DOKUZ EYLÜL ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİKSEL KALİTE KONTROL KARTLARINDA YAPAY SİNİR AĞLARI İLE ÖRÜNTÜ TANIMA

Caner KONUK

Eylül, 2023

İZMİR

İSTATİSTİKSEL KALİTE KONTROL KARTLARINDA YAPAY SİNİR AĞLARI İLE ÖRÜNTÜ TANIMA

Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi
İstatistik Anabilim Dalı, Veri Bilimi Programı

Caner KONUK

Eylül, 2023
İZMİR

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

CANER KONUK tarafından **PROF. DR. BURCU HÜDAVERDİ** yönetiminde hazırlanan **"İSTATİSTİKSEL KALİTE KONTROL KARTLARINDA YAPAY SİNİR AĞLARI İLE ÖRÜNTÜ TANIMA"** başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

.....

Prof. Dr. Burcu HÜDAVERDİ

Danışman

.....

Prof. Dr. Neslihan DEMİREL

Jüri Üyesi

.....

Prof. Dr. Gözde ULUTAGAY

Jüri Üyesi

Prof. Dr. Okan FISTIKOĞLU

Müdür

Fen Bilimleri Enstitüsü

TEŞEKKÜR

Tez çalışmam boyunca değerli bilgileri ve görüşleri ile bana yol gösteren, motive eden ve hiçbir zaman desteğini esirgemeyen danışmanım Prof. Dr. Burcu HÜDAVERDİ'ye en içten teşekkürlerimi sunarım.

İstatistiksel kalite kontrol alanındaki engin tecrübe ve bilgisi ile bu tez konusunu seçmemi ve bu alanda ilerlememi sağlayan, ayrıca uygulama aşamasındaki çok değerli katkıları ile bana yol gösteren danışmanım Dr. Senem VAHAPLAR'a en içten teşekkürlerimi sunarım.

Akademik alandaki deneyimi ve bilgisiyle bu süreçte bana en önemli zamanlarımda destek olan ablam Dr. Dilara KONUK TAKMA'ya, son olarak beni bugünlere getiren ve aldığım her kararda desteklerini hiçbir zaman esirgemeyen annem Hülya KONUK'a ve babam Yaşar KONUK'a sonsuz teşekkürlerimi sunarım.

Caner KONUK

İSTATİSTİKSEL KALİTE KONTROL KARTLARINDA YAPAY SİNİR AĞLARI İLE ÖRÜNTÜ TANIMA

ÖZ

İstatistiksel kalite kontrolü, veriye dayalı süreçlerden elde edilen ürünler veya hizmetler için kalite kontrol standartlarına uygunluğun ölçülmesini sağlayan bir yaklaşımdır. Kontrol kartları ise istatistiksel kalite kontrol yöntemlerinden birisidir. Kontrol kartları, süreçlerde ilgilenilen kalite karakteristiklerine göre sürecin görsel bir görünümünü sunmaktadır. Dolayısıyla, ilgili süreç için kontrol kartlarında ortaya çıkan örüntülerin incelenmesi sürecin kararlılığı açısından büyük önem taşımaktadır.

Bu çalışmanın asıl amacı, temel kontrol kartı örüntülerine ek olarak eş zamanlı kontrol kartı örüntülerinin de yapay sinir ağı modelleri ile tanınmasının gerçekleştirilmesidir. Literatür incelendiğinde çoğu yapay sinir ağı yöntemi içeren çalışmada sadece temel örüntülerin tanınmasının gerçekleştirildiği ve eşzamanlı örüntüler için ise çoğunlukla derin öğrenme yöntemlerinin kullanıldığı görülmektedir. Dolayısıyla, bu çalışmanın bir diğer amacı ise çok katmanlı algılayıcı ve öğrenmeli vektör kuantizasyon, yapay sinir ağı modellerinin eşzamanlı örüntüler açısından ortaya koydukları örüntü tanıma performanslarının da detaylıca incelenmesini sağlamaktır.

Bu çalışmada, kontrol kartlarında örüntü tanıma literatüründeki iki güçlü yapay sinir ağı modeli olan çok katmanlı algılayıcı ve öğrenmeli vektör kuantizasyon modelleri kullanılarak 7 temel ve 9 eşzamanlı örüntü olmak üzere toplamda 16 adet Shewhart kontrol kartı örüntüsü için kontrol kartı örüntü tanıma çalışması gerçekleştirilmiştir. Elde edilen sonuçlar detaylı olarak incelenmiş ve en iyi örüntü tanıyıcı modelinin belirlenmesi sağlanmıştır. Ayrıca sonuçlara göre kontrol kartı örüntü tanıma literatüründe sıkça kullanılan bu iki yapay sinir ağı modelinin detaylı olarak karşılaştırılması ve yorumlanması gerçekleştirilmiştir.

Anahtar kelimeler: Kontrol kartlarında örüntü tanıma, istatistiksel kalite kontrol, yapay sinir ağları, çok katmanlı algılayıcı, öğrenmeli vektör kuantizasyon

PATTERN RECOGNITION WITH NEURAL NETWORKS IN STATISTICAL QUALITY CONTROL CHARTS

ABSTRACT

Statistical quality control is an essential approach for assessing compliance with quality control standards in data – driven processes, ensuring the quality of products and services. Control charts play a vital role in this methodology by providing visual representations of process performance based on relevant quality characteristics. Consequently, analyzing patterns emerging in control charts holds significant importance in maintaining process stability.

The main objective of this study is to achieve the recognition of concurrent control chart patterns in addition to basic control chart patterns using artificial neural network models. Based on literature review, artificial neural network studies primarily focus on recognizing basic patterns, while deep learning methods are more commonly used for concurrent patterns recognition. Therefore, another aim of this study is to comprehensively examine the pattern recognition performance of multilayer perceptron and learning vector quantization, artificial neural network models, specifically in relation to concurrent patterns.

In this study, a control chart pattern recognition analysis was conducted using two powerful artificial neural network models in the control chart literature which are multilayer perceptron and learning vector quantization models. The study aimed to recognize a total of 16 Shewhart control chart patterns, including 7 basic patterns and 9 concurrent patterns. The obtained results were thoroughly examined, allowing for with detailed analysis and determination of the best pattern recognition model. In addition, a comprehensive comparison and interpretation of these two frequently used artificial neural network models in the control chart pattern recognition literature were carried out based on the results.

Keywords: Pattern recognition in control charts, statistical quality control, artificial neural networks, multilayer perceptron, learning vector quantization

İÇİNDEKİLER

Sayfa

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU ii
TEŞEKKÜR iii
ÖZ iv
ABSTRACT v
İÇİNDEKİLER vi
ŞEKİLLER LİSTESİ ix
TABLOLAR LİSTESİ xi
KISALTMALAR xii

BÖLÜM BİR – GİRİŞ 1

BÖLÜM İKİ – İSTATİSTİKSEL KALİTE KONTROLÜ 8

2.1 Kalite Kavramı ve Tanımı 8
2.2 Kalitenin Boyutları 9
2.3 Toplam Kalite Yönetimi 10
2.4 İstatistiksel Kalite Kontrolü 11
2.4.1 Histogram 12
2.4.2 Çetele Diyagramı 13
2.4.3 Pareto Diyagramı 13
2.4.4 Neden – Sonuç Diyagramı 16
2.4.5 Tabakalama 17
2.4.6 Serpme Diyagramı 17
2.4.7 Kontrol Kartları 18

BÖLÜM ÜÇ – İSTATİSTİKSEL KALİTE KONTROL KARTLARI 19

3.1 Kontrol Kartları 19
3.2 Kontrol Kartlarının Yapısı 20
3.3 Süreçte Varyasyona Neden Olan Faktörler 21
3.4 Kontrol Kartlarında I. tip (α) ve II. Tip (β) Hataları 23
3.5 Kontrol Kartlarında Aşama I ve Aşama II 26

3.6 Kontrol Kartlarının Çeşitleri 27
3.7 Shewhart Kontrol Kartları 28
3.8 Ölçülebilen Kalite Karakteristikleri için Shewhart Kontrol Kartları 29
3.8.1 $\overline{X} - R$ (Aritmetik Ortalama ve Değişim Aralığı) ve $\overline{X} - S$ (Aritmetik Ortalama ve Standart Sapma) Shewhart Kontrol Kartları 30
3.9 Ölçülemeyen Kalite Karakteristikleri için Shewhart Kontrol Kartları 33
3.9.1 p Kontrol Kartı 33
3.9.2 np Kontrol Kartı 34
3.9.3 c Kontrol Kartı 35
3.9.4 u Kontrol Kartı 36
3.10 Shewhart Kontrol Kartlarında Örüntüler 36

BÖLÜM DÖRT – YAPAY SİNİR AĞLARI 38

4.1 Yapay Sinir Ağlarına Giriş 38
4.2 Biyolojik Sinir Hücresi 38
4.3 Yapay Sinir Hücresi 39
4.3.1 Girdiler 41
4.3.2 Ağırlıklar 41
4.3.3 Toplama Fonksiyonları 42
4.3.4 Aktivasyon Fonksiyonları 43
4.3.5 Hücrenin Çıktısı 44
4.4 Yapay Sinir Ağlarının Yapısı 44
4.5 Yapay Sinir Ağı Mimarileri 46
4.5.1 İleri Beslemeli Yapay Sinir Ağları 46
4.5.2 Geri Beslemeli Yapay Sinir Ağları 47
4.6 Yapay Sinir Ağı Modelleri 48
4.7 Yapay Sinir Ağlarında Öğrenme ve Test 49
4.7.1 Hata Matrisi 51
4.8 Çok Katmanlı Algılayıcı (Multilayer Perceptron) Modeli 55
4.8.1 Geri Yayılım Algoritması 56
4.9 Öğrenmeli Vektör Kuantizasyon (Learning Vector Quantization) Modeli 57

BÖLÜM BEŞ – UYGULAMA 62

5.1 Çalışmanın Tanıtılması 62
5.2 Verilerin Elde Edilmesi 62
5.3 Çok Katmanlı Algılayıcı Modelinin Yapısı ve Eğitimi 67
5.4 Öğrenmeli Vektör Kuantizasyon Modelinin Yapısı ve Eğitimi 69

BÖLÜM ALTI – SONUÇLAR 72

KAYNAKLAR 84

EKLER 93

EK 1: $\overline{X}$ kontrol kartı için ilgili katsayılar ve örneklem boyutuna göre değerleri (Demir, 2008) 93
EK 2: R kontrol kartı için ilgili katsayılar ve örneklem boyutuna göre değerleri (Demir, 2008) 94
EK 3: S kontrol kartı için ilgili katsayılar ve örneklem boyutuna göre değerleri (Demir, 2008) 95
EK 4: ÇKA modeli ile gerçekleştirilen eğitimler için kullanılan MATLAB® kodları 96
EK 5: ÖVK modeli ile gerçekleştirilen eğitimler için kullanılan MATLAB® kodları 100

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1 Başlıca histogram şekilleri (Dengizler Kayaalp, 2007) 13
Şekil 2.2 Sipariş formları için oluşturulan pareto diyagramı örneği (Juran ve Godfrey, 1998) 15
Şekil 2.3 Bir süreçte meydana gelebilecek mobil vinç kazası sorunu için oluşturulan neden-sonuç diyagramı (Atalay ve Kılıç, 2015) 16
Şekil 2.4 X ve Y kalite karakteristikleri için bazı örnek serpme diyagramı türleri (Western Electric Company, 1956) 17
Şekil 3.1 Örnek bir kalite kontrol kartı (Montgomery, 2013) 20
Şekil 3.2 Oluşan varyasyonun yaygın ve özel faktörleri (Montgomery, 2013) 22
Şekil 3.3 Kontrol kartlarındaki varyasyonlar (Yılmaz, 2019) 23
Şekil 3.4 Bir $\overline{X}$ kontrol kartında I. tip (α) hata (Swamidass, 2000) 24
Şekil 3.5 Bir $\overline{X}$ kontrol kartında II. tip (β) hata (Swamidass, 2000) 25
Şekil 3.6 Ölçülebilen ve ölçülemeyen kalite karakteristikleri için SKK türleri (Koltan Yılmaz, 2014) 29
Şekil 3.7 Temel Shewhart kontrol kartı örüntüleri 37
Şekil 4.1 İnsan beyninde bulunan bir nöronun yapısı (Corson ve Aziz-Alaoui, 2009) 39
Şekil 4.2 Biyolojik sinir hücresi; (B) Yapay sinir hücresi; (C) Biyolojik sinaps; (D) YSA'daki sinapslar (Goncalves vd., 2013) 40
Şekil 4.3 Bir yapay sinir hücresinde bulunan elemanların detaylı gösterimi (Şenol, 2022) 40
Şekil 4.4 YSA'da kullanılan yaygın aktivasyon fonksiyonlarının grafikleri (Ghasemiyeh vd., 2017) 43
Şekil 4.5 Üç gizli katmana sahip bir çok katmanlı YSA modelinin yapısı (Bre vd., 2018) 45
Şekil 4.6 Örnek bir ileri beslemeli YSA mimarisi (Haykin, 2008) 47
Şekil 4.7 Örnek bir geri beslemeli YSA mimarisi (Suzuki, 2011) 48
Şekil 4.8 Mimarilerine göre temel YSA modelleri (Verma ve Kumar Singh, 2015) 49
Şekil 4.9 Çoklu sınıflandırma için hata matrisi yapısı (Krüger, 2016) 53

Şekil 4.10 ÇKA modeli eğitimi sırasında izlenen adımların diyagramı (Pham ve Oztemel, 1993) 55
Şekil 4.11 Örnek bir ÖVK ağ modelinin yapısı (Pelàez ve Castillo Piqueira, 2008) 59
Şekil 4.12 ÖVK modeli eğitimi sırasında izlenen adımların diyagramı (Pham ve Oztemel, 1993) 61
Şekil 5.1 Eş zamanlı kontrol kartı örüntüleri 65
Şekil 5.2 ÖVK modeli için kullanılan aktivasyon fonksiyonlarının grafikleri (Koltan Yılmaz, 2014) 70
Şekil 6.1 TRAINBR – 3 modeli için MATLAB® ortamından elde edilen ağ mimarisi diyagramı 74
Şekil 6.2 Test veri seti için MATLAB® ortamında TRAINBR – 3 modeli ile elde edilen hata matrisi grafiği (Toplam 6000 örnek, her örüntü için 375 örnek bulunmaktadır.) 76
Şekil 6.3 Eğitim veri seti için MATLAB® ortamında TRAINBR – 3 modeli ile elde edilen hata matrisi grafiği (Toplam 24000 örnek, her örüntü için 1500 örnek bulunmaktadır.) 77
Şekil 6.4 ÖVK – 1 modeli için MATLAB® ortamından elde edilen ağ mimarisi diyagramı 78
Şekil 6.5 Test veri seti için MATLAB® ortamında ÖVK – 1 modeli ile elde edilen hata matrisi grafiği (Toplam 6000 örnek, her örüntü için 375 örnek bulunmaktadır.) 79
Şekil 6.6 Eğitim veri seti için MATLAB® ortamında ÖVK – 1 modeli ile elde edilen hata matrisi grafiği (Toplam 24000 örnek, her örüntü için 1500 örnek bulunmaktadır.) 80

