



**EGE ÜNİVERSİTESİ**

**YÜKSEK LİSANS TEZİ**

**TELEKOMÜNİKASYON SİSTEMLERİNDE**

**SAHTEKÂRLIK TESPİTİ**

**VE YÖNETİMİ**

**Cansu SEZEN**

**Tez Danışmanı: Prof. Dr. Mehmet Emin DALKILIÇ**

**Uluslararası Bilgisayar Anabilim Dalı**

**Bilim Dalı Kodu: 619.02.04**

**Sunuş Tarihi: 11.08.2011**

**E. Ü. FEN BİLİMLERİ ENSTİTÜSÜ**

**Bornova-İZMİR**

**2011**



**EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**

**(YÜKSEK LİSANS TEZİ)**

**TELEKOMÜNİKASYON SİSTEMLERİNDE**

**SAHTEKÂRLIK TESPİTİ**

**VE YÖNETİMİ**

**Cansu SEZEN**

**Tez Danışmanı: Prof. Dr. Mehmet Emin DALKILIÇ**

**Uluslararası Bilgisayar Anabilim Dalı**

**Bilim Dalı Kodu: 619.02.04**

**Sunuş Tarihi: 11.08.2011**

**Bornova-İZMİR**

**2011**



Cansu Sezen tarafından yüksek lisans tezi olarak sunulan “Telekomünikasyon Sistemlerinde Sahtekârlık Tespiti ve Yönetimi” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 11.08.2011 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

**Jüri Üyeleri:**

**İmza**

**Jüri Başkanı** : Prof. Dr. Mehmet Emin DALKILIÇ

.....

**Raportör Üye** : Yrd. Doç. Dr. Muhammed CİNSDİKİCİ

.....

**Üye** : Doç. Dr. Aybars UĞUR

.....



## ÖZET

# TELEKOMÜNİKASYON SİSTEMLERİNDE SAHTEKÂRLIK TESPİTİ VE YÖNETİMİ

SEZEN, Cansu

Yüksek Lisans Tezi, Uluslararası Bilgisayar Enstitüsü  
Tez Yöneticisi: Prof. Dr. Mehmet Emin DALKILIÇ  
Ağustos 2011, 104 sayfa

Günümüzde yapay sinir ağları, sağladıkları başarılı sonuçlar nedeniyle birçok problemde geleneksel yöntemlere alternatif olarak tercih edilmeye başlamıştır. Bu çalışmada, telekomünikasyon şirketlerinde zarara yol açan sahtekârlık amaçlı kullanımların, arama detay kayıtlarından elde edilen veriler kullanılarak yapay sinir ağları yaklaşımları ile tespit edilmesi gerçekleştirilmiştir.

Daha yüksek oranlarda tespit başarısının elde edilmesi için sistem dâhilinde farklı işleyişlerdeki çıkarsama mekanizmaları tarafınan üretilen çıktıların göz önüne alınması amacıyla üç farklı yapay sinir ağı yaklaşımı birlikte kullanılarak bir sahtekârlık tespit sistemi geliştirilmiştir. Bu bileşik modelde Çok Katmanlı Algılayıcı (MLP: Multilayer Perceptron), Öğrenen Vektör Nicelendirmesi (LVQ: Learning Vector Quantization), Özörgütlemeli Harita (SOM: Self Organizing Map) ağ modelleri birlikte çalışmakta ve verdikleri sonuçlar arasından oyçokluğuyla karara varılmaktadır.

Yapılan testler sonucunda üç farklı yapay sinir ağı yaklaşımının birlikte kullanıldığı durumda elde edilen başarı oranının her bir yaklaşımın tek başına kullanıldığında sağladığı performanstan yüksek olduğu görülmüştür.

**Anahtar sözcükler:** Sahtekârlık tespiti, yapay sinir ağları, Telekomünikasyon sahtekârlığı



**ABSTRACT****FRAUD DETECTION AND MANAGEMENT ON  
TELECOMMUNICATION SYSTEMS**

SEZEN, Cansu

MSc in International Computer Institute

Supervisor: Prof. Dr. Mehmet Emin DALKILIÇ

August 2011, 104 pages

Nowadays neural networks has started being preferred as an alternative of traditional techniques due to the successful results they provide. In this study, detection of fraudulent use which causes harm for the telecommunication companies has been implemented with neural network approaches using data obtained from call detail records.

For higher detection success rates, to consider the outputs of the interference mechanisms with different treatments, a fraud detection system has been developed using three different neural network approaches together. In this combined model, Multilayer Perceptron (MLP), Learning Vector Quantization (LVQ) and Self Organizing Map (SOM) network models work together and the decision is obtained from their results using majority of votes.

As a result of the tests, it has been obtained that the success rate of the combined model was higher than the success rates of each net models.

**Keywords:** Fraud management, Artificial neural networks, Telecommunication fraud



## **TEŐEKKÖR**

Bana bu yüksek lisans tezi konusunda alıŐma imkânı tanıyan ve alıŐmalarımnda her koŐulda destek olan deęerli tez danıŐmanım Prof. Dr. Mehmet Emin Dalkılı'a ve eęitim hayatım boyunca benden desteklerini esirgemeyen aileme ve arkadaŐlarımna sonsuz teŐekkÖrler.



## İÇİNDEKİLER

### Sayfa

ÖZET .....	v
ABSTRACT .....	vii
TEŞEKKÜR .....	ix
ŞEKİLLER DİZİNİ .....	xix
ÇİZELGELER DİZİNİ .....	xxi
KISALTMALAR DİZİNİ .....	xxiii
1. GİRİŞ .....	1
2. TELEKOMÜNİKASYONDA SAHTEKÂRLIK YÖNETİMİNE GENEL BAKIŞ .....	3
2.1 Telekomünikasyona İlişkin Temel Kavramlar .....	3
2.1.1 Arama detayı kayıtları (CDR) .....	3
2.1.2 Yerel santral taşıyıcıları .....	7
2.1.3 Arabağlantı .....	7
2.1.4 Uluslararası dolaşım ortakları .....	8
2.2 Sahtekârlığa İlişkin Temel Kavramlar .....	8
2.2.1 Sahtekârlık yönetimi .....	8
2.2.2 Sahtekârlık önleme .....	8

**İÇİNDEKİLER (devam)**

	<u>Sayfa</u>
2.2.3 Sahtekârlık tespiti .....	8
2.2.4 Sahtekârlık yakalama .....	9
2.2.5 Sahtekârlık tekrarını engelleme .....	9
2.3 Telekomünikasyon Şirketlerine Yönelik Sahtekârlık .....	9
2.3.1 Rakip şirketler tarafından uygulanan sahtekârlıklar .....	10
2.3.2 Kullanıcılar tarafından uygulanan sahtekârlıklar .....	10
2.3.3 Üçüncü partiler tarafından uygulanan sahtekârlıklar .....	10
3. SAHTEKÂRLIK TESPİTİNDE TEMEL YAKLAŞIMLAR .....	12
3.1 Kural Tabanlı Sahtekârlık Tespiti .....	12
3.2 Çok Etmenli Sistemler ile Sahtekârlık Tespiti .....	13
3.3 Görsel Veri Madenciliği Tabanlı Sahtekârlık Tespiti .....	14
3.4. Markov Zincirleri ile Sahtekârlık Tespiti .....	15
3.5 Yapay Sinir Ağları Tabanlı Sahtekârlık Tespiti .....	15
3.6 Melez Sistemlerle Sahtekârlık Tespiti .....	17
4. YAPAY SİNİR AĞLARI .....	21
4.1 Yapay Zekâ .....	21
4.2 Yapay Sinir Ağları .....	24

**İÇİNDEKİLER (devam)**Sayfa

4.2.1 Yapay sinir ağlarının genel özellikleri .....	24
4.2.2 Yapay sinir ağlarının uygulama alanları .....	27
4.2.3 Yapay sinir ağı yapıları .....	28
4.2.4 Yapay sinir ağı algoritmaları .....	29
4.2.5 Yapay sinir ağlarında öğrenme kuralları .....	30
4.3 Yapay Sinir Ağlarının Yapısı ve Temel Elemanları .....	31
4.3.1 Biyolojik sinir hücresinin yapısı .....	32
4.3.2 Yapay sinir hücresinin yapısı .....	33
4.3.3 Yapay sinir ağının yapısı .....	36
4.3.4 Yapay sinir ağlarında öğrenme ve test .....	37
4.4 Yapay Sinir Ağı Modelleri .....	38
4.4.1 Çok Katmanlı Algılayıcı (MLP) .....	38
4.4.2 Öğrenen Vektör Nicelendirmesi (LVQ) .....	42
4.4.3 Özörgütlemeli Haritalar (SOM) .....	44
4.4.4 Bileşik ağ modeli .....	46
4.5 Yapay Sinir Ağı Modeli Geliştirme Süreci .....	46
4.5.1 Örneklerin seçilmesi .....	46

**İÇİNDEKİLER (devam)**

	<u>Sayfa</u>
4.5.2 Eğitim ve test setlerinin belirlenmesi .....	47
4.5.3 Girdi ve çıktıların sayısal gösterimi .....	47
4.5.4 Ağırlıkların başlangıç değerlerinin belirlenmesi .....	47
4.5.5 Katsayıların belirlenmesi .....	48
4.5.6 Örneklerin ağa sunuluş şekli .....	49
4.5.7 Ağırlıkların değiştirilme zamanı .....	49
4.5.8 Girdi ve çıktıların ölçeklendirilmesi .....	50
4.5.9 Durdurma kriterleri .....	50
4.5.10 Ağın topolojisinin (ara katman ve nöron aayısının) belirlenmesi .....	51
<b>5. GELİŞTİRİLEN UYGULAMA .....</b>	<b>52</b>
5.1 Veri İşleme Arayüzü .....	52
5.1.1 Veri setinin yüklenmesi .....	54
5.1.2 Veri setinin ölçeklendirilmesi .....	55
5.1.3 Veri setinin karıştırılması .....	56
5.1.4 Veri setinin bölünmesi .....	57
5.2 Ağ Arayüzü .....	57
5.2.1 Eğitim veri setinin yüklenmesi .....	59

**İÇİNDEKİLER (devam)**Sayfa

5.2.2 Ağın tasarımının yapılması ve eğitilmesi .....	59
5.2.3 Eğitilmiş ağın yüklenmesi .....	64
5.2.4 Test veri setinin yüklenmesi .....	65
5.2.5 Ağın test edilmesi .....	66
5.3 Birleştirme Arayüzü .....	66
5.3.1 Ağların yüklenmesi .....	67
5.3.2 Ağların birleştirilmesi .....	68
5.4 Görsel Simülasyon .....	69
5.4.1 Ağ tasarım simülasyonu .....	69
5.4.2 Ağ birleştirme simülasyonu .....	71
6. TELEKOMÜNİKASYONDA YAPAY SİNİR AĞLARI İLE SAHTEKÂRLIK TESPİTİ .....	72
6.1 Araştırmanın Amacı, Kapsamı ve Yöntemi .....	72
6.1.1 Araştırmanın amacı .....	72
6.1.2 Araştırmanın kapsamı .....	72
6.1.3 Araştırmanın yöntemi .....	72
6.2 Veri Setinin Hazırlanması .....	73
6.2.1 Örneklerin toplanması .....	73

**İÇİNDEKİLER (devam)**

	<u>Sayfa</u>
6.2.3 Girdi ve çıktılarının sayısal gösterimi .....	75
6.2.4 Girdi ve çıktılarının ölçeklendirilmesi .....	76
6.2.5 Eğitim ve test setlerinin belirlenmesi .....	77
6.3 Ağ Tasarımlarının Yapılması (Deney 1) .....	77
6.3.1 Ağın topolojisinin belirlenmesi .....	77
6.3.2 Katsayıların belirlenmesi .....	77
6.4 Ağların Eğitimi ve Test Edilmesi .....	78
6.4.1 MLP ağının eğitimi ve test edilmesi .....	78
6.4.2 LVQ Ağının eğitimi ve test edilmesi .....	81
6.4.3 SOM ağının eğitimi ve test edilmesi .....	81
6.5 Ağların Birleştirilmesi .....	82
6.6 Ağların Farklı Veri Setleri için Uyarlanması (Deney 2) .....	84
6.6.1 MLP ağının eğitimi ve test edilmesi .....	85
6.6.2 LVQ ağının eğitimi ve test edilmesi .....	85
6.6.3 SOM ağının eğitimi ve test edilmesi .....	86
6.6.4 Ağların birleştirilmesi .....	86
6.7 Aynı Eğitim Seti ile Eğitilen Ağların Farklı Test Setleri ile Test Edilmesi (Deney 3) .....	91

**İÇİNDEKİLER (devam)**Sayfa

6.7.1 MLP ağının eğitimi ve test edilmesi.....	92
6.7.2 LVQ ağının eğitimi ve test edilmesi.....	92
6.7.3 SOM ağının eğitimi ve test edilmesi .....	93
6.7.4 Ağların birleştirilmesi.....	94
6.8 Farklı Eğitim Setleri ile Eğitilen Ağların Aynı Test Seti ile Test Edilmesi (Deney 4) .....	95
6.8.1 MLP ağının eğitimi ve test edilmesi.....	95
6.8.2 LVQ ağının eğitimi ve test edilmesi.....	96
6.8.3 SOM ağının eğitimi ve test edilmesi .....	96
6.8.4 Ağların birleştirilmesi.....	97
7. SONUÇ.....	98



## ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
3.1 Kural tabanlı sahtekârlık yönetim sistemi mimarisi (Fawcett ve Provost, 1997) .....	12
3.2 İki aşamalı kural keşfi mimarisi (Rosset vd, 1999) .....	13
3.3 Kişisel iletişim ağlarında sahtekârlık için çok etmenli sistem yapısı (Abu-Hakima vd, 1997) .....	14
3.4 Görsel veri madenciliğine dayalı arama gösterimi (Cox vd, 1997) .....	15
3.5 İleri beslemeli yapay sinir ağı ROC grafiği (Taniguchi vd, 1998) .....	18
3.6 Gauss nodeli ROC grafiği (Taniguchi vd, 1998) .....	18
3.7 Bayes ağı ROC grafiği (Taniguchi vd, 1998) .....	19
4.1 Bir yapay sinir ağının üç temel bileşeni. (Haykin, 1999) .....	22
4.2 Basit makine öğrenmesi modeli. (Haykin, 1999) .....	23
4.3 Basit bir ileri beslemeli ağ örneği (Stergiou ve Siganos, 1996) .....	28
4.4 Bir geri beslemeli ağ örneği (Stergiou ve Siganos, 1996) .....	29
4.5 Biyolojik sinir hücresi (Uğur, 2005) .....	32
4.6 Yapay sinir hücresi .....	33
5.1 Veri işleme arayüzü .....	53
5.2 Küçük bir örnek veri setinin yüklenmesi .....	54
5.3 Küçük bir örnek veri setinin ölçeklendirilmesi .....	55

**ŞEKİLLER DİZİNİ (devam)**

<u>Şekil</u>	<u>Sayfa</u>
5.4 Küçük bir örnek veri setinin karıştırılması .....	56
5.5 Küçük bir örnek veri setinin bölünmesi .....	57
5.6 Ağ arayüzü .....	58
5.7 Küçük bir örnek eğitim veri setinin yüklenmesi .....	59
5.8 MLP ağının tasarımı .....	60
5.9 LVQ ağının tasarımı .....	61
5.10 SOM ağının tasarımı .....	62
5.11 SOM ağının oluşturduğu öbeklerin etiketlerinin düzeltilmesi .....	63
5.12 Eğitilmiş ağın yüklenmesi .....	64
5.13 Test veri setinin yüklenmesi .....	65
5.14 Ağın test edilmesi .....	66
5.15 Ağların bileşik ağ sistemine yüklenmesi .....	67
5.16 Bileşik ağ sisteminin çalıştırılması .....	68
5.17 Görsel simulasyon .....	70
5.18 Ağ birleştirme simulasyonu .....	71
6.1 Bileşik yapay sinir ağı karar mekanizması .....	73
7.1 Deney sonuçlarına ilişkin sütun grafiği .....	99

## ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1 Veraz Switch'lere ait CDR örneği .....	3
4.1 Biyolojik sinir ağı ile yapay sinir ağının karşılaştırılması (Elmas, 2003) .	36
6.1 Veri setinde bulunan sahalara ilişkin açıklamalar .....	74
6.2 Sayısal veri setinden 20 kayıt örneği .....	75
6.3 Ölçeklendirilmiş veri seti örneği .....	76
6.4 Her adımda azalan hata oranları (Deney 1) .....	79
6.5 Eğitim sonu hata yüzdeleri (Deney 1) .....	80
6.6 MLP ağının test sonuçları (Deney 1) .....	81
6.7 LVQ ağının test sonuçları (Deney 1) .....	81
6.8 SOM ağının test sonuçları (Deney 1) .....	82
6.9 Bileşik ağın test sonuçları (Deney 1) .....	82
6.10 Bileşik ağı oluşturan bileşenlerin ortak kararı (Deney 1) .....	83
6.11 MLP ağının test sonuçları (Deney 2) .....	85
6.12 LVQ ağının test sonuçları (Deney 2) .....	85
6.13 SOM ağının test sonuçları (Deney 2) .....	86
6.14 Bileşik ağın test sonuçları (Deney 2) .....	86
6.15 Bileşik ağı oluşturan bileşenlerin ortak kararı (Deney 2) .....	87

**ÇİZELGELER DİZİNİ (devam)**

<u>Çizelge</u>	<u>Sayfa</u>
6.16 MLP ağının test sonuçları (Deney 3) .....	92
6.17 LVQ ağının test sonuçları (Deney 3) .....	93
6.18 SOM ağının test sonuçları (Deney 3) .....	94
6.19 Bileşik ağın test sonuçları (Deney 3) .....	94
6.20 MLP ağının test sonuçları (Deney 4) .....	95
6.21 LVQ ağının test sonuçları (Deney 4) .....	96
6.22 SOM ağının test sonuçları (Deney 4) .....	97
6.23 Bileşik ağın test sonuçları (Deney 4) .....	97

**KISALTMALAR DİZİNİ**

<u>Kısaltmalar</u>	<u>Açıklama</u>
ANI	Automatic Number Identification.
ART	Adaptive Resonance Theory.
BAM	Bidirectional Associative Memory.
CDR	Call Detail Record.
CID	Caller ID.
FBA	Fraud Breaking Agents.
GNN	General Regression Neural Network.
LVQ	Learning Vector Quantization.
MLP	Multilayer Perceptron.
MNA	Mobility Network Agents.
NRI	Non-Receipt Issue.
PCA	Personal Communication Agents.
PNN	Probabilistic Neural Network.
P-RCE	Probability-Restricted Coulomb Energy
RBFNN	Radial Basis Function Neural Network
RMS	Root Mean Square
ROC	Receiver Operating Characteristic.

**KISALTMALAR DİZİNİ (devam)**

<u>Kısaltmalar</u>	<u>Açıklama</u>
SOM	Self Organizing Map.
TDNN	Time Delay Neural Network.
UP	User Profiles.
XML	Extensible Markup Language.

## 1. GİRİŞ

Son yıllarda iletişim sektörünün gelişmesiyle paralel olarak sahtekârlık (ing. fraud) giderek büyüyen bir sorun halini almıştır. Sahtekârlar (ing. fraudsters), çeşitli yöntemlerle telekomünikasyon şirketine para ödemedi hizmet almaktadırlar. Bu yöntemlerin sık kullanılan birkaçı, şirket için faturalandırmanın mümkün olmadığı bazı durumları keşfedip bunları kullanarak ücretsiz görüşmeler yapmak, şirketin yeni üyelere sağladığı ücretsiz görüşme sürelerini kullanmak için sahte üyelikler yaratmak, hizmeti almadığını iddia edip kredi kartı ile yapılan ödemeyi geri çekmektir. Ancak verdikleri kayıp bununla sınırlı değildir. Telekomünikasyon şirketleri, her konuşma için anlaşmalı olarak kullandıkları ağ sağlayıcıları (ing. carriers), arabağlantı operatörleri (ing. interconnection operators) ve uluslararası dolaşım ortaklarına (ing. roaming partners) ödeme yapmaktadırlar. Bu durumda ödemesi alınmayan konuşmalar telekom şirketlerinin kârını büyük oranda düşürmektedir.

Telekomünikasyon sektöründe sahtekârlık yönetimi, uygun yazılımlar aracılığıyla kaçak görüşmelerin gerçekleşmeden önlenmesi, gerçekleşme anında yakalanması, tekrar gerçekleşmesinin engellenmesini amaçlar. Detaylı arama kayıtları (CDR'lar) analiz edilerek sahtekârlık kuralları belirlenir. Bu kurallar kullanıcıların arama ve ödeme alışkanlıkları göz önüne alınarak belirlenebileceği gibi ülke, bölge, şehir temelindeki arama ve ödeme istatistiklerine göre de ayrı kurallar belirlenebilir. Ancak standart bir veri analizinde gözden kaçan detaylar, iyi bir müşterinin bu kurallara göre sahtekâr olarak belirlenip engellenmesi gibi sorunlara yol açarak şirketi daha fazla zarara uğratabilirler. Kriterlerin, ilgili tüm veriler üzerinde yapılacak veri madenciliği sonucu elde edilen çıkarımlar ile kriterlerin tüm özel durumları göz önüne alacak şekilde belirlendiği bir sahtekârlık yönetim sistemi, şirketlerin kaybını büyük oranda azaltacaktır.

Tez çalışması kapsamında çeşitli yapay sinir ağı yaklaşımları ile telekomünikasyonda sahtekarlık probleminde çözüm aranacak bu yöntemler optimum performans sağlayacak şekilde birleştirilerek bir sahtekârlık tespit sistemi geliştirilmiştir.

Öncelikle Bölüm 2'de telekomünikasyonda sahtekârlık konusuna ilişkin kavramlar üzerine detaylı açıklamalar yapılarak ön bilgi verilmektedir.

Bölüm 3'te sahtekârlık yönetimi üzerine geçmişten günümüze kadar gerçekleştirilmiş çalışmalar ve kullanılan teknikler incelenerek telekomünikasyon sektöründe sahtekârlık yönetimi için günümüzde ne tür çözümlerin mevcut olduğu ve tez kapsamında geliştirilen uygulamanın getireceği ilerlemeler üzerinde durulmaktadır.

Bölüm 4'te yapay sinir ağlarına ilişkin temel bilgiler verilmekte ve tez projesinde kullanılan ağ modelleri hakkında detaylı açıklamalar yapılmaktadır.

Bölüm 5'te tez kapsamı dâhilinde gerçekleştirilen uygulamanın detaylı sunumu yapılmaktadır. Kullanılan teknoloji ve yöntemler anlatılmakta, uygulamanın işleyiş aşamaları açıklanmaktadır.

Bölüm 6'da telekomünikasyonda sahtekârlık tespiti konusunda çözümü aranan problem ortaya konmakta, geliştirilen bileşik yapay sinir ağları tabanlı sahtekârlık tespit sistemi kullanılarak bu problemin çözümünde ulaşılan performans ölçümleri gösterilmektedir.

Son olarak Bölüm 7'de çalışmayla ilgili sonuç ve tartışmalara yer verilmekte, telekomünikasyonda yapay sinir ağları ile sahtekârlık tespiti üzerine ileride yapılabilecek çalışmalar hakkında fikir verilmektedir.

## 2. TELEKOMÜNİKASYONDA SAHTEKÂRLIK YÖNETİMİNE GENEL BAKIŞ

### 2.1 Telekomünikasyona İlişkin Temel Kavramlar

#### 2.1.1 Arama detayı kayıtları (CDR)

Arama detayı kayıtları, aramanın üzerinden geçtiği telefon switch'leri tarafından yaratılan ve konuşmaya ilişkin süre, arayan numara, aranan numara, arama başlama süresi, arama tipi gibi bilgileri içeren kayıtlardır. Bu kayıtlar genellikle faturalandırma sistemlerinde işlenerek arama ücretinin hesaplanmasında, başarısız aramalara ilişkin bilgi edinmede ve belli rotalardaki trafik yoğunluklarının ölçülmesinde kullanılabildiği gibi sahtekârlık tespitinde de önemli rol oynamaktadır. Çizelge 2.1'de, Veraz marka switch'lere ait bir CDR örneği görülmektedir.

Çizelge 2.1 Veraz Switch'lere ait CDR örneği

ZoneId	0
RecordSequenceIdentifer	8636547
FileSequenceIdentifier	
GlobalCallIdentifier	AAAAK0s9O1IiEAAEA51u+w
LastReceivedDate	01/01/10 00:02
EventOrder	oS1oSvc2oS3+S4+C5oC6oP7oR8+R9
CdrStatus	S
CallType	3
CallingPartyNumber	7548372540
ChargeNumber	
CalledPartyNumber	107738132408978
NatureOfAddressForCalledPartyNumber	4
OriginatingLineInformation	0
IngressLRN	
IngressCarrierIdentifierCode	
IngressCarrierSelectionInformation	0
IngressCallControlElementID	24
IngressTrunkGroupProtocol	7
IngressTrunkGroupType	1

IngressTrunkGroupID	6
IngressSignalStartDate	01/01/10 00:01
IngressGatewayID	14
IngressCardID	19
IngressSpanID	32
IngressChannelNumber	6
IngressDChannel	
IngressISDNCallReferenceNumber	
IngressCreateConnectionCompleteDate	01/01/10 00:01
IngressAddressCompleteDate	01/01/10 00:01
IngressCallAnswerDate	01/01/10 00:01
TranslatedNumber	497541372220
TerminatingLocalRoutingNumber	
TranslatedCarrierIdentificationCode	
IngressCallReleaseDate	01/01/10 00:02
IngressReleaseComplete	01/01/10 00:02
IngressReleaseCauseCode	16
EgressCallControlElementID	84
EgressTrunkProtocol	15
EgressTrunkGroupType	3
EgressTrunkGroupID	180
EgressCallStart	01/01/10 00:01
EgressGatewayID	0
EgressCardID	0
EgressSpanNumber	0
EgressChannelNumber	0
EgressDChannelNumber	
EgressISDNCallReference	
EgressCreateConnectionCompleteDate	01/01/10 00:01
EgressAddressCompleteDate	01/01/10 00:01
EgressCallAnswerDate	01/01/10 00:01
EgressCallReleaseDate	01/01/10 00:02
EgressReleaseCompleteDate	01/01/10 00:02
EgressReleaseCauseCode	16
FirstCheckPointDate	01/01/01 00:00
LastCheckPointDate	01/01/01 00:00
IngressGatewayAccessID	

EgressGatewayAccessID	Callax_Local_GW_01
IngressTrunkGroupName	CXFRA2
EgressTrunkGroupName	SNS1_BER
OriginatingGatewayIPAddress	
TerminatingGatewayIPAddresses	172.16.7.15:5060
H323ConferenceID	
IngressCardPortNumber	1
IngressCardPathNumber	1
EgressCardPortNumber	0
EgressCardPathNumber	0
IngressTrunkGroupNumber	8002
EgressTrunkGroupNumber	8947
OriginalDialedNumber	1,08E+016
OriginalDialedNumberNatureOfAddress	3
RedirectingInformation	
JurisdictionParameterFromIngress	
JurisdictionParameterFromEgress	
IngressTrunkBearerCapability	1
EgressTrunkBearerCapability	1
TransitNetworkSelectionCarrierCode	
NatureOfAddressOfCallingPartyNumber	3
IngressInternalReleaseCause	<E>15</E><E>0</E>
EgressInternalReleaseCause	<E>0</E><E>0</E>
EgressCalledNumber	38512408458
NatureOfAddressForEgressCalledNumber	4
EgressConnectedNumber	38512408458
NatureOfAddressForEgressConnectedNumber	4
PresentationIndicatorForCallingPartyNumber	0
IngressIRI	
EgressORI	
IngressExternalCallID	
EgressExternalCallID	6998778538941858951-1262904082-8285
IngressChargeInfo	
EgressChargeInfo	
IncomingPartialCallIndicator	
IncomingNationalForwardCallIndicator	
IncomingLastDivertingLineIdentity	

J7SpecificInformation	
IngressSS7GenericParameter	
EgressSS7GenericParameter	
IngressPacketsSent	1923
IngressPacketsReceived	962
IngressPacketsLost	0
IngressPacketsTransferred	0
IngressBytesSent	19230
IngressBytesReceived	19240
IngressBytesLost	0
IngressJitter	4
IngressLatency	0
EgressPacketsSent	0
EgressPacketsReceived	0
EgressPacketsLost	0
EgressPacketsTransferred	0
EgressBytesSent	0
EgressBytesReceived	0
EgressBytesLost	0
EgressJitter	0
EgressLatency	0
IngressCIC	998
EgressCIC	0
IngressCodec	0
EgressCodec	0
IngressLocalGatewayID	0
EgressLocalGatewayID	90
IngressCASCircuitSeizureTime	
EgressCASCircuitSeizureTime	
IngressZZCode	
EgressZZCode	
IngressCountryAddressType	
EgressCountryAddressType	
IngressPartitionNumber	0
EgressPartitionNumber	0
IngressCallingPartyCategory	10
IngressService	<PLCY><PAR>1077</PAR></PLCY>

EgressService	
EarlyEvents	
IngressReleaseCauseLocation	0
EgressReleaseCauseLocation	0
ReverseChargeIndicator	
CalledPartyCategory	1
CallDuration	20
IngressReleaseDirection	3
EgressReleaseDirection	2
IngressAnswerTimeInLocalTime	2010-01-01+01:01:41
EgressAnswerTimeInLocalTime	2010-01-01+01:01:41
IngressHLCCCharacteristicsIdentification	
EgressHLCCCharacteristicsIdentification	
IngressLLCTransferCapability	
EgressLLCTransferCapability	

### 2.1.2 Yerel santral taşıyıcıları

Yerel santral taşıyıcıları (ing. local exchange carriers), bir bölgeyi kapsama alanına alan telefon şirketleridir. Taşıyıcılar hiçbir kısıtlama ve ayırım olmadan telekomünikasyon hizmetlerini sunmak, numara taşınabilirliğini sağlamak, rakip hizmet sağlayıcılara arama eşitliği sağlamak ve telefon numaralarına, operatör hizmetlerine erişim izni vermekle yükümlüdür.

Kullanıcılar, kullandıkları tüm internet tabanlı ve diğer elektronik hizmetler için ayrı ayrı hizmet sağlayıcıları tarafından değil, yerel santral taşıyıcısına olan üyeliği üzerinden faturalandırılır.

### 2.1.3 Arabağlantı

Telekomünikasyonda arabağlantı (ing. interconnection), bir taşıyıcı ağının, o ağa ait olmayan ekipmanlar ile kurduğu fiziksel bağlantıdır. Bu bağlantı taşıyıcıyla müşterisinin ekipmanı arasında veya bir taşıyıcıyla başka taşıyıcılar arasında olabilir. Taşıyıcılar, kendilerine ait ekipmanları bulunmayan bir lokasyona gitmesi gereken aramaları bu bağlantılar aracılığıyla birbirine

yönlendirirler. Bu yönlendirmeler sonucu taşıyıcılar farklı bir taşıyıcıya ait hizmet ve ekipman için kullanım ücreti öderler.

#### **2.1.4 Uluslararası dolaşım ortakları**

GSM teknolojisi dahilinde bir özellik olan dolaşım (ing. roaming), bir GSM operatörünün üyesi olan kullanıcıların, üyeliğini diğer operatörler üzerinden de kullanabilmesine olanak sağlar. Örneğin Türkiye'deki bir operatörün kullanıcısı yurtdışına gittiğinde, üyesi olduğu operatörün anlaşmalı olduğu başka bir operatörün servislerini kullanarak Türkiye'de kullandığı telefonunu kullanmaya devam edebilir. Operatörler, dolaşım anlaşması olan operatörlere kullandıkları kendilerine ait hizmet ve ekipmanlar için faturalandırma yaparlar.

### **2.2 Sahtekârlığa İlişkin Temel Kavramlar**

Genel anlamda sahtekârlık, kişisel çıkar sağlama veya diğer bireylere zarar verme amacıyla yapılan kasıtlı aldatmadır. Sahtekârlar, sistemlerin zayıflıklarını kısa sürede tespit ederek onlardan faydalanmanın yollarını bulurlar.

