T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

METEOROLOJİK VERİLERİN AKILLI YÖNTEMLERLE SINIFLANDIRILMASI

Ömer Osman DURSUN

Tez Yöneticisi
Prof. Dr. Asaf VAROL

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

ELAZIĞ, 2005

T.C.

FIRAT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

METEOROLOJİK VERİLERİN AKILLI YÖNTEMLERLE SINIFLANDIRILMASI

Ömer Osman DURSUN

Yüksek Lisans Tezi

Elektronik ve Bilgisayar Eğitimi Anabilim Dalı

Bu tez, 02/09/2005 tarihinde aşağıda belirtilen jüri tarafından oybirliği /~~oyçokluğu~~ ile başarılı / başarısız olarak değerlendirilmiştir.

Üye: Doç. Dr. Z. Hakan AKPOLAT

Üye: Yrd. Doç. Dr. İbrahim TÜRKOĞLU

Üye: Yrd. Doç. Dr. Zafer AYDOĞMUŞ

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 07/09/2005 tarih ve 2005-291/8 sayılı kararıyla onaylanmıştır.

TEŞEKKÜR

Bu tez çalışması süresince benden yardımlarını esirgemeyen danışman hocam Sayın Prof. Dr. Asaf VAROL'a, Yrd. Doç. Dr. İbrahim TÜRKOĞLU'na, Arş. Gör. Engin AVCI'ya, Arş. Gör. Abdülkadir ŞENGÜR'e, Arş. Gör. Ferhat BAĞÇACI'ya, Arş. Gör. Murat KARABATAK'a, Arş. Gör. Davut HANBAY'a, bölümdeki diğer hocalarıma ve çok sevdiğim arkadaşlarım Cihan VAROL'a, Suat TORAMAN'a maddi, manevi her türlü desteği sağlayan çok sevdiğim ve değer verdiğim aileme teşekkürü bir borç bilirim.

Ömer Osman DURSUN

İÇİNDEKİLER

Sayfa

TEŞEKKÜR
İÇİNDEKİLER ..... **I**
ŞEKİLLER LİSTESİ ..... **III**
TABLOLAR LİSTESİ ..... **IV**
SİMGELER LİSTESİ ..... **V**
KISALTMALAR LİSTESİ ..... **VI**
ÖZET ..... **VII**
ABSTRACT ..... **VIII**

1. GİRİŞ **1**

2. ÖRÜNTÜ TANIMA **4**
2.1. Örüntü Tanıma Sistemleri 5
2.2. Örüntü Tanımada Öğrenme 8
2.2.1. Eğiticili ve Eğiticisiz Öğrenme 10
2.2.2. Veri Uygunluğu ve Genelleme 11
2.3. Örüntü Tanıma Sistemlerinin Bileşenleri 11
2.3.1. Özellik Çıkarma 12
2.3.2. Sınıflandırma 14
2.4. Örüntü Tanıma ve Veri Madenciliği 14

3. VERİ MADENCİLİĞİ **15**
3.1. Veri Tabanı Kavramı ve Veri Tabanlarının Tarihsel Gelişimi 15
3.2. Veri Madenciliğinin Tanımı 16
3.3. Veri Madenciliğinde Karşılaşılan Problemler 22
3.3.1. Veri Tabanı Boyutu 22
3.3.2. Gürültülü Veri 23
3.3.3. Boş Değerler 23
3.3.4. Eksik Veri 24
3.3.5. Artık Veri 24
3.3.6. Dinamik Veri 24
3.4. Veri Tabanlarında Bilgi Keşfi Süreci 25
3.4.1. Problemin Tanımlanması 25
3.4.2. Verilerin Hazırlanması 25

3.4.2.1. Toplama 25
3.4.2.2. Değer Biçme 26
3.4.2.3. Birleştirme ve Temizleme 26
3.4.2.4. Seçim 26
3.4.2.5. Dönüştürme 27
3.4.3. Modelin Kurulması ve Değerlendirilmesi 27
3.4.4. Modelin Kullanılması 30
3.4.5. Modelin İzlenmesi 30

4. SINIFLANDIRICILAR 31
4.1. Klasik sınıflandırıcılar 31
4.2. Bulanık sınıflandırıcılar 33
4.2.1. Matematiksel temel 33
4.2.2. Eğitme 34
4.2.3. Test etme 34
4.3. Yapay sinir ağları sınıflandırıcısı 35
4.3.1. Matematiksel temel 39
4.3.2. Eğitme 41
4.3.3. Test etme 42
4.4. Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi ile Sınıflandırma 42

5. UYGULAMA 45
5.1. Veri Madenciliği Çalışması Örnekleri 45
5.2. Elazığ İlinin Meteorolojik Verilerinin Sınıflandırılıp Değerlendirilmesi 48
5.2.1 Verilerin Sınıflandırılması 49
5.2.2. Sınıflandırılan Verilerin Değerlendirilmesi ve Sınıflandırıcı Performansları 58

6. SONUÇLAR VE ÖNERİLER 65
6.1. Sonuçlar ve Tartışma 65
6.2. Öneriler 65

KAYNAKLAR 67
ÖZGEÇMİŞ 69
EK-I 70

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1. Örüntü tanıma kavramı 5
Şekil 2.2. Örüntü tanıma sistemi 5
Şekil 2.3. Yapısal örüntü tanıma sistemi 6
Şekil 2.4. Akıllı örüntü tanıma yaklaşımı 7
Şekil 2.5. Öğrenme süreci 10
Şekil 2.6. Test etme süreci 10
Şekil 2.7. Örüntü tanıma sistemi 12
Şekil 3.1. Veri tabanı teknolojisinin evrimi 16
Şekil 3.2. Veri tabanı- Veri ambarı- Standart form 20
Şekil 3.3. Veri tabanlarında bilgi keşfi süreci ve veri madenciliği 21
Şekil 3.4. Denetimli Öğrenme 28
Şekil 4.1. Veri tabanındaki sınıflar ve en yakın komşu yöntemine göre k vektörünün sınıflandırılması 32
Şekil 4.2. Bulanık sınıflandırıcı 35
Şekil 4.3. Biyolojik nöronun şematik yapısı 36
Şekil 4.4.Yapay sinir ağ örüntü sınıflandırıcıları 39
Şekil 4.5. Çok katmanlı ileri beslemeli sinir ağ sınıflandırıcısı 41
Şekil 4.6. 2 girişli 9 kurallı bir ANFIS sınıflandırıcı yapısı 43
Şekil 5.1. Örüntü tanıma işlem süreci 49
Şekil 5.2. VM işlem süreci 49
Şekil 5.3. İşlem sürecinin basit gösterimi 52
Şekil 5.4. ANFIS sınıflandırıcısıyla eğitilen giriş verilerinin ve elde edilen çıkışın aynı eksen üzerindeki grafiği 53
Şekil 5.5. Arzu edilen çıkışla YSA'nın bulmuş olduğu çıkış arasındaki farkın grafiği..... 53

TABLOLAR LİSTESİ

Sayfa

Tablo 3.1. Sınıflar 29
Tablo 5.1. Bir veri tabanı uygulaması. 46
Tablo 5.2. Rakamsal değerlerle bir veri tabanı uygulaması 46
Tablo 5.3. Veri tabanında sınıflandırıcının eğitim sürecinde kullanılmak üzere tutulan veriler 50
Tablo 5.4. Karar sınıfları 52
Tablo 5.5. Bir günlük meteorolojik veri 52
Tablo 5.6. Test süreci sonucu elde edilen çıkışlar 54
Tablo 5.7. Sınıflandırıcıların performansı 59
Tablo 5.8. ANFIS sınıflandırıcısının kural tabanı arttırıldığında elde edilen çıkışlar 59

SİMGELER LİSTESİ

ϕ	: Modelin değişkenleri karakteristikleri
M	: Fonksiyonların uzayı
ψ	: Eğitme verisindeki bağımsız değişken
Ω_ψ	: Eğitme verisindeki bağımlı değişken
N	: Eğitme verisindeki örneklerin sayısı
ζ	: Test verisindeki bağımsız değişken
Ω_ζ	: Test verisindeki bağımlı değişken
E	: Test verisinin sınıflandırmasındaki hata
ε	: Öğrenme katsayısı
Net	: Giriş nesne öznitelik vektörü ile ağırlık vektörünün çarpımının toplamı
Nc	: Komşuluk kümesi
X	: Giriş nesne öznitelik vektörü
W	: Ağırlık vektörü,
msf( )	: İki küme merkezi arasındaki öklit uzaklığı
(x_a,y_a)	: Nesnenin ağırlık merkez koordinatı
YSA	: Yapay Sinir Ağı
H	:Gizli birimlerin sayısı
β	: Öğrenme oranı

KISALTMALAR LİSTESİ

ANFIS	: Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi
OLAP	: Online Analitik İşleme
RKDS	: Radar Karar Destek Sistemi
VA	: Veri Ambarı
VM	: Veri Madenciliği
VP	: Veri Pazarı
VS	: Veri Stoku
VTBK	: Veri Tabanı Bilgi Keşfi
VTYS	: Veri Tabanı Yönetim Sistemleri
YSA	: Yapay Sinir Ağı

ÖZET

Yüksek Lisans Tezi

METEOROLOJİK VERİLERİN AKILLI YÖNTEMLERLE SINIFLANDIRILMASI

Ömer Osman DURSUN

Fırat Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik ve Bilgisayar Eğitimi Anabilim Dalı

2005, Sayfa: 71

Bu tez çalışmasında, veri madenciliği konusunda yapılan araştırmalar ve yöntemler incelenmiştir. Elazığ iline ait, tek başına bir anlam ifade etmeyen sıcaklık, çiğ noktası, nem, basınç verileri yapay sinir ağı ve uyarlamalı ağ tabanlı bulanık çıkarım tarzı sınıflandırıcılardan geçirilerek hava tahmininin yapılması sağlanmıştır. Bunun yanı sıra yapay sinir ağı ve uyarlamalı ağ tabanlı bulanık çıkarım sınıflandırıcılarının performansları karşılaştırılmış, hangi sınıflandırıcının daha iyi sonuç verdiği saptanmıştır.

Anahtar Kelimeler: Veri madenciliği, Sınıflandırma, Hava tahmini, Yapay sinir ağı, Uyarlamalı ağ tabanlı bulanık çıkarım.

ABSTRACT

Master Thesis

THE CLASSIFICATION OF THE METEOROLOGICAL DATA USING THE INTELLIGENT METHODS

Ömer Osman DURSUN

Firat University

Graduate School of Natural and Applied Sciences

Department of Electronics and Computer Education

2005, Page: 71

In this thesis, we examine the methods and researches about the data mining. The weather forecast of Elazig is done using Elazig's temperature, dew point, humidity, air pressure data that are stand alone meaningless by neural network and adaptive network based fuzzy inference system classifiers. Moreover neural network and adaptive network based fuzzy inference system classifiers performance are compared. Which classifiers are gives a better result is determined by us.

Keywords: Data mining, Classification, Weather forecast, Neural network, Adaptive network based fuzzy inference system.

1. GİRİŞ

Gelişen iletişim teknolojileri sayesinde dünyamız küresel bir hal almaktadır. Dünya üzerinde herhangi bir ülkeye ya da kıtaya ulaşmak, bilgi iletişimi sağlamak daha çok kolay hale gelmiştir. Küreselleşmenin getirdiği rekabet ortamı, insanların bilgiye en hızlı ulaşmasını ve bunu en az maliyetle gerçekleştirdiğini ortaya koymuştur [1].

Verilerin dijital ortamda saklanmaya başlanması ile birlikte, yeryüzündeki bilgi miktarının her 20 ayda bir kendini iki katına çıkardığı günümüzde, veri tabanlarının sayısı da benzer hatta daha yüksek bir oranda artmaktadır. Yüksek kapasiteli işlem yapabilme gücünün ucuzlamasının bir sonucu olarak, veri saklama hem daha kolay olabilmekte hem de verinin kendisi de maliyet açısından daha ucuza elde edilebilmektedir [2].

Günümüzde oldukça yaygınlaşan elektronik ticaret ve internet üzeri alışveriş mekanizmalarının da artmasıyla birlikte, bu alanda birbirlerine rakip olan firmaların çalışmaları, örüntü tanıma ve onun alt bir dalı olan veri madenciliğinin önemini ön plana çıkarmaktadır [3].

Örüntü tanıma, aralarında ortak özellik bulunan ve aralarında bir ilişki kurulabilen karmaşık işaret örneklerini veya nesneleri bazı tespit edilmiş özellikler veya karakterler vasıtası ile tanımlama veya sınıflandırmadır. Bu bağlamda, örüntü tanımanın en önemli amaçları; bilinmeyen örüntü sınıflarına belirli bir şekil vermek ve bilinen bir sınıfa ait olan örüntüyü teşhis etmektir [3].

Örüntü tanıma tekniklerinin uygulamaları birçok mühendislik, tıp, askeri ve bilim alanına açıktır. Bunlardan bazıları; ses tanıma, haberleşme işaretlerini tanıma ve radar hedef sınıflama, biyomedikal kontrol, veri madenciliği verilebilir. Örüntü tanıma olarak bilinen bu uygulamalar, makine öğrenmesi, örüntü sınıflandırma, ayrım analizi ve nitelik tahmini gibi isimlerle de anılmaktadır [3].

Örüntü tanıma, görüntü tanıma, işaret işleme ve veri madenciliği gibi pek çok konuyu içine alan geniş içerikli bir konudur. VM (Veri Madenciliği) veri tabanlarında tutulan anlamsız veriler bütününden bir anlamlı veri elde etme sürecidir örüntü tanıma ise daha veriler veri tabanına gelmeden başlayan bir süreçtir [3].

Zaman ilerledikçe bilgi çağının daha dip noktalarına doğru ulaşıldığı aşikârdır. Birkaç sene öncesine kadar defter ve kalemle hesaplanan fatura-gelir-gider türü bilgiler, artık çok farklı boyutlarıyla birlikte bilgisayarla hesaplanmaktadır. Verilerin dijital ortamda tutulması ve hesaplanmasının faydaları göz ardı edilemez. Geçmiş yirmi yıl bize elektronik ortamlarda depolanmaya başlayan bilgi ve veri miktarının her geçen gün katlanarak arttığını ve söz konusu artış verilerinin saklanması için ihtiyaç duyulan depolama ortamlarının da büyümesinin gerekliliğini göstermektedir. Verilerin saklanması için dünyada gerekli olan alan ihtiyacı her 20

ayda büyüklük ve sayıları bakımından ikiye katlanmakta hatta bunun tahmin edilenden daha hızlı olduğu düşünülmektedir. Bilgisayar ile hedeflenen sonuçlara daha hızlı ulaşabilme ve düzenli kayıt tutulması bunun yanında hiç kuşkusuz günümüz teknolojisinde işlemci hızlarının artması, daha fazla veri saklayabilme, işleme koyabilme yeteneğinin artması ve buna rağmen bilgisayar fiyatlarının ucuzlaması paralelinde, kurumlar bilgilerini bilgisayar ortamında saklamaya başlamışlardır.

Veri miktarındaki artışın en önemli sebebi, elektronik veri toplayıcıları (uzaktan algılayan ya da satış noktası terminalleri) kullanımının yaygınlaşmasıdır. Öyle ki artık günlük alışverişler, alışverişler sırasında alınan ürün çeşitleri, işyerine giriş çıkış saatleri farkında olmadan yapılan rutin işlemler büyük veritabanlarına kaydedilen girdilerdir. Sadece uydu ve diğer uzay araçlarından dünyaya gönderilen görüntülerin saatte 50 gigabyte üzerinde olduğu düşünülürse, bu artışın boyutlarını daha açık bir şekilde göstermektedir. Bu kadar büyük veri yığınları arasından insan gözü mevcut verilerin ancak çok küçük bir kısmını görebilecek ve yorumlayabilecektir. Veri tabanı sistemlerinin kullanımının artışı ve hacimlerindeki muazzam büyüme ile birlikte bu verilerden nasıl faydalanabileceğimiz problemi ile karşılaşılmıştır. Mevcut veri sistemlerinde kullanılan sorgu veya raporlama araçlarının söz konusu veri yığınları karşısında yetersiz kalması veri tabanlarında bilgi keşfi adı altında yeni veri analiz yöntemleri arayışlarına neden olmaktadır [4].

VM ve bilgi keşfi, özellikle elektronik ticaret, bilim, tıp, iş ve eğitim alanlarındaki uygulamalarda yeni ve temel bir araştırma sahası olarak ortaya çıkmaya başlamıştır. VM; eldeki yapısız veriden anlamlı ve kullanışlı bilgiyi çıkarmaya yarayacak tümevarım işlemlerini formülle analiz etmeye, uygulamaya yönelik çalışmaları içermeye yarar. Geniş veri kümelerinden; desenleri, değişiklikleri, düzensizlikleri ve ilişkileri çıkarmakta kullanılır. Bu sayede, web üzerinde filtrelemeler, DNA sıraları içerisinde genlerin tespiti, ekonomideki eğilim ve düzensizliklerin tespiti, elektronik alışveriş yapan müşterilerin alışkanlıkları gibi karar verme mekanizmaları için önemli bulgular elde edilebilir [2].

VM; veri ambarlarındaki tutulan çok çeşitli verilere dayanan ve önceden keşfedilmemiş bilgileri ortaya çıkarmak, karar vermek ve eylem planını gerçekleştirmek için kullanma sürecidir. VM, veri içerisinden aranılan bilgiye ulaşma işidir. Madencilik teriminin kullanılma sebebi, büyük bir veri yığını arasından, uygun olanı arama ve seçme işleminin maden arama işine benzetilmesindendir. VM, büyük miktarda veri içinden, gelecekle ilgili tahmin yapılmasını sağlayacak bağıntı ve kuralların aranmasıdır. Bir başka deyişle, veri madenciliği büyük miktarda veri içinden gelecekle ilgili tahmin yapmayı sağlayacak, bağıntı ve kuralların bilgisayar programları kullanarak aranmasıdır. Veri kendi başına değersizdir. Önemli olan,

istenilen amaç doğrultusunda bilgilere ulaşabilinmesidir. Dolayısıyla bilgiye, bir amaca yönelik işlenmiş veridir, denilebilir.

Tezin ikinci bölümünde ses işleme, görüntü işleme ve VM gibi pek çok alanı içine alan örüntü tanıma kavramı ve örüntü tanımanın önemi kısaca belirtilmiştir. Üçüncü bölümde ise örüntü tanımanın alt süreci olan bilgisayar ortamına alınan verilerin yorumlanma süreci olarak bilinen VM ve veri tabanı kavramı ve veri tabanlarının tarihsel gelişimi, VM' nin kullanıldığı alanlar, VM' de karşılaşılan problemler, veri tabanlarında bilgi keşfi süreci konuları hakkında bilgi verilmiştir. Dördüncü bölümde bir veri tabanından elde edilen işlenmemiş verinin yorumlanma süreci olan sınıflandırma konusu işlenmiştir. Beşinci bölümde ise VM ile yapılmış çalışmalara yer verilmiş ve Elazığ ilinin meteorolojik verileri bir veri tabanında saklanarak bu verilerin değerlendirilmesi süreci örnek bir çalışma olarak ele alınmıştır. Verilen yöntemler içerisinde çeşitli şekiller kullanılarak kavramların daha iyi anlatılması hedeflenmiştir.

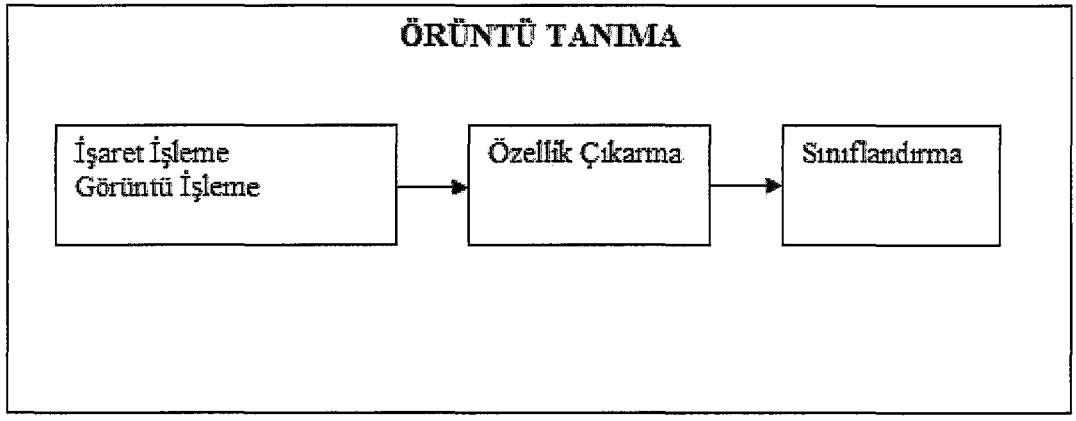
2. ÖRÜNTÜ TANIMA

Örüntü, varlıklar ile ilgili gözlenebilir veya ölçülebilir bilgilere verilen addır. Gerçek dünyadaki bu örüntüler, genellikle ilgilenilen verilerin nicel tanımlama şekilleridir. Örüntü tanıma, insanların çeşitli ses, görüntü, radar işareti ve benzeri tüm örüntülerin biçimsel şekillerinden çıkardıkları dilsel şekillendirmedir. Aslında, örüntü tanıma bilimin, mühendisliğin ve günlük hayatın geniş bir alanındaki etkinlikleri kapsamaktadır. Örüntü tanıma uygulamalarını insanların yaşantısında da görebiliriz: Hava değişimin algılanması, binlerce çiçek, bitki, hayvan türünü tanımlama, kitap okuma, yüz ve ses tanıma gibi bulanık sınırlara sahip birçok etkinlikte\ örüntü tanıma kullanılır. İnsanın örüntü tanıması, geçmiş tecrübelere dayalı öğrenme esaslıdır. Böylece, insanlar pratikte karşılaştığı örüntü tanıma olaylarını tecrübeleri ışığında değerlendirebilme yeteneğine sahiptirler. Belirli bir sesi tanımak için kullanılan kuralları tanımlamak mümkün değildir. İnsanlar bu işlemlerin birçoğunu oldukça iyi yapmalarına rağmen, bu işlemleri daha ucuz, iyi, hızlı ve otomatik olarak makinelerin yapmasını arzularlar. Örüntü tanıma, böyle akıllı ve öğrenebilen makineleri gerçekleştirmek için, çok boyutlu bir mühendislik disiplinidir [5].

Örüntü tanıma olayını şu şekilde irdeleyebiliriz: Aralarında ortak özellik bulunan ve aralarında bir ilişki kurulabilen karmaşık işaret örneklerini veya nesneleri bazı tespit edilmiş özellikler veya karakterler vasıtası ile tanımlama veya sınıflandırmadır. Bu bağlamda, örüntü tanımanın en önemli amaçları; bilinmeyen örüntü sınıflarına belirli bir şekil vermek ve bilinen bir sınıfa ait olan örüntüyü teşhis etmektir.

Örüntü tanıma tekniklerinin uygulamaları birçok mühendislik, tıp, askeri ve bilim alanına açıktır. Bunlardan bazıları; ses tanıma, EEG sınıflama, DTMF haberleşme işaretlerini tanıma ve radar hedef sınıflama, biyomedikal kontrol, veri madenciliği verilebilir. Örüntü tanıma olarak bilinen bu uygulamalar, makine öğrenmesi, örüntü sınıflandırma, ayrım analizi ve nitelik tahmini gibi isimlerle de anılmaktadır. Örüntü tanıma kavramı, Şekil 2.1. de gösterildiği gibi üç önemli birimden oluşmaktadır:

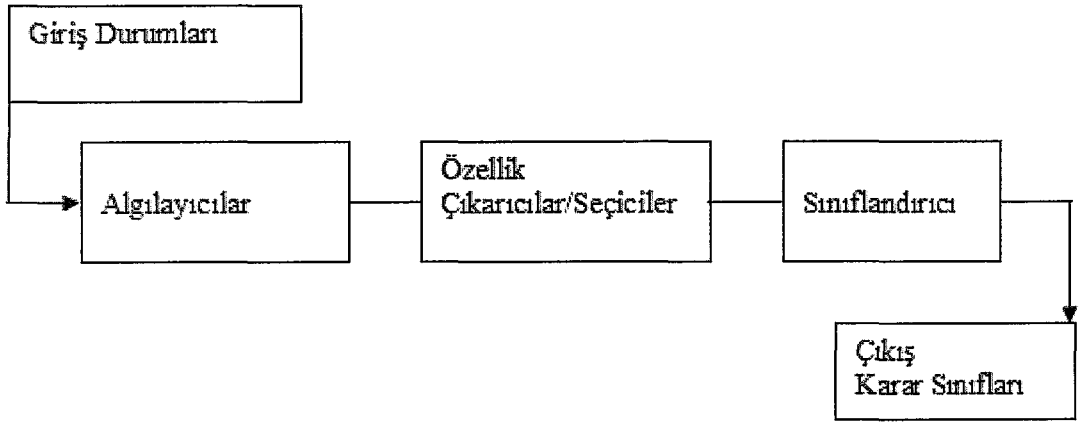
1. İşaret / Görüntü İşleme: Ön işlem aşamasıdır. İşaret veya görüntünün filtre edildiği, çeşitli dönüşüm ve gösterim teknikleri ile işlendiği, bileşenlerine ayrıldığı veya modellendiği kısımdır.
2. Özellik Çıkarma: İşaret ve görüntünün veri boyutunun indirgendiği ve tanımlayıcı anahtar özelliklerinin tespit edildiği ve aynı zamanda normalizasyona tabii tutulduğu aşamadır. Sistemin başarımında en etkili rolü oynar.
3. Sınıflandırma: Çıkarılan özellik kümesinin indirgendiği ve formüle edildiği tanımlayıcı karar aşamasıdır.