TABLOLAR LİSTESİ

Sayfa

Tablo 2.1 Bir sipariş sürecindeki formlar için oluşturulmuş örnek bir pareto tablosu (Juran ve Godfrey, 1998) 14
Tablo 4.1 YSA elemanlarının biyolojik sinir sistemindeki karşılıkları (Demir, 2018) 41
Tablo 4.2 Yaygın olarak kullanılan toplama fonksiyonları (Günay, 2018) 42
Tablo 4.3 İkili sınıflandırma problemleri için örnek bir hata matrisi örneği 51
Tablo 4.4 Sınıflandırma problemleri için yaygın olarak kullanılan performans metrikleri (Lever vd., 2016; Lipton vd., 2014) 52
Tablo 5.1 Temel yedi kontrol kartı örüntüsünü oluşturan süreçler için simülasyon formülleri (Hong vd., 2019) 63
Tablo 5.2 İlgili örüntülerin hedef çıktıları 66
Tablo 5.3 ÇKA modeli ve traingdx eğitim algoritması ile elde edilen sonuçlar 68
Tablo 5.4 ÇKA modeli ve trainlm eğitim algoritması ile elde edilen sonuçlar 69
Tablo 5.5 ÇKA modeli ve trainbr eğitim algoritması ile elde edilen sonuçlar 69
Tablo 5.6 ÖVK modeli ile elde edilen sonuçlar 71
Tablo 6.1 7 temel ve 13 eş zamanlı olmak üzere 20 adet kontrol kartı örüntüsü için eğitilen ÇKA modellerinden elde edilen sonuçlar 73
Tablo 6.2 Eğitim algoritmalarına göre test ve eğitim veri setleri için elde edilen ortalama örüntü tanıma oranları 81

KISALTMALAR

AAK	: Aşağı Ani Kayma
AKL	: Alt Kontrol Limiti
ART	: Artan Trend
ASQC	: Amerikan Kalite Kontrol Derneği
AZT	: Azalan Trend
CUSUM	: Kümülatif Toplam Kontrol Kartı
ÇKA	: Çok Katmanlı Algılayıcı
EOQC	: Avrupa Kalite Kontrol Organizasyonu
EWMA	: Üstel Ağırlıklı Hareketli Ortalama Kontrol Kartı
GN	: Gerçek Negatif
GP	: Gerçek Pozitif
İKK	: İstatistiksel Kalite Kontrolü
KKÖT	: Kontrol Kartı Örüntü Tanıma
MÇ	: Merkez Çizgi
NOR	: Normal
SİS	: Sistematik
SKK	: Shewhart Kontrol Kartı
Tansig	: Hiperbolik Tanjant Aktivasyon Fonksiyonu
TEÇ	: Tekrarlayan Çevrim
Trainbr	: Bayes Düzenlileştirme Geri Yayılım Algoritması
Traingdx	: Momentum ve Adaptif Öğrenme Oranı ile Gradyan İniş Geri Yayılım Algoritması
Trainlm	: Levenberg – Marquardt Geri Yayılım Algoritması
TSE	: Türk Standartları Enstitüsü
ÜKL	: Üst Kontrol Limiti
ÖVK	: Öğrenmeli Vektör Kuantizasyon
YAK	: Yukarı Ani Kayma
YN	: Yanlış Negatif
YP	: Yanlış Pozitif
YSA	: Yapay Sinir Ağları

BÖLÜM BİR
GİRİŞ

Geçmişten günümüze gelişen teknoloji, artan nüfus ve kaynaklara erişimin artması gibi birçok neden beraberinde artan tüketimi de getirmiştir. Artan talepler, rekabet ve globalleşme karşısında endüstriyel yeniliklerin gerçekleşmesi kaçınılmaz olmuştur. İkinci Dünya Savaşı'nın ardından seri üretimin yaygınlaşması ve sanayi devrimleri ile birlikte ihtiyaçlar ve tüketici istekleri değişime uğramıştır (Kaya vd., 2005). Bu durum ise geçmişi 1920'lerin başına dayanan istatistiksel kalite kontrolü (İKK) yöntemlerinin hızlı bir şekilde gelişerek günümüze kadar ilerlemesini sağlamıştır.

2011 yılında ortaya çıkan ve Dördüncü Sanayi Devrimi olarak da adlandırılan Endüstri 4.0 ile birlikte, akıllı sistemler içeren otonom üretim süreçlerinin oluşturulması hedeflenmiş ve süreçlerden daha çevik, tutarlı ve sağlam çıktılar elde edilmesi amaçlanmıştır. Genel olarak Endüstri 4.0, makine ağırlıklı üretim yerine dijital ağırlıklı üretime geçişi hedeflemektedir. Dolayısıyla Endüstri 4.0 ve teknoloji alanında yaşanan ciddi gelişmelerle birlikte, karanlık fabrika olarak da adlandırılan ve sıfır insan etkisi içeren üretim süreçlerinin oluşturulması için gereken motivasyonun artması sağlanmıştır (Oztemel ve Gursev, 2020).

Endüstri ve kalite kontrol alanlarındaki gelişmelerin yanı sıra son yıllarda gerçekleşen birçok yazılımsal ve donanımsal gelişme, büyük veri alanındaki gelişimin hızlı bir şekilde ivme kazanmasını sağlamıştır. Algılama teknolojilerindeki gelişim, veri sayısında gerçekleşen büyük miktardaki artışı da beraberinde getirmiştir. Ayrıca analitik platformların artışı, büyük miktardaki veriyi işlemek için gerekli hesaplama kaynaklarını arttırmıştır. Dolayısıyla, verilerden anlamlı bilgiler elde etmemiz daha da kolaylaşmıştır (Megahed ve Jones-Farmer, 2015; Qiu, 2020).

Büyük veri alanındaki bu gelişmeler yapay zekâ alanını da etkilemiştir. Veri sayısındaki artış özellikle yapay zekâ alanının bir alt alanı olan ve bilgisayar ortamında veriye dayalı olarak tahmin yapılmasını sağlayan makine öğrenmesi alanında büyük gelişmelere yol açmıştır. Bu gelişmelerle birlikte daha karmaşık makine öğrenmesi

modelleri eğitilebilir duruma gelmiştir. Bundan dolayı, birçok yeni makine öğrenmesi modeli ve bu modellerde öğrenmeyi sağlayan eğitim algoritmaları geliştirilmiştir. Dolayısıyla günümüzde, makine öğrenmesi alanında birçok farklı yöntem ve eğitim algoritması bulunmaktadır. Bu yöntemlerden birisi de yapay sinir ağlarıdır.

Yapay sinir ağları (YSA), insan beyninde bulunan biyolojik sinir hücrelerinin oluşturduğu sinaptik bağları, yapay sinir hücreleri ile taklit ederek insandaki öğrenme işlevinin bilgisayar ortamına aktarılmasını sağlayan bir makine öğrenmesi algoritmasıdır. YSA'da bulunan en temel işlem elemanı yapay sinir hücreleridir. Yapay sinir hücreleri, insan beynindeki biyolojik sinir hücrelerinin matematiksel olarak modellenmesiyle oluşturulmuş olan yapılardır. Dolayısıyla, insan beyninde karmaşık bir problem için belki de uzun yıllar gerçekleşebilecek olan bir öğrenme süreci, YSA ile bilgisayar ortamında daha kısa sürede gerçekleşebilmektedir. Böylelikle çok karmaşık bir problem için oluşturulmuş olan bir YSA modeli, ilgili problem için etkili çözümler sunabilmektedir. Ayrıca YSA, derin öğrenme alanının da temelini oluşturmaktadır.

Öte yandan süreç kavramı, hedeflenen bir sonucu üretmek için birlikte çalışan herhangi bir nedenler veya koşullar dizisidir. Dolayısıyla bir süreç, çok basit veya çok karmaşık işlemler içerebilmektedir (Western Electric Company, 1956). İKK, her ne kadar endüstriyel üretim süreçleri açısından ortaya çıkmış olsa da veriye dayalı yöntemlerden dolayı günümüzde birçok farklı alanda, birçok farklı süreç için uygulanabilmektedir. Bundan dolayı İKK yöntemlerini, sadece endüstriyel süreçlerde uygulanan yöntemler olarak düşünmemek gerekir. Kalite kontrol literatürünün öncüsü sayılan Walter Andrew Shewhart, herhangi bir süreçte gerçekleşebilecek olan varyasyonları, yaygın ve özel faktörler olmak üzere ikiye ayırmıştır. Ardından, varyasyonları oluşturan bu faktörleri inceleyebilmek için ise kontrol kartlarını geliştirmiştir. Temel İKK araçlarının en gelişmişi olan kontrol kartlarından ilkinin tanıtımı, 1924 yılının mayıs ayında Walter Andrew Shewhart tarafından gerçekleştirilmiştir (Best ve Neuhauser, 2006).

Kontrol kartları, süreçlerin grafiksel olarak görsel bir gösterimini ortaya koyduğu için gerçek zamanlı süreç izleme problemlerinde büyük kolaylık sağlamaktadır. Gerçek zamanlı süreçlerde, kontrol kartları sayesinde süreç içerisinde oluşabilecek herhangi bir sorun algılanabilir ve teşhis edilebilir. Dolayısıyla, ilgili sorunun çözümü en kısa sürede gerçekleştirilebilir ve ilgili süreç açısından maliyet kaybı minimum düzeye indirilebilir. Ayrıca ilgili süreç düzenli olarak kontrol kartları ile izlenerek, kararlı bir durumda işleyişine devam edebilir. Böylelikle, kontrol kartları ile bir süreçte ortaya çıkabilecek olan varyasyonlar ve kayıplar minimum düzeye indirgenebilir.

Kontrol kartlarındaki örüntülerin tanınması ile ilgili yapılan çalışmaların başlangıcı 1980'li yılların sonlarına dayanmaktadır. İlk çalışmalar, ham veri ile temel örüntülerin tanınmasına odaklanmış olup, zamanla ham veri yerine ham verilerden elde edilen belirli istatistiksel özelliklere göre örüntü tanıma çalışmaları gerçekleştirilmiştir. Kontrol kartlarındaki temel örüntü sayısı 6 ile 9 arasında değişmektedir. Temel örüntülerden bir tanesi olan normal örüntü, süreçte herhangi bir özel faktör bulunmadığını belirtir. Bundan dolayı kontrol kartı ile izlenilen bir süreçte normal örüntünün elde edilmesi sürecin kontrol altında olduğunun bir göstergesidir. Öte yandan literatür incelendiğinde, normal örüntü dışındaki diğer temel örüntülerin anormal örüntüler olarak tanımlandığı görülmektedir. Kontrol kartı ile izlenilen bir süreçte anormal örüntülerden birinin oluşması süreçte özel bir faktör bulunduğunun göstergesidir. Buradan yola çıkarak, kalite kontrol kartlarında YSA kullanılarak gerçekleştirilen literatürdeki önemli örüntü tanıma çalışmaları şu şekilde özetlenebilir:

Pham ve Oztemel (1993), çalışmalarında 6 temel örüntünün tanınması için öğrenmeli vektör kuantizasyon (ÖVK) ve çok katmanlı algılayıcı (ÇKA) YSA modelleri ile sırasıyla kontrol kartı örüntü tanıma (KKÖT) çalışması gerçekleştirmişlerdir. Ardından bu iki ağ tipinin kombinasyonu ile farklı bir KKÖT çalışması gerçekleştirmişlerdir. ÇKA ve ÖVK modellerinin kombinasyonu ile oluşturulan yeni modelden, diğer modellere göre daha iyi bir sonuç elde etmişlerdir.

Pham ve Oztemel (1994), çalışmalarında ÖVK modelini daha da geliştirerek 6 temel örüntünün tanınmasını gerçekleştirmişlerdir. Geliştirilmiş olan ÖVK

modelinden elde edilen sonuçlar ile standart ÖVK modeli ve geri yayılımlı YSA modelinden elde edilen sonuçları karşılaştırmışlardır. Sonuçlara göre geliştirilmiş ÖVK modeliyle, diğer modellere göre en iyi sonucu elde etmişlerdir.

Cheng (1997), çalışmasında 5 temel örüntünün tanınması için geri yayılımlı ÇKA modeli ve modüler YSA modeli ile KKÖT çalışması gerçekleştirmiştir. Modüler yapay sinir ağı modeli ile daha iyi bir sonuç elde etmiştir.

Pham ve Wani (1997), çalışmalarında 6 temel örüntünün tanınması için ham veri yerine, ham verilerden elde edilen 9 adet özelliği içeren özellikli veriyi kullanmışlardır. Örüntü tanıyıcı olarak ise buluşsal yöntemler ile elde edilen karar ağacı, tümevarım yöntemi ve geri yayılımlı ÇKA modelini kullanmışlardır. Özellikli veri ile eğitimi gerçekleştirilen geri yayılımlı ÇKA modelinden, diğer örüntü tanıyıcılara göre en iyi sonucu elde etmişlerdir.

Guh ve Tannock (1999), anormal örüntülerin tanınmasını ve anormal örüntülerin ortaya çıkmasında etkili olan özel faktörlerin belirlenebilmesinde büyük öneme sahip olan anahtar örüntü parametrelerini belirleyebilmek için çalışmalarında, 2 aşamalı YSA yöntemini kullanmışlardır. İki aşamalı YSA yönteminde her iki aşama için de geri yayılımlı ÇKA modelini kullanmışlardır. Böylelikle hem ilgili örüntünün hem de ilgili örüntü için anahtar parametrenin tanınmasını gerçekleştirmişlerdir. Sonuçlara göre %96,38 örüntü tanıma oranı elde etmişlerdir.

