



T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**SEZGİSEL ALGORİTMALAR YARDIMIYLA SİNİR AĞLARINDAN KURAL
ÇIKARIMI**

Özlem KANGA

Prof. Dr. Erdal EMEL
(Danışman)

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA - 2014

TEZ ONAYI

Özlem KANGA tarafından hazırlanan “Sezgisel Algoritmalar Yardımıyla Sinir Ağlarından Kural Çıkarımı” adlı tez çalışması aşağıdaki jüri tarafından oy birliği ile Uludağ Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Prof. Dr. Erdal EMEL

Başkan : Prof. Dr. Erdal EMEL

Üye : Doç. Dr. Barış SELÇUK

Üye : Yrd. Doç. Dr. Mehmet AKANSEL

Yukarıdaki sonucu onaylarım

Prof. Dr. Ali Osman DEMİR
Enstitü Müdürü

/ /

U.Ü Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

10/06/2014

Özlem KANGA

ÖZET

Yüksek Lisans Tezi

SEZGİSEL ALGORİTMALAR YARDIMIYLA SİNİR AĞLARINDAN KURAL ÇIKARIMI

Özlem KANGA

Uludağ Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Erdal EMEL

Günümüz bilgi çağında, veri madenciliği olarak da bilinen büyük miktardaki ham veriden mantıklı ve anlaşılabilir sonuçlar çıkarabilmek, önemli bir rekabet üstünlüğü yaratmaktadır. Yapay sinir ağları veri madenciliğinin bir dalı olan sınıflandırma alanında hatasız ya da yakın sonuçlar verdiği için, sıkça kullanılan tekniklerden biridir. Ancak yapay sinir ağlarının kara kutu özelliği, bu tekniğin veriden açık ve net sınıflandırma kuralları çıkarma konusunda yetersiz kalmasına sebep olmaktadır. Bu çalışma eğitilmiş yapay sinir ağlarından, ikili yapay arı kolonisi tabanlı olasılık eşik değerli (İYAKO) algoritması yardımıyla, verinin tümüne genellenebilen ve doğru kurallar çıkaran bir algoritma önermektedir. Yapay arı kolonisi algoritması bal arısı kolonilerinin akıllı besin arama davranışlarından esinlenerek geliştirilmiş sürü tabanlı meta-sezgisel yöntemlerdendir. Orijinal yapay arı kolonisi algoritması, ağırlıklı olarak sürekli optimizasyon problemleri için kullanılırken, İYAKO 0-1 tam sayılı programlama problemleri için geliştirilmiştir. Çok katmanlı yapay sinir ağı modeli, ikili girdi vektörlerine karşılık gelen çıktı vektörleri üzerinden eğitilmiş ve katmanlardaki ağırlıklar elde edilmiştir. Bu ağırlıklar kullanılarak çıkış katmanındaki transfer fonksiyonundan hesaplanan değer, daha sonra İYAKO algoritmasının amaç fonksiyonu olarak kullanılmıştır. Çıkarılan kurallar üzerinde kümeleme analizi uygulanarak kural sayısı azaltılmış ve veri sınıflarını temsil eden az sayıda anlamlı kurallar bulunmuştur. Elde edilen sonuçlar literatürdeki sınıflandırma karşılaştırma veri setleri üzerinde denenmiş, çeşitli makine-öğrenimi ve kural-tabanlı algoritmalar ile karşılaştırılmıştır. Önerilen algoritmanın gösterdiği performans, kural çıkarımı için alternatif bir çözüm yöntemi olduğunu göstermektedir.

Anahtar Kelimeler: Kural çıkarımı, veri madenciliği, yapay arı kolonisi algoritması, yapay sinir ağları

2014, xiv + 107 sayfa

ABSTRACT

MSc Thesis

NEURAL NETWORK RULE EXTRACTION USING HEURISTIC METHODS

Özlem KANGA

Uludağ University

Graduate School of Natural and Applied Sciences

Department of Industrial Engineering

Supervisor: Prof. Dr. Erdal EMEL

In today's information age, data mining, which is considered as a tool for drawing logical and understandable conclusions from large amounts of raw data, creates a significant competitive advantage. Artificial neural network classification, which is one of the commonly used techniques for data mining, gives close to flawless results in the field. However, black-box nature of artificial neural networks makes them insufficient when it comes to extracting clear and comprehensible rules from large databases. In this study a binary artificial bee colony based probability threshold (BABCP) algorithm is proposed to extract generalized accurate rules from artificial neural network weights that are trained on large databases. Artificial bee colony algorithm is a swarm-based meta-heuristic method, inspired by intelligent foraging behavior of honey bee colonies. The original artificial bee colony algorithm is mainly used for continuous optimization problems, whereas the proposed BABCP is for 0-1 integer programming problems. Network weights, obtained from trained artificial neural networks, are then used for maximizing the objective function value of the BABCP. The objective function used by the BABCP is the output activation function of the artificial neural network. A cluster analysis is applied to the rules extracted by the BABCP to reduce the number of rules and also to find more significant rules representing different classes of the data set. The results, obtained from applying the suggested algorithm on several classification data sets in the literature, are compared with various machine-learning and rule-based algorithms. The performance of the proposed algorithm shows that BABCP is an alternative solution method for rule extraction.

Key words: Rule extraction, Data mining, Artificial Bee Colony, Artificial Neural Networks

2014, xiv + 107 pages

TEŞEKKÜR

Yüksek Lisans eğitimim sırasında ve tez boyunca, bana tecrübelerini aktarırken ve zamanını paylaşırken göstermiş olduğu sabır, hoşgörü ve çaba dolayısıyla değerli ve saygıdeğer danışman hocam Prof. Dr. Erdal EMEL'e

Birlikte çalışmaktan mutluluk duyduğum asistan arkadaşlarıma,

Beni bugünlere getiren aileme teşekkürlerimi sunarım.

Özlem KANGA
10/06/2014

İÇİNDEKİLER

	Sayfa
TEZ ONAYI.....	i
ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGELER VE KISALTMALAR DİZİNİ	vi
ŞEKİLLER DİZİNİ.....	viii
ÇİZELGELER DİZİNİ	xii
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	3
2.1. Veri Madenciliği	3
2.1.2. Veri madenciliği yöntemleri	5
2.2.1. Yapay sinir ağı (YSA).....	10
2.2.2. YSA yapısı	10
2.2.3. YSA aktivasyon fonksiyonları	12
2.2.4. YSA türleri	14
2.2.5. Karar ağaçları	16
2.2.6. C4.5 karar ağacı algoritması	19
2.2.7. NBTree karar ağacı algoritması	20
2.2.8. Kural kümeleri	20
2.2.9. DecisionTable (“KararTablosu”) algoritması	22
2.2.10. PART algoritması	23
2.3. YSA, Karar Ağaçları ve Kural Kümelerinin Veri Madenciliği Açısından Karşılaştırılması	24
2.4. Meta-sezgisel Optimizasyon	26
2.4.1. Karınca kolonilerinin yiyecek bulma davranışları	28
2.4.2. Turlayan karınca kolonisi optimizasyon (TKKO) algoritması	28
2.4.3. Arı kolonilerinin yiyecek bulma davranışları	32
2.4.4. Yapay arı kolonisi (YAK) algoritması	34
2.4.5. İkili değişkenler için YAK kullanımı.....	37
2.4.6. Yapay arı kolonisinin ikili hali (<i>BABC</i>).....	38
2.4.7. İkili yapay arı kolonisi (<i>binABC</i>).....	39
2.4.8. Kaynak Özeti.....	39
3. MATERYAL VE YÖNTEM	41
3.1. Problemin Tanımı	41
3.2. YSA Eğitimi.....	43
3.3. İkili YAK Tabanlı Olasılık Eşik Değerli Algoritma	44
3.4. Kümeleme Analizi	51
4. UYGULAMA VE BULGULAR	53
4.1. Kullanılan Veri Kümeleri.....	53
4.2. Kural Çıkarımı Sonuçları	53
4.3. Kümeleme Analizi Sonuçları	57

5. TARTIŞMA VE SONUÇ	61
KAYNAKLAR	62
EKLER.....	68
EK 1 Kural Çıkarımı için Kullanılan Veri Kümeleri	69
EK 2 İkili Hale Dönüştürülmüş Veri Kümeleri	74
EK 3 STATISTICA Programı ile Eğitilen YSA Performans Tablosu.....	78
EK 4 MATLAB YSA Eğitimi Performans Grafikleri ve Tabloları	79
EK 5 Veri Kümesi Bazında MATLAB’de Eğitilmiş YSA Hata Matrisleri	87
EK 6 İkili Değer Alan Test Fonksiyonları	90
EK 7 Veri Kümeleri için Kümeleme Analizi Sonuçları	94
ÖZGEÇMİŞ	107

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler	Açıklama
P	Olasılık
τ	Karıncalar için feromon miktarı
ρ	Karınca feromonu için buharlaşma parametresi
x	İkili girdi vektör parametresi
v	Yapay arı kolonisi için ikili yeni kaynak parametresi
λ	İkili çıktı vektör parametresi
σ	Öznitelik parametrelerinin 1 olma olasılığı

Kisaltmalar	Açıklama
YSA	Yapay sinir ağı
ID3	Iterative dichotomiser 3 (Yinelemeli dikotomikleştirici 3)
NBTree	Naive Bayes tree (Naive Bayes ağacı)
DecisionTable	Decision table (Karar tablosu)
PART	Projective adaptive resonance theory (Yansıtımlı uyarlanabilir yankılaşım teorisi)
ART	Adaptive resonance theory (Uyarlanabilir yankılaşım teorisi)
GA	İşçi arı
K	Kâşif arı
G	Gözcü arı
DK	Deneyimli kâşif arı
DG	Deneyimli gözcü arı
YAK	Yapay arı kolonisi
BABC	Binary version of artificial bee colony (Yapay arı kolonisinin ikili hali)
binABC	Binary artificial bee colony (İkili yapay arı kolonisi)
A1	Sinir ağı girdi → gizli katman ağırlıkları
A2	Sinir ağı gizli → çıktı katman ağırlıkları
İYAKO	İkili YAK tabanlı olasılık eşik değerli algoritma
MPL	Multilayer perceptron (Çok katmanlı yapay sinir ağı)

SOS	Sum of squares (Karelerin toplamı)
OHK	Ortalama hata karesi
BCW	Breast cancer (Göğüs kanseri) Wisconsin veri kümesi
Özdevinimli Öğrenme	Machine Learning
Obur seçim	Greedy selection

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1.	1995-2014 yılları arasında bu alanda ilk 10'da yer alan dergilerdeki veri madenciliği alanındaki yayın sayıları (Web of Science 2014).....	3
Şekil 2.2.	Veri madenciliği yöntemlerinin özellikleri	5
Şekil 2.3.	Son 10 yıldaki yayınlarda kullanılan veri madenciliği yöntemlerinin dağılımı (Web of Science 2014)	7
Şekil 2.4.	Son 10 yıldaki yayınlarda veri madenciliğinde kullanılan özdevinimli modellerin dağılımı (Web of Science 2014)	7
Şekil 2.5.	Biyolojik nöronun temel yapısı (Kakıcı 2009).....	11
Şekil 2.6.	Biyolojik sinir sisteminin çalışma süreci.....	11
Şekil 2.7.	Temel YSA yapısı	12
Şekil 2.8.	Farklı tipteki aktivasyon fonksiyonları (Priddy ve Keller 2005'ten değiştirilerek alınmıştır)	13
Şekil 2.9.	Tek katmanlı YSA örneği.....	15
Şekil 2.10.	Tek gizli katmanı olan çok katmanlı YSA örneği	16
Şekil 2.11.	Çizelge 2.3'deki lens kullanımı veri kümesini tanımlayan STATISTICA programı kullanılarak ID3 yöntemiyle elde edilmiş karar ağacı.....	19
Şekil 2.12.	Karar tablosunun görsel anlatımı (Kohavi 1995'ten değiştirilerek alınmıştır)	23
Şekil 2.13.	PART yapısı	24
Şekil 2.14.	Son 10 yıldaki yayınlarda veri madenciliğinde kullanılan özdevinimli modellerin kural çıkarımı alanındaki dağılımı (Web of Science 2014)	25
Şekil 2.15.	Karınca kolonisinin yiyecek kaynağı ve yuva arasındaki yolu seçme süreci (Dorigo ve ark. 1997).....	28
Şekil 2.16.	TKKO algoritmasındaki bir karıncanın bulduğu yapay yol ve aday çözüm.....	29
Şekil 2.17.	TKKO için akış diyagramı	31

Şekil 2.18.	Balarısı kolonisinin yiyecek arama süreci (Baykasoglu ve ark. 2007'den değiştirilerek alınmıştır)	33
Şekil 2.19.	YAK için akış diyagramı.....	36
Şekil 3.1.	YSA girdi çıktı ilişkisi.....	41
Şekil 3.2.	İYAKO sınıflandırıcı için akış diyagramı	50
Şekil 3.3.	3 sınıflı bir kümeleme analizi örneği.....	51
Şekil EK 3.1.	STATISTICA programı ile eğitilen YSA performans tablosu.....	78
Şekil EK 4.1.	BCW veri kümesi MATLAB YSA eğitimi için ortalama hata karesi	79
Şekil EK 4.2.	BCW veri kümesi MATLAB YSA eğitimi için validasyon hatası	79
Şekil EK 4.3.	Iris veri kümesi MATLAB YSA eğitimi için ortalama hata karesi.....	80
Şekil EK 4.4.	Iris veri kümesi MATLAB YSA eğitimi için validasyon hatası	80
Şekil EK 4.5.	Monk veri kümesi MATLAB YSA eğitimi için ortalama hata karesi	81
Şekil EK 4.6.	Monk veri kümesi MATLAB YSA eğitimi için validasyon hatası	81
Şekil EK 4.7.	Nursery veri kümesi MATLAB YSA eğitimi için ortalama hata karesi	82
Şekil EK 4.8.	Nursery veri kümesi MATLAB YSA eğitimi için validasyon hatası.....	82
Şekil EK 4.9.	Sunburn veri kümesi MATLAB YSA eğitimi için ortalama hata karesi	83
Şekil EK 4.10.	Sunburn veri kümesi MATLAB YSA eğitimi için validasyon hatası.....	83
Şekil EK 4.11.	Titanic veri kümesi MATLAB YSA eğitimi için ortalama hata karesi	84

Şekil EK 4.12.	Titanic veri kümesi MATLAB YSA eğitimi için validasyon hatası	84
Şekil EK 4.13.	Weather veri kümesi MATLAB YSA eğitimi için ortalama hata karesi	85
Şekil EK 4.14.	Weather veri kümesi MATLAB YSA eğitimi için validasyon hatası.....	85
Şekil EK 4.15.	Wine veri kümesi MATLAB YSA eğitimi için ortalama hata karesi	86
Şekil EK 4.16.	Wine veri kümesi MATLAB YSA eğitimi için validasyon hatası	86
Şekil EK 6.1.	Ackley fonksiyonunun MATLAB programı ile çizimi	90
Şekil EK 6.2.	Bohachevsky fonksiyonunun MATLAB programı ile çizimi	91
Şekil EK 6.3.	Griewank fonksiyonunun MATLAB programı ile çizimi	91
Şekil EK 6.4.	Rastrigin fonksiyonunun MATLAB programı ile çizimi	92
Şekil EK 6.5.	Rosenbrock fonksiyonunun MATLAB programı ile çizimi.....	92
Şekil EK 6.6.	Sphere fonksiyonunun MATLAB programı ile çizimi.....	93
Şekil EK 7.1.	BCW veri kümesi için kural bazında iyicil sınıf doğruluk oranı	95
Şekil EK 7.2.	BCW veri kümesi için kural bazında kötücül sınıf doğruluk oranı	95
Şekil EK 7.3.	Iris veri kümesi için kural bazında tüm sınıf kümeleri	97
Şekil EK 7.4.	Monk veri kümesi için kural bazında istenen sınıf doğruluk oranı	98
Şekil EK 7.5.	Monk veri kümesi için kural bazında istenen sınıf doğruluk oranı 2	98
Şekil EK 7.6.	Nursery veri kümesi için kural bazında önerilemez sınıf doğruluk oranı.....	99
Şekil EK 7.7.	Nursery veri kümesi için kural bazında öncelikli sınıf doğruluk oranı.....	100

Şekil EK 7.8.	Nursery veri kümesi için kural bazında özel öncelikli sınıf doğruluk oranı.....	100
Şekil EK 7.9.	Sunburn veri kümesi için kural bazında güneş yanığı olur sınıf doğruluk oranı.....	101
Şekil EK 7.10.	Sunburn veri kümesi için kural bazında güneş yanığı olmaz sınıf doğruluk oranı.....	101
Şekil EK 7.11.	Titanic veri kümesi için kural bazında boğulur sınıf doğruluk oranı.....	102
Şekil EK 7.12.	Titanic veri kümesi için kural bazında boğulmaz sınıf doğruluk oranı.....	103
Şekil EK 7.13.	Weather veri kümesi için kural bazında tenis oynanır sınıf doğruluk oranı.....	104
Şekil EK 7.14.	Wine veri kümesi için kural bazında sınıf 1 doğruluk oranı.....	105
Şekil EK 7.15.	Wine veri kümesi için kural bazında sınıf 2 doğruluk oranı.....	105
Şekil EK 7.16.	Wine veri kümesi için kural bazında sınıf 3 doğruluk oranı.....	106

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 2.1.	YSA tarihçesi.....	10
Çizelge 2.2.	Sinir ağları terminolojisi (Mehrotra ve ark. 1997'den değiştirilerek alınmıştır)	12
Çizelge 2.3.	Örnek veri kümesi: lens kullanımı (https://archive.ics.uci.edu/ml/datasets/Lenses)	18
Çizelge 2.4.	Lens kullanımı veri kümesini tanımlayan kural kümesi.....	22
Çizelge 2.5.	Lens kullanımı veri kümesinin özet kural kümesi.....	22
Çizelge 2.6.	YSA'nın veri madenciliğinde avantaj ve dezavantajları	26
Çizelge 2.7.	YSA ve meta-sezgisel algoritmaların hibritlendiği literatür çalışmaları.....	26
Çizelge 2.8.	Kaynak özeti.....	40
Çizelge 4.1.	İYAKO kontrol parametre değerleri	54
Çizelge 4.2.	İYAKO performans değerlendirmesi	54
Çizelge 4.3.	İYAKO ve özdevinimli öğrenme yöntemlerinin performans hipotezlerinin p- değerleri	57
Çizelge 4.4.	Özdevinimli öğrenme yöntemleri ve İYAKO ile kümeleme kural çıkarımı sonuçları.....	58
Çizelge 4.5.	Kullanılan veri kümelerinin ana bileşen analizi	59
Çizelge 4.6.	İYAKO kümeleyici ve özdevinimli öğrenme yöntemlerinin performans hipotezlerinin p- değerleri	60
Çizelge EK 1.1.	BCW veri kümesi için öznitelik özellikleri.....	69
Çizelge EK 1.2.	Iris veri kümesi için öznitelik özellikleri.....	70
Çizelge EK 1.3.	Monk veri kümesi için öznitelik özellikleri	70
Çizelge EK 1.4.	Nursery veri kümesi için öznitelik özellikleri	71
Çizelge EK 1.5.	Sunburn veri kümesi için öznitelik özellikleri	72
Çizelge EK 1.6.	Titanic veri kümesi için öznitelik özellikleri.....	72

Çizelge EK 1.7.	Weather veri kümesi için öznitelik özellikleri	73
Çizelge EK 1.8.	Wine veri kümesi için öznitelik özellikleri	73
Çizelge EK 2.1.	BCW veri kümesinin ikili hali.....	74
Çizelge EK 2.2.	Iris veri kümesinin ikili hali	74
Çizelge EK 2.3.	Monk veri kümesinin ikili hali	75
Çizelge EK 2.4.	Nursery veri kümesinin ikili hali.....	75
Çizelge EK 2.5.	Sunburn veri kümesinin ikili hali	76
Çizelge EK 2.6.	Titanic veri kümesinin ikili hali	76
Çizelge EK 2.7.	Weather veri kümesinin ikili hali	77
Çizelge EK 2.8.	Wine veri kümesinin ikili hali	77
Çizelge EK 4.1.	BCW veri kümesi MATLAB YSA eğitimi sonuçları	79
Çizelge EK 4.2.	Iris veri kümesi MATLAB YSA eğitimi sonuçları	80
Çizelge EK 4.3.	Monk veri kümesi MATLAB YSA eğitimi sonuçları.....	81
Çizelge EK 4.4.	Nursery veri kümesi MATLAB YSA eğitimi sonuçları	82
Çizelge EK 4.5.	Sunburn veri kümesi MATLAB YSA eğitimi sonuçları.....	83
Çizelge EK 4.6.	Titanic veri kümesi MATLAB YSA eğitimi sonuçları	84
Çizelge EK 4.7.	Weather veri kümesi MATLAB YSA eğitimi sonuçları.....	85
Çizelge EK 4.8.	Wine veri kümesi MATLAB YSA eğitimi sonuçları.....	86
Çizelge EK 5.1.	BCW veri kümesi için YSA hata matrisi	87
Çizelge EK 5.2.	Iris veri kümesi için YSA hata matrisi	87
Çizelge EK 5.3.	Monk veri kümesi için YSA hata matrisi.....	87
Çizelge EK 5.4.	Nursery veri kümesi için YSA hata matrisi.....	88
Çizelge EK 5.5.	Sunburn veri kümesi için YSA hata matrisi.....	88
Çizelge EK 5.6.	Titanic veri kümesi için YSA hata matrisi	88

Çizelge EK 5.7.	Weather veri kümesi için YSA hata matrisi.....	89
Çizelge EK 5.8.	Wine veri kümesi için YSA hata matrisi.....	89
Çizelge EK 7.1.	BCW veri kümesi için anlaşılabilir kurallar.....	94
Çizelge EK 7.2.	BCW veri kümesi için kural bazında % doğruluk.....	96
Çizelge EK 7.3.	Iris veri kümesi için anlaşılabilir kurallar.....	96
Çizelge EK 7.4.	Iris veri kümesi için kural bazında % doğruluk	97
Çizelge EK 7.5.	Monk veri kümesi için anlaşılabilir kurallar	98
Çizelge EK 7.6.	Monk veri kümesi için kural bazında % doğruluk	99
Çizelge EK 7.7.	Nursery veri kümesi için anlaşılabilir kurallar	99
Çizelge EK 7.8.	Nursery veri kümesi için kural bazında % doğruluk	100
Çizelge EK 7.9.	Sunburn veri kümesi için anlaşılabilir kurallar	101
Çizelge EK 7.10.	Sunburn veri kümesi için kural bazında % doğruluk	101
Çizelge EK 7.11.	Titanic veri kümesi için anlaşılabilir kurallar.....	102
Çizelge EK 7.12.	Titanic veri kümesi için kural bazında % doğruluk	103
Çizelge EK 7.13.	Weather veri kümesi için anlaşılabilir kurallar	103
Çizelge EK 7.14.	Weather veri kümesi için kural bazında % doğruluk	104
Çizelge EK 7.15.	Wine veri kümesi için anlaşılabilir kurallar	104
Çizelge EK 7.16.	Wine veri kümesi için kural bazında % doğruluk	106

1. GİRİŞ

Bilgi teknolojilerinin gelişmesi ve internetin hız kazanması sonucunda, toplanan veri sayısı artmış ve veri depolama alanları genişlemiştir. . Bilgi teknolojisi sistemi, internet sayesinde hız kazanan veri toplama, veri tabanı oluşturma ve veri yönetimi dallarında geliştikçe, elindeki veriyi yararlı hale getirecek yollar aramaya başlamıştır. Veri madenciliği bilgi teknolojisinin geçirdiği evrimin bir sonucu olarak görülebilir. Hızla büyüyen veri depolarında saklanan ham veri, insanların algılama yeteneğinin sınırlarının dışına taşmış ve çözümlenebilmesi için güçlü araçlara ihtiyaç duyma durumu ortaya çıkmıştır. Jiawei ve ark. (2006) tarafından “veri zengini ancak bilgi fakiri durumların kurtarıcısı” olarak tanımlanan veri madenciliği de tam bu noktada devreye girer. Veri madenciliği araçları, saklı bilgileri ortaya çıkaracak veri analizini gerçekleştirerek, kurumlar için stratejik kararlarda, müşteri portföyü oluşturmada, bilimsel ve tıbbi araştırmalarda büyük katkı sağlayan destek mekanizmaları haline gelmiştir.

Veri madenciliği son yıllarda büyük miktardaki veri birikiminin getirdiği gereksinim nedeniyle enformasyon endüstrisi gibi işi bilgiyi işlemek olan sektörlerin oldukça ilgisini çeken bir alan haline gelmiştir. Karmaşık ve çok miktardaki ham veriden anlamlı ve aynı zamanda da yararlı bilgiye ulaşma işine kısaca veri madenciliği adı verilir. Kantardziec (2003) veri madenciliğini, “verinin analiz edilerek karar verme aşamasına yardımcı olacak bilgiyi çıkarma işi” olarak açıklamıştır. Veri madenciliğinde kullanılan veri ve bu veriden çıkarılan bilgi kavramlarının ayrıştırılmasında fayda vardır. Veri genellikle anketler, hesaplamalar veya deneylerle elde edilen rakamlar veya kavramlar bütünüdür. Bilgi ise bu verileri temel olarak kullanarak bir yargıya varma durumudur. Veri madenciliği sayesinde kazanılan bilgi, özellikle çok fazla veri toplamaya dayalı, müşteri ilişkileri, piyasa analizi ve üretim kontrolü gibi sektör uygulamaları için kullanılabilir. Kazanılan bilgi deneyimle birleşerek tecrübeye dönüşür bu da rekabet ve gücün önem arz ettiği piyasalarda kuvvetli bir avantaj olarak ortaya çıkar.

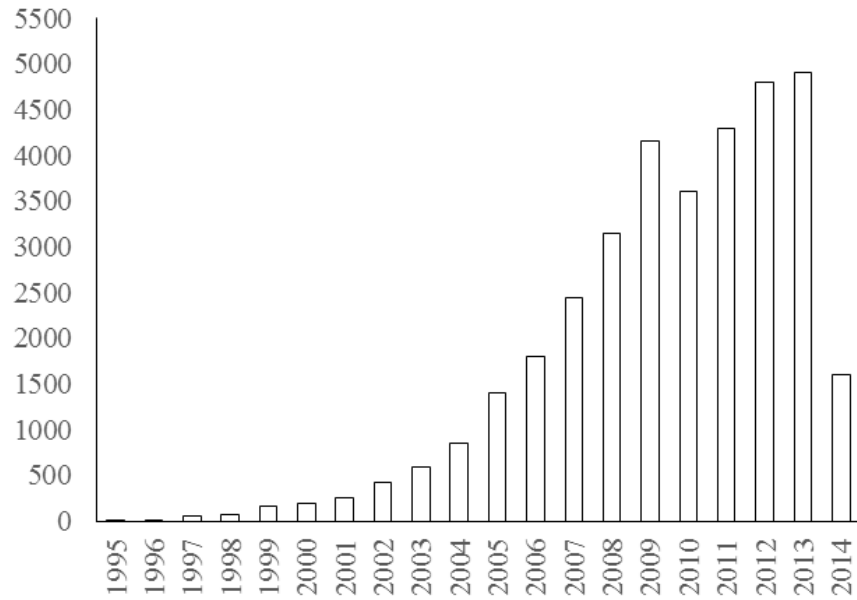
Bu çalışmanın amacı, geniş veri depolarında biriken, adı “Big Data” (Büyük Veri) olarak geçen, kullanıcı açısından yorumlanması güç verileri, anlamlı ve anlaşılabilir hale getirmektir. Bu çalışmada, veri kümelerinden anlamlı sınıflandırma kuralları çıkaracak

ve veri yapısı hakkında bilgi verecek bir yöntem önerilmektedir. Önerilen yöntemde, yapay sinir ağı temeline dayanan bir kural çıkarımı algoritması verilmektedir. Yapay sinir ağı veri kümesi üzerinde eğitildikten sonra, kullanıcı için anlaşılabilir haldeki “eğer-ise” kurallarını elde etmek için meta-sezgisel algoritmalarından yararlanan bir kural çıkarımı yöntemi uygulanmaktadır. Kural çıkarımından sonra ayrıca veri kümelerinin kendi içinde gösterdiği ilişkileri ve sınıflandırmada önemli olan bileşenleri ortaya çıkarmak amacıyla kümeleme analizi uygulanmıştır. Önerilen yöntem, literatürde sıklıkla kullanılan veri madenciliği yöntemlerinden karar ağaçları ve kural kümeleriyle, sekiz adet kıyas veri kümesi kullanılarak karşılaştırılmıştır.

2. KAYNAK ARAŞTIRMASI

2.1. Veri Madenciliği

Veri madenciliğinin önemi literatürde son 20 yıldaki veri madenciliği üzerine yapılan çalışmaların gösterdiği artan eğilimden de anlaşılabilir (Şekil 2.1). Veri madenciliğinin öneminin artmasıyla birlikte veri madenciliği uygulama alanları ve kullanılan araçların çeşitliliği de artmıştır. Geniş bir alana yayılan veri madenciliğine ilişkin detaylı bilgi bu bölümde verilecektir.



Şekil 2.1. 1995-2014 yılları arasında bu alanda ilk 10'da yer alan dergilerdeki veri madenciliği alanındaki yayın sayıları (Web of Science 2014)

2.1.1. Veri madenciliği aşamaları

Veri madenciliği yedi aşamada tanımlanabilir (Jiawei ve ark. 2006):

- (1) Veri temizliği (Gürültü ve tutarsız verilerin çıkarılması)
- (2) Veri entegrasyonu (Çoklu veri kaynaklarının birleştirilmesi)
- (3) Veri seçimi (Analize katkı sağlayacak gerekli verinin ayrıştırılması)

- (4) Veri dönüşümü (Verinin kullanıma uygun formata çevrilmesi)
- (5) Veri analizi (Akıllı metotlar kullanarak veriden kalıplar çıkarılması)
- (6) Veri kalıplarının incelenmesi (Veriyi gerçekten temsil eden kalıpların çeşitli ölçümlerle belirlenmesi)
- (7) Bilginin sunumu (Kullanıcının anlayacağı şekilde bilginin aktarılması)

İlk aşama olan “veri temizliği” aşamasında verinin güvenilirliğini arttırılır. Bu aşama eksik verinin değerlendirilmesi gerekirse tahmin edilmesi, gürültü ve aykırı değerlerin ortadan kaldırılması gibi önlemler içerir. Aykırı değerler, diğer gözlemlerle tutarlı olmayan değerlerdir. Aykırı değerler ölçüm hataları, kodlama ve hatalı kayıtlar sonucu bazen doğal bazen anormal şekilde oluşur. Aykırı değerlerin temizlenmemesi daha sonraki aşamalarda üretilcek modeli etkileyebilir. Aykırı değerlerle başa çıkmanın temizleme dışında diğer bir yolu ise daha sonraki aşamalarda üretilcek olan modelin aykırı değerlere karşı duyarsız kalacak sağlam yöntemlerle modellenmesi olabilir. Kullanıcı veri temizliğinin sonuçları ne kadar etkileyeceğine bağlı olarak bu aşamayı kullanıp kullanmamakta özgürdür. (Maimon ve ark. 2010).

Veri madenciliği aşamalardan en önemlisi veri analizi olarak değerlendirilebilir. Veri analizi yöntemleri tanımlayıcı ve tahmin yürütücü olmak üzere iki kategoriye ayrılır (Jiawei ve ark. 2006). Tanımlayıcı yöntemler verinin genel özelliklerini karakterize eder. Tahmin yürütücü yöntemler ise güncel veriyi bilinmeyen durumlar için öngörü yapmada kullanır. Bazı durumlarda kullanıcının verinin özellikleri hakkında hiçbir bilgisi olmayabilir. Bu tip durumlarda kullanıcı hem açıklayıcı hem de tahmin yürütücü yöntemleri eş zamanlı olarak kullanmak isteyebilir. Veri madenciliği sistemleri kullanıcıya istediği yöntemi seçme esnekliğini tanıyacak yapıda olmalıdır. Veri madenciliği sistemleri aynı zamanda kullanıcılara kılavuz görevi görerek verinin içerdiği ilginç kalıplar için ipuçları vermelidir. Veri tabanındaki bütün veriler için geçerli olmayabilen bu ilginç kalıpları kullanıcı belirleyeceği güvenilirlik seviyesine göre dikkate alıp almamayı seçer.

2.1.2. Veri madenciliği yöntemleri

Veri madenciliğinin ana görevi, veri kümelerini kullanarak modeller oluşturmaktır (Carrier ve Povel 2003). Bu modeller, kullanıcıya veri kümesinin karakteristikleri hakkında kullanışlı bilgiler sunmayı amaçlar. Literatürde kullanılan veri madenciliği yöntemlerinin oluşturduğu modeller incelendiğinde beş ana başlıkta toplanabileceği gözlemlenmiştir (Ahmed 2004, Mitra ve ark. 2002, Carrier ve Povel 2003, Shaw ve ark. 2001):

- i. Sınıflandırma
- ii. Kümeleme
- iii. Birliktelik
- iv. Regresyon
- v. Zaman serileri

Veri madenciliği yöntemlerinin karakteristik özellikleri Şekil 2.2’de özetlenmiştir.