Şekil 2.1. Örüntü tanıma kavramı

2.1. Örüntü Tanıma Sistemleri

Örüntü tanıma sistemleri gözlenen veya ölçülen verileri tanımlanmasında birçok uygulamanın merkezinde yer alır. Şekil 2.2'de yaygın olarak kullanılan genel anlamda örüntü tanıma sistemi verilmiştir. Algılayıcılar, herhangi bir anda mümkün olan birçok doğal durumlardan biri olabilen bazı fiziksel işlemleri ölçerler. Aşağıdaki blok diyagramın en önemli görevlerinden biri de, elde edilen ölçümlerin hepsinden oluşan giriş uzayından daha az boyutta özellik çıkartmaktır. Sonunda, sınıflandırıcının rolü örüntüyü özelliklerine göre kategorize ederek uygun sınıflara kaydetmektir [5].

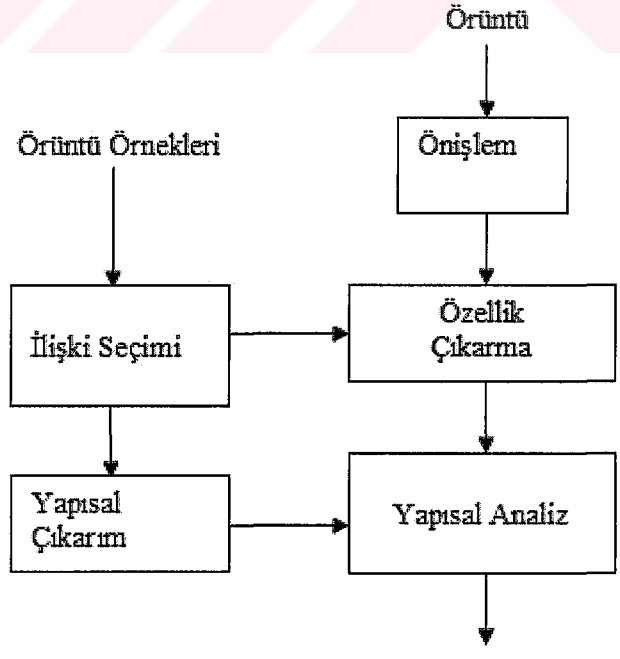


Şekil 2.2. Örüntü tanıma sistemi

Mevcut örüntü tanıma sistemleri üç grupta toplanmaktadır:

1. İstatistiksel örüntü tanıma: İstatistiksel örüntü tanıma yöntemin de sınıflama algoritmaları istatistiksel analiz üzerine kurulmuştur. Aynı sınıfa ait örüntüler, istatistiksel olarak tanımlanan benzer karakteristiklere sahiptirler. Bu yöntemde, özellik olarak nitelendirilen karakteristik ölçümler giriş örüntü örneklerinden çıkarılır. Her örüntü bir özellik vektörü ile tanımlanır. Genelde sınıflandırıcıyı oluşturan karar ve sınıflandırma yöntemleri üzerinde önemle durulur. Sınıflandırıcı tasarımı, ölçümler ve olasılıklar gibi işlenebilir örüntü bilgilerini birleştirmeyi esas alır. Böylece sınıflama, giriş veri uzayının olasılık yoğunluk fonksiyonlarının tahmini üzerine kurulu bir istatistiksel yapıdır. İstatistiksel örüntü tanıma\ Bayes Karar Teorisi üzerine kurulmuş olup uzun bir geçmişe sahiptirler [5].

2.Yapısal örüntü tanıma: Yapısal (geometriksel, kural dizilim) örüntü tanıma yaklaşımın da; verilen bir örüntü, şekilsel yapıdan temel karakteristik tanımlanmaya indirgenir. Çoğu zaman, örüntülerden çıkarılan bilgi yalnızca özellikler kümesinin sayısal değerlerinden değildir. Özelliklerin birbirine bağlanması veya aralarındaki karşılıklı ilişki, tanımlamayı ve sınıflandırmayı kolaylaştıran önemli yapısal bilgiye sahiptir. Bir başka deyişle örüntünün işlenmemiş halinden elde edilen tanımlayıcı biçimsel sentaks veya bunların sentezinden çıkarılan gramer ile tanımlama gerçekleşir (Örneğin, örüntünün köşe sayısı, kenar açıları vb.). Genel olarak yapısal yöntemde daha basit alt örüntüler karışık örüntülerin hiyerarşik tanımlamalarını formüle eder. Yapısal yöntemde her örüntü, bileşenlerinin bir kompozisyonu olarak ele alınır. Şekil 2.3'de bir yapısal örüntü tanıma sistemi görülmektedir [5].

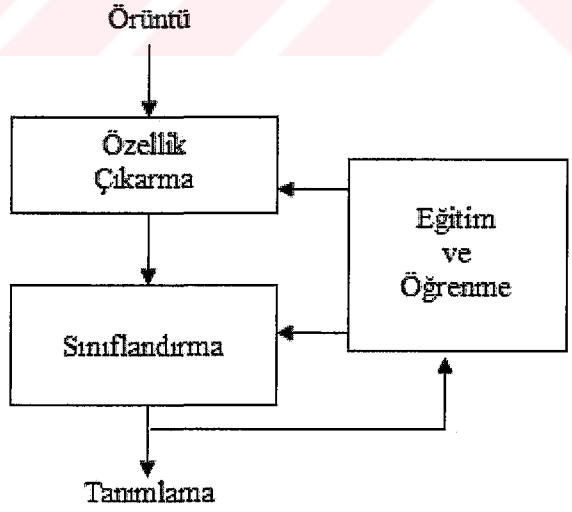


Şekil 2.3. Yapısal örüntü tanıma sistemi

Yapısal örüntü tanıma yönteminde çeşitli birimler arasındaki ilişki çok büyük önem taşır ve gerçek tanımada kullanılan bazı şekilsel notasyonlar tarafından belirtilir. Örneğin, ekrandaki bir masayı tanıma, "köşelerinden eşit uzunlukta bacaklar tarafından desteklenen yatay bir dikdörtgen yüzey" gibi yapısal tanımlamayı temel alarak gerçekleştirilebilir. Bu yöntemde, çevre uzunluğu, alan, ağırlık merkezi, eylemsizlik momenti ve Fourier tanımlayıcıları gibi genel özellikleri kullanır. Otoregresif model, poligonsal yaklaşım ve zincir kodları yapısal örüntü tanıma yöntemine örnek olarak verilebilir.

3. Akıllı örüntü tanıma: Örüntü tanıma sistemi, daha önceden öğrendiklerini tutabilecek bir hafızaya sahip, çıkarım, genelleme ve belirli bir hata toleransı ile karar verebilme yeteneklerini içermekte ise bu sistem akıllı örüntü tanıma sistemi olarak değerlendirilir. Şekil 2.4'de böyle akıllı ve öğrenebilen makineleri gerçekleştirmeye yönelik örüntü tanıma yaklaşımı verilmiştir.

Akıllı örüntü tanıma yaklaşımları, öğrenme tabanlı olup, karar aşamasında geçmiş tecrübelerinden sonuç üretmektedirler. Günümüzde, öğrenmeli örüntü tanıma algoritmaları yapay sinir ağ merkezli olarak gelişmektedir ve bu doğrultuda çalışmalar yoğunluktadır. YSA yaklaşımları istatistik yaklaşıma karşı belirleyici olarak ifade edilebilir. Çünkü öğrenme algoritmaları örüntü sınıflarının istatistiksel özellikleri hakkında hiçbir şey kullanmamaktadır. Bununla birlikte, istatistiksel ve YSA örüntü tanıma yaklaşımları şekil ve amaç olarak çok benzer olup, hatta YSA 'nın geleneksel istatistiksel örüntü tanımanın bir uzantısı olarak ifade edilen görüşlerde bulunmaktadır [5].



Şekil 2.4. Akıllı örüntü tanıma yaklaşımı

İstatistiksel, yapısal ve yapay sinir ağları ile örüntü tanıma yaklaşımları arasında kesin bir ayırım yoktur. Bunlar arasındaki sınırlar bulanıklık arz eder. Bu yaklaşımlar, genel ortak

özellikleri ve amaçları paylaşırlar. Verilen belirli bir örüntü tanıma probleminin çözümünde istatistiksel yaklaşıma göre örüntünün yapısı anlamsız olabilir. Yapı ancak uygun özellik seçimiyle yansıtılabilir. İstatistiksel örüntü tanımada; yapısal bilginin ifade edilmesinde görülen zorluk, yapısal örüntü tanımada kendini yapısal kuralların öğrenilmesinde gösterir. Buna karşın yapay sinir ağı yaklaşımı, istatistiksel ve yapısal yaklaşımlardan türetilmiştir. Açık bir şekilde örüntü hakkındaki yapısal bilgi değerli olduğunda, yapısal örüntü tanıma yaklaşımını seçmek daha doğrudur. Yapısal bilgi değersiz ve maksada uygun değilse, istatistiksel yöntemi seçmek daha doğrudur. Yapay sinir ağları, istatistiksel ve yapısal yaklaşıma alternatif teknikler sağlayan ve örüntü tanımaya öğrenme boyutu katarak akıllı tanıma niteliği kazandıran bir teknik olarak düşünülebilir [5].

2.2. Örüntü Tanımada Öğrenme

İstatistiksel örüntü tanımada öğrenme, genellikle veriden bilinmeyen bir olasılık yoğunluğunu tahminden meydana gelir. Parametre tahmin etme de model varsaymak veya bilinmeyen yoğunluğun parametrik bir şeklidir. Öğrenme süreci eğitme verisini en iyi uygunlaştıran modelin parametrelerini optimize etmeyi içerir. Maksimum olabilirlik ve Bayes tahmin edici parametrik karar vericilere örnektir. Parametrik olmayan teknikler ve tahmin ile veya verinin dağılım şekli hakkında tahmin yapmaksızın sınıflandırmaya devam edilir. Parzen pencereleri ve en yakın komşu kuralı ardışık öğrenmeye gerek duymayan parametrik olmayan yöntemlerdir. Doğrusal fark fonksiyonları öğrenmenin meydana geldiği parametrik olmayan bir tekniği dikkate alabilir [6].

Kümeleme eğiticisiz öğrenme gerektiren örüntü tanımada bir problemdir. Kümelemenin gayesi kategorilerin genelde bilinmeyen sayısı içinde sık sık tanımlanmayan veriyi, verinin kendisine bakarak gruplamaktır. Minimum karşıtlık, c-mean ve single-link geleneksel kümeleme algoritmalarına örnektir.

Öğrenmeli örüntü tanıma sistemlerinin başlıca iki amacı vardır:

1. Sınıflandırılacak verinin bazı yapı ve organizasyonunu birleştirmek.
2. Yeni veri için önceden bilinen sonuçlardan eğitilmiş sınıflandırıcıyı kullanmak.

Burada, öğrenme olayı sınıflandırıcıyı (tahmin ediciyi) ilgilendirir. Akıllı örüntü tanıma sistemlerinde, sınıflandırıcının parametreleri bir karar kuralına göre ardışık işlemlerle uyarlanarak öğrenme işlemi gerçekleştirilir. Sınıflandırıcı *"en iyinin ölçüsü"* ile değerlendirilir. Sınıflandırıcının *"en iyinin ölçüsü"* ne uyarlanma işlemine, kabul edilebilir sınırlar içinde devam edilebilir. Daha geniş anlamda, sınıflandırıcının öğrenmesi bir çıkış kümesi ile bir giriş kümesinden meydana gelir. Veri kümeleri eğitim verisi ve test etme verisi olarak iki parçaya

ayrılır; Öncelikle öğrenme işlemi gerçekleştirilir sonra sınıflandırıcının bir yetenek göstergesi olarak yeni bilgi ele alınarak test etme sürecine geçilir.

Problemin matematiksel formülasyonu için bazı değişkenleri tanımlamak gereklidir:

$\phi \in \Re^n$	Modelin n değişkenli parametreleri veya karakteristikleri
$J\left(\phi, \psi, \Omega_\psi\right) \in M$	"en iyinin ölçüsü", amaç fonksiyonu veya fonksiyon ölçütü,
M	Fonksiyonların uzayı $M: \Re^n x \Re^{m x N} x \Re^{s x N} \rightarrow \Re$
$\psi \in \Re^{m x N}$	Eğitme verisindeki bağımsız değişken, $\psi = \left\{\psi_i \in \Re^m, i = 1....N\right\}$
$\Omega_\psi \in \Re^{s x N}$	Eğitme verisindeki bağımlı değişken, $\Omega_\psi = \left\{\omega^\psi{}_i \in \Re^s, i = 1....N\right\}$
N	Eğitme verisindeki örneklerin sayısı
$g(\phi)$	*m* tane yan şart $g(\phi) = \left\{g_i(\phi) \middle\| \Re^n \rightarrow \Re, i = 1...m\right\}$
$\xi \in \Re^{m x N}$	Test verisindeki bağımsız değişken $\xi = \left\{\xi_i \in \Re^m, i = 1...N\right\}$
$\Omega_\xi \in \Re^{s x N}$	Test verisindeki bağımlı değişken $\Omega_\xi = \left\{\omega_{\xi_i} \in \Re^s, i = 1...N\right\}$
$f\left(\phi, v \in \Re^m\right)$	Sınıflandırıcı eşlemesi $f(.): \Re^n x \Re^m \rightarrow \Re^s$
$E\left[\Omega_\xi, f\right]$	Test verisinin sınıflandırılmasındaki hata $E[.]: \Re^s x \Re^s \rightarrow \Re$

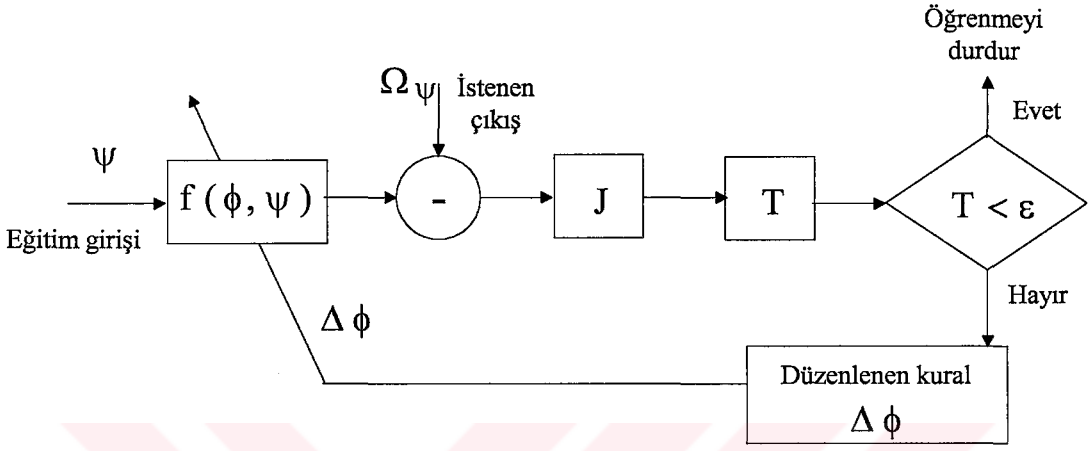
Bir giriş/çıkış eşlemesiyle sınıflandırıcının yorumunda tutarlılık olacak şekilde, genelde bağımsız değişkenler giriş ve bağımlı kopyası ise çıkış olarak adlandırılır. Burada tek gereksinim $f\left(\phi, v \in \Re^m\right)$ ile sınıflandırılan, ya bire bir veya birçoğundan birine eşlemektir. Bu nedenle, $f\left(\phi, v \in \Re^m\right)$ den yaklaştırma fonksiyonu olarak da söz edilir.

Öğrenme süreci, *J* amaç fonksiyonuna uyan ψ girişlerinden Ω_ψ çıkışlarını elde edebilen, $f\left(\phi, v \in \Re^m\right)$ en iyi yaklaşım fonksiyonunu elde etmek için her hangi bir $g(\varphi)$ yan şartını sağlayan bir ϕ parametre kümesi bulma çalışmalarını ihtiva eder. Yani:

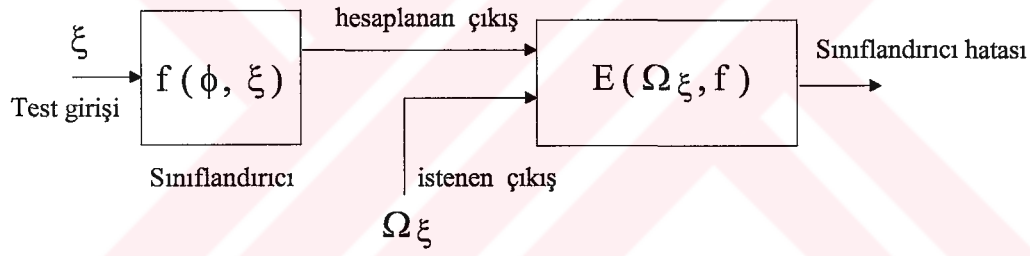
$g(\phi) \geq 0$ koşulu ile $J\left(\phi, \psi, \Omega_\psi\right)$ fonksiyonunu minimize eden ϕ 'yi bul.

Bu serbest parametrelerin istenilen şekilde ardışık olarak ayarlayıp belirlemek için bir matematiksel optimizasyon algoritması seçmek zorunludur. Ayrıca istenilen $T\left[\varphi, J\left(\varphi, \psi, \Omega_\psi\right)\right] < \varepsilon$ biçimi bir veya daha çok bitirme şartıdır. Burada, *T* birbirini izleyen ardışık değerlendirmelerin tipiksel olarak bir norm ölçüsüdür ve ε ise tavsiye edilen toleranstır.

Eğitme veya öğrenme bittikten sonra sınıflandırıcı bir veri kümesi (ξ, Ω_ξ) üzerinden test edilir. Seçilen hata ölçüsü *E[.]* göre, yeni ξ verisi için önceden hesaplanan sonucun *f(.)* ile sınıflandırılması iyi bir başarım göstergesidir. *E[.]* anlamlı bir uzaklık ölçüsü olabilir veya sınıflandırma senaryosunda karar kuralı ve hata oranını yaklaştırmak için bir sayma işlemi de olabilir. Süreçlerin tanımlanması Şekil 2.5. ve 2.6. 'da özetlenmiştir [6].



Şekil 2.5. Öğrenme süreci



Şekil 2.6. Test etme süreci

2.2.1. Eğiticili ve Eğiticisiz Öğrenme

Yukarıda varsayılan formülasyonda veri kümeleri o şekilde tanımlanır ki her bir ψ için Ω_ψ uygunluğu olur. Bu koşullar altındaki öğrenme eğiticili öğrenme olarak adlandırılır. Eğiticisiz öğrenme olarak bilinen durumda ise Ω_ψ giriş den bağımsız olarak tanımlanır ve öğrenme süreci daha zordur [3].

Eğiticili öğrenmede dışarıdan bir eğiticinin müdahalesi söz konusudur. Eğitici, sisteme ilgili girdi için üretmesi gereken sonucu verir. Yani sınıflandırıcı sisteme girdi/çıktı ikilisinden oluşan örnekler sunulur. Bu ikili, sistemin öğrenmesi gereken özellikleri temsil eder. Eğiticisiz öğrenmede ise hiç bir eğiticiye ihtiyaç yoktur. Bu nedenle çoğu zaman buna kendi kendine organize olma da denilmektedir. Tahmin edici, kendine gösterilen örnekleri alır ve belli bir

kritere göre sınıflandırır. Bu kriter önceden bilinmeyebilir. Sistem, kendi öğrenme kriterlerini kendisi oluşturmaktadır.

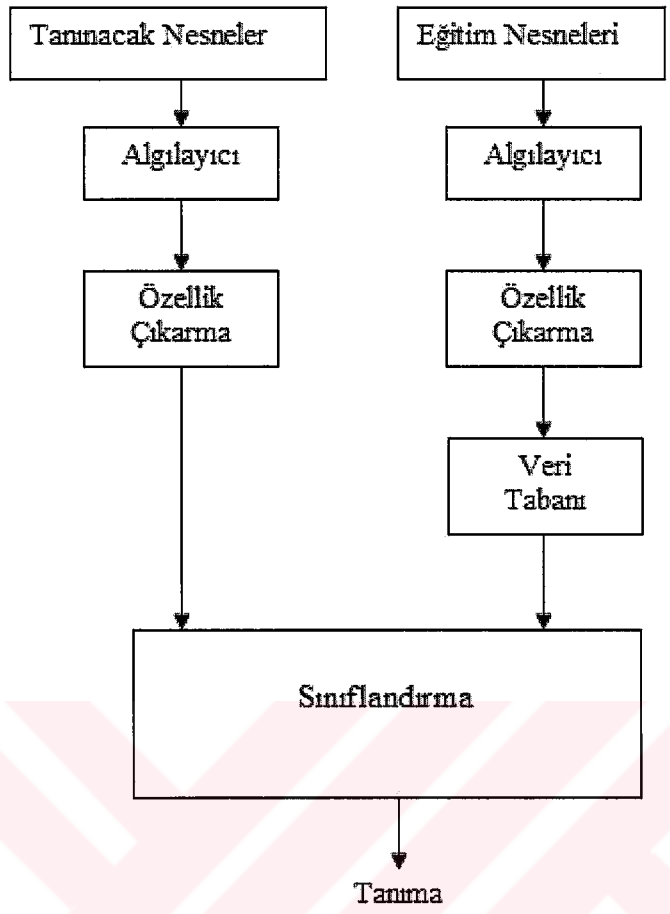
Eğiticili ve Eğiticisiz öğrenme türleri bazen birbirini destekleyici olarak kullanılırlar. Eğiticisiz tür ile özellik uzayı kümelenir. Bu kümeler eğiticili tür için bilinen yapı olarak eğitim kümesinde kullanılabilir. Böylece özellik uzayının boyutu azaltılır. Fakat bu işlemler zaman aldığından gerçek zaman uygulamaları zor ve pahalıdır.

2.2.2. Veri Uygunluğu ve Genelleme

Öğrenmenin tanımlanan iki maksadı; veri uygunlaştırma ve genellemedir. Eğer tahmin edici eğitim verisini tam olarak açıklamak zorundaysa, tahmin edicinin açıklaması sınırlı enterpolasyon (iç değerlendirme) yeteneği sebebiyle, test verisinde kötü bir performans icra edecektir. Bu eğitme veya uygunlaştırma üzerinde genel kavramdır. Eğitme, veri uygunlaştırma üzerinde daha az sıkı isteklerde bulunmakla birlikte, test verisini genelleştirmede önemli bir esneklikle tahmin edici karar verecektir [6].

2.3. Örüntü Tanıma Sistemlerinin Bileşenleri

Tipik bir örüntü tanıma sistemi, Şekil 2.7.de görülmektedir. Sistem, eğitim ve tanıma olmak üzere iki evreden oluşur. Sistemin en önemli elemanları; özellik çıkarma (niteleme), veri tabanı oluşturma ve sınıflandırma (eşleme) bileşenleridir. Eğitim ve tanıma evrelerindeki algılayıcılar ve özellik çıkarma elemanları değişik olabilir [3].



Şekil 2.7. Örüntü tanıma sistemi

2.3.1. Özellik Çıkarma

Özellik çıkarma örüntü tanımanın en önemli kısmı olup bir anlamda örüntü tanıma sisteminin başarımında anahtar rolü oynar. Örüntü sınıfları arasında ayrımı gerçekleştirmek için örüntü özelliklerinin çıkarılması gerekir. Günümüzde, çok başarılı sonuçlar veren örüntü sınıflandırıcı türleri mevcut olup sınıflandırıcının da doğrudan başarımını etkileyen özellik çıkarımı üzerine çalışmalar odaklanmıştır [5]. Özellik çıkarmanın ana sebepleri:

1. Ölçüm veya örüntü uzayından daha küçük boyuta dönüşmeyi sağlamaktır. Bu sınıflandırıcının küçük hatalar ile eğitimi ve karar aşamasının daha kısa sürede gerçekleşmesi demektir.
2. Boyut olarak daha düşük olan özellik uzayını sınıflandırıcının daha az parametre ile öğrenmesini mümkün hale getirecektir. Bunun yararı örüntü uzayı ile karar uzayı arasındaki dönüşüm aşamasının daha kısa sürede gerçekleşmesidir.

3. Durağan olmayan zaman serilerinde olduğu gibi karmaşık örüntülerin tanımlayıcı karakteristiklerini bulabilmek için özellik çıkarımı şarttır. Böylece karar aşamasının güvenirliliği artacaktır.

4. Örüntü sınıflandırma sisteminin, sistem içi veya dışındaki kontrolsüz girişimlerden etkilenmemesini sağlayacak bir özellik çıkarımı kararlı bir yapının oluşmasında etken olacaktır. Bu tür kararlı özellikler, sınıflandırıcının genelleme ve ayrışım yeteneğinin yüksek olmasında önemlidirler.

Örüntü özelliklerini belirlemede ana problem verilen esas örüntüden en iyi özellikleri seçmektir. Bunun için doğrudan ve dolaylı olmak üzere iki yaklaşım vardır. Birinci metot güçlü yapısal bağlantılara sahip olan ve basit yapılı belirli örüntü tanıma problemlerine uygulanabilmektedir. Dolaylı metotlarda ise aşağıdaki gibi formüle edilebilen bir dönüşüm veya gösterim tekniği ile daha kullanışlı bir yapıdan özellik çıkarımı yapılmaktadır. Böylece z uzayından, x uzayına bir dönüşüm gerçekleştirilir.

$$x = F(z)$$

- İşaretler örüntülerinin özelliklerinin çıkarımı ile ilgilenildiğinde, özellik çıkarımı için genelde zaman ve frekans bölgesi gösterimi ön plandadır. Böylece karmaşık örüntü yapısının hem geçici ve hem de zamana bağlı olarak frekans değişimlerini içeren tanımlayıcı özellik bilgileri çıkarılabilir. Bu özellikler, işaretin zaman ve frekans bölgelerindeki yerel bilgilerini karakterize eder.
- Nesne örüntülerinin özellik çıkarımı ile ilgilenildiğinde, özellikler kullanılan veritabanı ve uygulama alanına göre farklılık gösterir. Temel özellikler; kenar, köşe, doğru ve eğri çizgiler, delik ve sınır eğriselliğidir. Nesne örüntüsü tanımlamaları bu özelliklerin birinden veya birkaçının birleştirilmesinden elde edilir.