Sağıroğlu vd. (2000), çalışmalarında 6 temel kontrol kartı örüntüsünü tanıyabilmek için üç çeşit eğitim algoritmasını karşılaştırmışlardır. Bundan dolayı ÇKA modeli ile sırasıyla genişletilmiş delta bar delta, geri yayılım ve esnek yayılım eğitim algoritmalarını kullanmışlardır. Bu üç eğitim algoritması ile oluşturdukları ÇKA modellerinin her üçünün de KKÖT için başarılı olduğu sonucunu ortaya koymuşlardır. Ardından geri yayılımlı ÇKA modeli ile 6 temel kontrol kartı örüntüsü için %97,73 örüntü tanıma oranı ile en iyi sonucu elde etmişlerdir. Ayrıca yine geri yayılımlı ÇKA modeli ile hem eğitim hızı hem de eğitim parametreleri açısından en iyi sonuca ulaşmışlardır.

Gauri ve Chakraborty (2008), çalışmalarında 8 temel kontrol kartı örüntüsünün tanınması için 7 adet özellik içeren, özellikli veri ile geri yayılımlı ÇKA modelini eğitmişlerdir. İlk olarak, belirlenen 13 özellikten korelasyonu yüksek olan 6'sını çıkartmışlardır. Ardından geriye kalan 7 şekil özelliğini içeren özellikli veri ve MATLAB® ortamındaki üç çeşit geri yayılım algoritması ile üç farklı ÇKA modeli eğitmişlerdir. Bu ÇKA modellerinden eğitim ve doğrulama veri setleri ile sırasıyla ortalama %95,88 ve %95,75 örüntü tanıma oranları elde etmişlerdir.

Masood ve Hassan (2008), çalışmalarında 6 temel örüntüyü tanıyabilmek için geri yayılımlı YSA modelini kullanmışlardır. Eğitimi gerçekleştirilen modeller için eğitim parametreleri belirlenirken deney tasarımı yönteminden faydalanmışlardır. Sonuçlara göre en iyi YSA modeli ile normal örüntüler için %100 ve anormal örüntüler için ise %98,83 örüntü tanıma oranları elde etmişlerdir.

Gauri (2010), çalışmasında 8 temel örüntüyü tanıyabilmek için ÖVK ve geri yayılımlı ÇKA modellerini kullanmıştır. 7 özellik içeren özellikli veri ile hem ÇKA hem de ÖVK modelleri eğitmiştir. Ayrıca ham veri ile ÖVK modelleri eğitmiştir. Elde edilen modelleri hem eğitim süreleri hem de örüntü tanıma oranları açısından karşılaştırmıştır. Özellikli veri ile eğitilen modeller incelendiğinde, ÇKA modelinin en iyi örüntü tanıma oranlarına sahip olduğunu fakat ÖVK modeli ile çok yakın performans gösterdiği sonucunu elde etmiştir. Eğitim süresi açısından ise ham veri ile eğitilen ÖVK modelinin en kısa sürede ve özellikli veri ile eğitilen ÇKA modelinin en uzun sürede eğitimini gerçekleştirdiği sonucuna ulaşmıştır. Ayrıca özellikli veri ile eğitilen ÖVK modelinin, ham veri ile eğitilen ÖVK modelinden daha yüksek örüntü tanıma oranı ortaya koyduğunu fakat daha fazla eğitim süresi gerektirdiğini ortaya koymuştur.

Olgun (2011), çalışmasında hem ham veri hem de 6 özellik içeren özellikli veri ile 6 temel örüntüyü tanımak için geri yayılımlı ÇKA modelini ve Naive Bayes sınıflandırıcısını kullanmıştır. Geri yayılımlı ÇKA modeli için MATLAB® ortamındaki üç çeşit geri yayılım algoritmasını kullanmıştır. Özellikli veri ile eğitilen

ÇKA modellerinin ve Naive Bayes örüntü tanıyıcılarının, ham veri ile eğitilenlere göre daha yüksek örüntü tanıma oranları sunduğu sonucunu elde etmiştir.

Koltan Yılmaz (2014), çalışmasında hazır beton üreten bir işletmede gerçekleşen süreç için önemli bir kalite karakteristiği olan beton basınç dayanımı değerlerini kullanarak ilgili süreçte 6 temel örüntünün tanınmasını sağlayan bir KKÖT çalışması gerçekleştirmiştir. Çalışmasının ilk aşamasında ilgili süreçteki beton basınç dayanımı değerleri için ideal ortalama ve standart sapma değerlerini, kalite kontrol standartlarına göre belirlemiştir. Ardından elde edilen bu değerleri, ilgili sürece göre 6 temel örüntü verilerinin oluşturulmasını sağlayan simülasyonda parametre olarak kullanmıştır. Çalışmanın ikinci aşamasında ise simülasyon yoluyla elde edilen verilerden hem ham veri seti hem de 6 özellik içeren özellikli veri setini oluşturmuştur. Her iki veri seti ile ÖVK modelleri eğitmiştir. Sonuçlara göre ham verilerle eğitilen ÖVK modelinden eğitim ve test veri setleri için sırasıyla %99,49 ve %98 örüntü tanıma oranları ile en iyi sonucu elde etmiştir. Ardından bu modelden elde edilen ağırlık matrisi değerleri ile MATLAB® ortamında görsel bir program oluşturmuş ve bu program yardımıyla ilgili süreç için girdisi yapılan verilere göre örüntü tanıma işleminin gerçekleştirilmesini sağlamıştır.

Geleneksel yöntemler incelendiğinde, süreç hakkında birçok bilgi sağlayan kontrol kartlarında, örüntülerin tanınmasını sağlamak genellikle kalite kontrol personelinin deneyim ve becerisine bağlıdır. Fakat bu durum kararlılığı bozulmaya elverişli gerçek zamanlı olarak izlenen süreçler için bir açıklık yaratabilir. Çünkü bu tip süreçlerde kalite kontrol personelinin ilgili süreci, kararlılığının bozulmaması açısından sürekli olarak izlemesi gerekmektedir. Bundan dolayı, etkili bir kontrol kartı örüntü tanıma sistemi ile bu açıklık telafi edilebilir ve minimum yanlış alarmla kontrol kartlarının süreç üzerinde daha iyi bir şekilde uygulanması sağlanabilir. Ayrıca iyi bir KKÖT sistemi, ortaya çıkan örüntülerin tarafsız ve tutarlı bir şekilde yorumlanmasını sağlayabilir (Gauri ve Chakraborty, 2008).

Endüstri 4.0 ve yapay zekâ (özellikle derin öğrenme) yöntemleri ile birlikte sıfır insan etkisi içeren süreçlerin hedeflenmesi, kontrol kartlarında örüntü tanımayı

gerçekleştiren otonom sistemlerin oluşturulmasına yönelik çalışmaların artmasını sağlamıştır. Dolayısıyla son yıllardaki çalışmalarda, tanınması amaçlanan örüntü sayısı artmaktadır. Çünkü kontrol kartlarındaki örüntülerin detaylı bir şekilde tanınması süreçlerde ortaya çıkabilecek olan özel faktörlerin daha kolay bir şekilde algılanmasını ve çözülmesini sağlayabilir. Bundan dolayı, otonom sistemler için bu durum sıfır insan etkisi hedefi açısından büyük bir ilerleme sağlayabilir.

Bu tez kapsamında yapılan çalışmada, üçüncü bölümde detaylıca bahsedilecek olan Shewhart'ın $\overline{X}$ kontrol kartında ortaya çıkan örüntülerin tanınması amaçlanmaktadır. Çalışma ile kontrol kartlarında ortaya çıkan 7 temel örüntünün yanı sıra, türetilmiş olan 9 eş zamanlı örüntünün daha tanınması hedeflenmiştir. Eş zamanlı kontrol kartı örüntüleri iki veya daha fazla temel kontrol kartı örüntüsünün aynı anda ortaya çıkması ile oluşmaktadır. Son yıllarda özellikle derin öğrenme yöntemleri kullanılarak gerçekleştirilen, eş zamanlı örüntülerin tanınmasına yönelik çalışmalarda artış olduğu görülmektedir. Bu çalışmada ise çoğu çalışmadan farklı olarak temel ve eş zamanlı örüntülerin, KKÖT literatüründe sıklıkla kullanılan ve iki güçlü YSA modeli olan, ÇKA ve ÖVK modelleri kullanılarak tanınması hedeflenmiştir. Çalışmada, MATLAB® ortamında oluşturulan YSA modellerinden elde edilen sonuçlar incelenerek, toplamda 16 kontrol kartı örüntüsünü en iyi tanıyan YSA modelinin belirlenmesi gerçekleştirilecektir.

Bu tez toplamda beş bölümden oluşmaktadır. İkinci bölümde, kalite kontrol için büyük önem taşıyan İKK metodolojisi ve İKK'da kullanılan yedi önemli araç ile ilgili genel bilgi sağlanmaktadır. Üçüncü bölümde, temel İKK yöntemlerinin en gelişmişi olarak kabul edilen İKK kartlarından ayrıntılı olarak bahsedilmektedir. Dördüncü bölümde, YSA modellerinin genel yapısından ve bu tez kapsamında gerçekleştirilecek olan uygulamada kullanılacak olan ÇKA ve ÖVK, YSA modelleri hakkında detaylı bilgi sunulmaktadır. Son olarak beşinci bölüm ise bu tez için gerçekleştirilecek olan uygulamadan elde edilen sonuçların, detaylı olarak incelenmesini ve yorumlanmasını içermektedir.

BÖLÜM İKİ
İSTATİSTİKSEL KALİTE KONTROLÜ

2.1 Kalite Kavramı ve Tanımı

Kalite kavramı, Latince "Qualis" kelimesinden türetilmiştir. Türk Dil Kurumu'nun, Türkçe'de Batı Kökenli Kelimeler Sözlüğüne göre kalite, "bir şeyin iyi veya kötü olma özelliği, nitelik" şeklinde tanımlanmaktadır.

Kalitenin tanımı bilim insanlarına ve sektör özelliklerine özgü olarak değişiklik göstermektedir. Bundan dolayı kalite kavramı için özünde çok sayıda tanım bulunmaktadır. Çalışma yapan bazı önemli araştırmacılara ve kuruluşlara göre bu tanımlardan bazıları şu şekildedir (Kurt, 2018):

- W. Edwards Deming (1986) kaliteyi, "Mevcut ve gelecekteki müşteri gereksinimlerinin karşılanması için gayret sarf etmek" olarak tanımlamaktadır.
- Joseph M. Juran (1988) kaliteyi, "Kullanıma uygunluk" olarak tanımlamaktadır.
- Amerikan Kalite Kontrol Derneği (ASQC) kaliteyi, "Bir mal ya da hizmetin belirli bir gerekliliği karşılayabilme yeteneklerini ortaya koyan karakteristiklerin tümü" şeklinde tanımlamaktadır.
- Avrupa Kalite Kontrol Organizasyonu (EOQC) kaliteyi, "Belirli bir malın veya hizmetin, tüketicinin hizmetlerine uygunluk derecesi" olarak tanımlamaktadır.
- Türk Standartları Enstitüsü (TSE) kaliteyi, "Bir ürün ya da hizmetin belirlenen veya olabilecek gereksinimleri karşılama yeteneğine dayanan özelliklerin toplamı" şeklinde tanımlamaktadır.

Bu tanımlar kalitenin çok yönlü olduğunu ve sadece üretilen bir ürün için değil birçok alanda bulunan hizmetler için kullanılabileceğini göstermektedir. Son yıllarda mevcut kalite tanımlarının çoğundaki en büyük ortak nokta ise endüstri veya kültürden bağımsız olarak, daha çok müşteri memnuniyetine odaklanılmasıdır. Bunun nedeni genellikle kuruluşların birinci sınıf olarak nitelendirilebilmesi için daha çok müşteriyi,

ürün veya hizmet tasarım aşamalarında geliştirmelerin merkezinde tutmasından kaynaklanmaktadır. Dolayısıyla kalitenin ölçülebilmesi hem tüketiciler hem de kuruluşlar açısından büyük bir öneme sahiptir (Wicks ve Roethlein, 2009).

2.2 Kalitenin Boyutları

Bir ürün veya hizmetin kalitesi çeşitli özelliklerle belirlenebilir. Kaliteyi belirleyebilmek için ilgili ölçümün, ürünün veya hizmetin hangi özelliği üzerinden yapılacağını anlamak çok önemlidir. Garvin (1987), kaliteyi sekiz özellik için detaylı olarak boyutlandırmıştır. Kalitenin bu özelliklere göre tanımları ve önemli noktaları şu şekildedir (Montgomery, 2013):

- **Performans:** Ürünün amaçlanan işi istenen şekilde yapıp, yapamayacağının ölçütüdür.
- **Güvenirlik:** Ürünün ne kadar sıklıkla arızalandığının ölçütüdür. Birçok ürün, hizmet süreci içerisinde bir miktar onarım gerektirir.
- **Dayanıklılık:** Ürünün ne kadar dayanıklı olduğunun göstergesidir. Ürünün etkin hizmet ömrü şeklinde de tanımlanabilir.
- **Servis Kolaylığı:** Ürünün onarımının ne kadar kolay, hızlı ve ekonomik bir şekilde gerçekleştiğinin göstergesidir.
- **Estetik:** Ürünün dış görünüşünün ne kadar iyi olduğunun göstergesidir. Çoğunlukla renk, stil, şekil ve paketleme gibi faktörler bu özellik için ön plana çıkmaktadır.
- **İşlev:** Ürünün ne işe yaradığının göstergesidir. Müşteriler çoğunlukla yüksek kaliteyi, temel işlevlerin yanı sıra ek işlevlere de sahip olan ürünler şeklinde tanımlamaktadır.
- **Algılanan Kalite:** Ürünün veya şirketin genel olarak itibarının ölçütüdür. Ürün arızalandığında veya düzenli olarak bazı gereksinimlerinin karşılanması gerektiğinde, şirketin ürün ile ilgili müşteriye nasıl davrandığının göstergesidir.
- **Standartlara Uygunluk:** Ürünün kendi alanında standartlara uygunluğunun göstergesidir.