#### **2.2.1 Sahtekârlık yönetimi**

Sahtekârlık yönetimi, sahtekârlık ile mücadele için oluşturulmuş sistemdir. Dört katmandan oluşur.

#### **2.2.2 Sahtekârlık önleme**

Müşteri profilleri sürekli takip edilip önceden belirlenmiş kriterlere uygunluğu kontrol edilerek sahtekârlığın daha oluşmadan engellenmesine yöneliktir.

#### **2.2.3 Sahtekârlık tespiti**

Arama bilgilerinin incelenerek gerçekleşmiş bir sahtekârlığın en kısa sürede belirlenmesidir. Buna göre sahtekârlığı yapan kullanıcı bloklanabilir.

### **2.2.4 Sahtekârlık yakalama**

Aramalar gerçek zamanlı olarak takip edilerek sahtekârlık kriterlerine uyan bir durum oluştuğu anda aramanın sonlandırılmasına yöneliktir.

### **2.2.5 Sahtekârlık tekrarını engelleme**

Gerçekleşen bir sahtekârlığın sonucunda, sahtekârlık kriterlerinin güncellenmesi ile aynı tip bir sahtekârlığın bir daha gerçekleşmesini engellemeye yöneliktir.

Sahtekârlık ile mücadelede kullanılan farklı yöntemler vardır. Kural tabanlı yönetim, otomatik kullanıcı profillemeye, otomatik kural üretimi gibi ana başlıklarda incelenebilen bu yöntemlere bir sonraki bölümde ayrıntılı olarak değinilecektir.

## **2.3 Telekomünikasyon Şirketlerine Yönelik Sahtekârlık**

Telekomünikasyon alanında yapılan bir sahtekârlık, kullanıcının telekomünikasyon şirketini aldatmasına yönelik, telekomünikasyon şirketinin kullanıcıyı aldatmasına yönelik ve bir üçüncü partinin kullanıcı veya telekomünikasyon şirketini aldatmasına yönelik olmak üzere farklı şekillerde gerçekleşebilir. Tez kapsamında telekomünikasyon şirketlerine karşı yapılan sahtekârlıklara çözüm aranacağından bu bölümde bu tür sahtekârlık tipleri ele alınacaktır. Şirketleri hedef alan sahtekârlığı, telekomünikasyon servislerini ödeme yapmama niyetiyle kullanmak olarak tanımlayabiliriz.

Sahtekârlığın telekomünikasyon şirketine zararı iki durumda incelenebilir. Birinci durumda şirketin uğradığı zarar sadece sunduğu hizmet karşılığı ödeme alamamaktır. İkinci durumda ise daha ağır bir kayıp olarak ödemesini alamadığı hizmeti sağlamak için kasasından para çıkışı olmasıdır. Bu tip kayıplar, Ağ sağlayıcılarına, arabağlantı operatörlerine ve uluslararası dolaşım ortaklarına arama başına yapılan ödemelerdir.

Sahtekârların edindiği fayda yine iki durumda incelenebilir. Birinci durumda sahtekârın kazanımı sadece aldığı hizmet karşılığı ödeme yapmamaktır. İkinci bir durumda ise ayrıca kar elde ederek daha fazla çıkar sağlarlar. Bu kazanç, yüksek ücretli aramaları değerinin altında ücretten satmalarıyla veya sahip oldukları bir özel hattı (ing. premium rate service), yani gelen aramalar üzerinden kazanç sağladıkları hattı aramalarıyla gerçekleştirir..

### **2.3.1 Rakip şirketler tarafından uygulanan sahtekârlıklar**

*Arabağlantı sahtekârlığı*; taşıyıcıların kayıtları üzerinde oynanarak bir telekomünikasyon şirketinin diğerine olan borcunun yanlış hesaplanmasını sağlamaya yöneliktir.

Arabağlantı sahtekârlığının bir yöntemi olan *işlem tekrarı* (ing. refiling) *sahtekârlığı*; taşıyıcılardan birinin aramayı rakip şirkete aktarmadan önce CID veya ANI verisiyle oynaması ile gerçekleşir. Aramayı hangi taşıyıcı şirketin ne kadar mesafe taşıdığı bilgisi yanlış hesaplanacağından rakip şirketlerin birbirine yaptığı faturalandırma da yanlış olur.

### **2.3.2 Kullanıcılar tarafından uygulanan sahtekârlıklar**

*Üyelik sahtekârlığı*, telefon kullanıcılarının ücret ödeme niyeti olmadan arama yapmasına yöneliktir. Kullanıcı, yeni üyelikte verilen ücretsiz konuşma süreleri gibi ek hizmetlerden yararlanmak amacıyla farklı kullanıcı bilgileriyle üyelikler alarak ücret ödmeden telekomünikasyon hizmetini kullanır.

### **2.3.3 Üçüncü partiler tarafından uygulanan sahtekârlıklar**

*Telefon korsanlığı* (ing. phreaking), telefon ağının işleyişi hakkında bilgi edinen birinin yetkisiz aramalar yapabilmesine yöneliktir. Bu tür sahtekârlık, sosyal mühendislik yöntemleriyle telekomünikasyon çalışanlarından teknik bilgi sızdırmak ile mümkün olabilir.

***Switch yeniden programlama*** yöntemi ise daha teknik bir yöntemdir. Bu durumda sahtekâr telekomünikasyon şirketinin bilgisayar ağına veya faturalandırma sistemine giriş yaparak ücretsiz aramalar yapmasına imkân sağlar.

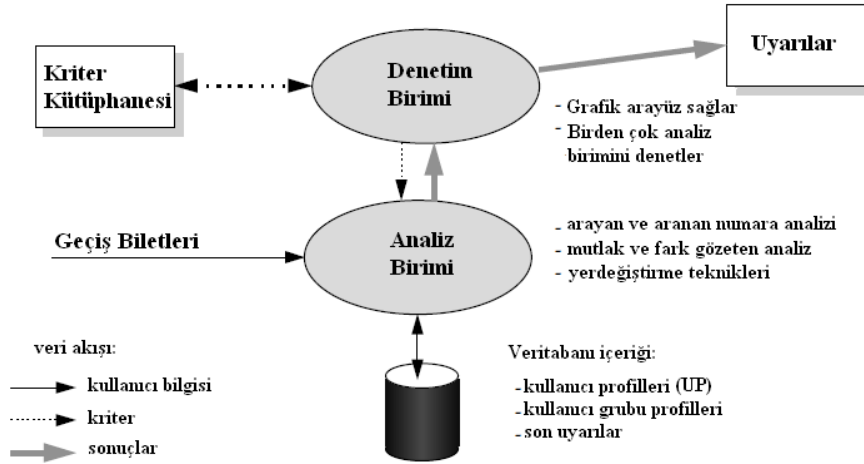
***Telefon klonlama*** ile sahtekârlar başka bir kullanıcının elektronik seri numarası ve telefon numarasını kopyalayarak bu numaradan arama yaparlar ve faturanın da kopyalanan kullanıcıya gitmesini sağlarlar.

### 3. SAHTEKÂRLIK TESPİTİNDE TEMEL YAKLAŞIMLAR

Geleneksel yaklaşımda kurallar bir uzmanın bilgisine dayanılarak belirlenir. Ancak bu bilgi, geçmiş konuşma verilerinden elde edilen bilgiyle otomatik kural keşfetme ve var olan kuralları optimize etme amaçlı bir sistem ile de desteklenmelidir. Geçmişten günümüze bu amaç için çeşitli yöntemler kullanılmıştır.

#### 3.1 Kural Tabanlı Sahtekârlık Tespiti

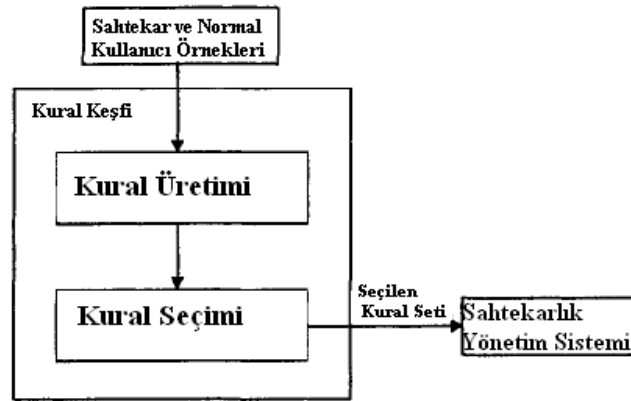
Fawcett ve Provost'un (1997) geliştirdiği uyarlanabilir kural tabanlı tespit altyapısında sahtekârlık göstergelerinin tespiti için veri madenciliği yapılır ve eşik değerler kullanılarak kurallar oluşturulur. Normal kullanım esnasında her hesabın profili çıkarılır. Kullanıcının her hareketi bu profil ile gösterdiği farklılıkları tespit etmek amacıyla izlenir. Günlük eşik değerleri ile karşılaştırılarak şüpheli hareketler ve kullanıcının kişisel profiline göre normalin üstünde olan ücretlendirmeler açığa çıkarılır. Sistem çıktıları birleştirilerek alarm verilip verilmeyeceğine karar verir. Şekil 3.1'de bu sistemin mimarisi görülmektedir.



Şekil 3.1 Kural tabanlı sahtekârlık yönetim sistemi mimarisi (Fawcett and Provost, 1997)

Sahtekârlık tespiti için CDR'lar tek başına yeterli değildir. Bir arama modeli bir kullanıcı için sahtekârlık olarak kabul edilirken başka bir kullanıcının olağan arama eğilimi dâhilinde olabilir. Sahtekârlığın yakalanması için arama detayları ile birlikte kullanıcının hizmetleri kullanım yoğunluğu, ödeme davranışları, güvenilirliği gibi etkenlerin de göz önünde bulundurulması gerekmektedir. Rosset vd, (1999) standart sınıflandırma ve kural üretiminin sahtekârlık tespiti için uygun olmadığını savunur.

Şekil 3.2'de görüldüğü gibi otomatik kural tespiti için müşteri özellikleri ve spesifik davranış özellikleri olarak iki veri katmanını kullanan, *kural üretimi* ve *kural seçimi* olmak üzere iki adımdan oluşan bir sahtekârlık yönetim sistemi geliştirmişlerdir. Kural üretim mekanizmasında normal kullanım ve sahtekâr kullanım örnekleri işlenerek çok sayıda aday kural üretilir ve kural seçimi mekanizmasında bu adaylar arasından kullanıcı ve davranış verilerine bağlı olarak optimum kural kümesi ortaya çıkarılır. Optimum kural kümesi, yüksek oranda sahtekârlık durumunu kapsamalı, doğru olmalı, hızlı tetiklenmeli ve küme boyutu küçük olmalıdır.

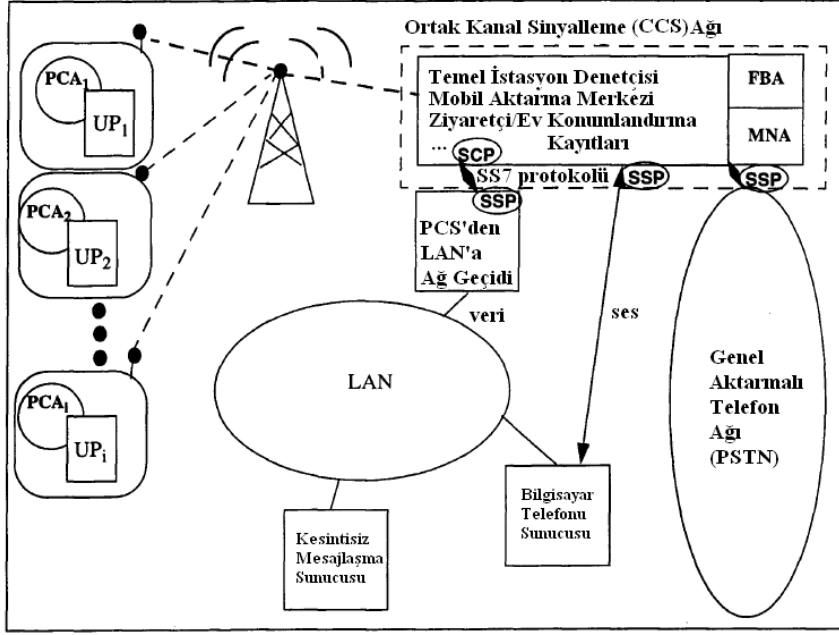


Şekil 3.2 İki aşamalı kural keşfi mimarisi (Rosset et al., 1999)

### 3.2 Çok Etmenli Sistemler ile Sahtekârlık Tespiti

Abu-Hakima vd, (1997) daha yüksek kullanıcı kişiselleştirmesi ve daha gelişmiş gerçek zamanlı sahtekârlık tespiti için çok etmenli (ing. agent) bir

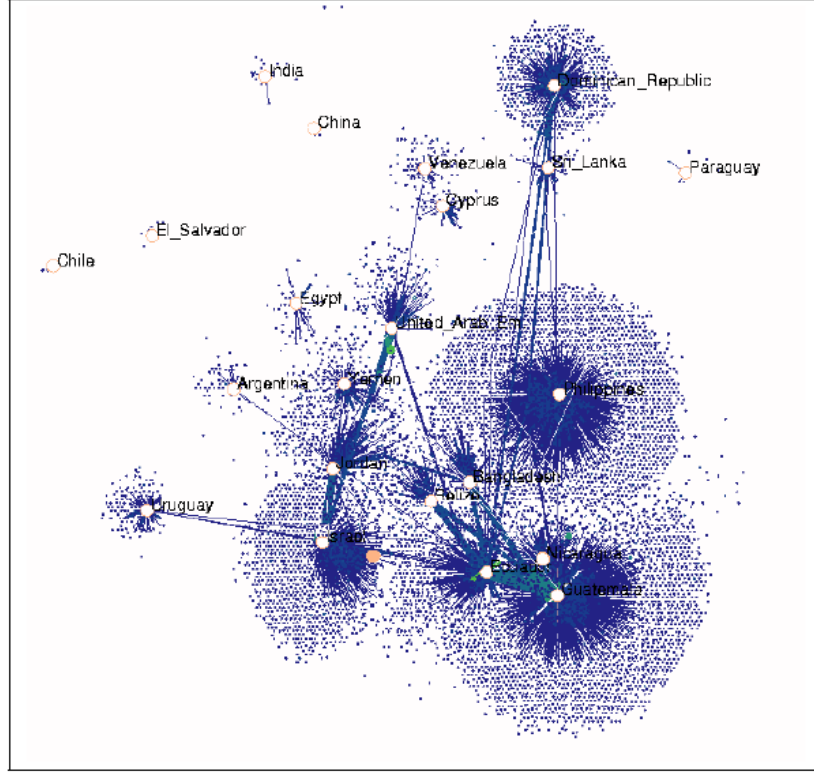
sahtekârlık tespit sistemi geliştirmiştir. Gömülü yapay zekâ tekniklerini kullanan çok yönlü dağıtık bir yaklaşım önerilmiştir. Bu yaklaşımda Kişisel İletişim Etmenleri (PCA's), Hareketlilik Ağı Etmenleri (MNA's), kullanıcı profilleri (UP's) ve Sahtekârlık Engelleme Etmenleri (FBA's) kullanılmıştır. Geliştirilen sistemin yapısı Şekil 3.3'te görülmektedir.



Şekil 3.3 Kişisel iletişim ağlarında sahtekârlık için çok etmenli sistem yapısı  
(Abu-Hakima et al., 1997)

### 3.3 Görsel Veri Madenciliği Tabanlı Sahtekârlık Tespiti

Cox vd, (1997) insan örüntü tanıma yeteneklerinin birçok durumda otomatik veri madenciliği algoritmalarından daha yüksek olduğunu ileri sürerek geliştirdikleri görsel veri madenciliğine dayalı telekomünikasyon sahtekârlığı tespit sisteminde insan ve makinelerin tespit gücünü birleştirerek daha büyük hesaplama kapasitesi elde etmeyi hedeflemiştir. Şekil 3.4'te, bu çalışmada elde edilen sekiz saatlik uluslararası aramaların görsel temsili görülmektedir. Aramaların olağan dışı şekilde yoğunlaştığı koyu renkli alanlar incelendiğinde sahtekârlıkla ilişkili olabildikleri görülmüştür.



Şekil 3.4 Görsel veri madenciliğine dayalı arama gösterimi (Cox et al., 1997)

### 3.4. Markov Zincirleri ile Sahtekârlık Tespiti

Taniguchi et al., 1999'da mobil iletişim ağları için bir sahtekârlık tespit sistemi sunmuştur. Hiyerarşik rejim değiştirme modeli tabanlı sistemde zamanla markov zincirlerine bağlı stokastik olarak gelişen üç değişken kullanılmıştır. Bunlar; bir hesap bir sahtekâr tarafından kurban edilmekte ise 1, diğer durumlarda 0 değerini alan “victimized”, sahtekârlık eylemi gerçekleşmekte ise 1, değilse 0 değerini alan “fraud” ve telefon kullanım halinde ise 1, değilse 0 değerini alan “call” değişkenleridir. Tespit problemi, rejim olasılıkları üzerinde bir çıkarım problemi olarak formüle edilmiştir. En alt hiyerarşi katmanında her bir aramanın dinamik davranış, bir üst katmanda normal davranıştan sahtekârlığa geçiş, en üst katmanda da “victimized” modelleri çıkarılmıştır.

### 3.5 Yapay Sinir Ağları Tabanlı Sahtekârlık Tespiti

Barson vd, (1996) mobil telefon ağları için geliştirdikleri öğreticili ileri beslemeli yapay sinir ağı tabanlı sistemi ile anormal kullanımı tespit etmeyi amaçlamıştır. Mobil telekomünikasyon sahtekârlığının değişen kullanım

örüntüleri gibi karakteristik özellikleri ve sahtekârlık tespiti için normal ve sahtekâr kullanım ayrımının gerekliliği yapay sinir ağı yaklaşımının tercih edilmesine neden olmuştur. Bu sistemde 18 girdi, çeşitli gizli birimler ve iki çıktı biriminden oluşan bir Çok Katmanlı Algılayıcı (MLP) ağı uygulaması yapılmıştır. Girdi olarak her kullanıcı için geçmiş ve güncel olmak üzere iki zaman periyoduna ait; düşük yerel kullanım ile yüksek uluslararası kullanım arasında altı gruba ayrılmış konuşma kayıtları kullanılmıştır. Çıktılardan biri normal kullanım, biri sahtekâr kullanımı temsil eder.

Britos vd, (2008) cep telefonu kullanıcılarının kullanımındaki olağan dışı değişiklikleri tespit etme amacıyla kullanıcıların davranış geçmişlerini temsil eden veri yapıları oluşturma yaklaşımı izlenmiştir. Yerel, ulusal ve uluslararası ağlara ait arama kayıtlarında ayrı ayrı Özörgütlemeli Harita (SOM) yapay sinir ağı algoritması uygulanarak örüntüler üretilmiştir kişinin güncel kullanım ile kullanım tarihi profilleri sıklık dağılımı vektörleri olarak tutulduğu için bunlar arasındaki farkın boyutunu ölçmek için vektörel uzaklıkları hesaplanır. Bunun için Hellinger mesafeleri kullanılarak 0-2 arasında değişen uzaklıklar elde edilmiştir. 0 dağılımların eşitliğini, 2 ise ortagonallliğini temsil eder ve 2'ye yaklaşıması sahtekârlık şüphesinin artması anlamına gelir.

Bu yöntemlerin çoğu kredi kartı sahtekârlığı problemlerinde de kullanılmıştır:

Ghosh ve Reilly'nin (1994) geliştirdiği sistemde yapay sinir ağı iki aylık zaman periyodunda gözlemlenen kayıp kartlar, çalıntı kartlar, uygulama sahtekârlığı, sahte kartlar, posta siparişi sahtekârlığı, NRI sahteciliği örnekleri ile eğitilmiştir. Bu ağ, kural tabanlı sahtekârlık tespit prosedürlerine göre daha az yanlış pozitif ile daha fazla sahtekârlık hesabı tespit etmiştir. Bu sistemde üç katmanlı bir ileri beslemeli ağ olan P-RCE yapay sinir ağı yöntemi kullanılmıştır. Çalışmada P-RCE'nin girdi katmanı güncel işlem bilgileri, işlem geçmişi bilgileri, ödeme geçmişi bilgileri gibi bilgilerden, çıktı katmanı, sahtekârlık skoru olarak ele alınan nümerik sonucu tutan tek bir hücreden oluşur.

Brause vd.'nin (1999) çalışmasında, düşük yanlış uyarı oranı ile yüksek sahtekârlık tespitinin sağlanması için ileri veri madenciliği teknikleri ile yapay sinir ağı algoritmasının birleştirilmesi gösterilmiştir. Bu çalışmada kredi kartı sahtekârlığı tanısında istatistik tabanlı yaklaşım kullanılmıştır.

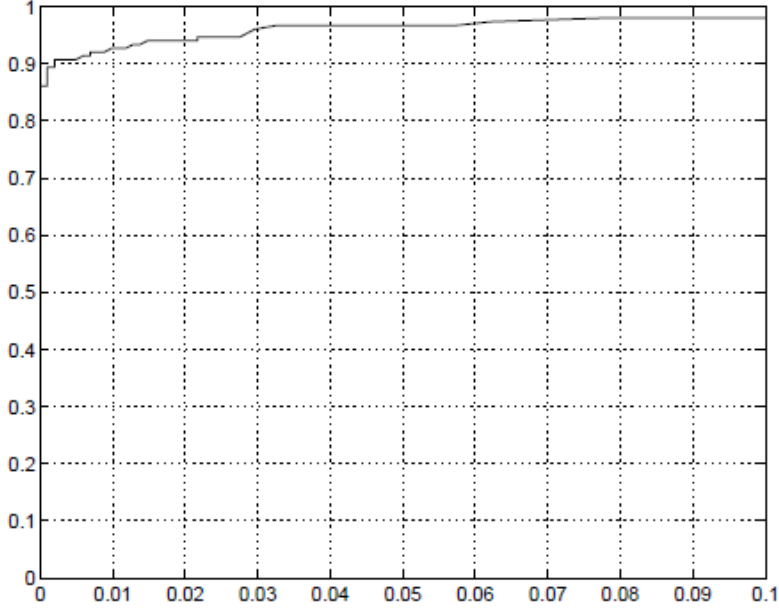
Zaslavsky ve Strizhak'ın (2006) geliştirdiği sistemde yapay sinir ağı algoritması olarak Özörgütlemeli Haritalar kullanılarak tipik bir kart kullanıcısı modeli yaratılmış ve işlem farklılıkları analiz edilerek şüpheli işlemler tespit edilmiştir.

### **3.6 Melez Sistemlerle Sahtekârlık Tespiti**

Burge ve Taylor, (1997) kural tabanlı, öğreticili öğrenmeli yapay sinir ağı tabanlı, arayan numara'yı kullanan öğreticisiz öğrenmeli yapay sinir ağı tabanlı ve arayan numara'yı kullanan öğreticisiz öğrenmeli yapay sinir ağı tabanlı olmak üzere dört bileşenden oluşan melez bir sahtekârlık tespit aracı geliştirmiştir. “BRUTUS” adı verilen bu sistemde güncel davranış profilleri ve davranış geçmişlerinin olasılık dağılımları Hellinger uzaklıkları kullanılarak karşılaştırılır. Mesafe uzadıkça sahtekârlık şüphesi artar.

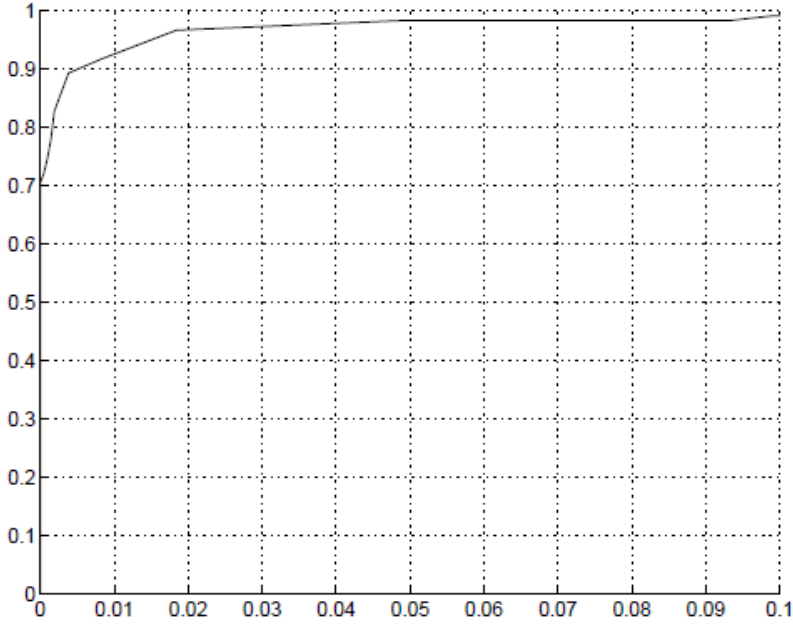
Taniguchi vd, (1998) Sahtekârlık tespiti için üç yaklaşım sunmuştur ve her metodun performans testi için ROC eğrileri kullanılmıştır:

Normal ve sahtekâr sınıfları arasında doğrusal olmayan fark gözetken fonksiyonları öğrenmek için öğreticili öğrenme tabanlı ileri beslemeli yapay sinir ağı. Şekil 3.5'teki ROC grafiğinde bu yaklaşım sonucu elde edilen sahtekârlık tespiti oranının %85'in üstünde olduğu görülmektedir.



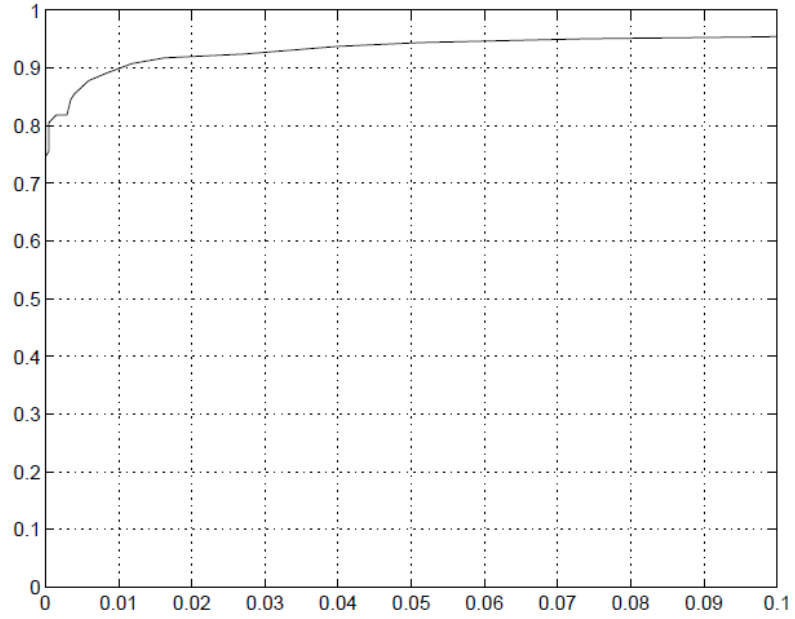
Şekil 3.5 İleri beslemeli yapay sinir ağı ROC grafiği (Taniguchi et al., 1998)

Her üyenin geçmiş davranışlarını modellemek ve geçmişe göre gösterdiği anormallikleri tespit etmek için Gauss karışım modelleri ile yoğunluk tahmini uygulaması. Şekil 3.6'daki ROC grafiğinde bu yaklaşım sonucu elde edilen sahtekârlık tespiti oranının %70'in üstünde olduğu görülmektedir.



Şekil 3.6 Gauss nodeli ROC grafiği (Taniguchi et al., 1998)

Normal ve sahtekâr kullanım varsayımları altında olasılıksal modeller tanımlamak için Bayes ağları. Bayes ağında değişken olarak çalışma ve tatil günleri için mesai saati içinde ve dışında arama sıklıkları, gece arama sıklığı, uluslararası arama sıklığı, çalışma ve tatil günleri için mesai saati içinde ve dışında ortalama konuşma süresi, gece ortalama konuşma süresi, ortalama uluslararası konuşma süresi bilgileri kullanılmıştır. Şekil 3.7'deki ROC grafiğinde bu yaklaşım sonucu elde edilen grafiğin Gauss modeline benzediği görülmektedir.



Şekil 3.7 Bayes ağı ROC grafiği (Taniguchi et al., 1998)

Bhargava vd, (2003) sahtekârları taklitçiler (ing. impersonators) ve dolandırıcılar (ing. swindlers) olarak ayırır. taklitçiler, kurbanının hesabını ele geçirerek onun kimliğine bürünür ve klon telefonlar yapar. Olağan dışı davranış kurban tarafından gerçekleştirilmiş görünür ve faturalandırma kurbanı yapılır. Dolandırıcılar, sistemi veya kullanıcıları aldatarak kasıtlı olarak zarar verir. Bu çalışmada dolandırıcılar ele alınmıştır. Üç yaklaşım vardır.

1. Bir kullanıcının hareketlerinin normal örüntüden farklılığı tespit edilir.
2. Mevcut sahtekârlık senaryoları için durum geçişi çizgeleri oluşturularak bilinen örüntülere benzer sahtekârlıklar yakalanır.
3. Kullanıcının geçmiş davranışlarındaki kastı araştırılır.

İlk iki yaklaşım aynı zamanda taklitçilerin tespitinden de kullanılır. Sonuncusu sadece dolandırıcı tespitine uygulanabilir. Bu üç yaklaşımı kullanan bir mimari sunulmuştur. Mimari dört bileşenden oluşur:

1. Profil tabanlı anormallik tespiti,
2. Durum geçişi analizi
3. Anlık eylemin tehlike teşkil etmesi durumunda alarm gönderimi
4. Aldatmanın kasıtlılık durumu tahmini

## 4. YAPAY SİNİR AĞLARI

### 4.1 Yapay Zekâ

Günümüzde kullanıcıların tekrarlanan işlerin yol açtığı verim kaybını önlemek amacıyla zeki yazılımları tercih etmesi nedeniyle zeki yazılımların geliştirilmesine yönelik çalışmalar önem kazanmıştır.

D. Lenat ve E. Feigenbaum'un tanımlarına göre zekâ;

“Karmaşık bir problemi çözmek için gerekli bilgileri toplayıp birleştirme kabiliyetidir”

“Karmaşık bir problemi, çözüm arama alanını daraltarak kısa yoldan çözebilme kabiliyetidir” (Kocabaş, 2011; Uğur ve Kınacı, 2006)

Zekâ'nın sözlük anlamı; İnsanın düşünme, akıl yürütme, nesnel gerçekleri algılama, kavrama, yargılama, sonuç çıkarma yeteneklerinin tümüdür. Ayrıca: Soyutlama, öğrenme ve yeni durumlara uyma gibi yetenekler de zekâ kapsamı içindedir. (Uğur ve Kınacı, 2006)

Yapay zekâ, organik olmayan ancak bu özellikleri taşıyan sistemlerin sahip olduğu zekâdır. İnsan zekâsının bilgisayarlar tarafından taklit edilmesini sağlayan yöntemler yapay zekâ'nın çalışma alanına girer. Luger ve Stubblefield, 1993 yılında yapay zekâyı “zeki davranışların otomasyonu ile ilgili bilgisayar bilimleri dalıdır” şeklinde tanımlamıştır. (Uğur ve Kınacı, 2006)

#### Yapay Zekâ'nın Amaçları

İnsan beyninin fonksiyonlarını bilgisayar modelleri yardımıyla anlamaya çalışmak.

İnsanların sahip olduğu zihinsel yetenekleri, bilgi kazanma, öğrenme ve buluş yapmada uyguladıkları strateji, metot ve teknikleri araştırmak.

Bu öğrenme metotlarını biçimsel hale getirmek ve bilgisayarlarda bilgi sistemleri halinde uygulamak.

İnsanlarını bilgisayar kullanımını kolaylaştıracak insan/bilgisayar ara birimleri geliştirmek.

Belli bir uzmanlık alanı içindeki bilgileri toplayarak “uzman sistemler” oluşturmak.