  Endüstriyel uygulamalarda çoğu zaman nesne örüntüsü sınırları ve bu sınırlardan türetilmiş ölçümler, özellik olarak kullanılır. Bu özellikler genel, yerel ve ilişkisel olmak üzere üç guruba ayrılabilir. Genel özelliklere örnek olarak çevre, ağırlık merkezi, sınır noktalarının ağırlık merkezine uzaklığı, eğrisellik, alan, eylemsizlik momenti gibi özellikler verilebilir. Doğru parçaları, sabit eğriselliği olan çember parçaları yerel özellikler için örnek verilebilir. İlişkisel özelliklere örnek olarak, nesne örüntüsünün alt parçalarının birbirlerine göre uzaklıkları, açıları gibi parametreler verilebilir [5].

2.3.2. Sınıflandırma

Bir sınıflandırıcıdan geçirilen verilerin belirli gruplara ayrılıp yorumlanması sürecidir. Sınıflandırma ile daha detaylı bilgi sınıflandırıcılar bölümünde verilecektir.

2.4. Örüntü Tanıma ve Veri Madenciliği

Örüntü tanıma, görüntü tanıma, işaret işleme ve veri madenciliği gibi pek çok konuyu içine alan geniş içerikli bir kavramdır. Veri madenciliği veri tabanlarında tutulan anlamsız veriler bütününden bir anlamlı veri elde etme sürecidir. Örüntü tanıma ise daha veriler veri tabanına gelmeden başlayan bir süreçtir. Mesela bir radar sinyali bilgisayara alınıp değerlendirilecekse örüntü tanıma, o işaretin işaret işleme tekniği kullanılarak yorumlanıp bilgisayar ortamına uygun hale getirilmesinden, veri tabanına depolanıp sınıflandırıcılarda değerlendirilme süreçlerinin bütününü kapsar. Veri madenciliği ise radar işareti ve bu işaretin bilgisayar ortamına alınması ile ilgilenmez. Veri madenciliği bilgisayar ortamına alınmış veri tabanlarında depolanan verilerin sınıflandırıcılarda değerlendirilme sürecini içerir. Bu yüzden veri madenciliğinin örüntü tanımanın bir kolu ya da bir bölümü olduğu söylenebilir [4].

Tezin bu bölümünde veri madenciliği konusunu kapsayan örüntü tanıma konusu hakkında bilgi verilmiştir. Tezin 3. bölümünde ise örüntü tanıma konusunun bir alt dalı olan veri madenciliği konusunda detaylı bilgi verilmiştir.

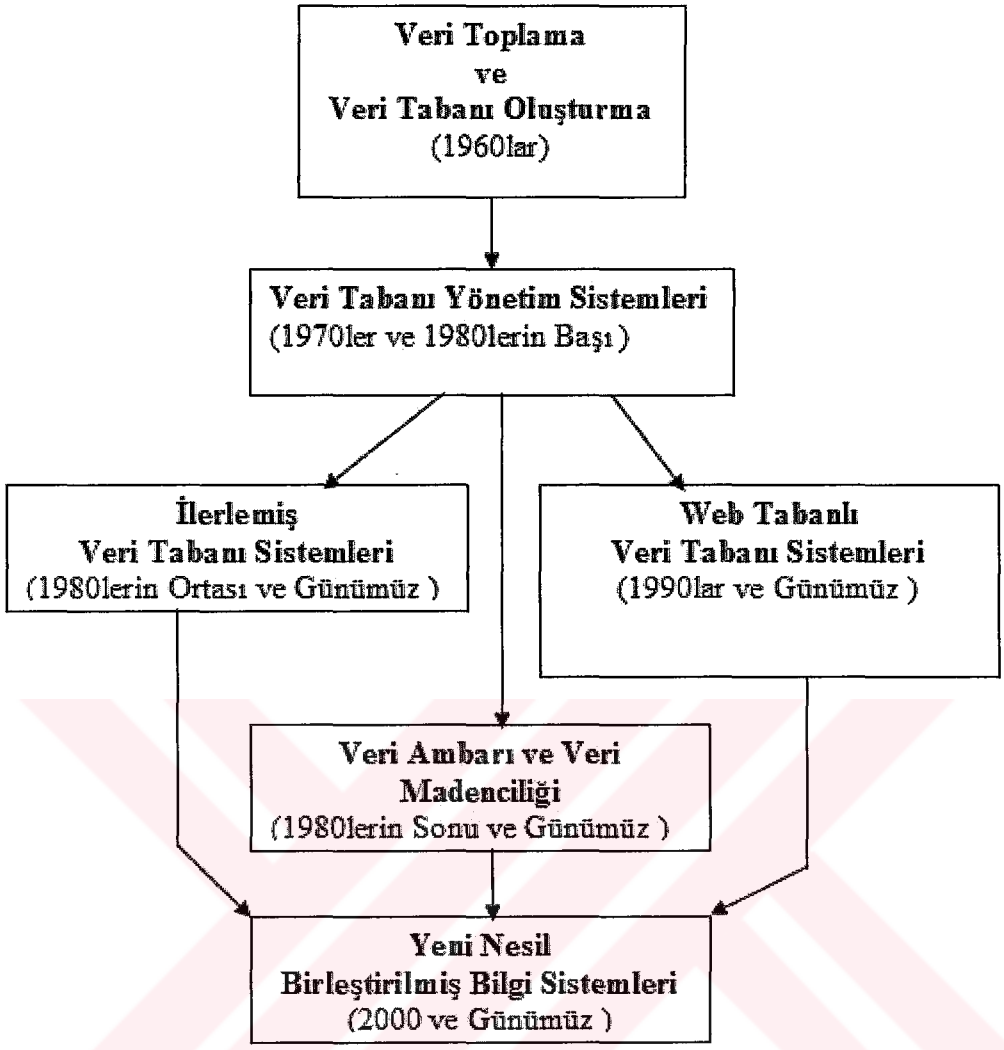
3. VERİ MADENCİLİĞİ

3.1. Veri Tabanı Kavramı ve Veri Tabanlarının Tarihsel Gelişimi

Veri madenciliği kavramını iyi anlayabilmek için veri ve veri tabanı kavramlarını bilmek gerekir. Veri işlenmemiş bilgi manasındadır. Veri tabanı ise bilgilerin depolanması amacıyla oluşturulmuş yapılardır.

Veri madenciliğini cazip kılan temel neden, büyük veri yığınları arasından kullanışlı olan bilgilerin ortaya çıkarılmasından kaynaklanmaktadır. Veri madenciliği bilgi teknolojisinin doğal bir evriminin sonucu olarak görülebilir. Evrimsel yol Şekil 3.1.'de görüldüğü gibidir.

1960'lar ve 1960'ların başlarında ilkel dosyalama işlemleri yapılmış veriler toplanmış ve veri tabanları oluşturulmuş, 1970'ler ve 1980'lerin başında hiyerarşik ve ağ yönetim sistemleri, ilişkisel veri tabanı sistemleri, veri tabanı yönetim sistemleri oluşturulmuş ve verileri sorgulamak için sorgulama dilleri geliştirilmiştir. 1980'lerin ortasından günümüze kadar olan süreçte çoklu ortam, bilimsel, aktif ve bilgi tabanına dayalı ilerlemiş veri tabanı sistemleri oluşturulmuştur. Veri tabanlarının artması ve ilerlemesiyle birlikte bilgiye olan ihtiyaç artmış ve veri tabanlarından daha fazla yararlanılması amacıyla 1980'lerin sonundan günümüze kadar olan süreçte verilerin depolandığı veri ambarı, veri analiz etmekte kullanılan teknikleri içeren OLAP, veri madenciliği ve veri tabanlarında bilgi keşfi kavramları ortaya çıkmıştır. 1990'lardan günümüze kadar olan süreçte ise internet ortamının kullanılmaya başlanılmasıyla birlikte XML tabanlı veri tabanı sistemleri ve web madenciliği kavramları ortaya çıkmıştır. 2000 yılından günümüze kadar olan süreç ise Şekil 3.1.'de görülen bütün sistemleri kapsar ve yeni nesil birleştirilmiş bilgi sistemleri olarak adlandırılır [7].



Şekil 3.1. Veri tabanı teknolojisinin evrimi

3.2. Veri Madenciliğinin Tanımı

Araştırma alanlarından biri olan VTBK (veri tabanı bilgi keşfi) disiplini, çok büyük verileri tam ya da yarı otomatik bir biçimde analiz eden yeni araç ve tekniklerin üretilmesi ile ilgilenen son yılların gözde araştırma konularından biridir. VTBK, veri seçimi, veri temizleme ve ön işleme, veri indirgeme, VM (veri madenciliği) ve değerlendirme aşamalarından oluşan bir süreçtir. VM, önceden bilinmeyen, veri içinde gizli, anlamlı ve faydalı bilgilerin büyük ölçekli veritabanlarından otomatik biçimde elde edilmesini sağlayan VTBK süreci içinde bir adımdır[8].

VM, eldeki verilerden üstü kapalı olan, çok net olmayan, önceden bilinmeyen kullanışlı bilginin çıkarılmasıdır. Bu da; kümeleme, veri özetleme, değişikliklerin analizi, sapmaların

tespiti gibi belirli sayıda teknik yaklaşımları içerir. Başka bir deyişle VM, verilerin içerisindeki desenlerin, ilişkilerin, değişimlerin, düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların yarı otomatik olarak keşfedilmesidir.

Temel olarak VM, geniş veri setleri arasındaki desenlerin ya da düzenin, verinin analizi ve yazılım tekniklerinin kullanılması ile ilgilidir. Veriler arasındaki ilişkiyi, kuralları ve özellikleri belirlemekten bilgisayar sorumludur. Amaç, daha önceden fark edilmemiş veri desenlerini tespit edebilmektir.

Aslında VM' yi uzman bir sistem gibide düşünebiliriz. Yalnız bir yönden uzman sistemlerden farklılık gösterir. Uzman bir sistem hazırlanırken konuya hakim bir uzman ve uygun bir çoklu ortam gerekir oysa her zaman bir uzman bulmak mümkün olamayabilir. Bu durumda öğrenme makine öğrenmesi ya da bilgisayarın kendi kendine öğrenmesi dediğimiz akıllı sınıflandırıcılar sayesinde gerçekleştirilir [9].

VM' yi istatistiksel bir yöntemler serisi olarak görmek mümkün olabilir. Ancak VM, geleneksel istatistikten birkaç yönde farklılık gösterir. VM' de amaç, kolaylıkla mantıksal kurallara ya da görsel sunumlara çevrilebilecek nitel modellerin çıkarılmasıdır. Bu bağlamda, VM insan merkezlidir ve bazen insan ve bilgisayar ara yüzü birleştirilir.

VM konusunda bahsi geçen geniş verideki "geniş" kelimesi, tek bir iş istasyonunun belleğine sığamayacak kadar büyük veri kümelerini ifade etmektedir. Yüksek hacimli veri ise tek bir iş istasyonundaki ya da bir grup iş istasyonundaki disklere sığamayacak kadar fazla veri anlamındadır. Dağıtık veri ise farklı coğrafi konumlarda bulunan verileri anlatır.

Büyük ya da küçük firmalar müşterilerin ilgilerini saptayabilmek ve müşterileri çekebilmek için öğrenme ilişkilerine ihtiyaç duyarlar bu ilişkiler de veri madenciliği ile ortaya çıkarılır. Veri madenciliği anlamlı örnek ve kuralları keşfetmek için büyük veri miktarları içinden otomatik veya yarı otomatik anlamları analiz etmekte ve incelemekte kullanılır. Veri madenciliği, şirketlerin müşterilerini daha iyi anlamaları için pazarlamalarına, satışlarına ve müşteri destek çalışmalarına izin verir [10,11].

VM, makine öğrenimi, istatistik, veritabanı yönetim sistemleri, veri ambarı, paralel programlama gibi farklı disiplinlerde kullanılan yaklaşımları birleştirmektedir. VA (veri ambarı) sorgulama, analiz etme ve raporlama için belirgin bir şekilde yapılandırılmış iş verisinin bir kopyasıdır. VA iş verilerinin kopyalarını içeren büyük bir veri tabanıdır. VA kavramı standartlaştırılan şartlar ve tanımlamaların akış durumu içindeki yeni bir ifadesidir. Bazı tanımlamalar veri üzerinde odaklanırken, diğerleri insanlara yazılımlar, araçlar ve iş süreçlerinden bahseder. VA' ya sahip olan kuruluşlarda gerekli veriler VP (veri pazarı) olarak isimlendirilen işleve özel veri tabanlarına aktarılır. VS (veri stoku) VA' nın en yaygın

bileşenidir. Verinin bir teki için fonksiyon günden güne depolanır bununla birlikte işlenmemiş veri ile VA' yı besler [11,12].

Bilgisayarın otomasyondaki önemini anlatmaya gerek yoktur. Her alanda olduğu gibi bilgisayar otomasyonda da vazgeçilmez araç konumundadır [13]. Makine öğrenimi, istatistik ve VM arasındaki yakın bağ kolaylıkla görülebilir. Bu üç disiplin veri içindeki ilginç düzenlilikleri ve örüntüleri bulmayı amaçlar. Makine öğrenimi yöntemleri VM algoritmalarında kullanılan yöntemlerin çekirdeğini oluşturur. Makine öğreniminde kullanılan karar ağacı, kural tümevarımı pek çok VM algoritmasında kullanılmaktadır. Makine öğrenimi ile VM arasında benzerliklerin yanı sıra farklılıklar da göze çarpmaktadır. Öncelikle VM algoritmalarında kullanılan örneklem boyutu, makine öğreniminde kullanılan veri boyutuna nazaran çok büyüktür. Genellikle makine öğreniminde kullanılan örneklem boyu 100 ile 1000 arasında değişirken, VM algoritmaları milyonlarca gerçek hayat nesneleri üzerinde uğraşmaktadır ki bunların karakteristiği boş, artık, eksik, gürültülü değerler olarak belirlenebilir. Aynı zamanda VM algoritmaları bilgi keşfetmeye uygun nesne niteliklerinin elde edilme sürecindeki karmaşıklıkla baş etmek zorundadır.

Olasılıksal veri nedenlemede VM, istatistik alanındaki birçok yöntemi kullanmasına rağmen, nesnelerin nitelik değerlerine bağlı çıkarsama yapmada bilinen istatistiksel yöntemlerden ayrılmaktadır. İstatistiksel yöntemler karar verme mekanizmasında VM disiplini ortaya çıkmadan önce çok sık kullanılırdı. Ancak bu yöntemlerin kullanım zorluğu (uzman kişileri tutma/başvurma), VM algoritmalarının uygulama kolaylılığı ile karşılaştırıldığında, veri nedenleme sürecindeki en güç adımı oluşturuyordu.

VTYS (veri tabanı yönetim sistemleri) büyük miktardaki yapısal bilgiyi saklama ve etkin bir biçimde erişim sağlamakla yükümlüdür. VTYS' lerde veri düzenlemesi, ilgili organizasyonun işletimsel veri ihtiyacı doğrultusunda gerçekleştirilir ki bu her zaman bilgi keşfi perspektifi ile bire bir çakışmaz. Bu açıdan veritabanındaki veriler temizleme, boyut indirgeme, transfer, vb. işlemlerinden geçirilerek VM kullanımına sunulurlar. VM teknikleri ayrı araç olarak sağlanabileceği gibi bir VTYS ile de entegre olabilirler [14].

VM algoritmalarında girdi olarak kullanılan verinin çok büyük olması ve işletim süresinin kabul edilebilir bir sınırda yer alması, VM algoritmalarının paralel programlama kullanılarak gerçekleştirimini beraberinde getirmiştir [15].

Etkin bir VM uygulayabilmek için dikkat edilmesi gereken noktalar aşağıdaki gibi özetlenebilir [16,17].

· *Değişik türdeki verileri ele alma:* Gerçek hayattaki uygulamalar makine öğreniminde olduğu gibi yalnızca sembolik veya kategorik veri türleri değil, fakat aynı zamanda tamsayı, kesirli sayılar, çoklu ortam verisi, coğrafi bilgi içeren veri gibi farklı tipteki veriler üzerinde işlem yapılmasını gerektirir. Kullanılan verinin saklandığı ortam, düz bir kütük veya ilişkisel

veritabanında yer alan tablolar olacağı gibi, nesneye yönelik veritabanları, çoklu ortam veritabanları, coğrafi veritabanları vb. olabilir. Saklandığı ortama göre veri, basit tipte veya karmaşık veri tipinde (çoklu ortam verisi, zaman içeren veri, yardımlı metin, coğrafi, vb.) olabilir. Bununla birlikte veri tipi çeşitliliğinin fazla olması, bir VM algoritmasının tüm veri tiplerini ele alabilmesini olanaksızlaştırmaktadır. Bu yüzden veri tipine özgü adanmış VM algoritmaları geliştirilmektedir.

· *VM algoritmasının etkinliği ve ölçeklenebilirliği:* Çok büyük veri içinden bilgi elde etmek için kullanılan VM algoritması etkin ve ölçeklenebilir olmalıdır. Bu, VM algoritmasının çalışma zamanının tahmin edilebilir ve kabul edilebilir bir süre olmasını gerektirir. Üssel veya çok terimli bir karmaşıklığına sahip bir VM algoritmasının uygulanması, kullanışlı değildir.

· *Sonuçların yararlılık, kesinlik ve anlamlılık kıstaslarını sağlaması:* Elde edilen sonuçlar analiz için kullanılan veritabanını doğru biçimde yansıtmalıdır. Bunun yanı sıra gürültülü ve aykırı veriler ele alınmalıdır. Bu işlem elde edilen kuralların kalitesini belirlemede önemli bir rol oynar.

· *Keşfedilen kuralların çeşitli biçimlerde gösterimi:* Bu özellik keşfedilen bilginin gösterim biçiminin seçilebilmesini sağlayan yüksek düzeyli bir dil tanımının yapılmasını ve grafik ara yüzünü gerektirir.

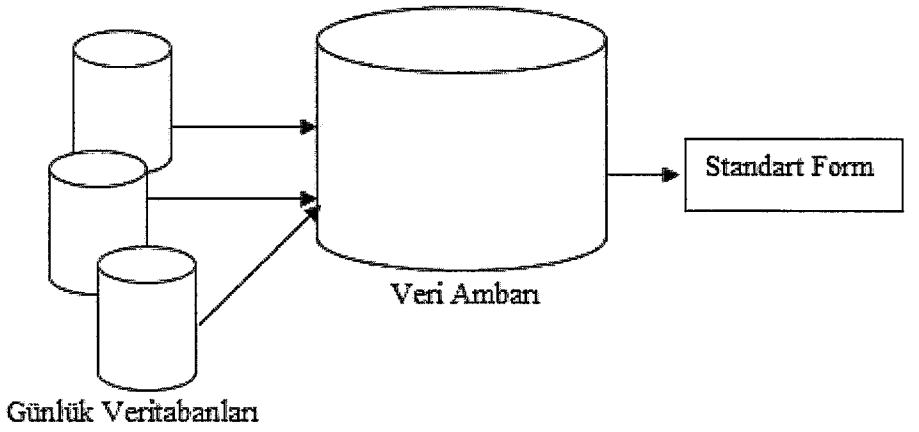
· *Farklı birkaç soyutlama düzeyi ve etkileşimli VM:* Büyük veritabanlarından elde edilecek bilginin tahmin edilmesi güçtür. Bu yüzden VM sorgusu, elde edilen bilgilere göre kullanıcıya etkileşimli olarak sorgusunu değiştirebilmeyi, farklı açılardan ve farklı soyutlama düzeylerinden keşfedilen bilgiyi inceleyebilme esnekliğini sağlamalıdır.

· *Farklı ortamlarda yer alan veri üzerinde işlem yapabilme:* Kurumlar yerel ağlar üzerinden pek çok dağıtık ve heterojen veritabanı üzerinde işlem yapmaktadır. Bu VM' nin farklı kaynaklarda birikmiş formatlı ya da formatsız veriler üzerinde analiz yapabilmesini gerektirir. Verinin büyüklüğünün yanı sıra dağıtık olması, yeni araştırma alanlarının ortaya çıkmasına sebep olmuştur. Bunlar, paralel ve dağıtık VM algoritmalarıdır.

· *Gizlilik ve veri güvenliğinin sağlanması: Bir* VTBK sisteminde keşfedilen bilgi pek çok farklı açıdan ve soyutlama düzeyinden izlenebildiği için gizlilik ve veri güvenliği, VM sistemini kullanan kullanıcının haklarına ve erişim yetkilerine göre sağlanmalıdır.

VM büyük miktarda veri inceleme amacı üzerine kurulmuş olduğundan, veri tabanları ile yakından ilişkilidir. Verinin hızla ulaşılabilecek şekilde amaca uygun bir şekilde saklanması gerekir. Günümüzde yaygın olarak kullanılmaya başlanan veri ambarları günlük kullanılan veri tabanlarının birleştirilmiş ve işlemeye daha uygun bir özetini saklamayı amaçlar. Şekil 3.2' de görüldüğü gibi günlük veri tabanlarından istenen özet bilgi seçilerek ve gerekli ön işlemeden

sonra veri ambarında saklanır. Ardından amaç doğrultusunda gerekli veri ambardan alınarak, VM çalışması için standart bir forma çevrilir [18].



Şekil 3.2. Veritabanı - veri ambarı - standart form

VA' da veri oluşturulduktan sonra bu verinin elle veya gözle analizi yapılabilir. Bunun için OLAP programları kullanılır. Bu programlar, veriye her boyutu veride bir alana karşılık gelen çok boyutlu bir küp olarak bakmayı ve incelemeyi sağlar. Böylece boyut bazında gruplama, boyutlar arasındaki korelasyonları inceleme ve sonuçları grafik veya rapor olarak sunma olanağı sağlar.

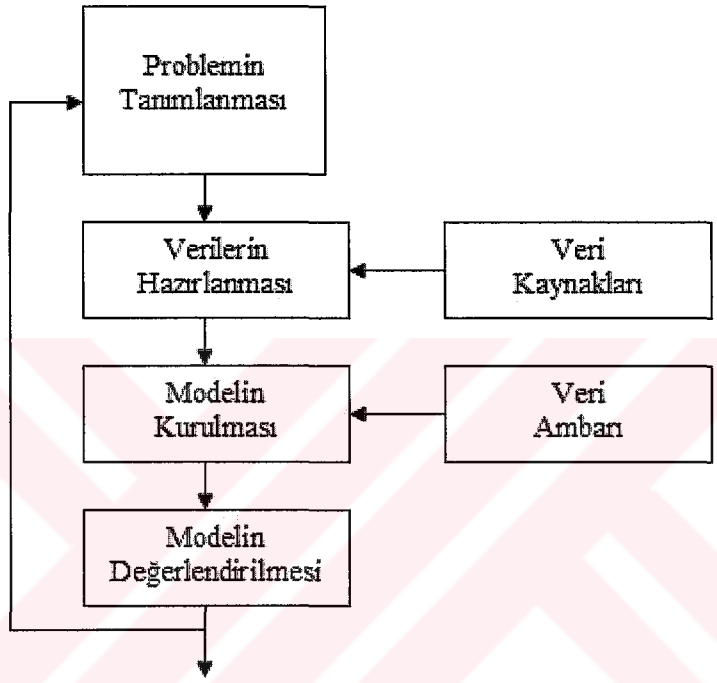
VM' de amaç, kullanıcının bilgi çıkarma sürecinde katkısının olabildiğince az tutulması, işin olabildiğince otomatik olarak yapılabilmesidir. Çünkü OLAP programlarını kullanırken bulunabilecek sonuçlar kullanıcının sormayı düşündüğü sorgularla sınırlıdır. Ama veri içinde çocuk bezi ile bira örneğindeki bağıntı gibi kullanıcının hiç aklına gelmeyecek bilgiler de olabilir. Zaten VM' de esas amaç, bu tip bilgileri bulabilmektir.

Şekil 3.3.' de görüldüğü gibi çeşitli veri kaynaklarından verilerin toplanması ile başlayan VTBK süreci, toplanan verilerin analiz için uygun hale getirilmesi aşaması ile devam etmektedir. Ancak veri ambarına sahip olan kuruluşlarda, gerekli verilerin VP olarak isimlendirilen işleve özel veri tabanlarına aktarılması ile doğrudan VM işlemlerine başlanabilmesi de mümkündür [18].

Fayyad'a göre VTBK sürecinde yer alan adımlar şöyledir [8]:

- Veri Seçimi: Bu adım birkaç veri kümesini birleştirerek, sorguya uygun örneklem kümesini elde etmeyi gerektirir.
- Veri Temizleme ve Ön işleme: Seçilen örneklemde yer alan hatalı tutanakların çıkarıldığı ve eksik nitelik değerlerinin değiştirildiği aşamadır. Bu aşama keşfedilen bilginin kalitesini arttırır.

- Veri İndirgeme: Seçilen örneklemden ilgisiz niteliklerin atıldığı ve tekrarlı tutanakların ayıklandığı adımdır. Bu aşama seçilen VM sorgusunun çalışma zamanını iyileştirir.
- Veri Madenciliği: Verilen bir VM sorgusunun (sınıflama, kümeleme, birliktelik, vb.) işletilmesidir.
- Değerlendirme: Keşfedilen bilginin geçerlilik, yenilik, yararlılık ve basitlik kıstaslarına göre değerlendirilmesi aşamasıdır.