Çoğu endüstriyel ve birçok iş alanında bu sekiz özellik kaliteyi boyutlandırmak için yeterlidir. Ayrıca yukarıda verilen bu özellikler, kalitenin çok yönlü olduğunun bir göstergesidir (Montgomery, 2013).

2.3 Toplam Kalite Yönetimi

Özellikle gelişen teknolojiyle birlikte işletmelerin hızlı ve sürekli değişen çevreye uyum sağlamaları gerekmektedir. İşletmeler rekabet ortamında varlıklarını devam ettirebilmek için değişiklikleri önceden tahmin etmek ve rekabet ortamında üstünlük sağlamak zorundadırlar. Bu rekabet ortamında ise üstünlük sağlamanın en etkili yolu; kalitenin geliştirilmesinden, verimliliğin arttırılmasından ve maliyetlerin azaltılmasından geçmektedir. Bu şartlar ortaya, ürün veya hizmet açısından müşteri memnuniyetinin en ön planda tutulduğu toplam kalite yönetimini ortaya çıkarmıştır. Toplam kalite yönetimi, işletmelerin bütün faaliyetlerinde muhtemel hataları önleyerek kalitenin arttırılmasını hedefleyen bir anlayıştır. Böylelikle hataların önlenmesi sağlanarak, günümüzde hızla artan arz ve talep açısından en temel faktör olan kalite için iyileştirmeler sağlanır. Bu iyileştirmelerin sonucu olarak ürün veya hizmet için maliyetin azalması, müşterinin beklentisinin karşılanması sağlanır (Değerli, 2006; Yıldırım, 2006).

Öte yandan İKK yöntemleri, toplam kalite yönetimi anlayışı açısından büyük bir öneme sahiptir. Çünkü İKK yöntemleri ilgilenilen süreç içerisinde ortaya çıkan olumsuzlukları ortaya çıkartarak, süreç için hedeflenen kalite iyileştirmelerini net bir şekilde sağlayabilir. Böylelikle süreçteki değişkenlik en aza indirgenerek müşteri tatmininde artış sağlanır. Toplam kalite yönetimi için müşteri tatminindeki artış durumu, özellikle gerçekleştirilmesi istenilen hedeflerden biridir (Değerli, 2006).

Bir diğer durum ise özellikle toplam kalite yönetiminin hedeflediği sıfır hata ile kalite anlayışının son yıllarda teknoloji alanındaki gelişmelerle birlikte daha da önem kazanmasıdır. Müşteriler son yıllarda herhangi bir ürün veya hizmet alırken çoğunlukla sıfır hata oluşmasını istemektedir. Bu durumun ortaya çıkmasında, elde edilen veri sayısının artması da çok büyük bir etkiye sahiptir. Sıfır hatanın elde

edilmesi zor olsa da elde edilen veriler ile sıfır hataya yakın bir süreç elde edilmesi olasıdır. Bundan dolayı ilgilenilen ürün veya hizmet için özellikle kalitenin genel boyutlarıyla ölçülmesi büyük önem taşımaktadır.

2.4 İstatistiksel Kalite Kontrolü

İKK, ilgilenilen süreç için elde edilen veriler üzerinde belirli yöntemler ile analizler gerçekleştirilerek, bu analizler ile süreçteki pratik sorunların çözümüne ve süreçten elde edilen ürün veya hizmetin kalitesinin arttırılmasına detaylı olarak olanak sağlayan bir kalite kontrol metodolojisidir. İKK her ne kadar kalite kontrol amacı ile uygulanıyor olsa da uygulanabilirliği bulunan her türlü faaliyet alanındaki süreçler için iyileştirme amaçlı kullanılabilir. İKK'nın kullanıldığı faaliyet alanlarına örnek olarak; mühendislik, işletme, denetim, yönetim ve muhasebe alanlarındaki süreçler verilebilir (Western Electric Company, 1956).

Son yıllarda, bilişim ve endüstri alanlarındaki gelişmelerle birlikte bir süreçten veri elde etmek çok daha kolay bir hâl almıştır. Geçmiş yıllarda kalite kontrolü denilince akla; ölçüm, algılama ve muayene gibi işlemler için geleneksel yöntemler gelmekteyken günümüzde büyük veri uygulamalarındaki gelişmelerle birlikte geleneksel kalite kontrol teknikleri yerini İKK tekniklerine bırakmaktadır.

Eğer bir ürün veya hizmet müşterilerin beklentilerini karşılayacaksa genellikle kararlı veya tekrarlanabilir bir süreç ile üretilmelidir. Açıkça ilgilenilen süreç, elde edilecek olan ürün veya hizmetin kalite özelliklerinin hedef boyutları için çok az değişkenlik ortaya koyarak çalışabilmelidir. İKK yöntemleri, sürecin kararlılığını sağlayıp, değişkenliği azaltarak sürecin kalitesini geliştirmede fayda sağlayan yedi güçlü problem çözme aracını içermektedir. Bu araçlar şu şekildedir (Montgomery, 2013):

i. Histogram
ii. Çetele diyagramı
iii. Pareto diyagramı

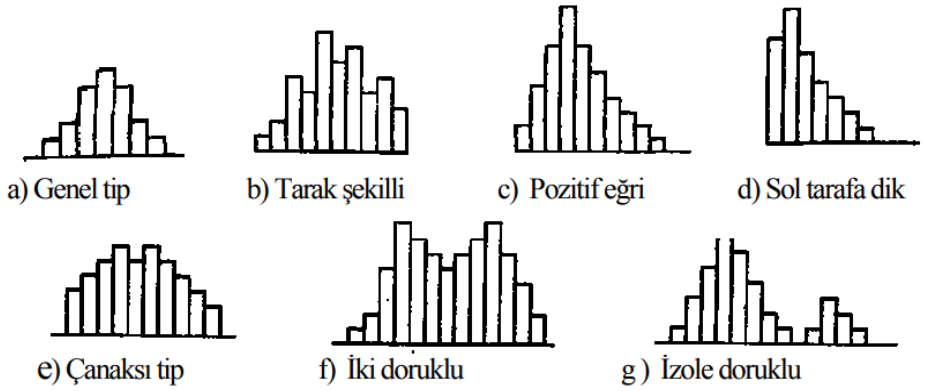
iv. Neden – sonuç diyagramı
v. Kusur konsantrasyon diyagramı (Tabakalama)
vi. Serpme diyagramı
vii. Kontrol kartları

Bu araçların tümü, genel süreci izlemek ve sürecin sürekli iyileştirilmesini sağlamak için yaygın olarak kullanılan İKK araçlarıdır. Bu araçlar, süreçte ortaya çıkabilecek sorunların, nedenlerinin tespit edilmesini ve bunların ortadan kaldırılmasını sağlayarak sürecin geliştirilebilmesi için kullanılmaktadır. Bu yedi araçtan alt başlıklarda detaylara girilmeden kısaca bahsedilecek olup kontrol kartlarından gelecek bölümde detaylı olarak bahsedilecektir.

2.4.1 Histogram

Histogramlar çubuk grafikler şeklinde tanımlanabilir. Uygun sınıf aralıklarında gruplandırılan ve büyüklüklerine göre düzenlenen gözlemler için dağılımı belirten grafiklerdir. Histogramlar dağılımlara dayalı süreçler ile ilgili sonuçları ortaya çıkarmakta oldukça faydalıdır (Magar ve Shinde, 2014). Histogramlar, ölçüm değerlerinin genel eğilimini, yakınsadıkları değeri ve dağılım sıklığını görmeyi kolaylaştırarak önemli bir kolaylık sağlamaktadır. Ek olarak histogramlar, veriler tarafından gösterilen varyasyonlar veya sapmaların kuşbakışı görünümünü sunmaktadır. (Dengizler Kayaalp, 2007).

Şekil 2.1’de histogram şekilleri isimleri ile birlikte verilmiştir.



Şekil 2.1 Başlıca histogram şekilleri (Dengizler Kayaalp, 2007)

2.4.2 Çetele Diyagramı

Verilerin toplanması herhangi bir analizin temel kısmını oluşturmaktadır. Süreçten toplanan verilerin ilgili ve kapsamlı bir şekilde yapılacak olan analiz için düzgün bir şekilde derlenmesi gerekir. Çetele diyagramları, veri toplanmasında ve derlenmesinde özellikle büyük bir öneme sahiptir. İlgilenilen süreç için toplanılacak olan verilerin türüne göre özel olarak oluşturulurlar ve sistematik bir şekilde bu verilerin derlenmesini sağlarlar (Magar ve Shinde, 2014). Bazı durumlarda çetele diyagramı, İKK analizinin başlangıcı olarak da sayılabilir (Akın, 2017).

2.4.3 Pareto Diyagramı

Pareto diyagramları, süreç içerisindeki elemanları sürece etkilerine göre büyüklük sırasıyla düzenleyen ve böylece, süreçte en yüksek etkiyi ortaya koyan elemanların tanımlanmasını sağlayan bir kalite kontrol aracıdır. Örnek olarak; hakkında en çok şikâyet gelen ürünlerin belirlenmesi, en sık ortaya çıkan şikayetlerin ortaya çıkış nedenlerinin belirlenmesi ve en sık geri iade alınan ürünlerin belirlenmesi gibi amaçlar için kullanılabilir. Ayrıca bu araç ile süreç içerisinde etki yaratan elemanlara göre iyileştirme uygulamaları da oluşturulabilir (Magar ve Shinde, 2014).

Tablo 2.1'de örnek bir pareto tablosu verilmiştir. Bu tablo da bir sipariş süreci sırasında kullanılan formların hata oranları verilmektedir. Bu hata oranlarına göre

formların sürece etkileri büyükten küçüğe doğru olacak şekilde tablo üzerinde sıralanmaktadır.

Tablo 2.1 Bir sipariş sürecindeki formlar için oluşturulmuş örnek bir pareto tablosu (Juran ve Godfrey, 1998)

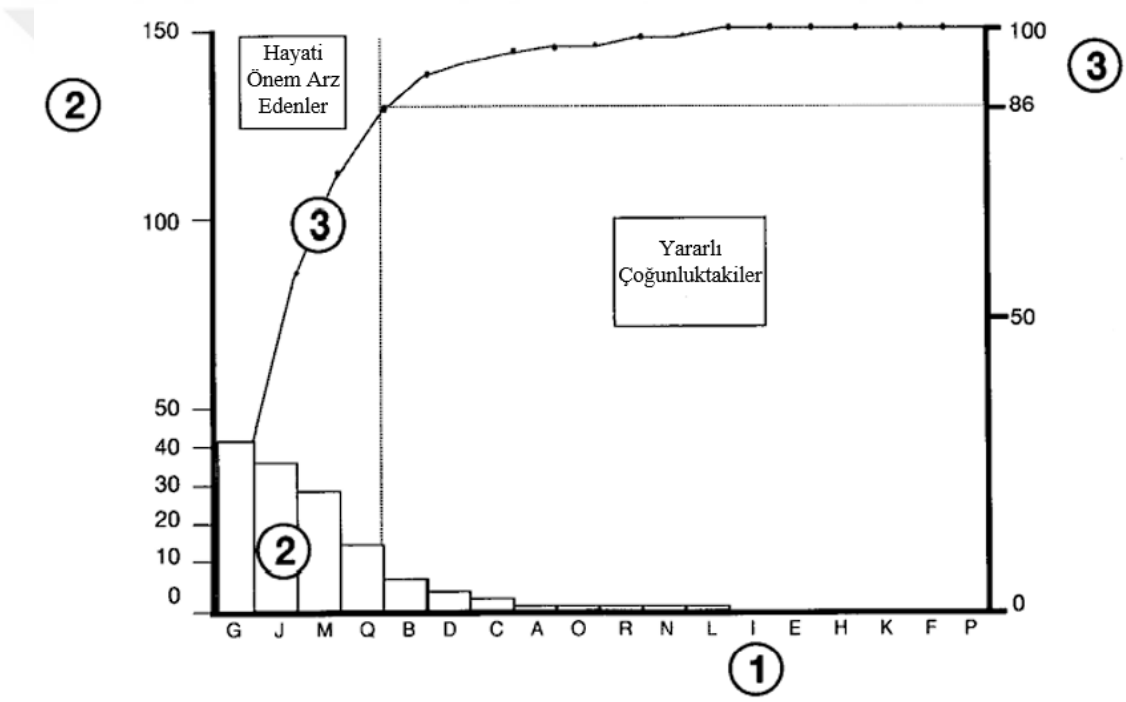
Sipariş Formları (1)	**Hata Sayısı (2)**	**Toplam Sipariş Formu Sayısına Gore Hata Yüzdesi**	**Toplam Sipariş Formu Sayısına Göre Kümülatif Hata Yüzdesi (3)**
G	44	29	0
J	38	25	29
M	31	21	54
Q	16	11	75
B	8	5	86
D	5	3	91
C	3	2	95
A	1	0.67	97
O	1	0.67	98
R	1	0.67	98
N	1	0.67	99
L	1	0.66	99
I	0	0	100
E	0	0	100
H	0	0	100
K	0	0	100
F	0	0	100
P	0	0	100
TOPLAM	**150**	**100**	**100**

Genel olarak pareto diyagramları çizdirilmeden önce Tablo 2.1'de bir örneği verilmiş olan pareto tablolarından faydalanılır. Bu tablolar pareto diyagramını çizdirirken kullanılan değerlerin üç temel adım ile hesaplanan değerlerini içermektedir. Bu temel adımlar şu şekildedir (Juran ve Godfrey, 1998):

- Elemanlar genel etkiye katkıda bulundukları düzeye göre sıralanır. Bu sıralama genellikle hata sayılarına göre yapılır.

- Her bir eleman için sayısal olarak elde edilen katkı yüzdesi miktarı hesaplanır. Bu miktar hesaplanırken hata miktarlarından ve toplam eleman sayısından faydalanılır. Örneğin, Tablo 2.1'de verilen tabloda G tipi sipariş formu için toplam eleman sayısına göre hata yüzdesi, $\frac{44}{150} = 0{,}29$ şeklinde hesaplanmıştır.
- Son olarak listelenen öğelerin hata yüzdelerinin kümülatif değerleri elde edilir.

Bu adımlarla elde edilen değerler Tablo 2.1'de sırasıyla 1,2 ve 3 ile numaralandırılmış sütunlarda görülmektedir. Aşağıda verilen Şekil 2.2'de, Tablo 2.1'den elde edilen değerlere göre çizdirilen pareto diyagramı örneği verilmiştir.