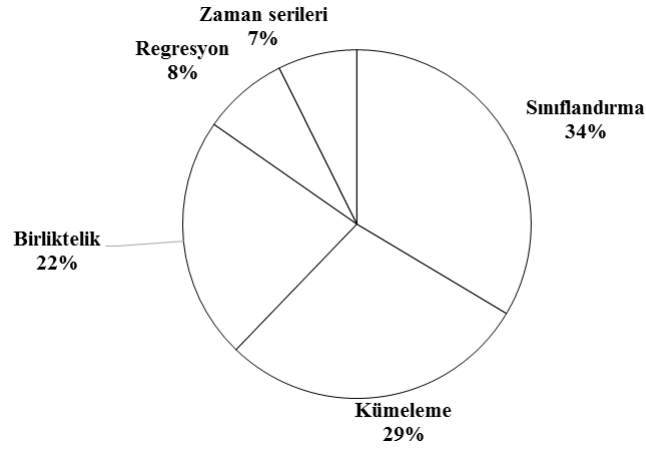


Şekil 2.2. Veri madenciliği yöntemlerinin özellikleri

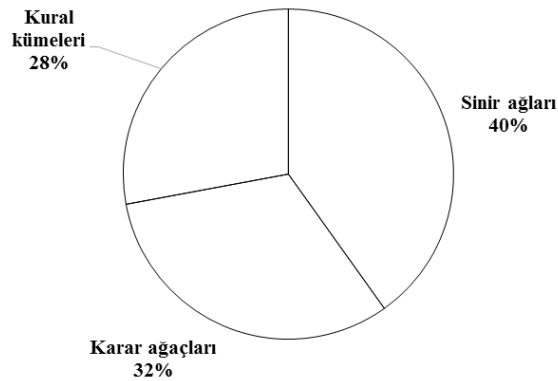
- Sınıflandırma: Sınıflandırma modellerinde, veri kümesinin belirli bir kısmı eğitim kalan kısmı da test için kullanılarak sınıflandırma kuralları oluşturulur. Test için kullanılan kısım sınıflandırma modelinin doğruluğunu değerlendirmede kullanılır. Sınıflandırma kurallarına “*Eğer İse*” kuralları da denir (Vijayarani ve Divya 2011). Sınıflandırma kuralları, önceden karşılaşılmayan bir durum ortaya çıktığında verilecek kararların belirlenmesinde yardımcı olur. Sınıflandırma kuralları kullanılarak, ortaya çıkan yeni durumun hangi veri sınıfına ait olduğu belirlenir (Hühn ve Hüllermeier, 2009). “*Doğrudan*” ve “*Dolaylı*” olmak üzere sınıflandırma kuralları iki şekilde çıkarılabilir. Doğrudan çıkarılan kurallar veri kümesinden çıkarılırken, Dolaylı çıkarılan kurallar diğer sınıflandırma modellerinden çıkarılır (Sasaki ve Kita 1998).
- Kümeleme: Kümeleme modellerinde amaç, karakteristikleri benzer verileri içeren gruplar oluşturarak birbirinden farklı özellikler gösteren kümeler yaratmaktır (Ahmed 2004). Kümeleme analizinin sınıflandırma analizinden farkı, küme özelliklerinin ve sayısının önceden bilinmiyor olmasıdır. Sınıflandırma analizinde veri kümesinin hangi sınıflardan oluştuğu bilinmekte ve buna göre eğitim yapılmaktadır.
- Birliktelik: Birliktelik modelleri, veri kümesindeki hangi olayların aynı anda gerçekleştiğine bakarak olaylar arasında ilişkiler kurar (Ahmed 2004, Mitra ve ark. 2002). Birliktelik modelleri özellikle pazarlama ve müşteri ilişkileri alanlarında sıkça karşılaşılan modellerdendir. Müşterilerin hangi ürünleri aynı anda satın aldıklarına bakılarak, hem finansal müşteri profili çıkaran, hem de müşterilere öneriler yapan birliktelik modelleri “*pazar sepeti analizi*” adı altında yaygın olarak kullanılmaktadır (Alpaydın 2000).
- Regresyon: Regresyon modelleri, veri kümesindeki her bir veriyi temsil edecek sürekli değerlerin hesaplanmasında kullanılan istatistiki tahmin yöntemlerindendir. Sınıflandırma ve birliktelik modellerinde veri kümesi genellikle kategorik haldeyken, regresyon modellerinde sürekli veri kümeleriyle çalışılır (Carrier ve Povel 2003, Mitra ve ark. 2002).

- Zaman serileri: Tahmin modelleri, veri kümesini zamana bağlı olarak inceleyerek gelecekteki verileri tahmin etmeye çalışır. Tahmin modelleri de regresyon modelleri gibi sürekli çıktı değerleriyle çalışır. Tahmin modellerinin en çok kullanıldığı alanların başında talep modelleri gelmektedir (Ahmed 2004, Linoff ve Berry 2004).

Veri madenciliğini modellerini oluşturmak için çok çeşitli özdevinimli öğrenme yöntemleri vardır. Kullanıcının bu yöntemlerden hangisini kullanacağı veri kümesindeki veri tipine ve elde edilmek istenen sonuçlara bağlıdır (Nigai ve ark. 2009).



Şekil 2.3. Son 10 yıldaki yayınlarda kullanılan veri madenciliği yöntemlerinin dağılımı (Web of Science 2014)



Şekil 2.4. Son 10 yıldaki yayınlarda veri madenciliğinde kullanılan özdevinimli modellerin dağılımı (Web of Science 2014)

Web of Science (2014)'e bakıldığında, veri madenciliği alanında yapılan yayınlarda yöntem olarak en çok sınıflandırma, özdevinimli öğrenme modeli olarak da en çok sinir ağları, daha sonra karar ağaçları ve kural kümeleri kullanılmıştır (Şekil 2.3 ve Şekil 2.4).

2.2. Özdevinimli Öğrenme ve Veri Madenciliği

Özdevinimli öğrenme ve veri madenciliği son yıllarda, büyük bilgi depolarındaki değerli bilgiyi ayıklama ihtiyacından kaynaklanan bilgisayar bilimleri ve veri analizi araştırma alanlarındaki hızlı büyüme ve gelişme sebebiyle, birbiriyle iç içe geçmiş iki konudur. Breiman (2001)'in da belirttiği üzere özdevinimli öğrenmenin gelişmesindeki en büyük etkenlerden biri istatistik alanındaki veri analizi yöntemlerine göre çok daha üstün sonuçlar vermesidir. İstatistik alanındaki yöntemler temel olarak lineer regresyon, lojistik regresyon ve “Cox” modeli gibi veri modellerine dayanır. Özdevinimli öğrenme ise karar ağaçları ve sinir ağları gibi algoritmik modelleri temel alır. Breiman (2001) yaptığı çalışmalar sonucunda algoritmik modellerin veri modellerine kıyasla daha iyi öngörü doğruluğu olduğunu ve girdilerle çıktılar arasındaki ilişkinin yapısına dair daha güvenli bilgiler verdiğini ortaya koymuştur. Girdilerle çıktılar arasındaki ilişki algoritmik modellerde insanlar tarafından anlaşılabilir yazılı dile dönüştürülebilirken, veri modellerinde karmaşık analitik bir şekilde anlatılır ki bu da gerçek hayat problemlerinde kişilerin veri ile ilgili basit ve kullanışlı öngörüler yapmasını zorlaştırır.

Mitchell (1997), özdevinimli öğrenmeyi kısaca verideki modelleri, örüntüleri ve düzensizlikleri keşfeden, gelişmiş ve iyi tanınan bir bilgisayar bilimi araştırma alanı olarak tanımlamıştır. Özdevinimli öğrenme yaklaşımları kabaca iki farklı grupta kategorize edilebilir (Fürnkranz ve ark. 2012):

- (1) Sembolik yaklaşımlar: Kurallar, karar ağaçları ve mantıksal temsiller gibi tümevarım öğrenimini kullanan yöntemler.
- (2) İstatistiksel yaklaşımlar: K-en yakın komşu, örnek tabanlı öğrenme, Bayes sınıflandırıcıları ve destek vektör makineleri gibi istatistiksel veya örüntü tanıyan yöntemler.

Yukarıda farklı iki kategoriye ayrılan veri madenciliğinin bir dalı olan özdevinimli öğrenme yaklaşımları temel olarak bir veri kümesi ile çalışırlar. Bu veri kümesi içinde öznitelikleri ve ait oldukları sınıf açısından analiz edilecek veriler bulunur. Verilerin öznitelikleri girdiyi, ait oldukları sınıf ise çıktıyı temsil eder. Özdevinimli öğrenme teknikleri bu girdi ve çıktılar üzerinden eğitilerek verinin ait olduğu sınıfı doğru olacak şekilde tespit etmeye çalışır. Bu eğitim iki farklı şekilde gerçekleştirilebilir (Kotsiantis ve ark. 2006):

(1) Denetimli öğrenme: Eğitim kümesinin sınıflandırılmış olduğunu varsayar. Denetimli öğrenmede örnekler daha önceki deneyimlerden alınır ve oluşturulan model ile yeni karşılaşılan bir örnek sınıflandırılmaya çalışılır.

(2) Denetimsiz öğrenme: Sınıflandırılmamış eğitim kümesi üzerinde çalışır.

Bu farklı tipteki öğrenme yöntemlerinden sonra özdevinimli öğrenme tekniği, veri madenciliğini iki farklı yoldan yapabilir (Fürnkranz ve ark. 2012):

(1) Tahmin edici veri madenciliği: Bu madencilik yöntemleri denetimli öğrenmeyi kullanır. Karar ağaçları ve kural kümeleri gibi model veya teoriler geliştirilerek, sınıflandırma ve öngörü için kullanılır. Karşılaşılmamış bir örneği veri kümesindeki sınıflardan birine atar.

(2) Tanımlayıcı veri madenciliği: Tanımlayıcı veri madenciliği yöntemleri tipik olarak denetimsiz öğrenme tekniğini kullanır. Bunlar sınıflandırılmamış veri kümesine ait olan örüntüleri ortaya çıkarırlar. Ortaya çıkan örüntüler genel olarak keşfedici veri çözümleme konularında kullanılır. Tahmin edici veri madenciliğiyle karşılaştırıldığında veriye bir sınıf etiketi atamaktansa onun öznitelikleri arasındaki ilişkileri ortaya çıkarmaya çalışır.

Tahmin veri madenciliği yöntemlerinden temel olanlar, sınıflandırma ve öngörü için model oluşturan yapay sinir ağları, karar ağaçları ve kural kümeleridir (Wu ve ark. 2008, Liao ve Wen 2007).

2.2.1. Yapay sinir ağı (YSA)

YSA sınıflandırıcıları veri madenciliği için çok önemli araçlardır. Son yıllarda, yeni YSA yapıları ve uygulamaları araştırmacılar tarafından üzerinde sıkça çalışılan konular haline gelmiştir. YSA'nın veri madenciliği yöntemlerinden olan sınıflandırma konusunda önem kazanması, teori ve uygulamalardan alınan başarılı sonuçlar sayesinde olmuştur (Wang ve Fu 2006).

YSA, insan beyninin çalışma prensibinden esinlenerek yaratılmıştır. İnsan beyninin tecrübe yoluyla öğrenme sonucu yeni bilgiler oluşturabilme ve keşfetme yeteneği bilgisayar sistemi haline dönüştürülmeye çalışılmıştır. YSA, insan beyninin bu öğrenme sürecini ve yapısını taklit ederek çalışır.

YSA ilk defa McCulloch ve Pitts tarafından 1943 yılında insan beyninden esinlenerek geliştirilmiş bir elektrik devresiyle modellenmiştir. YSA çalışmalarındaki önemli gelişmeler yıllarıyla birlikte Çizelge 2.1'de verilmiştir.

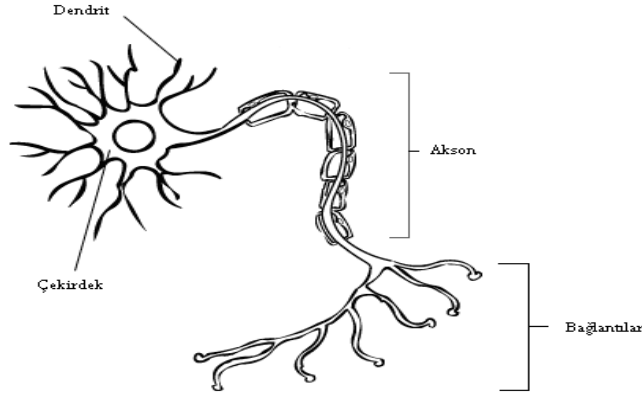
Çizelge 2.1. YSA tarihçesi

Yıl	Gelişim
1949	Hebb öğrenme kuralının bulunması ve öğrenme sürecinin bilgisayarlar tarafından yapılabilecek hale getirilmesi (Hebb 1949)
1958	Algılayıcı modelinin geliştirilmesi (Rosenblatt 1958)
1961	İki katmanlı ileri beslemeli algılayıcıların önerilmesi (Borsellino ve Gamba 1961)
1974	Çok katmanlı ağlarda geriye yayılım algoritmasının uygulanması (Werbos 1974)

2.2.2. YSA yapısı

YSA, biyolojik sinir sisteminden esinlendiğinden, yapısı da biyolojik sinir sistemine benzer. Biyolojik sinir sisteminin temel yapı taşı sinir hücreleri yani nöronlardır. Nöronun yapısı Şekil 2.5 ile gösterilmiştir. Nöronların uç kısmında bulunan ve kök yapısını andıran dendritlerin görevi, bağlı olduğu duyu organlarından veya diğer nöronlardan aldığı

sinyalleri çekirdeğe iletmektir. Çekirdeğe gelen bu sinyaller önce aksona, daha sonra da işlenerek nöronun diğer ucundaki bağlantı noktalarına yani sinapslara gönderilir. Sinapslar aldığı sinyalleri diğer nöronlara aktarır (Yegnanarayana 2009).



Şekil 2.5. Biyolojik nöronun temel yapısı (Kakıcı 2009)

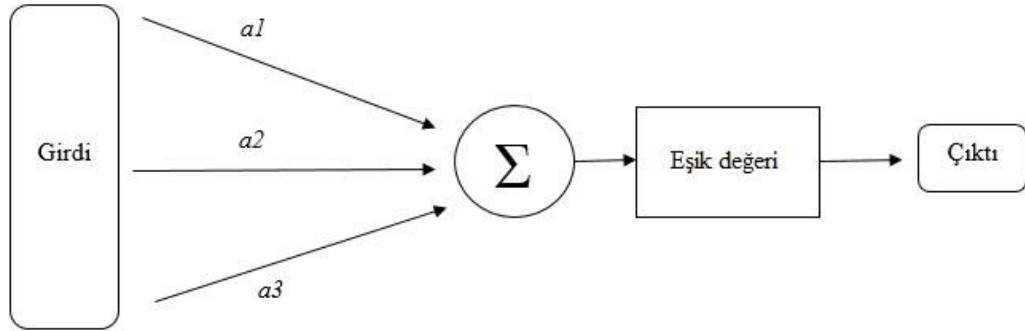
Bir insan beyinde ortalama 10 milyar nöron ve 60 trilyon nöronlar arası bağlantı olduğu tahmin edilmektedir. Duyu organlarından alıcı sinirler yoluyla edinilen bilgiler, taşıyıcı sinirler yardımıyla merkezi sinir sistemine aktarılırlar. Merkezi sinir sisteminde işlenen bilgiler, tepki sinirleri yoluyla organlara gönderilir. Canlı vücudunda oluşan etki tepki mekanizması Şekil 2.6 ile temsil edilmiştir (Yegnanarayana 2009).



Şekil 2.6. Biyolojik sinir sisteminin çalışma süreci

YSA'nın yapısı da nöronlar ve aralarındaki bağlantılardan oluşur. Nöronlar girdilerden aldıkları bilgiyi anlamlı bir sonuca dönüştürerek çıktı haline getirir. Bir sinir ağında, nöronlara gelen tüm bilgiler aynı önemi taşımayabilir. Mesela yüksek bir ses çıktığında kulaktan gelen bilginin burundan gelene göre çok daha fazla önemi vardır. Bu

yüzden nöronlara gelen bilgiler önem derecesine, yani “*sinaptik verimine*” göre, çıktıya katkı sağlar. YSA’da bu özellik, bağlantıların sahip oldukları ağırlıklar tanımlanmıştır.



Şekil 2.7. Temel YSA yapısı

Şekil 2.7 ile gösterilen YSA’da girdiler duyu organlarından gelen sinyalleri temsil eder. Nöronların aldıkları sinyalin önemi ve katkısına göre “*a*” ile gösterilen bağlantı ağırlıkları mevcuttur. Bu ağırlık değeri ve girdi sinyali çarpılarak toplama fonksiyonuna dâhil edilir. Eğer toplam değer eşik değerinden fazla ise sinyal iletilir, değilse iletilmez. İletilen sinyal en son olarak bir çıktı haline dönüşür (Braspenning ve ark. 1995). Biyolojik ve yapay sinir ağlarında kullanılan terminoloji Çizelge 2.2’de verilmiştir.

Çizelge 2.2. Sinir ağları terminolojisi (Mehrotra ve ark. 1997’den değiştirilerek alınmıştır)

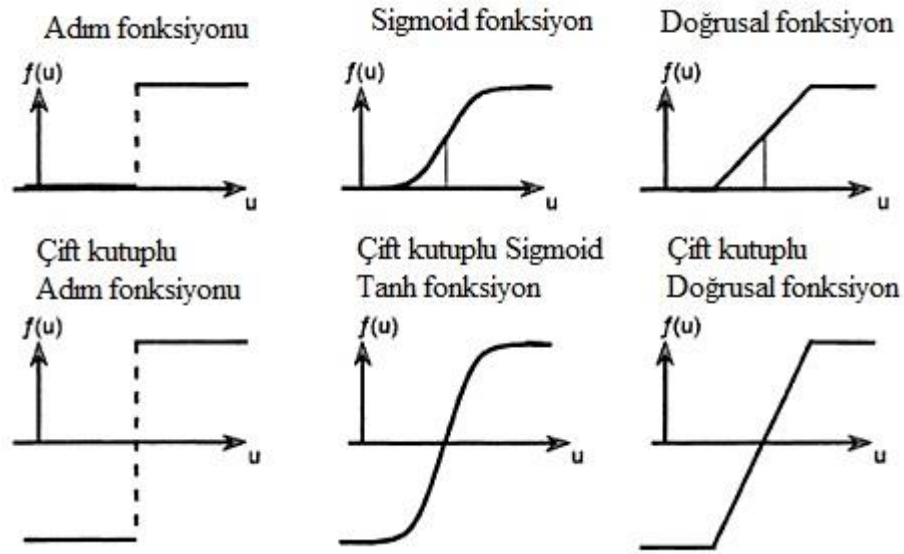
Biyolojik terminoloji	Yapay terminoloji
Nöron	Düğüm, Hücre
Sinaps	Bağlantı
Sinaptik verim	Bağlantı ağırlığı
Etki	Girdi
Tepki	Çıktı

2.2.3. YSA aktivasyon fonksiyonları

Toplama fonksiyonundan çıkan değer, YSA’nın çıktısını oluşturmak üzere aktivasyon fonksiyonuna iletilir. Aktivasyon fonksiyonu olarak farklı fonksiyonlar kullanılabilir. Kullanılacak aktivasyon fonksiyonu ağın yapısına bağlıdır. Örneğin geri beslemeli ağlarda aktivasyon fonksiyonunun türevi üzerinden de işlem yapıldığından, türevi kolay

hesaplanır fonksiyonlardan biri seçilir. YSA’da kullanılan fonksiyonlar Şekil 2.8’de özetlenmiştir.

- Adım fonksiyonu: Çıktı değeri olarak 0 veya 1 olmak üzere sadece 2 değer verir. Adım fonksiyonun çift kutuplu hali de çıktı değeri olarak 2 değer verir fakat bunlar orijinal fonksiyondan farklı olmak üzere -1 ve 1 değerleridir. Adım fonksiyonun verdiği değerler girdinin 0’dan büyük olup olmamasına bağlıdır.



Şekil 2.8. Farklı tipteki aktivasyon fonksiyonları (Priddy ve Keller 2005’ten değiştirilerek alınmıştır)

- Sigmoid fonksiyon: Türevi alınabilir bir fonksiyon olduğundan YSA yapılarında sıkça kullanılan bir aktivasyon fonksiyonudur. Sigmoid fonksiyonu (0,1) aralığında değerler üretir. Sigmoid fonksiyonu

$$f(u) = \frac{1}{1 + e^{-u}} \quad (2.1)$$

ile ifade edilir. Sigmoid fonksiyonun diğer bir çeşidi yine doğrusal olmayan tanjant hiperbolik (Tanh) fonksiyonudur. Tanh fonksiyonu (-1,1) aralığında değerler alır ve matematiksel ifadesi aşağıdaki gibidir:

$$f(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

- Doğrusal fonksiyon: Doğrusal problemleri çözmek amacıyla tercih edilir. E , doğrunun eğimini belirleyecek şekilde, E_x olarak ifade edilir.

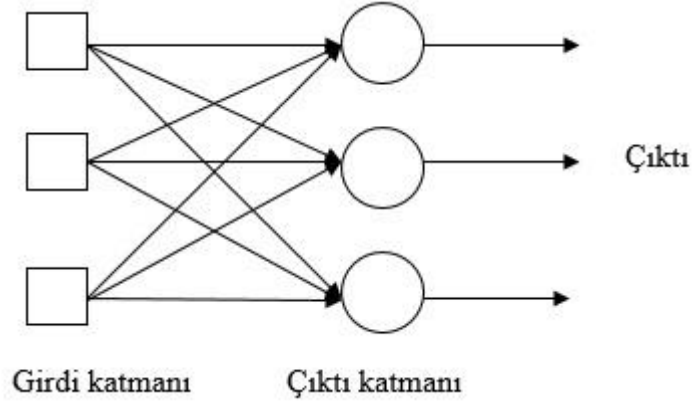
2.2.4. YSA türleri

Günümüzde farklı amaçlara ve alanlara yönelik çok çeşitli YSA modeli geliştirilmiştir. YSA türleri 3 ana başlıkta toplanabilir:

- i. Tek katmanlı YSA
- ii. Çok katmanlı YSA
- iii. Yinelenen YSA

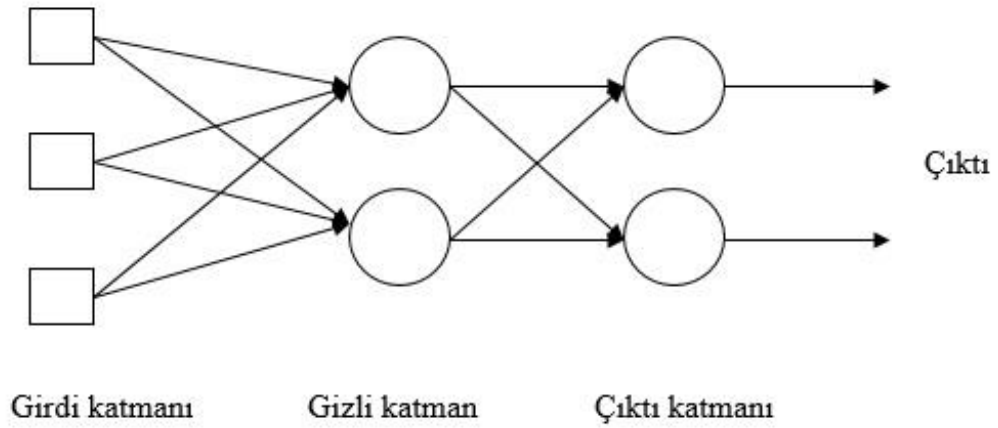
Tek katmanlı ve çok katmanlı YSA'lar ileri beslemeli, yinelenen YSA'lar ise geri beslemelidir. İleri beslemeli modellerde bilgi akışı girdi düğümlerinden çıktı düğümlerine doğru tek yönde olur. Geri beslemeli modellerin ileri beslemeli modellerden farklılaştığı nokta en az bir tane geri besleme döngüsünün olmasıdır. Geri besleme döngüsü sayesinde herhangi bir nöronun çıktıları diğer nöronların girdilerine beslenir. Geri besleme döngülerinin varlığı sinir ağının öğrenme kapasitesi ve performansı üzerinde oldukça etkilidir (Haykin 2009).

- Tek katmanlı YSA: Tek katmanlı YSA, katmanlı YSA'ların en basit halidir. Sadece girdi ve çıktı katmanlarından oluşur. Girdi katmanı eğitim bilgilerini alır, çıktı katmanı ise sonuçları verir. Şekil 2.9'da görülen 3 düğümlü bir tek katmanlı YSA yapısındaki tek kelimesi çıktı katmanını temsil eder, girdi katmanı katman sayısına katılmaz (Witten ve ark. 2005).



Şekil 2.9. Tek katmanlı YSA örneği

- Çok katmanlı YSA: Çok katmanlı YSA’larda girdi ve çıktı katmanlarının dışında bir de gizli katmanlar bulunur. Gizli katmanlarda yapılan işlemler gizli katman düğümleri tarafından gerçekleştirilir. Bu katmanlara gizli denmesinin nedeni, girdi veya çıktılar tarafından direkt olarak görülememelerindendir. Gizli katmanlar, girdi değerlerini çıktı değerleri için kullanışlı hale getirirler. Çok katmanlı YSA’larda her bir katmanın çıkışı sıradaki katmanın girdisi konumundadır. Böylece girdi, çıktıya ulaşana kadar gizli katmanlarda işlenir ve çıktıyı en iyileştirecek hale gelir. Gizli katman sayısı arttırıldıkça YSA lokal bir bakış açısı yerine global bir bakış açısı edinmeye başlar. Tek katmanlı YSA’larda girdiler çıktıları tek tek etkilerken, gizli katmanlar arttıkça girdilerin sadece önemli bölümleri çıktıları etkiler hale gelir (Churchland ve Sejnowski 1992). n katmanlı bir YSA, m girdi düğüm sayısı, h_i , i gizli katmanındaki düğüm sayısı ve k çıktı düğüm sayısı olacak şekilde $m-h_1-h_2-...-h_n-k$ ile gösterilir (Haykin 2009). Şekil 2.10’da, 3-2-2 ağı görülebilir.



Şekil 2.10. Tek gizli katmanlı olan çok katmanlı YSA örneği

2.2.5. Karar ağaçları

Özdevinimli öğrenme alanında, tek değişkenli karar ağacı yapısı kullanarak veri kümelerinden kurallara varma fikri ilk defa Hunt ve ark. (1966) tarafından ortaya atılmıştır. Daha sonra Quinlan (1979) *ID3* adı verilen ve günümüzde de halen yeni geliştirilen algoritmaların kıyaslanmasında kullanılan karar ağacı algoritmasını bulmuştur.

Karar ağacı yapısı bir dizi düğüm ve daldan oluşan bir sınıflandırma modelidir. Genel olarak her bir düğüm bir özniteliği, her bir düğümden çıkan dal da o düğümün temsil ettiği özniteliğin değerini gösterir. En üstte yer alan başlangıç düğümü kök, alttaki düğümler yaprak adını alır. Her yaprak bir sınıf ile eşleşir. Karar ağacı sınıflandırma için kullanılırken, sınıflandırılmak istenen örneğin özniteliklerine uygun değerler alan düğümlerden çıkan dallar yukarıdan aşağıya doğru takip edilir. Bu takip sonucunda sınıflandırılmak istenen örneğe, ulaşılan yapraktaki sınıf etiketi atanır. Bir karar ağacı en üstteki kökten başlayarak yukarıdan aşağıya doğru oluşturulur. Şekil 2.11, Çizelge 2.3 için bir karar ağacı örneğini göstermektedir. Karar ağacının en genel hali sadece kök olduğu haldir. Daha sonra bu kökten dallar çıkarılarak ağacın yaprakları oluşturulur ve bu yapraklar sayesinde veri kümesini tanımlayan genel durumdan özel durumlara geçiş yapılmış olunur. Sadece kökten oluşan ağaç tüm veri kümesi için çok genel kalır, bütün veriler için bir yaprak içeren ağaç ise çok özel kalır. Karar ağacını oluştururken göz önünde bulundurulması gereken temel prensip, ağaç yapısının veri kümesini belirli

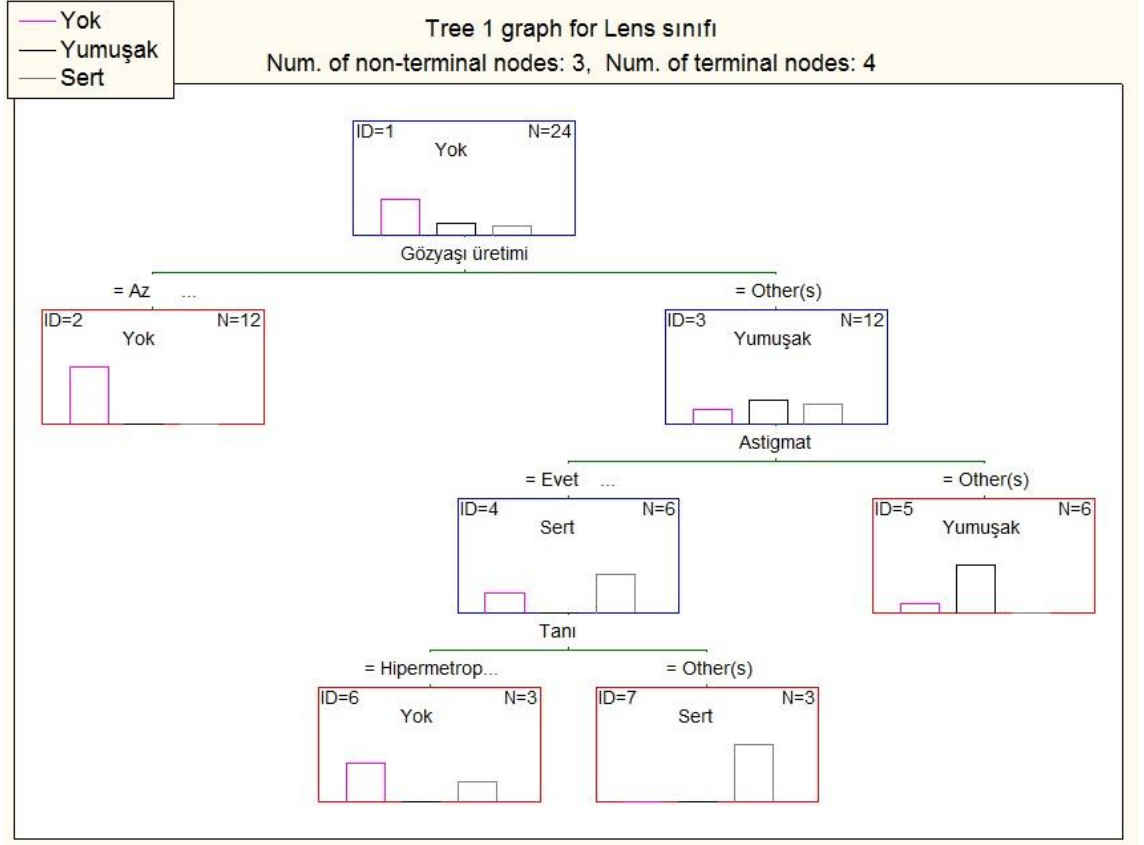
oranda genelleyerek karşılaşılmamış örnekler için yüksek öngörü doğruluğu sağlamasıdır.

Karar ağacını oluştururken en önemli adımlardan biri düğüm olarak seçilecek özniteliğe karar vermektir. Tipik öznitelik seçim ölçütü olarak düğümlerin “*saflığını*” ölçen fonksiyonlar kullanılır. Bu saflık derecesi düğümün tek bir sınıfa ait olan kaç tane veri içerdiğiyle doğrudan orantılıdır. Karar ağaçlarında özniteliklerin tümü bir düğüme atanmamış olabilir. Bu durum veri kümesinin bu özniteliğe bağlı kalınmadan sınıflandırılabilceğini gösterir. Örnek olarak Şekil 2.11’de, Çizelge 2.3 özniteliklerden biri olarak belirtilen yaş kullanılmamıştır. Bu da yaşı lens kullanımında çok etkili ölçütlerden biri olmadığını göstermektedir. Ayrıca, ağacın üst kısımlarındaki köke daha yakın olan düğümlere atanmış öznitelikler, hedef sınıf üzerinde daha güçlü bir etkiye sahiptir. Başka bir deyişle, bu düğümler daha fazla sayıdaki veriyi sınıflandırır. Ağacın alt kademelerine inildikçe her bir düğümdeki sınıflandırılmış örnek sayısı azalır. Eğer sınıflandırma için çok fazla düğüm kullanılırsa ortaya eğitim veri kümesini iyi sınıflandıran fakat karşılaşılmamış örnekleri sınıflandırmada başarısız kalan bir “*ezber*” ağacı ortaya çıkar.

C4.5 (Quinlan 1993) ve *NBTree* (Kohavi 1996) literatürde özellikle karşılaştırma algoritmaları olarak sıkça karşılaşılan karar ağacı algoritmalarındandır (Xindong ve ark. 2008).

Çizelge 2.3. Örnek veri kümesi: lens kullanımı
(<https://archive.ics.uci.edu/ml/datasets/Lenses>)

Örnek No.	Yaş	Tanı	Astigmat	Gözyaşı üretimi	Lens sınıfı
1	Genç	Miyop	Hayır	Az	Yok
2	Genç	Miyop	Hayır	Normal	Yumuşak
3	Genç	Miyop	Evet	Az	Yok
4	Genç	Miyop	Evet	Normal	Sert
5	Genç	Hipermetrop	Hayır	Az	Yok
6	Genç	Hipermetrop	Hayır	Normal	Yumuşak
7	Genç	Hipermetrop	Evet	Az	Yok
8	Genç	Hipermetrop	Evet	Normal	Sert
9	Ön-presbit	Miyop	Hayır	Az	Yok
10	Ön-presbit	Miyop	Hayır	Normal	Yumuşak
11	Ön-presbit	Miyop	Evet	Az	Yok
12	Ön-presbit	Miyop	Evet	Normal	Sert
13	Ön-presbit	Hipermetrop	Hayır	Az	Yok
14	Ön-presbit	Hipermetrop	Hayır	Normal	Yumuşak
15	Ön-presbit	Hipermetrop	Evet	Az	Yok
16	Ön-presbit	Hipermetrop	Evet	Normal	Yok
17	Presbit	Miyop	Hayır	Az	Yok
18	Presbit	Miyop	Hayır	Normal	Yok
19	Presbit	Miyop	Evet	Az	Yok
20	Presbit	Miyop	Evet	Normal	Sert
21	Presbit	Hipermetrop	Hayır	Az	Yok
22	Presbit	Hipermetrop	Hayır	Normal	Yumuşak
23	Presbit	Hipermetrop	Evet	Az	Yok
24	Presbit	Hipermetrop	Evet	Normal	Yok



Şekil 2.11. Çizelge 2.3’deki lens kullanımı veri kümesini tanımlayan STATISTICA programı kullanılarak ID3 yöntemiyle elde edilmiş karar ağacı

2.2.6. C4.5 karar ağacı algoritması

C4.5, sınıfı belirli girdi verilerini kullanarak karar ağacı oluşturan bir algoritmadır. C4.5, “bilgi kazanımı” kavramını kullanarak oluşturduğu karar ağacından, veri kümeleri için sınıflandırma kuralları çıkarır (Vijayarani ve Divya 2011). Bilgi kazanımı entropiye dayanan bir kavramdır. Entropi saflığı ya da düzensizliği ölçer. Karar ağacı dallandırılırken, bilgi kazanımı en yüksek olan öznelikten dallandırma yapılarak kökten yapraklara doğru gidilir. C4.5 algoritmasının adımları aşağıda verilmiştir (Quinlan 1993):

- (1) Her bir a özneliği için bilgi kazanımını, p pozitif gerçekleşen, n de negatif gerçekleşen sonuçların olasılıkları olacak şekilde

$$Entropi(a) = -p \log_2(p) - n \log_2(n)$$

olarak ifade edilen entropi fonksiyonu deęerini kullanarak hesapla.

(2) En yksek bilgi kazanımı olan znitelikten dallanma yap.

(3) Adım 1'e dn ve tm yapraklar saflaşana kadar hesaplama yapmaya devam et.

2.2.7. NBTree karar aęacı algoritması

“NBTree” (Kohavi 1996) , saf Bayes sınıflandırıcı ve karar aęacı ęrenme yntemlerini birleřtiren bir algoritmadır. Saf Bayes sınıflandırıcı,

$$p(a|b) = \frac{p(b|a)p(a)}{p(b)}$$

ifadesiyle verilen, kořullu olasılıkları hesaplayan “*Bayes teoremi*” ne dayanır. Karar aęacının her bir dęmnde saf Bayes deęeri hesaplanır ve her rnek, yesi olduęu dęmdeki saf Bayes deęerine gre sınıflandırılır. NBTree algoritmasındaki karar aęacı oluřumunun adımları C4.5 algoritmasındakine benzer. Karar aęacı oluřturulduktan sonra, her dęm iin kendi lokal verisi kullanılarak saf Bayes deęerleri hesaplanır. Her dęmdeki veri, hesaplanan deęerler gz nnde bulundurularak bir sınıfa atanır.