Şekil 3.3. Veri tabanlarında bilgi keşfi süreci ve veri madenciliği

VM astronomi, biyoloji, finans, pazarlama, sigorta, tıp, bankacılık, taşımacılık / ulaşım / konaklama, eğitim - öğretim ve birçok başka dalda uygulanmaktadır. Son 20 yıldır Amerika Birleşik Devletleri'nde çeşitli VM algoritmalarının gizli dinlemeden, vergi kaçakçılıklarının ortaya çıkartılmasına kadar çeşitli uygulamalarda kullanıldığı bilinmektedir. Bununla birlikte günümüzde VM teknikleri özellikle işletmelerde çeşitli alanlarda başarı ile kullanılmaktadır. Bu uygulamaların başlıcaları ilgili alanlara göre aşağıda özetlenmiştir [18].

Pazarlama

- Müşterilerin satın alma örüntülerinin belirlenmesi,
- Müşterilerin demografik özellikleri arasındaki bağlantıların bulunması,
- Posta kampanyalarında cevap verme oranının artırılması,
- Mevcut müşterilerin elde tutulması, yeni müşterilerin kazanılması,
- Pazar sepeti analizi,

- Müşteri ilişkileri yönetimi,
- Müşteri değerlendirme,
- Satış tahmini.

Bankacılık

- Farklı finansal göstergeler arasında gizli korelasyonların bulunması,
- Kredi kartı dolandırıcılıklarının tespiti,
- Kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi,
- Kredi taleplerinin değerlendirilmesi.

Sigortacılık

- Yeni poliçe talep edecek müşterilerin tahmin edilmesi,
- Sigorta dolandırıcılıklarının tespiti,
- Riskli müşteri örüntülerinin belirlenmesi.

3.3. Veri Madenciliğinde Karşılaşılan Problemler

Küçük veri kümelerinde hızlı ve doğru bir biçimde çalışan bir sistem, çok büyük veritabanlarına uygulandığında, tamamen farklı davranabilir. Bir VM sistemi tutarlı veri üzerinde mükemmel çalışırken, aynı veriye gürültü eklendiğinde kayda değer bir biçimde kötüleşebilir.

3.3.1. Veritabanı Boyutu

Veritabanı boyutları inanılmaz bir hızla artmaktadır ve oldukça küçük örneklemleri dahi ele alabilecek biçimde geliştirilmiştir. Aynı algoritmaların yüz binlerce kat büyük örneklemlerde kullanılabilmesi için azami dikkat gerekmektedir. Örneklemin büyük olması, örüntülerin gerçekten var olduğunu göstermesi açısından bir avantajdır. Ancak böyle bir örneklemden elde edilebilecek olası örüntü sayısı çok büyüktür. Bu yüzden VM sistemlerinin karşı karşıya olduğu en önemli sorunlardan biri, veritabanı boyutunun çok büyük olmasıdır. Dolayısıyla VM yöntemleri ya sezgisel/buluşsal bir yaklaşımla arama uzayını taramalıdır ya da örneklemi yatay/dikey olarak indirgemelidir.

Yatay indirgeme, nitelik değerlerinin önceden belirlenmiş genelleme sıradüzenine göre, bir üst nitelik değeri ile değiştirilme işlemi yapıldıktan sonra aynı olan çokluların çıkarılması

işlemidir. Dikey indirgeme, artık niteliklerin indirgenmesi işlemidir. Özellik seçimi yöntemleri ya da nitelik bağımlılık çizelgesi uygulanarak yapılır.

3.3.2. Gürültülü Veri

Büyük veritabanlarında pek çok niteliğin değeri yanlış olabilir. Bu hata, veri girişi sırasında yapılan insan hataları veya girilen değerin yanlış ölçülmesinden kaynaklanır. Veri girişi ya da veri toplanması sırasında oluşan sistem dışı hatalara gürültü adı verilir. Ancak günümüzde kullanılan ticari ilişkisel veritabanları; veri girişi sırasında oluşan hataları otomatik biçimde gidermek konusunda az bir destek sağlamaktadır. Hatalı veri, gerçek dünya veritabanlarında ciddi problem oluşturabilir. Bu durum, bir VM yönteminin kullanılan veri kümesinde bulunan gürültülü verilere karşı daha az duyarlı olmasını gerektirir. Gürültülü verinin yol açtığı problemler tümevarımsal karar ağaçlarında uygulanan yöntemler bağlamında kapsamlı bir biçimde araştırılmıştır. Eğer veri kümesi gürültülü ise sistem bozuk veriyi tanımalı ve ihmal etmelidir. Quinlan [19] gürültünün sınıflama üzerindeki etkisini araştırmak için bir dizi deneyler yapmıştır. Deneysel sonuçlar, etiketli öğrenmede etiket üzerindeki gürültü öğrenme algoritmasının performansını doğrudan etkileyerek, düşmesine sebep olmuştur. Buna karşın eğitim kümesindeki nesnelerin özellikleri/nitelikleri üzerindeki en çok %10'luk gürültü miktarı ayıklanabilmektedir. Chen ve Wong [16] gürültünün etkisini analiz etmek için istatistiksel yöntemler kullanmışlardır.

3.3.3. Boş Değerler

Veritabanlarında boş değer birincil anahtarda yer almayan herhangi bir niteliğin değeri olabilir. Bir çokluda eğer bir nitelik değeri boş ise o nitelik bilinmeyen ve uygulanamaz bir değere sahiptir. Bu durum ilişkisel veritabanlarında sıkça karşımıza çıkmaktadır. Bir ilişkide yer alan tüm çoklular aynı sayıda niteliğe (niteliğin değeri boş olsa bile) sahip olmalıdır. Örneğin kişisel bilgisayarların özelliklerini tutan bir ilişkide bazı model bilgisayarlar için ses kartı modeli niteliğinin değeri boş olabilir.

Lee boş değeri; "*bilinmeyen*", "*uygulanamaz*" ve "*bilinmeyen veya uygulanamaz*" olacak biçimde üçe ayıran bir yaklaşımı ilişkisel veritabanlarını genişletmek için öne sürmüştür[20]. Mevcut boş değer taşıyan veri için herhangi bir çözüm sunmayan bu yaklaşımın dışında bu konuda sadece bilinmeyen değer üzerinde çalışmalar yapılmıştır [21]. Boş değerli nitelikler veri kümesinde bulunuyorsa, ya bu çoklular tamamıyla ihmal edilmeli ya da bu çoklularda niteliğe olası en yakın değer atanmalıdır [21].

3.3.4. Eksik Veri

Evrendeki her nesnenin ayrıntılı bir biçimde tanımlandığını ve bu nesnelerin alabileceği değerler kümesinin belirli olduğu varsayılsın. Verilen bir bağlamda her bir nesnenin tanımı kesin ve yeterli olsa idi, sınıflama işlemi basitçe nesnelerin altkümelerinden faydalanılarak yapılırdı. Bununla birlikte veriler, kurum ihtiyaçları göz önünde bulundurularak düzenlenip toplandığından, mevcut veri gerçek hayatı yeterince yansıtmayabilir. Örneğin hastalığın tanısını koymak için kurallar sadece çok yaşlı insanların belirtilerinin bulunduğu bir veri kümesi kullanılarak üretilseydi, bu kurallara dayanarak bir çocuğa tanı koymak pek doğru olmazdı. Bu gibi koşullarda bilgi keşif modeli belirli bir güvenlik derecesinde tahmini kararlar alabilmelidir.

3.3.5. Artık Veri

Verilen veri kümesi, eldeki probleme uygun olmayan veya artık nitelikler içerebilir. Bu durum pek çok işlem sırasında karşımıza çıkabilir. Örneğin eldeki problem ile ilgili veriyi elde etmek için iki ilişki birleştirilirse, elde edilen ilişkide kullanıcının farkında olmadığı artık nitelikler bulunur. Artık nitelikleri elemek için geliştirilmiş algoritmalar, özellik seçimi olarak adlandırılır.

Özellik seçimi, tümevarıma dayalı öğrenmede budama öncesi yapılan işlem, hedef bağlamı tanımlamak için yeterli ve gerekli olan niteliklerin küçük bir alt kümesinin seçimi problemidir. Özellik seçimi yalnız arama uzayını küçültmekle kalmayıp, sınıflama işleminin kalitesini de arttırır [20,21].

3.3.6. Dinamik Veri

Kurumsal çevrim-içi veritabanları dinamiktir, yani içeriği sürekli olarak değişir. Bu durum, bilgi keşfi yöntemleri için önemli sakıncalar doğurmaktadır. İlk olarak sadece okuma yapan ve uzun süre çalışan bilgi keşfi yöntemi bir veritabanı uygulaması olarak mevcut veri tabanı ile birlikte çalıştırıldığında mevcut uygulamanın da performansı ciddi ölçüde düşer. Diğer bir sakınca ise veritabanında bulunan verilerin kalıcı olduğu varsayılıp, çevrimdışı veri üzerinde bilgi keşif yöntemi çalıştırıldığında, değişen verinin elde edilen örüntülere yansıması gerekmektedir. Bu işlem, bilgi keşfi yönteminin ürettiği örüntüleri zaman içinde değişen veriye göre sadece ilgili örüntüleri yığmalı olarak güncelleme yeteneğine sahip olmasını gerektirir. Aktif veritabanları tetikleme mekanizmalarına sahiptir ve bu özellik bilgi keşif yöntemleri ile birlikte kullanılabilir.

3.4. Veri Tabanlarında Bilgi Keşfi Süreci

Ne kadar etkin olursa olsun hiç bir VM algoritmasının üzerinde inceleme yapılan işin ve verilerin özelliklerinin bilinmemesi durumunda fayda sağlaması mümkün değildir. Bu nedenle aşağıda tanımlanan tüm aşamalardan önce, iş ve veri özelliklerinin öğrenilmesi / anlaşılması, başarının ilk şartı olacaktır.

Şekil 3.3.' de ayrıntılı olarak görüldüğü gibi,

- Problemin Tanımlanması,
- Verilerin Hazırlanması,
- Modelin Kurulması ve Değerlendirilmesi,
- Modelin Kullanılması,
- Modelin İzlenmesi

veri tabanlarında bilgi keşfi sürecinde izlenmesi gereken temel aşamalardır [4].

3.4.1. Problemin Tanımlanması

VM çalışmalarında başarılı olmanın ilk şartı, uygulamanın hangi işletme amacı için yapılacağının açık bir şekilde tanımlanmasıdır. İlgili işletme amacı işletme problemi üzerine odaklanmış ve açık bir dille ifade edilmiş olmalı, elde edilecek sonuçların başarı düzeylerinin nasıl ölçüleceği tanımlanmalıdır. Ayrıca yanlış tahminlerde katlanılacak olan maliyetlere ve doğru tahminlerde kazanılacak faydalara ilişkin tahminlere de bu aşamada yer verilmelidir [22].

3.4.2. Verilerin Hazırlanması

Modelin kurulması aşamasında ortaya çıkacak sorunlar, bu aşamaya sık sık geri dönülmesine ve verilerin yeniden düzenlenmesine neden olacaktır. Bu durum verilerin hazırlanması ve modelin kurulması aşamaları için bir analizcinin veri keşfi sürecinin toplamı içerisinde enerji ve zamanının % 50 - % 85'ini harcamasına neden olmaktadır. Verilerin hazırlanması aşaması kendi içerisinde toplama, değer biçme, birleştirme ve temizleme, seçme ve dönüştürme adımlarından meydana gelmektedir [22].

3.4.2.1. Toplama

Tanımlanan problem için gerekli olduğu düşünülen verilerin ve bu verilerin toplanacağı veri kaynaklarının belirlenmesi adımıdır. Verilerin toplanmasında kuruluşun kendi veri

kaynaklarının dışında, nüfus sayımı, hava durumu, merkez bankası gibi veri tabanlarından veya veri pazarlayan kuruluşların veri tabanlarından faydalanılabilir.

3.4.2.2. Değer Biçme

VM' de kullanılacak verilerin farklı kaynaklardan toplanması, doğal olarak veri uyumsuzluklarına neden olacaktır. Bu uyumsuzlukların başlıcaları farklı zamanlara ait olmaları, kodlama farklılıkları (örneğin bir veri tabanında cinsiyet özelliğinin e/k, diğer bir veri tabanında 0/1 olarak kodlanması), farklı ölçü birimleridir. Ayrıca verilerin nasıl, nerede ve hangi koşullar altında toplandığı da önem taşımaktadır.

Bu nedenlerle, iyi sonuç alınacak modeller ancak iyi verilerin üzerine kurulabileceği için, toplanan verilerin ne ölçüde uyumlu oldukları bu adımda incelenerek değerlendirilmelidir.

3.4.2.3. Birleştirme ve Temizleme

Bu adımda farklı kaynaklardan toplanan verilerde bulunan ve bir önceki adımda belirlenen sorunlar mümkün olduğu ölçüde giderilerek, veriler tek bir veri tabanında toplanır. Ancak basit yöntemlerle ve baştan savma olarak yapılacak sorun giderme işlemlerinin, ileriki aşamalarda daha büyük sorunların kaynağı olacağı unutulmamalıdır.

3.4.2.4. Seçim

Bu adımda kurulacak modele bağlı olarak veri seçimi yapılır. Örneğin tahmin edici bir model için bu adım bağımlı ve bağımsız değişkenlerin ve modelin eğitiminde kullanılacak veri kümesinin seçilmesi anlamını taşımaktadır.

Sıra numarası, kimlik numarası gibi anlamlı olmayan ve diğer değişkenlerin modeldeki ağırlığının azalmasına da neden olabilecek değişkenlerin modele girmemesi gerekmektedir. Bazı VM algoritmaları konu ile ilgisi olmayan bu tip değişkenleri otomatik olarak elese de pratikte bu işlemin kullanılan yazılıma bırakılmaması daha akılcı olacaktır. Verilerin görselleştirilmesine olanak sağlayan grafik araçlar ve bunların sunduğu ilişkiler, bağımsız değişkenlerin seçilmesinde önemli yararlar sağlayabilir.

Genellikle yanlış veri girişinden veya bir kereye özgü bir olayın gerçekleşmesinden kaynaklanan verilerin (Outlier), önemli bir uyarıcı enformasyon içerip içermediği kontrol edildikten sonra veri kümesinden atılması tercih edilir.

Modelde kullanılan veri tabanının çok büyük olması durumunda tesadüfîliği bozmayacak şekilde örnekleme yapılması uygun olabilir. Günümüzde hesaplama olanakları ne kadar gelişmiş olursa olsun, çok büyük veri tabanları üzerinde çok sayıda modelin denenmesi zaman kısıtlaması nedeni ile mümkün olamamaktadır. Bu nedenle tüm veri tabanını kullanarak bir kaç model denemek yerine, tesadüfî olarak örneklenmiş bir veri tabanı parçası üzerinde birçok modelin denenmesi ve bunlar arasından en güvenilir ve güçlü modelin seçilmesi daha uygun olacaktır.

3.4.2.5. Dönüştürme

Kredi riskinin tahmini için geliştirilen bir modelde, borç/gelir gibi önceden hesaplanmış bir oran yerine, ayrı ayrı borç ve gelir verilerinin kullanılması tercih edilebilir. Ayrıca modelde kullanılan algoritma, verilerin gösteriminde önemli rol oynayacaktır. Örneğin bir uygulamada bir yapay sinir ağı algoritmasının kullanılması durumunda, kategorik değişken değerlerinin evet/hayır olması; bir karar ağacı algoritmasının kullanılması durumunda ise örneğin gelir değişken değerlerinin yüksek/orta/düşük olarak gruplanmış olması, modelin etkinliğini artıracaktır.

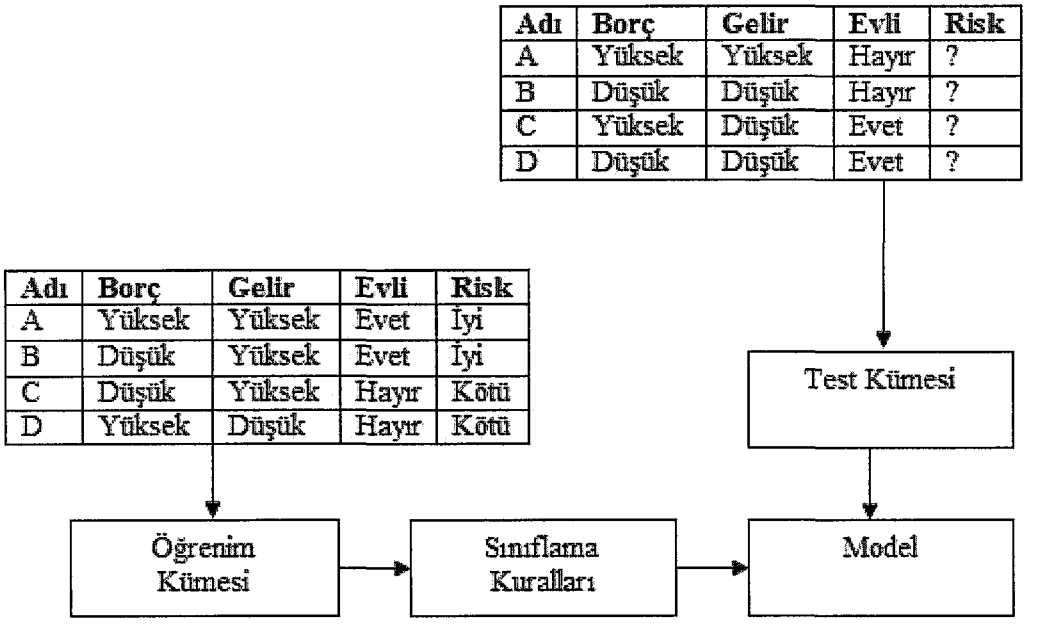
3.4.3. Modelin Kurulması ve Değerlendirilmesi

Tanımlanan problem için en uygun modelin bulunabilmesi, olabildiğince çok sayıda modelin kurularak denenmesi ile mümkündür. Bu nedenle veri hazırlama ve model kurma aşamaları, en iyi olduğu düşünülen modele varılıncaya kadar yinelenen bir süreçtir. Model kuruluş süreci denetimli ve denetimsiz öğrenimin kullanıldığı modellere göre farklılık göstermektedir.

Örnekten öğrenme olarak da isimlendirilen denetimli öğrenimde, bir denetçi tarafından ilgili sınıflar önceden belirlenen bir kritere göre ayrılarak, her sınıf için çeşitli örnekler verilir. Sistemin amacı verilen örneklerden hareket ederek her bir sınıfa ilişkin özelliklerin bulunması ve bu özelliklerin kural cümleleri ile ifade edilmesidir.

Öğrenme süreci tamamlandığında, tanımlanan kural cümleleri verilen yeni örneklere uygulanır ve yeni örneklerin hangi sınıfa ait olduğu, kurulan model tarafından belirlenir.

Denetimsiz öğrenmede, kümeleme analizinde olduğu gibi ilgili örneklerin gözlenmesi ve bu örneklerin özellikleri arasındaki benzerliklerden hareket ederek sınıfların tanımlanması amaçlanmaktadır.



Şekil 3.4. Denetimli öğrenme

Denetimli öğrenimde seçilen algoritmaya uygun olarak ilgili veriler hazırlandıktan sonra, ilk aşamada verinin bir kısmı modelin öğrenimi, diğer kısmı ise modelin geçerliliğinin test edilmesi için ayrılır. Modelin öğrenimi öğrenim kümesi kullanılarak gerçekleştirildikten sonra, test kümesi ile modelin doğruluk derecesi belirlenir.

Bir modelin doğruluğunun test edilmesinde kullanılan en basit yöntem basit geçerlilik testidir. Bu yöntemde tipik olarak verilerin % 5 ile % 33 arasındaki bir kısmı test verileri olarak ayrılır ve kalan kısım üzerinde modelin öğrenimi gerçekleştirildikten sonra bu veriler üzerinde test işlemi yapılır. Bir sınıflama modelinde yanlış olarak sınıflanan olay sayısının, tüm olay sayısına bölünmesi ile hata oranı, doğru olarak sınıflanan olay sayısının tüm olay sayısına bölünmesiyle de doğruluk oranı hesaplanır. (Doğruluk Oranı = 1 - Hata Oranı) Sınırlı miktarda veriye sahip olunması durumunda, kullanılabilecek diğer bir yöntem, çapraz geçerlilik testidir. Bu yöntemde veri kümesi tesadüfî olarak iki eşit parçaya ayrılır. İlk aşamada a parçası üzerinde model eğitimi ve b parçası üzerinde test işlemi; ikinci aşamada ise b parçası üzerinde model eğitimi ve a parçası üzerinde test işlemi yapılarak elde edilen hata oranlarının ortalaması kullanılır.

Bir kaç bin veya daha az satırdan meydana gelen küçük veri tabanlarında, verilerin n gruba ayrıldığı n katlı çapraz geçerlilik testi tercih edilebilir. Verilerin örneğin 10 gruba ayrıldığı bu yöntemde, ilk aşamada birinci grup test, diğer gruplar öğrenim için kullanılır. Bu süreç her defasında bir grubun test, diğer grupların öğrenim amaçlı kullanılması ile sürdürülür. Sonuçta elde edilen on hata oranının ortalaması, kurulan modelin tahmini hata oranı olacaktır.

Bootstrapping küçük veri kümeleri için modelin hata düzeyinin tahmininde kullanılan bir başka tekniktir. Çapraz geçerlilikte olduğu gibi model bütün veri kümesi üzerine kurulur. Daha sonra en az 200, bazen binin üzerinde olmak üzere çok fazla sayıda öğrenim kümesi tekrarlı örneklemelerle veri kümesinden oluşturularak hata oranı hesaplanır.

Model kuruluşu çalışmalarının sonucuna bağlı olarak, aynı teknikle farklı parametrelerin kullanıldığı veya başka algoritma ve araçların denendiği değişik modeller kurulabilir. Model kuruluş çalışmalarına başlamadan önce, imkânsız olmasa da hangi tekniğin en uygun olduğuna karar verebilmek güçtür. Bu nedenle farklı modeller kurarak, doğruluk derecelerine göre en uygun modeli bulmak üzere sayısız deneme yapılmasında yarar bulunmaktadır.

Özellikle sınıflama problemleri için kurulan modellerin doğruluk derecelerinin değerlendirilmesinde basit, ancak faydalı bir araç olan risk matrisi kullanılmaktadır. Aşağıda bir örneği görülen bu matriste sütunlarda fiili, satırlarda ise tahmini sınıflama değerleri yer almaktadır. Örneğin fiilen B sınıfına ait olması gereken 46 elemanın, kurulan model tarafından 2'sinin A, 38'inin B, 6'sının ise C olarak sınıflandırıldığı matriste kolayca görülebilmektedir[18].

Tablo 3.1. Sınıflar

	Fiili		
Tahmini	**A Sınıfı**	**B Sınıfı**	**C Sınıfı**
A Sınıfı	45	2	3
B Sınıfı	10	38	2
C Sınıfı	4	6	40

Önemli diğer bir değerlendirme kriteri modelin anlaşılabilirliğidir. Bazı uygulamalarda doğruluk oranlarındaki küçük artışlar çok önemli olsa da birçok işletme uygulamasında ilgili kararın niçin verildiğinin yorumlanabilmesi, çok daha büyük önem taşıyabilir. Çok ender olarak yorumlanamayacak kadar karmaşıklaşsalar da, genel olarak karar ağacı ve kural temelli sistemler, model tahmininin altında yatan nedenleri çok iyi ortaya koyabilmektedir.

Kaldıraç oranı ve grafiği, bir modelin sağladığı faydanın değerlendirilmesinde kullanılan önemli bir yardımcıdır. Örneğin kredi kartını muhtemelen iade edecek müşterilerin belirlenmesi amacını taşıyan bir uygulamada, kullanılan modelin belirlediği 100 kişinin 35'i gerçekten bir süre sonra kredi kartını iade ediyorsa ve tesadüfî olarak seçilen 100 müşterinin aynı zaman diliminde sadece 5'i kredi kartını iade ediyorsa, kaldıraç oranı 7 olarak bulunacaktır.

Kurulan modelin değerinin belirlenmesinde kullanılan diğer bir ölçü, model tarafından önerilen uygulamadan elde edilecek kazancın bu uygulamanın gerçekleştirilmesi için katlanılacak maliyete bölünmesi ile elde edilecek olan yatırımın geri dönüş oranıdır.

Kurulan modelin doğruluk derecesi ne denli yüksek olursa olsun, gerçek dünyayı tam anlamı ile modellediğini garanti edebilmek mümkün değildir. Yapılan testler sonucunda geçerli bir modelin doğru olmamasındaki başlıca nedenler, model kuruluşunda kabul edilen varsayımlar ve modelde kullanılan verilerin doğru olmamasıdır. Örneğin modelin kurulması sırasında varsayılan enflasyon oranının zaman içerisinde değişmesi, bireyin satın alma davranışını belirgin olarak etkileyecektir.

3.4.4. Modelin Kullanılması

Kurulan ve geçerliliği kabul edilen model doğrudan bir uygulama veya bir başka uygulamanın alt parçası olarak da kullanılabilir. Kurulan modeller risk analizi, kredi değerlendirme, dolandırıcılık tespiti gibi işletme uygulamalarında doğrudan kullanılabileceği gibi, promosyon planlaması simülasyonuna entegre edilebilir veya tahmin edilen envanter düzeyleri yeniden sipariş noktasının altına düştüğünde, otomatik olarak sipariş verilmesini sağlayacak bir uygulamanın içine de gömülebilir.

3.4.5. Modelin İzlenmesi

Zaman içerisinde bütün sistemlerin özelliklerinde ve dolayısıyla ürettikleri verilerde ortaya çıkan değişiklikler, kurulan modellerin sürekli olarak izlenmesini ve gerekiyorsa yeniden düzenlenmesini gerektirecektir. Tahmin edilen ve gözlenen değişkenler arasındaki farklılığı gösteren grafikler model sonuçlarının izlenmesinde kullanılan yararlı bir yöntemdir [18].