Şekil 2.2 Sipariş formları için oluşturulan pareto diyagramı örneği (Juran ve Godfrey, 1998)

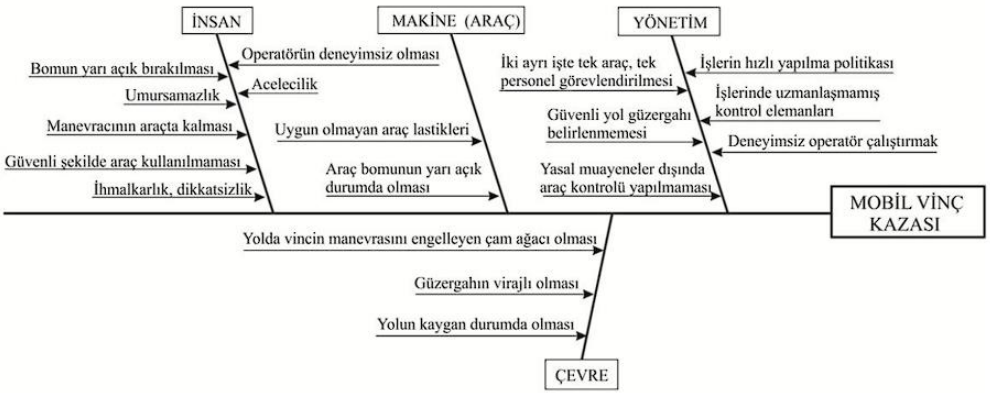
Tablo 2.1'de numaralandırılan sütunların pareto diyagramında bulunduğu kısımlar yine aynı numaralar ile Şekil 2.2'de de belirtilmiştir. Şekil 2.2'den de görüldüğü üzere pareto diyagramı sayesinde süreçte bulunan elemanlar hayati önem arz edenler ve yararlı çoğunluktakiler olacak şekilde iki gruba ayrılmıştır. Burada hayati önem arz edenler grubu, sürece etkisi büyük olan elemanları içermektedir. Sürece yapılan etki, sipariş formları örneği için hata sayısıyla doğru orantılıdır. Fakat süreçten sürece göre

bu durum değişiklik gösterebilir. Şekil 2.2'de verilen örnek pareto diyagramının hayati önem arz eden kısmında bulunan elemanlar için iyileştirilmeye gidilmesi gerekmektedir. Öte yandan yararlı çoğunluk grubunda bulunan elemanların, sürecin işleyişinde daha az sorun çıkartma ihtimaline sahip olmasından dolayı daha çok tercih edilebileceğini söyleyebiliriz.

2.4.4 Neden – Sonuç Diyagramı

Neden – sonuç diyagramları, süreç içerisinde, bir etki ile ortaya çıkabilecek olası bir sonucun ilişkisini sistematik bir şekilde gösteren araçlardır. Balık kılçığı diyagramı veya Kouro Ishikawa tarafından geliştirildiği için Ishiwaka diyagramı olarak da bilinmektedir. Bu araçta ilk olarak süreçte ortaya çıkabilecek nedenlerin kategorileri veya ana grupları belirlenir ve ardından omurga olarak da adlandırılan ana bağlantı bölümüne bağlanır. Oluşturulmuş olan ana grupların her biri altına olası nedenler ve ikincil nedenler eklenerek diyagram oluşturulur. Böylelikle süreç içerisinde bir sorun oluştuğunda hızlıca sorunun tespiti ve düzeltilmesi yapılabilir (Magar ve Shinde, 2014).

Şekil 2.3'te örnek bir süreç için elde edilen neden-sonuç diyagramı verilmiştir.



Şekil 2.3 Bir süreçte meydana gelebilecek mobil vinç kazası sorunu için oluşturulan neden-sonuç diyagramı (Atalay ve Kılıç, 2015)

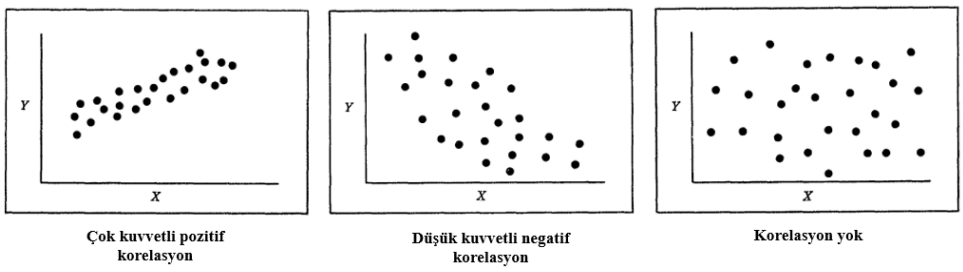
2.4.5 Tabakalama

Bir süreç içerisindeki bilgilerin, belirli özelliklere ve kategorilere göre sınıflandırmasına tabakalama işlemi denir. Bu işlem uygulanırken, belirlenen problem parçalara bölünür ve bu bölünen parçalar tek tek incelenir. İnceleme sırasında karışıklığın engellenmesi için farklı kaynaklardan elde edilen veriler ayrı ayrı değerlendirilir. Süreç içerisindeki farklı kaynaklar, süreçten sürece göre değişiklik gösterebilmektedir. Örnek olarak bir üretim süreci için vardiya, makine, operatör, malzeme vb. gibi kaynaklar süreç içerisindeki farklı kaynaklara örnek verilebilir (Yılmaz, 2009).

2.4.6 Serpme Diyagramı

Serpme diyagramı, ilgilenilen süreçten elde edilen herhangi iki nicel kalite karakteristiği arasında bir korelasyon olup olmadığının incelenmesini ve eğer korelasyon var ise yönünün ve şiddetinin belirlenmesini sağlayan bir kalite kontrol aracıdır. Diyagramı oluşturmak için belirlenen iki kalite karakteristiği süreç içerisinden rastgele belirlenebileceği gibi neden – sonuç diyagramından faydalanılarak da belirlenebilir. Özellikle neden – sonuç diyagramından faydalanılarak elde edilen kalite karakteristikleri sayesinde süreci kontrol dışına çıkaran olası nedenlerin tespit edilmesi sağlanır. Bundan dolayı neden-sonuç diyagramı ile serpme diyagramı birlikte kullanıldığında süreçler için güçlü problem çözücülerdir (Yılmaz, 2009).

Şekil 2.4'te bazı serpme diyagramı türleri verilmiştir.



Şekil 2.4 X ve Y kalite karakteristikleri için bazı örnek serpme diyagramı türleri (Western Electric Company, 1956)

2.4.7 Kontrol Kartları

İstatistiksel süreç kontrol kartlarının temeli 1920'lere dayanmaktadır. 1920'lerde Bell Telefon Laboratuvarlarından Walter A. Shewhart tarafından geliştirilen Shewhart kontrol kartı ile istatistiksel süreç kontrolünün modern bilimsel analizinin yolu açılmıştır. Shewhart kontrol kartı en temel kontrol kartıdır. İstatistiksel süreç kontrol kartlarının temelini oluşturan istatistiksel kavramları anlamak için önce Shewhart'ın önerdiği kontrol kartlarındaki temellerinin anlaşılması gerekmektedir (Montgomery, 2013).

Kontrol kartları sürecin kararlılığını incelerken aynı zamanda süreçten arzu edilen çıktıların elde edilmesinde kullanılan bir kalite kontrol aracıdır. Sürecin durumunun devamlı olarak görüntülenmesini sağlayan bu kontrol aracında; üst kontrol limiti, alt kontrol limiti ve merkez çizgi olmak üzere üç adet temel eleman bulunmaktadır (Akın, 2017).

Özellikle kontrol kartlarındaki limit çizgilerinin aşılması süreçte bir sorun olduğunu veya olabileceğini belirtmektedir. Eğer inceleme gerçekleştirilmez ise süreçten elde edilen çıktılarda sorunlar ortaya çıkabilir. Bu da ilgilenilen süreç için hem zaman hem de maliyet kaybı anlamına gelmektedir (Akın, 2017).

BÖLÜM ÜÇ
İSTATİSTİKSEL KALİTE KONTROL KARTLARI

3.1 Kontrol Kartları

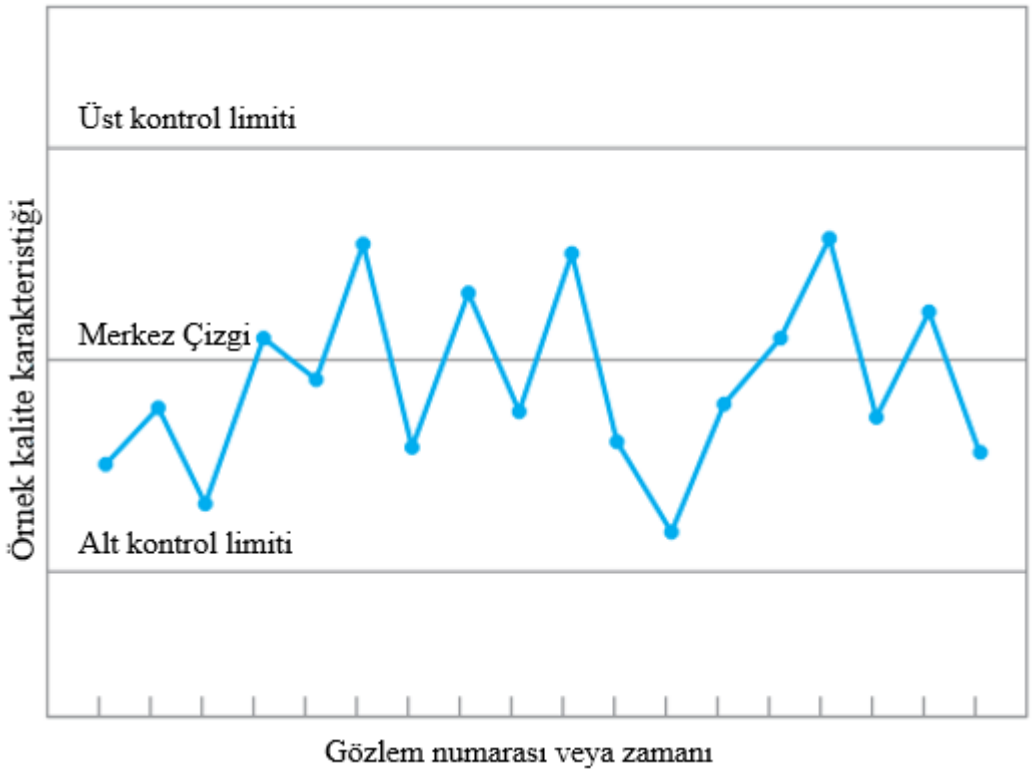
Kontrol kartları 1920'li yılların ortalarına doğru Walter Andrew Shewhart tarafından kullanımı önerilmiş olan bir İKK yöntemidir. Bir önceki bölümde de bahsedilen diğer 6 İKK yöntemine göre kontrol kartları literatürde en gelişmiş İKK yöntemi olarak belirtilmektedir. Kontrol kartlarının temel amacı, bir süreçteki kritik kalite karakteristiklerinin davranışları için görsel bir gösterim oluşturmak ve süreçte ortaya çıkmış veya çıkabilecek olan özel nedenlerin varlığını belirten kontrol dışı sinyalleri tespit etmektir.

Kontrol kartlarının popüler olmasında en az 5 neden vardır (Montgomery, 2013):

I. Kontrol kartlarının süreçteki üretkenliği arttırdığı kanıtlanmıştır. Başarılı bir kontrol kartı ile süreç takibi, süreçteki israfı ve yeniden çalışmayı azaltmaktadır. Bu da daha çok üretim ve daha az maliyet anlamına gelmektedir.
II. Hata önlemede kontrol kartları büyük önem taşımaktadır. Kontrol altında olan bir süreç için kontrol kartları sürecin kontrol altında tutulmasını sağlar.
III. Kontrol kartları süreç için gereksiz düzeltme yapılmasını önler. Bir kontrol kartı, özel faktörlerden kaynaklı varyasyonu ayırt edebilir ve bu ayrımı yaparken, süreçte görev alan personele göre daha verimli bir sonuç elde edilmesini sağlayabilir.
IV. Kontrol kartları süreçte oluşan bir sorun için teşhis bilgisi sağlar. Kontrol kartlarında oluşan örüntüler, deneyimli bir süreç çalışanı için teşhis açısından önemli bilgiler içermektedir. Elde edilen bu bilgiler sayesinde sürece, sürecin performansını arttırabilecek bir değişikliğin uygulanmasına olanak sağlar.
V. Kontrol kartları sürecin kabiliyeti hakkında bilgi sağlar. Kontrol kartları, süreç parametreleri hakkında zaman içindeki değişikliklerine ve

kararlılıklarına göre bilgi sağlar. Bu durum süreç için bir kabiliyet analizinin yapılmasına olanak sağlar.

3.2 Kontrol Kartlarının Yapısı



Şekil 3.1 Örnek bir kalite kontrol kartı (Montgomery, 2013)

Şekil 3.1'de verilen örnek kontrol kartındaki elemanları inceleyecek olursak (Montgomery, 2013):

- **Üst Kontrol Limiti (ÜKL):** Kontrol altındaki süreç için kalite karakteristiğinin alabileceği maksimum değerdir.
- **Merkez Çizgi (MÇ):** Kalite karakteristiğinin süreç devam ederken ki ortalama değeridir.
- **Alt Kontrol Limiti (AKL):** Kontrol altındaki süreç için kalite karakteristiğinin alabileceği minimum değerdir.

Eğer bir süreç kontrol kartında, AKL ve ÜKL arasında bir görünüm sunuyorsa, bu sürecin kontrol altında olduğunun bir göstergesidir. Bundan dolayı bir kontrol kartında bir veya daha fazla gözlem, AKL ve ÜKL'nin ötesine geçtiğinde ya da çizilen noktalar rastgele olmayan bir örüntü sergilediğinde, kontrol kartı süreç için kontrol dışı bir durumu belirtebilir.