2.2.8. Kural kmeleri

Kural kmesi, kural tabanlı bir sınıflandırma modelidir ve “*eęer-ise*” yapısındaki kuralların ıkarımını yapar. Her kural kořulları belirlerken “*eęer*” kısmında znitelikleri kullanır, “*ise*” kısmında ise bunlarla sınıflar arasındaki iliřkiyi oluřturur. Bu tip mantıksal kurallara alternatif olarak olasılık kuralları da ıkarılabilir. Olasılık kuralları ıkarılırken sınıf atamasının yanı sıra her sınıfa dřen verinin sayısı ya da olasılıęı da verilir (Clark ve Boswell 1991).

Kural kmeleri genellikle karar aęalarından daha basit ve anlaşıldır. Bunun nedeni ise karar aęalarının da aslında bir dizi eęer-ise kuralı olarak yorumlanabilmesidir. Karar aęacının her bir yapraęı bir kurala karřılık gelir. Bu kuralın kořulları, kkten yapraęa kadar olan dallar sonucu ise, yapraęa atanmıř sınıftır. Karar aęacından ıkan kuralların

yapısı kural kümelerine göre daha sabittir çünkü bütün kuralların ilk özniteliği her zaman kök ile temsil edilen öznitelik olur.

Bir karar ağacı ve kural kümesi tarafından oluşturulan kurallar arasındaki temel fark, kural kümesinin oluşturduğu kuralların bütün veri kümesini temsil eden ve birbiriyle çoğunlukla örtüşmeyen kurallar olmasıdır. Kural kümesi tarafından bulunan kurallar, mümkün tüm öznitelik kombinasyonlarıyla, bütün veri kümesini temsil ettiğinden, sayıları oldukça fazla olabilir. Örneğin Çizelge 2.3'deki veri kümesini temsil edecek kural kümesi tüm öznitelik kombinasyonları hesaplandığında 24 olmalıdır. Bu kural kümesi indirgendiğinde ise Çizelge 2.4'deki dokuz adet kurala ulaşılır. Bu sebeple kullanıcı kolaylığı açısından tercih edilen yöntem kuralların bir kısmını örtüştürerek kural sayısını azaltmaya gitmektir. Kural sayısı azaltmaya gidildiğinde, birden fazla kural ile temsil edilebilen veya hiçbir kural tarafından temsil edilemeyen örnekler ortaya çıkabilir. Genellikle, bu tip durumlarda fazla sayıdaki veriyi doğru sınıflandıran kurallar geçerli kabul edilir. Kullanıcı kolaylığı açısından Çizelge 2.4'deki kurallar daha da basite indirgenmek istenirse Çizelge 2.5'deki sadece iki adet kural içeren küme elde edilebilir. Çizelge 2.4 ve Çizelge 2.5'deki kural kümeleri karşılaştırıldığında ilk çizelgedekinin %100 doğruluk sağladığı fakat kural sayısının fazla olduğu görülür. İkinci çizelgedeki kural kümesi daha az sayıda kural çıkarımı yapmıştır fakat ikinci kuralda doğruluk oranı %83,33'e düşmüştür. Bu tip durumlarda basit fakat doğruluğu daha düşük ya da doğruluğu fazla fakat çok kural içeren kural kümelerini seçmek kullanıcının tercihidir.

Çizelge 2.4. Lens kullanımı veri kümesini tanımlayan kural kümesi

Kural No.	Yaş	Tanı	Astigmat	Gözyaşı üretimi	Lens sınıfı
1	Genç	---	Evet	Normal	Sert
2	---	Miyop	Evet	Normal	Sert
3	Genç	---	Hayır	Normal	Yumuşak
4	Ön-presbit	---	Hayır	Normal	Yumuşak
5	Presbit	Hipermetrop	Hayır	Normal	Yumuşak
6	---	---	---	Az	Yok
7	Ön-presbit	Hipermetrop	Evet	Normal	Yok
8	Presbit	Miyop	Hayır	Normal	Yok
9	Presbit	Hipermetrop	Evet	Normal	Yok

Çizelge 2.5. Lens kullanımı veri kümesinin özet kural kümesi

Kural No.	Yaş	Tanı	Astigmat	Gözyaşı üretimi	Lens sınıfı
1	---	---	---	Az	Yok
2	---	---	Hayır	Normal	Yumuşak

“DecisionTable” ve “PART” literatürde özellikle karşılaştırma algoritmaları olarak sıkça karşılaşılan karar ağacı algoritmalarındandır (Vijayarani ve Divya 2011).

2.2.9. DecisionTable (“KararTablosu”) algoritması

DecisionTable (Karar tablosu), oluşturması basit ve anlaşılır kural tabanlı yöntemlerdendir. Görsel olarak kullanıcıya girdi parametrelerinin tüm kombinasyonları için çıkacak sonuçları net bir şekilde ifade ederler. Karar tabloları 4 ana bölümden oluşur (Kohavi 1995):

- i. Koşullar
- ii. Eylemler

- iii. Koşul alternatifleri
- iv. Eylem seçimi

Karar tablolarının yapısı Şekil 2.12’de gösterilmiştir. Koşullar kısmında özniteliklerin ana başlıkları yazılır. Koşul alternatiflerinde ise öznitelik parametre değerlerinin alabileceği tüm alternatifler bulunur. Koşulların sayısı girdi verilerinin öznitelik sayısına, koşul alternatiflerinin sayısı ise girdi parametrelerinin olabilecek bütün kombinasyonlarının sayısına eşittir. Eylemler veri kümesinde bulunan çıktı değerlerinden oluşur. Veri kümesinde çeşitli parametre kombinasyonları için olan tüm çıktılar eylem olarak işlenir.

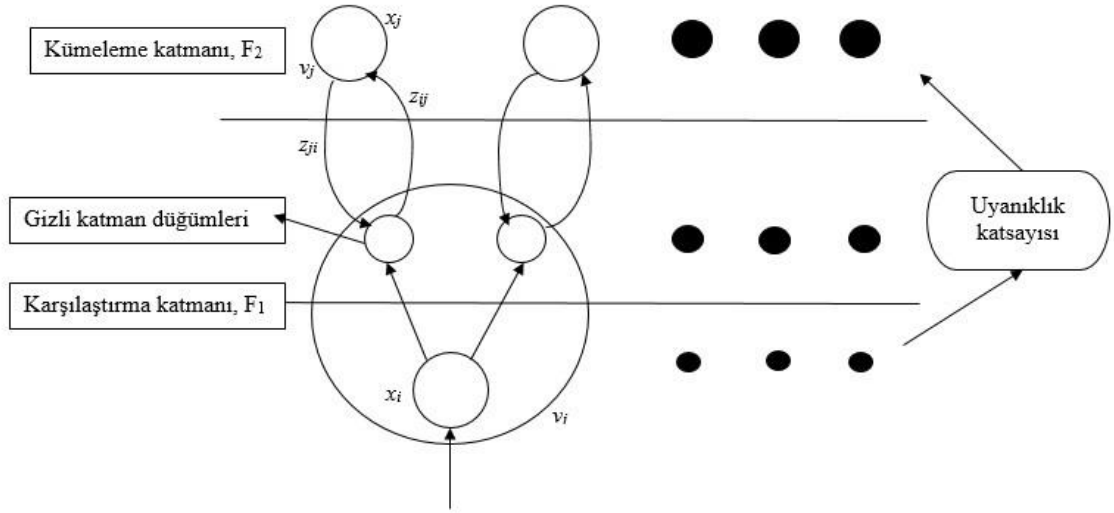
Koşullar	Koşul alternatifleri
Eylemler	Eylem seçimi

Şekil 2.12. Karar tablosunun görsel anlatımı (Kohavi 1995’ten değiştirilerek alınmıştır)

Veri kümesindeki bilgilere göre karar tablosu oluşturulur. Tablodaki her sütun bir kuralı temsil eder. Veri kümesinde her bir parametre kombinasyonu için bir eylem olmayabilir. Bu kombinasyonlar için eylem seçimi kısmı boş bırakılır. Tabloda eğer varsa, mümkün olmayan alternatifler için olan kuralların sütunları silinir. En son adım olarak tablonun basitleştirilmesi gerçekleştirilir. Tabloda eylem seçimi aynı olan sütunlar bulunur. Bu sütunlarda, eğer koşul alternatifleri tek bir koşul dışında aynı olan sütunlar varsa bunlar tek bir sütuna indirgenir. İndirgenmiş yeni kural sütununda farklı olan koşulun alternatif değeri boş bırakılır. Bu demektir ki boş bırakılan koşul değerinin eylem seçiminde bir etkisi yoktur (Werner 2005).

2.2.10. PART algoritması

PART’ın açılımı “*Yansıtılabilir Uyarlanabilir Yankılaşım Teorisi*” dir. PART (Cao ve Wu 2002) algoritması verileri, önceden belirlenmiş bir benzerlik kıstası ve mesafelere göre kümeler. Benzerlik kıstası “*uyanıklık katsayısı*” olarak adlandırılan değer ile belirlenir. PART algoritması ART (“*Uyarlanabilir Yankılaşım Teorisi*”) (Carpenter ve Grossberg 1987) algoritmasının geliştirilmiş halidir.



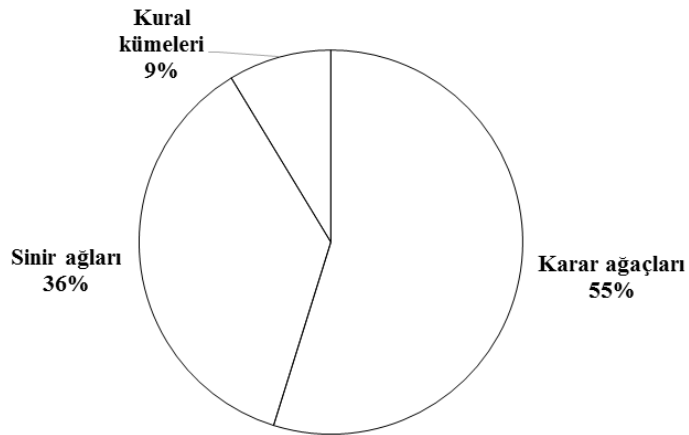
Şekil 2.13. PART yapısı

Uyarlanabilir yaklaşım teorili algoritmalar farklı katmanlardan oluşan bir ağ yapısındadır. PART ve ART algoritmalarının farklılaştığı yer PART'in yapısındaki, Şekil 2.13'de de görülen, karşılaştırma ve kümeleme katmanlarının arasında yer alan gizli katmandır. YSA'dakine benzer bir şekilde, F_1 katmanındaki v_i düğümlerinin x_i değerleri, z_{ij} ağırlıklarıyla çarpılarak bir aktivasyon değeri elde edilir. Bu elde edilen aktivasyon değeri, uyanıklık katsayısından küçük ise kümeleme katmanında yeni bir v_j düğümü oluşturulur. Her girdi için bu işlem gerçekleştirilir ve aktivasyon değerinin uyanıklık katsayısından büyük olduğu her döngüde z_{ji} katsayıları kullanılarak ağırlıklar güncellenir (Cao ve Wu 2002).

2.3. YSA, Karar Ağaçları ve Kural Kümelerinin Veri Madenciliği Açısından Karşılaştırılması

Literatürde yapılan çalışmalar göz önünde bulundurulduğunda, Şekil 1.4'de de görüldüğü üzere veri madenciliğinde YSA kullanım oranı, karar ağaçları ve kural kümelerinin kullanım oranına göre daha fazladır. Özellikle çok katmanlı YSA'ların, doğrusal olmayan yapıları sayesinde gösterdiği yüksek performans, sınıflandırma yöntemi olarak sıkça başvurulan bir yöntem haline gelmelerini sağlamıştır (Liao ve Wen 2007). Sınıflandırma yöntemi olarak YSA, karar ağaçları ve kural kümelerine göre daha iyi performans

göstererek, özellikle karşılaşılmamış örneklerin sınıflandırılmasında daha başarılı olmuştur (Brown ve ark. 1993, Perner ve ark. 2001, Westreich ve ark. 2009). YSA'ların yüksek performansına rağmen, sınıflandırma yaparken nasıl yaptıklarının içerdikleri karmaşık modeller yüzünden açıklanamıyor olması bir dezavantaj haline gelmiştir. Tek tek girdileri doğru bir şekilde sınıflandırmasında rağmen, YSA veri kümesi için kurallar çıkarmada yani veri kümesini kullanıcı için açıklayıcı hale getirmede yetersiz kalmıştır (Andrews ve ark. 1994). Bu yüzden de kural çıkarımı alanında diğer modellere göre kullanım oranı daha düşüktür. Kural çıkarımı alanında en çok kullanılan modeller karar ağaçlarıdır (Şekil 2.14). Karar ağaçlarının karmaşık ve çok sayıda veri içeren kümelerde çok fazla dallanması sebebiyle, bu tip problemlerde karar ağaçları da açıklayıcı basitliğini yitirmiştir (Mak ve ark. 2002).



Şekil 2.14. Son 10 yıldaki yayınlarda veri madenciliğinde kullanılan özdevinimli modellerin kural çıkarımı alanındaki dağılımı (Web of Science 2014)

YSA'nın yapısının karmaşıklığı sayesinde veri madenciliği alanında gösterdiği yüksek performans (Çizelge 2.6) araştırmacıları, kural çıkarımı için YSA'yı açıklayıcı hale getiren hibrit yöntemler geliştirmeye yöneltmiştir. Yapılan çalışmalarda meta-sezgisel algoritmaların optimizasyon problemleri için gösterdiği başarılı sonuçlar (Mahmud ve ark. 2012, Pedemonte ve ark. 2011, St. Hilarie ve Liu 2011) veri madenciliğinde yapay sinir ağlarını ve meta-sezgisel yöntemleri bir arada kullanma düşüncesini ortaya

çıkarmıştır. Hibrit algoritmaların çoğu optimizasyon problemlerinde başarılı sonuçlar veren meta-sezgisel algoritmalara sahiptir (Çizelge 2.7).

Çizelge 2.6. YSA'nın veri madenciliğinde avantaj ve dezavantajları

Avantajlar	Dezavantajlar
Veri sınıflandırma yöntemleri içinde son derece doğru ve kesin sonuçlar vermesi	Açıklayıcı gücünün yetersizliği: Çoklu gizli katmanlardaki karmaşık veri dönüşümünün girdi ve çıktılar arasındaki kuralları belirlemeyi zorlaştırması
İstatistiki yöntemlere nazaran normal dağılım zorunluluğu olmadan küçük veri kümeleriyle çalışabilmesi	
Gizli katmanlardaki lineer olmayan dönüşümler sayesinde karmaşık yapıdaki veriler üzerindeki yüksek performansı	

Çizelge 2.7. YSA ve meta-sezgisel algoritmaların hibritlendiği literatür çalışmaları

Ozbakir ve ark. (2011): PSO-Miner	Sinir ağını eğitmek ve çıktı fonksiyonlarını optimize etmek için Parçacık Sürü Optimizasyonundan yararlanılmıştır
Ozbakir ve ark. (2010): DIFACONN-Miner	Sinir ağını eğitmek için Diferansiyel Gelişim Algoritması, kural çıkarımı içinse TACO kullanılmıştır
Ozbakir ve ark. (2009): TACO-Miner	Sinir ağı eğitilmiş ve kural çıkarımı içinse TACO kullanılmıştır
Baykasoglu ve ark. (2007): MEPAR-Miner	Sinir ağı eğitilmiş ve kural çıkarımı içinse Çoklu-Denklem Programlama kullanılmıştır
Elalfi ve ark. (2004): Genetik Algoritma	Sinir ağı eğitilmiş ve kural çıkarımı ise Genetik Algoritma ile yapılmıştır

2.4. Meta-sezgisel Optimizasyon

Hem iş hayatında olsun hem gündelik hayatta, her gün insanlar çeşitli kararlar vererek bir takım sonuçlara ulaşmaktadırlar. Verilen kararlar arasında bir fabrikanın bir günde ne kadar üretim yapacağı veya bir eve bir gün için ne kadarlık alışveriş yapılacağı olabilir. Bu karar verme mekanizması, çeşitli seçenekler arasından belirli kısıt ve kıstaslara uygun olarak yapılan seçimler ile en iyi sonuca ulaşmayı hedefler. Eniyileme ile en iyi, yani

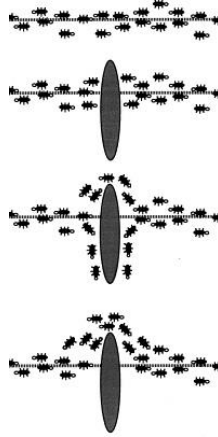
optimum sonuca ulaşmanın diğer bir adı da optimizasyondur. Murty (2003) optimizasyonu, karar değişkenlerinin şartlarını sağlamak koşuluyla optimize edilecek amaç fonksiyonlarının olması durumunda ortaya çıkan model olarak tanımlamıştır. Optimizasyon modelleri, karar değişkenlerinin alacağı değerleri belirleyerek, amaç fonksiyonlarını istendiği şekilde maksimize veya minimize etmeye çalışırlar.

Son yıllarda optimizasyon modellerini çözmek için, doğadan esinlenerek geliştirilen birçok optimizasyon algoritması ortaya çıkmıştır. Bu algoritmaların çoğu farklı tipteki, geniş bir problem kümesine uygulanabilen meta-sezgisel yöntemlerdir. Meta-sezgisel algoritmalar doğal fenomenlerden esinlenerek optimum sonuca erişmeyi temel alırlar. Meta-sezgisel algoritmaların en iyi çözümü buldukları kanıtlanamaz, fakat yakınsama özelliklerinden dolayı en iyi çözüme yakın bir çözüm bulacakları öngörülür. Meta-sezgisel algoritmalar, özellikle fazla sayıda veri içeren karmaşık problem tiplerinde, kesin çözüm yöntemlerine göre süre ve kullanıcı için yapısal anlaşılabilirlik avantajları nedeniyle, optimizasyon için ihtiyaç duyulan algoritmalarındandır (Akyol ve Alataş 2012). Bu tür algoritmalara örnek olarak karıncaların yiyecek kaynaklarına ulaşmasından esinlenen karınca kolonisi optimizasyonu (Dorigo ve ark. 2006), biyolojik evrimden esinlenen evrimsel algoritma (Angeline ve ark. 1994), parçacık sürüsü optimizasyonu (Kennedy ve Eberhart 1995) ve harmoni arama motoru (Geem ve ark. 2001) verilebilir. Benzer bir şekilde, doğadaki balarısı kolonilerinin akıllı yiyecek bulma davranışından esinlenen Yapay Arı Kolonisi algoritması (YAK), optimizasyon problemleri için teknik bir rapor olarak ilk defa Karaboga (2005) tarafından önerilmiştir. YAK algoritması, yaygın olarak kullanılmaya başlayan güncel sürü zekâsı optimizasyon algoritmalarından biridir. Yaygın olarak kullanılmasının sebebi, balarısı kolonilerinin akıllı yiyecek bulma davranışından esinlenen optimum çözümü bulma yönteminin başarılı sonuçlar vermiş olmasıdır (Karaboga ve Akay 2009).

Sürü optimizasyon algoritmaları sürekli, kesikli veya ikili gibi çeşitli yapıdaki karar değişkenleriyle çalışabilirler. Bu çalışmada önerilen yöntemin yapısı gereği, ikili değişkenlerle çalışma özelliklerinden dolayı, Hiroyasu (2000) tarafından geliştirilen Turlayan Karınca Kolonisi Optimizasyonu algoritması ve ikili değişkenli YAK algoritmaları ayrıntılı olarak incelenmiştir.

2.4.1. Karınca kolonilerinin yiyecek bulma davranışları

Karınca kolonilerinin yiyecek arama davranışını algoritma haline dönüştüren ilk çalışma Dorigo ve ark. (1991) tarafından yapılmıştır. Karıncalar yiyecek kaynakları bulmak için yuvalarının etrafını rasgele ararlar. Yiyecek kaynağı bulan karıncalar, yiyecek kaynağının miktarı ve kalitesine göre, yuvaya dönerken kullandıkları yola feromon adında bir madde bırakırlar. Karıncaların bıraktıkları feromon miktarı yolun uzunluğuyla ters orantılıdır. Feromon miktarı fazla olan yoldan daha fazla karınca geçer ve böylece yiyecek kaynağı ile yuva arasındaki en kısa yol bulunmuş olur. Karıncaların geçmediği yoldaki feromon ise zamanla buharlaşarak etkisini yitirir. Karınca kolonisinin yiyecek kaynağı bulma fenomeni Şekil 2.15 ile temsil edilmektedir. Dorigo ve ark.'ın (1991) “*karınca algoritması*” olarak adlandırdığı Karınca Kolonisi Optimizasyon (KKO) algoritması temel alınarak, daha sonraki yıllarda literatürde farklı versiyonlarda ve çeşitli problem tiplerinde KKO algoritmaları çalışılmıştır (Costa ve Hertz 1997, Gambardella ve Dorigo 2000, Merkle ve ark. 2002, Stützle ve Dorigo 2002, Dorigo ve Blum 2005).

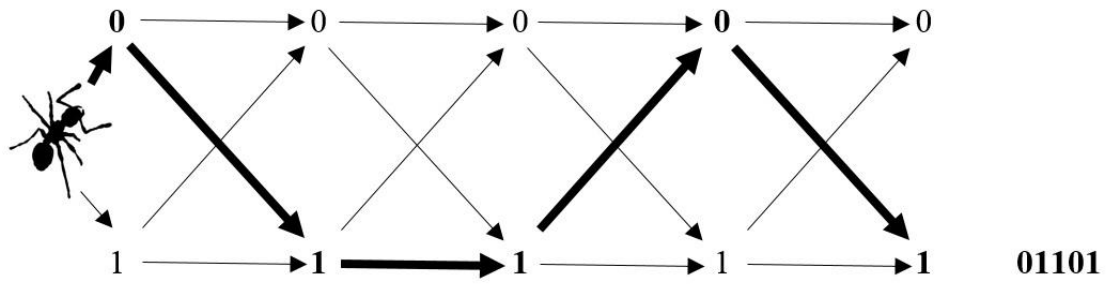


Şekil 2.15. Karınca kolonisinin yiyecek kaynağı ve yuva arasındaki yolu seçme süreci (Dorigo ve ark. 1997)

2.4.2. Turlayan karınca kolonisi optimizasyon (TKKO) algoritması

Hiroyasu ve ark. (2000) tarafından önerilen TKKO algoritmasında her bir çözüm, tasarım parametrelerinin değerlerinden oluşan bir vektör ile temsil edilir. Bu vektördeki tasarım

parametreleri ikili bitlerin dizisi olarak kodlanmıştır. Yani her bir çözüm ikili bitlerden oluşan bir vektördür. TKKO algoritmasındaki yapay karıncalar, dizideki her bitin değerini bulmak için uğraşırlar. Dizideki bitler sadece 0 veya 1 değerini alabildiğinden, yapay karıncalar çözüm vektöründeki tüm bitlerin değerinin 0 mı yoksa 1 mi olduğuna karar vermeye çalışırlar. Bu karar verme aşamasında, yapay karıncalar bit değerlerini seçerken sadece feromon bilgisinden yararlanır. Bir karınca dizideki bütün bitlere değer atadıktan sonra, problem için aday bir çözüm yaratmış olur. TKKO algoritmasında karıncanın bitlere değer atama aşaması Şekil 2.16’da gösterilmiştir.



Şekil 2.16. TKKO algoritmasındaki bir karıncanın bulduğu yapay yol ve aday çözüm

Bulunan çözümün, “*değerlendirme fonksiyonu*” olarak adlandırılan fonksiyon kullanılarak, kalitesini gösteren sayısal bir değer hesaplanır. Dizideki bitler arasındaki yapay alt yollara bırakılacak feromon miktarı bu kalite değeri kullanılarak hesaplanır. Kolonideki tüm karıncalar çözümlerini oluşturduktan ve bütün sonuçlar için feromon miktarı hesaplandıktan sonra, bitler arasındaki alt yolların feromon miktarı güncellenir. Güncelleme işlemi, önceki feromon miktarını belirlenen oranda azaltıp, yeni feromon değerini alt yollara ekleyerek gerçekleştirilir. Her bir aşamadaki yapay alt yol oluşturulurken karıncanın 0 ya da 1 değerini seçmesinin olasılığı

$$p_{ij} = \frac{\tau_{ij}}{\tau_{ij} + \tau_{ii}} \quad (2.2)$$

ifadesi ile hesaplanır. Buradaki p_{ij} , i-j alt yolunu seçme olasılığı, τ_{ii} ve τ_{ij} ise i-i ve i-j alt yollarındaki feromon miktarıdır. Olasılık tayinindeki feromon miktarı enbüyüklenme problemleri için

$$\Delta \tau_{ij}^k(t, t+1) = \begin{cases} Q * F_k & \text{eğer } k \text{ karıncası } i-j \text{ yolunu kullanırsa} \\ 0 & \text{eğer } k \text{ karıncası } i-j \text{ yolunu kullanmazsa} \end{cases} \quad (2.3)$$

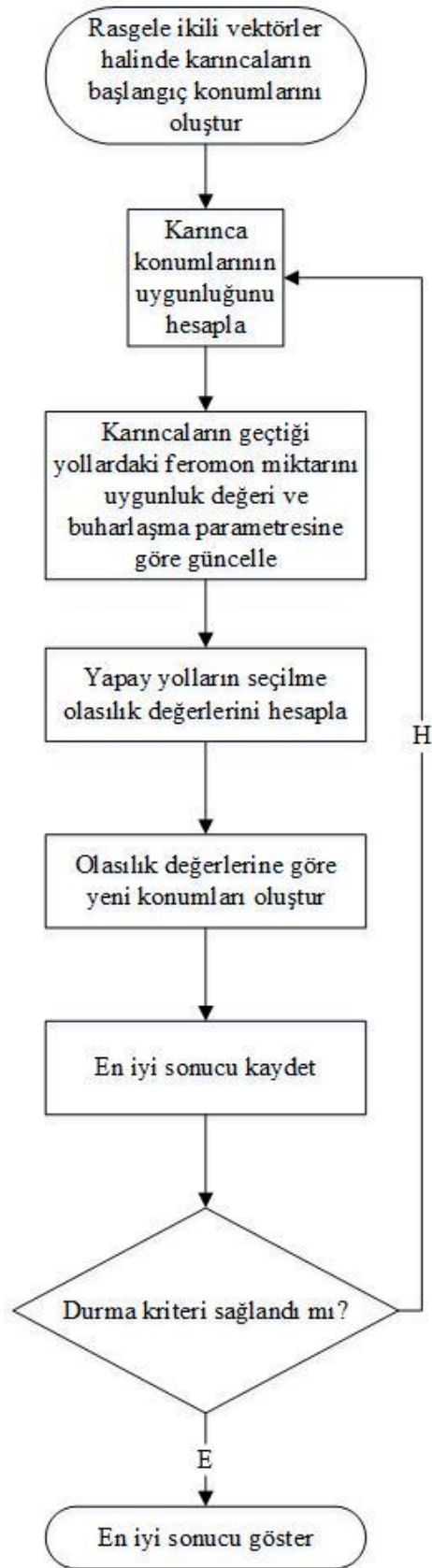
ile hesaplanmaktadır. Buradaki $\Delta \tau_{ij}^k$, k karıncası tarafından $i-j$ alt yoluna bırakılan feromon miktarı, Q , pozitif bir sabit sayı, F_k , ise k karıncası tarafından bulunan çözüm vektörü kullanılarak hesaplanan amaç fonksiyonu değeridir. Kolonideki tüm karıncalar arama sürecini tamamlayıp çözüm ürettikten sonra, $(t, t+1)$ zaman diliminde $i-j$ alt yoluna M karınca tarafından bırakılacak toplam feromon miktarı aşağıdaki ifade ile hesaplanır.

$$\Delta \tau_{ij}(t, t+1) = \sum_{k=1}^M \Delta \tau_{ij}^k(t, t+1) \quad (2.4)$$

En son olarak, $(t+1)$ zamanında $i-j$ alt yolundaki feromon miktarı

$$\tau_{ij}(t+1) = \rho \tau_{ij}(t) + \Delta \tau_{ij}(t, t+1) \quad (2.5)$$

ile güncellenir. Buradaki ρ katsayısı $[0,1]$ aralığında değer alan buharlaşma parametresidir (Sreejith ve ark. 2009). TKKO'nun adımları Şekil 2.17'de sunulmuştur.



Şekil 2.17. TKKO için akış diyagramı

2.4.3. Arı kolonilerinin yiyecek bulma davranışları

Doğada sürü halinde yaşayan arıların yaşamlarını devam ettirmek için yaptıkları en önemli işlerden biri yiyecek aramaktır. Balarısı kolonileri kendilerine özgü, zeki yiyecek bulma davranışlarıyla sürekli bir biçimde kovana buldukları kaynaklardan nektar taşır ve yeni yiyecek kaynakları ararlar.

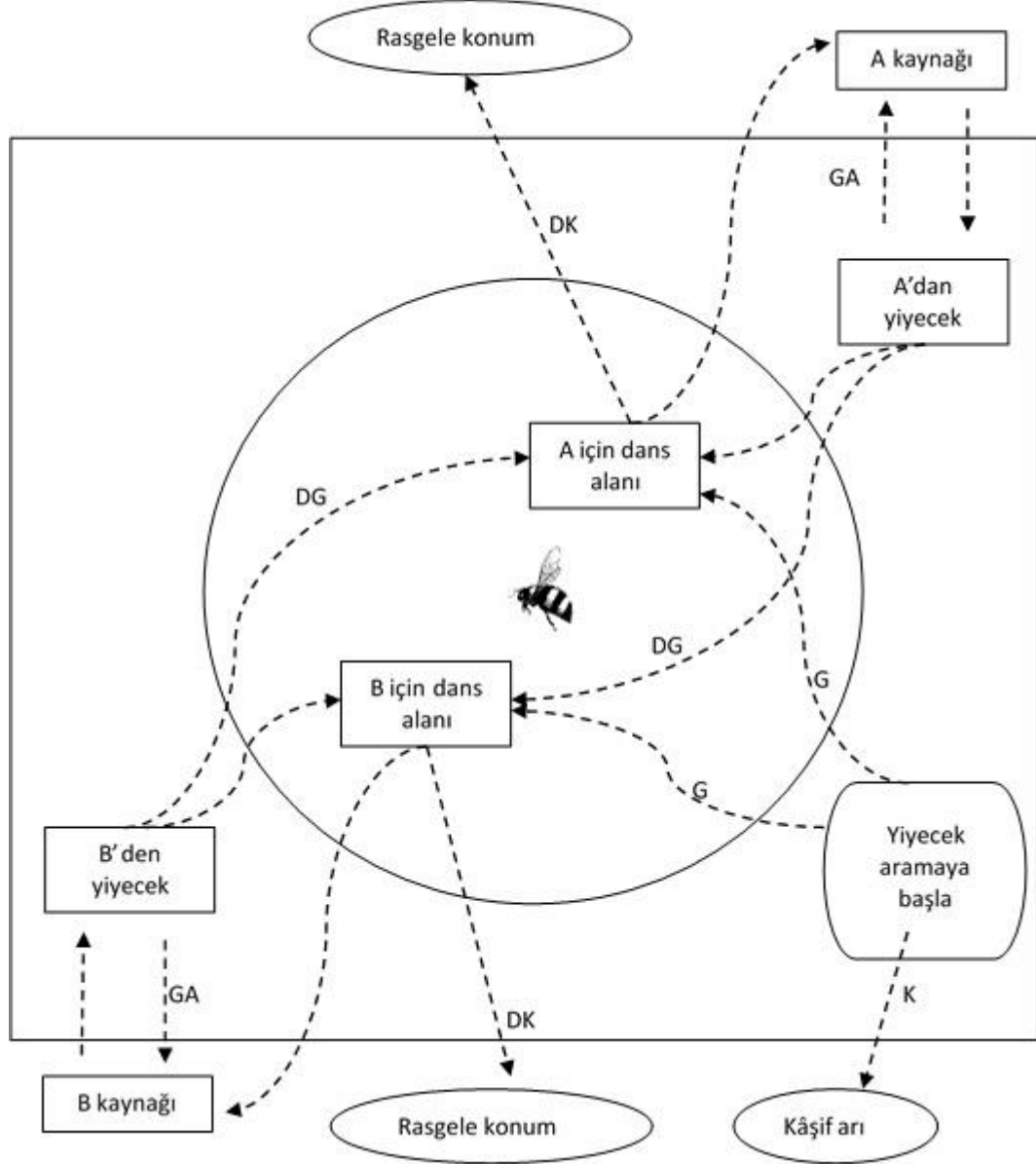
Balarılarının yiyecek arama süreci 3 ana bileşenden oluşur (Karaboga 2005):

- i. Yiyecek kaynakları
- ii. İşçi arılar
- iii. Görevsiz işçi arılar

Balarısı kolonisindeki işçi arılardan her biri bir yiyecek kaynağından sorumludur. İşçi arılar (GA) sorumlu oldukları kaynaktan kovana yiyecek taşırken, aynı zamanda koloniyi bu kaynağın kalitesi ve yeri konularında da bilgilendirir. Görevsiz işçi arılar ise yeni yiyecek kaynakları ararlar. 2 tip görevsiz işçi arı vardır, bunlar kâşif (K) ve gözcü (G) arılar olarak adlandırılır. Gözcü arılar kovanda görevli işçi arılardan gelecek bilgiyi bekler ve bu bilgi doğrultusunda yeni yiyecek kaynaklarına yönelir. Kâşif arılar ise, yeni yiyecek kaynaklarını rasgele bir şekilde arar. Kâşif arıların sayısının kovandaki diğer arıların sayısına oranı %5-10 arasında değişir (Seeley 1995).

Balarıları yiyecek kaynaklarından edindikleri bilgileri kovandaki “*dans alanı*” denen yerde paylaşırlar. İşçi arı, bu alanda sorumlu olduğu yiyecek kaynağının kalitesine bağlı olan bir dans yapar. İşçi arıların dansını izleyen gözcü arı, dansın şekline göre yöneleceği kaynağı seçer. Dansın şekli ve süresi, yiyecek kaynağının kalitesiyle doğru orantılıdır. Nektar miktarı yüksek kaynaklardan sorumlu görevli işçi arıların dansı, düşük kaynaktan sorumlu olanlara göre daha uzun sürer. Gözcü arıların, dansı uzun süren kaynağı seçmelerinin olasılığı daha yüksektir, böylece kalitesi yüksek kaynaklara daha fazla sayıda arı atanmış olur. Bir kaynağın yiyeceği tükendiği zaman, o kaynaktan sorumlu işçi arı deneyimli kâşif (DK) veya deneyimli gözcü (DG) arıya dönüşür. Eğer bulduğu kaynak

kaliteliyse, kâşif veya gözcü arı da işçi arıya dönüşebilir (Tereshko ve Loengarov 2005). Balarılarının yiyecek arama davranışı Şekil 2.18 ile gösterilmiştir.