4. SINIFLANDIRICILAR

Sınıflandırıcılar, herhangi bir giriş vektörünün çeşitli özelliklerini göz önüne alarak, daha önceden oluşturulmuş bir veri yapısında belirtilen özelliklerle kıyaslayarak kendisine en yakın özelliklere sahip olan sınıfı bulma işlemini gerçekleştirirler. Bu amaçla nesne tanımada kullanılan sınıflandırıcılar, girilen herhangi bir nesnenin öznitelik vektörünü işleyerek, kendisine en yakın özelliklere sahip sınıfı bulmasını sağlayarak nesneyi tanımlarlar [23]. Başlıca nesne sınıflandırıcıları 4 türdür. Bunlar:

1. Klasik yöntemleri kullanan sınıflandırıcılar,
2. Bulanık sınıflandırıcılar,
3. Yapay sinir ağları sınıflandırıcıları,
4. Uyarlamalı ağ tabanlı bulanık çıkarım sistemi sınıflandırıcılarıdır.

Nesneler çok sayıda değişkene ve belirsizliğe sahip olduklarından, nesne sınıflandırma işlemlerinde klasik seri programlama teknikleri ile genelde yeterli başarıya ulaşılamamıştır. Bu nedenle, nesne sınıflandırma işleminde kullanılan ilk iki sınıflandırıcı tipi, ardışık çalıştıklarından, istenilen (hız, tanıma doğruluğu vs.) şartlarda nesne tanımada başarı sağlayamamaktadırlar. Buna karşılık yapay sinir ağı sınıflandırıcıları, birbirine bağlı basit hesaplama elemanlarının paralel çalışmalarına ve insan sinir sisteminin işleyişini taklide dayanan, geniş amaçlı olarak nesne tanımada kullanılmaya başlanan umut verici bir modeldir. Model, basit hesaplama elemanları kullanmasına rağmen, çok sayıda elemanın paralel çalışmaları sonucunda, istenilen şartları çok yüksek oranlarda sağlayabilmektedir [24]. Alt bölümlerde bu sınıflandırıcılardan bahsedilmiştir.

4.1. Klasik sınıflandırıcılar

Klasik sınıflandırıcıların nesne tanıma problemine getirdiği çözümü en genel anlamda şu şekilde ifade edebiliriz; m tane farklı sınıf olup her biri (n_1, n_2, ... n_m) şeklinde elemanlardan oluşan bir veri tabanı olsun. Her bir sınıfın her elemanı ise p tane özelliğe sahip (x_1, x_2, ... x_p) bir vektördür. Tanınması istenen nesnenin öznitelik vektörü, aşağıda belirtilecek olan "en yakın komşu" yöntemini kullanan klasik sınıflandırıcıya göre, veri tabanında en çok benzediği sınıfa dahil edilerek tanımlanır [3].

Bu yöntem, nesne tanımanın en klasik metotlarından birisi olup, tanımlanması istenen nesnenin vektörünü, veritabanındaki en yakın komşusunun sınıfına dahil ederek tanımlar.

Yöntem, örnek vektörün istatistiksel dağılımından bağımsız olup, yalnızca en yakın komşunun sınıfına göre bir sınıflandırma yaparak tanıma işlemini gerçekleştirir.

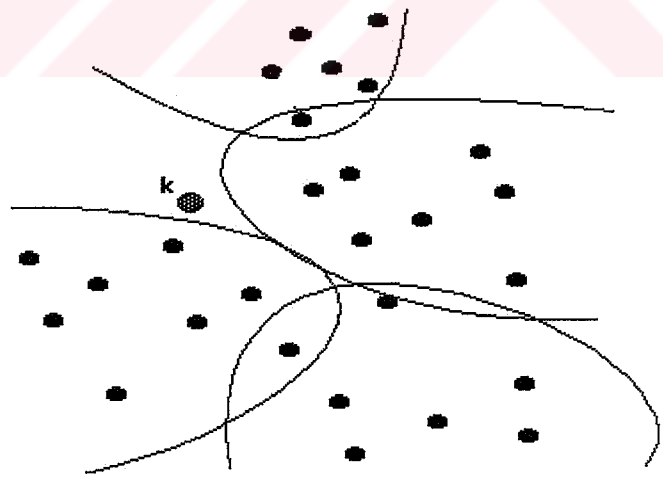
Bu yöntemde, örnek nesnenin vektörü alınarak, veri tabanındaki her bir vektöre olan uzaklığı ölçülür. En çok kullanılan mesafe ölçüsü öklit uzaklığı olsa da başka herhangi bir ölçü de kullanılabilir. Tanımlanacak olan örnek nesnenin vektörü, veri tabanındaki kendisine en çok benzeyen, nesnenin sınıfından sayılır (Şekil 4.1.). Böylece örnek nesnenin vektörü en çok hangi sınıfın içinde yer alıyorsa, örnek nesne bu sınıf türünden bir nesne tipi olarak tanımlanır. Yöntem, matematiksel olarak aşağıdaki gibi ifade edilebilir:

Herhangi bir k vektörünün sınıflandırılacağını varsayalım. Elimizdeki veri tabanında toplam vektör sayısı n olsun. Veri tabanındaki her vektörün sınıfı bilinmektedir ve sınıfla vektör arasındaki ilişki;

$C(x_i) = j$ ($j \in [1,m]$) olarak verilir.

Yine msf() adında, iki vektör arasındaki mesafeyi veren bir fonksiyon olduğunu varsayalım. O halde en yakın komşu yöntemi şu şekilde ifade edilebilir;

$C(k) = C(\min(msf(k,x_i)))$ $i \in [1,n]$ $\forall x_i$ için



Şekil 4.1. Veri tabanındaki sınıflar ve en yakın komşu yöntemine göre k vektörünün sınıflandırılması

4.2. Bulanık sınıflandırıcılar

Bir bulanık sınıflandırıcının temelinde, "bir örüntünün w_i sınıfına üyelik derecesi ß dır" tanımlaması yatmaktadır. Genel olarak bir bulanık sınıflandırıcı, bir örüntünün maksimum üyelik derecesine sahip olduğu sınıfın bulunmasını sağlar.

Bulanık sınıflandırıcıların, diğer bir önemli avantajı ise lineer olmayan olaylardaki başarılarıdır. Bir nesne hakkındaki bilgilerin belirsizlikler içerdiğini belirtmiştik; bu belirsizlikler üyelik fonksiyonları ile ifade edilebilirler. Sınıflandırma, bu fonksiyonların incelenmesinden elde edilen üyelik dereceleri ile gerçekleştirilir.

En çok bilinen bulanık sınıflandırıcıları ise bulanık isodata ve bulanık c-mean algoritmasıdır. Bulanık c-mean algoritması, bir bulanık objektif fonksiyonunu optimize etmeyi temel alan ve data noktalarının kümelendirilmesi için en bilinen ve en popüler algoritmadır. Bu sınıflandırıcı aynı zamanda serbest model tahmin edici ve bulanık c-mean kümelemenin genel kavramı üzerinde esaslanmıştır. Sınıflandırıcı eğitim verisi üzerinde üyelik fonksiyonlarını hesaplar ve kümeleme merkezinin yerini belirler. Yeni verinin sınıflandırılması öğrenilen üyelik fonksiyonları üzerine temellenmiştir. Bu algoritma temel olduğundan aşağıda kısaca verilmiştir[6].

4.2.1. Matematiksel temel

Her bir küme veya kategori prototipi veya merkezi, v_i ve bir üyelik fonksiyonu veya U ayrılma (bölünme) matrisi ile karakterize edilir. U'nun bir elamanı olarak, i kategorisindeki k örüntüsünün üyelik derecesi μ_{ik} ifadesi olarak yazılır. Bu üyelik ile verilir,

$$\mu_{ik} = \frac{\left(\frac{1}{\left\| x_k - v_i \right\|^2} \right)^{\frac{1}{m-1}}}{\sum_{j=1}^{N_C} \left(\frac{1}{\left\| x_k - v_i \right\|^2} \right)^{\frac{1}{m-1}}} \qquad (4.1)$$

Burada, N_C kategorilerin sayısıdır, m; $1 < m < \infty$ arasında bir fuzzy parametresidir ve x_k k. eğitim örneğidir. Çift çizgi sembolü bir uzaklık ölçüsüdür. Mevcut uygulamalarda, öklit uzaklığı olarak benimsenmektedir. x_k örneği v_i merkezine yakınlaştığında üyeliği 1'e yaklaşır,

uzağında ise üyeliği 0'a yaklaşır. Yukarıdaki denklemin payı i kümesine uygun olma (dahil olma) miktarı olarak yorumlanabilir ve payda ise kümelerin seçilen guruba uygun olma seçeneğinin toplam miktarı olarak yorumlanabilir. Bu nedenle, yukarıdaki denklem üyeliğin bir normalizasyonudur (ortalamasıdır).

i. küme merkezi şu şekilde verilir:

$$v_i = \frac{1}{\sum_{k=1}^{n} (\mu_{ik})^m} \sum_{k=1}^{n} (\mu_{ik})^m x_k \tag{4.2}$$

Burada, n eğitim örneklerinin sayısıdır. Küme merkezi eğitim örneklerinin bir ortalaması olarak düşünülebilir, üyeliklere uygun olarak ağırlıklaşır. Aynı zamanda sorgulamadaki kategori ile uygun olarak $\sum_{k=1}^{n} (\mu_{ik})^m$ örneğinin bulanıklık numarası ile normalize edilir [6].

4.2.2. Eğitme

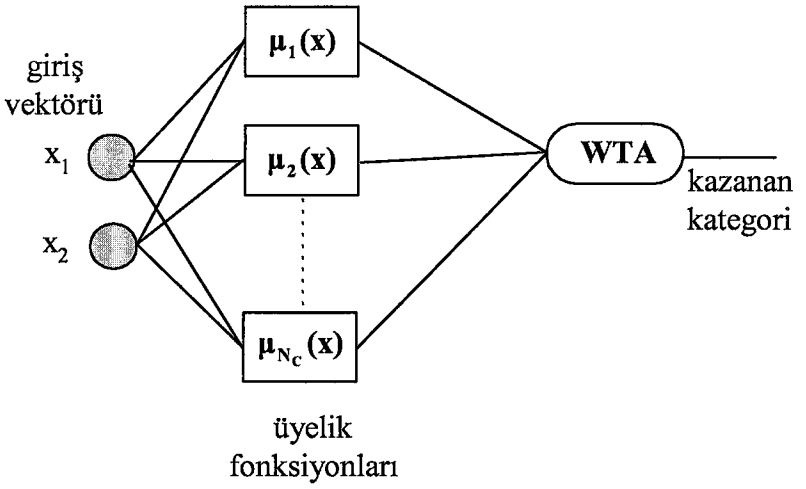
Eğitim esnasında, aşağıda belirtilen objektif (amaç) fonksiyonu minimize edilir:

$$\sum_{i=1}^{N_C} \sum_{k=1}^{n} (\mu_{ik})^m \left\| x_k - v_i \right\|^2 \tag{4.3}$$

v_i küme merkezleri ve μ_{ik} üyelik fonksiyonları minimizasyon esnasında yararlanılabilir parametrelerdir. Amaç fonksiyonunun gradyenti, üyelik fonksiyonları ve küme merkezleri μ_{ik} ve v_i denklemleri ile tanımlanan sistem kazancıdır. Bu sistem denklemleri fuzzy kümeleme algoritması olarak bilinen ardışık bir prosedür tarafından çözülebilir [6].

4.2.3. Test Etme

Yeni veri verildiğinde, çeşitli kümelere üyelikler hesaplanır. Bir kısmi matristen ziyade bir kısmi vektör, sorguda yalnız bir örnek alınarak hesaplanır. En yüksek üyelik ile kümeleme Şekil 4.2.' deki sınıflandırıcı kararı olarak seçilir [6].



Şekil 4.2. Bulanık sınıflandırıcı

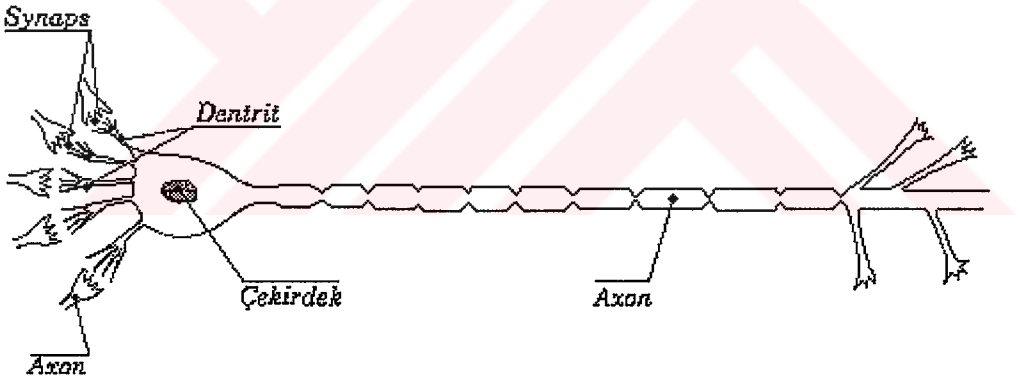
4.3. Yapay sinir ağları sınıflandırıcısı

YSA (Yapay sinir ağları), insan beyninden esinlenerek geliştirilmiş, ağırlıklı bağlantılar aracılığıyla birbirine bağlanan ve her biri kendi belleğine sahip işlem elemanlarından oluşan paralel ve dağıtılmış bilgi işleme yapılarıdır [24]. YSA, bir başka deyişle, insan beyninin çalışma prensibini taklit etme esası üzerine kurulmuş bilgi işleme yöntemleridir. Bu ağlar birbirine paralel olarak bağlanmış işlem elemanlarından (yapay sinir ağı hücresi, nöron, ünite, birim, düğüm) ve onların hiyerarşik bir organizasyonundan oluşurlar. YSA' nın temel düşüncesiyle insan beyninin fonksiyonları arasında benzerlikler vardır. YSA, her ne kadar temel yapı itibariyle bir kısım özellikleri insan beyninin fiziki özelliklerinden esinlenerek ortaya atılmış ise de, kesinlikle şu andaki halleri ile insan beyninin ne tam ne de yaklaşık bir modeli olarak değerlendirilemezler [25].

YSA bir programcının yeteneklerini gerektirmeyen kendi kendine öğrenme sistemidir. Bu ağlar öğrenmenin yanı sıra bilgiler arasında ilişkiler oluşturma yeteneğine de sahiptir. YSA insan beynine benzeyen organizasyon özelliklerini kullanmaktadır. YSA model seçimi ve sınıflandırılması, işlev tahmini, en uygun değeri bulma ve veri sınıflandırılması gibi işlerde başarılıdır [24].

Biyolojik sinir ağının en temel elemanı olan sinir hücresi, sinir sistemi içerisindeki fonksiyon ve görevlerine göre değişik şekil ve büyüklükte olabilir (Şekil 4.3). Bütün hücrelerin ortak bazı özellikleri vardır. Nöronun bir ucunda "dentrit" adı verilen ve hücreye diğer hücrelerden veya dış dünyadan gelen bilgileri toplayan bağlantı elemanı, diğer ucunda ise tek bir life benzer "axon" adı verilen ve hücreden diğer hücrelere ve dış dünyaya bilgi taşıyan

bağlantı elemanı vardır. Axon diğer hücrelerle birleşme esnasında dağınık dallara ayrılmaktadır. Bu iki uçtaki bağlantı noktalarının, elektrofizyolojik olarak hücrelerdeki bilgileri işlemede önemli yeri vardır. Hücrelerin birbiri ile elektrik sinyalleri vasıtasıyla irtibat kurduğu belirlenmiştir. Sinyaller bir hücrenin axon 'undan, diğerinin dentrit 'ine gönderilir. Bir axon birden fazla dentrit ile bağlantı kurabilir. Bu bağlantıların yapıldığı yere "synaps" denir. Hücreler, elektrik sinyalini hücre duvarlarındaki voltajı değiştirerek üretirler. Bu ise, hücrenin içinde ve dışında dağılmış iyonlar vasıtası ile olur. Bu iyonlar sodyum, kalsiyum, potasyum ve klorin gibi iyonlardır. Bir hücre, diğer hücreye elektrik enejisini bu kimyasal iyonlar vasıtasıyla transfer eder. Bazı iyonlar elektrik ve manyetik kutuplaşmaya sebep olurken, bazıları kutuplaşmadan kurtulup hücre zarını açarak iyonların hücreye geçmesine olanak sağlar. Zaten sinyallerin bir hücreden diğerine akmasını sağlayan da bu kutuplaşmanın azalması olayıdır. Sinyaller, hücrenin etkinliğini belirler. Bir hücrenin etkinliği, hücreye gelen synaps sayısı, synaps 'lardaki iyonların konsantrasyonu ve bir de synaps'ın sahip olduğu güç olmak üzere üç faktöre bağlıdır. Bir hücre sahip olduğu impuls miktarınca diğer hücreleri etkiler. Bazı hücreler diğerlerinin impulslarını pozitif yönde, bazı hücreler de negatif yönde etkiler. İnsan sinir ağı sistemi, bu şekilde çalışan milyonlarca hücrenin bir araya gelmesinden oluşur [3].



Şekil 4.3. Biyolojik nöronun şematik yapısı

Biyolojik beynin en önemli özelliklerinden birisi de öğrenme olayıdır. İnsanlar ve hayvanlar, sürekli olarak içerisinde bulundukları çevre ile ilişki neticesinde bir öğrenme işlemi içerisindedirler. Öğrenilen her yeni bilgi, beynin fonksiyonlarını hemen etkileyerek, yapılan davranışlara yansır. Yapay sinir ağlarının gerçekleştirilmesinde bu özellik esas teşkil eder.

Yapay sinir ağları, biyolojik sinir ağlarından esinlenerek modellendirilmiş olup, onlardan çok daha basit bir yapıya sahiptir. Geliştirilen birçok yapay sinir ağı biyolojik sinir ağlarının bilinen birkaç özelliğini (öğrenme kabiliyeti gibi) temsil etmek üzere geliştirilmiştir. Bir takım özellikler ise nörofizyolojik yaklaşımlar yerine mühendislik yaklaşımı ile geliştirilmektedir.

YSA' yı diğer sınıflandırıcılardan ayıran bir takım özellikleri vardır [3]. Bu özellikler;

1. **Öğrenme:** Yapay sinir ağları, örüntüler hakkındaki ilişkiyi belirli bir algoritmaya dayanarak çözmek yerine o ilişkiyi gösteren örüntü örneklerini incelemek suretiyle çözümler üretirler. Burada, sınıflandırılacak örüntü ile alakalı sinir ağına örneklerden başka hiç bir ön bilginin verilmemiş olması dikkat çekicidir. Ağ, kendisine gösterilen örnekleri tekrar tekrar inceleyerek aradaki ilişkiyi kavramaya çalışır. Her yeni örnek, ağın sahip olduğu bilgiye bir yenisini ekler ve bu işlem tekrar ettikçe ilgili örüntü sınıflandırma problemi hakkında bazı genellemeler yapılır.
2. **Genelleme:** Alışagelmiş bir takım sınıflandırma karakteristiklerinde, istenen çıkışı üretmek için tam olarak doğru girişlere gereksinim duyulmasıdır. Öte yandan yapay sinir ağları, girişlerinde değişimler olsa bile doğru çıkışı üretebilirler. Yani sistem, daha önce o tipten hiç bir şey görmemiş olmasına rağmen, insanlar gibi tamam olmayan veya kısmen hatalı girişlerle bile doğru tanımlama yapabilmektedir. Buda işaret etmektedir ki, yapay sinir ağları kendilerine gösterilen bir örüntüyü daha önce öğrendikleri ile mukayese ederek ve aradaki benzerlikleri ortaya koyarak, belirli sınıflara ayırma özelliklerine sahiptirler.
3. **Çıkarım yapma:** Yapay sinir ağları tam doğru olmayan bir eğitme kümesinden, tam doğruyu çıkarabilirler. Ses örüntülerini tanımak için eğitilmiş bir yapay sinir ağına, gürültü tarafından bozulmuş ses verilebilir. Buna rağmen eğitimden sonra, sistem girişi bozuk ses olmasına rağmen, çıkışta ses mükemmel bir şekilde oluşturulabilir. Yani sistem, eğitim kümesinin özünü çıkarıp saklamıştır. Böylece, eksik veya gürültülü girişlere karşın uygun şekilde cevap verebilmektedir.
4. **Hata toleransı:** Verilerde eğer bir eksiklik söz konusu olursa, geleneksel yöntemler çalışmazlar. Daha önce belirtildiği üzere, iyi eğitilmiş ve genelleme kapasitesi yüksek bir sinir ağı, kendisine takdim edilen veriler eksik olsa da karar verme işlemine devam eder. Aynı şekilde, yapay sinir ağı üzerinde birtakım problemler ve bozukluklar olabilir. Geleneksel sistemlerin tersine yapay sinir ağları, bu durumda da çalışmalarına devam ederler. Verilerdeki eksiklik veya yapay sinir ağındaki yapısal bozukluk arttıkça yapay sinir ağının performansı yavaş yavaş azalmaya başlar. Fakat sistem fonksiyonunu tamamen durdurmaz ve mutlaka bir sonuç üretilir. Bu özellikler yapay sinir ağının yapısından kaynaklanmaktadır. Çünkü ağın sahip olduğu bilgi, ağ üzerindeki hücrelerin birbiri ile olan bağlantıları üzerine dağıtılmıştır. Zaten böyle bir durumda tek bir bağlantı ve onun üzerindeki bilgi, başlı başına hiçbir zaman bir mana ifade etmez. Ancak, bir gurup halinde ve tam olarak bağlantıların birlikte düşünülmesi sonucu anlamlı bilgiler üretilir. Bundan dolayı birkaç bağlantının etkisiz hale gelmesi, sonucu

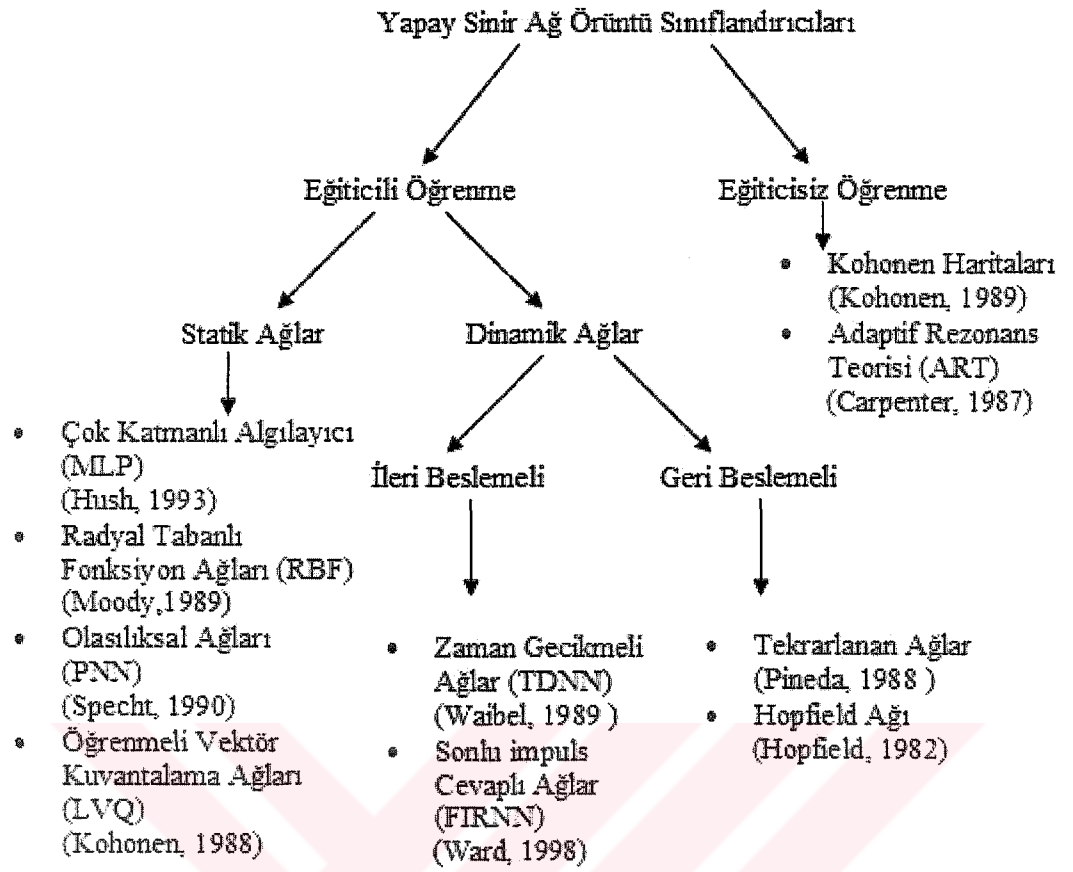
ya etkilemez veya performansı yavaş yavaş düşürür. Yapay sinir ağı, sahip olduğu diğer bağlantılar nedeniyle işlevine devam eder. Geleneksel sistemlerin ardışık çalışmalarından dolayı, sistemdeki en küçük bir hatanın veya bozulmanın ulaşabileceği boyutlar düşünülürse, bu özelliğin ne kadar önemli olduğu ortaya çıkar.

5. **Hız:** Gerçek zaman uygulamalarında bilgi işleme hızı önemli bir yer teşkil eder. Sistemlerin her geçen gün biraz daha karmaşık olduğu, dolayısı ile daha fazla hacimde veriyi daha verimli bir şekilde işleme gerekliliği, yeni yazılım/donanım sistemlerinin zorunluluğunu ortaya çıkarmıştır. Yapay sinir ağlarının da yine birbirlerine bağlı ve paralel işlem elemanlarından oluştuğundan böyle hızlı işleyebilmeleri, bu ağlara özellikle endüstriyel hayatta çok önemli olan gerçek zamanlı çalışma kabiliyeti kazandırır [24].