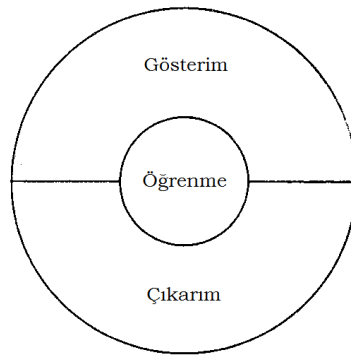
Geleceğin bilgi toplumunun kurulmasında önemli rol oynayacak genel bilgi sistemleri geliştirmek.

Yapay zekâ iş yardımcıları ve “zeki robot takımları” geliştirmek.

Bilimsel araştırma ve buluşlarda faydalanmak üzere, “araştırma yardımcıları” geliştirmek. (Kocabaş, 2011)

1990 yılında Sage’in yaptığı tanıma göre yapay zekânın amacı; makinelerin, hâlihazırda insanların daha iyi yapmakta olduğu bilişsel işleri yürütebilmesi için gereksinim duyduğu paradigmaların ve algoritmaların geliştirilmesidir. Bir yapay zekâ sisteminin üç yeteneğe sahip olması gerekir: bilginin saklanması, saklanan bilginin problemin çözümünde kullanılması, deneyimler sonucunda yeni bilgilerin elde edilmesi. (Haykin, 1999)

Sage’e göre Bir yapay zeka sisteminde; gösterim, çıkarım ve öğrenme olmak üzere üç temel bileşenin bulunması gerekmektedir. Şekil 4.1’de bir yapay sinir ağının üç temel bileşeni görülmektedir.



Şekil 4.1 Bir yapay sinir ağının üç temel bileşeni. (Haykin, 1999)

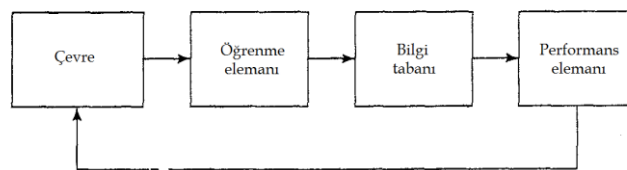
**Gösterim:** Bir yapay zekanın en belirgin özelliği, bir sembol yapıları dili ile probleme ilişkin hem genel hem de spesifik bilgilerin temsil edilmesidir. Semboller genellikle yapay zekânın sembolik gösterimlerinin kullanıcılar tarafından anlaşılmasının kolay olacağı alışılmış terimlerde ifade edilirler. (Haykin, 1999) Bilgi gösterimi yeteneği, yapay zekânın bildiklerini ve duyduklarını sistemde saklayabilmesini sağlar. (Russell and Norvig, 2003)

**Çıkarım:** Yapay zekânın problem çözme yeteneğidir. Bir sistemin çıkarım sistemi olarak nitelenmesi için bazı durumları sağlaması gerekir:

- Sistem geniş kapsam ve çeşitlilikte problemleri ifade edebilme ve çözebilme yeteneğine sahip olmalıdır.
- Sistem, belirgin ve örtülü bilgileri tanıyabilmelidir.
- Sistem probleme bir çözüm elde edildiğinde veya problem üzerinde çalışmanın sonlandırılması gerektiğinde hangi işlemlerin uygulanacağına karar veren bir kontrol mekanizmasına sahip olmalıdır. (Haykin, 1999)

Çıkarım yeteneği, saklanan bilgilerin soruları cevaplamak ve yeni sonuçlara varmak için kullanılmasını sağlar. (Russell and Norvig, 2003)

Öğrenme: Şekil 4.2’de gösterilen basit makine öğrenmesi modelinden anlaşıldığı üzere öğrenme elemanına çevre tarafından bazı bilgiler sağlanır. Öğrenme elemanı bu bilgileri bilgi tabanını geliştirmek güncellemek için kullanır ve son olarak performans elemanı, bilgi tabanını kullanarak işin yürütülmesini gerçekleştirir. (Haykin, 1999) Öğrenme yeteneği, sistemin yeni durumlara uyum sağlaması ve örüntüleri tespit ve tahmin etmesini sağlar. (Russell and Norvig, 2003)



Şekil 4.2 Basit makine öğrenmesi modeli. (Haykin, 1999)

Günlük yaşamda giderek yaygınlaşan öğrenebilen yazılımlar bu yetenekleri sayesinde deneyimlerine dayanarak davranışlarını değiştirebilirler. Makinelerin öğrenmesini sağlayan en güçlü yöntemlerden biri Yapay Sinir Ağları'dır. Yapay zekâ'nın diğer uygulama alanları ise; Uzman Sistemler, Bulanık Mantık, Genetik Algoritmalar, Robotbilim, Bilgisayarlı Görü ve görüntü İşleme, Ses Tanıma, Doğal Dil Anlama, Oyun Oynama, Yapay Yaşam, Veri Madenciliği, Yazılım Etmenleri, Anlamsal Web ve Dağıtık Yapay Zekâ'dır. (Uğur, 2008)

## 4.2 Yapay Sinir Ağları

Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir. (Öztemel, 2006) Sinir sisteminin davranışlarının kopyalayabilmesi için yapısının da kopyalanmasının gerektiği düşünen bilim adamları, yapay sinir ağlarının modellenmesinde sinir sisteminin yapısını örnek almışlardır. (Kakıcı, 2009)

1999 yılında Haykin'in yaptığı tanıma göre bir yapay sinir ağı, basit işlem birimlerinden meydana gelen ve deneysel bilgili saklamaya ve kullanıma uygun hale getirmeye doğal bir eğilimi olan bir paralel dağıtık işlemcidir. İki açıdan beyin ile benzerlik gösterir:

Ağ, öğrenme süreci boyunca çevreden bilgi toplar.

Sinaptik ağırlıklar olarak bilinen nöronlar arası bağlantı kuvvetleri, toplanan bilginin saklanmasında kullanılır. (Haykin, 1999)

### 4.2.1 Yapay sinir ağlarının genel özellikleri

Yapay sinir ağlarını diğer programlardan ayıran ve avantaj olarak görülen özellikleri aşağıda belirtilmiştir.

**Öğrenme:** Yapay sinir ağları bir olayı öğrenerek o olaya ilişkin örnekleri kullanarak genelleme yeteneği kazanırlar. Ağın eğitilmesi, bir olayı bütün yönleri

ile gösterebilecek kapsamda örneklerin tek tek ağı gösterilmesi ve ağındaki örneklerdeki olaylar arasındaki ilişkileri belirlemesidir. Ağı eğitilmesi için elde bulunan örnekler eğitim ve test setleri olarak kullanılmak üzere ikiye bölünür. Ağı eğitim setindeki bütün örnekler doğru sonuçlar üretmeye başladığı zaman eğitim tamamlanmış kabul edilir. Daha sonra test setindeki örnekler ağı gösterilir. Ağı daha önce hiç görmediği örnekler kabul edilir doğrulukta cevaplar veriyorsa ağı performansı iyi kabul edilir. Performansın yetersiz sayıldığı durumlarda ağı tekrar eğitilmesi veya eğitim için yeni örneklerin belirlenmesi yoluna gidilir.

**Paralel çalışma:** Yapay sinir ağı oluşturulan bütün yapay sinir hücreleri eşzamanlı olarak çalışmaktadır. Bu özellik, yapay sinir ağlarının geleneksel programlardan daha hızlı çalışmasını sağlamaktadır. (Akdağ, 2010)

**Bilginin saklanması:** Yapay sinir ağları bilgileri geleneksel programlarda olduğu gibi veritabanında veya programın içine gömülü olarak değil, ağı bağlantılarında saklarlar. Bu nedenle bilgilerin ortaya çıkarılıp yorumlanması zordur.

**Uyarlanabilirlik:** Yapay sinir ağları örnekler ile kendilerine gösterilen yeni durumlara uyum sağlamak amacıyla ağırlıklarını düzenleyebilir ve sürekli yeni olayları öğrenebilirler.

**Hata Toleransı:** Yapay sinir ağları eğitildikten sonra eksik veya belirsiz bilgilerle çalışabilir ve sonuç üretebilirler. Eksik bilginin ağı performansına etkisi, bu bilginin önemine bağlıdır.

**Dereceli bozulma:** Eksik bilgi veya hücrelerin bozulması nedeniyle ağlarda herhangi bir problem ortaya çıktığında hemen bozulmazlar. Hata toleransı özelliği sayesinde yavaş yavaş bozulurlar.

**Dağıtık bellek:** Yapay sinir ağlarında bilgi ağı yayılmış durumdadır. Ağıdaki bir bağlantının tek başına bir anlamı yoktur. Ağı tamamı, öğrendiği olayın bütünü temsil eder. Ağı bilgilerinin açıklanamamasının nedenlerinden biri budur. (Öztemel, 2006)

Yapay sinir ağlarının problemlerin çözümünde sağladığı birçok avantaja karşın dezavantaj olarak kabul edilen bazı özellikleri de vardır:

**Donanım bağımlılığı:** Yapay sinir ağlarının gerçek zamanlı bilgi işleyebilmeleri için paralel çalışan işlemcilerle ihtiyacı vardır. Paralel işlemlerin seri makinelerde yapılması zaman kaybına yol açar.

**Optimum çözüm garantisinin olmaması:** Probleme uygun ağ yapısının belirlenmesi için bir yöntem olmadığından genellikle deneme yanılma yöntemi tercih edilir. Bir problem için uygun ağ yapısının oluşturulmadığı durumlarda problem çözülemez veya düşük performanslı çözümler elde edilir.

**Parametre değerlerinin belirlenmesinde belli kuralların olmaması:** öğrenme katsayısı, katmanlarda bulunacak hücre sayısı gibi değişkenlerin belirlenmesi kullanıcının tecrübesine bağlıdır. Her problem için ayrı faktörlerin göz önüne alınması gereklidir.

**Problemin ağa gösteriminin yalnızca nümerik formatta olması:** Yapay sinir ağları sadece nümerik değerlerle çalışabildiği için problemin nümerik gösterime dönüştürülmesi gerekir. Uygun gösterim mekanizmasının kurulamaması problemin çözülememesine veya düşük performanslı çözümler üretilmesine neden olabilir. Aynı zamanda problemin ağa gösterilişi de performansı etkileyen faktörlerden biridir. Bir olay ayrık veya sürekli değerlerle gösterilebilir ve hangisinin daha başarılı bir öğrenme sağlayacağı bilinmemektedir.

**Eğitimin ne zaman bitirileceğine karar vermek için geliştirilmiş bir yöntemin olmaması:** Ağın kendisine gösterilen örnekler üzerindeki hatasının belli bir değerin altına inmesi eğitimin tamamlanması için yeterli görülebilir ancak bu durum optimum öğrenmenin gerçekleştiğini ve ağın optimum sonuçlar vereceğini göstermez.

**Ağın davranışlarının açıklanamaması:** Bir problem için üretilen çözümün nasıl ve neden üretildiğine dair bilgiye ulaşmanın mümkün olmaması ağın sonucuna olan güveni azaltır. (Öztemel, 2006)

#### **4.2.2 Yapay sinir ağlarının uygulama alanları**

Yapay sinir ağlarından yaygın olarak yararlanılan alanlar şu şekilde belirtilebilir;

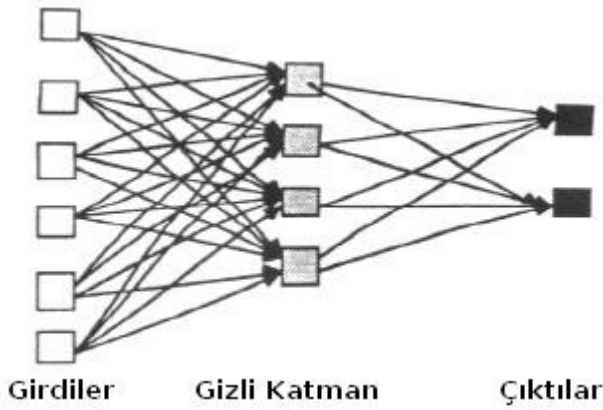
- Olasılıksal fonksiyon kestirimleri
- Sınıflandırma
- İlişkilendirme ve örüntü tanımlama
- Zaman serileri analizleri
- Sinyal filtreleme
- Veri sıkıştırma
- Örüntü tanıma
- Doğrusal olmayan sinyal işleme
- Doğrusal olmayan sistem modelleme
- Optimizasyon
- Zeki ve doğrusal olmayan kontrol (Öztemel, 2006).

### 4.2.3 Yapay sinir ağı yapıları

Yapay sinir ağları yapılarına göre ileri beslemeli ve geri beslemeli olmak üzere iki grupta sınıflandırılırlar.

#### 4.2.3.1 İleri beslemeli ağlar

İleri beslemeli yapay sinir ağları, sinyallerin yalnızca girdi katmanından çıktı katmanına doğru tek yönlü olarak taşınabilmesine olanak verir. Geri besleme döngüleri oluşmaz; örneğin bir katmanın çıktısı aynı katman üzerinde etkili olamaz. Yaygın olarak örüntü tanımlama problemlerinde kullanılırlar. Bu tür ağ yapılarından aynı zamanda “aşağıdan yukarıya” veya “yukarıdan aşağıya” olarak da söz edilir. (Stergiou and Siganos, 1996) Şekil 4.3’te basit bir ileri beslemeli ağ örneği görülmektedir.

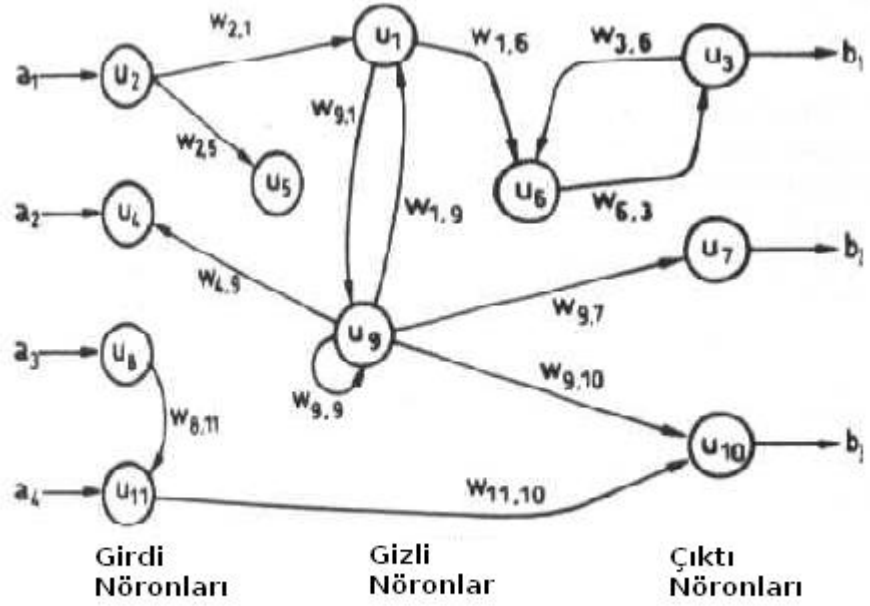


Şekil 4.3 Basit bir ileri beslemeli ağ örneği (Stergiou and Siganos, 1996)

#### 4.2.3.2 Geri beslemeli ağlar

Geri beslemeli ağlar, sinyallerin her iki yönde de iletilerek ağda döngüler oluşturmasına imkân verirler. Bu tür ağlar dinamiktir, yani kararlı hale gelene kadar durumları sürekli değişir. Kararlı hale geldiklerinde ise girdiler değişene kadar kararlılık noktasında kalırlar. Bu tür ağ yapılarından aynı zamanda “interaktif” veya “tekrarlı” olarak da söz edilir. (Stergiou ve Siganos)

Bir katmanın çıktı değerinin aynı katmana veya diğer katmanlara girdi olarak beslenebildiğinden dolayı, ağın herhangi bir andaki çıkış değeri hem o andaki hem de daha önceki giriş değerlerine göre belirlenir. Geri beslemeli ağlar dinamik yapılarından dolayı yaygın olarak tahminleme uygulamalarında kullanılırlar. (Güven, 2005) Şekil 4.4'te bir geri beslemeli ağ örneği görülmektedir.



Şekil 4.4 Bir geri beslemeli ağ örneği (Stergiou and Siganos, 1996)

MLP, LVQ, RBFNN, PNN, GNN ileri beslemeli ağlara, SOM, BAM, ART, Elman, Jordan, RecurrentBP, TDNN geri beslemeli ağlara örnektir. (Uğur,2005)

#### 4.2.4 Yapay sinir ağı algoritmaları

##### 4.2.4.1 Öğreticili öğrenme

Bu öğrenme yönteminde eğitim sırasında ağa hem girdiler, hem de o girdiler karşılığında oluşturulması gereken çıktılar verilerek ağın girdi ve çıktılar arasındaki ilişkileri öğrenmesi amaçlanır. (Uğur, 2005)

Öğrenme süresince ağın ürettiği çıktılar beklenen çıktı değerleri ile karşılaştırılarak hata oranı hesaplanır. Ağırlık değerleri hesaplanan hata oranına göre yeniden düzenlenir. (Kakıcı, 2009) Böylece bir sonraki adım veya döngüde beklenene daha yakın bir çıktı değerinin üretilmesi sağlanır. Eğitim aşaması genellikle uzun zaman almaktadır. Yapay sinir ağlarının birçoğu öğreticili öğrenme yöntemini kullanır. Bu yöntemi kullanan en yaygın ağ modeli MLP (Çok Katmanlı Algılayıcı) ağıdır. (Uğur, 2005)

#### **4.2.4.2 Destekleyicili öğrenme**

Bu tür öğrenmede sisteminde girdi değerleri için üretilmesi gereken çıktı değerleri sisteme gösterilmez, sistemin kendisine gösterilen girdilere karşılık çıktısını üretmesi beklenir ve ağa ürettiği çıktının doğru veya yanlış olduğuna dair bilgi verilir. Bu bilgi dikkate alınarak ağ yeniden düzenlenir. LVQ (Vektör Nicemeli Öğrenme) Ağı bu stratejiyi kullanan sistemlere bir örnektir. (Öztemel, 2006)

#### **4.2.4.3 Öğreticisiz öğrenme**

Bu öğrenme yönteminde ağa sadece girdi değerleri gösterilir ve parametreler arasındaki ilişkileri sistemin kendi kendine öğrenmesi beklenir. (Uğur, 2005)

Öğrenme, ağın her bir örneği kendi arasında sınıflandıracak şekilde kendi kurallarını oluşturması şeklinde gerçekleşir. Ağ, ağırlık değerlerini aynı özellikte olan örnekleri ayırabilecek şekilde düzenler. (Kakıcı, 2009) SOM (Özörgütlemeli Harita) Ağları öğreticisiz öğrenme stratejisini kullanmaktadır. (Öztemel, 2006)

#### **4.2.5 Yapay sinir ağlarında öğrenme kuralları**

Yapay sinir ağlarının öğrenmesinde kullanılmak üzere birçok öğrenme kuralı geliştirilmiştir. Bunların çoğunun geliştirilmesinde Hebb öğrenme kuralı temel alınmıştır.

#### **4.2.4.1 Hebb kuralı**

Bilinen en eski (1949) öğrenme kuralı olup diğer öğrenme kurallarının temelini oluşturan Hebb kuralına göre; birbiriyle bağlantılı iki hücre aynı işareti taşıyorsa, yani ikisi de aktif veya ikisi de pasif ise aralarındaki bağlantının kuvvetlendirilmesi gerekir. (Öztemel, 2006)

#### **4.2.4.2 Hopfield kuralı**

Hebb kuralı temel alınarak geliştirilmiş olan Hopfield kuralına göre, beklenen çıktı ve girdilerin ikisi de aktif/pasif ise bağlantıların öğrenme katsayısı kadar kuvvetlendirilmesi/zayıflatılması gerekir. Öğrenme katsayısı genel olarak 0–1 arasında değişen sabit ve pozitif bir değer olacak şekilde kullanıcı tarafından belirlenir. (Öztemel, 2006)

#### **4.2.4.3 Kohonen kuralı**

Bu kurala göre, ağdaki hücreler birbirleri ile yarışır. En büyük çıktıyı üreten hücre yarış kazanır ve hem kazanan elemanın hem de komşuları olan elemanların ağırlıklarını değiştirilir. (Öztemel, 2006)

#### **4.2.4.4 Delta kuralı**

Hebb öğrenme kuralına dayanan başka bir öğrenme kuralı olan delta kuralına göre ağın ürettiği çıktı ile üretmesi beklenen çıktı arasındaki hatanın karelerinin ortalamasını azaltılması amacıyla ağ ağırlık değerlerinin sürekli değiştirilmesi gerekir. (Öztemel, 2006)

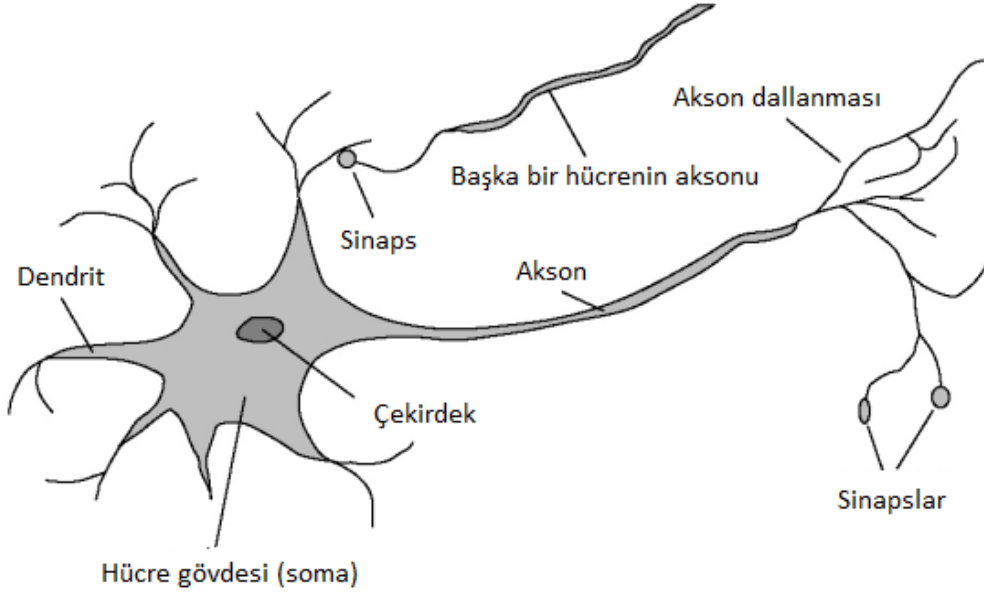
### **4.3 Yapay Sinir Ağlarının Yapısı ve Temel Elemanları**

Yapay sinir ağları, biyolojik sinir sisteminden esinlenilerek geliştirilmiştir. Bu nedenle işleyişinin tam olarak anlaşılabilmesi için önce biyolojik sinir sistemi hakkında bilgi sahibi olmak gerekir.

### 4.3.1 Biyolojik sinir hücresinin yapısı

Beynimizde 10 milyar adet sinir hücresi ve bunlarda altı trilyondan fazla sayıda bağlantının olduğu söylenmektedir. (Öztemel, 2006)

Sinir sistemi, birbiri ile iletişim halinde olan sinir hücrelerinden oluşur. Şekil 4.5'te bir sinir hücresinin sahip olduğu temel elemanlar gösterilmiştir.

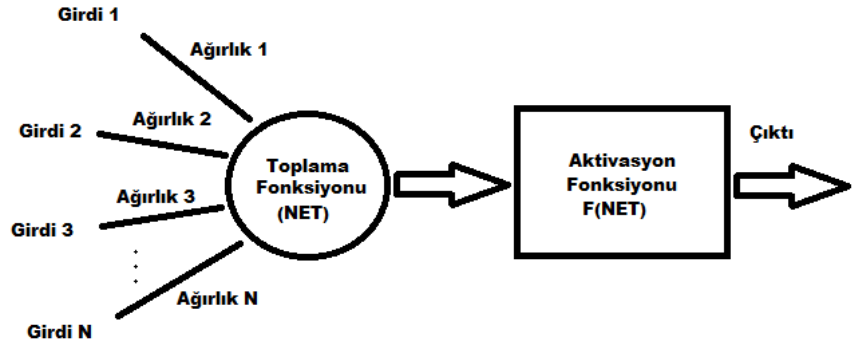


Şekil 4.5 Biyolojik sinir hücresi (Uğur, 2005)

Sinaps'ler, bir sinir hücresinden diğerine elektrik sinyallerinin geçişini sağlayan boşluklardır. İki hücre arasındaki fiziksel olmayan bağlantılar olarak görülebilirler. Sinaps'lerden geçen sinyaller hücre gövdesine giderler. Hücre gövdesi sinyalleri işleme tabi tutar, sinir hücresi kendi elektrik sinyalini oluşturur ve akson aracılığı ile dendrit'lere gönderir. Sinapsler dendritler tarafından sinapslere gönderilerek diğer hücelere iletilir. İki hücre arasındaki bilgi alış veriş, sinaptik bağlantılardaki nörotransmitter maddeler sayesinde sağlanır. Bu özellikte milyarlarca sinir hücresinin bir araya gelerek sinir sistemini oluşturur. Yapay sinir ağlarının geliştirilmesinde biyolojik hücrelerin bu özelliklerinden yararlanır. (Öztemel, 2006)

### 4.3.2 Yapay sinir hücresinin yapısı

Biyolojik sinir hücrelerinden esinlenilerek oluşturulan ve işlem elemanı olarak da adlandırılan yapay sinir hücrelerinin beş temel elemanı vardır. Bunlar şekil 4.6’da gösterildiği gibi girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktıdır. (Öztemel, 2006)



Şekil 4.6 Yapay sinir hücresi

#### 4.3.2.1 Girdiler

Bir yapay sinir hücresine dış dünyadan veya başka hücrelerden gelen bilgilerdir. Bunlar, ağa öğretilmek istenen örnekler tarafından belirlenir. (Öztemel, 2006)

#### 4.3.2.2 Ağırlıklar

Ağlar, öğrenme süresince girdi vektöründeki her bir bilginin önemini temsil eden ağırlıkları belirlemeye çalışırlar. Ağırlıkların değerinin büyük ya da küçük, pozitif ya da negatif olması önemli veya önemsiz olduğu anlamına gelmez ancak sıfır olması etkisinin olmadığını gösterir. (Öztemel, 2006) Ağın öğrendiği bilgiler bu ağırlık değerlerinde dağıtık olarak saklanır. Ağırlıkların güncellenmesi eğitim bitene kadar yapılır, test aşamasında ağırlıklar üzerinde değişiklik yapılmaz. Ağırlıkların ilk değerlerinin atanmasında izlenen yöntem ağın performansını etkileyen kriterlerden biridir.

### **4.3.2.3 Toplama Fonksiyonu**

Hücreye gelen net girdiyi hesaplayan fonksiyondur. Bu işi yapan çeşitli toplama fonksiyonları bulunmaktadır. Her bir işlem elemanının aynı toplama fonksiyonunu kullanabileceği gibi, ağ içinde bazı işlem elemanlarının farklı toplama fonksiyonu kullanması da mümkündür. (Öztemel, 2006) Literatürde sık kullanıldığı gözlenen toplama fonksiyonu örnekleri şunlardır:

#### **Çarpım Fonksiyonu:**

Net girdi:  $\prod_i G_i A_i$

#### **Maksimum Fonksiyonu:**

Net girdi:  $\text{Max}(G_i A_i), i = 1 \dots N$

#### **Minimum Fonksiyonu:**

Net girdi:  $\text{Min}(G_i A_i), i = 1 \dots N$

#### **Çoğunluk Fonksiyonu:**

Net girdi:  $:\sum_i \text{sgn}(G_i A_i)$

#### **Kümülatif Toplam Fonksiyonu:**

Net girdi:  $:\text{Net}_i(\text{eski}) + \sum_i \text{sgn}(G_i A_i)$

En yaygın kullanılan toplama fonksiyonu kümülatif toplamdır.

### **4.3.2.4 Aktivasyon Fonksiyonu**

Hücreye gelen net girdiyi işleyerek üretilecek çıktıyı belirleyen fonksiyondur. Bu fonksiyonların da farklı çeşitleri bulunmaktadır. Bazı ağ modellerinde aktivasyon fonksiyonu olarak türevi alınabilir bir fonksiyonun seçilmesi gereklidir. (Öztemel, 2006) Çok katmanlı algılayıcılar bu modellerden biridir. Literatürde sık kullanıldığı gözlenen toplama fonksiyonu örnekleri şunlardır:

**Sigmoid Fonksiyonu:**

$$F(NET) = \frac{1}{1+e^{-NET}}$$

**Linear Fonksiyon:**

$$F(NET) = NET$$

**Step Fonksiyonu:**

$$F(NET) = \begin{cases} 1 & \text{if } NET > \text{eşik} \\ 0 & \text{if } NET \leq \text{eşik} \end{cases}$$

**Sinüs Fonksiyonu:**

$$F(NET) = \text{Sin}(NET)$$

**Eşik Değer Fonksiyonu:**

$$F(NET) = \begin{cases} 0 & \text{if } NET \leq 0 \\ NET & \text{if } 0 < NET < 1 \\ 1 & \text{if } NET \geq 1 \end{cases}$$

**Hiperbolik Tanjant Fonksiyonu:**

$$F(NET) = (e^{NET} + e^{-NET}) / (e^{NET} - e^{-NET})$$

En yaygın kullanılan aktivasyon fonksiyonu sigmoid fonksiyonudur.

**4.3.2.5 Çıktılar**

Ağın çıktısı aktivasyon fonksiyonu tarafından belirlenir ve dış dünyaya veya girdi olarak tekrar ağa gönderilir. (Öztemel, 2006)

Yapay sinir hücresi ile biyolojik sinir hücresi arasındaki benzerlikler, Çizelge 4.1'de gösterilmiştir.