Şekil 2.18. Balarısı kolonisinin yiyecek arama süreci (Baykasoglu ve ark. 2007'den değiştirilerek alınmıştır)

2.4.4. Yapay arı kolonisi (YAK) algoritması

YAK algoritması, balarılarının yiyecek arama davranışından esinlenerek geliştirilen, sürü tabanlı bir meta-sezgisel algoritmadır (Karaboga 2005). Başlangıçta kısıtsız optimizasyon problemleri için geliştirilen YAK algoritması (Karaboga ve Basturk 2007a), daha sonra kısıtlı optimizasyon problemlerini de çözecek şekilde düzenlenmiştir (Karaboga ve Basturk 2007b).

Balarılarının yiyecek bulma sürecini temel alan YAK algoritmasında modelin basitleştirilmesi açısından aşağıdaki kabuller yapılmıştır (Karaboga 2005):

- i. Kovanda işçi, kâşif ve gözcü arılar olmak üzere 3 tipte arı vardır.
- ii. Başlangıçta kovandaki arıların yarısı işçi diğer yarısı da gözcü arıdır.
- iii. Her bir yiyecek kaynağına atanmış tek bir işçi arı vardır, yani işçi arıların sayısı yiyecek kaynağı sayısına eşittir.
- iv. Nektarı tükenmiş kaynağın işçi arısı kâşif arıya dönüşmektedir.

YAK algoritmasında balarısı kovanının çevresini temsil eden arama uzayı karar değişkenlerinin alabileceği alt ve üst sınır değerlerine göre oluşturulur. Başlangıçtaki yiyecek kaynakları bu arama uzayından, N yiyecek kaynağı sayısı ve M parametre sayısı olmak üzere

$$x_{ij} = x_j^{\min} + \text{rassal}(0,1)(x_j^{\max} - x_j^{\min}) \quad (2.6)$$

$$i = 1, \dots, N$$

$$j = 1, \dots, M$$

ifadesi ile rasgele seçilir ve iyileştirme sayaçları sıfırlanır.

YAK'daki her döngü üç aşamadan oluşur: işçi ve gözcü arılar yiyecek kaynaklarına gönderilir, yiyecek kaynaklarının kalitesi tayin edilir ve kâşif arılar olası yiyecek kaynaklarına yönlendirilir. Yiyecek kaynaklarının her birinin yeri problemin

çözümlerinden birini, yiyecek kaynağının nektar miktarı da çözümün kalitesini temsil eder. Her kaynaktaki görevli arı için

$$v_{ij} = x_{ij} + rassal(-1,1)(x_{ij} - x_{kj}) \quad (2.7)$$

kullanılarak yeni bir kaynak üretilir. Burada, $k \in \{1, \dots, N\}$ ve $k \neq i$ durumu geçerlidir. Yeni üretilen kaynağın kovanın sınırlarının dışına çıkması durumunda parametre değerleri

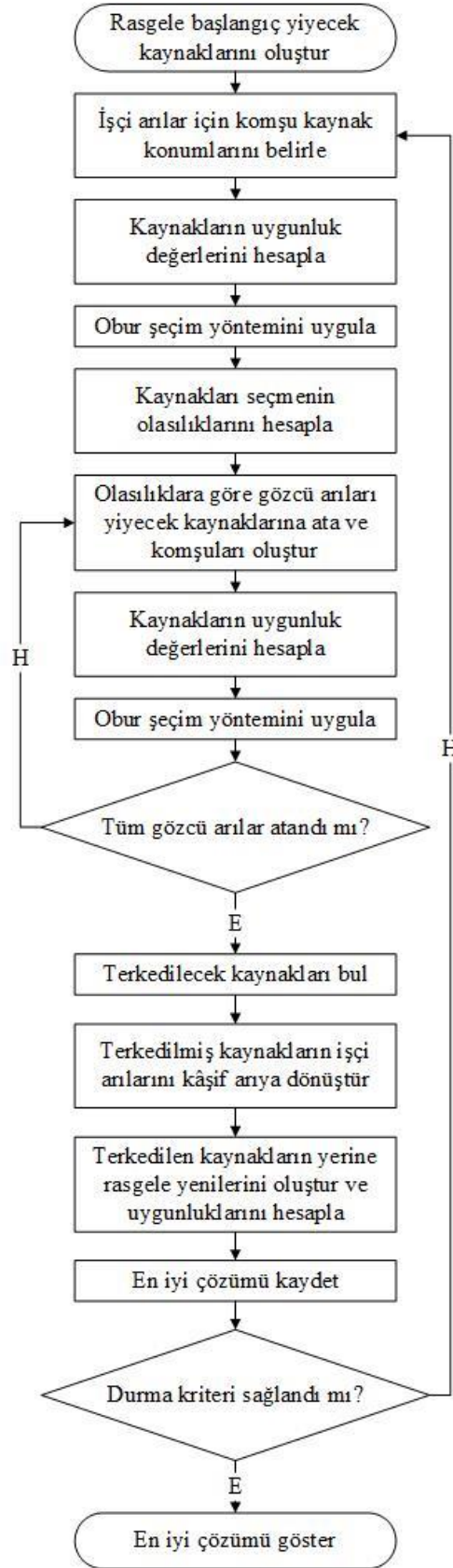
$$v_{ij} = \begin{cases} x_j^{\min}, & v_{ij} < x_j^{\min} \\ v_{ij}, & x_j^{\min} \leq v_{ij} \leq x_j^{\max} \\ x_j^{\max}, & v_{ij} > x_j^{\max} \end{cases} \quad (2.8)$$

ile sınırların içine taşınır.

Yiyecek kaynağı ve komşusu için amaç fonksiyon değerine bağlı uygunluk değeri hesaplandıktan sonra bunlar arasında bir obur seçim yapılır. Aynı zamanda gözcü arı, Denklem (2.9)'daki f_i 'nin i kaynağı için uygunluk değeri olduğu

$$p_i = \frac{f_i}{\sum_{k=1}^N f_k} \quad (2.9)$$

p_i olasılığıyla rasgele bir kaynak seçtikten sonra, Denklem (2.7) ile aday bir kaynak belirler. Bu aday kaynak ve ana kaynak arasında da obur seçim yapılır. Obur seçimlerin sonunda, en iyi uygunluk değeri olan kaynak ana kaynak haline gelir. Her döngüde obur seçimler sonunda değişmeyen ana kaynağın iyileştirme sayacı bir arttırılır. Önceden belirlenen döngü sayısı boyunca iyileştirilemeyen yiyecek kaynakları terkedilir, bu kaynaklardan sorumlu işçi arılar kâşif arı haline gelir ve Denklem (2.6) ile sıfırdan kaynak üretmeye başlarlar (Karaboga ve Akay 2009). YAK'ın adımları Şekil 2.19'da sunulmuştur.



Şekil 2.19. YAK için akış diyagramı

2.4.5. İkili değişkenler için YAK kullanımı

YAK son zamanlarda, sürekli optimizasyon problemlerinde gösterdiği başarılı performans sebebiyle etkili bir yaklaşım olduğunu kanıtlamış ve popülerlik kazanmıştır (Bao ve Zeng 2009, Karaboga ve Basturk 2008, Tsai ve ark. 2009, Zhang ve ark. 2010).

YAK algoritmasının diğer optimizasyon algoritmalarıyla karşılaştırıldığında daha az sayıda kontrol parametresi olması ve hızlı yakınsama özelliğinden dolayı rekabetçi bir algoritma olduğu sonucu ortaya çıkmıştır (Wei ve Hanning 2012). YAK algoritmasının basit ve etkin bir algoritma olması, kümeleme, sinir ağları ve kısıtlı optimizasyon gibi çeşitli mühendislik problemlerine başarılı bir şekilde uygulanmasını sağlamıştır (Karaboga ve Basturk 2007a, Karaboga ve Basturk 2007b, Karaboga ve Basturk 2008). YAK algoritması sürekli değişkenlere sahip problemlere uygulanmak için geliştirildiğinden, ikili değişken yapısındaki problemlere uygulanamaz. YAK algoritmasının kesikli değişkenlere uygulanabilmesi için algoritma yapısında çeşitli değişikliklere gidilmesi gerekmektedir. Gerçek hayat problemlerinde ikili değişken tipindeki problemler ile sıklıkla karşılaşıldığından, YAK algoritması gibi başarılı bir algoritmanın bu tipteki problemlere uygulanabilir hale getirilmesi önem kazanmıştır. Özellikle çizelgeleme ve rotalama problemlerinde kullanılan sıralama ve yerleştirme yöntemleri ikili değişkenlerle çalışmayı zorunlu kılar. Orijinal YAK algoritmasının aksine, ikili değişkenler için olan YAK algoritması yiyecek kaynaklarını sürekli değişkenler ile değil de, ikili değişkenler ile temsil eder. Bu yapıda, kolonideki arılar, aday çözümlerdeki parametre değerlerinin 0 veya 1 değerlerinden hangisini alacağına karar verirler.

İkili değişken tipindeki problemlerin uygulama alanlarının fazlalığı literatürde YAK algoritmasını ikili uzayda kullanılabilecek hale getiren birçok yaklaşımın geliştirilmesine yol açmıştır. Bu algoritmalara örnek olarak Yapay Arı Kolonisinin İkili Hali (Wei ve Hanning 2012) ve İkili Yapay Arı Kolonisi (Pampara ve Engelbrecht 2011) verilebilir. Örnek verilen algoritmalarda çoğunlukla adımlar YAK algoritması ile aynıdır, farklılık, yiyecek kaynaklarını güncelleyen fonksiyonun çeşitli yöntemlerle ikili yapıya uydurulmasıyla oluşur.

2.4.6. Yapay arı kolonisinin ikili hali (BABC)

BABC, Wei ve Hanning (2012) tarafından iş çizelgeleme dalında ikili eniyileme problemlerinin çözümünde kullanılmak üzere geliştirilmiştir. Önerilen algoritmada karar değişkenlerinin değerleri, geliştirilen diferansiyel ifade ile her döngüde değişen olasılıklara göre hesaplanır. Geliştirilen yeni operatör ile BABC algoritması YAK'ın özelliklerini korurken aynı zamanda ikili uzayda çalışabilme özelliğini kazanmaktadır. Buna ek olarak BABC algoritmasında değişkenleri $[0,1]$ aralığına kısıtlamak için “pozisyon kırpması sınır koşulu” stratejisi uygulanır (Wei ve Hanning 2012).

BABC algoritmasında başlangıçtaki yiyecek kaynaklarını rasgele bir şekilde tanımlamak üzere, parametrelerin alt ve üst sınırları sırasıyla AS ve $ÜS$, yiyecek kaynağı sayısı N ve parametre sayısı M olmak üzere

$$\begin{aligned} x_{ij} &= AS_j + rassal[0,1]*(ÜS_j - AS_j) \\ i &= 1, \dots, N \\ j &= 1, \dots, M \end{aligned} \quad (2.10)$$

ifadesi oluşturulur.

Daha sonraki adımda Denklem (2.7) ile komşu kaynakların konumları hesaplanır. Bu adımda YAK algoritmasına ek olarak parametre değerlerini 0 ve 1 haline getirmek için, normalizasyon işleminden sonra

$$v_{ij}^{yeni} = yuvarla(v_{ij}) \quad (2.11)$$

ile en yakın 0 veya 1 değerine yuvarlama yapılır.

$$\begin{aligned} \text{eğer } v_{ij}^{yeni} > ÜS, \text{ ise } v_{ij}^{yeni} &= ÜS \\ \text{eğer } v_{ij}^{yeni} < AS, \text{ ise } v_{ij}^{yeni} &= AS \end{aligned} \quad (2.12)$$

ifadesi ile eşik değer sınırları kullanılır.

Bundan sonraki adımlar, YAK algoritmasındaki gözcü ve kâşif arıların adımlarıyla aynıdır (Wei ve Hanning 2012).

2.4.7. İkili yapay arı kolonisi (*binABC*)

Pampara ve Engelbrecht (2011) tarafından önerilen “*binABC*”, sürekli uzaydaki değerleri (0,1) aralığına dönüştürmek için

$$x_{ij}^{(0,1)} = f(x_{ij}^{sürekli}) = \frac{1}{1 + e^{-x_{ij}^{sürekli}}} \quad (2.13)$$

ifadesiyle verilen sigmoid transfer fonksiyonunu kullanır. Bu fonksiyonla (0,1) aralığına sıkıştırılan değerler daha sonraki adımda τ_{ij} olarak belirlenen eşik değerine göre,

$$x_{ij}^{[0,1]} = \begin{cases} 1, & \text{eğer } \tau_{ij} \leq x_{ij}^{(0,1)} \\ 0, & \text{eğer } \tau_{ij} > x_{ij}^{(0,1)} \end{cases} \quad (2.14)$$

ifadesiyle 0 veya 1 değerlerinden birini alır (Arpitam ve ark. 2013).

2.4.8. Kaynak Özeti

Çizelge 2.8 ile, kaynak araştırması bölümünde anlatılan ve bir kısmı bu çalışma kapsamında kullanılan, veri madenciliği yöntemleri ve meta-sezgisel algoritmalar için bir özet verilmiştir.

Çizelge 2.8. Kaynak özeti

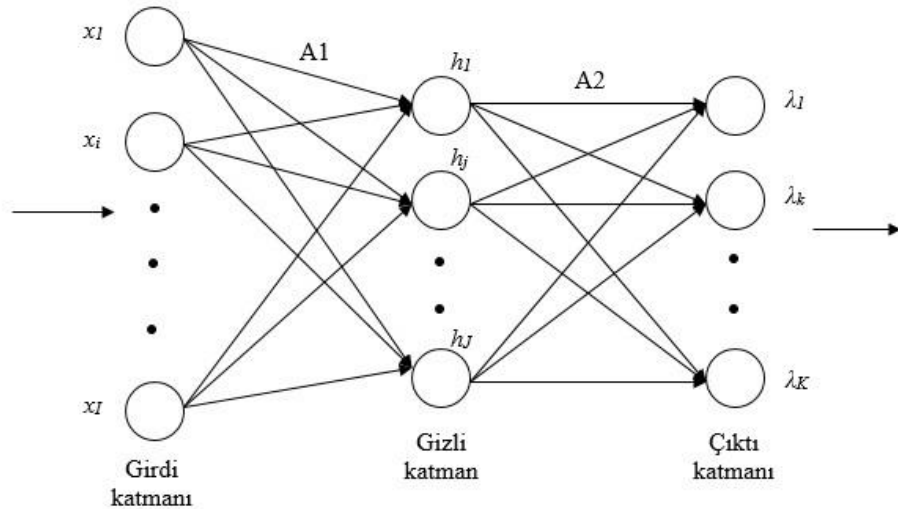
Yayın	Kullanılan yöntem
Veri Madenciliği	
Parpinelli ve ark. (2002)	Ant-Miner: karınca kolonisi ile veri madenciliği
Elalfi ve ark. (2004)	ANN/GA rule extraction: yapay sinir ağları ve genetik algoritma ile kural çıkarımı
Markowska-Kaczmar (2005)	NN/Evolutionary Algorithm: sinir ağları ve evrimsel algoritma ile veri madenciliği
Chen ve ark. (2006)	GARC: birliktelik kurallarının çıkarımıyla veri madenciliği
Hruschka ve Ebecken (2006)	NN/Genetic Algorithm: sinir ağları ve genetik algoritma ile veri madenciliği
Smaldon ve Freitas (2006)	Unordered rule set Ant-Miner: karınca kolonisi ile kural kümelerinin oluşturulması
Baykasoğlu ve ark. (2007)	MEPAR-Miner: sinir ağları ve çoklu-denklemler programlayıcı ile veri madenciliği
Holden ve Freitas (2007)	Hybrid PSO/ACO: parçacık sürü optimizasyonu ve karınca kolonisinin veri madenciliği yöntemi olarak hibritlenmesi
Thabtah ve Crowling (2007)	RMR: önem derecesine göre birliktelik kurallarının çıkarımı ile veri madenciliği
Kahramanli ve Allahverdi (2009)	NN/Artificial Immune System: sinir ağları ve yapay bağışıklık sistemi ile veri madenciliği
Özbakır ve ark. (2009)	TACO-Miner: sinir ağları ve turlayan karınca kolonisi ile kural çıkarımı
Ozbakir ve ark. (2010)	DIFACONN-Miner: sinir ağını eğitmek için diferansiyel gelişim algoritması, kural çıkarımı içinse TACO kullanılmıştır
Ozbakir ve ark. (2011)	PSO-Miner: Sinir ağları ve parçacık sürü optimizasyonu ile veri madenciliği
Setiono ve ark. (2012)	NN: sinir ağları kullanılarak veri madenciliği
Meta-sezgisel Algoritmalar	
Hiroyasu ve ark. (2000)	TACO: turlayan karınca kolonisi
Karaboga (2005)	ABC: yapay arı kolonisi
Pampara ve Engelbrecht (2011)	binABC: ikili yapay arı kolonisi
Wei ve Hanning (2012)	BABC: yapay arı kolonisinin ikili hali

3. MATERYAL VE YÖNTEM

3.1. Problemin Tanımı

Veri kümelerinden kural çıkarımı için, YSA ile elde edilmiş yüksek performanslı sınıflandırma çıktılarını, anlaşılabilir sınıflandırıcı kurallar haline dönüştürecek bir yöntem gereklidir. Bu çalışmada, eğitilmiş YSA üzerinden kural çıkarımını, istenen çıktıyı verecek girdilerin belirlenmesi şeklinde bir optimizasyon problemi olarak tanımlayıp, bir meta-sezgisel algoritma yardımıyla optimale yakın çözümler elde edilmesi amaçlanmaktadır.

Tek gizli katmanlı YSA'nın gizli katmanı ile çıktı katmanında kullanılan sigmoid aktivasyon fonksiyonu, YSA'nın "geri yayılım algoritması" ile eğitime imkân vermekte ve eğitim sonucunda iki grup ağırlık hesaplanabilmektedir. Bunlardan ilki girdi katmanı ile gizli katman arasındaki ağırlıklar ($A1$), ikincisi ise gizli katman ile çıktı katmanı arasındaki ağırlıklardır ($A2$). Eğitim için $X = \{x_1 \dots x_i \dots x_I\}$ şeklinde ikili girdi vektörleri ile $S = \{\lambda_1 \dots \lambda_k \dots \lambda_K\}$ şeklindeki çıktı sınıf vektörleri kullanılır. I , toplam girdi düğümü sayısı, K ise toplam çıktı düğümü sayısıdır (Şekil 3.1).



Şekil 3.1. YSA girdi çıktı ilişkisi

j gizli katman düğümünün toplam girdisi,

$$\sum_{i=1}^I x_i (A1)_{ij}$$

olarak ifade edilir. j gizli katman düğümünün sigmoid fonksiyonu üzerinden çıktısı ise

$$\frac{1}{1 + e^{-\left[\sum_{i=1}^I x_i (A1)_{ij}\right]}}$$

ile verilir. Çıktı katmanındaki k 'nci düğümünün toplam girdisi, J , gizli katmandaki toplam düğüm sayısı olmak üzere,

$$\sum_{j=1}^J (A2)_{jk} \frac{1}{1 + e^{-\left[\sum_{i=1}^I x_i (A1)_{ij}\right]}}$$

şeklinde ifade edilir. Böylece çıktı katmanındaki k 'nci düğümün sigmoid fonksiyonu üzerinden çıktısı λ_k ,

$$\lambda_k(x_i) = \frac{1}{1 + e^{-\left[\sum_{j=1}^J (A2)_{jk} \left(\frac{1}{1 + e^{-\left[\sum_{i=1}^I x_i (A1)_{ij}\right]}} \right)\right]}}$$

ifadesi ile hesaplanır. λ_k , enbüyük değeri 1 olan üssel bir fonksiyondur. Sınıflandırıcı olarak çalışan bir YSA'da herhangi bir girdi vektörünün k çıktı sınıfına ait olması için, λ_k değerinin 1, diğer çıktı katman düğümlerinin de 0 değerini alması gerekir. Yani k sınıfını tam anlamıyla temsil eden girdi vektörü, λ_k değerini enbüyükleyen vektördür. Bu durumda problem doğrusal olmayan bir enbüyükleme problemi olarak tanımlanabilir. Buna göre,

Enbüyükle

$$\lambda_k(x_i) = \frac{1}{1 + e^{\left[\sum_{j=1}^J (A2)_{j,k} \left(\frac{1}{1 + e^{\left[\sum_{i=1}^I x_i (A1)_{i,j} \right]}} \right) \right]}}$$
$$x_i \in \{0,1\}$$

olarak ifade edilebilen optimizasyon probleminde karar değişkeni olan x_i 'nin değerleri ancak meta-sezgisel yöntemlerle belirlenebilir.

3.2. YSA Eğitimi

Tek gizli katmanlı YSA'ların birçok veri kümesi üzerinde başarı sonuçlar vermesine karşılık, gizli katman sayısı arttıkça sinir ağı karmaşıklığının artması nedeniyle eğitimlerde tek gizli katman kullanılmıştır (Hornik ve ark. 1989). Eğitilecek YSA gizli katman düğüm sayısını hesaplamak üzere STATISTICA programından yararlanılmıştır. İkili hale dönüştürülmüş veri kümeleri kullanılarak STATISTICA'da ortalama hata karesini enküçükleyecek şekilde geri yayılım eğitim algoritmasıyla YSA eğitimi yapılmıştır. Gizli katman ve çıktı katmanının her ikisinde de Denklem (2.1) ile tanımlanan sigmoid aktivasyon fonksiyonu kullanılmıştır. Eğitilen 100 adet farklı ağ yapısından en iyi performansı gösteren beş tanesi seçilmiştir. MATLAB ortamında kodlanarak, eğitilecek YSA'nın gizli katman düğüm sayısı, seçilen beş ağdan performansı en iyi olanın gizli katman düğüm sayısı olarak belirlenmiştir. Veri kümeleri için seçilen gizli katman yapıları EK 3'te detaylı olarak verilmiştir.

STATISTICA ile belirlenen ağ yapısı kullanılarak yapılacak sınıflandırma test veri kümeleri için ağ ağırlıkları, MATLAB ortamında ortalama hata karesini en küçükleyecek şekilde geri yayılım eğitim algoritması kodlanarak elde edilmiştir. Enbüyük epok sayısı 5000, hata eşik değeri ise 10^{-3} olarak seçilmiştir. Eğitimlerde veri kümesindeki örneklerin 2/3'ü eğitim, geri kalanı da test için kullanılmıştır. EK 4'te de görüleceği üzere çoğu veri kümesi için eğitim, epok sayısı 5000'e gelmeden bitmiştir. Geri yayılım, girdiler ve çıktılar arasındaki ilişkiyi en iyi şekilde ifade edecek fonksiyonel karakteristiği bulmak için ağırlıkların güncellenmesini sağlayan kısmı türev tabanlı bir algoritmadır. Her bir

girdi için hata çıktıdan girdi katmanına doğru yayılmaya çalışılır (Haykin 2009). Geri yayılım algoritmasının adımları aşağıdaki gibidir:

Başla

Gizli katman düğüm sayısını belirle

Epok sayısını ve hata eşik değerini belirle

Başlangıç ağırlıklarını sıfıra yakın küçük rassal değerler olarak ata

Eğer (epok sayısı < enbüyük epok sayısı veya hata > hata eşiği) *ise*

Her veri kümesindeki girdi *için*

Ağdaki çıktı değerini hesapla

Ağ çıktısı ile hedef çıktı arasındaki hatayı hesapla

Hata karesini hesapla

Hata değeri üzerinden ağırlıkları güncelle

Bitir

Bitir

Bitir

MATLAB ortamında, her bir veri kümesi için, eğitim algoritması 10 kez çalıştırılmış ve veri kümesini en yüksek doğrulukla temsil eden ağırlıklar saklanmıştır. Ağ performans grafikleri ve hata matrisleri EK 5’te verilmiştir.

3.3. İkili YAK Tabanlı Olasılık Eşik Değerli Algoritma

YSA eğitimlerinden sonra elde edilen katmanlar arası ağırlıklar kullanılarak, Bölüm 3.1’de tanımlanan optimizasyon problemini çözmek amacıyla burada İkili YAK Tabanlı Olasılık Eşik Değerli Algoritma (İYAKO) önerilmektedir. Yapılan kaynak araştırmasında bu yapıdaki problem tipinde YAK algoritmasıyla yapılmış bir çalışmaya rastlanmamıştır. İYAKO, YAK algoritmasının adımlarını (Şekil 2.19) temel alır ve YSA eğitimi sonucu elde edilen ağ ağırlıklarını kullanarak veri kümesini en iyi şekilde ifade eden kuralları çıkarmayı hedefler.

Algoritmaya başlanmadan önce, her bir veri kümesi için, önsel (apriori) olasılık hesabı yapılır. İkili değer alabilen her bir öznitelik parametresinin $x_j = 1$ olma olasılığı σ_j , Z kullanılan veri kümesindeki örnek sayısını, M parametre sayısını gösterecek şekilde verilen

$$\sigma_j = olasılık(x_j = 1) = \frac{\sum_{z=1}^Z x_{z,j}}{Z} \quad (3.1)$$

$$j \in \{1, 2, \dots, M\}$$

eşitliğiyle, 1 değerini almış parametrelerin toplam sayısının örnek sayısına bölünmesiyle bulunur.

Önerilen İYAKO'da işçi arılar için başlangıç değişkenleri x_{ij} , (0,1) sürekli aralığından değer alacak şekilde, AS ve ÜS sırasıyla parametrelerin alt ve üst sınırları, N yiyecek kaynağı sayısı ve M parametre sayısı olmak üzere

$$x_{ij}^{sürekli} = AS_j + rassal(0,1) * (ÜS_j - AS_j) \quad (3.2)$$

$$i = 1, \dots, N$$

$$j = 1, \dots, M$$

fonksiyonu ile rassal olarak atanır. Değişkenler (0,1) aralığında olduğundan sigmoid fonksiyonu gibi sürekli değişkenleri (0,1) aralığına çeken bir dönüştürücü fonksiyona ihtiyaç duyulmamıştır.

Yeni kaynakların bulunmasında

$$v_{ij}^{sürekli} = x_{ij}^{sürekli} + rassal(-1,1)(x_{ij}^{sürekli} - x_{kj}^{sürekli}) \quad (3.3)$$

fonksiyonu kullanılır. Burada, $k \in \{1, \dots, N\}$ ve $k \neq i$ durumu geçerlidir. Komşu kaynak olan k ve değiştirilecek parametre değeri olan j , rassal olarak seçilir. Eğer $v_{ij}^{sürekli}$ değeri (0,1) aralığının dışında kalırsa,

$$\begin{aligned} \text{eğer } v_{ij}^{sürekl} > 1, \text{ ise } v_{ij}^{sürekl} &= 1 \\ \text{eğer } v_{ij}^{sürekl} < 0, \text{ ise } v_{ij}^{sürekl} &= 0 \end{aligned} \quad (3.4)$$

kullanılarak aralığın içine taşınır. i kaynağındaki j 'nci parametre olan x_{ij} yerine v_{ij} alınarak, i kaynağına alternatif bir diğer potansiyel kaynak üretilir.

Bir sonraki adımda uygunluk değerlerini,

$$\lambda_k(x_i) = \frac{1}{1 + e^{\left[\sum_{j=1}^J (A2)_{j,k} \left(\frac{1}{1 + e^{\left[\sum_{l=1}^I x_l(A1)_{l,j} \right]}} \right) \right]}} \quad (3.5)$$

şeklindeki amaç fonksiyonu ifadesi ile hesaplayabilmek için sürekli değişkenler eşik değere göre ikili hale dönüştürülür. Bu eşik değer girdi değişkeni parametrelerinin 1 olma olasılığı olarak seçilmiştir. Eşik değer τ_j , Denklem (3.1) ile bulunan σ_j değerini 1'den çıkararak hesaplanır.

$$\tau_j = 1 - \sigma_j$$

Sürekli halde bulunan girdi değerleri eşik değeri kullanılarak

$$x_{ij}^{[0,1]} = \begin{cases} 1, & \text{eğer } \tau_j \leq x_{ij}^{sürekl} \\ 0, & \text{eğer } \tau_j > x_{ij}^{sürekl} \end{cases} \quad (3.6)$$

ile $[0,1]$ şeklinde ikili haline getirilir.

Uygunluk değerleri, Denklem (3.5) ile belirtilen amaç fonksiyonu ile hesaplanan i kaynağı ve i kaynağına alternatif olarak oluşturulan v_{ij} değerini almış potansiyel kaynak arasında, uygunluk değerinin büyüklüğüne göre bir obur seçim yapılır. Eğer obur seçim sonucunda i kaynağı aynı kalırsa, i kaynağının iyileştirme sayacı 1 arttırılır.

İşçi arılar için en iyi kaynak obur seçim yöntemiyle belirlendikten sonra, her bir işçi arının kaynağının uygunluk değeri f_i olacak şekilde, Denklem (2.9) ile verilen

$$p_i = \frac{f_i}{\sum_{k=1}^N f_k}$$

eşitliğiyle bulunan, seçilme olasılıkları hesaplanır.

Her bir gözcü arı, işçi arıların kaynaklarının p_i değerine göre, kendine iyileştirilecek bir kaynak seçer. Bu iyileştirme için, Denklem (3.3) ile verilen

$$v_{ij}^{süreklî} = x_{ij}^{süreklî} + rassal(-1,1)(x_{ij}^{süreklî} - x_{kj}^{süreklî})$$

ifadesi kullanılır. Burada, $k \in \{1, \dots, N\}$ ve $k \neq i$ durumu geçerlidir. Komşu kaynak olan k ve değiştirilecek parametre değeri olan j , rassal olarak seçilir. Eğer $v_{ij}^{süreklî}$ değeri (0,1) aralığının dışında kalırsa, Denklem (3.6) ile [0,1] haline getirilir.

Gözcü arının oluşturduğu kaynağın uygunluk değeri Denklem (3.5)'teki amaç fonksiyonu ile hesaplanır. Uygunluk değerine göre, gözcü arının kaynağı ve seçtiği işçi arının kaynağı arasında bir obur seçim yapılarak, işçi arı için en iyi kaynak belirlenir.

En son olarak, iyileştirme sayacı, is_i , önceden belirlenen limit değeri aşmış işçi arı, kâşif arıya dönüşür ve Denklem (3.2) ile sıfırdan rassal yiyecek kaynağı yeri bulur. Limiti aşmış işçi arının kaynağı ile kâşif arının bulduğu kaynak yer değiştirir.

Önerilen algoritma yapısında kural çıkarımı için Denklem (3.5)'teki amaç fonksiyonu ile hesaplanan uygunluğa da bir eşik değer atanır. Her tekrarda, eşik değerin üzerinde kalan uygunluk değerine sahip çözümler kural olarak tanımlanmak üzere saklanır. Bu çalışma için eşik değer olarak 0,9 seçilmiştir. İYAKO algoritmasının güvenilir şekilde çalıştığı literatürdeki ikili kıyas fonksiyonları kullanılarak test edilmiştir. Kullanılan ikili kıyas fonksiyonlarına ait detaylı bilgi EK 6'de verilmiştir.

İYAKO algoritması sonucu saklanan çözümlerin her biri potansiyel bir kuralı temsil etmektedir ve kullanılan veri kümesinin yapısına bağlı olarak sayıca fazla kural içeren durumlar ortaya çıkmaktadır. Ayrıca kural kümesindeki kuralların ağırlıkları da birbiriyle

aynı değildir, veri kümesinin zayıf şekilde temsil eden kurallar da bulunmaktadır. Veri kümesini en iyi şekilde temsil edecek kaliteli kuralları bulmak için bir filtreleme işlemi uygulanmıştır. Filtreleme işlemi Brazdil ve Torgo (1990) tarafından önerilen,

$$Kalite(kural) = Tutarlılık(kural) * e^{Kapsama(kural)-1} \quad (3.7)$$

ifadesiyle verilen kalite fonksiyonu yardımıyla yapılmaktadır. Kalite hesabında kullanılan tutarlılık fonksiyonu

$$Tutarlılık(kural) = \frac{\alpha \cap \beta}{\alpha} \quad (3.8)$$

ifadesiyle, kapsama fonksiyonu ise

$$Kapsama(kural) = \frac{\alpha \cap \beta}{\beta} \quad (3.9)$$

eşitliğiyle hesaplanır. Buradaki α kuralın koşullarına uyan veri sayısını, β kuralın ait olduğu sınıftaki toplam veri sayısını, $\alpha \cap \beta$ ise koşullara uyan kural sınıfındaki veri sayısını temsil eder. İYAKO algoritmasıyla elde edilen potansiyel kurallar kalite değerlerine göre büyükten küçüğe doğru sıralanır ve kesin kural olarak atanmaya başlanır. Kural olarak atanan bir çözümün temsil ettiği veriler eğitim kümesinden silinir. Son olarak oluşturulan kural kümesinin test doğruluğu, veri kümesi üzerinden hesaplanarak performans değerleri elde edilir.

Doğruluk hesabında,

$$Doğruluk = \frac{\text{Gerçek pozitif sayısı} + \text{Gerçek negatif sayısı}}{\text{Gerçek pozitif} + \text{Sahte pozitif} + \text{Gerçek negatif} + \text{Sahte negatif}} \quad (3.10)$$

ifadesi kullanılır.