YSA bilgi sınıflama ve bilgi yorumlamanın da içinde bulunduğu çok değişik problemlerin çözümünde kullanılmaktadır. YSA' nın kullanıldığı alanlar şu şekilde sıralanabilir;

- Sistem modelleme,
- Denetim,
- El yazısı tanıma,
- Ses tanıma,
- Elektrik işareti tanıma,
- Parmak izi tanıma,
- Meteorolojik yorumlama,
- Kalp fonksiyonlarını izleme, tanıma ve yorumlama [26].

Tüm YSA modellerini bir örüntü sınıflandırıcısı olarak kullanmak mümkündür, fakat en yaygın kullanılan ve en güçlü örüntü sınıflandırıcısı çok katmanlı ileri beslemeli ağ olup, tüm yapay sinir ağı uygulamalarının %90 nını kaplamaktadır. Örüntü sınıflandırıcısı olarak kullanılan YSA türleri hiyerarşik bir biçimde Şekil 4.4.de gösterilmiştir [5].



Şekil 4.4. Yapay sinir ağ örüntü sınıflandırıcıları

Çok katmanlı ileri beslemeli yapay sinir ağı parametrik olmayan bir sınıflandırıcıyı göz önünde tutmaktadır. Verinin temelini teşkil eden yapı hakkında tahmin yapılamayacağı gibi, aynı zamanda bu bir serbest model tahmin edicidir. Üç katmanlı ağ eğitim verisinin sonrasal olasılıkları, doğrudan tahmin etmek için yaygın olarak kullanılır. Yeterince verilen örnekler, giriş verisinden çıkış kategorisi elde edilmesiyle herhangi bir rasgele seçilmiş doğrusal olmayan eşleme ile öğrenme gerçekleşebilir [3].

4.3.1. Matematiksel temel

Şekil 4.5.de görülen sinir ağ yapısında her bir bağlantı, eğitme veya öğrenme süreci esnasında uyarlanabilen bir ağırlık değişkenine sahiptir. Giriş nöronları dağıtım katmanında olduğu gibi benzer şekilde hareket ederler ve PNN deki role çok benzerdir. Bununla birlikte her bir gizli nöron, iki işleme tabi tutulur. İlkinde, yeni girişlerin ağırlık toplamı hesap edilir. Gizli j birimi ile çıkış için matematiksel olarak şu yazılabilir [3];

$$\sum_{k=1}^{m} w_{jk} x_k - \theta_j \tag{4.4}$$

Burada, w_{jk} gizli j birimi ile k giriş birimi arasındaki ağırlıktır, x_k k giriş biriminden gelen sinyaldir, θ_j gizli j birimi için eşik değeridir ve m ise girişin boyutudur ve aynı zamanda giriş birimlerinin sayısına denktir. Bir doğrusal olmayan dönüşümle nöron çıkışını üretmek için bir toplam uygulanır. $w_{j0} = \theta_j$ ve $x_0 = -1$ verilmesi ile, j gizli birimin çıkışı şu şekilde yazılabilir [3];

$$f\left(\sum_{k=0}^{m} w_{jk} x_k\right) \tag{4.5}$$

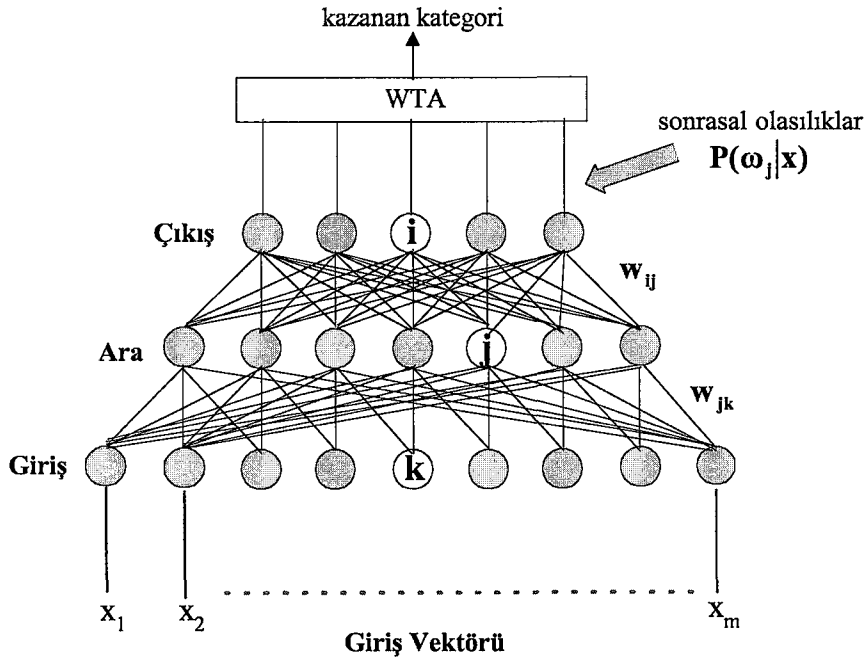
Burada f(.) doğrusal olmayan dönüştürücü olup, aşağıda denklemi verilen bir sigmoid fonksiyondur.

$$f(h) = \frac{1}{1+e^{-h}} \tag{4.6}$$

Çıkış birimleri gizli katmandan alınan girişler üzerindeki, doğrusal olmayan işlemlerin bir benzer icrasıdır. Girişten çıkışı üreten, yaklaştırma fonksiyonu şu şekilde yazılabilir;

$$y_i = f\left(\sum_{j=0}^{H} w_{ij} f\left(\sum_{k=0}^{m} w_{jk} x_k\right)\right) \tag{4.7}$$

Burada H gizli birimlerin sayısıdır, y_i 1. çıkış biriminin değeridir ve w_{ij} j gizli birim ile i çıkış birimi arasındaki bağlantının ağırlığıdır. Bir sınıflandırıcı, x giriş vektörü ile verilen, i kategorisinin sonrasal olasılığının tahmin edicisi olarak bu yaklaştırma fonksiyonudur.



Şekil 4.5. Çok katmanlı ileri beslemeli sinir ağ sınıflandırıcısı

4.3.2. Eğitme

Bir optimizasyon problemi olarak, çok katmanlı sinir ağı eğitme esnasında belirtilen amaç fonksiyonunu minimize etmek için çalışır.

$$E[w] = \frac{1}{2}\sum_{\mu i}\left[\zeta_i^{\mu} - y_i^{\mu}\right]^2 \tag{4.8}$$

Burada, w ağdaki tüm ağırlıkların kapsamları; ζ_i^{μ} ağ μ girişi ile sunulduğunda i çıkışı için istenen çıkıştır ve y_i^{μ} μ girişi ağa sunulduğu zaman i çıkışının gerçek değeridir. w_{ij} ve w_{jk} ağırlıkları E[w] 'nin minimizasyonunda değişen parametrelerdir. Optimizasyon problemi bir ardışık gradyent (eğim) kazanç formülasyonu olarak, aşağıda belirtildiği üzere düzenlenen ağırlık kuralı ile belirtilir:

$$w_{i+1} = w_i - \beta\, \nabla E\left[w_i\right] \tag{4.9}$$

Burada β öğrenme oranı, $\nabla E[w_i]$ 'dün gradyentidir, w_{i+1} yeni ağırlık değeri ve w_i ise önceki değerdir. $\nabla E[w_i]$ gradyent hesaplamak için geri yayınım (backpropagation) etkili bir

yoldur. Böylece sinir ağ sınıflandırıcısı, μ girişleri olarak verilen ve w ağırlıklarının kurulmasında kullanılan eğitme örneklerinden gelen bilgileri kodlarlar [3].

4.3.3. Test Etme

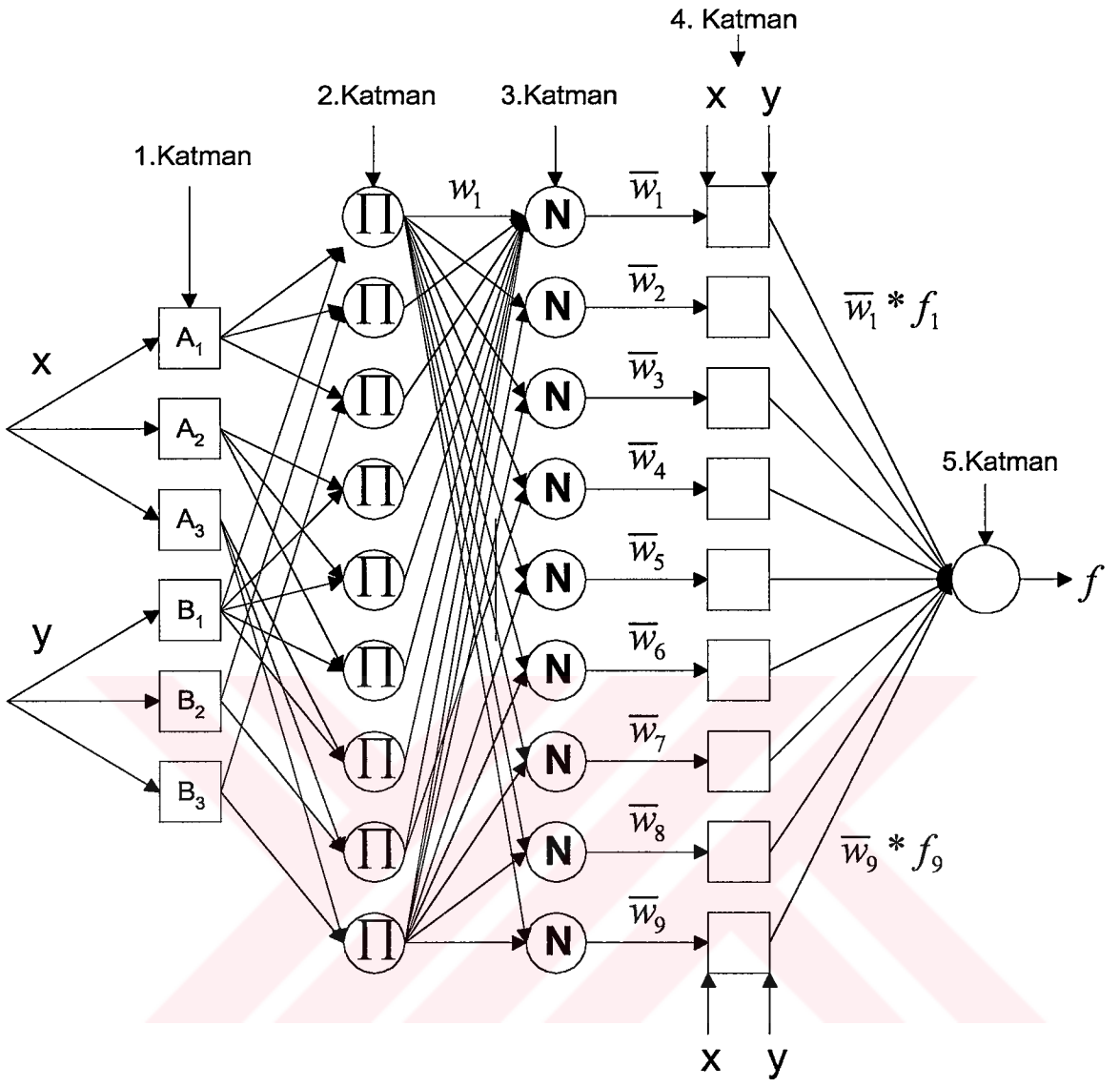
Sinir ağ sınıflandırıcısının öğrenmesi, belirtilen mimarisi için denklem *E[.]* ile minimize edilen ağırlıklar ayarlanarak gerçekleşir, yeni giriş sinyali için sonrasal olasılıklar önceden bildirilebilir. Katmandan katmana gelen basit sinyal üretimlerin de, denklem $y_i()$ ile dikte edilen işlemler gerçekleşir [3].

4.4. Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi ile Sınıflandırma

ANFIS' in, yapısında hem yapay sinir ağları hem de bulanık mantık kullanılır. Yapı bakımından ANFIS, bulanık çıkarım sistemindeki eğer-ise kuralları ve giriş çıkış bilgi çiftlerinden oluşur. Ancak sistem eğitiminde yapay sinir ağı öğrenme algoritmaları kullanılır. x ve y giriş, z ise çıkış olarak alınırsa temel kural yapısı şu şekilde yazılabilir:

Eğer x A_1 ve y B_1 ise $f_1 = p_1 x + q_1 y + r_1$ (4.10)

Burada p ve q lineer çıkış parametreleridir. İki girişli ve bir çıkışlı bir ANFIS 'in temel yapısı Şekil 4.6.' da görülmektedir. Bu yapı, 5 katman ve 9 adet eğer-ise kuralı kullanılarak oluşturulmuştur:



Şekil 4.6. 2 girişli 9 kurallı bir ANFIS sınıflandırıcı yapısı

1.Katman: Bu katmandaki hücre sayısı, iki giriş ve bu iki girişin her birine üç üyelik fonksiyonu tanımlandığına göre altı adettir (i=6'dır). Buna göre,

$O_{1,i}=\mu_{Ai}(x)$, i=1,2,3 için $O1_{,i}=\mu_{Bi-3}(y)$, i=4,5,6 için (4.11)

Burada x ve y girişlerdir. Bu katmanın çıkışı kuralların varsayım (eğer) kısımlarının üyelik fonksiyonlarına olan üyelik dereceleridir. Buradan da görüldüğü gibi üyelik fonksiyonu olarak Gauss üyelik fonksiyonu kullanılmıştır.

$\mu_{Ai}(x), \mu_{Bi-3}(y) = \exp((-(x_i - c_i)/(a_i))^2)$ (4.12)

2. Katman: Burada kuralların kesinlik dereceleri cebirsel çarpım kullanılarak bulunur.

$O_{2,i}=w_i=\mu_{Ai}(x)*\mu_{Bi}(y), \quad i=1,2,3,\ldots,9$ (4.13)

3. Katman: Burada kuralların normalizasyon işlemi yapılmaktadır:

$O_{3,i}=\overline{w_i}=w_i/(w_1+w_2), \quad i=1,2,3,\ldots,9$ (4.14)

4. Katman: Bu katmanda normalize edilmiş her bir kural kendine ait çıkış fonksiyonu ile çarpılır.

$O_{4,i}=\overline{w_i}*f_i=w_i*(p_ix+q_iy+r_i)$ (4.15)

Buradaki p,q ve r lineer parametreleri sonuç parametreleri olarak adlandırılır.

5. Katman: 4. Katman çıkışlarının toplanarak ANFIS çıkışının sayısal değerinin bulunduğu kısımdır [3,21].

$O_{5,i}$=toplam çıkış = $\sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$ (4.16)

Tezin 5. bölümünde YSA ve ANFIS tarzı sınıflandırıcılar kullanılarak veri tabanında tutulan Elazığ iline ait meteorolojik verilerin yorumlanması gerçekleştirilmiştir. Sınıflandırma işleminde en iyi sonuçları YSA ve ANFIS sınıflandırıcıları verdiği için bu iki sınıflandırıcı kullanılmış ve bu sınıflandırıcıların performansları karşılaştırılmıştır [23,27].

5. UYGULAMA

5.1. Veri Madenciliği Çalışması Örnekleri

Veri madenciliği çeşitli sektörler tarafından kullanılabilmektedir. Bu sektörlere ait örnekler verilecek olursa; yapılan bir araştırmaya göre çocuk bezi alan müşterilerin %30'u bira da almaktadır. Bu tür araştırmalarda müşterilerin birlikte aldığı ürünler incelenir. Bu analiz yöntemine sepet analiz yöntemi denilmektedir. Normal düşünüldüğü takdirde çocuk bezi alacak birisinin bebek maması veya domates alan kişinin salatalık veya biber alma ihtimali vardır ve bu tür ihtimaller istatistikî kavramlar tarafından incelenirken, örnekte de belirtildiği insanların aklına kolay kolay gelmeyecek ilişkileri göstermek, VM (Veri Madenciliği)'nin en büyük farkıdır [28].

Bir diğer çalışma ise kişiler ve kişilerin aldığı araba üzerine yoğunlaşmıştır. Bu çalışma ile genç kadınların küçük araba satın aldıkları, yaşlı, zengin erkekler büyük, lüks araba satın aldıkları sonucuna varılmıştır. Bu analiz ile piyasaya giren bir şirket küçük arabalarının reklamlarını kadın dergilerinde, büyük arabalarının reklamlarını ise akşam haber bültenleri, spor müsabakaları öncesi ve sonrasında yaparak satışlarda artış sağlayabilir [29].

Bankacılık sektöründe de bir takım araştırmalar yapılmıştır. Bir finans kurumundan kredi almak için başvuran kişiler 0 ile 1000 arasında puan ile değerlendirirler ve yapılan araştırmaya göre ev sahibi olan, evli, aynı iş yerinde beş yıldan fazladır çalışan, geçmiş kredilerinde geç ödemesi bir ayı geçmemiş bir erkeğin kredi skoru 825'dir [29].

Davranış skoru, kredi almış ve taksitleri ödeyen bir kişinin sonraki taksitlerini ödeme veya geciktirme davranışını notlamayı amaçlayan bir çalışmadır. Bu çalışma ile ilk üç taksitinden iki veya daha fazlasını geç ödemiş olan müşterilerin %60 olasılıkla kanuni takibe gittikleri sonucu elde edilmiştir [29].

Yukarıda VM ile yapılacak çeşitli örneklere yer verilmiştir. Bunlara ilaveten bir veri yığınında belirli bir nesne aramak veya kurala aykırı çeşitli sonuçlar bulmakta veri madenciliği kullanılarak yapılan araştırmalar arasındadır.

VM' nin ilişkili olduğu bir diğer teorem ise veri ambarıdır. VM büyük miktarda bulunan veriyi incelemek için kullanıldığından, bu teorem veri tabanları ile yakından ilgilidir. Ulaşılmak istenen verinin amaca uygun bir formatta saklanması ve gerektiğinde hızlı bir şekilde ulaşılabilecek durumda olması gerekmektedir. Günümüzde yaygın olarak kullanılmaya başlanan veri ambarları günlük kullanılan veri tabanlarının birleştirilmiş ve işlemeye daha uygun bir özetini saklamayı amaçlar. Aynı zamanda VM' nin asıl amacı, sonuçların çıkarılma aşamasında

kullanıcının katkısının olabildiğince az tutulmasıdır. Çünkü VM programlarını kullanırken bulunabilecek sonuçlar kullanıcının sormayı düşündüğü sorgularla sınırlıdır.

VM if-then-else kalıbı gibidir. Aşağıda verilecek örnek, bir hava durumu tahmin raporudur ve buna bağlı olarak çıkabilecek sonuçları incelenecektir [28, 29].

Tablo 5.1. Bir veri tabanı uygulaması

Gökyüzü	Sıcaklık	Nem	Rüzgar	Futbol
Güneşli	Sıcak	Yüksek	Yok	Yok
Güneşli	Sıcak	Yüksek	Var	Yok
Bulutlu	Sıcak	Yüksek	Yok	Var
Yağmurlu	Ilık	Normal	Yok	Var

Tablo 5.1 bir hava durumuna göre dışarıda oyun oynanıp oynanamayacağını belirten bir rapordur. Elde 4 tane kriter bulunmaktadır. Bunlar gökyüzünün güneşli olup olmadığı, sıcaklığın durumu, nemin seviyesi ve rüzgârın olup olmadığıdır.

Eldeki veriye göre şöyle sonuçlar çıkartılabilir.

Eğer Gökyüzü=Güneşli ve Sıcaklık =Sıcak ise Futbol=Yok

Eğer Gökyüzü=Yağmurlu ve Rüzgar=Var ise Futbol=Yok

Eğer Gökyüzü=Bulutlu ise Futbol=Var

Eğer Nem=Normal ise Futbol=Var

Eğer Yukarıdakilerden Hiçbiriyse Futbol=Var

Bunun yanında bu örnek rakamlar ile verilebilirdi. Söz gelimi Sıcaklık ibaresindeki Sıcak-Ilık yerine dereceler verilebilir. Sıcak=80 Ilık=65 gibi o zaman Sıcaklık<65 Sıcaklık>85 gibi kriterler de ortaya çıkacaktır. Rakamlar kullanılarak bu örnekte yapılan bazı değişiklikler, aşağıda verilmiştir (Tablo 5.2).

Tablo 5.2. Rakamsal değerlerle bir veri tabanı uygulaması

Gökyüzü	Sıcaklık (°F)	Nem (%)	Rüzgar	Futbol
Güneşli	85	85	Yok	Yok
Güneşli	80	90	Var	Yok
Bulutlu	83	86	Yok	Var
Yağmurlu	75	80	Yok	Var

Eğer Gökyüzü=Güneşli ve Nem>83 ise Futbol=Yok

Eğer Gökyüzü=Yağmurlu ve Rüzgar=Var ise Futbol=Yok

Eğer Gökyüzü=Bulutlu ise Futbol=Var

Eğer Nem<85 ise Futbol=Var

Eğer Yukarıdakilerden Hiçbiriyse Futbol=Var

VM konusunda temel olarak iki teorem üzerinde durulacaktır. Birincisi Sınıflandırma (Classification), önceden tanımlanmış sıfatın değerini tahmin etmeye yarar. İlk hava durumu istatistiğine göre bir örnek verecek olursak,

Eğer Gökyüzü=Güneşli ve Nem=Yüksek ise Futbol=Yok

İkincisi ise ilişkilendirme (Assosciations), sıfatın rasgele değerini hesaplar veya sıfatların kombinasyonlarını temsil eder. Tekrar ilk hava durumu örneğinden yola çıkacak olursak, aşağıdaki sonuçları elde etmiş oluruz.

Eğer Nem=Normal ve Rüzgâr=Yok ise Futbol=Var

Eğer Gökyüzü=Güneşli ve Futbol=Yok ise Nem=Vardır.

Başka bir örnekte Kanada çevresinin radar karar destek sistemi veritabanından gelen hacimsel fırtına verisini; hangi yaklaşımın en iyi sınıflayacağını saptamak için sınıflama strateji teknikleri kullanılmıştır. Bu çalışmada karşılaştırma için kriter, test edilen bir veri üzerindeki sınıflamadaki katsayının doğruluğuna göre yapılmıştır. Fırtına olaylarının farklı tiplerini sınıflamak için bir fırtına değişim yaklaşımı kullanılmıştır. Farklı fırtına stratejileri ve ilk işlem teknikleri alınmıştır. Ayrıca, sunulan farklı fırtına değişim sınıflama stratejileriyle Kanada çevresindeki radar karar destek sistem verisinden gelen hücresel fırtına hücre verisini, hangi yaklaşımın en iyi sınıflayacağı konusunda araştırma yapılmıştır. 360^0 etrafın taranması ile toplanan radar verileri, hacimsel tarama olarak bilinir. Hacimsel tarama verileri RKDS (Radar Karar Destek Sistemi) tarafından işlenir. RKDS örüntü tanımlama radar karar destek araçları meteoroloji uzmanları için fırtınaların analizi ve gerçek zamanlı taramalar için bir bilgi tabanıdır. Hacimsel verinin yüksekliği, hacimsel verinin açısı, hız gibi değerler özellik olarak çıkarılmıştır. 4 karar sınıfı dolu, şiddetli yağmur, tornado, rüzgâr ve bunların ikili birleşimlerinden oluşan dolu, şiddetli yağmur, tornado, rüzgâr, dolu yada yağmur, dolu yada tornado, dolu yada rüzgâr, yağmur yada tornado, yağmur yada rüzgâr, tornado yada rüzgâr 10 karar sınıfı kullanılmıştır. Yapay Sinir Ağları, Bulanık Sınıflandırıcı (Fuzzy C-Mean) ve Karar Ağaçları (Decision Tree) sınıflandırıcı olarak kullanılmıştır. 4 karar sınıflı verilerin sınıflandırmasında YSA (Yapay Sinir Ağları) sınıflandırıcının en iyi sonucu verdiği ve 10 karar sınıflı verilerin sınıflandırmasında ise Karar Ağaçlarının en iyi sonucu verdiği görülmüştür [30].

İncelenen bu çalışmalar ışığında Elazığ iline ait meteorolojik verilerin sınıflandırılarak değerlendirilmesi gerçekleştirilmiştir.

5.2. Elazığ İlinin Meteorolojik Verilerinin Sınıflandırılıp Değerlendirilmesi

Bu bölümde, incelenen yöntemlerin MATLAB ortamında programları geliştirilerek meteorolojik veriler üzerinde uygulamaları yapılmıştır.

Veri madenciliği; veri ambarlarındaki tutulan çok çeşitli verilere dayanarak daha önce keşfedilmemiş bilgileri ortaya çıkarmak, bunları karar vermek ve eylem planını gerçekleştirmek için kullanma sürecidir.

Veri madenciliği veri içerisinden aranılan bilgiye ulaşma işidir. Madencilik teriminin kullanılma sebebi, büyük bir veri yığını arasından uygun olanı arama ve seçme işleminin maden arama işine benzetilmesindendir.

Veri madenciliği, büyük miktarda veri içinden, gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların aranmasıdır. Bir başka deyişle, veri madenciliği, büyük miktarda veri içinden gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların bilgisayar programları kullanarak aranmasıdır.