Çizelge 4.1 Biyolojik sinir ağı ile yapay sinir ağının karşılaştırılması (Elmas, 2003)

<b>Biyolojik Sinir Ağı</b>	<b>Yapay Sinir Ağı</b>
Sinir Sistemi	Sinirsel Hesaplama Sistemi
Sinir	Düğüm (Sinir, İşlem Elemanı)
Sinaps	Sinirsel Bağlantı Ağırlıkları
Dendrit	Toplama Fonksiyonu
Hücre Gövdesi	Aktivasyon Fonksiyonu
Akson	Sinir Çıkışı

### **4.3.3 Yapay sinir ağının yapısı**

Genel olarak hücreler üç katman halinde ve her katman içinde paralel olarak bir araya gelerek ağı oluştururlar. Her katmanda birbirine paralel elemanlar söz konusudur. (Öztemel, 2006)

#### **4.3.3.1 Girdi katmanı**

Bu katmandaki hücrelerin görevi, dış dünyadan bilgileri alarak üzerinde herhangi bir işlem yapmadan ara katmanlara iletmektir. (Öztemel, 2006)

#### **4.3.3.2 Ara katmanlar**

Girdi katmanından gelen bilgilerin işlenerek çıktı katmanına gönderilmesinin gerçekleştirildiği katmanlardır. Bir ağın bir veya birden fazla ara katmanı olabilir. (Öztemel, 2006) Bunun yanı sıra ara katman barındırmayan ağlar da mevcuttur. Ağdaki her bir ara katmanda bulunan nöron sayıları farklı olabilir. Ara katmanların ve bu katmanlardaki nöron sayısının artması hesaplama karmaşıklığının ve süresinin artmasına neden olur ancak aynı zamanda ağın daha karmaşık problemlerin çözümünde de kullanılmasına olanak verir. (Kakıcı, 2009)

### **4.3.3 Çıktı katmanı**

Ara katmanlardan gelen bilgilerin işlenerek ağın girdi katmanına sunulmuş olan girdi seti için üretmesi gereken çıktıyı üretilip dış dünyaya gönderen katmandır. (Öztemel, 2006) Geri beslemeli ağlarda bu çıktılar ağın ağırlık değerlerinin yeniden hesaplanması için de kullanılır. (Kakıcı, 2009)

Yapay sinir ağlarının genel çalışma prensibi, bir girdi setini alarak onları çıktı setine çevirmek olarak açıklanabilir. Ağın kendisine gösterilen girdiler için doğru çıktıları üretecek hale getirilmesi, ağın eğitilmesi sonucunda gerçekleşir. Ağa gösterilecek olan örnekler bir vektör haline getirilir. Bu vektör ağa gösterilir ve ağ bu vektör için gerekli çıktı vektörünü üretir. Ağın parametre değerleri doğru çıktıyı üretecek şekilde düzenlenir. Yapay sinir ağları, herhangi bir girdi vektörünün çıktı vektörüne nasıl dönüştükleri konusunda bilgi vermemeleri nedeniyle *kara kutu* gibi görülebilirler. Sonuçların nasıl oluşturulduğunu açıklama yeteneği olmamasının ağa duyulan güveni sarsmasına rağmen verdiği başarılı sonuçlar sayesinde kendilerine duyulan ilgi sürekli artmaktadır. Açıklama yeteneğinin kazandırılması konusu günümüzde bilim dünyasına büyük katkı sağlayabilecek önemli bir araştırma konusu olarak görülebilir. (Öztemel, 2006)

### **4.3.4 Yapay sinir ağlarında öğrenme ve test**

Yapay sinir ağlarında işlem elemanlarının bağlantılarının ağırlık değerlerinin belirlenmesi işleme ağın eğitilmesi denir. Başlangıçta bu ağırlıklar rastgele atanır. Ağ, kendisine örnekler gösterildikçe doğru çıktıları üretecek ağırlık değerlerini bulmak amacıyla bu ağırlık değerlerini günceller. En doğru ağırlık değerlerinin bulunabilmesi için örnekler ağa defalarca gösterilir. Ağ doğru ağırlık değerlerine ulaştığında örneklerin temsil ettiği olay hakkında genellemeler yapabilme yeteneğine kavuşmasına *ağın öğrenmesi* denir. Ağırlık değerlerinin değişmesi *öğrenme kurallarına* göre yürütülür. (Öztemel, 2006)

Öğrenme olayı iki aşamada gerçekleşir. Bunlardan ilki ağın gösterilen örnek için üreteceği çıktının belirlenmesidir. İkincisi ise bu çıktının doğruluk derecesine göre ağın bağlantılarındaki ağırlıkların değiştirilmesidir. (Öztemel, 2006)

Ağın eğitimi tamamlandıktan sonra performansının ölçülmesi için öğreneme sırasında görmediği örnekler ile yapılan denemelere *ağın test edilmesi* denir. Ağ eğitimde belirlenen ağırlıkları kullanarak çıktı üretir. Test sırasında ağırlıkların değerleri değiştirilmez. Yapay sinir ağlarının bilinen örneklerden belirli bilgileri çıkararak bilinmeyen örnekler hakkında yorum veya genelleme yapabilme yeteneğine *adaptif öğrenme* denir. Ağ daha önce hiç görmediği örneklere kabul edilir doğrulukta cevaplar veriyorsa ağın performansı iyi kabul edilir. Performansın yetersiz sayıldığı durumlarda ağın tekrar eğitilmesi veya eğitim için yeni örneklerin belirlenmesi yoluna gidilir. (Öztemel, 2006)

#### 4.4 Yapay Sinir Ağı Modelleri

Bir yapay sinir ağının modeli, işlem elemanlarının bağlanması sonucu oluşan topoloji, toplama fonksiyonu, aktivasyon fonksiyonu, öğrenme stratejisi ve öğrenme kuralı ile belirlenir. (Öztemel, 2006) Tez projesi kapsamında öğreticili ağ modeli olarak Çok Katmanlı Algılayıcı (MLP), destekleyicili ağ modeli olarak Öğrenen Vektör Nicelendirmesi (LVQ) ve öğreticili ağ modeli olarak Özörgütlemeli Harita (SOM) ağları geliştirilmiş ve bileşik bir ağ tasarımında kullanılmıştır. Bu bölümde bu ağ modellerinin yapıları hakkında bilgiler verilecektir.

##### 4.4.1 Çok Katmanlı Algılayıcı (MLP)

Bir problemin doğrusal olması, çıktıların arasına bir doğru veya doğrular çizerek onları iki veya daha fazla sınıfa ayırmanın mümkün olması anlamına gelmektedir. Yapay sinir ağlarının geliştirilen ilk modelleri yalnızca doğrusal problemleri çözebilmektedir. Bu modeller ile doğrusal olmayan ilişkilerin öğrenilmesi mümkün değildir. 1960'lı yılların sonunda Minsky ve Pappert'in doğrusal olmayan XOR probleminin çözülememesini örnek göstererek yapay sinir ağlarının bilimsel değerinin olmadığını iddia etmesi, yapay sinir ağlarının tarihinde bir duraklama devrine neden olmuştur. Yapay sinir ağları üzerine çalışmaların yok denecek kadar azaldığı bu dönem, 1982 yılında Rumelhart ve arkadaşları tarafından tek katmanlı algılayıcıların çözemediği XOR problemini

özebilen ok Katmanlı Algılayıcılar'ın geliştirilmesi ile sona ermiştir. (Öztemel, 2006)

ok Katmanlı Algılayıcı modeline aynı zamanda **“hata yayma modeli”** ve **“geri yayım modeli”** (ing. back propagation) de denmektedir. Öğrenme yöntemi olarak **“delta öğrenme kuralı”** kullanılır. Ağın beklenen çıktısı ile ürettiği çıktı arasındaki hatayı en aza indirmek amacıyla hatayı ağı yayarak çalışmasını sürdürür. (Öztemel, 2006)

#### **4.4.1.1 MLP ağının yapısı**

MLP, ileriye doğru bağlantılı üç katmandan oluşur:

**1. Girdi Katmanı:** Dış dünyadan gelen bilgileri işlemeden ara katmana gönderir. Her bir işlem elemanının sadece bir girdisi ve bir çıktısı vardır. Bu çıktı bir sonraki katmanda bulunan bütün işlem elemanlarına gönderilir. Girdi katmanındaki her işlem elemanı bir sonraki katmanda bulunan bütün işlem elemanlarına bağlıdır.

**2. Ara Katmanlar:** Girdi katmanından gelen bilgileri işleyerek bir sonraki katmana gönderirler. Ağda birden fazla ara katman ve bir ara katmanda birden fazla işlem elemanı bulunabilir. Ara katmandaki her işlem elemanı bir sonraki katmanda bulunan bütün işlem elemanlarına bağlıdır.

**3. Çıktı Katmanı:** Ara katmandan gelen bilgileri işleyerek ürettiği çıktıları dış dünyaya gönderir. Bir çıktı katmanında birden fazla işlem elemanı bulunabilir. Her işlem elemanı bir önceki katmanda bulunan bütün işlem elemanlarına bağlıdır. Her işlem elemanının sadece bir çıktısı vardır.

#### **4.4.1.2 MLP ağının öğrenme kuralı**

MLP ağları öğretmenli öğrenme stratejisine göre çalışırlar. Ağı eğitim sırasında hem girdiler hem de ağın o girdi için üretmesi eklenen çıktılar gösterilir. Öğrenme kuralı olarak en küçük kareler yöntemine dayalı delta öğrenme kuralının genelleştirilmiş halidir.

Genelleştirilmiş delta kuralı iki aşamadan oluşur:

### 1. İleri doğru hesaplama: Ağın çıktısının hesaplanması

Ara katmandaki işlem elemanları, girdi katmanındaki işlem elemanlarından gelen bilgileri ağırlıklarının etkisi ile alır ve aşağıdaki formül ile net girdi hesaplanır:

$$NET_j^a = \sum_{k=1}^n A_{kj} \cdot \zeta_k^i$$

Elde edilen net girdiler aktivasyon fonksiyonundan geçirilerek ara katmanın net çıktısı hesaplanır.

Çıktı katmanındaki işlem elemanları, ara katmandan gelen net girdinin hesaplanması ve aktivasyon fonksiyonundan geçirilmesi ile ağın çıktısı hesaplanmış olur.

### 2. Geri doğru hesaplama: Ağırlıkların değiştirilmesi.

Her işlem elemanı için ağın ürettiği çıktı ile beklenen çıktı arasındaki farklar hesaplanıp toplanarak ağın toplam hatası bulunur. Toplam hatanın azaltılması için hatanın ağa dağıtılması amacıyla ağırlıkların değiştirilmesi gerekmektedir.

Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi: ara katmandaki j. işlem elemanı ile çıktı katmanındaki m. işlem elemanı arasındaki ağırlığın değiştirilmesi aşağıdaki formüle göre yapılır:

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

Eşitlikteki  $\delta_m$ , m. Çıktı biriminin hatasını gösterir ve aşağıdaki formül ile hesaplanır:

$$\delta_m = f'(NET) \cdot E_m$$

Ara katmanlar arası veya ara katman ile girdi katmanı arasındaki ağırlıkların değiştirilmesi:

Ağın ürettiği çıktı değerindeki hata üzerinde girdi katmanı ile ara katman arasındaki ağırlıkların payı vardır. Bu nedenle bu katmanlar arasındaki ağırlıkların değiştirilmesinde çıktı katmanındaki bütün elemanların hatalarının dikkate alınması gereklidir. Bu ağırlıkların değişim miktarı aşağıdaki formülle hesaplanır:

$$\Delta A_{kj}^i(t) = \lambda \delta_j^a C_k^i + \alpha \Delta A_{kj}^i(t-1)$$

Buradaki hata terimi  $\delta^a$  aşağıdaki formülle hesaplanır:

$$\delta_j^a = f'(NET) \cdot \sum_m \delta_m A_{jm}^a \quad (\text{Öztemel, 2006})$$

#### **4.4.1.2 MLP ağının çalışma prosedürü**

Aşağıdaki adımlar, ağın çıktıları ile beklenen çıktılar arasındaki hatalar kabul edilebilir düzeye inene kadar tekrarlanır.

1. Örnekler toplanır.
2. Ağın topolojik yapısı belirlenir.
3. Öğrenme parametreleri belirlenir.
4. Ağırlıkların başlangıç değerleri atanır
5. Öğrenme setinden örnekler seçilerek ağa gösterilir.
6. Öğrenme sırasında ileri hesaplama yapılır.
7. Ağırlıklar değiştirilir.

#### 4.4.2 Öğrenen Vektör Nicelendirmesi (LVQ)

1984 yılında Kohonen tarafından geliştirilmiş olan LVQ ağının temel felsefesi  $n$  boyutlu bir vektörün referans vektörleri denen bir vektörler setine haritalanmasıdır. Referans vektörleri ağın ağırlık değerlerinden oluşur. LVQ ağının temel görevi, öğrenme yolu ile girdi vektörlerini temsil edecek referans vektörlerinin belirlenmesidir. Genellikle sınıflandırma problemlerinin çözümünde kullanılan LVQ ağının çıktılarından bir tanesi 1 ve diğerleri 0 değeri alır. Bu durum girdinin, 1 değerini alan referans vektörü ile temsil edilen sınıfa ait olduğu anlamına gelir. Eğitim sırasında girdilerin sınıflara ayrılması, *en yakın komşu* kuralına göre gerçekleştirilir. Girdi vektörü ile referans vektörleri arasındaki uzaklıklar karşılaştırılır ve girdi vektörünün kendisine en kısa mesafede bulunan vektör grubuna ait olduğu varsayılır. Ağın ağırlıkları değiştirilerek girdileri doğru sınıflara ayıracak referans vektörlerinin bulunması sağlanır. Çıktı değerlerinin belirlenmesinde *kazanan her şeyi alır* stratejisi uygulanır; yani ağda sadece kazanan vektöre ait ağırlıkları güncellenir. LVQ ağlarında destekleyicili öğrenme yöntemi olduğundan dolayı ağa ürettiği çıktıların gerçek değeri değil sadece doğru olup olmadığı bilgisi verilir ve ağırlıkların güncellenmesi bu bilgiye bağlı olarak yapılır. Ağırlıkların güncellenmesinde kullanılan öğrenme katsayısının düzenli olarak azaltılarak 0'a ulaşması hedeflenir. Ağda ağırlıklar yalnızca girdi ile Kohonen katmanları arasında bulunur.

##### 4.4.2.1 LVQ ağının yapısı

LVQ ağları, girdi katmanı, Kohonen katmanı ve çıktı katmanı olmak üzere üç katmandan oluşur.

**1. Girdi katmanı:** Dış dünyadan gelen örnekler ağa gösterilerek girdi vektörü oluşturulur. İşleme olmaz.

**2. Kohonen Katmanı:** Bu katmandaki her eleman bir referans vektörüdür. Her bir girdi setine en yakın ağırlık vektörü belirlenerek ilgili referans vektörüne haritalama yapılır. Girdi katmanındaki her işlem elemanı kohonen katmanındaki her işlem elemanı arasında bağlantı vardır.

**3. Çıktı Katmanı:** Girdinin ait olduğu sınıf belirlenir. Kohonen katmanındaki işlem elemanlarından her biri çıktı katmanındaki yalnızca bir işlem elemanına bağlıdır.

#### **4.4.2.2 LVQ ağının öğrenme kuralı**

LVQ ağının, kohonen katmanındaki işlem elemanlarının birbiri ile yarışmaları ilkesine dayanan öğrenme kuralına Kohonen öğrenme kuralı denir. Bu yarışma, girdi vektörü ile referans vektörler arasındaki euclid uzaklıklarının hesaplanarak girdi vektörüne en yakın referans vektörünün bulunmasına dayanır.

i. işlem elemanı için Girdi vektörü  $X$  ile referans vektörü  $A$  arasındaki mesafe aşağıdaki şekilde hesaplanır:

$$d_i = \|A_i - X\| = \sqrt{\sum_j (A_{ij} - X_j)^2}$$

Kazanan işlem elemanı doğru sınıfta ise; referans vektörü giriş vektörüne yaklaştırılarak aynı örnek ağa tekrar gösterildiğinde yine aynı işlem elemanının kazanması amaçlanır. Bu işlem aşağıdaki formüle göre yapılır.

$$A_{i+1} = A_i + \lambda(X - A_i)$$

Kazanan işlem elemanı yanlış sınıfta ise; referans vektörü giriş vektöründen uzaklaştırılarak bir daha aynı işlem elemanının kazanmaması amaçlanır Bu işlem aşağıdaki formüle göre yapılır.

$$A_{i+1} = A_i - \lambda(X - A_i)$$

Öğrenme katsayısının 0 değerine ulaşana kadar düzenli olarak azaltılması ile, girdi vektörünün referans vektörüne çok yaklaştığında aksi yönde tekrar uzaklaşmaya başlamadan durmasını sağlamak amaçlanır. (Öztemel, 2006)

#### **4.4.2.3 LVO ağının dezavantajları:**

- Öğrenme katsayısının zamanında 0 değerine ulaşmaması durumunda ağ öğrendiklerini unutabilir.
- Bazı problemlerde sürekli aynı referans vektörü yarışmayı kazanır
- İki sınıfın tam ortasındaki veya sınırlara çok yakın olan vektörlerin sınıflandırması yapılamayabilir.

#### **4.4.2.4 LVO Ağının Çalışma Prosedürü**

1. Örnekler belirlenir
2. Ağın topolojik yapısı belirlenir (girdi ve çıktı sayısı, referans vektör sayısının belirlenmesi)
3. Öğrenme parametreleri belirlenir
4. Ağırlıkların başlangıç değerleri atanır
5. Öğrenme setinden bir örnek ağa gösterilir
6. Kazanan işlem elamanını belirlenir
7. Ağırlıkların değiştirilir
8. Bütün örnekler doğru sınıflandırılana kadar 5-7. adımlar tekrarlanır.  
(Öztemel, 2006)

#### **4.4.3 Özörgütlemeli Haritalar (SOM)**

1984 yılında Teuvo Kohonen tarafından sunulan ileri beslemeli bir model olan SOM, *ilişkisel bellek* modeline dayanmaktadır. İlişkisel bellek, iki nesnenin birbiri ile ilişkilendirilmesi ile öğrenmeyi temel alır. Buna örnek olarak limon-ekşi tat ilişkisi verilebilir. SOM modeli ilişkisel belleğin yanı sıra *kendi kendine öğrenmeyi* de temel almaktadır. SOM yapısında küme adı verilen grup merkezleri bulunur. Popülasyondaki iyi temsilciler seçilebilirse popülasyon bu temsilciler ile ifade edilebilir. (Cinsdikici, 2011)

Özörgütlemeli ağlar genellikle yüksek boyutlu girdilerin daha düşük boyutlu çıktılar ile temsil edilmesi amacıyla kullanılır. (Kohonen vd, 1995)

#### **4.4.3.1 SOM ağıının yapısı**

SOM ağıları, girdi ve çıktı katmanları olmak üzere iki katmandan oluşmaktadırlar.

**1. Girdi katmanı:** Dış dünyadan girdi değerleri alınır.

**2. Çıktı katmanı:** İki boyutlu bir düzlemi, işlem elemanları bu düzlem üzerine dağılmış vektörleri gösterir. (Öztemel, 2006)

#### **4.4.3.2 SOM ağıının öğrenme kuralı**

SOM ağıları, yarışmayı kazanan elemanın 1, diğerlerinin 0 değerini alması ilkesine dayanmaktadır. SOM ağıının eğitilmesinde kazanan elemanın bulunması için iki yol kullanılabilir. Bunlardan ilkinde her elemanın çıktısı ağırlıklarla girdilerin çarpımının toplamı ile bulunur. Bu çıktı değerlerinden en yüksek değere sahip olan işlem elemanı yarışmayı kazanmış olur. İkinci yöntemde ise kazanan, *euclid mesafesi* kullanılarak girdi vektörüne en yakın ağırlık vektörüdür. Kazanan eleman belirlendikten sonra bu eleman ve komşularının ağırlıkları değiştirilir. (Öztemel, 2006)

#### **4.4.3.3 SOM Ağıının Çalışma Prosedürü**

SOM ağ yapısının algoritması 1984 yılında Kohonen tarafından aşağıdaki gibi belirtilmiştir.

**Adım 0:** Ağırlıklara rastgele küçük sayılarla başlangıç değerlerinin atamasını yap.

$$0 < w_i < 1$$

Öğrenme katsayısını ata

$$0 < \mu < 1$$

**Adım 1:** Yeterli adım sayısına ulaşıncaya dek 2-8. Adımlar arasını tekrarla

**Adım 2:** Her bir girdi örüntüsü için 3-5. Adımlar arasını tekrarla

**Adım 3:** Her bir küme (j) için uzaklık hesapla

$$\text{uzaklık}(j) = \sum (w_{ij} - x_i)^2$$

**Adım 4:** Minimum uzaklığa sahip olan j. kümeyi bul.

$$\text{index} = \min\{j, \text{uzaklik}(j)\}$$

**Adım 5:** index'in gösterdiği kümenin ve komşularının ağırlıklarını güncelle.

$$w_{i,\text{index}}(\text{yeni}) = w_{i,\text{index}}(\text{eski}) + \mu * (X_{i,\text{index}} - w_{i,\text{index}}(\text{eski}))$$

Komşuları için ise “r” uzaklık katsayısını gözeterek güncelleme yap.

**Adım 6:** Öğrenme katsayısını hassaslaştır.

$$\mu = \mu * 0.6 \text{ (0.6 rastgele bir sayıdır)}$$

**Adım 7:** Komşuluk derecesini azalt

**Adım 8:** Duruşu kontrol et. (Cinsdikici, 2011)

#### 4.4.4 Bileşik ağ modeli

Birden fazla ağın aynı probleme çözüm üretmek üzere geliştirildiği sistemlere bileşik ağlar denir. Sistemi oluşturan ağların her biri girdileri birbirinden bağımsız olarak işlemekte olup birbirinin kararlarını etkileyecek herhangi bir etkileşimde bulunmaları söz konusu değildir. Bu tür ağların işleyişi, birden fazla uzmanın bir probleme çözüm üretmesine benzetilebilir. Bileşik ağ oluştur ağlardan her biri olayın farklı bir yönünü daha iyi öğrenebilir, böylece hepsinin kararları bir araya getirilerek oluşturulan ortak karardan, ağların tek tek oluşturduğundan daha yüksek performans elde edilir. Bileşik ağ oluşturan ağların tümü aynı ağ modeline sahip olabileceği gibi farklı modeller de olabilirler.

### 4.5 Yapay Sinir Ağı Modeli Geliştirme Süreci

Yapay sinir ağı modelinin geliştirilmesi, birkaç temel noktanın dışında uyulması gereken belli kuralların olmaması ve çoğunlukla deneme yanılma yöntemi kullanılması nedeniyle zor bir iştir. Ağın başarısı büyük oranda geliştiricinin deneyimine bağlıdır. Bir “çok katmanlı algılayıcı” ağının oluşturulmasında temel olarak izlenen adımlar aşağıda belirtilmiştir.

#### 4.5.1 Örneklerin seçilmesi

Öğrenme boyunca ağ ağırlıkları girdi olarak kullanılan örnekler dikkate alınarak değiştirildiğinden dolayı örneklerin seçilmesi ağın performansını

yakından etkileyen bir konudur. Oluşturulan girdi çıktı setlerinin bir kısmı eğitim, bir kısmı test verileri olarak kullanılır. Ağın daha önce karşılaşmadığı senaryolara ilişkin doğru yorum yapabilmesi için, ağın eğitiminde kullanılan örneklerin, problem uzayının tümünü temsil edebilecek şekilde seçilmesi gerekmektedir. (Öztemel, 2006)

#### **4.5.2 Eğitim ve test setlerinin belirlenmesi**

Yapay sinir ağı uygulamalarında genellikle veriler genellikle eğitim ve test seti olarak ikiye ayrılır, ancak bazı çalışmalarda aşırı uyum probleminden kaçınmak ve eğitim sırasında durma noktasını belirlemek için onaylama seti olarak adlandırılan üçüncü bir set kullanılmıştır. Verilerin ayrılmasında kesin bir oran bulunmamaktadır. Problemin özellikleri, verilerin türü ve eldeki veri sayısı gibi farklı faktörler göz önünde bulundurularak bir oran belirlenebilir. Örneğin; Yao ve Tan verileri %70'i eğitim, %20'si onaylama, %10 ise test seti olarak ayırırken, C. Ntongo ve M. Boyd verilerin %90'ını eğitim % 10'unu ise test için ayırmıştır. Bir başka çalışmada ise verilerin yaklaşık %89'u eğitim %11 ise test için ayrıştırılmıştır (Akdağ, 2010)

#### **4.5.3 Girdi ve çıktıların sayısal gösterimi**

Ağın girdileri işleyebilmesi için değerlerin tümünün sayısal olması gereklidir. Bu nedenle sayısal olmayan değerlerin sayısal değerlerle temsil edilmesi sağlanmalıdır. Girdileri sayısal gösterime çevirmek için kullanılan yöntem ağın performansını etkileyebilir.

Ağın çıktı değerleri ile beklenen değerler arasındaki hatanın bulunabilmesi için çıktı değerlerinin de sayısal olarak temsil edilmesi gereklidir. Bazı durumlarda ağın çıktıları sayısal olsa bile bunların ikili sistemde ifade edilmesi daha uygun olabilir. (Öztemel, 2006)

#### **4.5.4 Ağırlıkların başlangıç değerlerinin belirlenmesi**

Ağırlıkların başlangıç değerleri, ağın performansını büyük oranda etkileyebilmektedir. Genellikle belirli bir aralık içinde rastgele değerler atanarak

belirlenirler. Bu aralığın fazla büyük olması ağın yerel çözümler arasında dolaşıp global çözümü bulamamasına, fazla küçük olması ise öğrenmenin çok uzun sürmesine neden olur. Literatürdeki geçmiş çalışmalarda bu -1.0 ile 1.0 arasında değerlerin başarılı sonuç verdiğini ortaya koymuştur. (Öztemel, 2006) Ancak öğretilen probleme bağlı olarak hangi aralığın ağa en yüksek performansı sağlayacağını deneme yoluyla bulunması gereklidir.

#### **4.5.5 Katsayıların belirlenmesi**

Ağın performansını en yüksek oranda etkileyecek bileşenlerden biri, katsayıların belirlenmesidir. Bu değerlerin belirlenmesi için bir kural bulunmamaktadır. Her problem için geçerli standart bir değer olmadığından tasarımcı kendi tecrübelerine ve değerlendirmelerine dayanarak probleme bağlı uygun değerleri tespit eder. Çok katmanlı algılayıcıların kullandığı katsayılar, öğrenme ve momentum katsayılarıdır. Farklı ağ modellerinde belirlenmesi gereken farklı katsayılar olacaktır.

##### **4.5.5.1 Öğrenme katsayısı ( $\lambda$ )**

Ağırlıkların değişim miktarını belirler. Genellikle belirli bir aralık içinde rastgele değerler atanarak belirlenirler. Bu aralığın fazla büyük olması ağ yerel çözümler arasında dalgalanmaya (osilasyon) neden olabilir, fazla küçük olması ise öğrenmenin çok uzun sürmesine neden olur. Literatürdeki geçmiş çalışmalarda 0.2 ile 0.4 arasındaki değerlerin seçilmesinin başarılı sonuç verdiğini ortaya koymuştur. (Öztemel, 2006) Ancak öğretilen probleme bağlı olarak hangi katsayının ağa en yüksek performansı sağlayacağını deneme yoluyla bulunması gereklidir.

##### **4.5.5.2 Momentum katsayısı ( $\alpha$ )**

Bir önceki adımdaki değişimin hangi oranda yeni değişim miktarına ekleneceğini belirler. Bu katsayının küçük olması yerel çözümler arasında dolaşma riskini doğururken büyük olması tek bir çözüme ulaşmada sorunlara neden olur. Literatürdeki geçmiş çalışmalarda 0.6 ile 0.8 arasındaki değerlerin seçilmesinin başarılı sonuç verdiğini ortaya koymuştur. (Öztemel, 2006)

#### **4.5.6 Örneklerin ağı sunuluş şekli**

Örnekler genel olarak ağı sıralı veya rastgele şekilde sunulabilmektedir.

##### **4.5.6.1 Sıralı sunum**

Örnekler ağı tek tek sıra ile sunulurlar. Tüm örneklerin sunulması bittiğinde başa dönülerek aynı sıra ile tekrar sunulurlar. Bu işlem ağın öğrenmesi tamamlanana kadar tekrarlanır. Bu tür sunuşta her bir örneğin ağı sunulma şansı eşittir.

##### **4.5.6.2 Rastgele sunum**

Örnekler ağı rastgele seçilerek sunulurlar. Bu sunum şeklinde de iki yöntem vardır. Birincide seçilen örnekler veri setinde tutulmaya devam ederler ve tekrar seçilebilirler. Bu durumda örneklerin seçilme şansları eşit değildir. İkinci durumda ise seçilen örnekler bütün veri seti ağı gösterilene kadar veri setinden çıkarılır ve sonraki örnek kalanlar arasından seçilir. Bu durumda örneklerin seçilme şansları eşittir. (Öztemel, 2006)

#### **4.5.7 Ağırlıkların değiştirilme zamanı**

Ağırlıkların değiştirilme zamanı öğrenme kuralı tarafından belirlenmiştir. Üç seçenek vardır:

- Ağı gösterilen her örnekten sonra hata bulunur ve ağırlıklara dağıtılır.
- Ağı belli bir sayıda örnek tek tek gösterildikten sonra toplam hata hesaplanır ve ağırlıklara dağıtılır.
- Tüm örnekler ağı gösterildikten sonra tüm hata toplanıp ağırlıklara dağıtılır. Örnek sayısının az olduğu problemlerin çözümünde bu yöntem kullanılabilir. (Öztemel, 2006)

#### 4.5.8 Girdi ve çıktıların ölçeklendirilmesi

Problemin örnek veri setinde çok büyük veya çok küçük birkaç uç değerin bulunması durumunda, ağ optimizasyon işlemini gerçekleştiremeyebilir. (Akdağ, 2010) Veri setindeki değerlerin birbirinden farklı ölçeklere sahip olmasının ağı yanlış yönlendirmesinin engellenmesi için girdi ve çıktıların tek ölçekte incelenmesi amacıyla belli bir aralıkta ölçeklendirilmesi gereklidir. Literatürdeki geçmiş çalışmaların çoğunda bu ölçeklendirme 0-1 arasında yapılmıştır.

Ağın oluşturacağı çıktıların ölçeği, ağ tasarımında kullanılan aktivasyon fonksiyonuna bağlıdır. Örneğin sigmoid fonksiyonunda yalnızca 0-1 arasında çıktılar üretilir. Ağın çıktılarının bu aralığın dışında değerler olması gerekiyorsa çıktı değerlerinin ölçeklenmesi gerekir. (Öztemel, 2006)

#### 4.5.9 Durdurma kriterleri

Ağın eğitime gereğinden fazla devam etmesi, doğru sonuçları veren ağırlıkların bulunduktan sonra kaybedilmesine neden olur. Bunun engellenmesi için ağın eğitiminin ne zaman tamamlanmış sayılacağını belirten bir kriter belirlenmelidir Genel olarak iki tür durdurma kriteri kullanılır:

- Hatanın belli bir değerin altına düşmesi

Kabul edilebilir hata değeri, problemin niteliğine bağlı olarak tasarımcı tarafından belirlenir. Tüm örneklerin hatalarının ortalama değeri bu değerin altına düştüğü zaman eğitim sonlanır.

- Ağın önceden belirlenen adım sayısına ulaşması

Kabul edilebilir hata değerinin belirlenemediği durumlarda ağın hata grafiği izlenerek ağın eğitiminin tamamlanması için gereken adım sayısı tespit edilir. (Öztemel, 2006)

C. Ntongo ve M. Boynd iki yöntemi birlikte kullanarak, ağın test aşamasında %2 veya daha az bir oranda hata verdiğinde ya da 4000 adım sayısına ulaştığında, eğitimi sonlandırmışlardır. (Akdağ, 2010)

#### **4.5.10 Ađın topolojisinin (ara katman ve nöron aayısının) belirlenmesi**

Tasarımcılar tarafından tecrübelerle dayanarak yapılır. Sürekli daha iyi performansa ulaşmak için farklı sayılarda ara katman ve işlem elemanları ile ve denemeler yapılır. Geliştirilen bir ađın işlem elemanı sayısını sürekli artırmak veya azaltmak ve oluşan hata grafiđini gözlemlemek, en uygun ađ topolojisinin belirlenmesinin bir yöntemidir. (Öztemel, 2006) Çođu problem için bir gizli katman veya en çok iki gizli katmanın yeterli olacađı konusunda ortak noktaya varılmıřtır. (Akdađ, 2010)

## 5. GELİŞTİRİLEN UYGULAMA

Java dilinde geliştirilen tez uygulaması, veri işleme, ağ tasarımı ve ağ birleştirme olmak üzere üç görsel kullanıcı ara yüzünden oluşmuştur. Proje kapsamında üç adet yapay sinir ağı modelinin gerçekleştirimi yapılmıştır. Bu modeller;

MLP: Çok Katmanlı Algılayıcı (Multilayer Perceptron )

LVQ: Öğrenen Vektör Nicelendirmesi (Learning Vector Quantization)

SOM: Özörgütlemeli Harita (Self Organizing Map)

Uygulamanın temel işlevleri;

Verilerin hazırlanması

Ağın tasarlanması

Ağın eğitilmesi

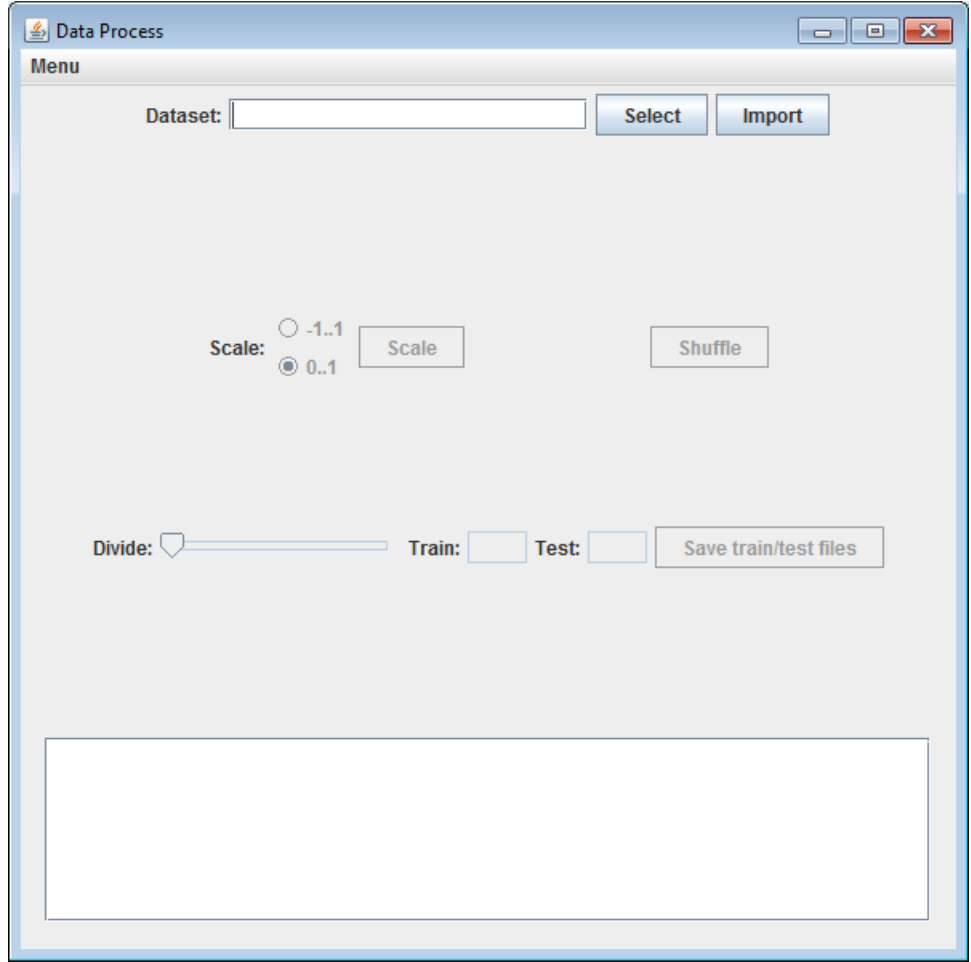
Ağın kaydedilmesi

Ağın test edilmesi

Ağların birleştirilmesi

### 5.1 Veri İşleme Arayüzü

Uygulamanın bu kısmında, verilerin ağın işleyebileceği şekilde hazırlanarak eğitim ve test veri setleri olarak ayrılması sağlanır. Şekil 5.1'de veri işleme arayüzü görülmektedir.