Şekil 3.2’de İYAKO akış diyagramı sunulmuştur. Aşağıda, İYAKO algoritmasının adımları verilmiştir:

Başla

- 1) Veriyi ikili hale getir
- 2) Veri kümesi üzerinde YSA’yı eğit ve ağ ağırlıklarını sakla
- 3) İYAKO parametre değerlerini belirle
- 4) Kural olarak saklanacak çözümler için uygunluk eşik değerini $\alpha = 0,9$ olarak ata
- 5) Önsel (apriori) olasılık değeri olarak, ikili hale dönüştürme eşik değeri, τ_j ’yi hesapla (Denklem (3.1))
- 6) İşçi arılar için başlangıç popülasyonunu oluştur (Denklem (3.2))

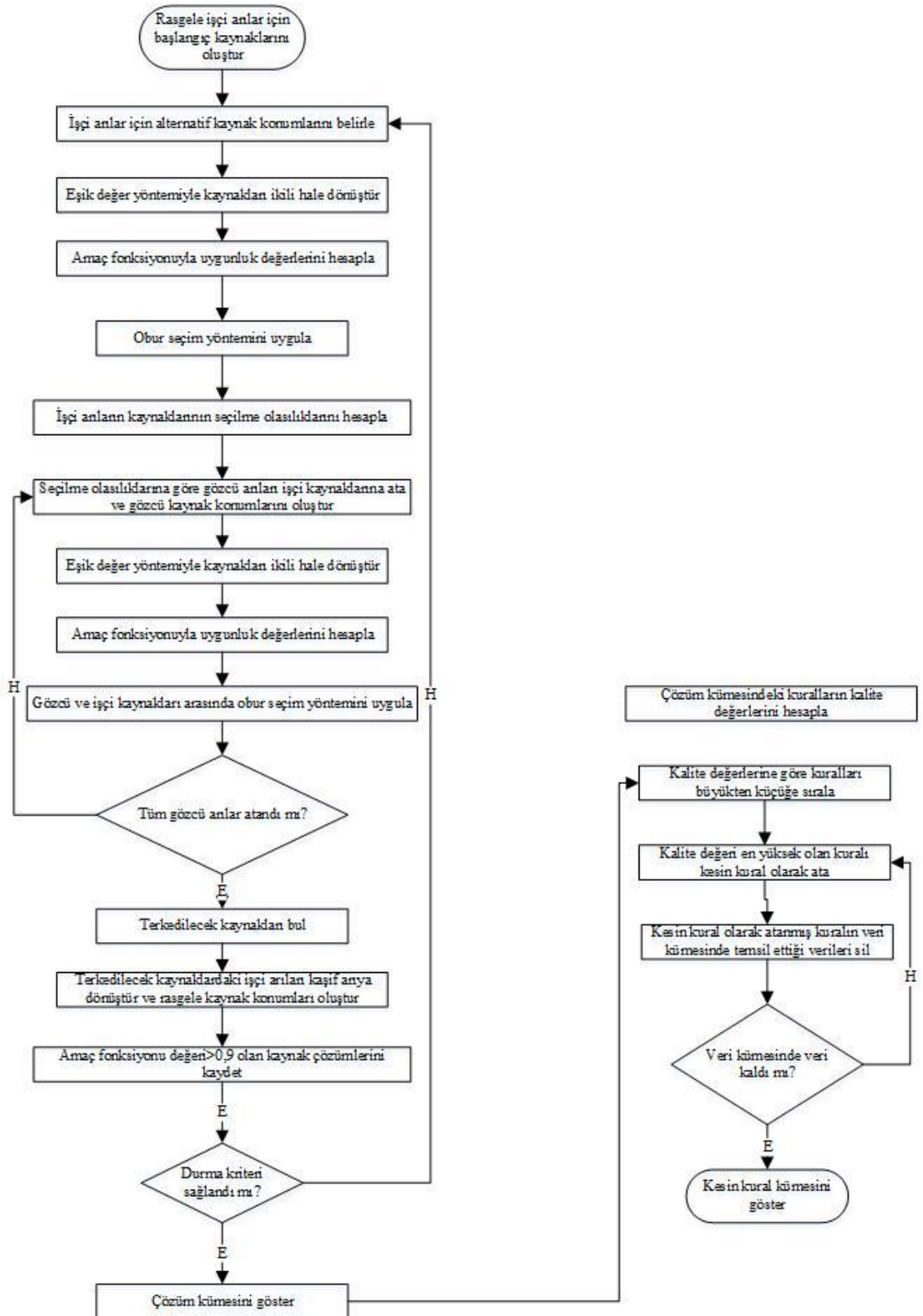
Tekrarla

- 1) İşçi arılar için potansiyel yeni kaynaklar üret (Denklem (3.3))
- 2) Sürekli çözümleri ikili hale getir (Denklem (3.6))
- 3) İşçi arı kaynakları ve potansiyel kaynaklar için uygunluk değerlerini amaç fonksiyonundan hesapla (Denklem (3.5))
- 4) İşçi arı kaynakları ve potansiyel kaynaklar arasında uygunluk değerine göre obur seçim uygula
- 5) Potansiyel kaynak tarafından iyileştirilemeyen işçi kaynaklarının iyileştirme sayacılarını 1 arttır
- 6) Gözcü arının işçi arıyı seçme olasılığı p_i ’yi, işçi kaynaklarının uygunluk değerlerine göre hesapla
- 7) Gözcü arılar için rulet tekerleği yöntemiyle p_i değerlerine göre işçi kaynağı seç
- 8) Gözcü arı için seçtiği işçi arıya bağlı olarak yeni bir çözüm oluştur
- 9) Sürekli çözümleri ikili hale getir
- 10) Gözcü arılar için uygunluk değerlerini amaç fonksiyonu ile hesapla
- 11) Gözcü arı ve seçtiği işçi arı arasında uygunluk değerine göre obur seçim yap
- 12) İyileştirme sayacı limit değere ulaşmış işçi arıları kâşif arıya dönüştür ve sıfırdan çözüm üret
- 13) Amaç fonksiyonu değeri $0,9$ ’dan büyük olan çözümleri sakla

Bitir

Kural Çıkarımı

- 1) Çözüm kümesindeki kuralların kalitelerini hesapla (Denklem (3.7))
- 2) Çözüm kümesini kalite değerlerine göre azalan şekilde sırala
- 3) Kalite değeri en büyük çözümden başlayarak, veri kümesinde temsil edilecek veri kalmayıncaya kadar, çözümleri kural olarak ata ve kural olarak atanmış çözümün temsil ettiği verileri kümeden sil

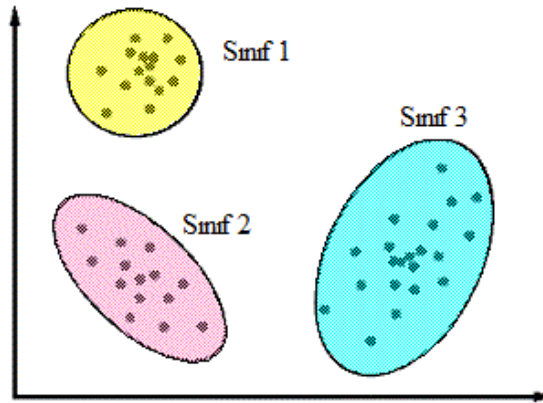


Şekil 3.2. İYAKO sınıflandırıcı için akış diyagramı

3.4. Kümeleme Analizi

İYAKO algoritmasından elde edilen kurallar veri kümesindeki tüm var olan ve yeni karşılaşılabilecek veriyi açıklayıcı niteliğe sahip olduğundan sayıca fazladır. Bu sayıyı azaltmak ve veri kümesini genel olarak ifade eden kurallar bulmak amacıyla İYAKO ile elde edilen kurallara kümeleme analizi yapılmıştır.

Kümeleme analizi her bir sınıf için, o sınıfta bulunan verilerin ortak özelliklerinden yola çıkarak sınıfları temsil eden veri kümeleri oluşturur. Aynı kümenin elemanı olan nesneler birbirleriyle diğer kümelerin elemanı olan nesnelerle olduğundan daha çok benzerlik gösterir. Aynı kümedeki verilerin ortak öznitelikleri o sınıf için genel kuralları oluşturur. Öznitelik özelliklerine göre nesnelerin oluşturduğu sınıf kümelerinin görseli Şekil 3.2 ile verilmiştir.



Şekil 3.3. 3 sınıflı bir kümeleme analizi örneği

Kümeleme analizi için, İYAKO uygulanarak elde edilen filtreleme işleminden geçmemiş kurallar, çıktı sınıflarına göre ayrılır. Tüm sınıflar için her bir girdi parametresinin 1 değerini aldığı kuralların toplam sayısının sınıfa ait kural sayısına oranı m_i ,

$$m_j = \frac{\sum_{j=1}^Z x_j}{Z} \quad x_j = 1$$
$$j = 1 \dots M$$

eşitliğiyle, tüm j parametreleri için hesaplanır. Burada Z , sınıfa ait toplam kural sayısı, N ise parametre sayısıdır. Eğer

$$m_j > \%60$$

koşulu sağlanırsa, i parametresi, sınıfı temsil eden aday kural koşulu olarak eklenir. 10 kere çalıştırılan İYAKO algoritmasında her bir çalıştırmada aday kural koşulu olarak eklenen parametre değeri, 10 çalıştırmanın %80'inden fazlasında bulunuyorsa kesin kural koşulu olarak atanır. Bu işlem sonunda sınıfları temsil eden kümeler ve genel kurallar oluşturulur. Ayrıca aday kural koşulları belirlenirken yüzde değeri en yüksek olan koşul parametresi sınıfı temsil etmede ana bileşen olarak belirlenmiştir. Ana bileşen parametresinin aldığı değere göre verinin sınıfının tayin edilme durumu incelenmiştir.

Bir sonraki adımda girdi parametreleri cinsinden $[0,1]$ olarak ifade edilen kurallar “eğer-ise” cinsinden anlaşılabilir sözlü kurallara dönüştürülür. Kuralların “eğer” kısmı, özniteliklerin parametre koşullarını, “ise” kısmı ise ait oldukları sınıfı belirtir. Aynı öznitelige ait parametreler “veya”, farklı öznitelige ait parametreler ise “ve” bağlacıyla ayrılarak yazılı hale dönüştürülür.

4. UYGULAMA VE BULGULAR

4.1. Kullanılan Veri Kümeleri

Önerilen algoritmanın performansını ölçmek için UCI (University of California at Irvine) makine öğrenimi veri tabanından¹ alınan 8 adet veri kümesi kullanılmıştır. Kullanılan veri kümelerine dair detaylı bilgi EK 1’de verilmiştir.

Sürekli veri içeren kümelere WEKA² programında denetimli ayırıklaştırma uygulanmış ve elde edilen aralık değerlerine göre veri ikili hale dönüştürülmüştür. WEKA programı denetimli ayırıklaştırma yöntemi olarak “*entropi-tabanlı ayırıklaştırma*” yı kullanır (Fayyad ve Irani 1993). Entropi-tabanlı ayırıklaştırma yönteminde, sınıf bilgisine bağlı olarak aralıklar oluşturulur. Veri aralıklarına düşen örneklerin ait olduğu sınıfın saflık derecesinin en yüksek olması amaçlanır. Denetimli ayırıklaştırma uygulanmış ve ikili hale dönüştürülmüş veri kümelerinin detayları EK 2’de sunulmuştur.

Veri kümeleri için doğruluk hesabı yapılırken veri sayısı 150’den büyük olan kümeler için 10-kat çapraz geçişleme yöntemi, 150’den küçük olan kümeler içinse %66 eğitim, %33 test bölünmesi uygulanmıştır.

4.2. Kural Çıkarımı Sonuçları

MATLAB ile eğitilen YSA’lardan elde edilen ağırlık değerleri İYAKO algoritmasında kullanılarak λ_k değeri enbüyüklenmeye çalışılmıştır. İYAKO kontrol parametreleri literatürde YAK için önerilen şekilde kabul edilen değerler (Karaboga ve Basturk 2008) Çizelge 4.1 ile verilmiştir.

¹ <https://archive.ics.uci.edu/ml/datasets.html>

² <http://www.cs.waikato.ac.nz/ml/weka/index.html>

Çizelge 4.1. İYAKO kontrol parametre değerleri

Tekrar sayısı	Arı sayısı	İyileştirme limiti
1000	Veri kümesi boyutu*2	Veri kümesi boyutu ²

MATLAB ortamında kodlanan İYAKO algoritması, her bir veri kümesi için 10 kez çalıştırılmıştır. Kural çıkarımından sonra, veri madenciliği için sıklıkla kullanılan özdevinimli öğrenme yöntemleri olan C4.5, NBTree, DecisionTable ve PART algoritmaları, her bir veri kümesi için WEKA programında 10 kez çalıştırılmıştır. Elde edilen doğruluk değerleri ve kural sayılarına ait bilgiler Çizelge 4.2’de sunulmuştur.

Çizelge 4.2. İYAKO performans değerlendirmesi

Veri kümesi			J48 (C4.5)	Decision Table	NBTree	PART	İYAKO sınıflandırıcı
BCW	Doğruluk	Minimum	93,7%	91,2%	94,1%	93,3%	94,3%
		Maksimum	95,8%	96,2%	98,3%	97,1%	98,4%
		Ortalama	94,4%	94,0%	96,0%	95,3%	96,0%
		Standart sapma	0,7%	1,6%	1,5%	1,4%	2,3%
	Kural sayısı	Minimum	6	18	1	6	13
		Maksimum	14	47	6	11	17
		Ortalama	9,3	30,1	3,4	8,1	14,8
		Standart sapma	2,7	10,1	1,9	1,6	2,0
Iris	Doğruluk	Minimum	92,2%	90,2%	90,2%	92,2%	97,1%
		Maksimum	98,0%	98,0%	98,0%	98,0%	100,0%
		Ortalama	95,0%	94,1%	93,7%	94,9%	98,9%
		Standart sapma	3,0%	2,6%	2,4%	3,0%	1,6%
	Kural sayısı	Minimum	3	3	1	3	4
		Maksimum	5	3	4	6	7
		Ortalama	4,0	3,0	2,1	3,8	5,4
		Standart sapma	0,8	0,0	1,2	1,1	1,5
Monk	Doğruluk	Minimum	57,1%	57,1%	78,6%	64,3%	84,1%
		Maksimum	85,7%	83,3%	92,9%	100,0%	100,0%
		Ortalama	72,7%	72,7%	84,8%	89,5%	92,6%
		Standart sapma	9,0%	7,8%	4,4%	11,8%	7,1%
	Kural sayısı	Minimum	5	8	3	8	9
		Maksimum	22	30	5	10	14
		Ortalama	13,5	14,5	4,0	8,5	11,6
		Standart sapma	4,2	8,4	0,9	0,7	2,3

Çizelge 4.2. İYAKO performans değerlendirmesi (devamı)

Nursery	Doğruluk	Minimum	95,8%	93,9%	95,2%	97,9%	94,3%
		Maksimum	96,6%	94,4%	97,0%	99,1%	96,1%
		Ortalama	96,2%	94,2%	96,4%	98,7%	95,0%
		Standart sapma	0,3%	0,2%	0,5%	0,4%	1,0%
	Kural sayısı	Minimum	243	810	75	131	189
		Maksimum	313	810	167	186	216
		Ortalama	280,2	810,0	135,8	159,1	200,4
		Standart sapma	18,4	0,0	25,4	15,1	12,8
Sunburn	Doğruluk	Minimum	33,3%	0,0%	66,7%	33,3%	66,7%
		Maksimum	66,7%	66,7%	100,0%	66,7%	100,0%
		Ortalama	58,3%	48,3%	83,3%	58,3%	87,1%
		Standart sapma	14,2%	22,8%	17,6%	14,2%	3,1%
	Kural sayısı	Minimum	1	1	1	1	3
		Maksimum	3	2	1	2	5
		Ortalama	2,1	1,3	1,0	1,8	3,6
		Standart sapma	0,7	0,5	0,0	0,4	0,9
Titanic	Doğruluk	Minimum	76,7%	76,2%	76,6%	77,7%	74,9%
		Maksimum	80,9%	80,7%	79,7%	80,9%	84,0%
		Ortalama	78,5%	78,6%	78,1%	79,0%	79,3%
		Standart sapma	1,3%	1,3%	1,0%	1,0%	4,9%
	Kural sayısı	Minimum	5	2	1	3	4
		Maksimum	10	14	5	7	8
		Ortalama	7,3	11,9	2,1	5,4	5,8
		Standart sapma	2,3	3,9	1,6	2,1	1,6
Weather	Doğruluk	Minimum	25,0%	25,0%	0,0%	25,0%	42,9%
		Maksimum	60,0%	75,0%	80,0%	66,7%	71,4%
		Ortalama	51,5%	58,2%	59,2%	53,2%	54,3%
		Standart sapma	11,6%	15,2%	23,0%	12,5%	15,6%
	Kural sayısı	Minimum	1	1	1	1	3
		Maksimum	4	2	1	3	7
		Ortalama	2,0	1,3	1,0	2,1	6,2
		Standart sapma	1,1	0,5	0,0	0,7	1,5
Wine	Doğruluk	Minimum	85,0%	78,7%	93,3%	85,3%	93,8%
		Maksimum	95,1%	93,4%	98,4%	95,1%	98,3%
		Ortalama	90,8%	88,2%	95,6%	91,1%	95,1%
		Standart sapma	3,7%	5,4%	1,9%	3,3%	1,8%
	Kural sayısı	Minimum	4	14	1	3	5
		Maksimum	6	29	5	5	7
		Ortalama	5,2	20,3	2,3	4,3	5,8
		Standart sapma	0,6	4,4	1,3	0,7	1,1

İYAKO algoritmasının performansının diğer özdevinimli yöntem performanslarına göre farklılığının istatistiksel olarak anlamlılığı için, MINITAB programında, algoritmalar arasında %95 güvenirlilikte t- testi yapılmıştır. Yapılan test için kullanılan hipotezler

$$\begin{aligned} \text{Sıfır hipotezi:} & \quad H_0 : \mu_1 = \mu_2 \\ \text{Alternatif hipotez:} & \quad H_1 : \mu_1 > \mu_2 \end{aligned}$$

şeklindedir. MINITAB’de yapılan t-testinin p değerleri Çizelge 4.3’te sunulmuştur. Çizelge 4.3’te altı çizili durumlarda doğruluk testleri için μ_1 olarak özdevinimli öğrenme yöntemi, siyah durumlarda ise μ_1 olarak İYAKO alınmıştır. Altı çizili durumlarda ise μ_1 olarak özdevinimli öğrenme yöntemini kullanmanın sebebi, İYAKO algoritmasının doğruluğu bu durumlarda diğer yöntemlere göre düşük çıktığından, diğer yöntemlerin yüksek performansın geçerliliğini test etmektir. Kalın punto ile işaretlenmiş durumlar, sıfır hipotezinin reddedilerek alternatif hipotezin kabul edildiği durumlardır. Yani bu durumlarda μ_1 değeri, μ_2 değerine göre daha yüksek çıkmıştır. İYAKO algoritması, BCW veri kümesinde J48 ve DecisionTable, Iris kümesinde tüm diğer yöntemler, Monk kümesinde J48, DecisionTable ve NBTree, Nursery kümesinde DecisionTable, Sunburn kümesinde J48, DecisionTable ve PART, Wine kümesinde J48, DecisionTable ve PART algoritmalarına göre daha üstün bir doğruluk oranı göstermiştir. Sadece Nursery veri kümesinde J48, NBTree ve PART algoritmalarının doğruluğu İYAKO’ya göre daha fazladır. Diğer durumlar içinse p değerlerine göre yeterince kanıt yoktur. Kural sayısına bakılacak olur ise İYAKO algoritması genellikle diğer yöntemlere göre daha fazla sayıda kural bulmaktadır.

Çizelge 4.3. İYAKO ve özdevinimli öğrenme yöntemlerinin performans hipotezlerinin p- değerleri

p-değerleri		Decision			
		J48(C4.5)	Table	NBTree	PART
Doğruluk	BCW	0,0330	0,0200	0,5180	0,2490
	Iris	0,0010	0,0000	0,0000	0,0010
	Monk	0,0000	0,0000	0,0050	0,2460
	Nursery	0,0020	0,0130	0,0010	0,0000
	Sunburn	0,0000	0,0000	0,2620	0,0000
	Titanic	0,3260	0,3380	0,2320	0,4480
	Weather	0,3270	<u>0,2910</u>	<u>0,2940</u>	0,4300
	Wine	0,0030	0,0010	<u>0,2910</u>	0,0020
Kural Sayısı	BCW	0,0000	<u>0,0010</u>	0,0000	0,0000
	Iris	0,0120	0,0000	0,0000	0,0090
	Monk	<u>0,1160</u>	<u>0,1590</u>	0,0000	0,0010
	Nursery	<u>0,0000</u>	<u>0,0000</u>	0,0000	0,0000
	Sunburn	0,0000	0,0000	0,0000	0,0000
	Titanic	<u>0,0540</u>	<u>0,0000</u>	0,0000	0,3190
	Weather	0,0000	0,0000	0,0000	0,0000
	Wine	0,0780	<u>0,0000</u>	0,0000	0,0010

4.3. Kümeleme Analizi Sonuçları

Kümeleme analizi sonucu elde edilen kuralların diğer kıyaslama için kullanılan özdevinimli öğrenme yöntemleriyle olan performans karşılaştırmaları Çizelge 4.4’de verilmiştir. Çizelge 4.4’te verilen değerler, özdevinimli öğrenme yöntemlerinin kümeleme analizi olmaksızın WEKA algoritmasında çalıştırmasından elde edilen sonuçları içermektedir. Burada sadece İYAKO algoritmasıyla kümeleme analizi yapılmıştır. İYAKO algoritmasındaki kural sayısını azaltmak için kümeleme analizinin göstereceği performans incelenmiştir.

Çizelge 4.4. Özdevinimli öğrenme yöntemleri ve İYAKO ile kümeleme kural çıkarımı sonuçları

Veri kümesi			J48(C4.5)	Decision Table	NBTree	İYAKO	
						PART	kümeleyici
BCW	Doğruluk	Minimum	93,7%	91,2%	94,1%	93,3%	56,3%
		Maksimum	95,8%	96,2%	98,3%	97,0%	100,0%
		Ortalama	94,4%	94,0%	96,0%	95,3%	92,3%
		Standart sapma	0,7%	1,6%	1,5%	1,4%	10,0%
	Kural sayısı	Minimum	6	18	1	6	2
		Maksimum	14	47	6	11	19
		Ortalama	9,3	30,1	3,4	8,1	10,5
		Standart sapma	2,7	10,0	1,9	1,6	12,0
Iris	Doğruluk	Minimum	92,2%	90,2%	90,2%	92,2%	90,7%
		Maksimum	98,0%	98,0%	98,0%	98,0%	100,0%
		Ortalama	95,0%	94,1%	93,7%	95,0%	97,7%
		Standart sapma	3,0%	2,6%	2,4%	3,0%	3,3%
	Kural sayısı	Minimum	3	3	1	3	3
		Maksimum	5	3	4	6	7
		Ortalama	4,0	3,0	2,1	3,8	5,0
		Standart sapma	0,8	0,0	1,2	1,1	2,9
Monk	Doğruluk	Minimum	57,1%	57,1%	78,6%	64,3%	100,0%
		Maksimum	85,7%	83,3%	92,9%	100,0%	100,0%
		Ortalama	72,7%	72,7%	84,8%	89,5%	100,0%
		Standart sapma	9,0%	7,8%	4,4%	11,8%	0,0%
	Kural sayısı	Minimum	5	8	3	8	2
		Maksimum	22	30	5	10	2
		Ortalama	13,50	14,50	4,00	8,50	2,0
		Standart sapma	4,20	8,42	0,94	0,71	0,0
Nursery	Doğruluk	Minimum	95,8%	93,9%	95,2%	98,0%	52,0%
		Maksimum	96,6%	94,4%	97,0%	99,1%	100,0%
		Ortalama	96,2%	94,2%	96,4%	98,7%	67,3%
		Standart sapma	0,3%	0,2%	0,5%	0,4%	22,2%
	Kural sayısı	Minimum	243	810	75	131	3
		Maksimum	313	810	167	186	7
		Ortalama	280,2	810,0	135,8	159,1	5,0
		Standart sapma	18,4	0,0	25,4	15,1	2,8
Sunburn	Doğruluk	Minimum	33,3%	0,0%	66,7%	33,3%	60,0%
		Maksimum	66,7%	66,7%	100,0%	66,7%	100,0%
		Ortalama	58,3%	48,3%	83,3%	58,3%	93,3%
		Standart sapma	14,2%	22,8%	17,6%	14,2%	16,3%
	Kural sayısı	Minimum	1	1	1	1	2
		Maksimum	3	2	1	2	6
		Ortalama	2,1	1,3	1,0	1,8	4,0
		Standart sapma	0,7	0,5	0,0	0,4	2,8

Çizelge 4.4. Özdevinimli öğrenme yöntemleriyle ve İYAKO ile kümeleme kural çıkarımı sonuçları (devamı)

Titanic	Doğruluk	Minimum	76,6%	76,2%	76,6%	77,7%	37,6%
		Maksimum	80,9%	80,7%	79,7%	80,9%	75,9%
		Ortalama	78,5%	78,6%	78,0%	79,0%	61,2%
		Standart sapma	1,3%	1,3%	1,0%	1,0%	14,2%
	Kural sayısı	Minimum	5	2	1	3	2
		Maksimum	10	14	5	7	6
		Ortalama	7,3	12,0	2,1	5,4	4,0
		Standart sapma	2,3	3,9	1,6	2,1	2,8
Weather	Doğruluk	Minimum	25,0%	25,0%	0,0%	25,0%	85,7%
		Maksimum	60,0%	75,0%	80,0%	66,7%	100,0%
		Ortalama	51,5%	58,2%	59,2%	53,2%	95,2%
		Standart sapma	11,6%	15,2%	23,0%	12,5%	8,3%
	Kural sayısı	Minimum	1	1	1	1	1
		Maksimum	4	2	1	3	3
		Ortalama	2,0	1,3	1,0	2,1	2,0
		Standart sapma	1,1	0,5	0,0	0,7	1,4
Wine	Doğruluk	Minimum	85,0%	78,7%	93,3%	85,3%	30,4%
		Maksimum	95,1%	93,4%	98,4%	95,1%	100,0%
		Ortalama	90,8%	88,2%	95,6%	91,2%	76,00%
		Standart sapma	3,7%	5,4%	1,9%	3,3%	25,4%
	Kural sayısı	Minimum	4	14	1	3	3
		Maksimum	6	29	5	5	18
		Ortalama	5,2	20,3	2,3	4,3	10,5
		Standart sapma	0,6	4,4	1,3	0,7	10,6

Kümeleme analizi sonucunda belirlenen ana bileşenler kümeler bazında Çizelge 4.5 ile özetlenmiştir. İYAKO algoritması sona erdikten sonra yapılan kümeleme analizinde çıkarılan kuralların doğruluk oran ve grafikleriyle ana bileşen sınıflandırma doğrulukları EK 7’ de detaylı olarak verilmiştir.

Çizelge 4.5. Kullanılan veri kümelerinin ana bileşen analizi

Veri kümesi	<i>BCW</i>	<i>Iris</i>	<i>Monk</i>	<i>Nursery</i>
Ana bileşen	Mitoz	Taç yaprak genişliği	Ceket rengi	Sağlık durumu
Veri kümesi	<i>Sunburn</i>	<i>Titanic</i>	<i>Weather</i>	<i>Wine</i>
Ana bileşen	Güneş kremi	Cinsiyet	Nem	Flavanoidler

Çizelge 4.6. İYAKO kümeleyici ve özdevinimli öğrenme yöntemlerinin performans hipotezlerinin p- değerleri

p-değerleri		Decision			
		J48(C4.5)	Table	NBTree	PART
Doğruluk	BCW	<u>0,1930</u>	<u>0,3640</u>	<u>0,0680</u>	<u>0,1060</u>
	Iris	0,0460	0,0170	0,0100	0,0460
	Monk	0,0000	0,0000	0,0000	0,0100
	Nursery	<u>0,0070</u>	<u>0,0090</u>	<u>0,0070</u>	<u>0,0050</u>
	Sunburn	0,0010	0,0000	0,1370	0,0010
	Titanic	0,0150	0,0150	0,0170	0,0140
	Weather	0,0010	0,0010	0,0010	0,0010
	Wine	<u>0,0130</u>	<u>0,0320</u>	<u>0,0020</u>	<u>0,1200</u>
# Kurallar	BCW	0,4560	<u>0,1240</u>	0,2790	0,4130
	Iris	0,3540	0,2500	0,1950	0,3300
	Monk	0,0000	<u>0,0010</u>	0,0000	0,0000
	Nursery	0,0000	0,0000	0,0000	0,0000
	Sunburn	0,3300	0,2030	0,1870	0,2350
	Titanic	<u>0,1820</u>	<u>0,0920</u>	0,2630	<u>0,3130</u>
	Weather	0,5000	0,3070	0,2500	<u>0,4690</u>
	Wine	0,3040	<u>0,2360</u>	0,2800	0,2110

Çizelge 4.6'ya bakıldığında kümeleme analizi sonucu ortaya çıkan kuralların Iris, Monk, Sunburn, Titanic ve Weather kümelerinde diğer yöntemlere göre daha yüksek doğruluk oranına sahip olduğu görülmektedir. Nursery ve Wine kümelerinde ise karşılaştırılan yöntemler daha üstün bir doğruluk performansı göstermiştir. BCW veri kümesi için hangi yöntemin daha iyi olduğunda dair bir kanıt bulunamamıştır. Kural sayıları incelendiğinde ise Monk ve Nursery veri kümelerinde kümeleme analizi sonucunda diğer yöntemlere göre çok daha az sayıda kural bulunmuştur. Diğer durumlarda ise kümeleme analizinin daha fazla sayıda kural çıkardığında dair bir kanıt yoktur.

5. TARTIŞMA VE SONUÇ

Bu çalışmada, veri kümelerini temsil edecek kurallar çıkarmayı amaçlayan ikili yapay arı kolonisine dayanan İYAKO algoritması önerilmiştir. Önerilen algoritma hem YSA'ların hem de YAK algoritmasının yüksek performansından yararlanmaktadır. İYAKO algoritmasıyla YSA'ların sağladığı yüksek performanslı sonuçların kurallara dönüştürülerek daha açıklayıcı ve kullanıcı için anlaşılabilir hale gelmesi sağlanmıştır. Aynı zamanda kümeleme analizi de gerçekleştirilerek, veri kümelerinin kendi içinde gösterdiği ilişkiler ve sınıflandırmada önemli olan bileşenler açığa çıkarılmıştır. Performans değerlendirmesinden sonra İYAKO algoritmasının ve önerilen yöntemin veri madenciliği için başarılı olduğu görülmektedir.

Önerilen İYAKO algoritmasında, YSA'lar veri kümeleri üzerinde eğitildikten sonra ikili değişkenler ile çalışan yapay arı kolonisi algoritması ile “eğer-ise” yapısında, veri kümesini temsil eden kuralların çıkarımı yapılmıştır. Kural çıkarımında sonra kümeleme analizi gerçekleştirilerek verilerin kendi arasında gösterdiği ilişkiler saptanmış ve sınıfları gruplandıracak az sayıda kurala ulaşılmıştır.

Elde edilen sonuçlar doğrultusunda, önerilen yöntemin “Big Data” (Büyük Veri) olarak adlandırılan ve kullanıcı açısından anlaşılabilir durumda olmayan veri kümelerinden açıklayıcı kuralları çıkarabildiği görülmüştür. Böyle bir yöntem, geniş veri depolarıyla çalışan birçok alanda uygulama şansı bulmaktadır. Temel uygulama alanlarına örnek verilecek olursa özellikle, internet üzerinden pazarlamanın önem kazandığı günümüzde, müşteri profili oluşturmada yarar sağlayabilecek bir yöntemdir. Müşterilerin alışkanlıklarını kaydeden iyi düzenlenmiş bir veri tabanı kullanılarak, bir sonraki satın alınacak ürün belirlenebilir. Bilimsel çalışmalar açısından ise özellikle tıp alanında, çeşitli hastalıkların gösterdiği semptom verileri kullanılarak hastalıkların tanısıyla birlikte hastalık tiplerine dair çıkarımlar için, önerilen algoritma fayda sağlayıcı özelliğe sahiptir.

Bundan sonraki çalışmalarda önerilen yöntemin gerçek hayat problemlerinde uygulanması ve algoritma parametrelerinin performans üzerinde etkisinin deneysel olarak incelenmesi hedeflenmektedir.

KAYNAKLAR

- Ahmed, S.R. 2004.** Applications of data mining in retail business. *Information Technology: Coding and Computing*, 2: 455-459.
- Akyol, S., Alataş, B. 2012.** Güncel Sürü Zekâsı Optimizasyon Algoritmaları. *Nevşehir Üniversitesi Fen Bilimleri Dergisi*, 1: 36-50.
- Aliev, R.A., Aliev, R.R., Guirimov, B., Uyar, K. 2008.** Dynamic data mining technique for rules extraction in a process of battery charging. *Applied Soft Computing*, 8(3): 1252-1258.
- Alpaydın, E. 2000.** Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri, Bilişim 2000 Eğitim Semineri, İstanbul, 2000.
- Andrews, R., Diederich, J., Tickle, A.B. 1995.** A survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Syst*, 8(6): 373-389.
- Angeline, P.J., Saunders, G.M., Pollack, J.B. 1994.** An evolutionary algorithm that constructs recurrent neural networks. *Neural Networks, Transactions on*, 5(1): 54-65.
- Arpitam, C., Bipan T., Kanai C.P. 2013.** Binary grayscale halftone pattern generation using binary artificial bee colony (bABC). *Signal, Image and Video Processing*, 7(6): 1195-1209.
- Bao, L., Zeng, J.C. 2009.** Comparison and analysis of the selection mechanism in the artificial bee colony Algorithm, Ninth International Conference on Hybrid Intelligent Systems, 12-14 August, 2009, Shenyang, LiaoNing, China.
- Baykasoglu, A., Ozbakir, L., Tapkan, P. 2007.** Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem: Swarm Intelligence, Focus on Ant and Particle Swarm Optimization, Ed.: Chan, F.T.S., Tiwari, M.K., Itech Education and Publishing, Vienna, Austria, pp: 113-144.
- Bolaji, A.B., Khader, A.T., AlBetar, M.A., Awadallah, M.A. 2013.** Artificial Bee Colony algorithm, its variants and applications: a survey. *Journal of Theoretical and Applied Information Technology*, 47(2): 426-433.
- Borsellino, A., Gamba, A. 1961.** An outline of a mathematical theory of PAPA. *Nuovo Cimento Suppi.* 2, 20: 221-231.
- Braspenning, P.J., Thuijsman, F., Weijters, A.J.M.M. 1995.** Artificial Neural Networks: An Introduction to ANN Theory and Practice. Springer-Verlag, Berlin, Heidelberg, Germany, 293 pp.
- Brazdil, P., Torgo, L. 1990.** Knowledge Acquisition via Knowledge Integration. Current Trends in Knowledge Acquisition, Ed.: Wielinga, B. et al, IOS Press.
- Breiman, L. 2001.** Statistical Modeling: The Two Cultures. *Statistical Science*, 16(3): 199-231.
- Cao, Y., Wu, J. 2002.** Projective ART for clustering data sets in high dimensional spaces. *Neural Netw.*, 15: 105-120.

- Cao, Y., Wu, J. 2002.** Projective ART for clustering data sets in high dimensional spaces. *Neural Networks*, 15: 105-120.
- Carpenter, G.A., Grossberg, S. 1987.** A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37: 54-115.
- Carrier, C.G., Povel, O. 2003.** Characterizing data mining software. *Intelligent Data Analysis*, 7: 181-192.
- Clark, P., Boswell, R. 1991.** Rule Induction with CN2: Some Recent Improvements: Machine Learning – Proceedings of the Fifth European Conference (EWSL-91), Ed.: Kodratoff, Y., Berlin, pp: 151-163.
- Costa, D., Hertz, A. 1997.** Ants can color graphs, *J. Oper. Res. Soc.* 48: 295-305.
- Dorigo M., Maniezzo V., Colorni A. 1991.** Positive Feedback as a Search Strategy, *Teknik Rapor*, N. 91-016, Politecnico di Milano, Italy.
- Dorigo, M., Birattari M., Stutzle, T. 2006.** Ant colony optimization. *Computational Intelligence Magazine*, 1(4): 28–39.
- Dorigo, M., Blum, C. 2005.** Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344: 243-278.
- Dorigo, M., Gambardella, L.M. 1997.** Ant colonies for the travelling salesman problem. *Biosystems*, 43(2): 73-81.
- Fisher, D.L. 1966.** Data, Documentation and Decision Tables. *Comm ACM*, 9(1): 26–31.
- Fürnkranz, J., Gamberger, D., Lavrac, N. 2012.** Foundations of Rule Learning. Springer-Verlag, Berlin Heidelberg, 334 pp.
- Gambardella, L.M., Dorigo, M. 2000.** Ant colony system hybridized with a new local search for the sequential ordering problem, *INFORMS J. Comput.* 12(3): 237-255.
- Geem, Z.W., Kim, J.H., Loganathan, G.V. 2001.** A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2): 60.
- Giles, C.L., Omlin, C.W. 1993.** Extraction, Insertion and Refinement of Symbolic Rules in Dynamically Driven Recurrent Neural Networks. *Connection Sci.*, 5(3/4): 307–328.
- Hassoun, M.H. 1995.** Fundamentals of Artificial Neural Networks. MIT Press, USA, 511 pp.
- Haykin, S. 2009.** Neural Networks and Learning Machines. Pearson Education, Upper Saddle River, New Jersey, USA, 934 pp.
- Hiroyasu, T., Miki, M., Ono, Y., Minami, Y. 2000.** Ant colony for continuous functions. The Science and Engineering, Doshisha University.
- Hühn, J., Hüllermeier, E. 2009.** FURIA: an algorithm for unordered fuzzy rule induction, *Data Mining and Knowledge Discovery*, 19(3): 293-319.
- Hunt, E., Marin, J., Stone, P. 1966.** Experiments in induction. Academic Press Inc., 247 pp.