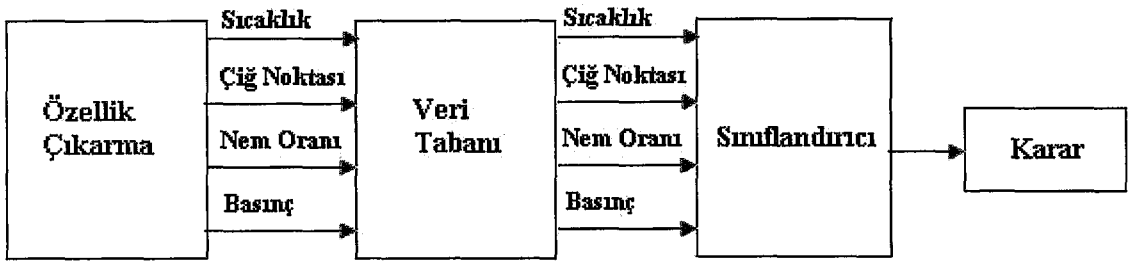
Meteorolojik olaylar, insanoğlunun yaşamını ilk çağlardan itibaren etkilemiş, insanlar durmadan günümüze kadar dünya atmosferinde olup biten olayların nedenlerini, zamanın koşullarına göre inceleyip araştırmışlardır. Bu amaçla da çeşitli gözlem ve incelemeler yaparak hava olaylarını önceden tahmin edebilme yollarını bulmaya çalışmışlar, bunların olumlu etkilerinden faydalanma, olumsuz etkilerinden de kurtulma ve korunma yollarını aramışlardır.

Tüm meteorolojik parametrelerin değişimi, etkileşimi ve sonuçlarını bu anlamda birer kurala dayandırmak mümkündür. Küçük, orta ve küresel ölçekli sayısal hava tahmin modellemelerinin temeli, fiziğin temel hareket kanunlarına dayanmaktadır. Elde edilen meteorolojik verilerin toplanması, kalite kontrolünden geçirilmesi ve değerlendirilmesi çok önemlidir. Depolanan bu anlamsız (ham) meteorolojik verilerden anlamlı bir sonuç çıkarmak için veri madenciliği teknikleri kullanılarak doğru hava tahminlerinin yapılması gerçekleştirilir.

Bu çalışmada, Elazığ iline ait meteorolojik verilerin veri madenciliği teknikleri kullanılarak değerlendirilmesi ve akıllı örüntü tanıma yaklaşımıyla otomatik hava tahmini yapılması gerçekleştirilmiştir. Otomatik hava tahmini, meteorolojik verilerin, veri madenciliği algoritmaları kullanılarak bilgisayar destekli olarak gerçekleştirilmesi, bu çalışmanın asıl konusunu oluşturmaktadır. Radarlardan alınan günlük meteorolojik verileri sağlayan www.wunderground.com internet sitesinden, Elazığ iline ait meteorolojik veriler alınmıştır. Her bir güne ait meteorolojik veriler bir veri tabanında depolanmıştır. Veri tabanında farklı hava şartları için depolanan işlenmemiş sıcaklık, çiğ noktası, nem oranı ve basınç verileri veri madenciliği ile değerlendirilerek, Elazığ iline ait o günün hava tahmini akıllı sınıflandırıcılar ile

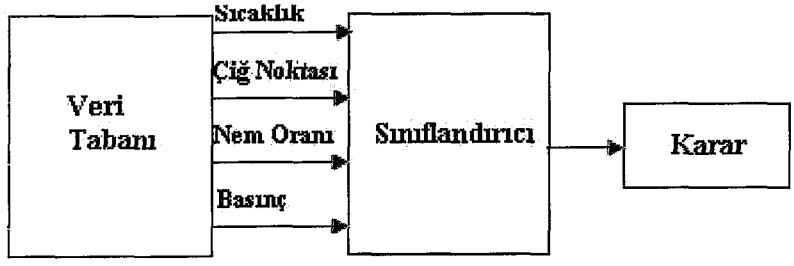
gerçekleştirilmiştir. Böylece insan faktöründen kaynaklanan hatalar minimuma indirgenmiş olur.

Elazığ iline ait hava tahmini gerçekleştirilirken işaret işlemeye tabi tutulan radar işaretlerinin sıcaklık, çiğ noktası, nem oranı, basınç verileri özellikleri çıkarılmış bir şekilde www.wuderground.com adresinden alınmıştır ve bu veriler giriş olarak veri madenciliğindeki akıllı sınıflandırıcılara verilerek, değerlendirilip bir hava tahmininin yapılması gerçekleştirilmiştir.



Şekil 5.1. Örüntü tanıma işlem süreci

Şekil 5.1.'de görüldüğü gibi verilerin özellikleri çıkarılmasından karar aşamasına kadar olan işlem süreci birinci bölümde anlatıldığı üzere örüntü tanıma konusunun içinde yer alır. Tezde kullanılan meteorolojik verilerin özellik çıkarma işlemi yapılmamış özellikleri çıkarılmış bir şekilde www.wunderground.com internet adresinden alınmıştır. Verilerin veri tabanından alınarak yorumlanıp değerlendirilmesi ise Şekil 5.2.' de görüldüğü gibi VM işlem süreci içerisine girer. Şekil 5.1. ve Şekil 5.2.' de de görüldüğü gibi VM işlem süreci örüntü tanıma işlem süreci içerisindedir.



Şekil 5.2. VM işlem süreci

5.2.1. Verilerin Sınıflandırılması

Elazığ iline ait sıcaklık, çiğ noktası, nem oranı ve basınç verileri, veri tabanından alınarak ANFIS (Adaptive Network Based Fuzzy Inference System) ve YSA (Yapay Sinir Ağı) sınıflandırıcılarından ayrı ayrı geçirildikten sonra hem hava tahmininin yapılması hem de bu

sınıflandırıcıların performanslarının değerlendirilmesi sağlanmıştır. 4 sınıftan oluşan 40 tane giriş verisi, toplam veri sayısı 160 olan veriler sınıflandırıcılara verilerek, eğitim sürecinden geçirilmiştir [31]. Eğitim sürecinden geçirilen bu verileri bilgisayarın algılayıp hangi karar sınıfına ait olduğunun yorumlaması sağlanmıştır. Eğitim işlemi tamamlandıktan sonra yaklaşık 4 sınıftan oluşan 100 veri toplam 400 veri test süreci için sınıflandırıcılara verilmiştir. Elde edilen sonuçlar aşağıdadır.

Tablo 5.3. Veri tabanında sınıflandırıcının eğitim sürecinde kullanmak üzere tutulan veriler

S.N.	Tarih	Saat	Sıcaklık	Çiğ noktası	Nem oranı	Basınç	Durum
1-	14.02.2003	11:50	37.40 °F / 3.0 °C	33.80 °F / 1.0 °C	87%	29.77 in / 1008.0 hPa	Hafif Kar
2-	19.02.2003	11:50	30.20 °F / -1.0 °C	28.40 °F / -2.0 °C	93%	29.53 in / 1000.0 hPa	Hafif Kar
3-	20.02.2003	11:50	33.80 °F / 1.0 °C	32.00 °F / 0.0 °C	93%	29.47 in / 998.0 hPa	Hafif Kar
4-	24.02.2003	11:50	33.80 °F / 1.0 °C.	32.00 °F / 0.0 °C	93%	29.80 in / 1009.0 hPa	Kar
5-	04.03.2003	11:50	37.40 °F / 3.0 °C	32.00 °F / 0.0 °C	81%	30.12 in / 1020.0 hPa	Hafif Kar
6-	29.10.2003	11:50	33.8 °F / 1.0 °C	32.0 °F / 0.0 °C	93%	29.71 in / 25.3 hPa	Kar
7-	19.12.2003	11:50	35.6 °F / 2.0 °C	26.6 °F / -3.0 °C	70%	29.92 in / 30.7 hPa	Kar
8-	09.01.2004	11:50	21.2 °F / -6.0 °C	15.8 °F / -9.0 °C	80%	30.04 in / 14.9 hPa	Kar
9-	02.01.2003	11:50	37.40 °F / 3.0 °C	33.80 °F / 1.0 °C	87%	30.04 in / 1017.0 hPa	Yağmur
10-	03.01.2003	11:50	39.20 °F / 4.0 °C	33.80 °F / 1.0 °C	81%	30.06 in / 1018.0 hPa	Yağmur
11-	04.01.2003	10:50	41.00 °F / 5.0 °C	32.00 °F / 0.0 °C	70%	29.95 in / 1014.0 hPa	Yağmur
12-	14.01.2003	11:50	39.20 °F / 4.0 °C	37.40 °F / 3.0 °C	93%	29.98 in / 1015.0 hPa	Yağmur
13-	18.03.2003	11:50	42.80 °F / 6.0 °C	35.60 °F / 2.0 °C	76%	29.74 in / 1007.0 hPa	Yağmur
14-	19.03.2003	11:50	41.00 °F / 5.0 °C	35.60 °F / 2.0 °C	81%	29.59 in / 1002.0 hPa	Yağmur
15-	24.03.2003	11:50	39.20 °F / 4.0 °C	33.80 °F / 1.0 °C	81%	29.74 in / 1007.0 hPa	Yağmur
16-	25.03.2003	11:50	42.80 °F / 6.0 °C	39.20 °F / 4.0 °C	87%	29.68 in / 1005.0 hPa	Yağmur
17-	14.06.2003	11:50	82.40 °F / 28.0 °C	46.40 °F / 8.0 °C	28%	29.89 in / 1012.0 hPa	Açık

Tablo 5.3. Veri tabanında sınıflandırıcının eğitim sürecinde kullanmak üzere tutulan veriler (Devamı)

S.N.	Tarih	Saat	Sıcaklık	Çiğ noktası	Nem oranı	Basınç	Durum
18-	15.06.2003	11:50	86.00 °F / 30.0 °C	48.20 °F / 9.0 °C	27%	29.83 in / 1010.0 hPa	Açık
19-	19.06.2003	11:50	80.60 °F / 27.0 °C	48.20 °F / 9.0 °C	32%	29.83 in / 1010.0 hPa	Açık
20-	20.06.2003	11:50	86.00 °F / 30.0 °C	48.20 °F / 9.0 °C	27%	29.74 in / 1007.0 hPa	Açık
21-	21.06.2003	11:50	84.20 °F / 29.0 °C	37.40 °F / 3.0 °C	19%	29.71 in / 1006.0 hPa	Açık
22-	23.06.2003	11:50	82.40 °F / 28.0 °C	46.40 °F / 8.0 °C	28%	29.89 in / 1012.0 hPa	Açık
23-	24.06.2003	11:50	78.80 °F / 26.0 °C	39.20 °F / 4.0 °C	24%	29.92 in / 1013.0 hPa	Açık
24-	05.08.2003	11:50	93.20 °F / 34.0 °C	44.60 °F / 7.0 °C	19%	29.80 in / 1009.0 hPa	Açık
25-	11.01.2003	10:50	37.40 °F / 3.0 °C	28.40 °F / -2.0 °C	70%	30.39 in / 1029.0 hPa	Kapalı
26-	22.01.2003	10:50	35.60 °F / 2.0 °C	26.60 °F / -3.0 °C	70%	30.12 in / 1020.0 hPa	Kapalı
27-	03.03.2003	11:50	32.00 °F / 0.0 °C	26.60 °F / -3.0 °C	80%	30.18 in / 1022.0 hPa	Kapalı
28-	23.03.2003	11:50	33.80 °F / 1.0 °C	24.80 °F / -4.0 °C	70%	29.98 in / 1015.0 hPa	Kapalı
29-	31.10.2003	11:50	42.8 °F / 6.0 °C	39.2 °F / 4.0 °C	87%	30.18 in / 38.6 hPa	Kapalı
30-	05.01.2004	11:50	44.6 °F / 7.0 °C	28.4 °F / -2.0 °C	53%	29.83 in / 38.1 hPa	Kapalı
31-	07.01.2004	11:50	41.0 °F / 5.0 °C	33.8 °F / 1.0 °C	76%	29.56 in / 34.3 hPa	Kapalı
32-	11.01.2004	11:50	28.4 °F / -2.0 °C	21.2 °F / -6.0 °C	74%	30.15 in / 21.2 hPa	Kapalı
33-	07.01.2003	10:50	46.40 °F / 8.0 °C	35.60 °F / 2.0 °C	66%	30.27 in / 1025.0 hPa	Çoğunlukla Bulutlu
34-	12.01.2003	12:50	39.20 °F / 4.0 °C	32.00 °F / 0.0 °C	75%	30.21 in / 1023.0 hPa	Çoğunlukla Bulutlu
35-	18.01.2003	11:50	37.40 °F / 3.0 °C	24.80 °F / -4.0 °C	60%	30.15 in / 1021.0 hPa	Çoğunlukla Bulutlu
36-	28.01.2003	11:50	44.60 °F / 7.0 °C	24.80 °F / -4.0 °C	46%	29.65 in / 1004.0 hPa	Çoğunlukla Bulutlu
37-	08.02.2003	11:50	44.60 °F / 7.0 °C	33.80 °F / 1.0 °C	66%	29.74 in / 1007.0 hPa	Çoğunlukla Bulutlu
38-	05.04.2003	11:50	62.60 °F / 17.0 °C	41.00 °F / 5.0 °C	45%	29.80 in / 1009.0 hPa	Çoğunlukla Bulutlu
39-	26.10.2003	11:50	68.0 °F / 20.0 °C	37.4 °F / 3.0 °C	33%	29.89 in / 37.4 hPa	Çoğunlukla Bulutlu
40-	07.11.2003	11:50	66.2 °F / 19.0 °C	44.6 °F / 7.0 °C	46%	30.18 in / 44.6 hPa	Çoğunlukla Bulutlu

Şekil 5.3. İşlem sürecinin basit gösterimi

Sıcaklık, çiğ noktası, nem oranı ve basınç verileri sınıflandırıcılardan geçirildikten sonra bilgisayarda değerlendirilen çıkış verileri, Tablo 5.4' de görüldüğü üzere 5 karar sınıfı olarak ifade edilmektedir. Bu karar sınıflarından 0 durumu havanın karlı olacağını, 1 durumu havanın yağmurlu olacağını, 2 durumu havanın açık olacağını, 3 durumu havanın kapalı olacağını ve 4 durumu ise havanın bulutlu olacağını ifade eder.

Tablo 5.4. Karar sınıfları

0	Kar
1	Yağmur
2	Açık
3	Kapalı
4	Bulutlu

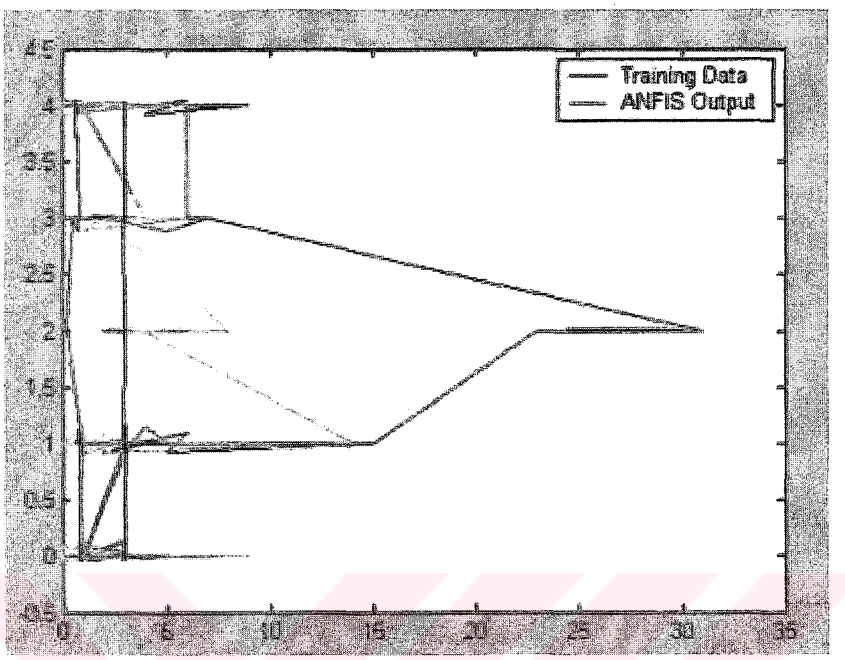
Tablo 5.3. ve Şekil 5.3.'de görüldüğü üzere 40 satır ve 4 sütundan oluşan X giriş verileri eğitim sürecinde sınıflandırıcılardan geçirildikten sonra bilgisayarın bu verileri yorumlayıp hangi karar sınıfına ait olacağının saptanması sağlanmıştır.

Tablo 5.5. Bir günlük meteorolojik veri

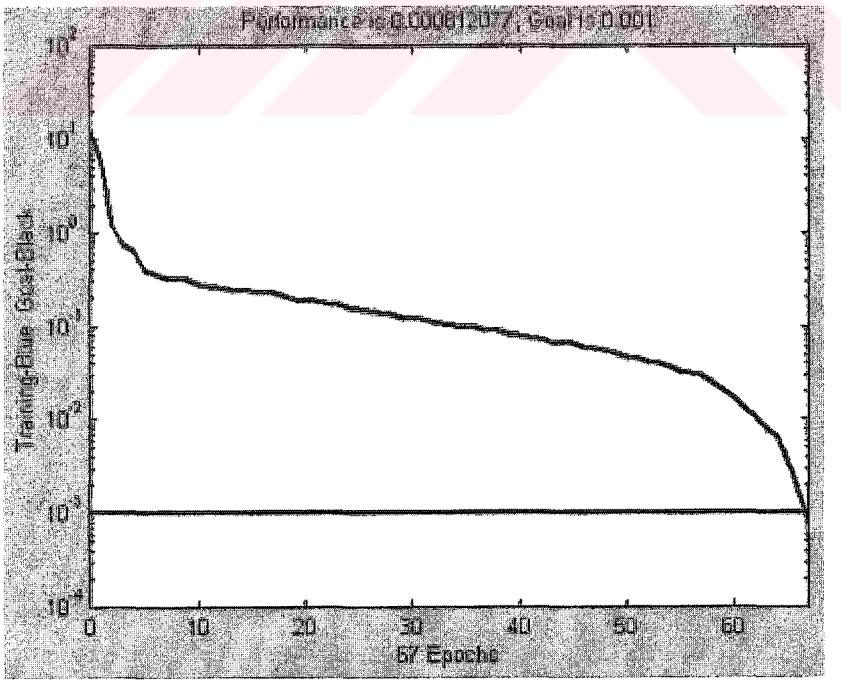
Tarih	**Saat**	**Sıcaklık**	**Çiğ noktası**	**Nem**	**Basınç**	**Durum**
20.02.2003	11:50	3.80 °F / 1.0 °C	32.00 °F / 0.0 °C	93%	29.47 in / 998.0 hPa	Hafif Kar

Tablo 5.5.' deki veriler bilgisayar ortamına aktarılırken, X giriş matrisi için sıcaklık değeri 1, çiğ noktası değeri 0, nem değeri 0.93, basınç 2.947 olarak alınmaktadır. Giriş verileri alınırken bilgisayarın hızlı derleyebilmesi için nem oranı 100'e basınç verileri ise 10'a bölünerek normalizasyona uğratılmaktadır, yani 93 olan nem 0.93 ve 29.47 olan basınç ise 2.947 olarak alınmaktadır. Tablo 5.3.' de görülen bütün veriler için aynı işlem uygulanmaktadır. Tablo 5.5.' deki girişler sonucunda oluşan Y çıkış matrisi hava durumu karlı olduğu için 0 olarak elde edilir. Toplam 400 veri test etme sürecinde sınıflandırıcıdan geçirildikten sonra elde edilen veriler Tablo 5.10.'da topluca verilmiştir. 100 satır ve 1 sütundan oluşan Y çıkışları, yani

karar sınıfları elde edilmektedir. Elde edilen eğitim sürecinin grafikleri Şekil 5.4. ve Şekil 5.5. de görülmektedir.



Şekil 5.4. ANFIS sınıflandırıcısıyla eğitilen giriş verilerinin ve elde edilen çıkışın aynı eksen üzerindeki grafiği



Şekil 5.5. Arzu edilen çıkışla YSA'nın bulmuş olduğu çıkış arasındaki farkın grafiği

Test sürecinde sınıflandırıcılara verilen giriş verileri ve sınıflandırıcılardan elde edilen çıkış verileri ise Tablo 5.6.' daki gibidir.

Tablo 5.6. Test süreci sonucu elde edilen çıkışlar

	Girişler				Çıkışlar			
S.N.	**Sıcaklık (°C)**	**Çiğ Noktası (°C)**	**Nem Oranı (%)**	**Basınç (in)**	**ANFIS**	**YSA**	**İstenen Sonuç**	**Durum**
1-	1	3	0.75	3.004	2.1511	3.0488		
2-	1	2	0.81	3.001	0.9373	1.7807		
3-	0	0	1	2.974	1.5351	0.0337		
4-	0	0	1	2.970	1.5078	0.0312		
5-	0	1	0.93	2.983	0.6379	-0.2088		
6-	0	1	0.93	2.986	0.6693	-0.1970		
7-	0	1	0.93	2.992	0.7880	-0.1725		
8-	0	2	0.87	3.001	-0.2945	0.7801		
9-	3	1	0.87	2.992	0.2671	0.5471		
10-	3	2	0.93	2.992	0.8498	1.8876	0	Kar
11-	1	2	0.86	3.000	0.2318	1.9406		
12-	0	2	0.85	3.001	- 0.2699	0.8561		
13-	1	0	0.93	2.980	-0.0373	0.0270		
14-	3	1	0.87	2.977	0.1309	0.0029		
15-	1	2	0.93	2.953	-0.0002	0.0114		
16-	1	0	0.93	2.947	0.0890	-0.0084		
17-	3	0	0.81	3.012	0.0396	0.0143		
18-	1	0	0.93	2.971	-0.0448	0.0153		
19-	6	9	0.80	3.004	-0.0001	0.0044		
20-	1	1	0.87	3.004	- 0.0087	0.0112		

Tablo 5.6. Test süreci sonucu elde edilen çıkışlar (Devamı)

	Girişler				Çıkışlar			
S.N.	**Sıcaklık (°C)**	**Çiğ Noktası (°C)**	**Nem Oranı (%)**	**Basınç (in)**	**ANFIS**	**YSA**	**İstenen Sonuç**	**Durum**
21-	15	11	0.77	3.006	-0.7720	1.4090		
22-	14	13	0.94	3.009	-1.3413	2.4816		
23-	14	12	0.88	3.012	-1.7070	2.7145		
24-	13	11	0.88	3.009	-1.0956	3.0701		
25-	11	11	1	3.015	- 1.5020	3.4673		
26-	11	10	1	3.015	- 1.4098	3.5883		
27-	12	11	0.94	3.009	-1.0024	3.1283		
28-	11	10	0.94	3.009	-0.8409	3.3462		
29-	13	10	0.82	3.012	-1.3349	3.9412		
30-	13	10	0.82	3.009	-0.9579	3.7596	1	Yağmur
31-	3	2	0.93	2.995	0.0012	2.0085		
32-	2	1	0.93	2.998	0.0014	1.0590		
33-	3	1	0.87	3.004	0.9214	1.0046		
34-	4	1	0.81	3.006	1.1595	1.0035		
35-	5	0	0.70	2.995	1.0049	1.0375		
36-	4	3	0.93	2.998	1.0237	0.9950		
37-	2	1	0.93	2.992	0.9177	1.0097		
38-	6	2	0.76	2.974	1.1048	1.0234		
39-	5	2	0.81	2.959	0.9259	0.9952		
40-	15	14	0.94	2.977	1.0000	1.0067		
41-	3	4	0.60	2.989	0.0162	2.8252	2	Açık
42-	1	4	0.70	2.989	0.0024	2.4979		

Tablo 5.6. Test süreci sonucu elde edilen çıkışlar (Devamı)

	Girişler				Çıkışlar			
S.N.	**Sıcaklık (°C)**	**Çiğ Noktası (°C)**	**Nem Oranı (%)**	**Basınç (in)**	**ANFIS**	**YSA**	**İstenen Sonuç**	**Durum**
43-	6	8	0.86	2.998	0.0970	-1.0918		
44-	7	8	0.93	3.001	0.0959	1.3641		
45-	1	13	0.35	3.006	0.5800	3.0203		
46-	9	16	0.58	3.006	0.3706	2.8868		
47-	2	12	0.47	3.012	1.0425	1.6284		
48-	29	10	0.30	2.968	0.1561	1.5842		
49-	32	10	0.26	2.977	0.1381	1.7317		
50-	23	3	0.27	2.974	- 0.0037	1.9200		
51-	20	1	0.28	2.974	- 0.0100	1.6232	2	Açık
52-	29	10	0.30	2.980	0.1436	1.6310		
53-	23	4	0.29	3.004	1.9998	1.9534		
54-	26	5	0.26	2.998	2.0003	1.9611		
55-	28	6	0.25	2.995	2.0002	1.9734		
56-	26	4	0.24	2.992	1.9997	1.9720		
57-	30	2	0.17	2.974	2.0002	2.1333		
58-	26	5	0.26	2.992	1.9999	1.9550		
59-	28	4	0.21	2.983	1.9998	1.9633		
60-	31	8	0.24	2.998	1.9999	1.9617		
61-	2	0	0.87	2.980	-0.5925	0.0551		
62-	1	0	0.93	2.980	-0.0373	0.0270	3	Kapalı
63-	0	0	1	2.977	1.5783	0.0355		
64-	1	0	0.93	2.977	-0.0493	0.0230		

Tablo 5.6. Test süreci sonucu elde edilen çıkışlar (Devamı)

	Girişler				Çıkışlar			
S.N.	**Sıcaklık (°C)**	**Çiğ Noktası (°C)**	**Nem Oranı (%)**	**Basınç (in)**	**ANFIS**	**YSA**	**İstenen Sonuç**	**Durum**
65-	1	2	0.93	2.977	-0.0528	1.1513		
66-	1	2	0.93	3.009	4.9411	2.6558		
67-	1	3	0.75	3.006	2.0245	3.1603		
68-	1	1	1	3.027	14.4858	0.1468		
69-	2	2	1	3.036	18.5970	3.5654		
70-	1	1	0.87	3.021	- 0.1471	0.1790		
71-	0	2	0.87	3.024	0.8792	1.5906		
72-	2	2	1	3.033	18.5299	3.5399	3	Kapalı
73-	7	2	0.53	2.983	3.0016	3.0065		
74-	5	1	0.76	2.956	2.8896	2.9814		
75-	2	6	0.74	3.015	3.0000	3.0037		
76-	0	3	0.80	3.018	2.9962	3.0145		
77-	3	2	0.70	3.039	2.9968	3.0107		
78-	2	3	0.70	3.012	3.0132	3.0065		
79-	1	4	0.70	2.998	3.0017	3.0037		
80-	6	4	0.87	3.018	2.9979	2.9980		
81-	3	7	0.74	2.998	-27.6783	-0.1362		
82-	4	10	0.64	2.998	50.8717	0.2971		
83-	4	1	0.70	2.980	5.7793	1.0360	4	Bulutlu
84-	4	2	0.65	2.980	12.6581	0.6267		
85-	5	2	0.61	2.980	13.0243	2.2749		
86-	8	3	0.46	2.998	-20.2389	2.9264		

Tablo 5.6. Test süreci sonucu elde edilen çıkışlar (Devamı)

	Girişler				Çıkışlar			
S.N.	**Sıcaklık (°C)**	**Çiğ Noktası (°C)**	**Nem Oranı (%)**	**Basınç (in)**	**ANFIS**	**YSA**	**İstenen Sonuç**	**Durum**
87-	6	1	0.61	2.998	2.0614	4.9696		
88-	2	4	0.86	3.006	-4.4058	2.9711		
89-	2	4	0.86	3.009	1.9775	3.1128		
90-	1	3	0.75	3.004	2.1511	3.0488		
91-	2	3	0.70	3.004	6.6947	2.6155		
92-	4	6	0.86	3.039	-89.2280	3.2827		
93-	6	1	0.70	2.995	3.9547	4.0053	4	Bulutlu
94-	6	3	0.81	3.006	3.9614	3.9997		
95-	8	2	0.66	3.027	3.9974	4.0111		
96-	9	5	0.76	3.018	4.0023	3.9981		
97-	4	0	0.75	3.021	3.9292	4.0036		
98-	6	0	0.66	3.006	4.0343	4.0026		
99-	4	6	0.49	3.001	3.9999	4.0035		
100-	3	4	0.60	3.015	3.9975	4.0025		

5.2.2. Sınıflandırılan Verilerin Değerlendirilmesi ve Sınıflandırıcı Performanslarının Karşılaştırılması

Giriş verileri biri birine çok yakın olduğundan, çıkış değerlerindeki hata oranı yüksek çıkmaktadır. Bu durum istenen çıkış değerlerinin dışında değerlerin çıkmasına neden olmaktadır. Ayrıca bazı değerler negatif çıkmaktadır. Negatif olan bu değerler ihmal edilip pozitif değermiş gibi düşünülmektedir. Tablo 5.7.' de görüldüğü gibi ANFIS ve YSA sınıflandırıcılarının performans değerleri birbirine yakın çıkmıştır. ANFIS %54 oranında, YSA ise %55 oranında bir performans göstermektedir.