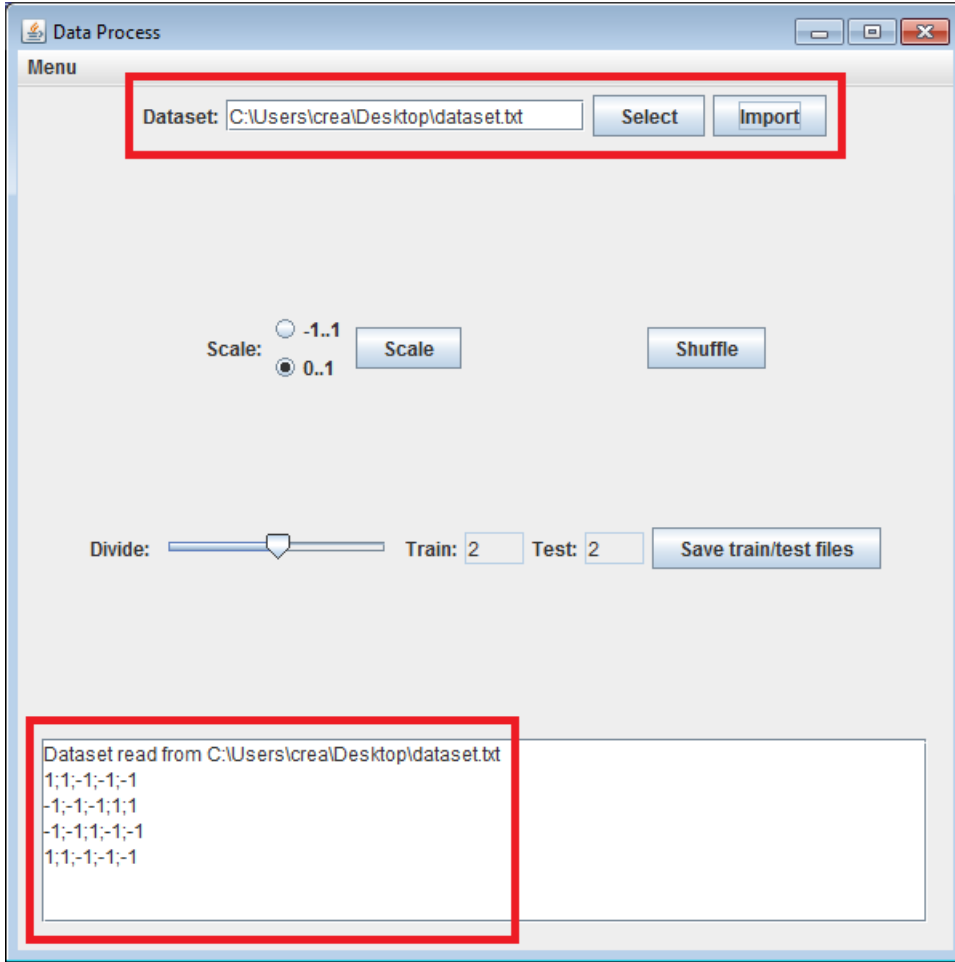


Şekil 5.1 Veri işleme arayüzü

### 5.1.1 Veri setinin yüklenmesi

Ağın kullanılacağı olaya ilişkin sayısal formattaki bilgilerin bulunduğu veri seti dosyası, ağda kullanılmaya hazır hale getirilmek üzere seçilir ve yüklenir.

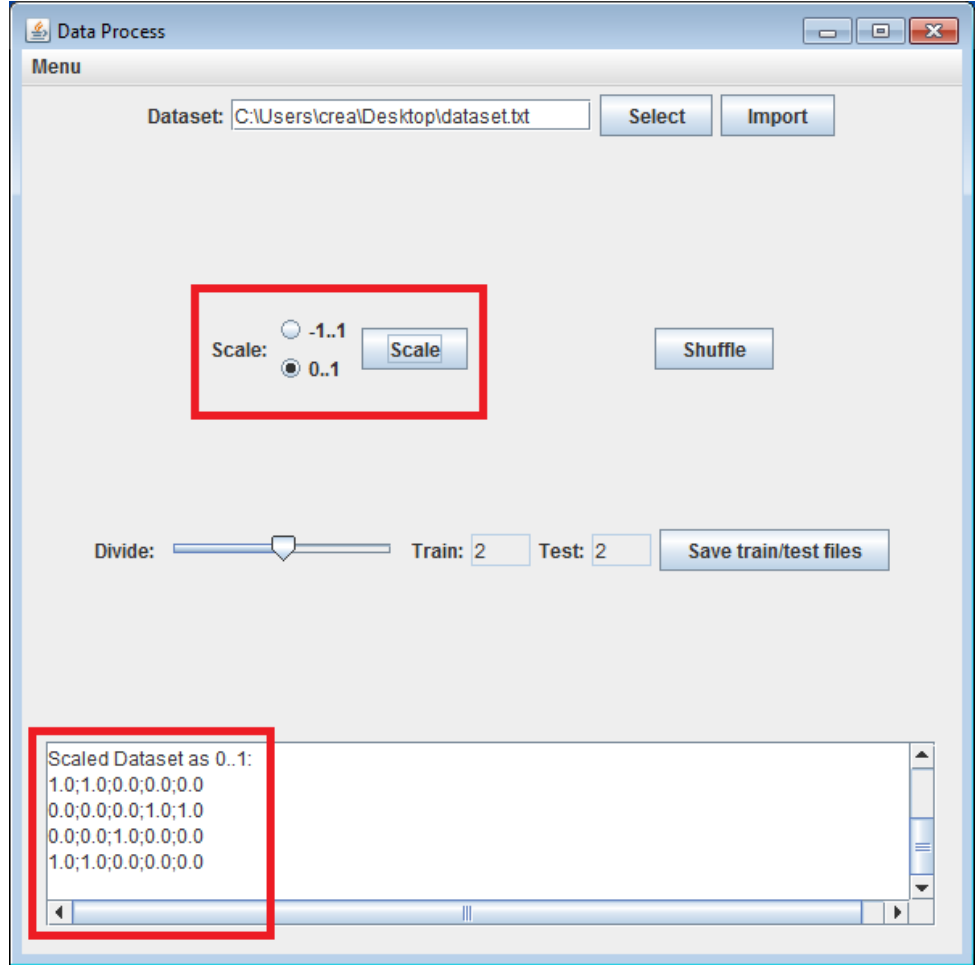
Veri setindeki her satır bir deseni (ing. pattern), ilk dört rakam girdi değerlerini, son rakam çıktı değerini temsil eder. Şekil 5.2’de Küçük bir örnek veri setinin yüklenmesi gösterilmektedir.



Şekil 5.2 Küçük bir örnek veri setinin yüklenmesi

### 5.1.2 Veri setinin ölçeklendirilmesi

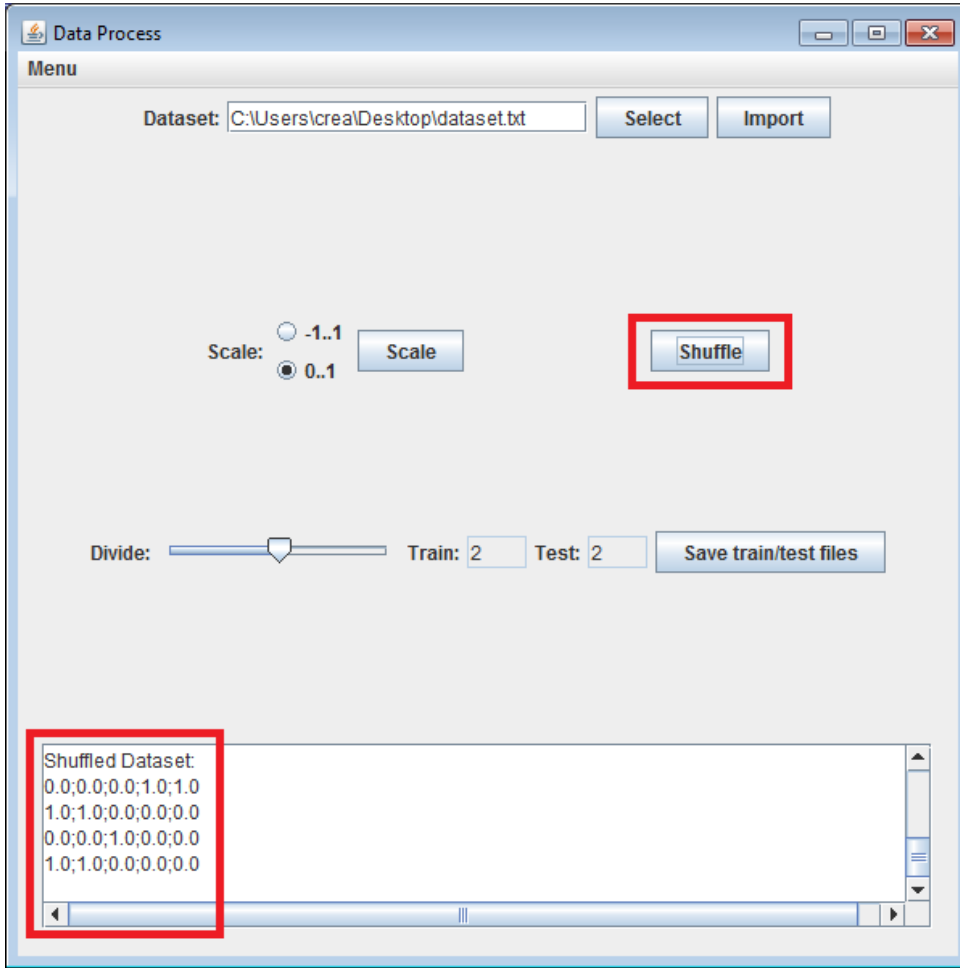
Yüklenen veri setinde bulunan veriler birbirinden bağımsız ölçeklerde olabileceği için, ağa girdi olarak verilmeden önce sabit sınırlar arasına çekilmeleri gerekir. Uygulamada -1-1 veya 0-1 arasında veri ölçeklendirmesi yapılabilmektedir. Şekil 5.3'te küçük bir örnek setindeki -1 ile 1 arasındaki verilere ölçekli olarak 0 ile 1 arasında temsili değerler verildiği görülmektedir.



Şekil 5.3 Küçük bir örnek veri setinin ölçeklendirilmesi

### 5.1.3 Veri setinin karıştırılması

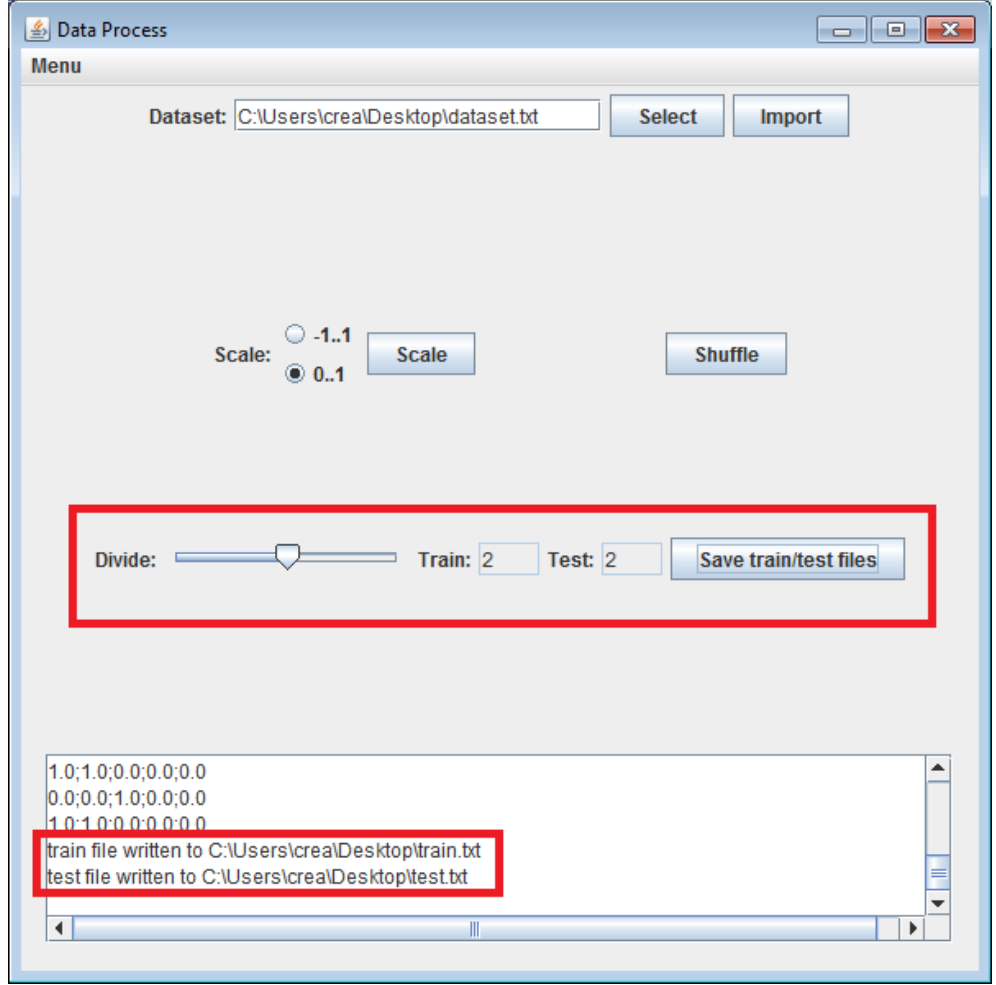
Veri setinin eğitim ve test verileri olarak ayrılmasında dağıtımın tamamen rastgele yapılması önemlidir. Eğitim sırasında kullanılan verilerin olayın tamamını kapsayan örnekler içermesi gereklidir. Veri setinin belli bir düzen içinde olduğu durumlarda, ağırlık eğitiminin bazı noktalara yoğunlaşarak bazılarında eksik kalmasının engellenmesi, karıştırma işlemi ile sağlanır. Şekil 5.4'te küçük bir örnek veri setinin karıştırılması gösterilmektedir.



Şekil 5.4 Küçük bir örnek veri setinin karıştırılması

### 5.1.4 Veri setinin bölünmesi

Veri setinin ölçeklendirilip karıştırıldıktan sonra ne kadarının eğitim, ne kadarının test verisi olarak kullanılacağını belirlemek ve bölünen veri setlerinin iki ayrı dosya olarak kaydedilmesi bu kısımda sağlanır. Şekil 5.5'te küçük bir örnek veri setinin bölünmesi gösterilmektedir.

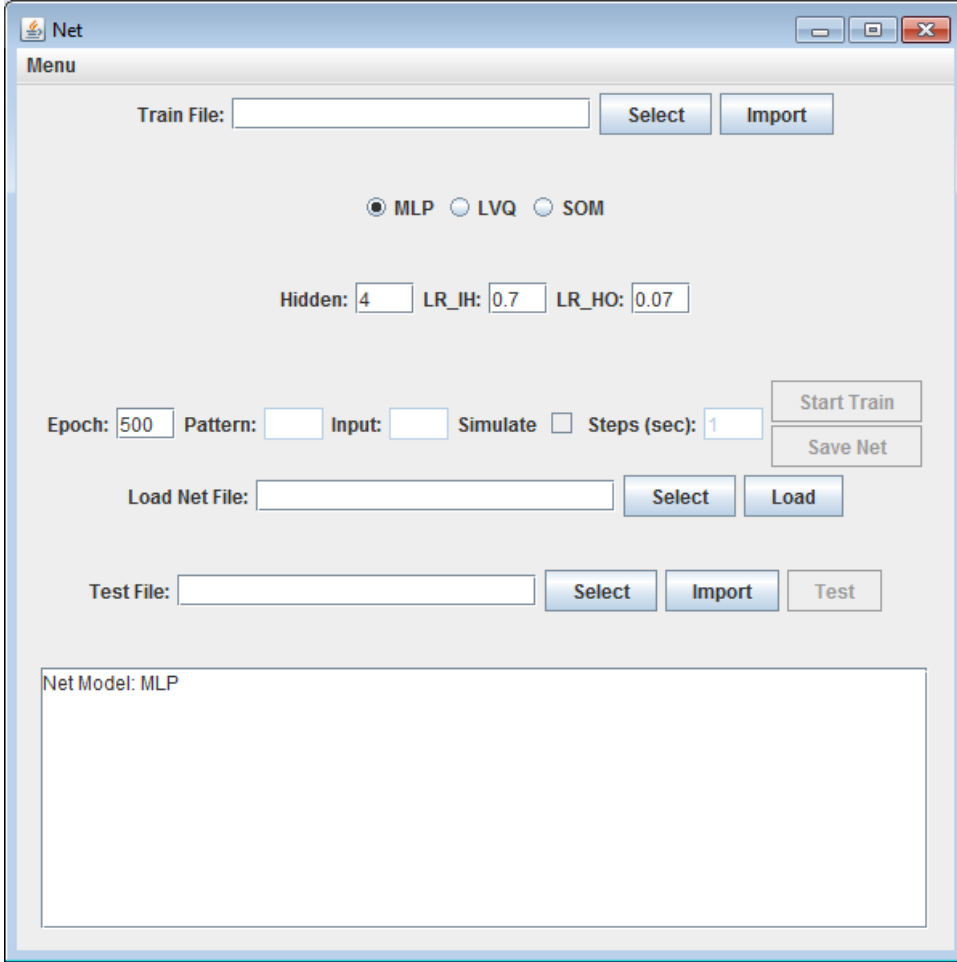


Şekil 5.5 Küçük bir örnek veri setinin bölünmesi

### 5.2 Ağ Arayüzü

Uygulamanın bu kısmında, oluşturulan eğitim ve test veri setleri kullanılarak ağın eğitilmesi ve test edilmesi gerçekleştirilir. Eğitilen ağın kaydedilmesi ve daha sonra tekrar yüklenerek test için kullanılması mümkündür. Eğitim ve test süreçlerinde gerçekleşen olayların adım adım izlenebilmesine

imkan sađlayan opsiyonel bir simülasyon ekranı mevcuttur. Şekil 5.6'da ađ arayüzü görölmektedir.

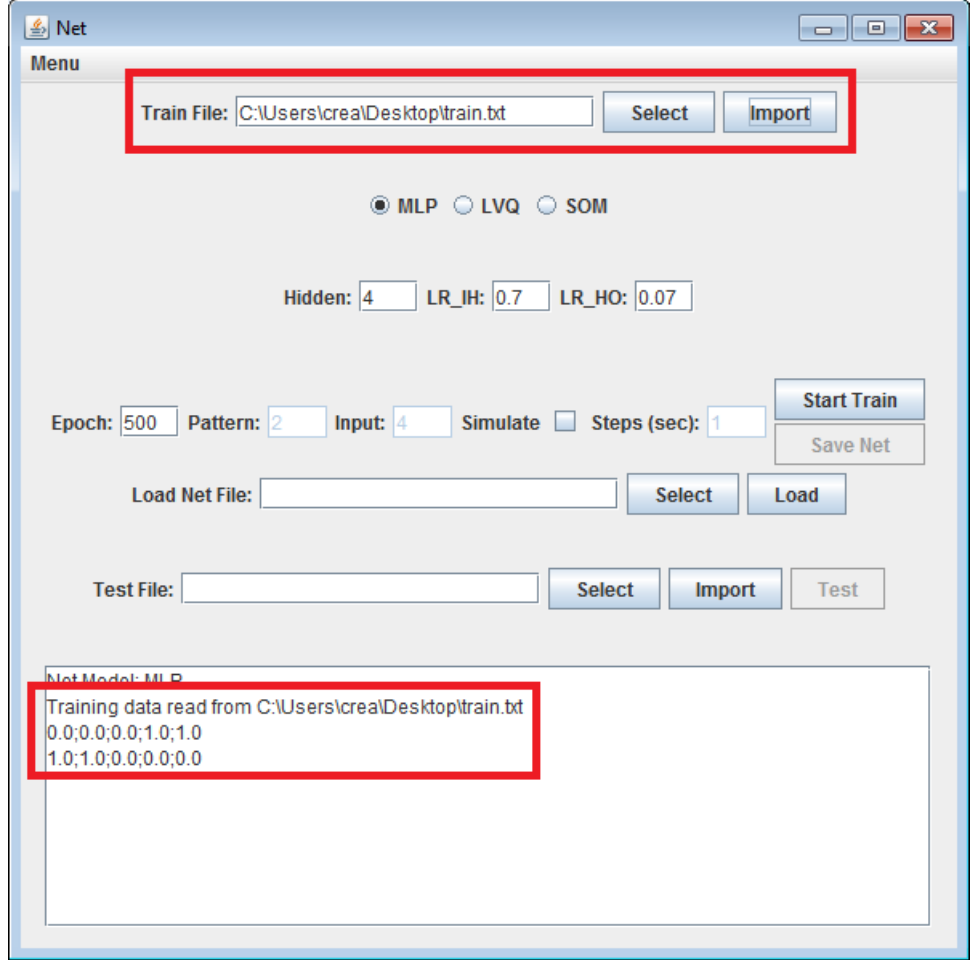


The image shows a software window titled "Net" with a "Menu" header. The interface includes several input fields and buttons for configuring a neural network simulation. At the top, there is a "Train File:" field with "Select" and "Import" buttons. Below this, three radio buttons are present: "MLP" (selected), "LVQ", and "SOM". Further down, there are input fields for "Hidden: 4", "LR\_IH: 0.7", and "LR\_HO: 0.07". A row of controls includes "Epoch: 500", "Pattern:" (empty), "Input:" (empty), a "Simulate" checkbox (unchecked), and "Steps (sec): 1". To the right of these are "Start Train" and "Save Net" buttons. Below this row is a "Load Net File:" field with "Select" and "Load" buttons. At the bottom, there is a "Test File:" field with "Select", "Import", and "Test" buttons. A large text area at the bottom left displays "Net Model: MLP".

Şekil 5.6 Ađ arayüzü

### 5.2.1 Eğitim veri setinin yüklenmesi

Ağın işleyebileceği formatta hazırlanmış olan eğitim verilerinin bulunduğu dosya seçilir ve yüklenir. Şekil 5.7’de küçük bir örnek eğitim veri setinin yüklenmesi gösterilmektedir.



Şekil 5.7 Küçük bir örnek eğitim veri setinin yüklenmesi

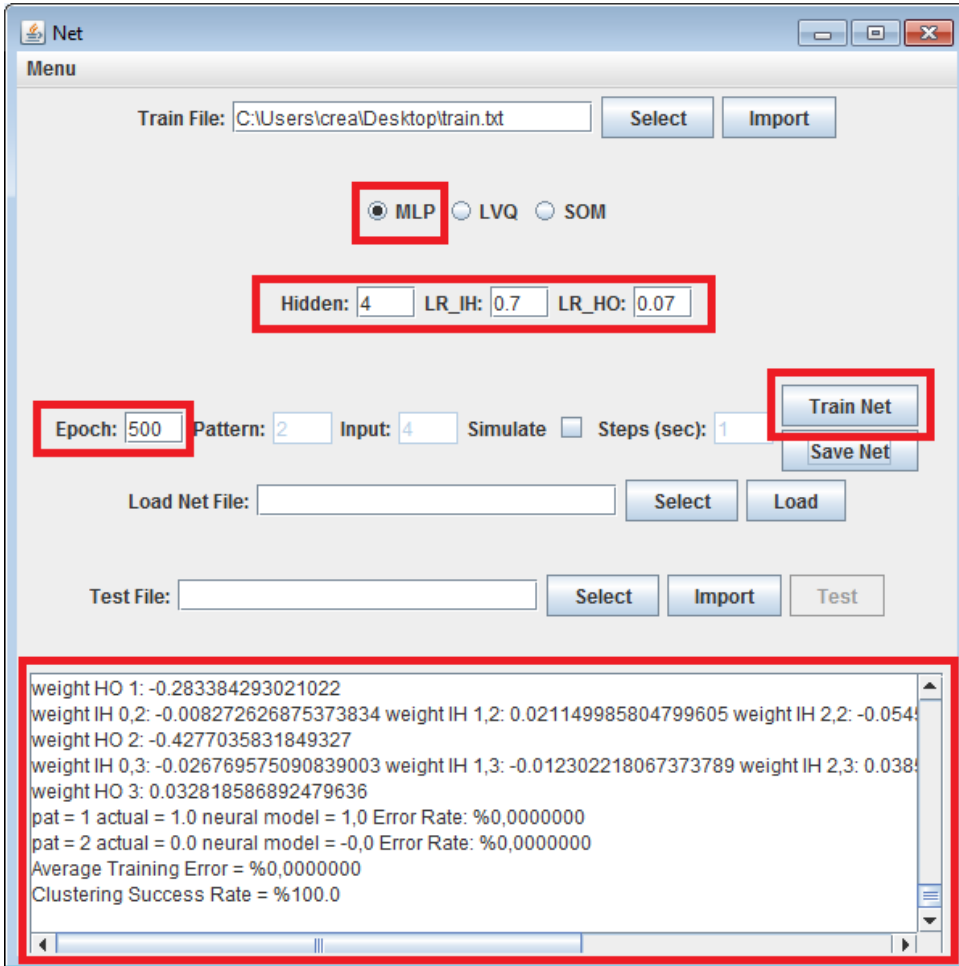
### 5.2.2 Ağın tasarımının yapılması ve eğitilmesi

Ağ arayüzünde üç adet yapay sinir ağı modelinin tasarımının yapılması mümkündür. Tasarlanacak model seçildiğinde tasarım ekranı seçilen modele ilişkin parametrelerin girilmesine imkân sağlayacak şekilde değişmektedir. Pattern (Desen) ağa verilecek örnek eğitim verisi sayısını, girdi ise her bir örnekte bulunan girdi değeri sayısını temsil ettiği için bu alanlar yüklenen eğitim veri seti

tarafından belirlenir. Epoch (Devir) değeri tüm örneklerin ağı gösterilmesi işleminin tekrar sayısını temsil eder ve kullanıcı tarafından belirlenir.

### 5.2.2.1 MLP ağının tasarımı

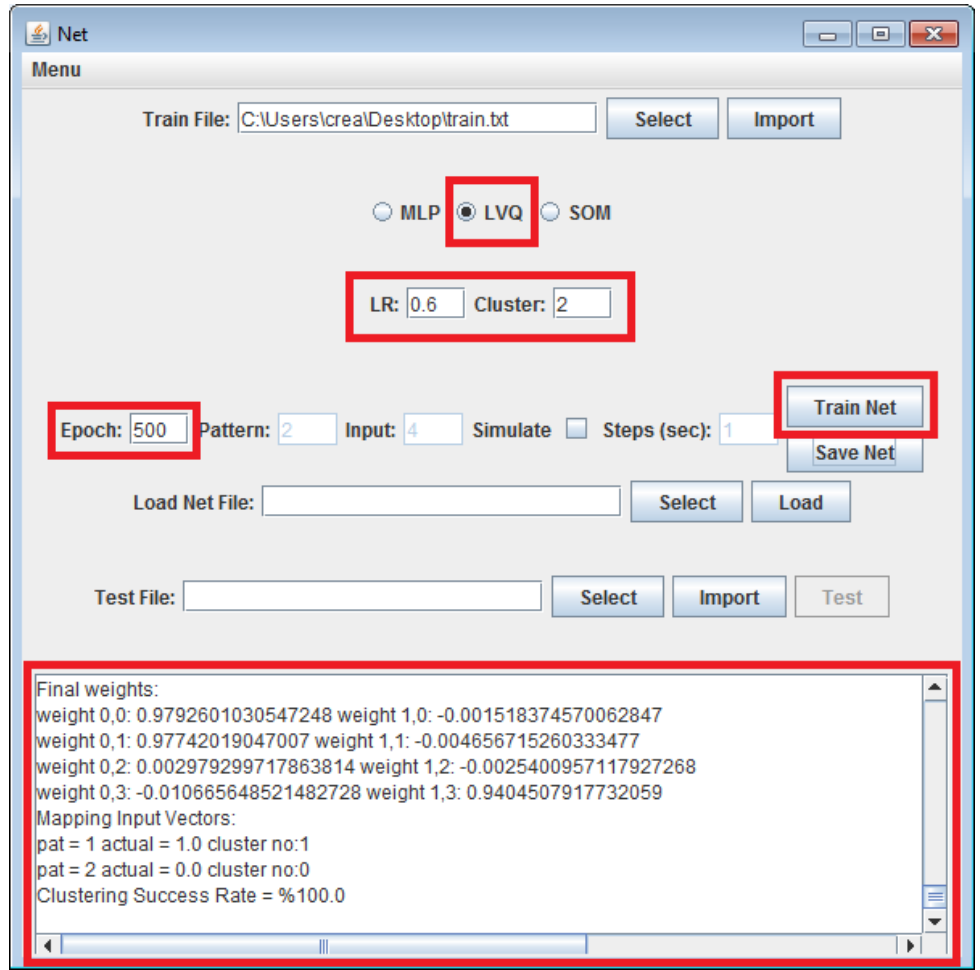
Hidden (Gizli) değeri girdi katmanı ile çıktı katmanı arasındaki gizli katmanda bulunacak nöron sayısını, LR\_IH girdi katmanı ile gizli katman arasındaki ağırlıkların öğrenme katsayısını, LR\_HO ise gizli katman ile çıktı katmanı arasındaki ağırlıkların öğrenme katsayısını temsil eder. Bu bilgiler girilerek ağın tasarımı tamamlandığında, oluşturulan ağın yüklenen eğitim verileri ile eğitimi başlatılır. Şekil 5.8’de MLP ağının eğitilmesi gösterilmektedir.