- Husseinzadeh Kashan, M., Nahavandi, H., Husseinzadeh Kashan, A. 2012.** DisABC: A new artificial bee colony algorithm for binary optimization. *Applied Soft Computing*, 12: 342-352.
- Jaccard, P. 1912.** The distribution of the flora in the Alpine zone. *The New Phytologist*, 11(2): 37-52.
- Jiawei, H., Kamber, M. 2006.** Data Mining: Concepts and Techniques, Second Edition. Morgan Kaufmann Publishers, San Francisco, CA, USA, 743 pp.
- Kakıcı, A. 2009.** Yapay Sinir Ağlarının Mimarisi ve Yapı Elemanları. <http://www.ahmetkakici.com/yapay-sinir-aglari/yapay-sinir-aglarinin-mimarisi-ve-yapi-elemanlari/> - (Erişim tarihi: 16.01.2014)
- Kantardzic, M. 2003.** Data Mining: Concepts, Models, Methods and Algorithms. John Wiley & Sons, Hoboken, NJ, USA, 343 pp.
- Karaboga, D. 2005.** An Idea based on Honey Bee Swarm for Numerical Optimization. Erciyes University, Engineering Faculty, Computer Engineering Department, Technical Report TR06 pp 1-10
- Karaboğa, D. 2005.** An idea based on honey bee swarm for numerical optimization, Technical Report-TR06. Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri, Türkiye.
- Karaboga, D., Akay, B. 2009.** A Survey: Algorithms Simulating Bee Swarm Intelligence, *Artificial Intelligence Review*, 31(1-4), pp: 61-85.
- Karaboga, D., Basturk, B. 2007.** A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. *Journal of Global Optimization*, 39(3): 459-471.
- Karaboga, D., Basturk, B. 2007.** Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. *Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, 4529: 789-798.
- Karaboga, D., Basturk, B. 2007a.** Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. *Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, 4529: 789-798.
- Karaboga, D., Basturk, B. 2007b.** Artificial Bee Colony Algorithm on Training Artificial Neural Networks. 15th. Signal Processing and Communications Applications, 11-13 June, 2007, Eskisehir, Turkey.
- Karaboga, D., Basturk, B. 2008.** On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8(1): 687-697.
- Karaboga, D., Ozturk, C. 2011.** A novel clustering approach: artificial bee colony (ABC) algorithm, *Applied Soft Computing*, 11(1): 652-657.
- Kennedy, J., Eberhart, R. 1995.** Particle swarm optimization. *Neural Networks, International Conference*, 4: 1942-1948.
- Kim, S-S., Byeon, J-H., Liu, H., Abraham, A., McLoone, S. 2013.** Optimal job scheduling in grid computing using efficient binary artificial bee colony optimization. *Soft Computing*, 17(5): 867-882.

- Kiran, M.S., Gündüz, M. 2013.** XOR-based artificial bee colony algorithm for binary optimization. *Turkish Journal of Electrical Engineering & Computer Sciences*, 21: 2307-2328.
- Kohavi, R. 1995.** The Power of Decision Tables. *Lecture Notes in Computer Science, ECML*, 912: 174-189.
- Kohavi, R. 1996.** Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. *Proceedings of Second International Conference on Knowledge Discovery and Data Mining*, 114-119.
- Kotsiantis, S., Kanellopoulos, D., Pintelas, P. 2006.** Data Preprocessing for Supervised Learning. *International Journal of Computer Science*, 1(2): 111-117.
- Kotsiantis, S., Zaharakis, I., Pintelas, P. 2006.** Supervised Machine Learning: A Review of Classification Techniques. *Artificial Intelligence Review*, 26(3): 159-190.
- Linoff, G.S., Berry, M.J.A. 2011.** *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. John Wiley & Sons, Indianapolis, IN, USA, 888 pp.
- Maimon, O., Rokach, L. 2010.** *Data Mining and Knowledge Discovery Handbook*, Second Edition. Springer, New York, NY, USA, 1285 pp.
- Mehrotra, K., Mohan, C.K., Ranka, S. 1997.** *Elements of Artificial Neural Networks*. MIT Press, USA, 344 pp.
- Merkle, D. Middendorf, M. Schmeck, H. 2002.** Ant colony optimization for resource-constrained Project scheduling, *IEEE Trans. Evol. Comput.* 6(4): 333–346.
- Mitchell, T.M. 1997.** *Machine Learning*. McGraw-Hill, Maidenhead, UK, 414 pp.
- Mitra, S., Pal, S.K., Mitra, P. 2002.** Data mining in soft computing framework: A survey. *IEEE Transactions on Neural Networks*, 13(1): 3-14.
- Murty, K.G. 2003.** Models for Decision Making: Optimization Models For Decision Making, vol. 1, Internet Edition, Michigan, Ann Arbor, USA, pp: 1-18.
- Nigai, E.W.T, Xiu, L., Chau, D.C.K. 2009.** Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, 36: 2592-2602.
- Pampara, G., Engelbrecht, A.P. 2011.** Binary artificial bee colony optimization. *IEEE Symposium on Swarm Intelligence*, 11-15 April, 2011, Paris, France.
- Priddy, K.L., Keller, P.E. 2005.** *Artificial Neural Networks: An Introduction*. SPIE Press, Bellingham, Washington, USA, 165 pp.
- Quinlan, J. R. 1979.** Discovering rules from large collections of examples: a case study. *Expert Systems in the Micro-Electronic Age*. Edinburgh University Press. pp. 168-201
- Quinlan, J. R. 1993.** *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers San Mateo.
- Sasaki, M. Kita, K. 1998.** Rule-based text categorization using hierarchical categories, *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, 3: 2827-2830.

- Seeley, T.D. 1995.** The Wisdom of the Hive. Harvard University Press, Cambridge, MA.
- Shaw, M.J., Subramaniam, C., Tan, G.W., Welge, M.E. 2001.** Knowledge management and data mining for marketing. *Decision Support Systems*, 31: 127-137.
- Shukran, M.A., Chung, Y.Y., Yeh, W., Wahid, N., Zaidi, A.M. 2011.** Artificial Bee Colony based Data Mining Algorithms for Classification Tasks. *Modern Applied Science*, 5(4): 217-231.
- Sreejith, S., Chandrasekaran, K., Simon, S.P. 2009.** Application of Touring Ant colony Optimization technique for Optimal Power Flow incorporating Thyristor Controlled Series Compensator. TENCON, 23-26 Jan. 2009, Singapore.
- Stützle, T. Dorigo, M. 2002.** A short convergence proof for a class of ACO algorithms. *IEEE Trans. Evol. Comput.* 6(4): 358–365.
- Tereshko, V., Loengarov, A. 2005.** Collective Decision-Making in Honey Bee Foraging Dynamics. *Journal of Computing and Information Systems*, 9(3): 1-7.
- Thrun, S.B. 1994.** Extracting provably correct rules from artificial neural networks. Tech. Rep. IAITR-93-5, Inst. Informatik III, Univ. Bonn, Germany.
- Tickle, A.B., Andrews, R., Golea, M., Diederich, J. 1998.** The Truth Will Come to Light: Directions and Challenges in Extracting the Knowledge Embedded Within Trained Artificial Neural Networks. *Transactions on Neural Networks*, 9(6): 1057-1068.
- Towell, G., Shavlik, J. 1993.** The extraction of refined rules from knowledge-based neural networks. *Machine Learning*, 131: 71–101.
- Tsai, P.W., Pan, J.S., Liao, B.Y., Chu, S.C. 2009.** Enhanced artificial bee colony optimization. *International Journal of Innovative Computing, Information and Control*, 5(12): 5081-5092.
- Vijayarani, S., Divya, M. 2011.** An Efficient Algorithm for Generating Classification Rules. *International Journal of Computer Science and Technology*, 2(4): 512-515.
- Wang, L., Fu, X. 2006.** Data Mining with Computational Intelligence. Springer, Germany, 288 pp.
- Wei, L., Hanning, C. 2012.** BABC: A Binary Version of Artificial Bee Colony Algorithm for Discrete Optimization. *International Journal of Advancements in Computing Technology*, 4(14): 307-314.
- Werbos, P. 1974.** “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences,” Harvard University dissertation.
- Werner, L. 2005.** Decision Tables, <http://classes.soe.ucsc.edu/cmeps115/Spring05/supplements/DecisionTables.htm>-14.01.2012 – (Erişim tarihi: 18.11.2013)
- Witten, I.H., Frank, E., Hall, M.A. 2005.** Data Mining: Practical Machine Learning Tools and Techniques, Third Edition. Morgan Kaufmann Publishers, Burlington, MA, USA, 629 pp.
- Witten, I.H., Frank, E., Hall, M.A. 2011.** Data Mining Practical Learning Tools and Techniques, Third Edition. Morgan Kaufmann Publishers, Burlington, MA, USA, 629 pp.

Xindong Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z-H., Steinbach, M., Hand, D.J., Steinberg, D. 2008. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1): 1-37.

Yegnanarayana, B. 2009. Artificial Neural Networks. PHI Learning Pvt. Ltd, New Delhi, India, 476 pp.

Yuko Aratsu, Y., Mizuno, K., Sasaki, H., Nishihara, S. 2013. Solving Constraint Satisfaction Problems by Artificial Bee Colony with Greedy Scouts. World Congress on Engineering and Computer Science, 23-25 October, 2013, San Francisco, USA.

Zhang, C., Ouyang, D., Ning, J. 2010. An artificial bee colony approach for clustering. *Expert Systems with Applications*, 37(7): 4761-4767.

EKLER

- | | |
|-------------|---|
| EK 1 | Kural Çıkarımı için Kullanılan Veri Kümeleri |
| EK 2 | İkili Hale Dönüştürülmüş Veri Kümeleri |
| EK 3 | STATISTICA Programı ile Eğitilen YSA Performans Tablosu |
| EK 4 | MATLAB YSA Eğitimi Performans Grafikleri ve Tabloları |
| EK 5 | Veri Kümesi Bazında MATLAB’de Eğitilmiş YSA Hata Matrisleri |
| EK 6 | İkili Değer Alan Test Fonksiyonları |
| EK 7 | Veri Kümeleri için Kümeleme Analizi Sonuçları |

EK 1 Kural Çıkarımı için Kullanılan Veri Kümeleri

“*Breast Cancer Wisconsin (BCW)*” Veri Kümesi:

Wisconsin, Amerika Birleşik Devletleri eyaletinin sınırları içindeki göğüs kanseri vakalarındaki tümörlerin tipinin, tümörlerin gösterdiği karakteristik özelliklere göre sınıflandırılması.

Veri sayısı: 699

Öznitelik sayısı: 9

Sınıf sayısı: 2 (iyicil, kötücül)

Tip: Sürekli

Eksik değer sayısı: 16

Öznitelik detayları:

Çizelge EK 1.1. BCW veri kümesi için öznitelik özellikleri

Öznitelik Numarası	Öznitelik tanımı	Kesikli halde seçenek sayısı
Öznitelik 1	Parça kalınlığı	3
Öznitelik 2	Hücre boyunun düzgünlüğü	4
Öznitelik 3	Hücre şeklinin düzgünlüğü	4
Öznitelik 4	Sınır birleşmeleri	3
Öznitelik 5	Tek epitel hücre boyu	3
Öznitelik 6	Çekirdek	4
Öznitelik 7	Kromatin	3
Öznitelik 8	Çekirdekçik	3
Öznitelik 9	Mitoz	2

“*Iris*” Veri Kümesi:

Çiçeklerin taç ve çanak yaprak karakteristiklerine göre sınıflandırılması.

Veri sayısı: 150

Öznitelik sayısı: 4

Sınıf sayısı: 3 (*setosa*, *versicolor*, *virginic*)

Tip: Sürekli

Eksik değer sayısı: --

Öznitelik detayları:

Çizelge EK 1.2. Iris veri kümesi için öznitelik özellikleri

Öznitelik Numarası	Öznitelik tanımı	Kesikli halde seçenek sayısı
Öznitelik 1	Çanak yaprak uzunluğu	3
Öznitelik 2	Çanak yaprak genişliği	3
Öznitelik 3	Taç yaprak uzunluğu	3
Öznitelik 4	Taç yaprak genişliği	3

“Monk” Veri Kümesi:

Robotların fiziksel özelliklerine göre sınıflandırılması.

Veri sayısı: 432

Öznitelik sayısı: 6

Sınıf sayısı: 2 (istenen sınıf, diğer)

Tip: Kategorik

Eksik değer sayısı: --

Öznitelik detayları:

Çizelge EK 1.3. Monk veri kümesi için öznitelik özellikleri

Öznitelik Numarası	Öznitelik tanımı	Seçenek sayısı
Öznitelik 1	Kafa şekli	3
Öznitelik 2	Vücut şekli	3
Öznitelik 3	Gülme durumu	2
Öznitelik 4	Elinde ne tutuyor	3
Öznitelik 5	Ceket rengi	4
Öznitelik 6	Kravat durumu	2

“Nursery” Veri Kümesi:

Ailelerin kreş başvurularının ailevi özellikler değerlendirilerek, öncelik durumuna göre sınıflandırılması.

Veri sayısı: 12960

Öznitelik sayısı: 8

Sınıf sayısı: 5 (özel öncelikli, öncelikli önerilen, öncelikli, önerilen, önerilmeyen)

Tip: Kategorik

Eksik değer sayısı: --

Öznitelik detayları:

Çizelge EK 1.4. Nursery veri kümesi için öznitelik özellikleri

Öznitelik Numarası	Öznitelik tanımı	Seçenek sayısı
Öznitelik 1	Ebeveynler	3
Öznitelik 2	Bakıcı durumu	5
Öznitelik 3	Form	4
Öznitelik 4	Çocuk	4
Öznitelik 5	Ev durumu	3
Öznitelik 6	Finansman	2
Öznitelik 7	Sosyal	3
Öznitelik 8	Sağlık	3

“Sunburn” Veri Kümesi:

Fiziksel özellikler ve güneş kremi kullanımına bağlı olarak, güneş yanıklarının sınıflandırılması.

Veri sayısı: 8

Öznitelik sayısı: 4

Sınıf sayısı: 2 (güneş yanığı olur, güneş yanığı olmaz)

Tip: Kategorik

Eksik değer sayısı: --

Öznitelik detayları:

Çizelge EK 1.5. Sunburn veri kümesi için öznitelik özellikleri

Öznitelik numarası	Öznitelik tanımı	Seçenek sayısı
Öznitelik 1	Saç rengi	3
Öznitelik 2	Boy	3
Öznitelik 3	Kilo	2
Öznitelik 4	Güneş kremi	2

“Titanic” Veri Kümesi:

Titanik gemisindeki yolcuların fiziksel özellikleri ve kamara tipine göre, batma esnasında boğulup boğulmayacağını sınıflandırılması.

Veri sayısı: 2201

Öznitelik sayısı: 3

Sınıf sayısı: 2 (boğulur, boğulmaz)

Tip: Kategorik

Eksik değer sayısı: --

Öznitelik detayları:

Çizelge EK 1.6. Titanic veri kümesi için öznitelik özellikleri

Öznitelik numarası	Öznitelik tanımı	Seçenek sayısı
Öznitelik 1	Kamara	4
Öznitelik 2	Yaş	2
Öznitelik 3	Cinsiyet	2

EK 1.7. “Weather” Veri Kümesi

Hava durumuna göre tenis oynamaya uygunluk durumunun sınıflandırılması.

Veri sayısı: 14

Öznitelik sayısı: 4

Sınıf sayısı: 2 (tenis oynanır, tenis oynanmaz)

Tip: Sürekli, kategorik

Eksik değer sayısı: --

Öznitelik detayları:

Çizelge EK 1.7. Weather veri kümesi için öznitelik özellikleri

Öznitelik numarası	Öznitelik tanımı	Seçenek sayısı
Öznitelik 1	Hava durumu	3
Öznitelik 2	Sıcaklık	3
Öznitelik 3	Nem	2
Öznitelik 4	Rüzgâr	2

“Wine” Veri Kümesi:

Şarapların fiziksel özellikleri ve içeriklerine göre sınıflandırılması.

Veri sayısı: 178

Öznitelik sayısı: 13

Sınıf sayısı: 3 (1,2,3)

Tip: Kategorik

Eksik değer sayısı: --

Öznitelik detayları:

Çizelge EK 1.8. Wine veri kümesi için öznitelik özellikleri

Öznitelik numarası	Öznitelik tanımı	Kesikli halde seçenek sayısı
Öznitelik 1	Alkol	3
Öznitelik 2	Malik asit	3
Öznitelik 3	Kül	2
Öznitelik 4	Külün alkalinitesi	2
Öznitelik 5	Magnezyum	2
Öznitelik 6	Toplam fenoller	3
Öznitelik 7	Flavanoidler	4
Öznitelik 8	Flavanoid olmayan fenoller	2
Öznitelik 9	Proantosiyeninler	2
Öznitelik 10	Renk yoğunluğu	3
Öznitelik 11	Renk derecesi	4
Öznitelik 12	OD280/OD315 oranı	3
Öznitelik 13	Prolin	4

EK 2 İkili Hale Dönüştürülmüş Veri Kümeleri

Çizelge EK 2.1. BCW veri kümesinin ikili hali

Veri kümesi	Öznitelik	Parametre değeri	İkili düğüm
BCW	Parça kalınlığı	$(-\infty-4,5]$	x_1
		$(4,5-6,5]$	x_2
		$(6,5-\infty)$	x_3
	Hücre boyunun düzgünlüğü	$(-\infty-1,5]$	x_4
		$(1,5-2,5]$	x_5
		$(2,5-4,5]$	x_6
		$(4,5-\infty)$	x_7
	Hücre şeklinin düzgünlüğü	$(-\infty-1,5]$	x_8
		$(1,5-2,5]$	x_9
		$(2,5-4,5]$	x_{10}
		$(4,5-\infty)$	x_{11}
	Sınır birleşmeleri	$(-\infty-1,5]$	x_{12}
		$(1,5-3,5]$	x_{13}
		$(3,5-\infty)$	x_{14}
	Tek epitel hücre boyu	$(-\infty-2,5]$	x_{15}
		$(2,5-3,5]$	x_{16}
		$(3,5-\infty)$	x_{17}
	Çekirdek	$(-\infty-1,5]$	x_{18}
		$(1,5-2,5]$	x_{19}
		$(2,5-5,5]$	x_{20}
		$(5,5-\infty)$	x_{21}
	Kromatin	$(-\infty-2,5]$	x_{22}
		$(2,5-3,5]$	x_{23}
		$(3,5-\infty)$	x_{24}
	Çekirdekçik	$(-\infty-2,5]$	x_{25}
		$(2,5-9,5]$	x_{26}
		$(9,5-\infty)$	x_{27}
	Mitoz	$(-\infty-1,5]$	x_{28}
		$(1,5-\infty)$	x_{29}
	Sınıf	İyicil	λ_1
		Kötücül	λ_2

Çizelge EK 2.2. Iris veri kümesinin ikili hali

Veri kümesi	Öznitelik	Parametre değeri	İkili düğüm
Iris	Çanak yaprak uzunluğu	$(-\infty-5,55]$	x_1
		$(5,55-6,15]$	x_2
		$(6,15-\infty)$	x_3
	Çanak yaprak genişliği	$(-\infty-2,95]$	x_4
		$(2,95-3,35]$	x_5
		$(3,35-\infty)$	x_6
	Taç yaprak uzunluğu	$(-\infty-2,45]$	x_7
		$(2,45-4,75]$	x_8
		$(4,75-\infty)$	x_9
	Taç yaprak genişliği	$(-\infty-0,8]$	x_{10}
		$(0,8-1,75]$	x_{11}
		$(1,75-\infty)$	x_{12}
	Sınıf	<i>Setosa</i>	λ_1
		<i>Versicolor</i>	λ_2
		<i>Virginic</i>	λ_3

Çizelge EK 2.3. Monk veri kümesinin ikili hali

Veri kümesi	Öznitelik	Parametre değeri	İkili düğüm
Monk	Kafa şekli	Yuvarlak	x_1
		Sekizgen	x_2
		Kare	x_3
	Vücut şekli	Yuvarlak	x_4
		Sekizgen	x_5
		Kare	x_6
	Gülme durumu	Evet	x_7
		Hayır	x_8
	Elinde ne tutuyor	Kılıç	x_9
		Bayrak	x_{10}
		Balon	x_{11}
	Ceket rengi	Kırmızı	x_{12}
		Sarı	x_{13}
		Yeşil	x_{14}
		Mavi	x_{15}
	Kravat durumu	Evet	x_{16}
		Hayır	x_{17}
	Sınıf	İstenen	λ_1
		Diğer	λ_2

Çizelge EK 2.4. Nursery veri kümesinin ikili hali

Veri kümesi	Öznitelik	Parametre değeri	İkili düğüm
Nursery	Ebeveynler	Normal	x_1
		İddialı	x_2
		Çok iddialı	x_3
	Bakıcı durumu	Uygun	x_4
		Az uygun	x_5
		Uygunsuz	x_6
		Kritik	x_7
		Çok kritik	x_8
	Form	Tamam	x_9
		Eksik	x_{10}
		Evlatlık	x_{11}
		Tamamlandı	x_{12}
	Çocuklar	1	x_{13}
		2	x_{14}
		3	x_{15}
		>3	x_{16}
	Ev durumu	Kritik	x_{17}
		Uygun	x_{18}
		Az uygun	x_{19}
	Finansman	Uygunsuz	x_{20}
		Uygun	x_{21}
	Sosyal	Problemsiz	x_{22}
		Az problemlili	x_{23}
		Problemlili	x_{24}
	Sağlık	Öncelikli	x_{25}
		Önerilmeyen	x_{26}
		Önerilen	x_{27}
	Sınıf	Önerilmeyen	λ_1
		Öncelikli	λ_2
		Özel öncelikli	λ_3
		Önerilen	λ_4
		Öncelikli önerilen	λ_5

Çizelge EK 2.5. Sunburn veri kümesinin ikili hali

Veri kümesi	Öznitelik	Parametre değeri	İkili düğüm
Sunburn	Saç rengi	Sarı	x_1
		Kahverengi	x_2
		Kızıl	x_3
	Boy	Kısa	x_4
		Ortalama	x_5
		Uzun	x_6
	Kilo	İnce	x_7
		Normal	x_8
		Kilolu	x_9
	Güneş kremi	Evet	x_{10}
		Hayır	x_{11}
	Sınıf	Güneş yanığı olur	λ_1
		Güneş yanığı olmaz	λ_2

Çizelge EK 2.6. Titanic veri kümesinin ikili hali

Veri kümesi	Öznitelik	Parametre değeri	İkili düğüm
Titanic	Kamara	Tayfa	x_1
		Birinci sınıf	x_2
		İkinci sınıf	x_3
		Üçüncü sınıf	x_4
	Yaş	Yetişkin	x_5
		Çocuk	x_6
	Cinsiyet	Kadın	x_7
		Erkek	x_8
	Sınıf	Boğulur	λ_1
		Boğulmaz	λ_2

Çizelge EK 2.7. Weather veri kümesinin ikili hali

Veri kümesi	Öznitelik	Parametre değeri	İkili düğüm
Weather	Hava durumu	Güneşli	x_1
		Bulutlu	x_2
		Yağmurlu	x_3
	Sıcaklık	Sıcaklık	x_4
		Ilık	x_5
		Serin	x_6
	Nem	Yüksek	x_7
		Normal	x_8
	Rüzgar	Az	x_9
		Kuvvetli	x_{10}
	Sınıf	Tenis oynanır	λ_1
		Tenis oynanmaz	λ_2

Çizelge EK 2.8. Wine veri kümesinin ikili hali

Veri kümesi	Öznitelik	Parametre değeri	İkili düğüm
Wine	Alkol	$(-\infty-12,185]$	x_1
		$(12,185-12,78]$	x_2
		$(12,78-\infty)$	x_3
	Malik asit	$(-\infty-1,42]$	x_4
		$(1,42-2,235]$	x_5
		$(2,235-\infty)$	x_6
	Kül	$(-\infty-2,03]$	x_7
		$(2,03-\infty)$	x_8
	Külün alkalitesi	$(-\infty-17,9]$	x_9
		$(17,9-\infty)$	x_{10}
	Magnezyum	$(-\infty-88,5]$	x_{11}
		$(88,5-\infty)$	x_{12}
	Toplam fenoller	$(-\infty-1,84]$	x_{13}
		$(1,84-2,335]$	x_{14}
		$(2,335-\infty)$	x_{15}
	Flavanoidler	$(-\infty-0,975]$	x_{16}
		$(0,975-1,575]$	x_{17}
		$(1,575-2,31]$	x_{18}
		$(2,31-\infty)$	x_{19}
	Flavanoid olmayan fenoller	$(-\infty-0,395]$	x_{20}
		$(0,395-\infty)$	x_{21}
	Proantosiyeninler	$(-\infty-1,27]$	x_{22}
		$(1,27-\infty)$	x_{23}
	Renk yoğunluğu	$(-\infty-3,46]$	x_{24}
		$(3,46-7,55]$	x_{25}
		$(7,55-\infty)$	x_{26}
	Renk derecesi	$(-\infty-0,785]$	x_{27}
		$(0,785-0,975]$	x_{28}
		$(0,975-1,295]$	x_{29}
		$(1,295-\infty)$	x_{30}
	OD280/OD315 oranı	$(-\infty-2,115]$	x_{31}
		$(2,115-2,475]$	x_{32}
		$(2,475-\infty)$	x_{33}
	Prolin	$(-\infty-468]$	x_{34}
		$(468-755]$	x_{35}
		$(755-987,5]$	x_{36}
		$(987,5-\infty)$	x_{37}
	Sınıf	1	λ_1
		2	λ_2
		3	λ_3

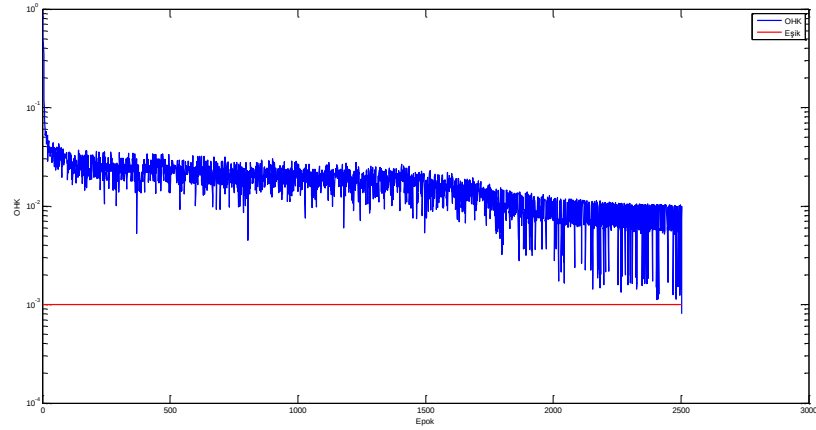
EK 3 STATISTICA Programı ile Eğitilen YSA Performans Tablosu

Veri kümesi	İndeks	Ağ adı	Eğitim perf.	Test perf.	Eğitim algoritması	Hata fonk.	Gizli katman aktivasyon fonksiyonu	Çıktı katmanı aktivasyon fonksiyonu	Ortalama perf.	Gizli katman sayısı
BCW	1	MLP 29-16-2	97,99%	97,79%	BFGS 22	SOS	Logistic	Logistic	97,89%	13
	2	MLP 29-13-2	97,99%	97,79%	BFGS 24	SOS	Logistic	Logistic	97,89%	
	3	MLP 29-16-2	97,07%	97,06%	BFGS 3	SOS	Logistic	Logistic	97,07%	
	4	MLP 29-23-2	95,98%	97,06%	BFGS 3	SOS	Logistic	Logistic	96,52%	
	5	MLP 29-10-2	97,07%	97,06%	BFGS 4	SOS	Logistic	Logistic	97,07%	
Iris	1	MLP 12-14-3	95,83%	96,67%	BFGS 7	SOS	Logistic	Logistic	96,25%	4
	2	MLP 12-5-3	95,83%	96,67%	BFGS 6	SOS	Logistic	Logistic	96,25%	
	3	MLP 12-8-3	95,83%	96,67%	BFGS 3	SOS	Logistic	Logistic	96,25%	
	4	MLP 12-4-3	96,67%	96,67%	BFGS 20	SOS	Logistic	Logistic	96,67%	
	5	MLP 12-8-3	95,83%	96,67%	BFGS 6	SOS	Logistic	Logistic	96,25%	
Monk	1	MLP 17-5-2	100,00%	100,00%	BFGS 11	SOS	Logistic	Logistic	100,00%	5
	2	MLP 17-12-2	100,00%	100,00%	BFGS 21	SOS	Logistic	Logistic	100,00%	
	3	MLP 17-10-2	100,00%	100,00%	BFGS 20	SOS	Logistic	Logistic	100,00%	
	4	MLP 17-7-2	99,42%	100,00%	BFGS 12	SOS	Logistic	Logistic	99,71%	
	5	MLP 17-15-2	99,42%	100,00%	BFGS 18	SOS	Logistic	Logistic	99,71%	
Nursery	1	MLP 27-10-5	93,03%	93,13%	BFGS 42	SOS	Logistic	Logistic	93,08%	18
	2	MLP 27-10-5	95,18%	95,91%	BFGS 98	SOS	Logistic	Logistic	95,54%	
	3	MLP 27-5-5	93,14%	93,56%	BFGS 136	SOS	Logistic	Logistic	93,35%	
	4	MLP 27-13-5	93,53%	93,94%	BFGS 73	SOS	Logistic	Logistic	93,74%	
	5	MLP 27-18-5	96,47%	96,33%	BFGS 102	SOS	Logistic	Logistic	96,40%	
Sunburn	1	MLP 11-4-2	57,14%	100,00%	BFGS -1	SOS	Logistic	Logistic	78,57%	4
	2	MLP 11-5-2	57,14%	100,00%	BFGS -1	SOS	Logistic	Logistic	78,57%	
	3	MLP 11-5-2	57,14%	100,00%	BFGS -1	SOS	Logistic	Logistic	78,57%	
	4	MLP 11-5-2	57,14%	100,00%	BFGS -1	SOS	Logistic	Logistic	78,57%	
	5	MLP 11-4-2	57,14%	100,00%	BFGS -1	SOS	Logistic	Logistic	78,57%	
Titanic	1	MLP 8-3-2	78,42%	81,59%	BFGS 28	SOS	Logistic	Logistic	80,01%	3
	2	MLP 8-12-2	78,42%	81,59%	BFGS 10	SOS	Logistic	Logistic	80,01%	
	3	MLP 8-12-2	78,42%	81,59%	BFGS 11	SOS	Logistic	Logistic	80,01%	
	4	MLP 8-12-2	78,42%	81,59%	BFGS 21	SOS	Logistic	Logistic	80,01%	
	5	MLP 8-9-2	78,42%	81,59%	BFGS 19	SOS	Logistic	Logistic	80,01%	
Weather	1	MLP 10-11-2	58,33%	100,00%	BFGS -1	SOS	Logistic	Logistic	79,17%	3
	2	MLP 10-9-2	58,33%	100,00%	BFGS 1	SOS	Logistic	Logistic	79,17%	
	3	MLP 10-3-2	58,33%	100,00%	BFGS 1	SOS	Logistic	Logistic	79,17%	
	4	MLP 10-10-2	58,33%	100,00%	BFGS -1	SOS	Logistic	Logistic	79,17%	
	5	MLP 10-9-2	58,33%	100,00%	BFGS 1	SOS	Logistic	Logistic	79,17%	
Wine	1	MLP 37-29-3	96,50%	100,00%	BFGS 7	SOS	Logistic	Logistic	98,25%	7
	2	MLP 37-7-3	99,30%	100,00%	BFGS 6	SOS	Logistic	Logistic	99,65%	
	3	MLP 37-6-3	97,90%	100,00%	BFGS 6	SOS	Logistic	Logistic	98,95%	
	4	MLP 37-34-3	96,50%	100,00%	BFGS 6	SOS	Logistic	Logistic	98,25%	
	5	MLP 37-11-3	97,90%	100,00%	BFGS 7	SOS	Logistic	Logistic	98,95%	

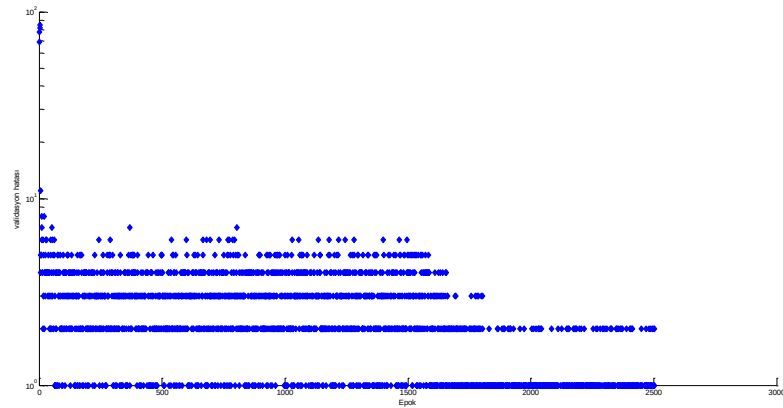
Şekil EK 3.1. STATISTICA programı ile eğitilen YSA performans tablosu

EK 4 MATLAB YSA Eğitimi Performans Grafikleri ve Tabloları

BCW Veri Kümesi:



Şekil EK 4.1. BCW veri kümesi MATLAB YSA eğitimi için ortalama hata karesi

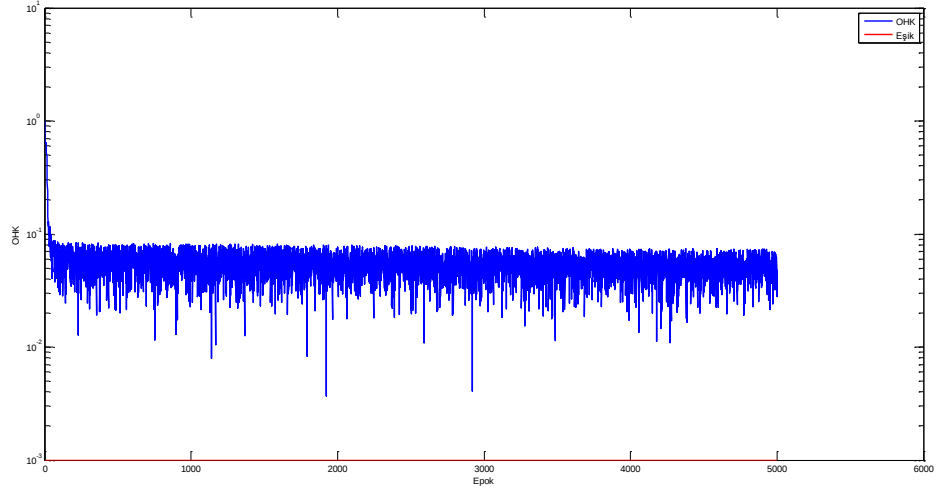


Şekil EK 4.2. BCW veri kümesi MATLAB YSA eğitimi için validasyon hatası

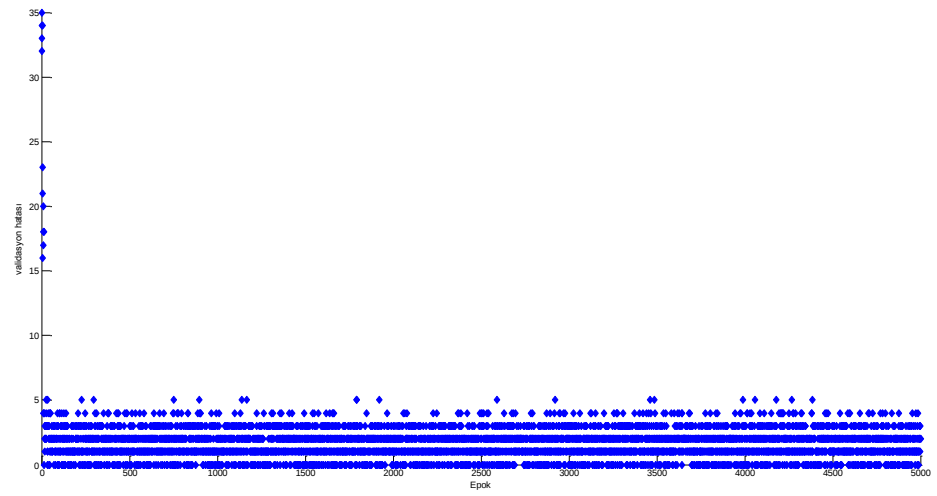
Çizelge EK 4.1. BCW veri kümesi MATLAB YSA eğitimi sonuçları

Epok sayısı	2503
OHK	0,0008
Test edilen girdi sayısı	227
Hata sayısı	2
Hata yüzdesi	1%

Iris Veri Kümesi:



Şekil EK 4.3. Iris veri kümesi MATLAB YSA eğitimi için ortalama hata karesi

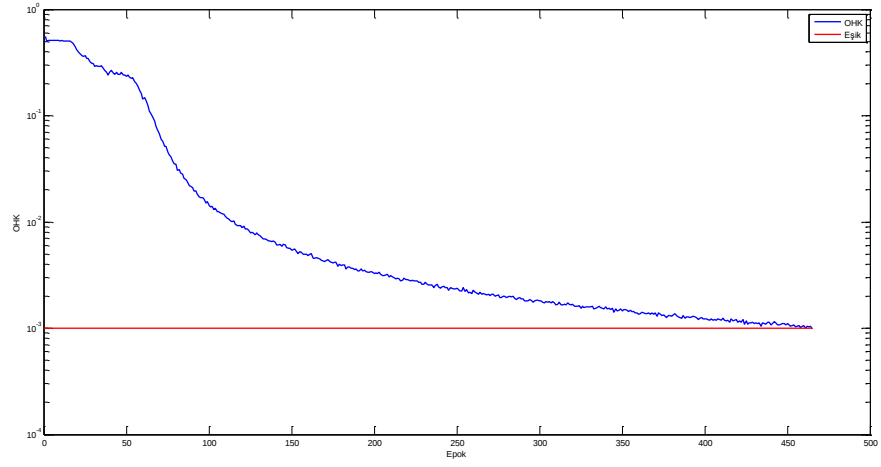


Şekil EK 4.4. Iris veri kümesi MATLAB YSA eğitimi için validasyon hatası

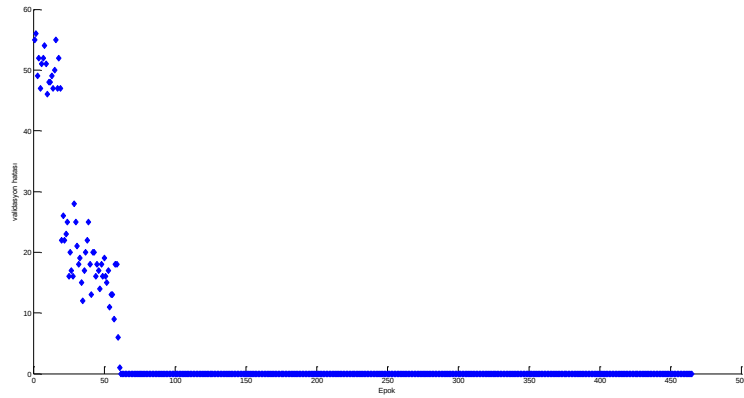
Çizelge EK 4.2. Iris veri kümesi MATLAB YSA eğitimi sonuçları

Epok sayısı	5000
OHK	0,04807
Test edilen girdi sayısı	48
Hata sayısı	2
Hata yüzdesi	4%

Monk Veri Kümesi:



Şekil EK 4.5. Monk veri kümesi MATLAB YSA eğitimi için ortalama hata karesi

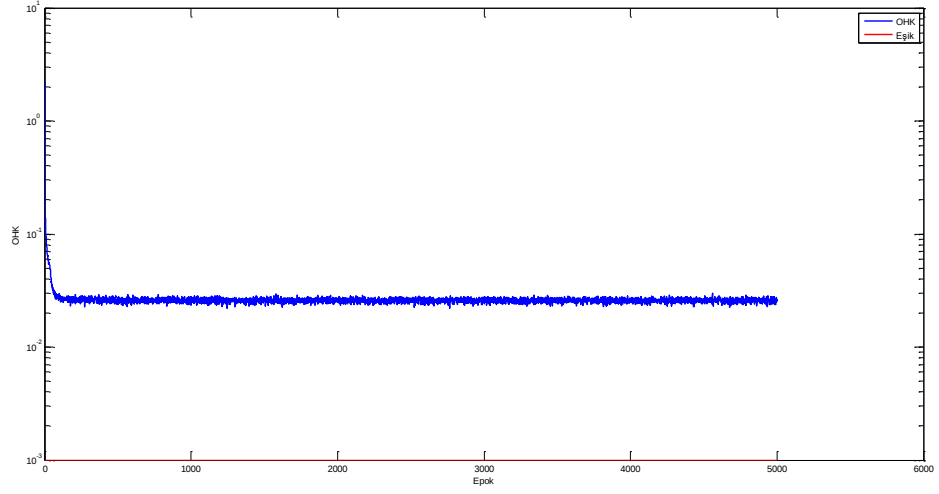


Şekil EK 4.6. Monk veri kümesi MATLAB YSA eğitimi için validasyon hatası

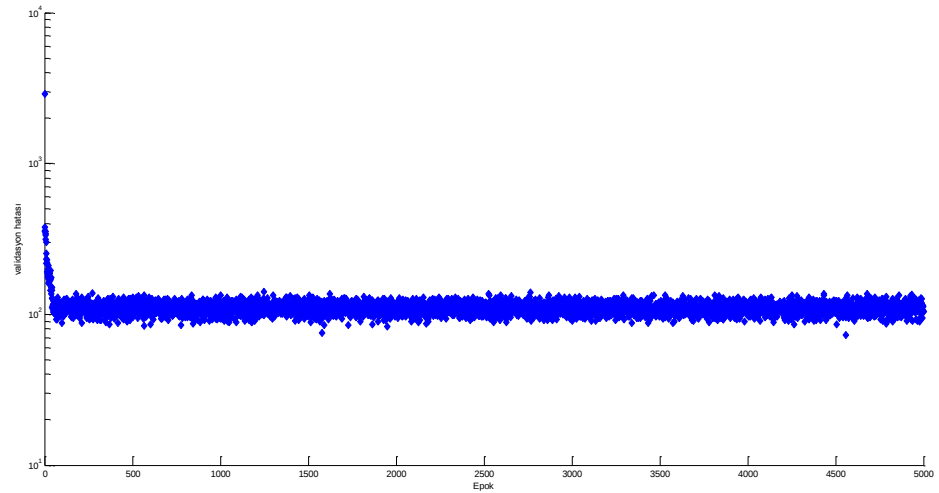
Çizelge EK 4.3. Monk veri kümesi MATLAB YSA eğitimi sonuçları

Epok sayısı	464
OHK	0,00099
Test edilen girdi sayısı	100
Hata sayısı	0
Hata yüzdesi	0%

Nursery Veri Kümesi:



Şekil EK 4.7. Nursery veri kümesi MATLAB YSA eğitimi için ortalama hata karesi

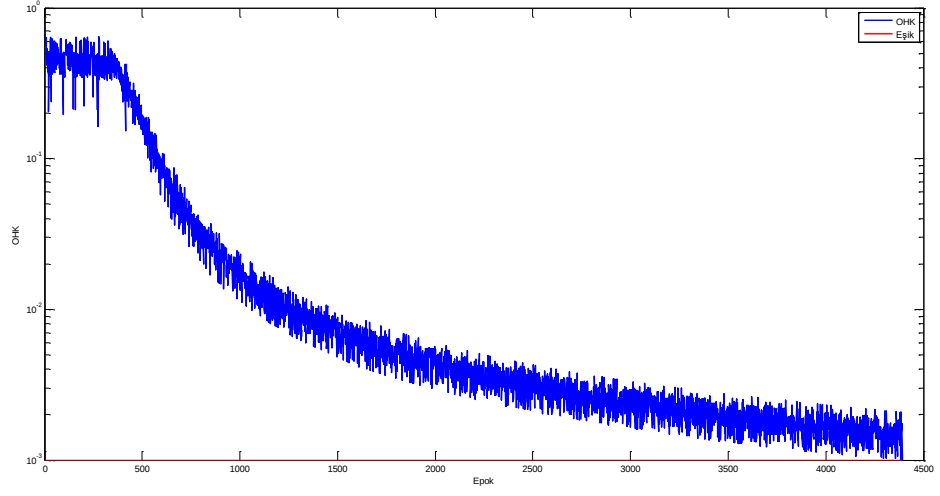


Şekil EK 4.8. Nursery veri kümesi MATLAB YSA eğitimi için validasyon hatası

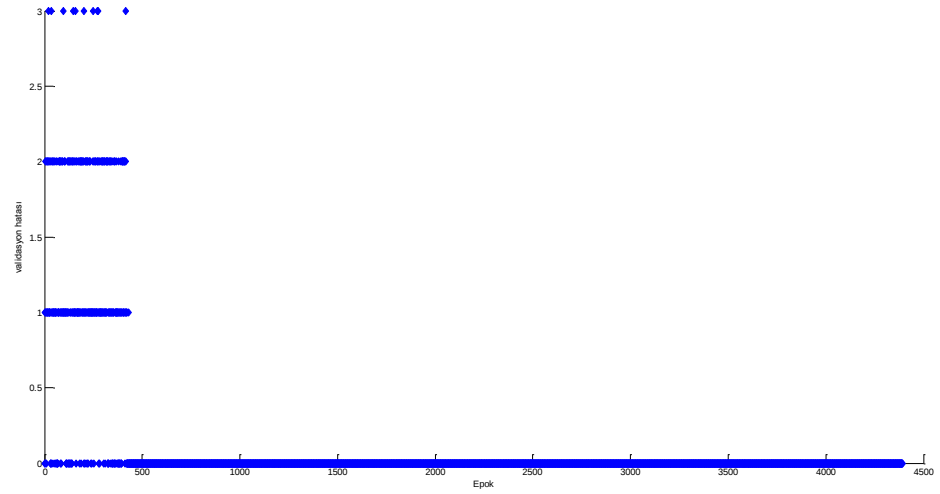
Çizelge EK 4.4. Nursery veri kümesi MATLAB YSA eğitimi sonuçları

Epok sayısı	5000
OHK	0,02665
Test edilen girdi sayısı	4320
Hata sayısı	104
Hata yüzdesi	2%

Sunburn Veri Kümesi:



Şekil EK 4.9. Sunburn veri kümesi MATLAB YSA eğitimi için ortalama hata karesi

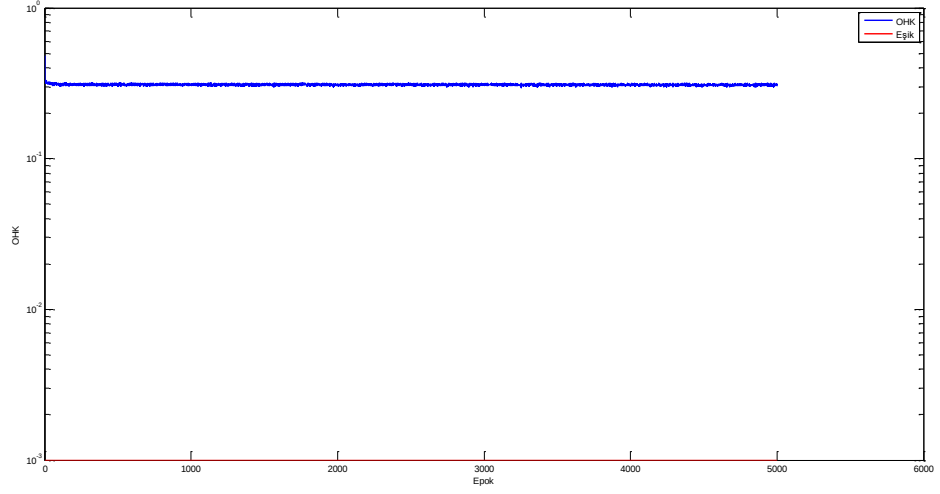


Şekil EK 4.10. Sunburn veri kümesi MATLAB YSA eğitimi için validasyon hatası

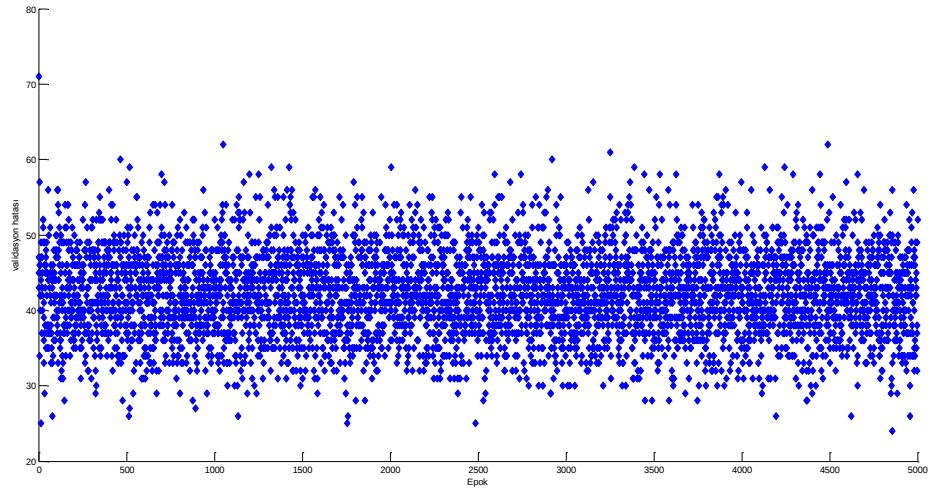
Çizelge EK 4.5. Sunburn veri kümesi MATLAB YSA eğitimi sonuçları

Epok sayısı	4391
OHK	0,001
Test edilen girdi sayısı	3
Hata sayısı	0
Hata yüzdesi	0%

Titanic Veri Kümesi:



Şekil EK 4.11. Titanic veri kümesi MATLAB YSA eğitimi için ortalama hata karesi

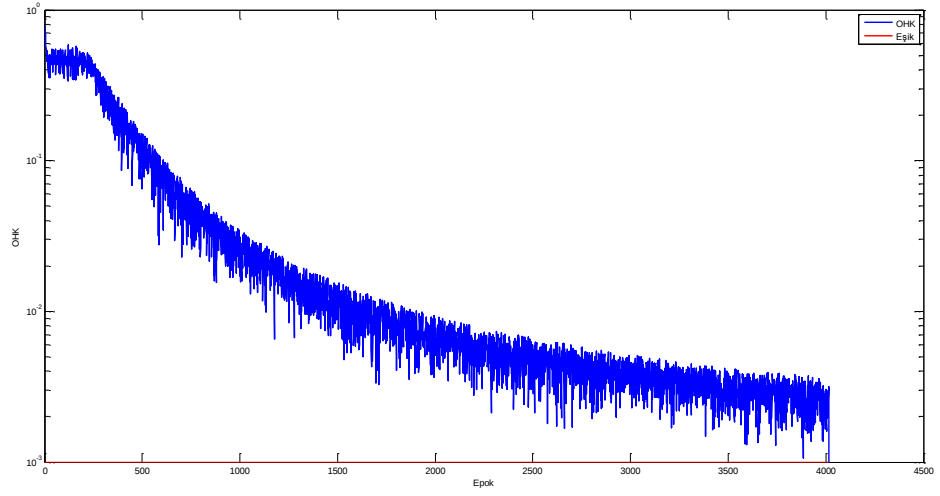


Şekil EK 4.12. Titanic veri kümesi MATLAB YSA eğitimi için validasyon hatası

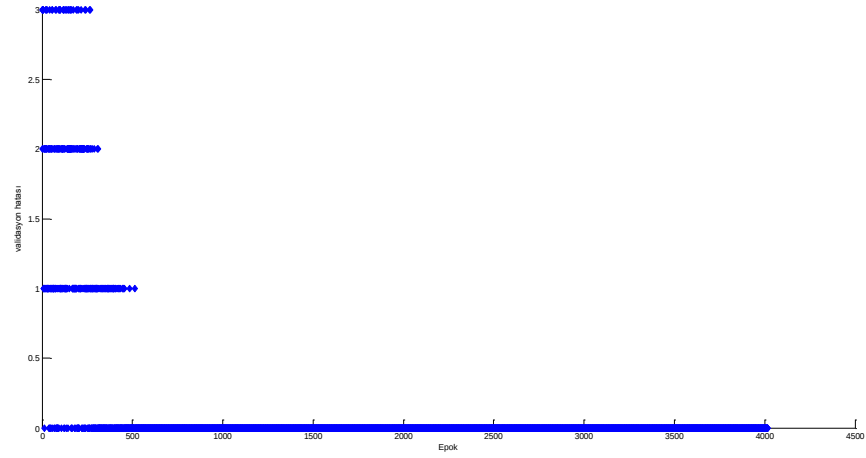
Çizelge EK 4.6. Titanic veri kümesi MATLAB YSA eğitimi sonuçları

Epok sayısı	5000
OHK	0,3254
Test edilen girdi sayısı	201
Hata sayısı	52
Hata yüzdesi	26%

Weather Veri Kümesi:



Şekil EK 4.13. Weather veri kümesi MATLAB YSA eğitimi için ortalama hata karesi

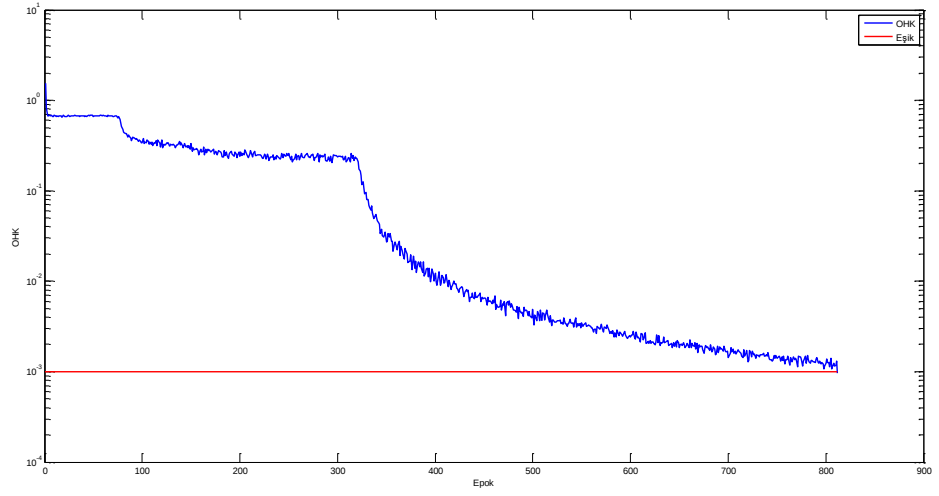


Şekil EK 4.14. Weather veri kümesi MATLAB YSA eğitimi için validasyon hatası

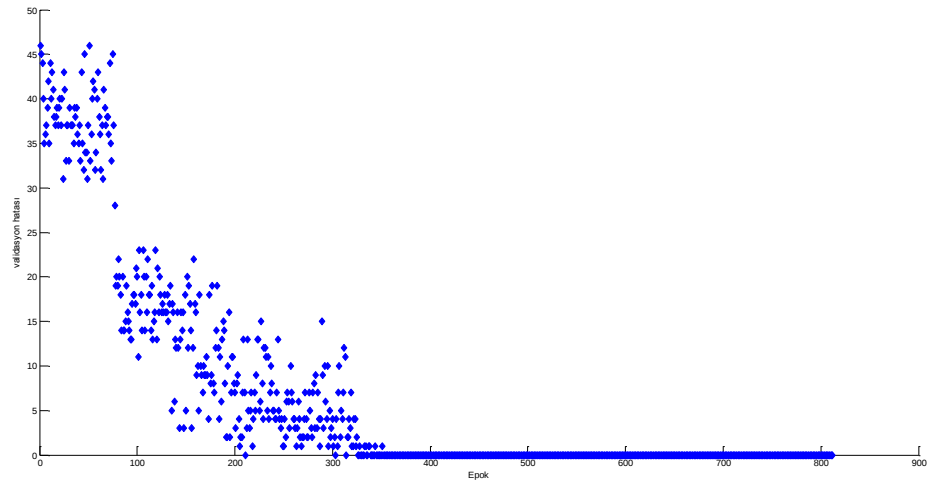
Çizelge EK 4.7. Weather veri kümesi MATLAB YSA eğitimi sonuçları

Epok sayısı	4016
OHK	0,000998
Test edilen girdi sayısı	4
Hata sayısı	0
Hata yüzdesi	0%

Wine Veri Kümesi:



Şekil EK 4.15. Wine veri kümesi MATLAB YSA eğitimi için ortalama hata karesi



Şekil EK 4.16. Wine veri kümesi MATLAB YSA eğitimi için validasyon hatası

Çizelge EK 4.8. Wine veri kümesi MATLAB YSA eğitimi sonuçları

Epok sayısı	811
OHK	0,000964
Test edilen girdi sayısı	59
Hata sayısı	0
Hata yüzdesi	0%

EK 5 Veri Kümesi Bazında MATLAB’de Eğitilmiş YSA Hata Matrisleri

Çizelge EK 5.1. BCW veri kümesi için YSA hata matrisi

SINIF	İYİCİL	KÖTÜCÜL	Sınıflandırma	Kesinlik
İYİCİL	437	7	444	98,42%
KÖTÜCÜL	2	237	239	99,16%
Gerçeklik	439	244	683	
Doğruluk	99,54%	97,13%	Toplam	98,68%

Çizelge EK 5.2. Iris veri kümesi için YSA hata matrisi

SINIF	SETOSA	VERSICOLOR	VIRGINIC	Sınıflandırma	Kesinlik
SETOSA	50	0	0	50	100,00%
VERSICOLOR	0	49	1	50	98,00%
VIRGINIC	0	5	45	50	90,00%
Gerçeklik	50	54	46	150	
Doğruluk	100,00%	90,74%	97,83%	Toplam	96,00%

Çizelge EK 5.3. Monk veri kümesi için YSA hata matrisi

SINIF	İSTENEN	DİĞER	Sınıflandırma	Kesinlik
İSTENEN	216	0	216	100,00%
DİĞER	0	216	216	100,00%
Gerçeklik	216	216	432	
Doğruluk	100,00%	100,00%	Toplam	100,00%

Çizelge EK 5.4. Nursery veri kümesi için YSA hata matrisi

SINIF	ÖNERİLMEYEN	ÖNCELİKLİ	ÖZEL ÖNCELİKLİ	ÖNERİLEN	ÖNCELİKLİ ÖNERİLEN	Sınıflandırma	Kesinlik
ÖNERİLMEYEN	4320	0	0	0	0	4320	100,00%
ÖNCELİKLİ	10	4145	0	47	64	4266	97,16%
ÖZEL ÖNCELİKLİ	0	0	4044	0	0	4044	100,00%
ÖNERİLEN	0	0	0	2	0	2	100,00%
ÖNCELİKLİ ÖNERİLEN	0	0	0	0	328	328	100,00%
Gerçeklik	4330	4145	4044	49	392	12960	
Doğruluk	99,77%	100,00%	100,00%	4,08%	83,67%	Toplam	99,07%

Çizelge EK 5.5. Sunburn veri kümesi için YSA hata matrisi

SINIF	GÜNEŞ YANIĞI OLUR	GÜNEŞ YANIĞI OLMAZ	Sınıflandırma	Kesinlik
GÜNEŞ YANIĞI OLUR	3	0	3	100,00%
GÜNEŞ YANIĞI OLMAZ	0	5	5	100,00%
Gerçeklik	3	5	8	
Doğruluk	100,00%	100,00%	Toplam	100,00%

Çizelge EK 5.6. Titanic veri kümesi için YSA hata matrisi

SINIF	BOĞULUR	BOĞULMAZ	Sınıflandırma	Kesinlik
BOĞULUR	1453	37	1490	97,52%
BOĞULMAZ	427	284	711	39,94%
Gerçeklik	1880	321	2201	
Doğruluk	77,29%	88,47%	Toplam	78,92%

Çizelge EK 5.7. Weather veri kümesi için YSA hata matrisi

	TENİS OYNANIR	TENİS OYNANMAZ	Sınıflandırma	Kesinlik
TENİS OYNANIR	9	1	10	90,00%
TENİS OYNANMAZ	0	4	4	100,00%
Gerçeklik	9	5	14	
Doğruluk	100,00%	80,00%	Toplam	92,86%

Çizelge EK 5.8. Wine veri kümesi için YSA hata matrisi

	SINIF 1	SINIF 2	SINIF 3	Sınıflandırma	Kesinlik
SINIF 1	59	0	0	59	100,00%
SINIF 2	0	71	0	71	100,00%
SINIF 3	0	0	48	48	100,00%
Gerçeklik	59	71	48	178	
Doğruluk	100,00%	100,00%	100,00%	Toplam	100,00%

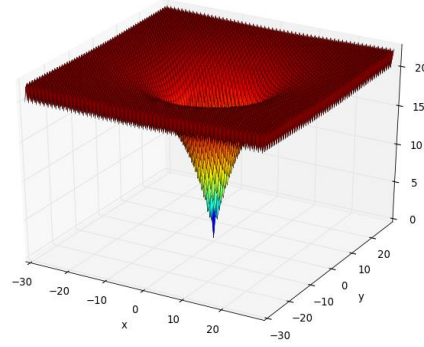
EK 6 İkili Değer Alan Test Fonksiyonları

“Ackley” Fonksiyonu:

Tip: Enküçükleme

$$\text{Fonksiyon: } f(x) = 20 - 20 \exp \left(-0.2 \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \right) + e - \exp \left(\frac{1}{N} \sum_{i=1}^N \cos(2\pi x_i) \right)$$

$$\begin{aligned} \text{Bütünsel enküçük: } & x_i = 0 \quad \forall_i \in \{1 \dots N\} \\ & f(x) = 0 \end{aligned}$$



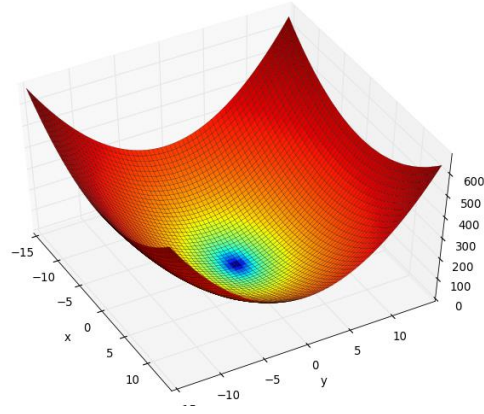
Şekil EK 6.1. Ackley fonksiyonun MATLAB programı ile çizimi

“Bohachevsky” Fonksiyonu:

Tip: Enküçükleme

$$\text{Fonksiyon: } f(x) = \sum_{i=1}^{N-1} (x_i^2 + 2x_{i+1}^2 - 0.3 \cos(3\pi x_i) - 0.4 \cos(4\pi x_{i+1})) + 0.7$$

$$\begin{aligned} \text{Bütünsel enküçük: } & x_i = 0 \quad \forall_i \in \{1 \dots N\} \\ & f(x) = 0 \end{aligned}$$



Şekil EK 6.2. Bohachevsky fonksiyonun MATLAB programı ile çizimi

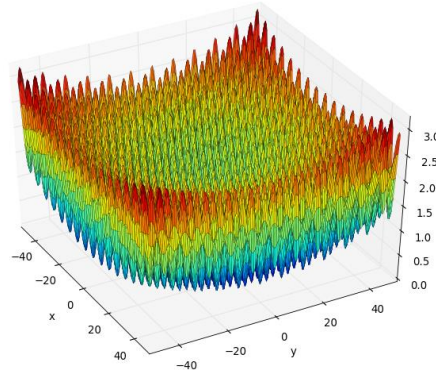
“*Griewank*” Fonksiyonu:

Tip: Enküçkleme

$$\text{Fonksiyon: } f(x) = \frac{1}{4000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$\text{Bütünsel enküçük: } x_i = 0 \quad \forall_i \in \{1 \dots N\}$$

$$f(x) = 0$$



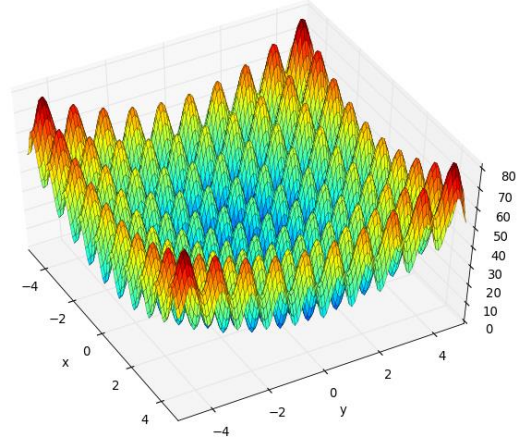
Şekil EK 6.3. Griewank fonksiyonunun MATLAB programı ile çizimi

“*Rastrigin*” Fonksiyonu:

Tip: Enküçkleme

$$\text{Fonksiyon: } f(x) = 10N \sum_{i=1}^N x_i^2 - \cos(2\pi x_i)$$

Bütünsel enküçük: $x_i = 0 \quad \forall_i \in \{1...N\}$
 $f(x) = 0$



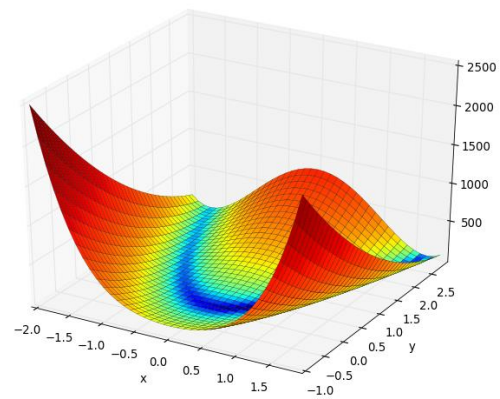
Şekil EK 6.4. Rastrigin fonksiyonunun MATLAB programı ile çizimi

“*Rosenbrock*” Fonksiyonu:

Tip: Enküçkleme

Fonksiyon: $\sum_{i=1}^{N-1} (1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2$

Bütünsel enküçük: $x_i = 1 \quad \forall_i \in \{1...N\}$
 $f(x) = 0$



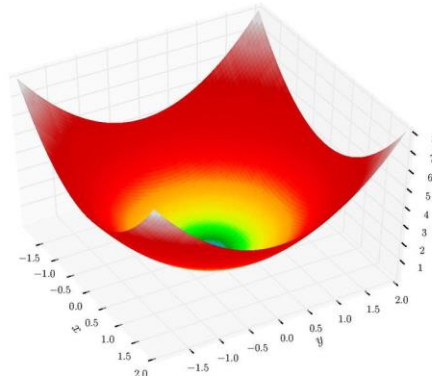
Şekil EK 6.5. Rosenbrock fonksiyonunun MATLAB programı ile çizimi

“Sphere” Fonksiyonu:

Tip: Enküçkleme

Fonksiyon: $f(x) = \sum_{i=1}^N x_i^2$

Bütünsel enküçük: $x_i = 0 \quad \forall_i \in \{1...N\}$
 $f(x) = 0$



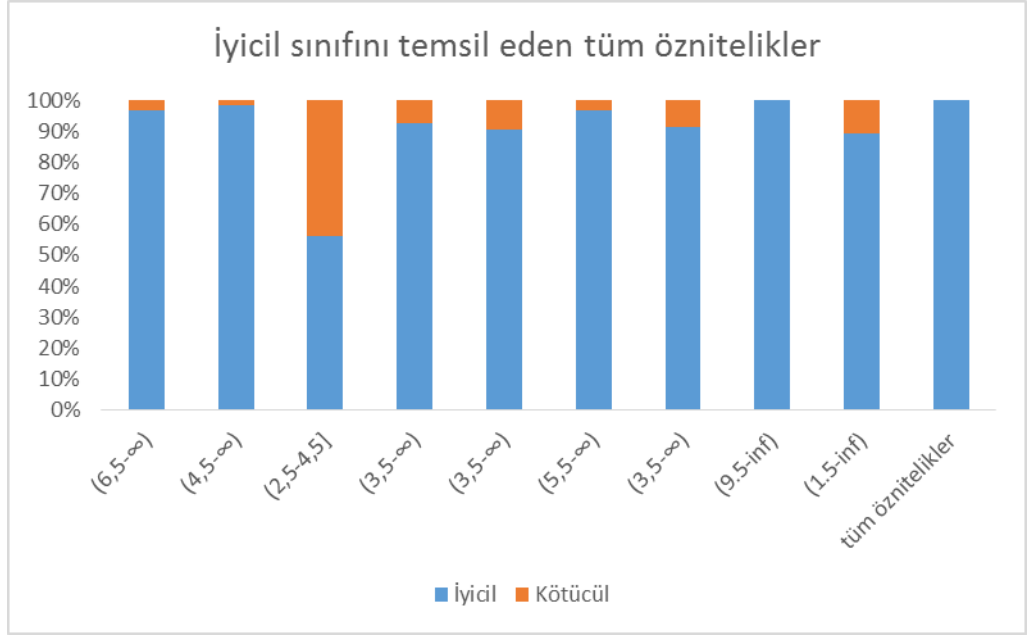
Şekil EK 6.6. Sphere fonksiyonunun MATLAB programı ile çizimi