Tablo 5.7. Sınıflandırıcıların performansı

ANFIS	YSA
%54 (Kural Tabanı $=2^4$)	%55
%65 (Kural Tabanı $=5^4$)	%55

Ancak ANFIS sınıflandırıcısının kural tabanı (numMFs değeri 5 olarak alınmış, $5^4=625$) 625 olarak alındığında ANFIS sınıflandırıcısının performansının YSA sınıflandırıcıya göre daha yüksek olduğu Tablo 5.7.'de görülmektedir. Kural tabanı arttırıldığında eğitim süresi uzamıştır. ANFIS' in performansının %65 olduğu gözlemlenmiştir.

Tablo 5.8. ANFIS sınıflandırıcısının kural tabanı arttırıldığında elde edilen çıkışlar

	Girişler				**Çıkış**		
S.N.	**Sıcaklık (°C)**	**Çiğ Noktası (°C)**	**Nem Oranı (%)**	**Basınç (in)**	**ANFIS**	**İstenen Sonuç**	**Durum**
1-	1	3	0.75	3.004	2.7506		
2-	1	2	0.81	3.001	0.7360		
3-	0	0	1	2.974	0.0314		
4-	0	0	1	2.970	0.0442		
5-	0	1	0.93	2.983	-0.5315		
6-	0	1	0.93	2.986	-0.5336		
7-	0	1	0.93	2.992	-0.5252	0	Kar
8-	0	2	0.87	3.001	0.1890		
9-	3	1	0.87	2.992	0.0565		
10-	3	2	0.93	2.992	0.8871		
11-	1	2	0.86	3.000	0.4003		
12-	0	2	0.85	3.001	0.4233		
13-	1	0	0.93	2.980	0.0000		

Tablo 5.8. ANFIS sınıflandırıcısının kural tabanı arttırıldığında elde edilen çıkışlar (Devamı)

	Girişler				Çıkış		
S.N.	**Sıcaklık (°C)**	**Çiğ Noktası (°C)**	**Nem Oranı (%)**	**Basınç (in)**	**ANFIS**	**İstenen Sonuç**	**Durum**
14-	3	1	0.87	2.977	0.0000	0	Kar
15-	1	2	0.93	2.953	0.0000		
16-	1	0	0.93	2.947	0.0000		
17-	3	0	0.81	3.012	0.0000		
18-	1	0	0.93	2.971	-0.0000		
19-	6	9	0.80	3.004	0.0000		
20-	1	1	0.87	3.004	0.0000		
21-	15	11	0.77	3.006	-0.0007	1	Yağmur
22-	14	13	0.94	3.009	1.0564		
23-	14	12	0.88	3.012	0.0117		
24-	13	11	0.88	3.009	1.0046		
25-	11	11	1	3.015	-0.0138		
26-	11	10	1	3.015	-0.0130		
27-	12	11	0.94	3.009	-0.0071		
28-	11	10	0.94	3.009	1.0148		
29-	13	10	0.82	3.012	1.0370		
30-	13	10	0.82	3.009	-0.0396		
31-	3	2	0.93	2.995	0.8809		
32-	2	1	0.93	2.998	0.9797		
33-	3	1	0.87	3.004	1.0000		
34-	4	1	0.81	3.006	1.0000		
35-	5	0	0.70	2.995	1.0000		

Tablo 5.8. ANFIS sınıflandırıcısının kural tabanı arttırıldığında elde edilen çıkışlar (Devamı)

	Girişler				Çıkış		
S.N.	**Sıcaklık (°C)**	**Çiğ Noktası (°C)**	**Nem Oranı (%)**	**Basınç (in)**	**ANFIS**	**İstenen Sonuç**	**Durum**
36-	4	3	0.93	2.998	1.0000	1	Yağmur
37-	2	1	0.93	2.992	1.0000		
38-	6	2	0.76	2.974	1.0000		
39-	5	2	0.81	2.959	1.0000		
40-	15	14	0.94	2.977	1.0000		
41-	3	4	0.60	2.989	3.3023	2	Açık
42-	1	4	0.70	2.989	2.0004		
43-	6	8	0.86	2.998	0.2131		
44-	7	8	0.93	3.001	0.0060		
45-	1	13	0.35	3.006	0.0000		
46-	9	16	0.58	3.006	0.0000		
47-	2	12	0.47	3.012	0.0000		
48-	29	10	0.30	2.968	-0.0000		
49-	32	10	0.26	2.977	2.0193		
50-	23	3	0.27	2.974	0.0329		
51-	20	1	0.28	2.974	2.0014		
52-	29	10	0.30	2.980	2.0365		
53-	23	4	0.29	3.004	2.0000		
54-	26	5	0.26	2.998	2.0000		
55-	28	6	0.25	2.995	2.0000		
56-	26	4	0.24	2.992	2.0000		
57-	30	2	0.17	2.974	2.0000		

Tablo 5.8. ANFIS sınıflandırıcısının kural tabanı arttırıldığında elde edilen çıkışlar (Devamı)

	Girişler				Çıkış		
S.N.	**Sıcaklık (°C)**	**Çiğ Noktası (°C)**	**Nem Oranı (%)**	**Basınç (in)**	**ANFIS**	**İstenen Sonuç**	**Durum**
58-	26	5	0.26	2.992	2.0000	2	Açık
59-	28	4	0.21	2.983	2.0000		
60-	31	8	0.24	2.998	2.0000		
61-	2	0	0.87	2.980	0.0212	3	Kapalı
62-	1	0	0.93	2.980	0.0000		
63-	0	0	1	2.977	-0.0398		
64-	1	0	0.93	2.977	0.0001		
65-	1	2	0.93	2.977	3.0076		
66-	1	2	0.93	3.009	3.2914		
67-	1	3	0.75	3.006	2.8185		
68-	1	1	1	3.027	2.5554		
69-	2	2	1	3.036	0.0148		
70-	1	1	0.87	3.021	2.9758		
71-	0	2	0.87	3.024	0.3993		
72-	2	2	1	3.033	0.0493		
73-	7	2	0.53	2.983	3.0000		
74-	5	1	0.76	2.956	3.0000		
75-	2	6	0.74	3.015	3.0000		
76-	0	3	0.80	3.018	3.0000		
77-	3	2	0.70	3.039	3.0000		
78-	2	3	0.70	3.012	3.0000		
79-	1	4	0.70	2.998	3.0000		

Tablo 5.8. ANFIS sınıflandırıcısının kural tabanı arttırıldığında elde edilen çıkışlar (Devamı)

	Girişler				Çıkış		
S.N.	**Sıcaklık (°C)**	**Çiğ Noktası (°C)**	**Nem Oranı (%)**	**Basınç (in)**	**ANFIS**	**İstenen Sonuç**	**Durum**
80-	6	4	0.87	3.018	3.0000	3	Kapalı
81-	3	7	0.74	2.998	3.9501		
82-	4	10	0.64	2.998	-0.0698		
83-	4	1	0.70	2.980	-1.2486		
84-	4	2	0.65	2.980	1.4519		
85-	5	2	0.61	2.980	3.7398		
86-	8	3	0.46	2.998	1.8489		
87-	6	1	0.61	2.998	3.6041		
88-	2	4	0.86	3.006	1.0400		
89-	2	4	0.86	3.009	1.0542		
90-	1	3	0.75	3.004	3.7506	4	Bulutlu
91-	2	3	0.70	3.004	3.7772		
92-	4	6	0.86	3.039	0.0434		
93-	6	1	0.70	2.995	4.0000		
94-	6	3	0.81	3.006	4.0000		
95-	8	2	0.66	3.027	4.0000		
96-	9	5	0.76	3.018	4.0000		
97-	4	0	0.75	3.021	4.0000		
98-	6	0	0.66	3.006	4.0000		
99-	4	6	0.49	3.001	4.0000		
100-	3	4	0.60	3.015	4.0000		

ANFIS ile YSA' yı karşılaştırdığımızda YSA' nın ANFIS' e göre daha az bir işlem yükü olduğunu ve hızlı cevap verebildiğini görmekteyiz. Fakat YSA' nın bu avantajlarına karşılık ANFIS' in işlem yükü fazla olmasına rağmen genelleme yeteneği ve hatanın fazla düşürülmesi yönünden daha iyi sonuçlar verdiği görülmüştür.

Tezin bu bölümünde meteorolojik verileri sınıflandırmak için ANFIS ve YSA tarzı sınıflandırıcılar kullanılmış ve bu sınıflandırıcıların performansları karşılaştırılmıştır.

6. SONUÇLAR VE ÖNERİLER

6.1. Sonuçlar ve Tartışma

Bu tez çalışmasında, meteorolojik hava tahmini uygulaması gerçekleştirilmiştir. Bu uygulama için 40 günlük meteorolojik veriler alınmıştır. Alınan sıcaklık, çiğ noktası, nem, basınç gibi meteorolojik veriler toplanarak geniş bir veri tabanı oluşturulmuştur. Oluşturulan veri tabanındaki sıcaklık, çiğ noktası, nem ve basınç verileri normalizasyon işleminden geçirilmiştir. Normalize edilen sıcaklık, çiğ noktası, nem, basınç değerleri YSA ve ANFIS sınıflandırıcılarına giriş değeri olarak verilmiştir. Sınıflandırıcıdan doğru çıkış alabilmek için girişlerin eğitilerek bilgisayarın öğrenmesi sağlanmıştır. Böylece sonraki günün hava tahmini olayı gerçekleştirilmiştir. Hata tahmini aralığını azaltabilmek için ANFIS sınıflandırıcısının kural tabanı sayısı 16'dan 625' e çıkarılmıştır. Böylece hata oranının azaldığı görülmüştür. Bu durum Tablo 5.7.'de görülmektedir.

14.02.2003 – 09.02.2004 tarihleri arasında rasgele alınan 100 günlük veri kullanılarak hava tahmini yapılmıştır. Gerçekleştirilen bu hava tahmininde ANFIS sınıflandırıcısının kural tabanı 16 iken 54 gününün hava tahmini doğru olarak bulunmuştur. ANFIS sınıflandırıcısının kural tabanı 625'e çıkarıldığında, 65 günün hava tahmini doğru olarak bulunmuştur. YSA sınıflandırıcısı ile yapılan tahminde ise 55 günün hava durumu doğru olarak tahmin edilmiştir.

Bu çalışmada kullanılan yöntemlerin ışığında ve Matlab ortamında geliştiren program sayesinde, hava tahminlerinde %50'nin üzerinde çok daha doğru tahminlerin yapılabileceği Bölüm 5.'de ayrıntılı bir şekilde anlatılmıştır. Yukarıda belirtilen sonuçlardan hareket ederek, ANFIS sınıflandırıcısının kural tabanının 625'e çıkarılarak, %65 gibi doğru hava tahminlerinin yapılabileceği, bu çalışma ile ispatlanmıştır. Bu amaçla geliştirilen veri madenciliği yazılımı kullanılarak, bundan böyle Elazığ için hava tahminleri yapılabilecektir.

6.2. Öneriler

Bu tez çalışması ile hava tahmini yapabilmek için, yeni bir veri madenciliği yazılımı geliştirilmiştir. Ancak, hava tahminlerinde çok daha iyi sonuçlar elde edebilmek için;

- Daha farklı özelikler alınabilir yani giriş sayısı arttırılabilir.
- Daha çok ve farklı veri kullanılabilir.
- İlerideki çalışmalarda daha iyi hava tahmini yapabilecek farklı bir sınıflandırıcı tasarlanabilir.

- Verilerin on-line bir şekilde doğrudan veri tabanına aktarılarak ilgili yazılımda anında değerlendirilebilir.

İleri düzeyde sürdürülecek bir çalışma ile daha başarılı sonuçların elde edilebileceğini ve daha mükemmel yazılımların geliştirilebileceğini öngörmekteyiz.

KAYNAKLAR

[1] Varol, A., Daş, R., 2005, Comparison of Video Conference Application Performed Through Two Different Methods Using Frame Relay Line, Journal of Polytechnic, Volume 8, Number 1, pp.1-10.

[2] Frawley, W. J., Piatetsky-Shapiro, G., Matheus, C. J., 1991, "Knowledge discovery databases: An overview", Knowledge Discovery in Databases (G. Piatetsky-Shapiro and W. J. Frawley, eds.), 1-27, Cambridge, MA: AAAI/MIT.

[3] Türkoğlu, İ., 1996, Yapay Sinir Ağları İle Nesne Tanıma. Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ, 112 s.

[4] Öğer, S.G., Eylül, 2003, Osteoporoz Hastalığının Tanısı İçin Veri Madenciliği Kullanımı, Yüksek Lisans Tezi, F. Ü. Fen Bilimleri Enstitüsü.

[5] Türkoğlu, İ., 2002, Durağan Olmayan İşaretler İçin Zaman-Frekans Entropilerine Dayalı Akıllı Örüntü Tanıma , Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü.

[6] Türkoğlu, İ., 1999, Örüntü Tanıma Yöntemlerinin İncelenmesi, Doktora Semineri, Fırat Üniversitesi Fen Bilimleri Enstitüsü.

[7] Han, J., Kamber, Micheline, 2001, Data Mining Concepts and Techniques, Academic Press, Newyork, p.500.

[8] Fayyad, P. S. U. M, Piatetsky-Shapiro, G., Uthurusamy, R., 1996, "Advances in knowledge discovery and data mining", Cambridge, MA: MIT Press.

[9] Varol, A., Varol, N., 21-23 Mayıs 1998, Uzman Sistem Hazırlanırken Hangi Kriterler Göz Önünde Bulundurulmalı, GAP 2. Mühendislik Kongresi, Bildiri Kitabı, Şanlıurfa, s. 559-566.

[10] Berry, Micheal J. A., Gordon, Linoff, 1997, Data Mining Techniques, Wiley Computer Press, Newyork, p.454.

[11] Marakas, George M., 2003, Modern Data Warehousing Mining and Vizulation, Prentice Hall Press, New Jersey, p.274.

[12] Todman, C., 2003, Designing a Data Warehouse, Prentice Hall Press, New Jersey, p.323.

[13] Varol, A., 2000, Robotik, Meb Yayınları, İstanbul.

[14] Oğuz, B., 2000, "Eşleştirme haznelemesinin biçimsel kavram analizi ile modellenmesi", Hacettepe Ünv., Yüksek Lisans Tezi.

[15] Agrawal, R. and Shafer, J.C., 1996, "Parallel mining of association rules: Design, Implementation and Experience", IBM Research Report RJ 10004.

[16] Chen, M. S., Han, J., Yu, P. S., 1996, "Data mining: An overview from a database perspective", IEEE Transactions on Knowledge and Data Engineering, 8(6):866-883.

[17] Chen, Z., 2001, Data Mining and Uncertain Reasoning, Wiley Computer Press, Newyork, 454p.

[18] Akpınar, H., 2000, "Veri tabanlarında bilgi keşfi ve veri madenciliği", İ.Ü. İşletme Fakültesi Dergisi, C.29, 1-22.

[19] Quinlan, J. R., 1986a, "The effect of noise on concept learning. In Machine Learning: In An Artificial Intelligence Aproach", R. Michalski, J. Carbonell, and T. Mitchell, (eds.), vol. 2, 149-166, SanMateo, CA: Morgan Kauffmann Inc.

[20] Lee, S. K., 1992, "An extended relational database model for uncertain and imprecise information", Proc. of the ISth VLDB conference, (Vancouver, British Columbia, Canada), p.211-218.

[21] Luba, T. and Lasocki, R., 1994, "On unknown attribute values in functional dependencies", Proc. of the International Workshop on Rough Sets and Soft Computing, (San Jose, CA), p.490-497.

[22] Doğan, Ş., 2005, Veri Madenciliği ve Biyoinformatik Uygulamaları, Yüksek Lisans Semineri, Fırat Üniversitesi Fen Bilimleri Enstitüsü.

[23] Avcı, E., 2005, Akıllı Radar ile Hedef Tanıma Sistemi , Doktora Tezi, F.Ü. Fen Bilimleri Enstitüsü.

[24] Şengür, A., 2003, Yapay sinir ağları ile analog modülasyonlu haberleşme işaretlerinin sınıflandırılması, Yüksek lisans tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü.

[25] Elmas, Ç., 2003, Yapay sinir ağları, Seçkin yayınları, Ankara, 192 s.

[26] Türkoğlu, İ.,Şengür, A., Toraman, S., 2003, Tıbbi Görüntülerden İstenen Bir Örüntünün Ayrıştırılması, DAUM, 3, 97-101.

[27] Avcı, E., Türkoğlu İ., 2003, "Tünel Diyodun Uyarlamalı Ağ Tabanlı Bulanık Çıkarım ile Modellenmesi", XII. International Twelfth Turkish Symposium on Artificial Intelligence and Neural Network, Çanakkale , Cilt T-1, s. 39-41.

[28] Kittisak Kerdprasop and Nittaya Kerdprasop, 2002, The Performance Of Learning Algorithms On Reduced Data Sets, Tokyo, Japan, Proceeding of the IASTED International Conference Artificial and Computational Intelligence.

[29] R. Gunnalan and T, Menzies and K. Appukuttyand A, Srinivasan and Y. Hu, 2003, Feature Subset Selection with TAR2less.

[30] Peters, J., F., Suraj, Z., Shan, S., Ramanna, S., Pedrycz, W., Pizzi, N., 2003, Classification of meteorological volumetric radar data using rough set methods, Pattern recognition letters, Elsevier sience, p.911-920.

[31] www.wunderground.com

ÖZGEÇMİŞ

Ömer Osman DURSUN

Ahmet Kabaklı
Anadolu Öğretmen Lisesi
23000, Elazığ

Tel: 424-2476792
E.posta : omerdursun23@yahoo.com

1979	Elazığ'da doğdu.
1985–1990	Atatürk İlkokulunu tamamladı.
1990–1993	Elazığ Ortaokulunu bitirdi.
1993–1997	Balakgazi Lisesini bitirdi.
1998–2002	Fırat Üniversitesi T.E.F. Elektronik ve Bilgisayar Eğitimi Bölümünden mezun oldu.
2002–2004	M.E.B. Elazığ Vali Tevfik Gür İlköğretim Okulunda öğretmenlik yaptı.
2005-	M.E.B. Elazığ Ahmet Kabaklı Anadolu Öğretmen Lisesinde bilgisayar öğretmeni olarak görev yapmaktadır.

EK-I

MATLAB Program Kodları

ANFIS sınıflandırıcısının eğitim programının kodları

```
x=[1 0 0.93 2.980;3 1 0.87 2.977;1 2 0.93 2.953;1 0 0.93 2.947;3 0 0.81 3.012;
1 0 0.93 2.971;6 9 0.80 3.004;1 1 0.87 3.004;3 1 0.87 3.004;4 1 0.81 3.006
5 0 0.70 2.995;4 3 0.93 2.998;2 1 0.93 2.992;6 2 0.76 2.974;5 2 0.81 2.959
15 14 0.94 2.977;23 4 0.29 3.004;26 5 0.26 2.998;28 6 0.25 2.995;26 4 0.24 2.992
30 2 0.17 2.974;26 5 0.26 2.992;28 4 0.21 2.983;31 8 0.24 2.998;7 2 0.53 2.983
5 1 0.76 2.956;2 6 0.74 3.015;0 3 0.80 3.018;3 2 0.70 3.039;2 3 0.70 3.012
1 4 0.70 2.998;6 4 0.87 3.018;6 1 0.70 2.995;6 3 0.81 3.006;8 2 0.66 3.027
9 5 0.76 3.018;4 0 0.75 3.021;6 0 0.66 3.006;4 6 0.49 3.001;3 4 0.60 3.015];
y=[0;0;0;0;0;0;0;0;0;1;1;1;1;1;1;1;1;2;2;2;2;2;2;2;2;3;3;3;3;3;3;3;3;4;4;4;4;4;4;4;4
trnData = [x y];
numMFs = 2;
mfType = 'gauss2mf';
epoch_n = 1000;
in_fismat = genfis1(trnData,numMFs,mfType);
out_fismat = anfis(trnData,in_fismat,1000);
Plot(x,y,x,evalfis(x,out_fismat));
legend('Training Data','ANFIS Output');
save rsomer.mat out_fismat
```

ANFIS sınıflandırıcısının test programının kodları

```
load rsomer.mat out_fismat
x=[6 1 0.70 2.995
6 3 0.81 3.006
8 2 0.66 3.027
9 5 0.76 3.018
4 0 0.75 3.021
6 0 0.66 3.006
4 6 0.49 3.001
3 4 0.60 3.015];
y=evalfis(x,out_fismat);
```

YSA sınıflandırıcısının eğitim programının kodları

```
clc;
P=[1 0 0.93 2.980;3 1 0.87 2.977;1 2 0.93 2.953;1 0 0.93 2.9473 0 0.81 3.012
1 0 0.93 2.971;6 9 0.80 3.004;1 1 0.87 3.004;3 1 0.87 3.004;4 1 0.81 3.006
5 0 0.70 2.995;4 3 0.93 2.998;2 1 0.93 2.992;6 2 0.76 2.974;5 2 0.81 2.959
15 14 0.94 2.977;23 4 0.29 3.004;26 5 0.26 2.998;28 6 0.25 2.995;26 4 0.24 2.992
30 2 0.17 2.974;26 5 0.26 2.992;28 4 0.21 2.983;31 8 0.24 2.998;7 2 0.53 2.983
5 1 0.76 2.956;2 6 0.74 3.015;0 3 0.80 3.018;3 2 0.70 3.039;2 3 0.70 3.012;1 4 0.70 2.998
6 4 0.87 3.018;6 1 0.70 2.995;6 3 0.81 3.006;8 2 0.66 3.027;9 5 0.76 3.018;4 0 0.75 3.021
6 0 0.66 3.006;4 6 0.49 3.001;3 4 0.60 3.015]';
T=[0;0;0;0;0;0;0;0;1;1;1;1;1;1;1;1;2;2;2;2;2;2;2;2;3;3;3;3;3;3;3;3;4;4;4;4;4;4;4;4]';
net = newff([0 31;0 14;0.17 0.94;2.8 3.1],[40 25 1],{'tansig' 'tansig' 'purelin'},'trainlm');
net.inputs{1}.size=4;
net.trainParam.epochs=1000;
net.trainParam.goal=1e-3;
net.trainParam.lr=0.9;
net = train(net,P,T);
save omerpsrr1.mat net;
```

YSA sınıflandırıcısının test programının kodları

```
load omerpsrr1.mat net;
P=[3    7    0.74    2.998;4 10    0.64    2.998;4 1    0.70    2.980
4    2    0.65    2.980;5 2    0.61    2.980;8 3    0.46    2.998
6    1    0.61    2.998;2 4    0.86    3.006;2 4    0.86    3.009
1    3    0.75    3.004;2 3    0.70    3.004;4 6    0.86    3.039
6 1 0.70 2.995;6 3 0.81 3.006;8 2 0.66 3.027;9 5 0.76 3.018;4 0 0.75 3.021
6 0 0.66 3.006;4 6 0.49 3.001;3 4 0.60 3.015]';
y=sim(net,P)
```