Şekil 5.8 MLP ağının tasarımı

### 5.2.2.2 LVQ ağının tasarımı

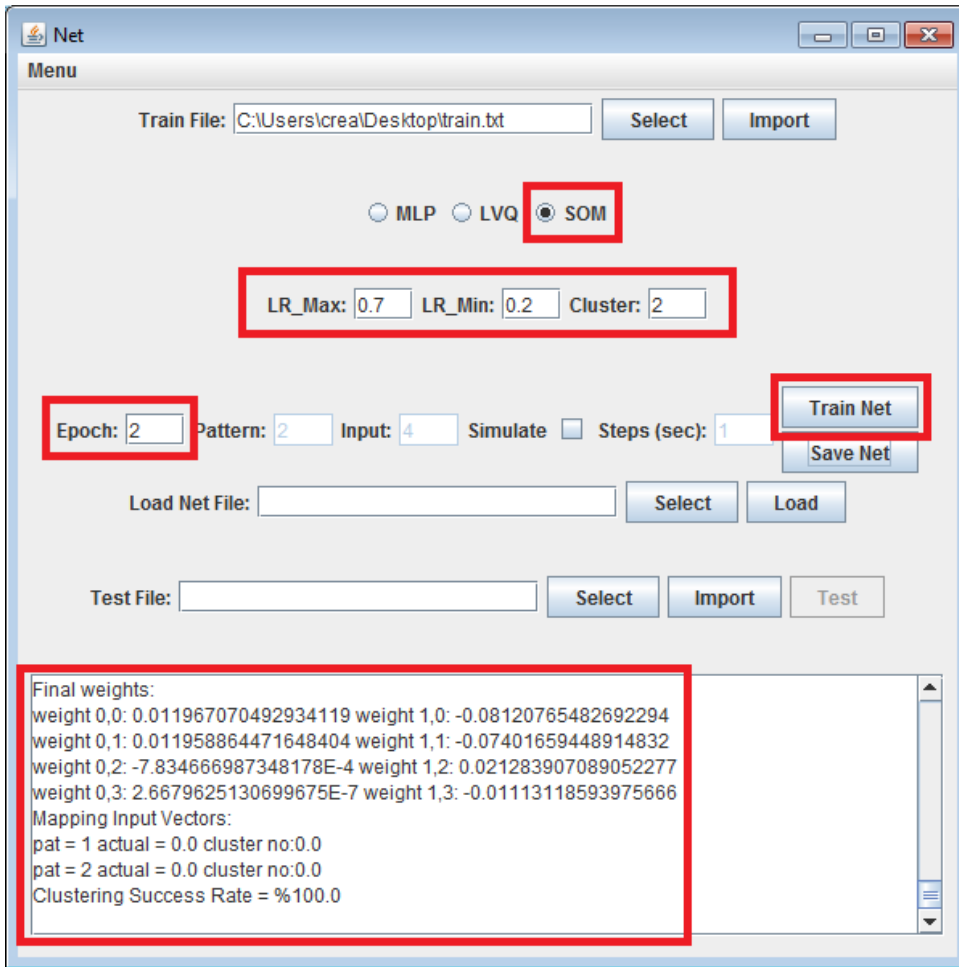
LR ağın öğrenme katsayısını, “Cluster” ise girdi desenlerinin öbeklenmesi sonucu oluşacak öbek sayısını temsil eder. Bu bilgiler girilerek ağın tasarımı tamamlandığında, oluşturulan ağın yüklenen eğitim verileri ile eğitimi başlatılır. Ağın bir eğitim desenini atadığı öbeğin doğru olup olmadığı, eğitim girdilerinin çıktı değeri ile karşılaştırma yapılarak tespit edilir ve buna göre ağırlıklar değiştirilir (Destekleyicili öğrenme). Şekil 5.9’da LVQ ağının eğitilmesi gösterilmektedir.



Şekil 5.9 LVQ ağının tasarımı

### 5.2.2.3 SOM ağının tasarımı

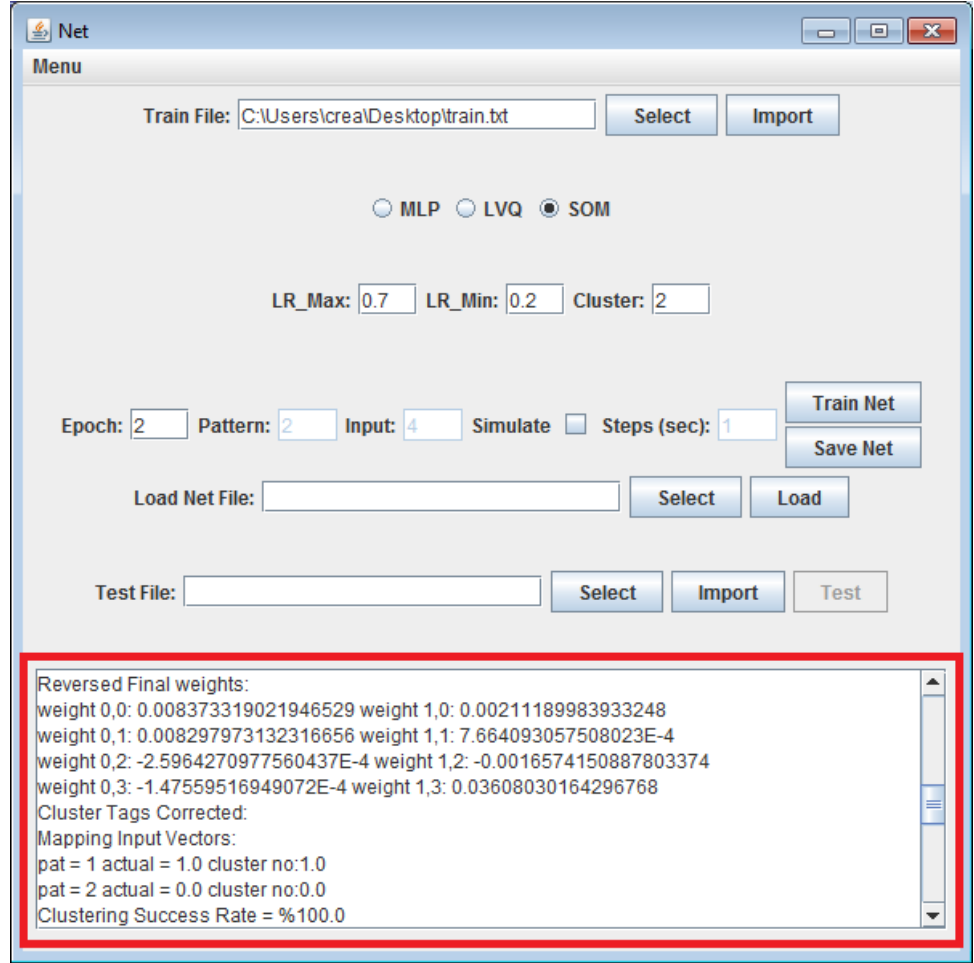
LVQ ağında olduğu gibi LR ağın öğrenme katsayısını, “Cluster” ise girdi desenlerinin öbeklenmesi sonucu oluşacak öbek sayısını temsil eder. Bu bilgiler girilerek ağın tasarımı tamamlandığında, oluşturulan ağın yüklenen eğitim verileri ile eğitimi başlatılır. Eğitim desenlerini öbekleme işlemi süresince eğitim girdilerine karşılık gelen çıktı değeri ağa gösterilmemektedir. (Öğreticisiz öğrenme). Şekil 5.10’da SOM ağının eğitilmesi gösterilmektedir.



Şekil 5.10 SOM ağının tasarımı

MLP ve LVQ ağlarının oluşturduğu öbekler, bu öbeklerin barındırdığı desenlerin gerçek çıktıları (0 veya 1) ile etiketlenmiştir. SOM Ağının oluşturduğu öbekler ise eğitim sırasında desenlerin gerçek çıktıları bilinmediğinden dolayı

etiketlenmemiştir. SOM ağının bileşik ağ sisteminde diğer ağların oluşturduğu çıktılarla uyumlu çıktılar üretmesinin sağlanması için eğitim tamamlandıktan sonra gerçek çıktılar ve ağın oluşturduğu öbekler karşılaştırılır ve eğer ağın öbeklere rastgele atadığı id'ler ile çıktı değerleri zıt ise sistemde öbek id'sinin index olarak kullanıldığı veri yapılarında gerekli değişiklikler yapılır. Şekil 5.11'de bu duruma bir örnek gösterilmiştir:



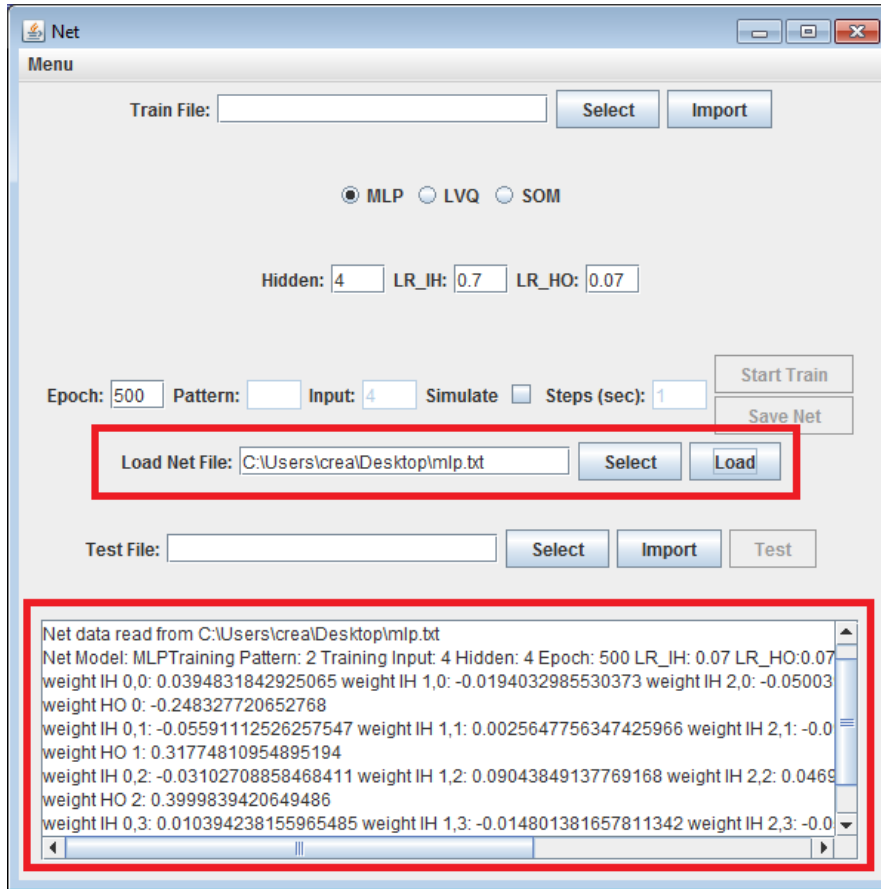
Şekil 5.11 SOM ağının oluşturduğu öbeklerin etiketlerinin düzeltilmesi

### 5.2.3 Eğitilmiş ağı yüklenmesi

Test aşamasından önce ağı eğitilmesi yerine daha önce eğitilmiş bir ağı kaydı yüklenerek test işlemi bu ağı üzerinde yapılabilir. Şekil 5.12’de ağı yükleme işlemi gösterilmektedir. Kayıtlı ağı dosyasının içinde bulunan değerler, ağı modeline göre değişiklik gösterir:

MLP ağı dosyasında; ağı modeli, devir sayısı, desen sayısı, girdi sayısı, gizli katmanda bulunan nöron sayısı, girdi katmanı-gizli katman için öğrenme katsayısı, gizli katman-çıkış katmanı için öğrenme katsayısı, girdi katmanı-gizli katman için ağırlık değerleri, gizli katman-çıkış katmanı için ağırlık değerleri bulunur.

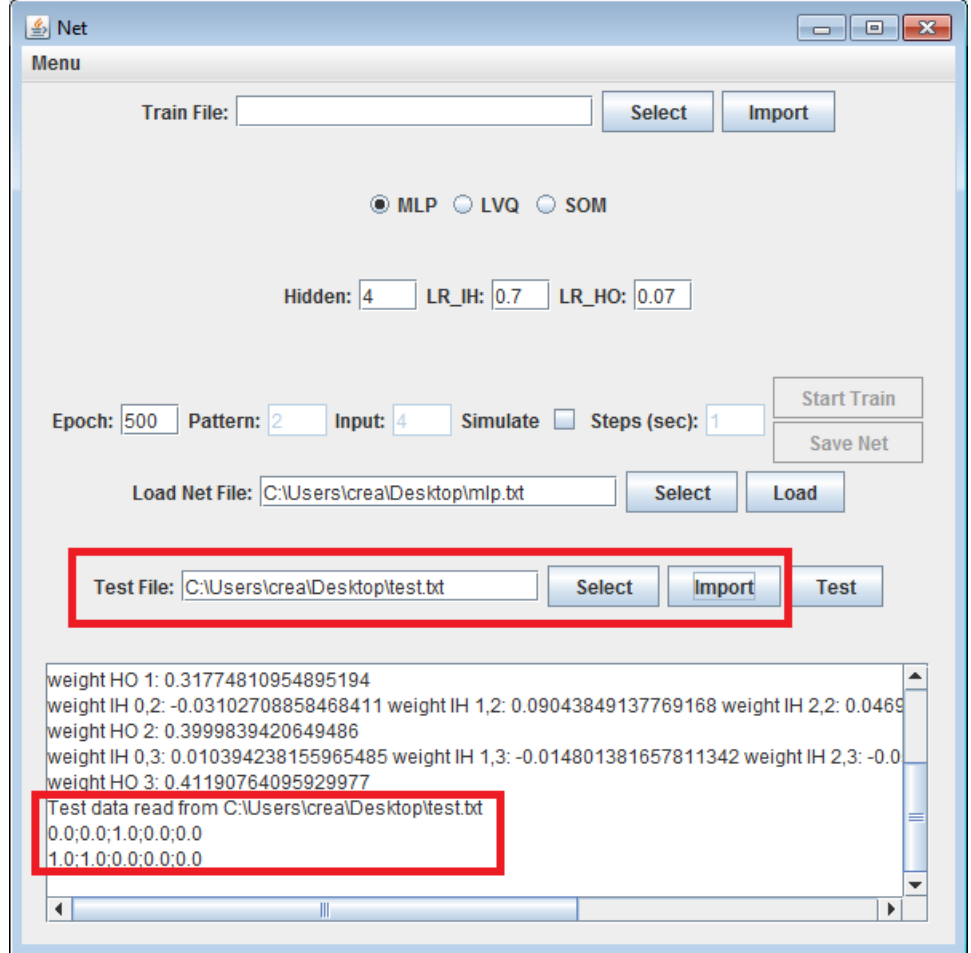
LVQ ve SOM ağı dosyalarında; ağı modeli, devir sayısı, desen sayısı, girdi sayısı, öğrenme katsayısı, öbek sayısı, ağırlık değerleri bulunur.



Şekil 5.12 Eğitilmiş ağı yüklenmesi

### 5.2.4 Test veri setinin yüklenmesi

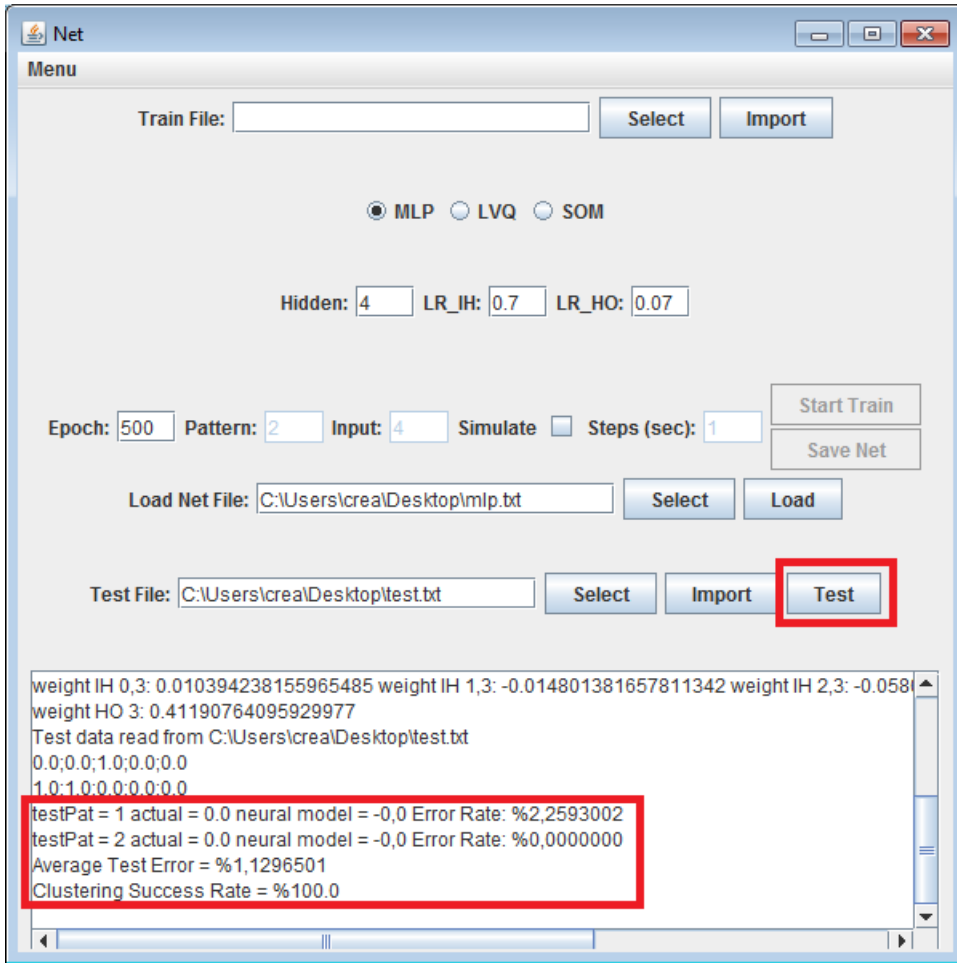
Ağın işleyebileceği formatta hazırlanmış olan test verilerinin bulunduğu dosya seçilir ve yüklenir. Şekil 5.13'te test veri setinin yüklenmesi gösterilmektedir.



Şekil 5.13 Test veri setinin yüklenmesi

### 5.2.5 Ađın test edilmesi

Eđitimi tamamlandı ve ađırlık deđerleri son halini almıř ađa test verileri tek tek gsterilerek sonuları ıktı olarak gsterilir. Beklenen ıktı ile ađın verdiđi ıktı karřılařtırılarak ađın tahmin bařarısı hesaplanarak performansı llr. Ađın rettiđi sonuların gvenilirliđinin yksek olması iin mmkn olduđu kadar fazla veri ile eđitilmesi gerekmektedir. Őekil 5.14'te ađın test edilmesi gsterilmektedir.



Őekil 5.14 Ađın test edilmesi

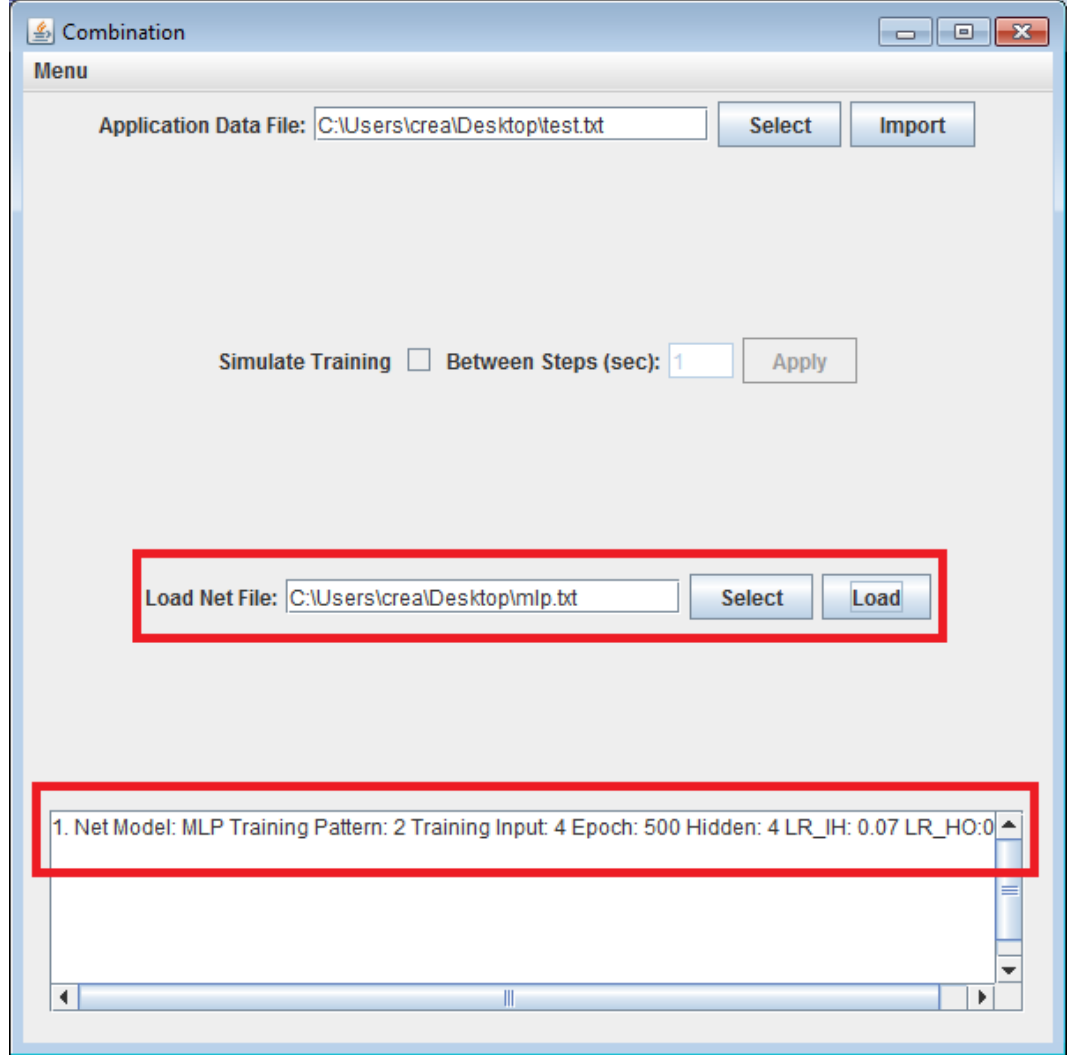
### 5.3 Birleřtirme Arayz

Uygulamanın bu kısmında, tasarlanan ađlar kullanılarak bileřik bir yapay sinir ađı sisteminin oluřturulması ve bu sistemin alıřtırılması gerekleřtirilir. alıřtırma srecinde her girdi iin ađların ve bileřik sistemin ıktı deđerlerinin

adım adım izlenebilmesine imkan sağlayan opsiyonel bir simülasyon ekranı mevcuttur.

### 5.3.1 Ağların yüklenmesi

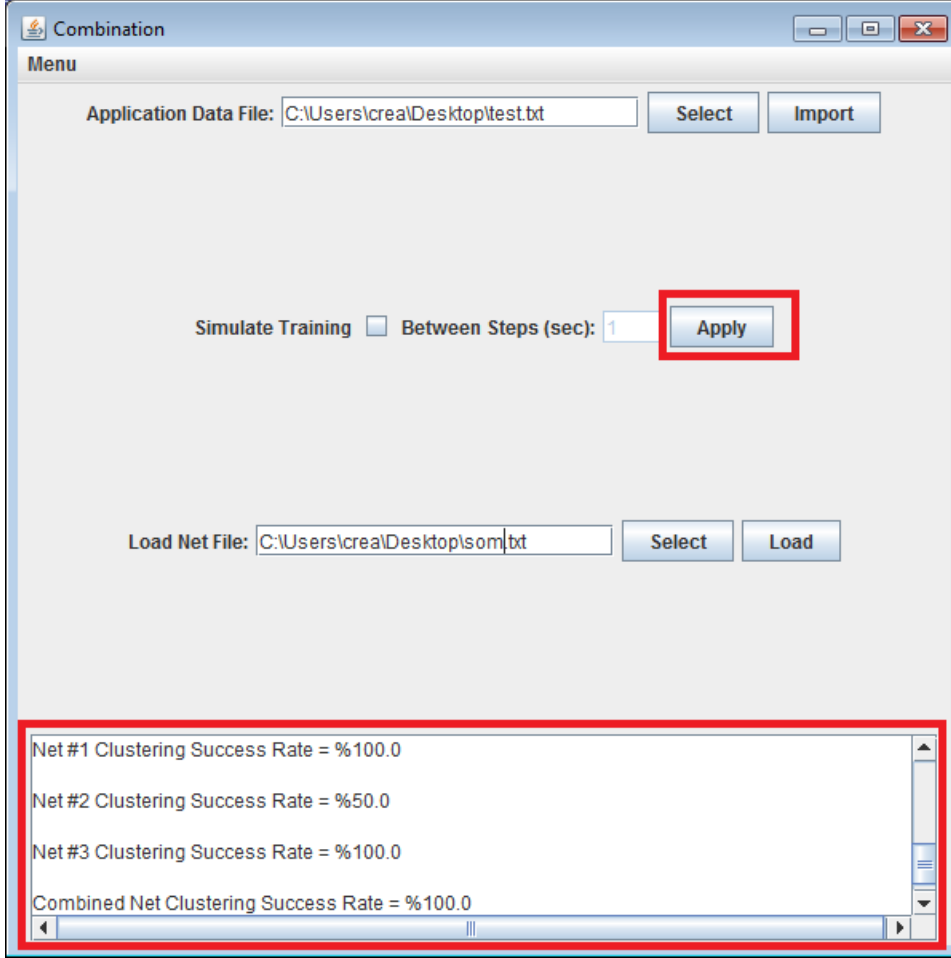
Bileşik sistemde yer alması istenen önceden eğitilmiş ağlar sırayla yüklenir. Şekil 5.15'te ağların bileşik ağ sistemine yüklenmesi gösterilmektedir.



Şekil 5.15 Ağların bileşik ağ sistemine yüklenmesi

### 5.3.2 Ağların birleştirilmesi

Bileşik sistemde yer alacak ağların ve sisteme sunulacak veri setinin yüklenmesi tamamlandıktan sonra sistem, ağların çıktılarından ortak bir karar üretmek üzere çalıştırılır. Şekil 5.16'da bileşik ağ sisteminin çalıştırılması gösterilmektedir.



Şekil 5.16 Bileşik ağ sisteminin çalıştırılması

Bu işlem sonucunda girdi veri setindeki her bir desen için ağların ayrı ayrı ürettiği çıktılar ve bileşik ağ sisteminin ürettiği çıktı aşağıda gösterilmiştir:

```

Pat #1 Actual = 0.0
Net1 (MLP)= pat = 1 actual = 0.0 cluster no:0.0
Net2 (LVQ)= pat = 1 actual = 0.0 cluster no:1.0
Net3 (SOM)= pat = 1 actual = 0.0 cluster no:0.0
Winner: 0.0

```

Pat #2 Actual = 0.0

Net1 (MLP)= pat = 2 actual = 0.0 cluster no:0.0

Net2 (LVQ)= pat = 2 actual = 0.0 cluster no:0.0

Net3 (SOM)= pat = 2 actual = 0.0 cluster no:0.0

Winner: 0.0

Net #1 Clustering Success Rate = %100.0

Net #2 Clustering Success Rate = %50.0

Net #3 Clustering Success Rate = %100.0

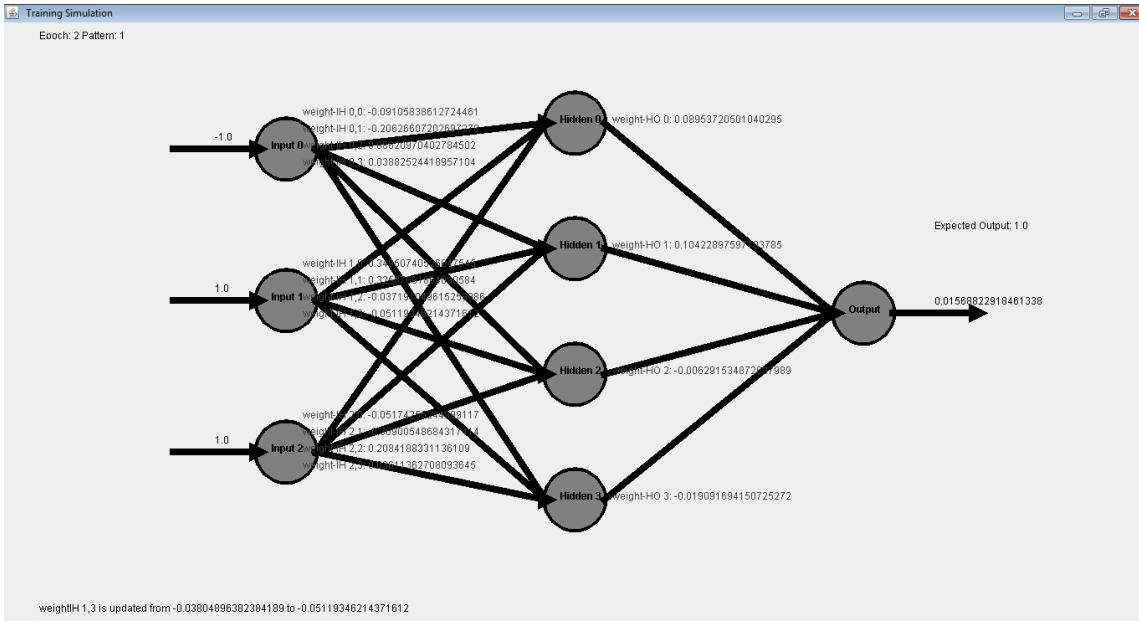
Combined Net Clustering Success Rate = %100.0

## **5.4 Görsel Simülasyon**

### **5.4.1 Ağ tasarım simülasyonu**

Ağ tasarımı arayüzünde, ağın eğitilmesi başlatılmadan önce simülasyon seçeneği aktif edilmiş ise eğitim sürecinde gerçekleşen her olay, kullanıcının belirlediği bekleme süreleri adım adım izlenebilir. Her bir tekrar süresince ağa gösterilen her bir desen için ağırlıklarda yapılan her güncellenme bu sayede görülebilmektedir.

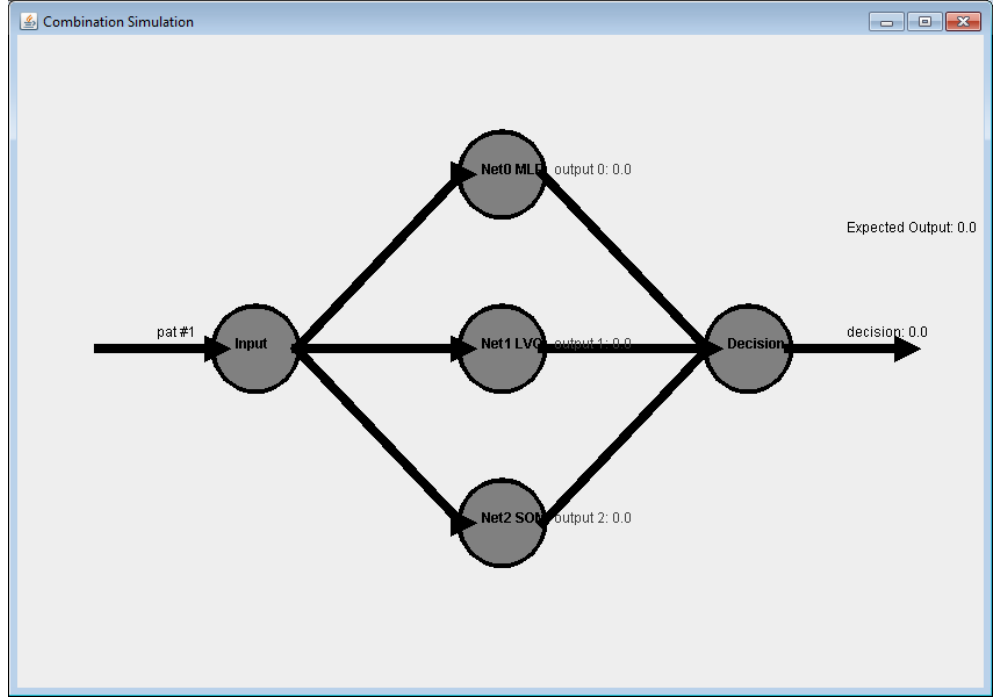
Görsel simülasyon ekranı, test aşamasında her bir test verisi için çıktı oluşumunun gösterilmesi için de kullanılabilir. Şekil 5.17'de görsel simülasyon arayüzü görülmektedir.