EK 7 Veri Kümeleri için Kümeleme Analizi Sonuçları

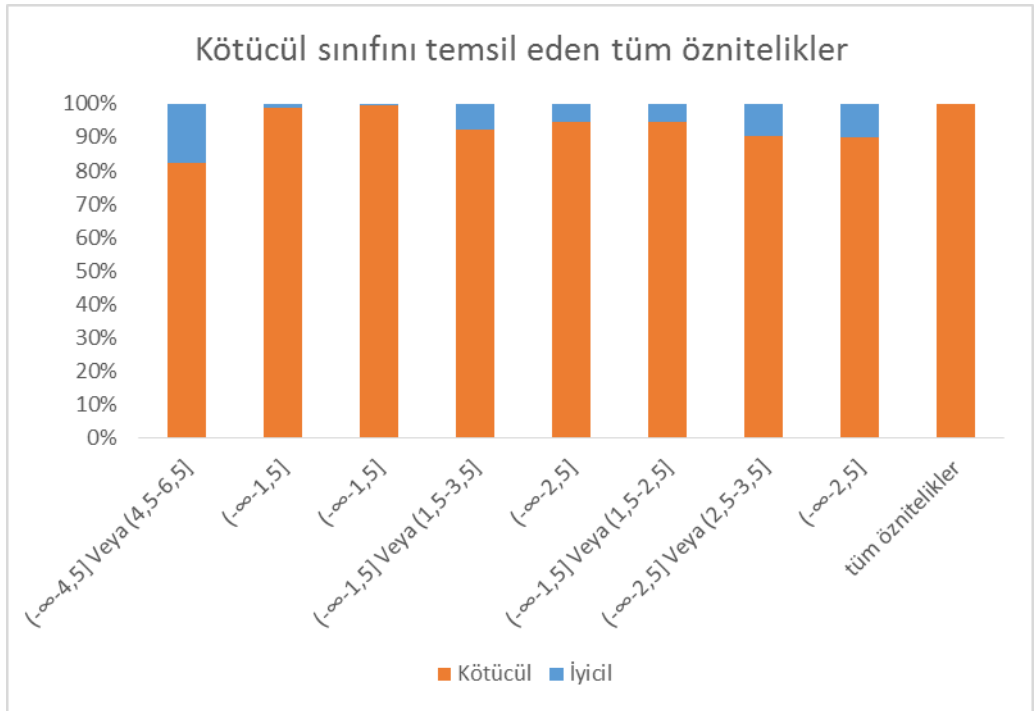
BCW Veri Kümesi:

Çizelge EK 7.1. BCW veri kümesi için anlaşılabilir kurallar

	Kural 1	Kural 2
Koşullar	EĞER parça kalınlığı (6,5-∞) VE hücre boyunun düzgünlüğü (4,5-∞) VE hücre şeklinin düzgünlüğü (2,5-4,5] VE sınır birleşmeleri (3,5-∞) VE tek epitel hücre boyu (3,5-∞) VE çekirdek (5,5-∞) VE kromatin (3,5-∞) VE çekirdekçik (9.5-inf) VE mitoz (1.5-inf) İSE	EĞER parça kalınlığı ($-\infty$-4,5] Veya (4,5-6,5] VE hücre boyunun düzgünlüğü ($-\infty$-1,5] VE hücre şeklinin düzgünlüğü ($-\infty$-1,5] VE sınır birleşmeleri ($-\infty$-1,5] Veya (1,5-3,5] VE tek epitel hücre boyu ($-\infty$-2,5] VE çekirdek ($-\infty$-1,5] Veya (1,5-2,5] VE kromatin ($-\infty$-2,5] Veya (2,5-3,5] VE çekirdekçik ($-\infty$-2,5] İSE
Kümeler	İyicil	Kötücül



Şekil EK 7.1. BCW veri kümesi için kural bazında iyicil sınıf doğruluk oranı



Şekil EK 7.2. BCW veri kümesi için kural bazında kötücül sınıf doğruluk oranı

Çizelge EK 7.2. BCW veri kümesi için kural bazında % doğruluk

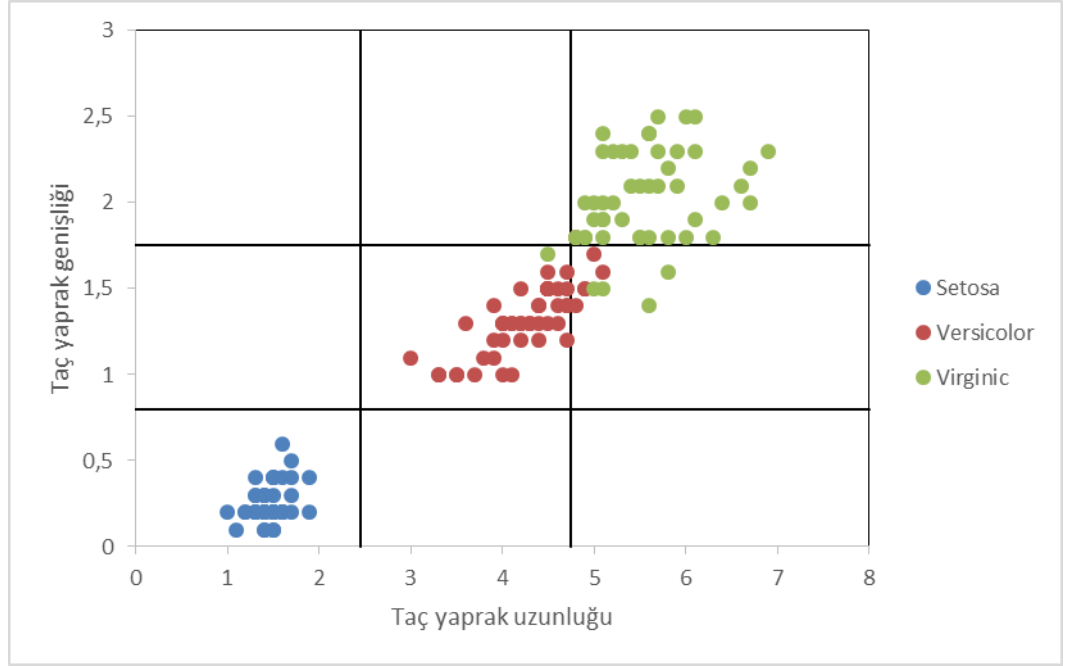
Öznitelik	Sınıf	İyicil	Kötücül	Doğruluk
parça kalınlığı	(6,5- ∞)	145	5	97%
hücre boyunun düzgünlüğü	(4,5- ∞)	172	3	98%
hücre şeklinin düzgünlüğü	(2,5-4,5]	54	42	56%
sınır birleşmeleri	(3,5- ∞)	161	13	93%
tek epitel hücre boyu	(3,5- ∞)	174	18	91%
çekirdek	(5,5- ∞)	168	6	97%
kromatin	(3,5- ∞)	194	18	92%
çekirdekçik	(9,5-inf)	60	0	100%
mitoz	(1,5-inf)	107	13	89%
tüm öznitelikler	tüm öznitelikler	7	0	100%

parça kalınlığı	($-\infty$ -4,5] Veya (4,5-6,5]	94	439	82%
hücre boyunun düzgünlüğü	($-\infty$ -1,5]	4	369	99%
hücre şeklinin düzgünlüğü	($-\infty$ -1,5]	2	344	99%
sınır birleşmeleri	($-\infty$ -1,5] Veya (1,5-3,5]	30	363	92%
tek epitel hücre boyu	($-\infty$ -2,5]	22	398	95%
çekirdek	($-\infty$ -1,5] Veya (1,5-2,5]	24	408	94%
kromatin	($-\infty$ -2,5] Veya (2,5-3,5]	45	426	90%
çekirdekçik	($-\infty$ -2,5]	47	421	90%
tüm öznitelikler	tüm öznitelikler	0	279	100%

Iris Veri Kümesi:

Çizelge EK 7.3. Iris veri kümesi için anlaşılabilir kurallar

Kurallar	Koşullar	Kümeler
Kural 1	EĞER taç yaprak uzunluğu ($-\infty$ -2,45] VE taç yaprak genişliği ($-\infty$ -0,8]	Setosa
Kural 2	EĞER taç yaprak uzunluğu (2,45-4,75] VE taç yaprak genişliği (0,8-1,75]	Versicolor
Kural 3	EĞER taç yaprak genişliği (1,75- ∞)	Virginic
Küme oluşumunda genel belirleyiciler taç yaprak genişliği ve uzunluğudur.		



Şekil EK 7.3. Iris veri kümesi için kural bazında tüm sınıf kümeleri

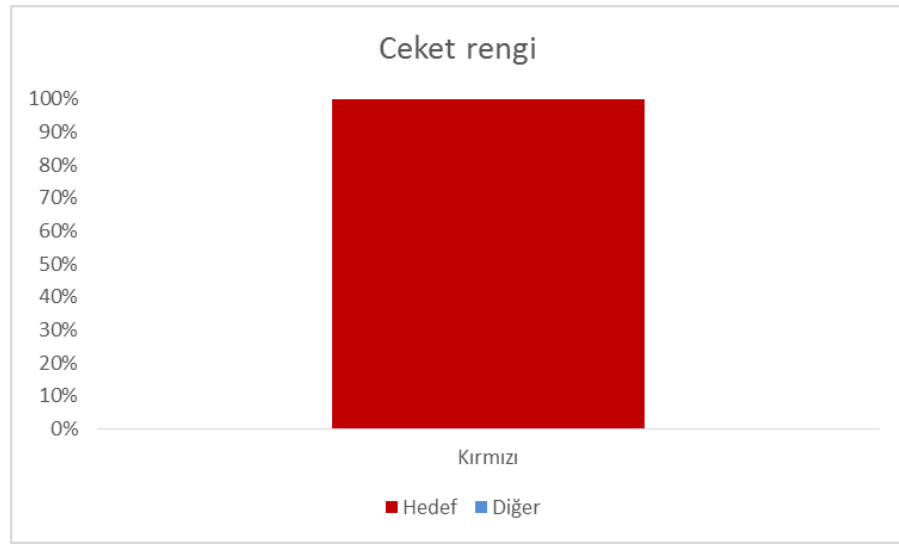
Çizelge EK 7.4. Iris veri kümesi için kural bazında % doğruluk

Öznitelik	Sınıf	Setosa	Versicolor	Virginic	Doğruluk
Taç yaprak uzunluğu	Birinci aralık	49	0	0	100%
	İkinci aralık	0	44	1	98%
	Üçüncü aralık	0	6	48	89%
Taç yaprak genişliği	Birinci aralık	49	0	0	100%
	İkinci aralık	0	49	5	91%
	Üçüncü aralık	0	1	44	98%
Taç yaprak uzunluğu- genişliği	Birinci-Birinci aralık	49	0	0	100%
	İkinci-İkinci aralık	0	45	0	100%
	Üçüncü-Üçüncü aralık	0	1	44	98%

Monk Veri Kümesi:

Çizelge EK 7.5. Monk veri kümesi için anlaşılabilir kurallar

Kurallar	Koşullar			Kümeler
Kural 1	EĞER	ceket rengi (kırmızı)	İSE	İstenen
Kural 2	EĞER	kafa şekli == vücut şekli	İSE	Diğer
Küme oluşumunda genel belirleyici ceket rengi ve kafa ve vücut şeklinin aynılığıdır				



Şekil EK 7.4. Monk veri kümesi için kural bazında istenen sınıf doğruluk oranı



Şekil EK 9.5. Monk veri kümesi için kural bazında istenen diğer doğruluk oranı

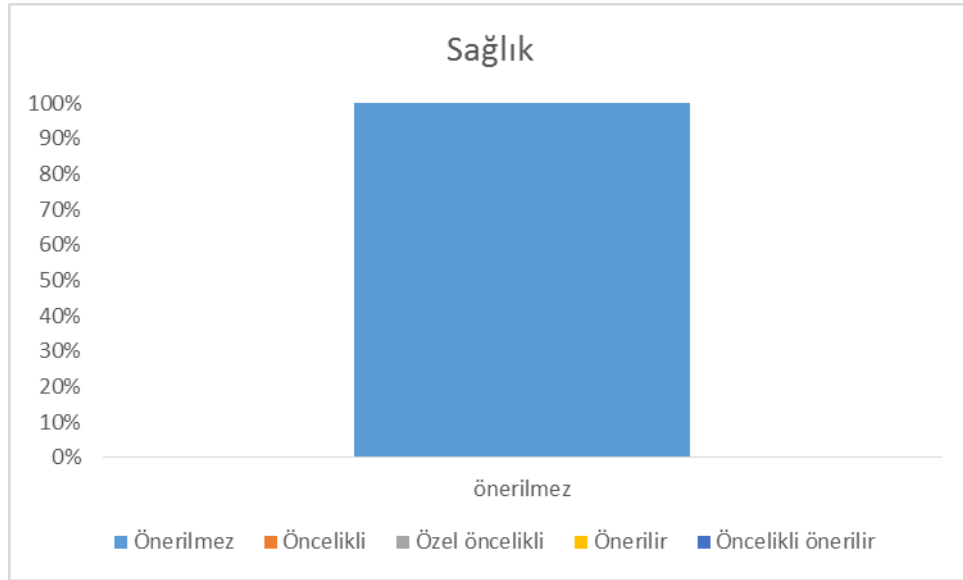
Çizelge EK 7.6. Monk veri kümesi için kural bazında % doğruluk

Öznitelik	Sınıf	Hedef	Diğer	Doğruluk
Ceket rengi	Kırmızı	108	0	100%
Kafa-vücut şekli	Kafa şekli=Vücut şekli	144	0	100%

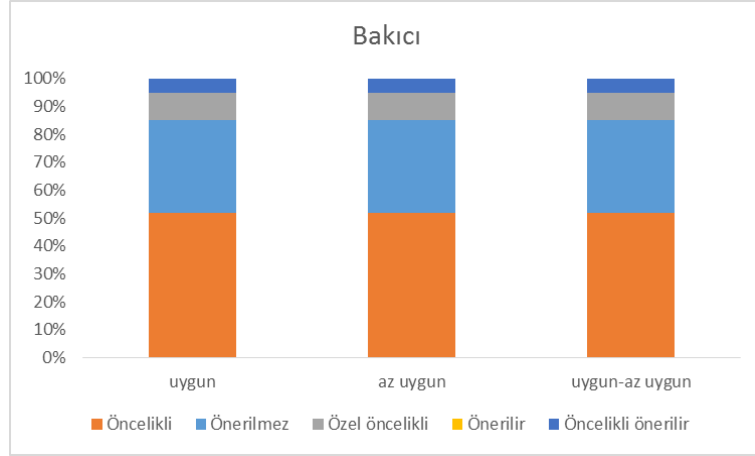
Nursery Veri Kümesi:

Çizelge EK 7.7. Nursery veri kümesi için anlaşılabilir kurallar

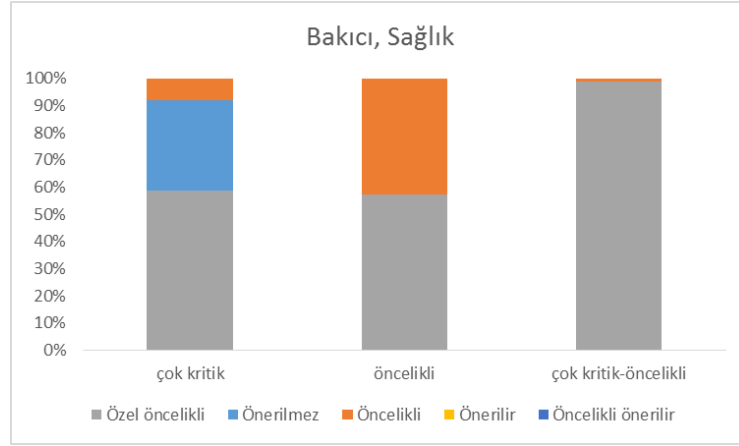
Kurallar	Koşullar	Kümeler
Kural 1	EĞER sağlık (önerilmez) İSE	Önerilmez
Kural 2	EĞER bakıcı (uygun) Veya (az uygun) VE İSE	Öncelikli
Kural 3	EĞER bakıcı (çok kritik) VE sağlık (öncelikli) İSE	Özel öncelikli
Küme oluşumunda genel belirleyiciler bakıcı ve sağlık durumudur.		



Şekil EK 7.6. Nursery veri kümesi için kural bazında önerilmez sınıf doğruluk oranı



Şekil EK 7.7. Nursery veri kümesi için kural bazında öncelikli sınıf doğruluk oranı



Şekil EK 7.8. Nursery veri kümesi için kural bazında özel öncelikli sınıf doğruluk oranı

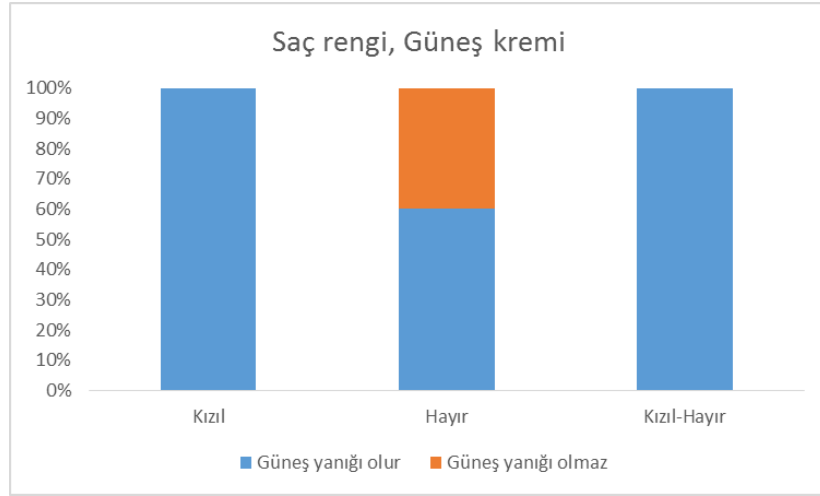
Çizelge EK 7.8. Nursery veri kümesi için kural bazında % doğruluk

Öznitelik	Sınıf	Önerilmez	Öncelikli	Özel öncelikli	Önerilir	Öncelikli önerilir	Doğruluk
Sağlık	önerilmez	4320	0	0	0	0	100%
Bakıcı	uygun	864	1344	252	2	130	52%
	az uygun	864	1344	252	0	132	52%
	uygun-az uygun	1728	2688	504	2	262	52%
Bakıcı	çok kritik	864	210	1518	0	0	59%
Sağlık	öncelikli	0	1854	2466	0	0	57%
Bakıcı-Sağlık	çok kritik-öncelikli	0	9	855	0	0	99%

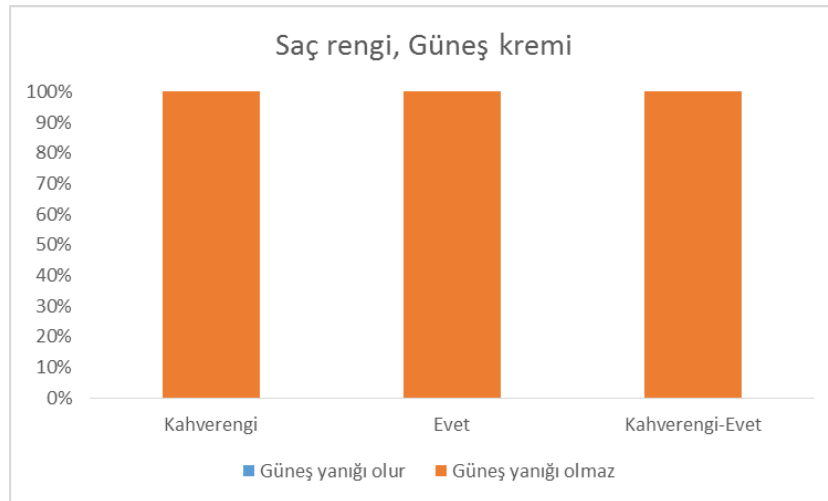
Sunburn Veri Kümesi:

Çizelge EK 7.9. Sunburn veri kümesi için anlaşılabilir kurallar

Kurallar	Koşullar	Kümeler
Kural 1	EĞER saç rengi (kızıl) VE güneş kremi (hayır) İSE	Güneş yanığı olur
Kural 2	EĞER saç rengi (kahverengi) VE güneş kremi (evet) İSE	Güneş yanığı olmaz
Küme oluşumunda genel belirleyiciler saç rengi ve güneş kremidir		



Şekil EK 7.9. Sunburn veri kümesi için kural bazında güneş yanığı olur sınıf doğruluk oranı



Şekil EK 7.10. Sunburn veri kümesi için kural bazında güneş yanığı olmaz sınıf doğruluk oranı

Çizelge EK 7.10. Sunburn veri kümesi için kural bazında % doğruluk

Öznitelik	Sınıf	Güneş yanığı olur	Güneş yanığı olmaz	Doğruluk
Saç rengi	Kızıl	1	0	100%
Güneş kremi	Hayır	3	2	60%
Saç rengi-Güneş kremi	Kızıl-Hayır	1	0	100%

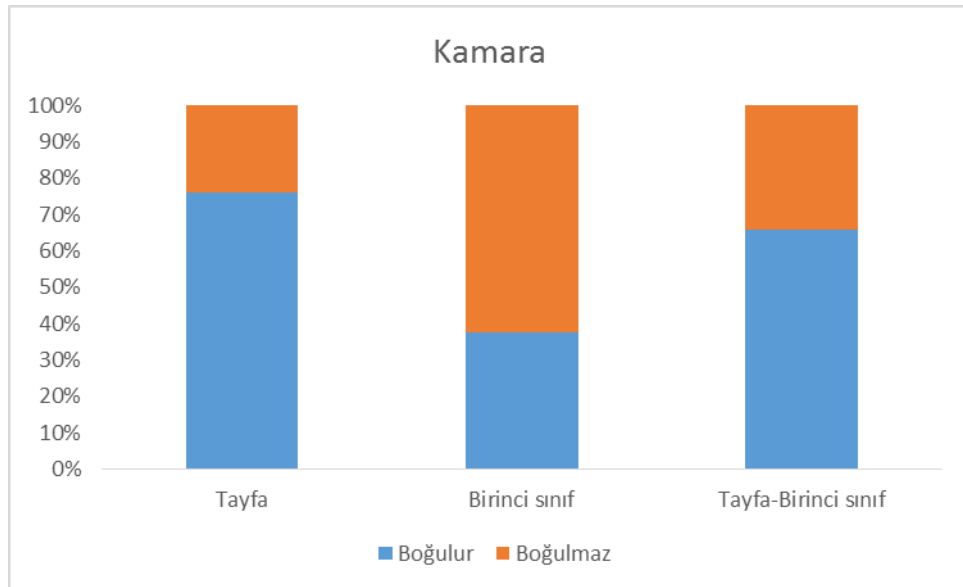
Saç rengi	Kahverengi	0	3	100%
Güneş kremi	Evet	0	3	100%
Saç rengi-Güneş kremi	Kahverengi-Evet	0	1	100%

Titanic Veri Kümesi:

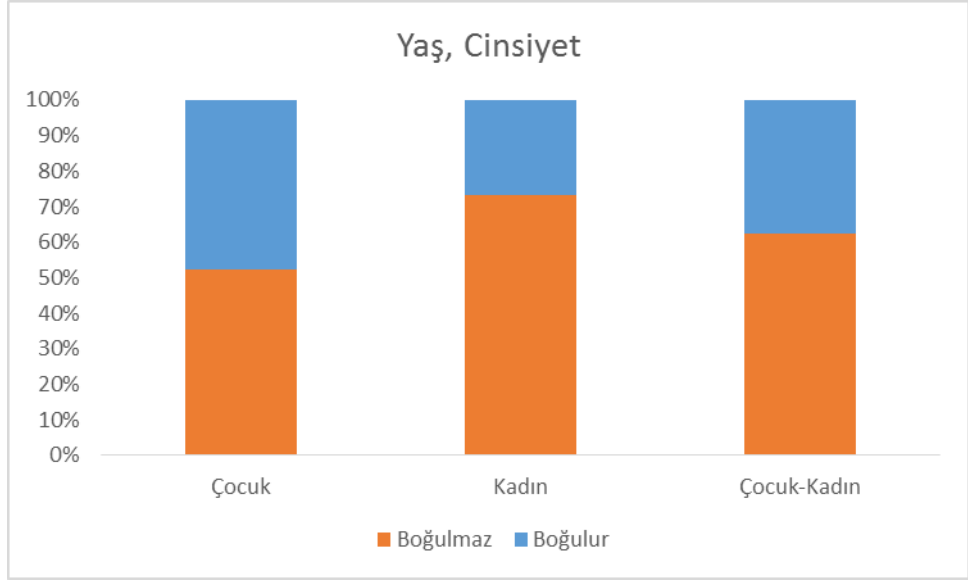
Çizelge EK 7.11. Titanic veri kümesi için anlaşılabilir kurallar

Kurallar	Koşullar	Kümeler
Kural 1	EĞER sınıf (crew) Veya (first class) İSE	Boğulur
Kural 2	EĞER yaş (çocuk) VE cinsiyet (bayan) İSE	Boğulmaz

Küme oluşumunda genel belirleyiciler sınıf, yaş ve cinsiyettir.



Şekil EK 7.11. Titanic veri kümesi için kural bazında boğulur sınıf doğruluk oranı



Şekil EK 7.12. Titanic veri kümesi için kural bazında boğulmaz sınıf doğruluk oranı

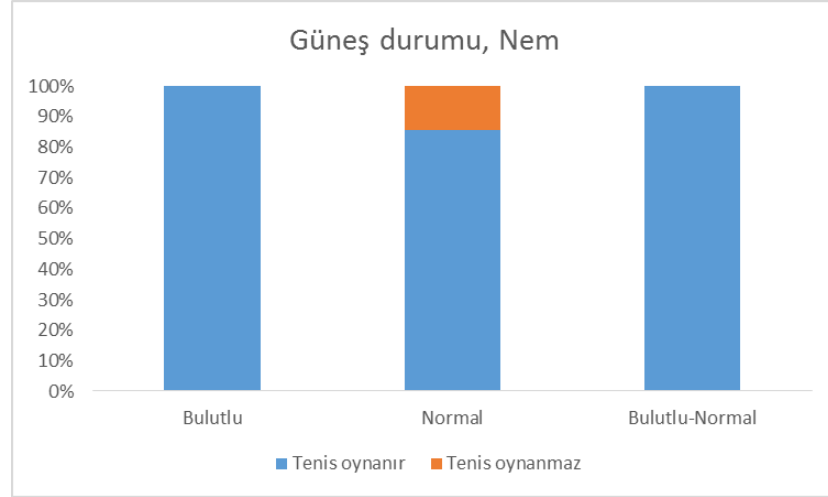
Çizelge EK 7.12. Titanic veri kümesi için kural bazında % doğruluk

Öznitelik	Sınıf	Boğulur	Boğulmaz	Doğruluk
Kamara	Tayfa	673	212	76%
	Birinci sınıf	122	203	38%
	Tayfa-Birinci sınıf	795	415	66%
Yaş	Çocuk	52	57	52%
Cinsiyet	Kadın	126	344	73%
Yaş-Cinsiyet	Çocuk-Kadın	17	28	62%

Weather Veri Kümesi:

Çizelge EK 7.13. Weather veri kümesi için anlaşılabilir kurallar

Kurallar	Koşullar	Kümeler
Kural 1	EĞER güneş durumu (bulutlu) VE nem (normal) İSE	Tenis oynanır
Küme oluşumunda genel belirleyiciler güneş durumu ve nemdir		



Şekil EK 7.13. Weather veri kümesi için kural bazında tenis oynanır sınıf doğruluk oranı

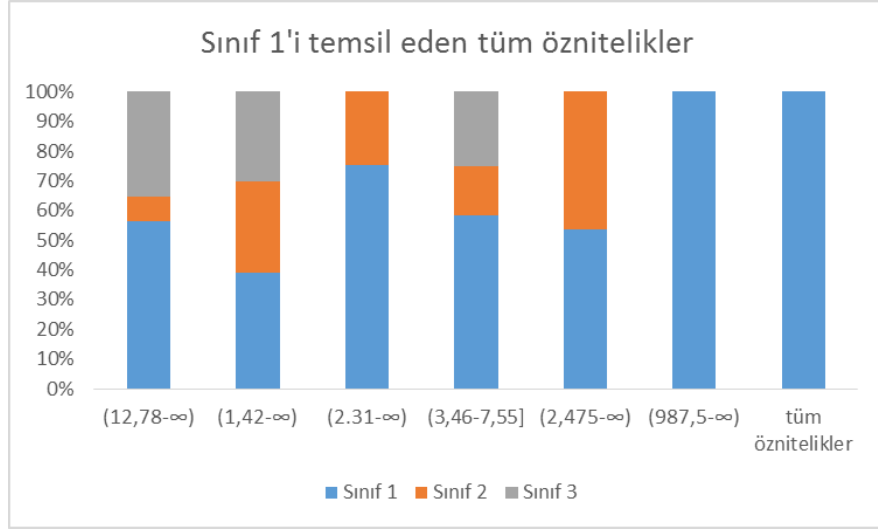
Çizelge EK 7.14. Weather veri kümesi için kural bazında % doğruluk

Öznitelik	Sınıf	Tenis oynanır	Tenis oynanmaz	Doğruluk
Güneş durumu	Bulutlu	4	0	100%
Nem	Normal	6	1	86%
Güneş durumu-Nem	Bulutlu-Normal	2	0	100%

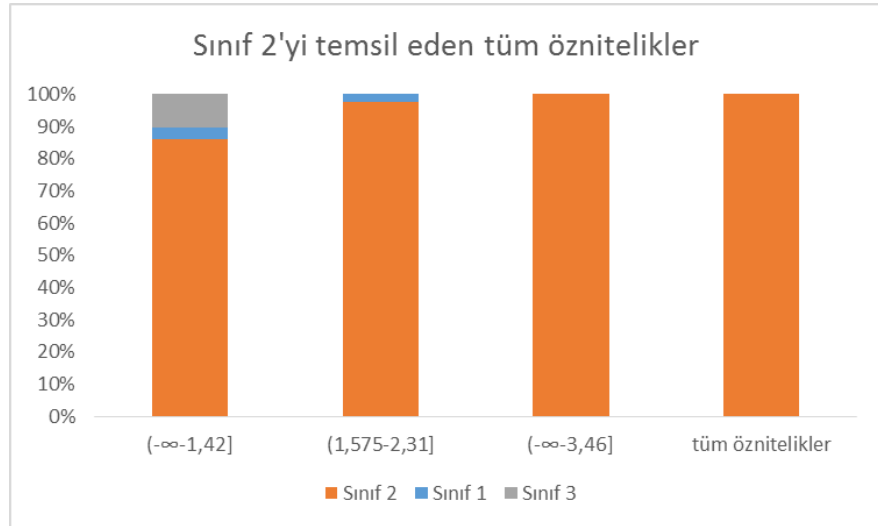
Wine Veri Kümesi:

Çizelge EK 7.15. Wine veri kümesi için anlaşılabilir kurallar

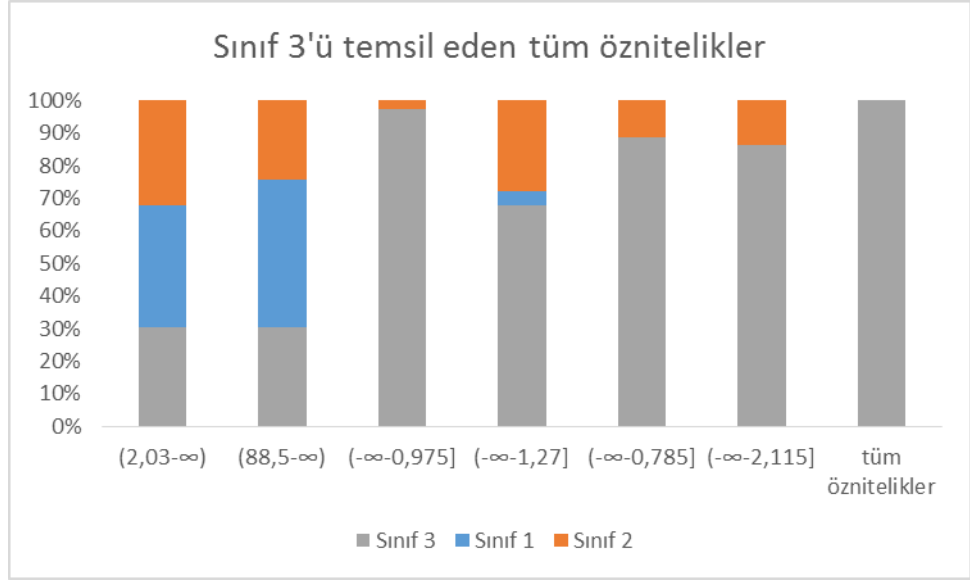
	Kural 1	Kural 2	Kural 3
Koşullar	EĞER alkol (12,78-∞)	EĞER malik asit (-∞-1,42]	EĞER kül (2,03-∞)
	VE malik asit (1,42-∞)	VE flavanoidler (1,575-2,31]	VE magnezyum (88,5-∞)
	VE flavanoidler (2,31-∞)	VE renk yoğunluğu (-∞-3,46]	VE flavanoidler (-∞-0,975]
	VE renk yoğunluğu (3,46-7,55]		VE proantosiyeninler (-∞-1,27]
	VE OD280/OD315 oranı (2,475-∞)		VE renk derecesi (-∞-0,785]
	VE prolin (987,5-∞)		VE OD280/OD315 oranı (-∞-2,115]
	İSE	İSE	İSE
Kümeler	Sınıf 1	Sınıf 2	Sınıf 3



Şekil EK 7.14. Wine veri kümesi için kural bazında sınıf 1 doğruluk oranı



Şekil EK 7.15. Wine veri kümesi için kural bazında sınıf 2 doğruluk oranı



Şekil EK 7.16. Wine veri kümesi için kural bazında sınıf 3 doğruluk oranı

Çizelge EK 7.16. Wine veri kümesi için kural bazında % doğruluk

Öznitelik	Sınıf	Sınıf 1	Sınıf 2	Sınıf 3	Doğruluk
alkol	(12,78-∞)	59	9	37	56%
malik asit	(1,42-∞)	58	46	45	39%
flavanoidler	(2,31-∞)	58	19	0	75%
renk yoğunluğu	(3,46-7,55]	56	16	24	58%
OD280/OD315 oranı	(2,475-∞)	59	51	0	54%
prolin	(987,5-∞)	44	0	0	100%
tüm öznitelikler	tüm öznitelikler	39	0	0	100%

malik asit	(-∞-1,42]	1	25	3	86%
flavanoidler	(1,575-2,31]	1	38	0	97%
renk yoğunluğu	(-∞-3,46]	0	55	0	100%
tüm öznitelikler	tüm öznitelikler	0	10	0	100%

kül	(2,03-∞)	59	51	48	30%
magnezyum	(88,5-∞)	59	32	40	31%
flavanoidler	(-∞-0,975]	0	1	38	97%
proantosiyaninler	(-∞-1,27]	2	14	34	68%
renk derecesi	(-∞-0,785]	0	5	40	89%
OD280/OD315 oranı	(-∞-2,115]	0	7	45	87%
tüm öznitelikler	tüm öznitelikler	0	0	11	100%

ÖZGEÇMİŞ

Adı Soyadı	: Özlem KANGA
Doğum Yeri ve Tarihi	: Bursa 29/05/1983
Yabancı Dili:	: İngilizce, Almanca
Eğitim Durumu	
Lise	: Özel Amerikan Robert Koleji/ 2001
Lisans	: İTÜ Kimya Mühendisliği/ 2005
Yüksek Lisans	: Sabancı Üniversitesi MBA/ 2007
Çalıştığı Kurum	: UÜ Endüstri Mühendisliği Bölümü/2011-...
İletişim	: ozlemkanga@uludag.edu.tr