3.3 Süreçte Varyasyona Neden Olan Faktörler

Herhangi bir süreç, her ne kadar iyi tasarlanmış veya özenle korunmuş olsa da sürecin doğası gereği belirli bir varyasyon ya sürecin başlangıcından itibaren ya da doğal olarak sürecin işleyişi sırasında ortaya çıkacak şekilde mevcuttur. Genel olarak süreçlerde kontrol edilebilir veya kontrol edilemez varyasyonlar bulunmaktadır. Süreçlerde varyasyonu oluşturan faktörler, İKK için iki şekilde tanımlanmaktadır (Montgomery, 2013; Qiu, 2014):

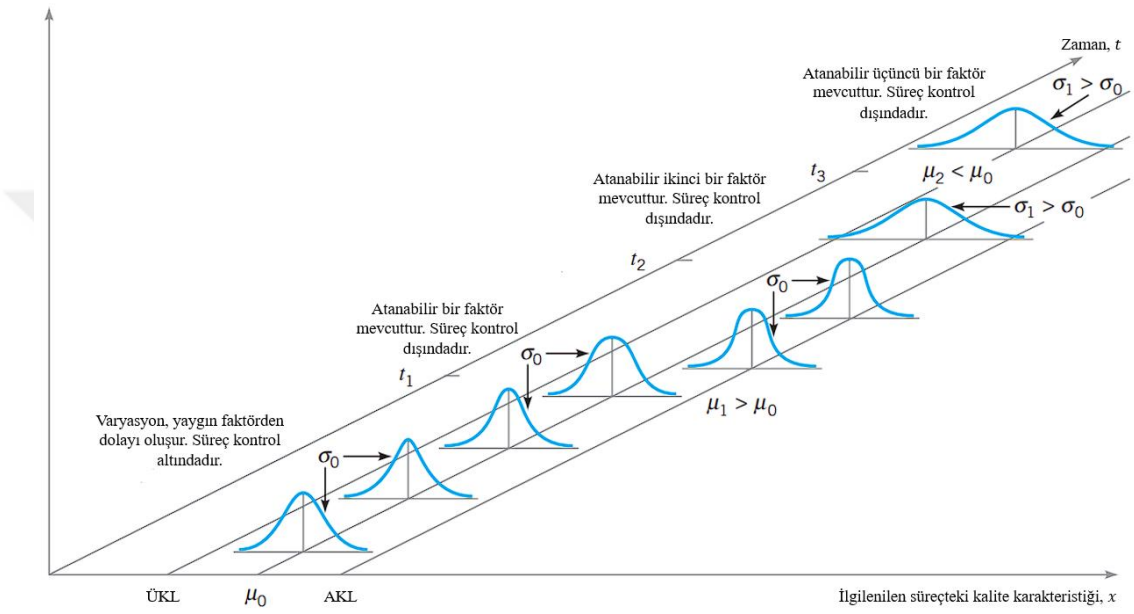
Yaygın Faktörler (Common Causes): Süreçlerde doğal olarak ortaya çıkan varyasyonu oluşturan faktörlerdir. Genellikle ortaya çıkan bu doğal varyasyon süreçteki yaygın faktörlerin kümülatif etkisi ile ortaya çıkmaktadır. Yaygın faktörler aracılığıyla oluşan varyasyon ile işleyen bir sürecin kontrol altında olduğu varsayılmaktadır. Diğer bir deyişle yaygın faktörler, oluşumundan itibaren süreçlerin bir parçasıdır.

Özel Faktörler (Assignable Causes): Süreçlerde yaygın faktörlerin oluşturduğu varyasyonlar dışında ortaya çıkan diğer varyasyonların oluşmasına neden olan faktörlerdir. Yaygın faktörler ile karşılaştırıldığında, özel faktörler aracılığıyla ortaya çıkan varyasyonlar genellikle süreçler için büyük bir sorunu ve süreç performans seviyesi için kabul edilemez bir durumu belirtmektedir. Bundan dolayı özel faktörler ile oluşmuş varyasyon içeren bir süreç, kontrol dışı bir süreç olarak tanımlanmaktadır.

Kontrol kartlarının amaçlarından biri de süreçlerdeki özel faktörler ile ortaya çıkan olağandışı varyasyonları tespit etmektir. Eğer bir kontrol kartı, ilgilenilen süreçte özel bir faktörün ortaya çıktığına işaret ediyorsa, mutlaka bu özel faktörün bulunarak

süreçten çıkarılması için çaba sarf edilmesi gerekmektedir. Bundan dolayı, özellikle endüstriyel süreçlerde özel faktörlerin sağlıklı bir şekilde algılanması, tespit edilmesi ve düzeltilmesi süreçler için büyük önem taşımaktadır (Reis ve Gins, 2017; Woodall ve Montgomery, 1999).

Şekil 3.2 yaygın faktörler ve özel faktörler için görsel bir anlatım sağlamaktadır.



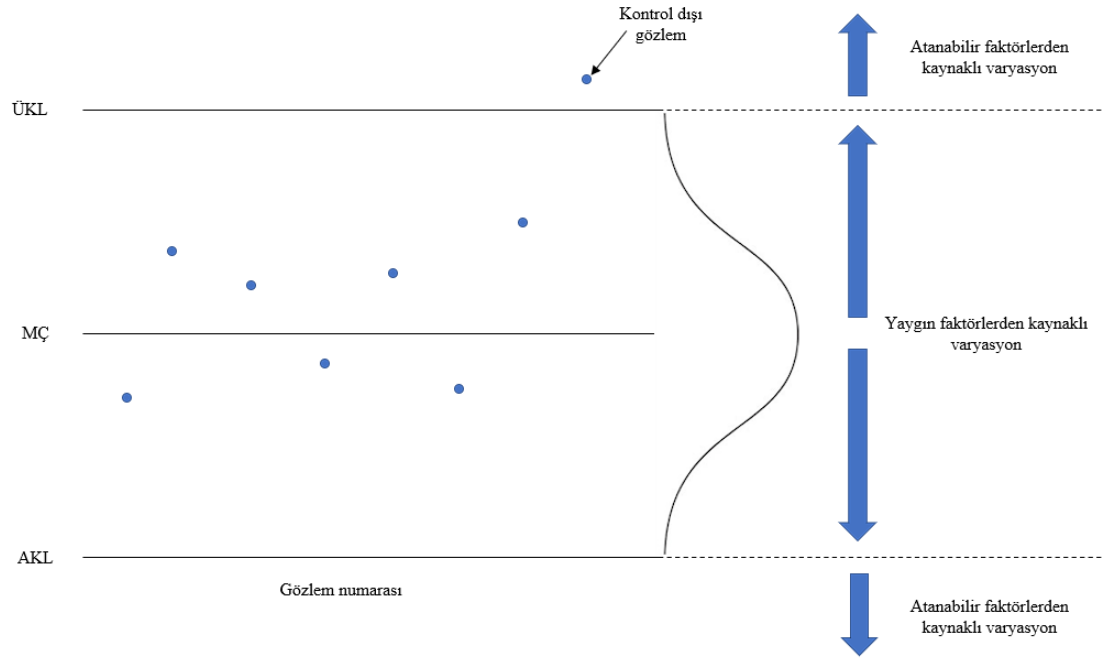
Şekil 3.2 Oluşan varyasyonun yaygın ve özel faktörleri (Montgomery, 2013)

Şekil 3.2 incelendiğinde sürecin t_1 zamanına kadar kontrol altında olduğu görülmektedir. Bundan dolayı t_1 zamanına kadar oluşan varyasyonların, yaygın faktörlere bağlı varyasyonlar olduğu söylenebilir. Ayrıca t_1 zamanına kadar sürecin standart sapması (σ_0) ve ortalamasının (μ_0) süreç için istenilen kontrol değerlerinde olduğu görülmektedir. t_1 zamanına gelindiğinde özel bir faktör ortaya çıkar ve Şekil 3.2'de görüldüğü gibi sürecin ortalaması ve standart sapması, yeni değerler olan μ_1 ve σ_1 değerlerini alır. t_2 zamanında ise yeni bir özel faktör ortaya çıkar ve sürecin ortalaması tekrar eski haline yani μ_0 değerine döner. Fakat artık sürecin standart sapması daha büyük bir değer olan σ_1 değerini almıştır. Burada önemli olan durum, t_1 zamanından itibaren oluşan bütün özel faktörlerin, sürecin kontrol dışı olarak nitelendirilmesini sağlamasıdır. Ayrıca Şekil 3.2'de görüldüğü üzere özel bir faktör

ortaya çıktığında sürecin büyük bir kısmı kontrol limitlerinin dışında kalmaktadır (Montgomery, 2013).

Endüstriyel uygulamalarda ve ilgilenilen kalite karakteristiğinin normal dağılım gösterdiği kontrol kartlarında, ÜKL ve AKL değerleri genellikle $\pm 3\sigma$ olacak şekilde belirlenir. Üç sigma yöntemi ile belirlenen ÜKL ve AKL değerleri kontrol kartlarında normal dağılım gösteren bir kalite karakteristiğinin %99,74'ünü kapsamaktadır (Yılmaz, 2019).

Şekil 3.3'te varyasyonun kontrol grafiğindeki daha detaylı bir görüntüsü sunulmaktadır.

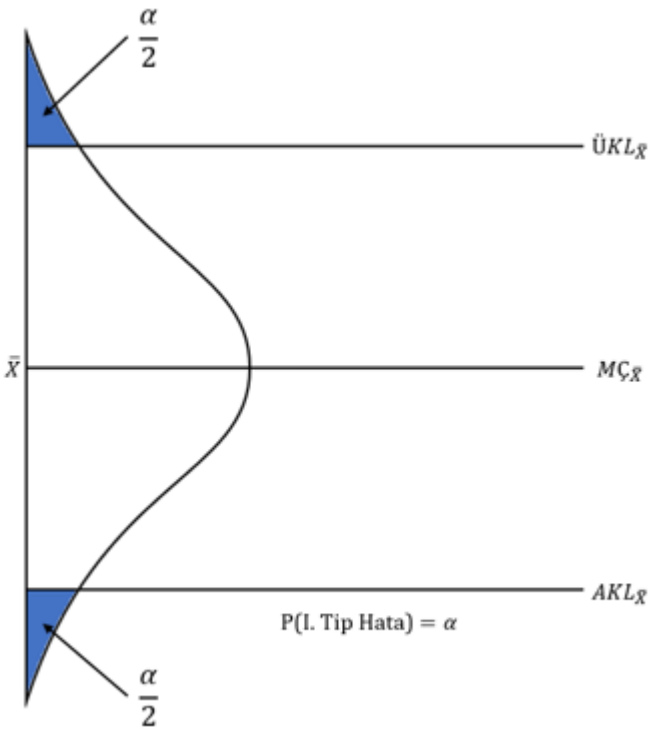


Şekil 3.3 Kontrol kartlarındaki varyasyonlar (Yılmaz, 2019)

3.4 Kontrol Kartlarında I. tip (α) ve II. Tip (β) Hataları

Kontrol kartları ile süreç hakkında karar verilirken genel olarak I. tip (α) ve II. tip (β) hata olmak üzere iki tip hata meydana gelmektedir.

Kontrol kartları aracılığıyla süreç hakkında karar verilirken eğer ilgilenilen süreç gerçekten kontrol altındayken kontrolden çıktığı sonucuna varılırsa, I. tip hata işlenmiş olur. Bir önceki bölümde de bahsedildiği gibi kontrol kartlarındaki AKL ve ÜKL değerleri genellikle ilgilenilen kalite karakteristiği örneklerinden elde edilen standart sapma (σ) değerinin ± 3 katı olacak şekilde belirlenir. Normal dağılım teorisine göre ilgilenilen kalite karakteristiğinin gözlem değerlerinin $\pm 3\sigma$ değerlerinin ötesine geçmesi için %0,26'lık (1-0,9974) bir olasılık bulunmaktadır. Bu durum gerçekleştiğinde kontrol kartlarını gözlemleyen kullanıcı tarafından verilecek olan karar, sürecin kontrolden çıktığı şeklinde olacaktır ve eğer süreç gerçekten kontrol altındaysa bu karar ortaya yanlış bir sonuç çıkaracaktır (Swamidass, 2000).

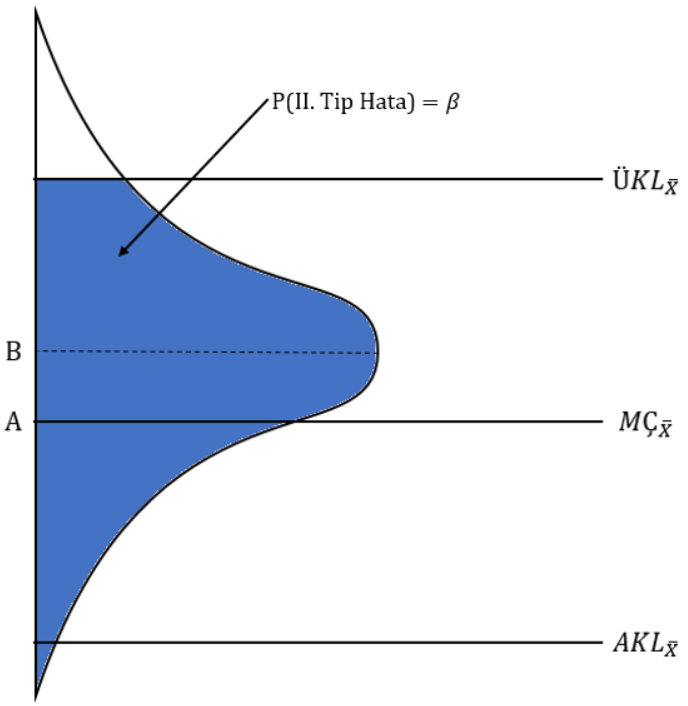


Şekil 3.4 Bir $\overline{X}$ kontrol kartında I. tip (α) hata (Swamidass, 2000)

Şekil 3.4 incelendiğinde I. tip hata olasılığı AKL ve ÜKL değerlerinin dışındaki iki kuyruk alanının toplamıdır. Kuyruk alanlarının her biri $\alpha/2$ ile belirtilirse, I. tip hata olasılığı α ile belirtilir. Aynı zamanda I. tip hatalar, ilgilenilen sürecin kalite kontrol operatörlerini, özel faktörlerle oluşan varyasyonlara yönlendiren yanlış alarmlar olarak

da bilinmektedir. Fakat süreçte herhangi bir özel faktör bulunmadığı için harcanan iş gücü ve zaman, I. tip hatanın maliyetidir (Swamidass, 2000).