Şekil 5.17 Görsel simülasyon

### 5.4.2 Ağ birleştirme simülasyonu

Ağ birleştirme ara yüzünde, her bir test girdisi için her bir ağın üreteceği çıktı değeri, ve bu çıktılara bağlı olarak bileşik ağı vereceği çıktı değerinin görsel olarak gösterimi de simülasyon seçeneği aktif edildiği takdirde mümkündür. Şekil 5.18’de ağ birleştirme simülasyonu arayüzü görülmektedir..



Şekil 5.18 Ağ birleştirme simülasyonu

## **6. TELEKOMÜNİKASYONDA YAPAY SİNİR AĞLARI İLE SAHTEKÂRLIK TESPİTİ**

### **6.1 Araştırmanın Amacı, Kapsamı ve Yöntemi**

#### **6.1.1 Araştırmanın amacı**

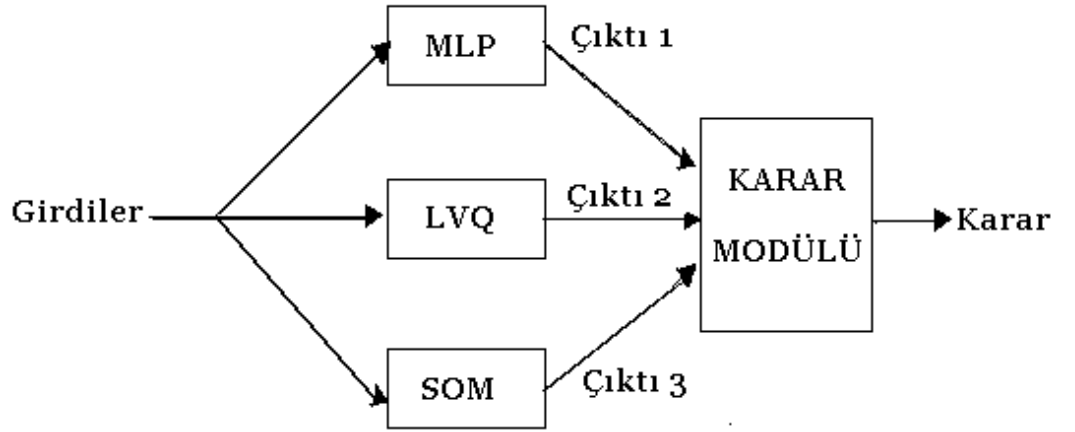
Bu çalışmanın amacı, geliştirilen bir bileşik yapay sinir ağı modeli ile telekomünikasyon şirketlerinde zarara yol açan sahtekârlık amaçlı kullanımların arama detay kayıtları kullanılarak tespit edilmesidir. Yapay sinir ağı modelinin tespit başarısının hesaplanmasında referans olarak kullanılan sahtekârlık ve normal kullanım etiketleri, örnek veri setinin alındığı telekomünikasyon şirketi bünyesinde kural tabanlı bir sahtekârlık tespit sisteminin yönlendirmesi ve sahtekârlık yönetimi çalışanlarının incelemesi sonucunda atanmıştır.

#### **6.1.2 Araştırmanın kapsamı**

Projede bir telekomünikasyon şirketinin iki yıllık kullanım verileri esas alınmıştır. Bu veriler, daha önce karşılaşılmış sahtekârlık olaylarında görülen ortak davranışlar göz önünde bulundurulmak suretiyle işlenerek sahtekârlık tespitinde kullanılacak bilgiler haline getirilmiştir. Yapay sinir ağı modellerinin geliştirme aşamasında belirlenmesi gereken ağ yapısı, başlangıç ağırlık değerlerinin atanması, öğrenme algoritması, toplama ve aktivasyon fonksiyonları, öğrenme oranı, katman sayıları, her katmanda kullanılacak işlem elemanı sayısı gibi parametrelerin belirlenmesi için belli bir yöntem olmadığından dolayı uzman görüşleri göz önüne alınarak çok sayıda deneme yapılması gerekmiştir. Yapılan denemelerde farklı ağ yapılarının oluşturulmasında esneklik sağlanması amacıyla bir paket program kullanmak yerine çalışmanın amacına uygun yeni bir program geliştirilmiştir.

### 6.1.3 Araştırmanın yöntemi

Telekomünikasyonda sahtekârlık probleminin çözümü için, üç farklı yapay sinir ağı modelinden oluşan bir ortak karar mekanizması geliştirilmiştir. Bu bileşik modelde Çok Katmanlı Algılayıcı (MLP: Multilayer Perceptron), Öğrenen Vektör Nicelendirmesi (LVQ: Learning Vector Quantization), Özörgütlemeli Harita (SOM: Self Organizing Map) ağ modelleri birlikte çalışmaktadır. Bu üç ağ modeli aynı eğitim verileriyle eğitilir ve çıktı olarak verdikleri sonuçlar arasında oyçokluğuyla karara varılır. Şekil 6.1'de bileşik yapay sinir ağı karar mekanizması görülmektedir.



Şekil 6.1 Bileşik yapay sinir ağı karar mekanizması

## 6.2 Veri Setinin Hazırlanması

### 6.2.1 Örneklerin toplanması

Örnekler, telekomünikasyon hizmeti veren bir şirketin karşılaştığı sahtekârlık olaylarına ilişkin verilerden oluşmaktadır. Girdi olarak telefon hizmeti kullanıcılarının konuşma detay kayıtları (CDR), kişisel bilgileri ve ödeme bilgileri kullanılmıştır. Buna karşılık olarak sistemin, verilen girdilerle temsil edilen olayın sahtekârlık olup olmadığına karar vermesi beklenmektedir. Oluşturulan girdi

setlerinin bir kısmı eğitim, bir kısmı test verileri olarak kullanılacaktır. Ağın daha önce karşılaşmadığı senaryolara ilişkin doğru yorum yapabilmesi için, ağın eğitiminde kullanılan örneklerin, öğretilecek sahtekârlık problemi uzayının tümünü temsil edebilecek şekilde seçilmesi hedeflenmiştir.

### 6.2.2 Veri setinin oluşturulması

Telekomünikasyon şirketinden alınan CDR'lar, üyelik ve ödeme bilgilerinden gerekli olan sahalara ayıklanarak oluşturulan veri seti, tasarlanan tüm ağların girdilerini oluşturmaktadır. Veri setinin oluşturulmasında her bir kullanıcıyı temsil eden bir kayıt tasarlanmış, kaydın sahalarda da bu kullanıcıya ait üyelik, arama ve ödeme davranışlarına ilişkin istatistikler tutularak sahtekâr veya normal kullanıcı olduğunu belirten etiketleme yapılmıştır.

Hazırlanan veri setinde 22 girdi ve 1 çıktı sahası bulunmaktadır. Bu sahalara ve açıklamaları Çizelge 6.1'de görülmektedir.

Çizelge 6.1 Veri setinde bulunan sahalara ilişkin açıklamalar

Saha No	Saha Adı	Açıklama
#1	country_ID	Kullanıcının bulunduğu ülke kodu
#2	Accounts From Same IP	Kullanıcı ile aynı IP'den üye olan kullanıcı sayısı
#3	Accounts With Same PassWord	Kullanıcı ile aynı parolayı kullanan kullanıcı sayısı
#4	Member Since (days)	Kullanıcının üyelik tarihinden itibaren geçen süre
#5	Usage Range	Kullanıcının ilk ve son kullanımı arasındaki süre
#6	Success Rate	Kullanıcının gerçekleştirdiği aramaların başarı oranı
#7	Successful Calls	Kullanıcının gerçekleştirdiği başarılı aramaların sayısı
#8	Total Calls	Kullanıcının gerçekleştirdiği tüm aramaların sayısı
#9	Total Charge	Kullanıcının gerçekleştirdiği aramaların toplam ücreti
#10	Total Duration	Kullanıcının gerçekleştirdiği aramaların toplam süresi
#11	CalledPartyNumbers	Kullanıcının aradığı farklı numaraların sayısı
#12	Landline Calls	Kullanıcının sabit hat aramalarının sayısı
#13	LandLine Charge	Kullanıcının sabit hat aramalarının toplam ücreti
#14	LandLine Duration	Kullanıcının sabit hat aramalarının toplam süresi
#15	Mobile Calls	Kullanıcının mobil hat aramalarının sayısı
#16	Mobile Charge	Kullanıcının mobil hat aramalarının toplam ücreti
#17	Mobile Duration	Kullanıcının sabit hat aramalarının toplam süresi
#18	Onnet Calls	Kullanıcının internet aramalarının toplam ücreti
#19	Onnet Duration	Kullanıcının internet aramalarının toplam süresi
#20	Number of Payments	Kullanıcının yaptığı ödeme sayısı
#21	dur per day	Kullanıcının günlük ortalama arama süresi
#22	charge per min	Kullanıcının aramalarının dakika başına ortalama ücreti
#23	Fraud	Kullanıcının sahtekâr olup olmadığı bilgisi

### 6.2.3 Girdi ve çıktılarının sayısal gösterimi

Ağın girdileri işleyebilmesi için değerlerin tümünün sayısal olması gerektiğinden dolayı sayısal olmayan değerlerin sayısal değerlerle temsil edilmesi sağlanmıştır. Örneğin kullanıcının ülke bilgisinin kullanılabilmesi için her ülkeye birer sayısal kimlik atanmış ve veri setinde bu kimlikler ile gösterilmiştir. Kullanıcıların IP ve Password'lerine ilişkin olarak, sahtekârlık tespitinde göz önünde bulundurulmuş bir ölçüt olan aynı IP adresinden veya aynı parola ile üye olan kullanıcı sayısı bilgileri kullanılmıştır.

Ağlar, üretecekleri çıktı değerleri 0 ve 1 olacak şekilde tasarlanmıştır. Ağdaki hata oranlarının hesaplanabilmesi için, ağın üretmesi beklenen değeri temsil eden etiket sahası (fraud), sahtekârlık durumu için 1, normal kullanım için 0 değerleri ile temsil edilmiştir.

Verilerin sayısal formata çevrilmiş görünümü Çizelge 6.2'de gösterilmiştir.

Çizelge 6.2 Sayısal veri setinden 20 kayıt örneği

#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22	#23	
87	3	65	574	16	50.66	38	75	9988	3646	32	6	1148	884	32	8840	2762	0	0	1	21447.05	0.27	1	
182	2	14	667	0	81.81	36	44	9996	5184	15	36	9996	5184	0	0	0	0	0	1	518400.00	0.19	1	
182	1	14	667	0	96.00	24	25	9996	5483	13	24	9996	5483	0	0	0	0	0	1	548300.00	0.18	1	
182	1	68	655	1	36.36	4	11	3870	2447	5	0	0	0	4	3870	2447	0	0	1	122350.00	0.15	1	
0	3	1	363	97	25.32	59	233	0	16735	51	10	0	74	49	0	16661	0	0	0	17076.53	0.00	1	
182	1	68	667	17	62.22	28	45	8504	11949	23	15	4568	9250	13	3936	2699	0	0	1	66383.33	0.07	1	
182	3	68	668	9	73.91	17	23	10000	9690	9	2	1890	1210	15	8110	8480	0	0	1	96900.00	0.10	1	
182	1	68	662	14	20.00	5	25	3670	3648	14	0	0	0	5	3670	3648	0	0	1	24320.00	0.10	1	
182	1	3	717	44	46.32	126	272	9964	43492	89	47	528	21098	79	9436	22394	0	0	1	96648.88	0.02	1	
182	1	68	684	6	27.77	5	18	10000	6625	5	1	10	20	4	9990	6605	0	0	1	94642.85	0.15	1	
82	2	2	83	1	75.00	9	12	9984	7357	1	0	0	0	9	9984	7357	0	0	1	367850.00	0.13	0	
90	1	1	171	119	20.00	20	100	9960	8067	28	12	120	925	8	9840	7142	0	0	1	6722.50	0.12	0	
90	1	1	173	106	33.33	42	126	29960	31687	11	35	24650	28340	7	5310	3347	0	0	3	29614.01	0.09	0	
90	1	1	289	69	46.31	44	95	9996	28198	17	24	4061	22838	20	5935	5360	0	0	1	40282.85	0.03	0	
26	1	1	422	128	22.85	8	35	80	10408	8	8	80	10408	0	0	0	0	0	1	8068.21	0.00	0	
10	1	2	540	137	56.14	32	57	9977	8811	15	21	5729	6734	11	4248	2077	0	0	1	6384.78	0.11	0	
10	1	1	298	114	37.41	52	139	3161	15131	18	39	2053	14729	13	1108	402	0	0	1	13157.39	0.02	0	
10	1	1	376	0	67.64	23	34	230	750	11	23	230	750	0	0	0	0	0	1	75000.00	0.03	0	
10	1	180	53	35	48.71	38	78	9660	8713	15	0	0	0	38	9660	8713	0	0	1	24202.77	0.11	0	
0	1	11	615	7	8.33	1	12	0	25	6	0	0	0	0	0	0	0	1	25	0	312.50	0.00	0

### 6.2.4 Girdi ve çıktıların ölçeklendirilmesi

Veri setindeki değerlerin birbirinden farklı ölçeklere sahip olmasının ağı yanlış yönlendirmesinin engellenmesi için girdi ve çıktıların tek ölçekte incelenmesi amacıyla belli bir aralıkta ölçeklendirilmiştir. Aynı anda MLP, LVQ ve SOM ağ modellerinin yapısına uyum sağlaması açısından bu ölçeklendirme 0-1 arasında yapılmıştır.

MLP ağında aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanıldığı için yalnızca 0-1 arasında çıktılar üretilir. LVQ ve SOM ağları, örnekleri iki sınıfa ayıracak şekilde tasarladıkları için çıktıları ya 0 ya 1 olmaktadır. Bu ifade şekli sistemin anlaşılabilirliğini kolaylaştırmaktadır. MLP ağının çıktıları, diğer iki ağın çıktıları ile karşılaştırma yapılabilmesi için 0 veya 1 değerlerinden birine yuvarlanmaktadır.

Verilerin ölçeklendikten sonraki görünümü Çizelge 6.3'te gösterilmiştir.

Çizelge 6.3 Ölçeklendirilmiş veri seti örneği

#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22	#23
0.47	1.0	0.35	0.78	0.11	0.48	0.29	0.24	0.33	0.08	0.35	0.12	0.04	0.03	0.40	0.88	0.12	0.0	0.0	0.33	0.03	1.0	1.0
1.0	0.5	0.07	0.92	0.0	0.83	0.28	0.12	0.33	0.11	0.15	0.76	0.40	0.18	0.0	0.0	0.0	0.0	0.0	0.33	0.94	0.70	1.0
1.0	0.0	0.07	0.92	0.0	1.0	0.18	0.05	0.33	0.12	0.13	0.51	0.40	0.19	0.0	0.0	0.0	0.0	0.0	0.33	1.0	0.66	1.0
1.0	0.0	0.37	0.90	0.01	0.31	0.02	0.0	0.12	0.05	0.04	0.0	0.0	0.0	0.05	0.38	0.10	0.0	0.0	0.33	0.22	0.57	1.0
0.0	1.0	0.0	0.46	0.70	0.19	0.46	0.85	0.0	0.38	0.56	0.21	0.0	0.01	0.62	0.0	0.74	0.0	0.0	0.0	0.03	0.0	1.0
1.0	0.0	0.37	0.92	0.12	0.61	0.21	0.13	0.28	0.27	0.25	0.31	0.18	0.32	0.16	0.39	0.12	0.0	0.0	0.33	0.12	0.25	1.0
1.0	1.0	0.37	0.92	0.06	0.74	0.12	0.04	0.33	0.22	0.09	0.04	0.07	0.04	0.18	0.81	0.37	0.0	0.0	0.33	0.17	0.37	1.0
1.0	0.0	0.37	0.91	0.10	0.13	0.03	0.05	0.12	0.08	0.14	0.0	0.0	0.0	0.06	0.36	0.16	0.0	0.0	0.33	0.04	0.36	1.0
1.0	0.0	0.01	1.0	0.32	0.43	1.0	1.0	0.33	1.0	1.0	1.0	0.02	0.74	1.0	0.94	1.0	0.0	0.0	0.33	0.17	0.08	1.0
1.0	0.0	0.37	0.95	0.04	0.22	0.03	0.02	0.33	0.15	0.04	0.02	4.05	7.05	0.05	1.0	0.29	0.0	0.0	0.33	0.17	0.55	1.0
0.45	0.5	0.01	0.04	0.01	0.76	0.06	0.01	0.33	0.16	0.0	0.0	0.0	0.0	0.11	0.99	0.32	0.0	0.0	0.33	0.67	0.49	0.0
0.49	0.0	0.0	0.17	0.86	0.13	0.15	0.34	0.33	0.18	0.30	0.25	0.01	0.03	0.10	0.98	0.31	0.0	0.0	0.33	0.01	0.45	0.0
0.49	0.0	0.0	0.18	0.77	0.28	0.32	0.44	1.0	0.72	0.11	0.74	1.0	1.0	0.08	0.53	0.14	0.0	0.0	1.0	0.05	0.34	0.0
0.49	0.0	0.0	0.35	0.50	0.43	0.34	0.32	0.33	0.64	0.18	0.51	0.16	0.80	0.25	0.59	0.23	0.0	0.0	0.33	0.07	0.12	0.0
0.14	0.0	0.0	0.55	0.93	0.16	0.05	0.09	0.01	0.23	0.07	0.17	0.00	0.36	0.0	0.0	0.0	0.0	0.0	0.33	0.01	0.01	0.0
0.05	0.0	0.01	0.73	1.0	0.54	0.24	0.17	0.33	0.20	0.15	0.44	0.23	0.23	0.13	0.42	0.09	0.0	0.0	0.33	0.01	0.41	0.0
0.05	0.0	0.0	0.36	0.83	0.33	0.40	0.49	0.10	0.34	0.19	0.82	0.08	0.51	0.16	0.11	0.01	0.0	0.0	0.33	0.02	0.07	0.0
0.05	0.0	0.0	0.48	0.0	0.67	0.17	0.08	0.01	0.01	0.11	0.48	0.01	0.02	0.0	0.0	0.0	0.0	0.0	0.33	0.13	0.11	0.0
0.05	0.0	1.0	0.0	0.25	0.46	0.29	0.25	0.32	0.19	0.15	0.0	0.0	0.0	0.48	0.96	0.38	0.0	0.0	0.33	0.04	0.40	0.0
0.0	0.0	0.05	0.84	0.05	0.0	0.0	0.01	0.0	0.0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0

### **6.2.5 Eğitim ve test setlerinin belirlenmesi**

Yapay sinir ağlarını test etmek için eğitim işleminde kullanılmayan veri kümeleri kullanılmıştır. Verilerin %50'si eğitim % 50'si ise test için ayrılmıştır.

### **6.3 Ağ Tasarımlarının Yapılması (Deney 1)**

Sisteme sunulacak ilk deney (Deney 1) için yapılan ağ tasarımları aşağıda açıklanmıştır.

#### **6.3.1 Ağın topolojisinin belirlenmesi**

Tüm ağların girdi katmanında 22, çıktı katmanında bir nöron bulunmaktadır. MLP ağının topolojisinde bir ara katman ve bu katmanda beş nöron vardır. En iyi performansı sağlayan kombinasyonu bulmak için farklı sayılarda ara katman ve işlem elemanları ile ve denemeler yapılmış ve hata yüzdeleri gözlemlemek en uygun ağ topolojisinin belirlenmesi hedeflenmiştir.

#### **6.3.2 Katsayıların belirlenmesi**

##### **6.3.2.1 MLP ağının katsayıların belirlenmesi**

MLP ağında hem girdi katmanı ile gizli katman arasındaki bağlantılar hem de gizli katman ile çıktı katmanı arasındaki bağlantılar için ayrı ayrı öğrenme katsayıları kullanılmıştır. 0.01 ile 1 arasında yapılan denemeler sonucunda en yüksek sınıflandırma performansı ve en düşük RMS yüzdesini veren ağırlıkların girdi katmanı ile gizli katman arası bağlantılar için 0.7, gizli katman ile çıktı katmanı arasındaki bağlantılar için 0.07 değerlerinin verdiği görülmüştür.

##### **6.3.2.2 LVQ ağının katsayıların belirlenmesi**

LVQ ağında eğitim boyunca öğrenme katsayısı her adım sonunda %20 azaltılmaktadır. Öğrenme katsayısının başlangıç değerinin bulunması için yapılan denemelerde, ağın yaptığı kümeleme işleminde girdi vektörlerinin beklenen çıktının bulunduğu kümeyi temsil eden referans vektörüne olan uzaklığının düşük, diğer kümeyi temsil eden referans vektöre olan uzaklığının yüksek

olmasını sađlayan katsayıların bulunması hedeflenmiştir. Yapılan denemelerde, 0.4 deęerinin bu kořulun sađladığı tespit edilmiştir.

### **6.3.2.3 SOM ađının katsayıların belirlenmesi**

SOM ađının eđitimine başlamadan önce maksimum ve minimum öğrenme katsayılarının kullanıcı tarafından belirlenmesi gerekmektedir. Bu deęerler, düzenli olarak azaltılacak olan öğrenme katsayısının başlangıç deęerini ve hangi deęere kadar küçültüleceğini ifade eder. Öğrenme katsayısının eğitim sonunda mümkün olduğu kadar 0'a yaklařmış olması hedeflendiğinden bu deęer 0 olarak belirlenmiştir. Maksimum öğrenme katsayısı için ise 0.01 ile 1 arasında denemeler yapılmış ve en iyi performansı veren deęerin 0.8 olduğu tespit edilmiştir.

## **6.4 Ađların Eđitimi ve Test Edilmesi**

Ađlıkların başlangıç deęerleri, 0 ile 1 arasındaki rastgele deęerler olarak atanmıştır.

Ađın önceden belirlenen adım sayısına ulaşması durumunda eğitim sona erer. Maksimum adım sayısı MLP ađında 30000, diđer ađlarda 500 olarak belirlenmiştir.

Ađ yapılarının eđitilmesi sırasında, 29 normal kullanım 21 sahtekârlık durumunu içeren 50 örnek olay kullanılmıştır.

Ađ yapılarının test edilmesi sırasında, eğitim işlemi sırasında kullanılmayan 21 normal kullanım 29 sahtekârlık durumunu içeren 50 örnek olay kullanılmıştır.

### **6.4.1 MLP ađının eđitimi ve test edilmesi**

Örnekler ađa rastgele seçilerek sunulmuştur. Bir örneğin her adımda ađa gösterilme garantisi yoktur. Her adımda ađa gösterilen her bir örnekten sonra ađın ürettiği çıktı ile beklenen çıktı hata bulunarak ađlıklara dağıtılır. Hatanın hesaplanması, RMS (Kare Ortalamalarının Karekökü) yöntemi ile yapılır. Her adım sonunda o adım için toplam hata oranı hesaplanır. Bu hata oranının gittikçe

azalması öğrenmenin gerçekleştiğini gösterir. Eğitim sonunda kabul edilir bir değere ulaşmış olması gerekir. Toplama fonksiyonu olarak kümülatif toplam, aktivasyon fonksiyonu olarak hiperbolik tanjant fonksiyonu kullanılmıştır. MLP ağlarının eğitiminde delta öğrenme kuralı kullanılmaktadır.

Geliştirilen MLP ağının, adım numarası arttıkça elde edilen hata oranları Çizelge 6.4'te gösterilmiştir.

Çizelge 6.4 Her adımda azalan hata oranları (Deney 1)

Adım Numarası	RMS
0	0,248505
2	0,185899
5	0,151068
10	0,139488
100	0,058959
250	0,036544
500	0,020221
1000	0,010358
2500	0,005391
5000	0,002957
10000	0,001934
20000	0,001248
30000	0,000841

Ağın eğitim seti ile eğitimi sonucunda sınıflandırma başarısı %100 olarak elde edilmiştir

30000 adım sonunda elde edilen ortalama eğitim hatası %0,053212 olarak hesaplanmıştır. Eğitimin sonlandığı anda ağın her bir eğitim deseni için ürettiği çıktının hata oranları Çizelge 6.5'te görülmektedir. Örneğin hata oranının %0,1 olması, gerçek değeri 1 olan çıktının 0,9 veya 1,1 olarak tahmin edildiğini ifade eder. Sınıflandırma yapılabilmesi için MLP ağının tüm çıktılarının 0 veya 1 ile temsil edilmesi gerekmektedir. Bu nedenle ağın çıktıları bileşik ağ mekanizmasında girdi olarak kullanılırken bu değerler yuvarlanmıştır.

Çizelge 6.5 Eğitim sonu hata yüzdeleri (Deney 1)

<b>Desen No</b>	<b>Hata Oranı</b>	<b>Desen No</b>	<b>Hata Oranı</b>
1	%0,021070	26	%0,029229
2	%0,017097	27	%0,009100
3	%0,027504	28	%0,005278
4	%0,123851	29	%0,007133
5	%0,021869	30	%0,045752
6	%0,000134	31	%0,038750
7	%0,045743	32	%0,041652
8	%0,020571	33	%0,237202
9	%0,016837	34	%0,100018
10	%0,019852	35	%0,001807
11	%0,015737	36	%0,025337
12	%0,023939	37	%0,141661
13	%0,085145	38	%0,001051
14	%0,109117	39	%0,172923
15	%0,004256	40	%0,267276
16	%0,024793	41	%0,038069
17	%0,025265	42	%0,010911
18	%0,027378	43	%0,208122
19	%0,230543	44	%0,017497
20	%0,008365	45	%0,045769
21	%0,016602	46	%0,052226
22	%0,063479	47	%0,033293
23	%0,003551	48	%0,040873
24	%0,013569	49	%0,028564
25	%0,029445	50	%0,065390

Ağın kendisine daha önce gösterilmemiş olan test seti ile test edilmesi sonucunda elde edilen sınıflandırma başarısı %96'dır.

Çizelge 6.6 MLP ağının test sonuçları (Deney 1)

Sınıf	Test Örneği Sayısı	Doğru	Yanlış
Sahtekâr	29	29	0
Normal	21	20	1
Toplam	50	49	1

### 6.4.2 LVQ Ağının eğitimi ve test edilmesi

Örnekler ağa sırayla sunulmuştur. Her adımda her bir örnek ağa bir kere gösterilmiştir. Her adımda ağa gösterilen her bir örnekten sonra ağın iki kümeye olan euclid uzaklıkları hesaplanır ve kazanan eleman belirlenir. Ağ bu belirlemenin doğru olup olmadığı bilgisini verir. Doğru ise bağlantıların güçlenmesi, yanlış ise zayıflaması sağlanacak şekilde ağırlıklar güncellenir. Adım sonunda öğrenme katsayısı %20 azaltılır. LVQ ağlarının eğitiminde kohonen öğrenme kuralı kullanılmaktadır.

Ağın eğitim seti ile eğitimi sonucunda sınıflandırma başarısı %98 olarak elde edilmiştir.

Ağa daha önce gösterilmemiş olan test seti ile test edilmesi sonucunda elde edilen sınıflandırma başarısı %98'dir.

Çizelge 6.7 LVQ ağının test sonuçları (Deney 1)

Sınıf	Test Örneği Sayısı	Doğru	Yanlış
Sahtekâr	29	28	1
Normal	21	21	0
Toplam	50	49	1

### 6.4.3 SOM ağının eğitimi ve test edilmesi

Örnekler ağa sırayla sunulmuş ve her adımda her bir örnek ağa bir kere gösterilmiştir. LVQ ağında olduğu gibi her adımda ağa gösterilen her bir örnekten

sonra ađın iki kümeye olan euclid uzaklıkları hesaplanır ve kazanan eleman belirlenir.

Adım sonunda öğrenme katsayısı, tüm adımlar tamamlandığında 0'a ulaşılmasını sağlayacak şekilde belirlenen oranla azaltılır. Daha sonra azaltılan öğrenme katsayısı kullanılarak kazanan elemanın ağırlıkları kuvvetlendirilir.

Ađın eğitim seti ile eğitimi sonucunda sınıflandırma başarısı %96 olarak elde edilmiştir.

Ađa daha önce gösterilmemiş olan test seti ile test edilmesi sonucunda elde edilen sınıflandırma başarısı %96'dır.

Çizelge 6.8 SOM ađının test sonuçları (Deney 1)

Sınıf	Test Örneđi Sayısı	Dođru	Yanlış
Sahtekâr	29	29	0
Normal	21	19	2
Toplam	50	48	2

## 6.5 Ađların Birleřtirilmesi

Eđitilen üç ađın aynı test girdileri üzerinde ortak çalışmak üzere birleřtirildiđinde, verilen kararların %100'ünün dođru olduđu görülmüřtür.

Çizelge 6.9 Bileřik ađın test sonuçları (Deney 1)

Sınıf	Test Örneđi Sayısı	Dođru	Yanlış
Sahtekâr	29	29	0
Normal	21	21	0
Toplam	50	50	0

Çizelge 6.10'da ađa sunulan test desenleri için her bir ađın oluřturduđu çıktıları ve bileřik ađın verdiđi karar gösterilmiştir. 16. desende LVQ ađının sahtekâr bir kullanıcı için verdiđi normal kararı, MLP ve SOM ađları tarafından

düzeltilmiştir. 49. desende MLP ağı normal bir kullanıcı için sahtekâr tahmininde bulunmuş, bu hata da LVQ ve SOM ağları tarafından düzeltilmiştir. 18. ve 42. desenlerde SOM ağının verdiği yanlış karar diğer ağlar tarafından düzeltilmiştir.

Çizelge 6.10 Bileşik ağı oluşturan bileşenlerin ortak kararı (Deney 1)

Desen No	Beklenen Çıktı	MLP Çıktısı	LVQ Çıktısı	SOM Çıktısı	Ortak Karar
1	0	0	0	0	0
2	1	1	1	1	1
3	1	1	1	1	1
4	0	0	0	0	0
5	1	1	1	1	1
6	1	1	1	1	1
7	0	0	0	0	0
8	1	1	1	1	1
9	1	1	1	1	1
10	1	1	1	1	1
11	0	0	0	0	0
12	1	1	1	1	1
13	1	1	1	1	1
14	0	0	0	0	0
15	1	1	1	1	1
16	1	1	0	1	1
17	1	1	1	1	1
18	0	0	0	1	0
19	0	0	0	0	0
20	1	1	1	1	1
21	1	1	1	1	1
22	0	0	0	0	0
23	0	0	0	0	0
24	0	0	0	0	0
25	1	1	1	1	1
26	1	1	1	1	1
27	1	1	1	1	1
28	0	0	0	0	0
29	0	0	0	0	0
30	1	1	1	1	1
31	1	1	1	1	1

32	0	0	0	0	0
33	0	0	0	0	0
34	1	1	1	1	1
35	1	1	1	1	1
36	1	1	1	1	1
37	1	1	1	1	1
38	0	0	0	0	0
39	1	1	1	1	1
40	0	0	0	0	0
41	0	0	0	0	0
42	0	0	0	1	0
43	0	0	0	0	0
44	1	1	1	1	1
45	1	1	1	1	1
46	0	0	0	0	0
47	1	1	1	1	1
48	1	1	1	1	1
49	0	1	0	0	0
50	1	1	1	1	1

Sonuç olarak ağlar ayrı ayrı kullanıldıklarında performansları MLP ağı için %98.0, LVQ için %98.0 ve SOM için %96.0 iken bileşik ağda bu oranın %100'e çıktığı görülmektedir.

## 6.6 Ağların Farklı Veri Setleri için Uyarlanması (Deney 2)

Tüm problem uzayını temsil edecek şekilde hazırlanan yeni veri seti, 63 sahtekârlık, 265 normal kullanım durumunu içeren 328 adet örnekten oluşmaktadır. Bu veri seti rastgele sıralanarak ikiye bölünmüş ve 164 eğitim, 164 test örneği elde edilmiştir. Eğitim setinde 138 normal kullanım 26 sahtekârlık durumunu, test seti ise 127 normal kullanım 37 sahtekârlık durumunu içermektedir.