Kalite kontrol kartlarında süreç hakkında karar verilirken eğer ilgilenilen süreç gerçekten kontrol dışındayken kontrol altında olduğu sonucuna varılırsa, II. tip (β) hata işlenmiş olur. Kalite kontrol kartları genellikle, ÜKL ve AKL değerlerinin dışında herhangi bir gözlem yoksa sürecin kontrol altında olduğu varsayılarak yorumlanır. Fakat, kontrolden çıkmış süreçlerdeki bazı özel faktörlerden kaynaklı olarak ilgilenilen kalite karakteristiği istatistiğinin, ÜKL ve AKL değerlerinin dışına çıkmaması durumu da mümkündür. Bu durumda süreç hakkında karar verilirken, sürecin kontrol altında olduğu ve ilgilenilen süreçte herhangi bir özel faktör bulunmadığı sonucuna ulaşılması kaçınılmazdır. Ancak gerçekte süreç, kontrol altında değildir. Bu da II. tip hata anlamına gelmektedir (Swamidass, 2000).



Şekil 3.5 Bir $\overline{X}$ kontrol kartında II. tip (β) hata (Swamidass, 2000)

Şekil 3.5, ortalaması A noktasından B noktasına doğru kayma gösteren bir sürecin kontrolden çıkması durumunu belirtmektedir. İlgilenilen kalite karakteristiğinin ortalamasının ($\overline{X}$) örneklem üzerindeki dağılımı normaldir fakat, B noktası etrafında birikim göstermeye başlamıştır. Kontrol kartında bulunan taralı alan, II. tip hatanın bölgesini belirtmektedir. II. tip hatanın değeri β ile ifade edilmektedir (Swamidass, 2000).

I. tip ve II. tip hatadan kaçınmak için kontrol kartları incelenirken sadece ilgilenilen kalite karakteristiği istatistiğinin, AKL ve ÜKL değerlerini aşmamasına göre karar verilmemelidir. Limitlere ek olarak kontrol kartlarındaki örüntülerde incelenmelidir. Böylelikle ilgilenilen süreç için daha detaylı bilgiler elde edilerek daha detaylı kararlar verilebilir.

3.5 Kontrol Kartlarında Aşama I ve Aşama II

Geleneksel kontrol kartları, esas olarak imalat endüstrisindeki üretim hatlarını izlemek, gözlemlemek ve süreçte ortaya çıkan herhangi bir özel faktörü belirlemek için kullanılmaktadır. Fakat son yıllarda, özellikle büyük veri alanındaki gelişmelerle birlikte kontrol kartları açısından birçok yeni yöntemde ortaya çıkmaktadır. Son yıllarda yapılan çalışmalar ile birlikte kontrol kartlarının sadece endüstriyel uygulamalarda değil birçok alanda da kullanılabilecek önemli bir istatistiksel araç olduğu bilinmektedir (Qiu, 2020).

Herhangi bir süreç ilk olarak çalışmaya başladığında, sürecin performansı ve kontrol kartındaki parametreler hakkında elimizde herhangi bir bilgi bulunmamaktadır. Bu gibi durumlarda süreçten genellikle az sayıda çıktı elde edilmesine izin verilir ve ardından elde edilen veriler analiz edilir. Eğer tasarlanan süreç için istenilen gereksinimler karşılanmıyorsa süreç tekrar ayarlanmalıdır. Bu kontrol ve ayarlama adımı ilgilenilen sürecin kalitesi istenilen gereksinimleri karşılayana kadar genellikle birkaç kez tekrarlanır. Bu aşama için kullanılan kontrol kartları, aşama I kontrol kartları olarak adlandırılır (Qiu, 2020).

Aşama I kontrol kartları ile ilgilenilen süreç için parametreler tahmin edildikten sonra sürecin çıktı elde etmesine izin verilir. Ardından elde edilen çıktılar periyodik olarak incelenir. İlgilenilen kalite karakteristiği için oluşturulan kontrol sınırları, aşama I kontrol kartına dayanmaktadır. Eğer bu kontrol sınırları aşılırsa ilgilenilen kalite karakteristiğinin kontrol altındaki durumdan farklı bir dağılıma sahip olduğuna dair bir uyarı alırız. Bu da sürecin kontrol dışında olması anlamına gelmektedir. Bu aşama için kullanılan kontrol kartları genellikle aşama II kontrol kartı olarak adlandırılır (Qiu, 2020).

İKK literatüründe aşama I ve aşama II kontrol kartları arasındaki ayrım çok da eskiye dayanmamaktadır. Aşama I kontrol kartları, ilgilenilen süreç için uygun kontrol limitlerinin türetilebileceği herhangi bir önceki bilgiye sahip değildir. Ayrıca aşama I kontrol kartları, süreç için daha çok keşif niteliği taşıyan araçlardır. Bundan dolayı aşama I kontrol kartları için parametreler ne kadar iyi tahmin edilirse aşama II kontrol kartları ile süreç izlenirken o kadar iyi performans elde edilmektedir (Vining, 2009).

3.6 Kontrol Kartlarının Çeşitleri

İKK literatüründe en çok bilinen ve kullanılan kontrol kartı çeşitleri; Shewhart, CUSUM (Kümülatif Toplam) ve EWMA (Üstel Ağırlıklı Hareketli Ortalama) kontrol kartlarıdır. Literatürdeki ilk kontrol kartı 1924 yılının mayıs ayında Walter Andrew Shewhart tarafından tanıtılmıştır. Ardından Walter Andrew Shewhart, 1931 yılında yayınladığı kitabı ile Shewhart kontrol kartlarını (SKK) derinlemesine tanıtarak kullanımının önünü açmıştır (Best ve Neuhauser, 2006). Günümüze kadar SKK'nın birçok farklı versiyonu önerilmiştir ve pratikte yaygın olarak kullanılmaktadır.

Fakat SKK, süreçteki ilgilenilen kalite karakteristiğinin son gözlem değerinin bilgilerini kullanmaktadır ve $1{,}5\sigma$ değerine yakın olan süreçlerdeki küçük kaymaları iyi bir şekilde belirleyememektedir. Bu durum özellikle aşama II için ilgilenilen süreçlerde büyük sorunlara neden olabilmektedir. Bundan dolayı küçük kaymalara sahip olan süreçler için CUSUM ve EWMA kontrol kartlarının kullanımı daha

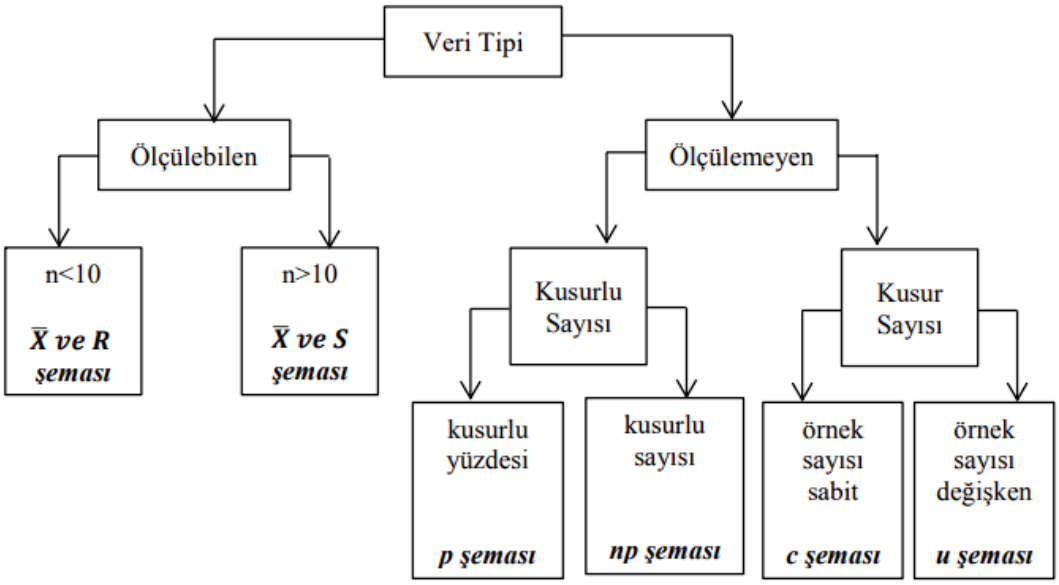
uygundur. CUSUM ve EWMA kontrol kartları, özellikle aşama II için SKK yerine kullanılabilecek güçlü alternatiflerdir (Montgomery, 2013).

Öte yandan CUSUM ve EWMA kontrol kartlarının geliştirilme amacı, küçük kaymalara sahip süreçleri izlemek olduğundan dolayı her iki kontrol kartı da doğal olmayan örüntüleri tanıma yeteneğinden yoksundur (Guh ve Tannock, 1999). Bu nedenle bu tez içerisinde yapılan çalışma için SKK temel alınmıştır ve dolayısıyla CUSUM ve EWMA kontrol kartları ile ilgili teorik bilgiden bahsedilmeyecektir.

3.7 Shewhart Kontrol Kartları

SKK, kalite karakteristiğinin özelliğine, ölçülen verinin türüne ve kullanım amacına bağlı olarak kendi içerisinde birçok farklı kontrol kartı türü barındırmaktadır (Özcan, 2020). Bu kontrol kartı türlerinden en yaygın kullanılanları Şekil 3.6'da verilmiştir.

Süreçte ilgilenilen kalite karakteristiği için SKK oluşturulmadan önce, kalite karakteristiğinin ölçülebilir (nicel) veya ölçülemez (nitel) olması durumuna bakılmalıdır. Belirleme yapıldıktan sonra kullanılacak olan SKK'nın türüne karar verilebilir. Eğer ilgilenilen kalite karakteristiğinin ölçülemez olması durumu söz konusu ise kusurlu parça sayısına veya kusur sayısına göre analiz yapılmasını sağlayacak olan kontrol kartı türlerinin, ilgilenilen kalite karakteristiği için uygulanması gerçekleştirilir (Temelatma, 2021).



Şekil 3.6 Ölçülebilen ve ölçülemeyen kalite karakteristikleri için SKK türleri (Koltan Yılmaz, 2014)

Şekil 3.6'da verilen SKK türlerinden bağımsız olarak temel bir kalite kontrol kartı; bir ÜKL, bir MÇ ve bir AKL'den oluşmaktadır. Öte yandan SKK için süreç açısından önemli bir konu olan kontrol limitleri belirlenirken genellikle MÇ değerinden $\pm 3\sigma$ uzaklıkta bulunan değerler belirlenir. Bunun nedeni 3σ limitlerinin tip I ve tip II hatalar için iyi bir denge noktası olmasından kaynaklanmaktadır. MÇ değeri ise, kontrol kartının türüne bağlı olarak incelenen kalite karakteristiğinin genellikle ortalamasına eşittir (Özcan, 2020):

$$\text{Ü}KL = M\text{Ç} + 3\sigma \tag{3.1}$$

$$AKL = M\text{Ç} - 3\sigma \tag{3.2}$$

3.8 Ölçülebilen Kalite Karakteristikleri için Shewhart Kontrol Kartları

Ölçülebilir kalite karakteristikleri için kullanılan kontrol kartları, ilgilenilen süreçten elde edilen ürünlerin özellikleri ölçülebilir (nicel) olduğunda kullanılır. Örnek olarak; genişlik, sıcaklık, uzunluk, derinlik ve hacim gibi rakamlarla ifade edilebilen özellikler verilebilir. İlgilenilen bir süreç için ölçülebilir kalite karakteristiği mevcut

ise, bu kalite karakteristiğinin hem ortalaması hem de standart sapması kontrol edilmelidir (Kaymakamtorunları Deniz, 2019). Bu tip kalite karakteristikleri için kullanılan kontrol kartları, birden fazla örnek alt sınıfı eleman sayısına sahip olan ve anlamlı örnek sınıfları oluşturabilen süreçler için kullanılır. Bunlar; ortalama ($\overline{X}$), standart sapma (σ) ve dağılım aralığı (R) kontrol kartlarıdır. Ayrıca bu kontrol kartları, pratikte en sık kullanılan kontrol kartlarıdır (Ertuğrul ve Karakaşoğlu, 2006).

3.8.1 $\overline{X} - R$ (Aritmetik Ortalama ve Değişim Aralığı) ve $\overline{X} - S$ (Aritmetik Ortalama ve Standart Sapma) Shewhart Kontrol Kartları

$\overline{X}$ kontrol kartı sadece aritmetik ortalama değerini dikkate alan bir kontrol kartı tipidir. Ölçülebilen kalite karakteristikleri için kullanılan bu kontrol kartlarında, $\overline{X}$ kontrol kartı ile R ya da S kontrol kartları genellikle birlikte kullanılır. Bunun nedeni ilgilenilen süreçten elde edilen örneklem ortalamaları aynı olmasına rağmen değişim aralıkları veya standart sapmaları büyük olursa sürecin kontrol dışına çıkabileceği görüşüdür. Çünkü kontrol altındaki bir süreçte, örneklem değerleri için hem uygun ortalama değerine sahip olunması hem de ölçümlerin uygun bir aralıkta bulunması beklenmektedir. Bundan dolayı $\overline{X}$, R ve S kontrol kartları $\overline{X} - R$ ya da $\overline{X} - S$ olacak şekilde birbirleri ile birlikte kullanılır (Kurt, 2018).

Süreç için kontrol kartları kullanılırken, tüm çıktıların tamamen incelenmesi yerine örnekleme yolu ile kontrol işlemi gerçekleştirilir. Öte yandan $\overline{X} - R$ ve $\overline{X} - S$ kontrol kartlarının hangisinin kullanılacağı belirlenirken de örneklem hacmine bakılır. Örneklem boyutu 1 ile 10 arasında ise $\overline{X} - R$ kontrol kartı kullanılır. Örneklem boyutu 10'dan büyük olduğunda ise değişim aralığı açısından güvenirliğin azalmasından dolayı $\overline{X} - S$ kontrol kartı kullanılır (Demir ve Mirtagioğlu, 2016).