### 6.6.1 MLP ağının eğitimi ve test edilmesi

MLP ağının eğitiminde maksimum adım sayısı 1000, gizli katman sayısı 1, gizli katmanda bulunan eleman sayısı 5, girdi katmanı ile gizli katman arası öğrenme katsayısı 0.7, gizli katman ile çıktı katmanı arası öğrenme katsayısı 0.07 olarak belirlenmiştir.

Ağın eğitim seti ile eğitimi sonucunda sınıflandırma başarısı %100 olarak elde edilmiştir. Ağın kendisine daha önce gösterilmemiş olan test seti ile test edilmesi sonucunda elde edilen sınıflandırma başarısı %96.34'tür.

Çizelge 6.11 MLP ağının test sonuçları (Deney 2)

Sınıf	Test Örneği Sayısı	Doğru	Yanlış
Sahtekâr	37	31	6
Normal	127	127	0
Toplam	164	158	6

### 6.6.2 LVQ ağının eğitimi ve test edilmesi

LVQ ağının eğitiminde maksimum adım sayısı 500, başlangıç öğrenme katsayısı ise 0.4 olarak belirlenmiştir.

Ağın eğitim seti ile eğitimi sonucunda sınıflandırma başarısı %98.78 olarak elde edilmiştir.

Ağa daha önce gösterilmemiş olan test seti ile test edilmesi sonucunda elde edilen sınıflandırma başarısı %96.34'tür.

Çizelge 6.12 LVQ ağının test sonuçları (Deney 2)

Sınıf	Test Örneği Sayısı	Doğru	Yanlış
Sahtekâr	37	33	4
Normal	127	125	2
Toplam	164	158	6

### 6.6.3 SOM ağının rğitimi ve test edilmesi

SOM ağının eğitiminde maksimum adım sayısı 500, minimum öğrenme katsayısı 0, maksimum öğrenme katsayısı ise 0.9 olarak belirlenmiştir.

Ağın eğitim seti ile eğitimi sonucunda sınıflandırma başarısı %99.39 olarak elde edilmiştir.

Ağa daha önce gösterilmemiş olan test seti ile test edilmesi sonucunda elde edilen sınıflandırma başarısı %96.95'tir.

Çizelge 6.13 SOM ağının test sonuçları (Deney 2)

Sınıf	Test Örneği Sayısı	Doğru	Yanlış
Sahtekâr	37	32	5
Normal	127	127	0
Toplam	164	159	5

### 6.6.4 Ağların birleştirilmesi

Eğitilen üç ağın aynı test girdileri üzerinde ortak çalışmak üzere birleştirildiğinde, verilen kararların %97.56'sının doğru olduğu görülmüştür.

Çizelge 6.14 Bileşik ağın test sonuçları (Deney 2)

Sınıf	Test Örneği Sayısı	Doğru	Yanlış
Sahtekâr	37	33	4
Normal	127	127	0
Toplam	164	160	4

Çizelge 6.15'te ağa sunulan test desenleri için her bir ağın oluşturduğu çıktılar ve bileşik ağın verdiği karar gösterilmiştir. 32. desende SOM ağının sahtekâr bir kullanıcı için verdiği normal kararı, MLP ve SOM ağları tarafından düzeltilmiştir. 78. ve 160. desenlerde MLP ağı sahtekâr kullanıcılar için normal tahmininde bulunmuş, bu hata da LVQ ve SOM ağları tarafından düzeltilmiştir.

86. ve 117. desenlerde LVQ ağıının normal kullanıcılar için sahtekâr tahmininde bulunma hatası, diğer ağılar tarafından düzeltilmiştir. SOM ağıının verdiği yanlış karar diğer ağılar tarafından düzeltilmiştir. 33, 95, 148 ve 153. desenlerde tüm ağılar sahtekâr bir kullanıcı için normal kararı verdiğinden dolayı bu hatalar düzeltilememiştir.

Çizelge 6.15 Bileşik ağı oluşturan bileşenlerin ortak kararı (Deney 2)

Desen No	Beklenen Çıktı	MLP Çıktısı	LVQ Çıktısı	SOM Çıktısı	Ortak Karar
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	1	1	1	1	1
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	1	1	1	1	1
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0
21	0	0	0	0	0
22	1	1	1	1	1
23	0	0	0	0	0
24	0	0	0	0	0
25	0	0	0	0	0
26	1	1	1	1	1
27	0	0	0	0	0
28	0	0	0	0	0

29	0	0	0	0	0
30	0	0	0	0	0
31	0	0	0	0	0
32	1	1	1	0	1
33	1	0	0	0	0
34	0	0	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	0	0	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	0	0	0	0	0
41	0	0	0	0	0
42	0	0	0	0	0
43	0	0	0	0	0
44	1	1	1	1	1
45	1	1	1	1	1
46	0	0	0	0	0
47	0	0	0	0	0
48	0	0	0	0	0
49	1	1	1	1	1
50	0	0	0	0	0
51	0	0	0	0	0
52	0	0	0	0	0
53	1	1	1	1	1
54	1	1	1	1	1
55	1	1	1	1	1
56	1	1	1	1	1
57	0	0	0	0	0
58	0	0	0	0	0
59	0	0	0	0	0
60	0	0	0	0	0
61	1	1	1	1	1
62	0	0	0	0	0
63	1	1	1	1	1
64	1	1	1	1	1
65	0	0	0	0	0

66	0	0	0	0	0
67	0	0	0	0	0
68	1	1	1	1	1
69	0	0	0	0	0
70	0	0	0	0	0
71	1	1	1	1	1
72	0	0	0	0	0
73	0	0	0	0	0
74	0	0	0	0	0
75	1	1	1	1	1
76	0	0	0	0	0
77	0	0	0	0	0
78	1	0	1	1	1
79	0	0	0	0	0
80	0	0	0	0	0
81	0	0	0	0	0
82	0	0	0	0	0
83	0	0	0	0	0
84	0	0	0	0	0
85	0	0	0	0	0
86	0	0	1	0	0
87	0	0	0	0	0
88	0	0	0	0	0
89	0	0	0	0	0
90	0	0	0	0	0
91	0	0	0	0	0
92	0	0	0	0	0
93	0	0	0	0	0
94	0	0	0	0	0
95	1	0	0	0	0
96	0	0	0	0	0
97	0	0	0	0	0
98	0	0	0	0	0
99	0	0	0	0	0
100	0	0	0	0	0
101	0	0	0	0	0
102	0	0	0	0	0

103	0	0	0	0	0
104	0	0	0	0	0
105	0	0	0	0	0
106	0	0	0	0	0
107	0	0	0	0	0
108	0	0	0	0	0
109	0	0	0	0	0
110	1	1	1	1	1
111	0	0	0	0	0
112	1	1	1	1	1
113	0	0	0	0	0
114	0	0	0	0	0
115	1	1	1	1	1
116	0	0	0	0	0
117	0	0	1	0	0
118	0	0	0	0	0
119	1	1	1	1	1
120	1	1	1	1	1
121	1	1	1	1	1
122	0	0	0	0	0
123	1	1	1	1	1
124	0	0	0	0	0
125	0	0	0	0	0
126	0	0	0	0	0
127	0	0	0	0	0
128	0	0	0	0	0
129	0	0	0	0	0
130	0	0	0	0	0
131	0	0	0	0	0
132	1	1	1	1	1
133	0	0	0	0	0
134	0	0	0	0	0
135	0	0	0	0	0
136	0	0	0	0	0
137	0	0	0	0	0
138	0	0	0	0	0
139	0	0	0	0	0

140	0	0	0	0	0
141	0	0	0	0	0
142	1	1	1	1	1
143	0	0	0	0	0
144	0	0	0	0	0
145	0	0	0	0	0
146	1	1	1	1	1
147	0	0	0	0	0
148	1	0	0	0	0
149	0	0	0	0	0
150	0	0	0	0	0
151	0	0	0	0	0
152	0	0	0	0	0
153	1	0	0	0	0
154	0	0	0	0	0
155	1	1	1	1	1
156	1	1	1	1	1
157	0	0	0	0	0
158	1	1	1	1	1
159	0	0	0	0	0
160	1	0	1	1	1
161	0	0	0	0	0
162	0	0	0	0	0
163	0	0	0	0	0
164	0	0	0	0	0

Sonuç olarak ağlar ayrı ayrı kullanıldıklarında performansları MLP ve LVQ ağları için %96.34 ve SOM için %96.95 iken bileşik ağda bu oranın %97.56'ya çıktığı görülmektedir.

### 6.7 Aynı Eğitim Seti ile Eğitilen Ağların Farklı Test Setleri ile Test Edilmesi (Deney 3)

Tüm problem uzayını temsil edecek şekilde bir adet eğitim seti ve 4 adet test seti hazırlanmıştır. Eğitim seti 9 sahtekârlık 32 normal kullanım, birinci test seti 3 sahtekârlık 38 normal kullanım, ikinci test seti 5 sahtekârlık 36 normal kullanım,

üçüncü test seti 13 sahtekârlık 28 normal kullanım, dördüncü test seti 8 sahtekârlık 33 normal kullanım durumunu içeren 41 adet örnekten oluşmaktadır. Eğitim ve test setlerindeki sahtekârlık ve normal kullanım durumu oranları, örneklerin rastgele seçilmesi sonucunda oluşmuştur.

### 6.7.1 MLP ağının eğitimi ve test edilmesi

MLP ağının eğitiminde maksimum adım sayısı 30000, gizli katman sayısı 1, gizli katmanda bulunan eleman sayısı 10, girdi katmanı ile gizli katman arası öğrenme katsayısı 0.7, gizli katman ile çıktı katmanı arası öğrenme katsayısı 0.07 olarak belirlenmiştir.

Ağın eğitim seti ile eğitimi sonucunda sınıflandırma başarı oranı %100 olarak elde edilmiştir.

Ağın kendisine daha önce gösterilmemiş örneklerden oluşan 4 adet test seti ile test edilmesi sonucunda elde edilen sınıflandırma başarı oranları Çizelge 6.16'da görülmektedir.

Çizelge 6.16 MLP ağının test sonuçları (Deney 3)

Test Seti	MLP Test Başarı Oranı	False Positive	False Negative
1	%95.12	2	0
2	%97.56	1	0
3	%100	0	0
4	%97.56	0	1

MLP ağının test başarı oranlarının ortalaması %97.56, standart sapması ise 1.99 olarak hesaplanmıştır.

### 6.7.2 LVQ ağının eğitimi ve test edilmesi

LVQ ağının eğitiminde maksimum adım sayısı 500, başlangıç öğrenme katsayısı ise 0.4 olarak belirlenmiştir.

Ağın eğitim seti ile eğitimi sonucunda başarı oranı %97.56 olarak elde edilmiştir.

Ağın kendisine daha önce gösterilmemiş örneklerden oluşan 4 adet test seti ile test edilmesi sonucunda elde edilen sınıflandırma başarı oranları Çizelge 6.17'de görülmektedir.

Çizelge 6.17 LVQ ağının test sonuçları (Deney 3)

Test Seti	LVQ Test Başarı Oranı	False Positive	False Negative
1	% 100	0	0
2	% 100	0	0
3	%95.12	0	2
4	%97.56	0	1

LVQ ağının test başarı oranlarının ortalaması %98.17, standart sapması ise 2.33 olarak hesaplanmıştır.

### 6.7.3 SOM ağının eğitimi ve test edilmesi

SOM ağının eğitiminde maksimum adım sayısı 500, minimum öğrenme katsayısı 0, maksimum öğrenme katsayısı ise 0.95 olarak belirlenmiştir.

Ağın eğitim seti ile eğitimi sonucunda başarı oranı %97.56 olarak elde edilmiştir.

Ağın kendisine daha önce gösterilmemiş örneklerden oluşan 4 adet test seti ile test edilmesi sonucunda elde edilen sınıflandırma başarı oranları Çizelge 6.18'de görülmektedir.

Çizelge 6.18 SOM ağının test sonuçları (Deney 3)

Test Seti	SOM Test Başarı Oranı	False Positive	False Negative
1	%100	0	0
2	%100	0	0
3	%97.56	0	1
4	%95.12	1	1

SOM ağının test başarı oranlarının ortalaması %98.17, standart sapması ise 2.33 olarak hesaplanmıştır.

#### 6.7.4 Ağların birleştirilmesi

Eğitilen üç ağ modeli ortak çalışmak üzere birleştirildiğinde verilen kararların başarı oranları Çizelge 6.19’da görülmektedir.

Çizelge 6.19 Bileşik ağın test sonuçları (Deney 3)

Test Seti	Bileşik Ağ Test Başarı Oranı	False Positive	False Negative
1	%100	0	0
2	%100	0	0
3	%97.56	0	1
4	%97.56	0	1

Bileşik ağın test başarı oranlarının ortalaması %98.78, standart sapması ise 1.40 olarak hesaplanmıştır.

Sonuç olarak ağlar ayrı ayrı kullanıldıklarında elde edilen başarı oranlarının standart sapmalar MLP ağı için 1.99, LVQ ve SOM ağları için 2.33 ve iken bileşik ağda 1.40’a düştüğü görülmektedir. Bu durum, bileşik ağın test setindeki değişimlerden daha az etkilendiğini, yani daha kararlı olduğunu ortaya koyar.

## 6.8 Farklı Eğitim Setleri ile Eğitilen Ağların Aynı Test Seti ile Test Edilmesi (Deney 4)

Tüm problem uzayını temsil edecek şekilde 4 adet eğitim seti ve 1 adet test seti hazırlanmıştır. Birinci eğitim seti 8 sahtekârlık 33 normal kullanım, ikinci eğitim seti 12 sahtekârlık 29 normal kullanım, üçüncü eğitim seti 13 sahtekârlık 28 normal kullanım, dördüncü eğitim seti 11 sahtekârlık 30 normal kullanım, test seti 7 sahtekârlık 34 normal kullanım durumunu içeren 41 adet örnekten oluşmaktadır. Eğitim ve test setlerindeki sahtekârlık ve normal kullanım durumu oranları, örneklerin rastgele seçilmesi sonucunda oluşmuştur.

### 6.8.1 MLP ağının eğitimi ve test edilmesi

MLP ağlarının eğitimlerinde maksimum adım sayısı 30000, gizli katman sayısı 1, gizli katmanda bulunan eleman sayısı 10, girdi katmanı ile gizli katman arası öğrenme katsayısı 0.7, gizli katman ile çıktı katmanı arası öğrenme katsayısı 0.07 olarak belirlenmiştir.

4 eğitim seti ile gerçekleştirilen eğitimler sonucunda oluşan ağların eğitim başarı oranları ve ağların kendilerine daha önce gösterilmemiş örneklerden oluşan test seti ile test edilmeleri sonucunda elde edilen sınıflandırma başarı oranları Çizelge 6.20’de görülmektedir.

Çizelge 6.20 MLP ağının test sonuçları (Deney 4)

Eğitim Seti	MLP Eğitim Başarı Oranı	MLP Test Başarı Oranı	False Positive	False Negative
1	%100	%100	0	0
2	%100	%100	0	0
3	%100	%95.12	1	1
4	%100	%95.12	2	0

MLP ağının test başarı oranlarının ortalaması %97.56, standart sapması ise 2.81 olarak hesaplanmıştır.

### 6.8.2 LVQ ađının eđitimi ve test edilmesi

LVQ ađlarının eđitimlerinde maksimum adım sayısı 500, bařlangıç öğrenme katsayıları ise 1. eđitim seti için 0.2, 2. 3. ve 4. eđitim setleri için 0.4 olarak belirlenmiřtir.

4 eđitim seti ile gerekleřtirilen eđitimler sonucunda oluřan ađların eđitim bařarı oranları ve ađların kendilerine daha önce gsterilmemiř örneklerden oluřan test seti ile test edilmeleri sonucunda elde edilen sınıflandırma bařarı oranları izelge 6.21’de grlmektedir.

izelge 6.21 LVQ ađının test sonuları (Deney 4)

Eđitim Seti	LVQ Eđitim Bařarı Oranı	LVQ Test Bařarı Oranı	False Positive	False Negative
1	%97.56	%100	0	0
2	%100	%100	0	0
3	%100	%100	0	0
4	%97.56	%100	0	0

LVQ ađının test bařarı oranlarının ortalaması %100, standart sapması ise 0 olarak hesaplanmıřtır.

### 6.8.3 SOM ađının eđitimi ve test edilmesi

SOM ađlarının eđitimlerinde maksimum adım sayısı 500, minimum öğrenme katsayısı 0 olarak belirlenmiřtir. Maksimum öğrenme katsayıları ise 1. eđitim seti için 0.96, 2. ve 4. eđitim setleri için 0.92, 3. eđitim seti için 0.99 olarak belirlenmiřtir.

4 eđitim seti ile gerekleřtirilen eđitimler sonucunda oluřan ađların eđitim bařarı oranları ve ađların kendilerine daha önce gsterilmemiř örneklerden oluřan test seti ile test edilmeleri sonucunda elde edilen sınıflandırma bařarı oranları izelge 6.22’de grlmektedir.

Çizelge 6.22 SOM ağıının test sonuçları (Deney 4)

Eğitim Seti	SOM Eğitim Başarı Oranı	SOM Test Başarı Oranı	False Positive	False Negative
1	%95.12	%97.56	1	0
2	%100	%100	0	0
3	%97.56	%95.12	2	0
4	%97.56	%97.56	1	0

SOM ağıının test başarı oranlarının ortalaması %96.95, standart sapması ise 2.33 olarak hesaplanmıştır.

#### 6.8.4 Ağların birleştirilmesi

4 eğitim setinden her biri ile eğitilen üç ağ modeli ortak çalışmak üzere birleştirildiğinde verilen kararların başarı oranları Çizelge 6.23'te görülmektedir.

Çizelge 6.23 Bileşik ağıın test sonuçları (Deney 4)

Eğitim Seti	Bileşik Ağ Test Başarı Oranı	False Positive	False Negative
1	%100	0	0
2	%100	0	0
3	%100	0	0
4	%100	0	0

Bileşik ağıın test başarı oranlarının ortalaması 100, standart sapması ise 0 olarak hesaplanmıştır.

Sonuç olarak ağlar ayrı ayrı kullanıldıklarında elde edilen başarı oranlarının standart sapmalar MLP ağı için 2.81, LVQ ağı için 0 ve SOM ağı için 2.33 iken bileşik ağda 0'a düştüğü görülmektedir. Bu durum, bileşik ağıın eğitim setindeki değişimlerden daha az etkilendiğini, yani daha kararlı olduğunu ortaya koyar.

## 7. SONUÇ

Bu tez çalışması ile telekomünikasyon hizmetlerinde gerçekleştirilen sahtekârlık işlemlerini tespit eden bir sistem tasarlanmıştır. Bir telekomünikasyon şirketinin kullanıcılarının 06.11.2008 ile 19.03.2011 tarihleri arasındaki üyelik, kullanım ve ödeme verileri incelemeye alınmıştır. Bu veriler işlenerek her kullanıcıyı temsil eden 23 sahadan oluşan birer kayıttan oluşan bir veriseti hazırlanmıştır.

İlk deneyde verisetinden 50 tane sahtekâr, 50 tane normal kullanıcıya ilişkin kayıtlar işleme alınmıştır. Bu deneyde doğru sınıflandırma performansları MLP ağında %98.0, LVQ ağında %98.0, SOM ağında %96.0 ve bileşik ağda %100 olarak ölçülmüştür.

İkinci deneyde 62 tane sahtekâr, 266 tane normal kullanıcıya ilişkin 308 kayıt işleme alınmıştır. Bu deneyde doğru sınıflandırma performansları MLP ve LVQ ağlarında %96.34, SOM ağında %96.95 ve bileşik ağda %97.56 olarak ölçülmüştür.

İlk iki deneyde görüldüğü üzere üç farklı yapay sinir ağı yaklaşımının birlikte kullanıldığı durumda, her bir yaklaşımın tek başına kullanıldığında olduğundan daha yüksek başarı oranı elde edilmiştir. Bu durum, bir ağ modelinin bir desen için verdiği hatalı sonucun, diğer ağ modelleri tarafından düzeltilebildiğini göstermektedir.

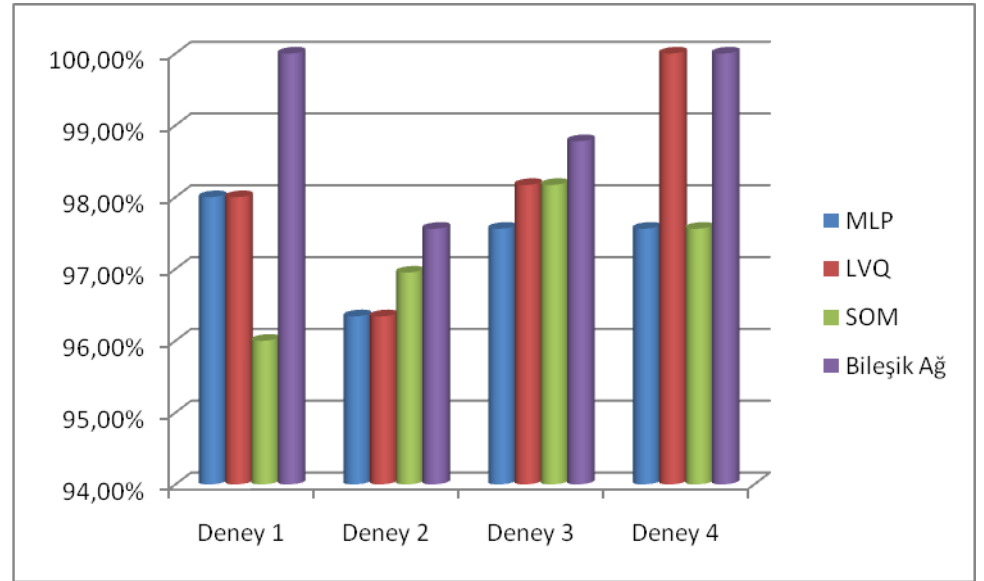
Üçüncü deneyde bir eğitim seti ile eğitilen ağın farklı test setleri ile test edilmesi durumunda elde edilen başarı oranları gösterilmesi amacıyla rastgele seçilmiş 41 kayıttan oluşan bir eğitim seti ve 41'er kayıttan oluşan 4 adet test seti işleme alınmıştır. Bu deneyde doğru sınıflandırma performanslarının ortalamaları MLP ağında %97.56, LVQ ve SOM ağlarında %98.17 ve bileşik ağda %98.78 olarak ölçülmüştür. Bu performansların standart sapmaları MLP ağında 1.99, LVQ ve SOM ağlarında 2.33 ve iken bileşik ağda 1.40 olarak ölçülmüştür.

Dördüncü deneyde ise farklı eğitim setleri ile eğitilen ağlarının aynı test setleri ile test edilmesi durumunda elde edilen başarı oranlarının gösterilmesi

amacıyla rastgele seçilmiş 41'er kayıttan oluşan 4 eğitim seti ve 41 kayıttan oluşan bir test seti işleme alınmıştır. Bu deneyde doğru sınıflandırma performanslarının ortalamaları MLP ağında %97.56, LVQ ağında %100 ve SOM ağında %97.56 ve bileşik ağda %100 olarak ölçülmüştür. Bu performansların standart sapmaları MLP ağında 2.81, LVQ ağında 0 ve SOM ağında 2.33 ve iken bileşik ağda 0 olarak ölçülmüştür.

Son iki deneyde görüldüğü üzere üç farklı yapay sinir ağı yaklaşımının birlikte kullanıldığı durumda, her bir yaklaşımın tek başına kullanıldığında olduğundan daha düşük bir standart sapma değeri elde edilmiştir. Bu durum, bileşik ağın eğitim ve test setlerindeki değişimlerden daha az etkilendiğini, yani daha kararlı olduğunu göstermektedir.

Deney sonuçlarına ilişkin sütun grafiği Şekil 7.1'de görülmektedir.



Şekil 7.1 Deney sonuçlarına ilişkin sütun grafiği

Algoritmaların çalışma süreleri; katman sayıları, katmanlardaki nöron sayıları ve adım sayılarına bağlı olarak değişmektedir. Öğrenme süreçleri, girdi ve çıktı katmanlarından oluşan ve en fazla 500 adım sayısına ulaşan LVQ ve SOM modellerinde ekran gösterimleri ile birlikte 1 saniye kadar sürerken 2. deneyde ara katmanında 5 nöron bulunan ve 1000 adım sayısı ile çalışan MLP ağında bu süre 2-3 saniyeye, 1., 2. ve 4. deneylerde ara katmanında 10 nöron bulunan ve 30000

adım sayısı ile çalışan MLP ađında ise 20-30 saniyeye kadar ıkabilmektedir. Test srelerinin alıřma sreleri tm ađlar iin 1 saniyeden kısa srmektedir.

Sistemin en nemli dezavantajı, ađ tasarımlarında bulunan katman ve iřlem elemanı sayısı arttıka đrenme sresinin artmasıdır. Yapay sinir ađlarının sonuların nasıl oluřturulduđunu aıklama yeteneđinin olmaması, ađa duyulan gveni azaltan bir dezavantaj olarak deđerlendirilebilir.

Gelecekte yapılacak alıřmalarda bileřik karar mekanizmasına dâhil edilmek zere farklı yapay sinir ađı modelleri, uzman sistemler, kural tabanlı yntemler veya istatistiksel yntemlere dayalı algoritmalar geliřtirilebilir. Her bir yntemin ortak kararı belirlemede farklı derecelerde etkili olması iin ađırlık deđerleri belirlenerek daha yksek bařarı oranlarının elde edilmesi sađlanabilir.

Veri setinde ve ıktı formatlarında yapılacak uyarlamalarla, geliřtirilen yazılımın ve kullanılan yntemlerin sađlık, mhendislik, finans gibi eřitli alanlara iliřkin tahmin, tespit, sınıflandırma teřhis gibi problemlerinin zmnde kullanılabilir hale getirilmesi mmkndr.

## KAYNAKLAR DİZİNİ

- Abu-hakima, S., Toloo, M., White, T.,** 1997, A multi-agent systems approach for fraud detection in personal communication systems, Proceedings of the 15<sup>th</sup> International Joint Conference on Artificial Intelligence (IJCAI-97), pp. 1-8
- Akdağ, Y.,** 2010, Vadeli İşlem ve Opsiyon Borsası'nda TL Dolar Vadeli İşlem Sözleşmelerinin Gün Sonu Uzlaşma Fiyatının Yapay Sinir Ağları ile Tahmini, Marmara Üniversitesi Bankacılık ve Sigortacılık Enstitüsü Sermaye Piyasası ve Borsa Anabilim Dalı, 117 s.
- Barghava, B., Zhong, Y., Lu, Y.,** 2003, Fraud Formalization and Detection , Data Warehousing and Knowledge Dscovery, Fifth International Conference, pp. 330-339
- Barson P., Field, S., Davey, N., McAskie, G., and Frank, R.,** 1996, The detection of fraud in mobile phone networks, Neural Network World, pp. 477-484
- Brause, R., Langsdorf, T., Hepp, M.,** 1999, Neural data mining for credit card fraud detection, Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, pp. 103-106
- Britos, p., Grosser, H., Rodríguez, D., Garcia-Martinez, R.,** 2008, Detecting unusual changes of users consumption, Artificial Intelligence in Theory and Practice II, IFIP 20th World Computer Congress, 297-306
- Burge, P., Shawe-Taylor, J., Cooke, C., Moreau, Y., Preneel, B., Stoermann, C.,** 1997, Fraud detection and management in mobile telecommunications networks, In Proceedings of the European Conference on Security and Detection ECOS 97, pp. 91-96
- Burge, P., Shawe-Taylor, J., Cooke, C., Moreau, Y., Preneel, B., Stoermann, C.,** 1997, Novel techniques for fraud detection in mobile telecommunication networks, ACTS Mobile Summit
- Cinsdikici, M.,** “Yapay Sinir Ağları”, Ders Notları, Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü, 28s.
- <http://ube.ege.edu.tr/~cinsdiki/UBI521/Chapter-1/cinsdikici-neural-net-giris.pdf> (Erişim tarihi: 2 Haziran 2011)

## KAYNAKLAR DİZİNİ (devam)

- Cox, K., Eick, S., Wills, G., Brachman, R.,** 1997, Visual data mining: recognizing telephone calling fraud, Data Mining and Knowledge Discovery, pp. 225-231
- Elmas, Ç.,** 2003, Yapay Sinir Ağları, Seçkin Yayıncılık, 192 s.
- Fausett, L.,** 1994, Fundamentals of Neural Networks, Prentice Hall
- Fawcett, T. and Provost, F.,** 1997, Adaptive fraud detection in data mining and knowledge discovery, Kluwer Academic Publishers, pp. 1-28.
- Ghosh, S., Reilly, D.,** 1994, Credit card fraud detection with a neural-network, Proceedings of the Annual International Conference on System Science, pp. 621-630.
- Güven, A.,** 2005, Göze Ait Elektrofizyolojik Sinyaller Kullanılarak Yapay Sinir Ağları Destekli Bazı Göz Hastalıklarının Teşhisi, Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalı, 92 s.
- Haykin, S.,** 1999, Neural Networks: A Comprehensive Foundation, Prentice Hall
- Kakıcı, A.,** 2009, "Yapay Sinir Ağlarına Giriş",  
<http://www.ahmetkakici.com/yazilim/yapay-sinir-aglarina-giris/> (Erişim tarihi: 2 Haziran 2011)
- Kocabaş, Ş.,** "Yapay Zeka Araştırma ve Uygulama Alanları",  
<http://www.sakirkocabas.com/files/bumat.rtf> (Erişim tarihi: 31 Mayıs 2011)
- Öztemel, E.,** 2006, Yapay Sinir Ağları, Papatya Yayınları, 232 s.
- Pejic-Bach, M.,** 2010, Invited Paper: Profiling intelligent systems applications in fraud detection and prevention: Survey of research articles, International Conference on Intelligent Systems, Modelling and Simulation, pp.80-85
- Rosset, S., Murad, U., Neumann, E., Idan, Y., Pinkas, G.,** 1999, Discovery of fraud rules for telecommunications - challenges and solutions, Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 409-413

**KAYNAKLAR DİZİNİ (devam)**

- Russell, J.S., Norvig, P.**, 2003, "Artificial Intelligence A Modern Approach", Second Edition
- Stergiou, C., Siganos, D.**, 1996, "Neural Networks", [http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/cs11/report.html#Feed-forward%20networks](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Feed-forward%20networks) (Erişim tarihi: 31 Mayıs 2011)
- Taniguchi, M., Haft, M., Hollmen, J., Tresp, V.**, 1998, Fraud detection in communication networks using neural and probabilistic methods, Proceedings of the 1998 IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP'98), pp. 1241-1244
- Uğur, A.**, 2005, "Yapay Sinir Ağları", Ders Notları, Ege Üniversitesi Bilgisayar Mühendisliği Bölümü, 46s.
- Uğur, A., Kınacı, A. C.**, 2006, "Yapay Zeka Teknikleri ve Yapay Sinir Ağları Kullanılarak Web Sayfalarının Sınıflandırılması"
- Uğur, A.**, 2008, "Günlük Yaşamda Yapay Zeka", Ders Notları, Ege Üniversitesi Bilgisayar Mühendisliği Bölümü, 57s.
- Wikipedia**, CDR, <http://en.wikipedia.org/wiki/Cdr>, (Erişim Tarihi: 20 Mayıs 2010)
- Zaslavsky, V., Strizhak, A.**, 2006, Credit card fraud detection using self organizing maps, Cybercrime and Cybersecurity, pp. 48-63

**ÖZGEÇMİŞ**

**Adı Soyadı** : Cansu SEZEN  
**Doğum Tarihi** : 09.03.1987  
**Doğum Yeri** : İzmir  
**Uyruđu** : T.C.  
**Lise** : İzmir Özel Türk Fen Lisesi (2001-2004)  
**Lisans** : Ege Üniversitesi Bilgisayar Mühendisliđi (2004-2008)  
**Yüksek Lisans** : Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü (2008-2011)  
**Çalıřtıđı Kurum** : Ünipa A.Ş.  
**Görevi** : Yazılım Mühendisi  
**Yabancı Dil** : İngilizce  
**E-posta adresi** : [cansusezen@gmail.com](mailto:cansusezen@gmail.com)  
**Tel** : 90 (555) 839 25 26