Bir diğer önemli durum ise bu tip kontrol kartlarında standartların bilinmesi veya bilinmemesi durumudur. Eğer süreç ortalaması (μ) ve süreç standart sapması (σ) bilinmiyorsa bu durumda bu değerlerin tahmin edilmesi gerekir. μ için en iyi tahmin değeri örneklem ortalamalarının ortalaması olan $\overline{\overline{X}}$ değeridir. σ için ise en iyi tahmin

değeri örneklem standart sapmalarının ortalaması olan $\overline{S}$ değeridir. Ayrıca σ değeri ile değişim aralığı arasında ilişki bulunmaktadır ve bundan dolayı R kontrol kartı için σ değeri bilinmediğinde, örneklemlerin değişim aralıklarının ortalaması olan $\overline{R}$ değeri kullanılır. Öte yandan standartların bilindiği ve bilinmediği her iki durum için de MÇ, ÜKL ve AKL değerleri belirlenirken EK 1, EK 2 ve EK 3'te verilen ve örneklem boyutuna göre değişiklik gösteren sabitlerden faydalanılır (Demir, 2008; Juran ve Godfrey, 1998).

Kontrol kartları için MÇ ve limit değerleri bulunurken kullanılan parametreler şu şekildedir (Özcan, 2020):

- μ : Süreç ortalaması
- σ : Süreç standart sapması
- $\overline{\overline{X}}$: Örneklem ortalamalarının ortalaması
- $\overline{R}$: Örneklem değişim aralıklarının ortalaması
- $\overline{S}$: Tahmini standart sapma

Standartların belli olmaması durumunda $\overline{X}$, R ve S kontrol kartları için MÇ, ÜKL ve AKL değerleri verilen eşitliklere göre hesaplanır (Demir, 2008; Juran ve Godfrey, 1998; Montgomery, 2013):

$\overline{X}$ kontrol kartı için ($\overline{R}$ yerine $\overline{S}$ da kullanılabilir):

$$\text{ÜKL} = \overline{X} + A_2\overline{R} \tag{3.3}$$

$$\text{MÇ} = \overline{\overline{X}} \tag{3.4}$$

$$\text{AKL} = \overline{X} - A_2\overline{R} \tag{3.5}$$

R kontrol kartı için:

$$\text{ÜKL} = D_4\overline{R} \tag{3.6}$$

$$\text{MÇ} = \overline{R} \tag{3.7}$$

$$\text{AKL} = D_3\overline{R} \tag{3.8}$$

S kontrol kartı için:

$$\text{ÜKL} = B_4\overline{S} \tag{3.9}$$

$$\text{MÇ} = \overline{S} \tag{3.10}$$

$$\text{AKL} = B_3\overline{S} \tag{3.11}$$

Standartların belli olması durumunda ise $\overline{X}$, R ve S kontrol kartları için MÇ, ÜKL ve AKL değerleri verilen eşitliklere göre hesaplanır (Demir, 2008; Juran ve Godfrey, 1998; Montgomery, 2013):

$\overline{X}$ kontrol kartı için:

$$\text{ÜKL} = \mu + A\sigma \tag{3.10}$$

$$\text{MÇ} = \mu \tag{3.11}$$

$$\text{AKL} = \mu - A\sigma \tag{3.12}$$

R kontrol kartı için:

$$\text{ÜKL} = D_2\sigma \tag{3.12}$$

$$\text{MÇ} = d_2\sigma \tag{3.13}$$

$$\text{AKL} = D_1\sigma \tag{3.14}$$

S kontrol kartı için:

$$\text{ÜKL} = B_2\sigma \tag{3.15}$$

$$\text{MÇ} = c_4\sigma \tag{3.16}$$

$$\text{AKL} = B_1\sigma \tag{3.17}$$

Şeklinde MÇ, ÜKL ve AKL değerleri, standartların belli olduğu ve belli olmadığı durumlar için hesaplanır.

3.9 Ölçülemeyen Kalite Karakteristikleri için Shewhart Kontrol Kartları

Ölçülemeyen kalite karakteristikleri için kullanılan kontrol kartları, ilgilenilen süreçteki özellikler ölçüme uygun olmadığında kullanılır. Süreçteki kalite karakteristiği herhangi bir ölçü birimi ile ölçülemediği için kusurlu – kusursuz, uygun – uygunsuz, onaylandı – onaylanmadı, iyi – kötü gibi ikili kategorilere ayrılabilir. Özellikle kusurlu – kusursuz kategori örneği bu tip kontrol kartlarında kullanılan temel bir kavramdır. Yaygın olarak kullanılan bu tip kontrol kartları, Şekil 3.6'da da verildiği gibi dört gruba ayrılabilir (Ertuğrul ve Karakaşoğlu, 2006; Kaymakamtorunları Deniz, 2019):

- Kusurlu oranı (p) kontrol kartı
- Kusurlu sayısı (np) kontrol kartı
- Kusur sayısı (c) kontrol kartı
- Birim başına kusur sayısı (u) kontrol kartı.

İlgilenilen örneklemin içerisinde herhangi bir kusurlu örnek aranıyorsa p veya np kontrol kartlarından faydalanılır. Eğer ilgilenilen örneklemin içerisinden elde edilen örneklerde özel bir kusur aranıyorsa c veya u kontrol kartlarından faydalanılır (Temelatma, 2021).

3.9.1 p Kontrol Kartı

p kontrol kartı, bir örneklemde bulunan kusurlu oranı (uygunsuz birim) için kullanılan ve binom dağılımına dayalı olan bir kontrol kartıdır (Özcan, 2020). Genel

olarak p kontrol kartı, örneklem boyutu büyük olduğunda ve örneklem başına kusurlu sayısının dört veya dörtten büyük olduğu durumlarda çok yararlıdır. Fakat örneklem boyutu 25'ten küçük ya da tahmin edilen kusurlu sayısı 1'den az ise p kontrol kartı, ilgilenilen süreç için kontrolün durumu açısından güvenilir bilgi sağlayamayabilir (ASTM Committee E-11 on Quality and Statistics, 1990).

p kontrol kartı oluşturulmak istendiğinde ilk olarak örneklemlere ait kusurlu oranlarının bulunması gerekmektedir. Ardından bu oranların ortalaması alınarak ortalama kusurlu oranı olan $\overline{p}$ değeri elde edilir. Ardından $\overline{p}$ değerini içeren aşağıdaki eşitliklerle MÇ, ÜKL ve AKL değerleri elde edilir (Aslam vd., 2021; Özcan, 2020):

$$\text{ÜKL} = \overline{p} + 3\sqrt{\frac{\overline{p}(1-\overline{p})}{n}} \tag{3.18}$$

$$\text{MÇ} = \overline{p} \tag{3.19}$$

$$\text{AKL} = \overline{p} - 3\sqrt{\frac{\overline{p}(1-\overline{p})}{n}} \tag{3.20}$$

3.9.2 np Kontrol Kartı

np kontrol kartı, p kontrol kartına çok benzer olup kusur oranı yerine kusurlu sayısına göre eşitlikleri oluşturur. Bu kontrol kartı ilgilenilen süreçten elde edilen örneklemlerin boyutlarının eşit olduğu durumlarda kullanılmaktadır. Bu kontrol kartında ortalama kusurlu sayısı olan $n\overline{p}$ değeri, kontrol kartı parametreleri belirlenirken kullanılır. Öte yandan p kontrol kartındaki eşitliklerde kullanılan ve ortalama kusurlu oranı olarak ifade edilen $\overline{p}$ değeri, bu kontrol kartı için de kullanılmaktadır (ASTM Committee E-11 on Quality and Statistics, 1990; Özcan, 2020). np kontrol kartı için MÇ, ÜKL ve AKL değerleri aşağıdaki eşitliklerle elde edilir (Aslam vd., 2021):

$$\text{ÜKL} = n\overline{p} + 3\sqrt{n\overline{p}(1-\overline{p})} \tag{3.21}$$

$$\text{MÇ} = n\overline{p} \tag{3.22}$$

$$\text{AKL} = n\overline{p} - 3\sqrt{n\overline{p}(1-\overline{p})} \tag{3.23}$$

3.9.3 c Kontrol Kartı

c kontrol kartı, ilgilenilen süreçten elde edilen örneklemlerden kusur oranı belirlenemediğinde kullanılır. Kusur oranı belirlenememesinin nedeni süreçten elde edilen çıktıların, eşit bir alandan veya zamandan oluşmasından kaynaklanır. Örnek olarak; müşterilerden gelen haftalık şikâyet sayısı, bir metrekare halıdan elde edilen kusur sayısı verilebilir. Bundan dolayı c kontrol kartı, örneklem boyutları aynı olduğunda kullanılmaktadır. Öte yandan kusur oranı elde edilemediği için yerine örneklemlerden elde edilen kusur sayısı kullanılır (ASTM Committee E-11 on Quality and Statistics, 1990; Özcan, 2020).

c kontrol kartı oluşturulmak istendiğinde ilk olarak ortalama kusur sayısı değeri olan $\overline{c}$ değeri elde edilir. Bu değer örneklemlerden elde edilen kusur sayılarının ortalaması alınarak hesaplanır. Aşağıdaki eşitliklerle c kontrol kartı için MÇ, AKL ve ÜKL değerleri elde edilir (Aslam vd., 2021):

$$\text{ÜKL} = \overline{c} + 3\sqrt{\overline{c}} \tag{3.24}$$

$$\text{MÇ} = \overline{c} \tag{3.25}$$

$$\text{AKL} = \overline{c} - 3\sqrt{\overline{c}} \tag{3.26}$$

c kontrol kartı, u kontrol kartına eş değerdir. Fakat süreçten elde edilen örneklemlerin boyutları aynı olduğunda pratiklik açısından u kontrol kartının yerini almaktadır (ASTM Committee E-11 on Quality and Statistics, 1990).

3.9.4 u Kontrol Kartı

Temeli Poisson dağılımına dayanan u kontrol kartı, birim başına kusur adetini kontrol etmek için kullanılır. Örneklem boyutu u kontrol kartında, c kontrol kartından farklı olarak sabit veya değişken olabilir. Öte yandan örneklem boyutu 1 olduğu durumlarda, u kontrol kartı ile c kontrol kartı aynıdır (ASTM Committee E-11 on Quality and Statistics, 1990; Özcan, 2020).

u kontrol kartı oluşturulurken ilk olarak birim başına ortalama kusur sayısı olan $\overline{u}$ değeri hesaplanmalıdır. Bu değer hesaplanırken ilk olarak her örneklem için elde edilen kusur sayıları, ilgili örneklem boyutlarına bölünerek her örneklem için birim başına kusur sayısı değerleri hesaplanır. Ardından her örneklem için hesaplanan birim başına kusur sayısı değerleri toplanarak örneklem sayısına bölünür ve böylelikle birim başına ortalama kusur sayısı olan $\overline{u}$ değeri elde edilir. Ardından aşağıdaki eşitliklerle u kontrol kartı için MÇ, AKL ve ÜKL değerleri hesaplanır (Kurt, 2018):

$$\text{ÜKL} = \overline{u} + 3\sqrt{\frac{\overline{u}}{n}} \tag{3.27}$$

$$\text{MÇ} = \overline{u} \tag{3.28}$$

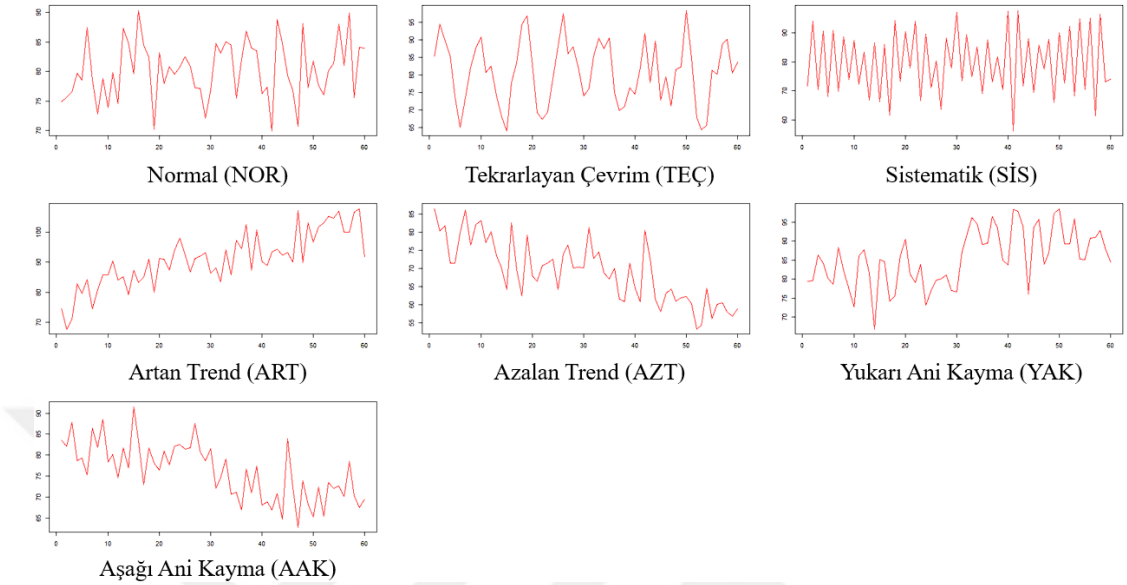
$$\text{AKL} = \overline{u} - 3\sqrt{\frac{\overline{u}}{n}} \tag{3.39}$$

3.10 Shewhart Kontrol Kartlarında Örüntüler

SKK'da sürecin kontrol altında olup olmadığı incelenirken sadece AKL ve ÜKL değerlerinin aşılmasına bakılmaz. Ayrıca kontrol kartındaki örüntülerde incelenir. Bir dizi nokta eğer doğal olmayan bir örüntü sergiliyorsa, bu sürecin kontrol dışında olduğunun veya olabileceğinin göstergesidir (Awadalla ve Sadek, 2012).

SKK'da ilgilenilen süreç için izlenme gerçekleştirilirken belirli örüntüler elde edilir. Literatür incelendiğinde elde edilen temel örüntü sayısı 6 ila 9 arasında

değişiklik göstermektedir. Şekil 3.7'de, SKK'da ortaya çıkabilecek olan yedi temel örüntü verilmiştir.



Şekil 3.7 Temel Shewhart kontrol kartı örüntüleri

Bu örüntüler, ilgilenilen sürecin kalite karakteristiğine göre sürecin uzun vadeli durumunu gösterebilir. Özellikle normal örüntünün elde edilmesi süreç için büyük önem taşımaktadır. Çünkü normal örüntü, sürecin kontrol altında olduğunun bir göstergesidir. Öte yandan normal örüntü dışında elde edilen diğer örüntüler anormal örüntüler olarak adlandırılır. Anormal örüntüler, sürecin ya kontrol dışında olduğunu ya da kontrol dışına çıkabileceğini belirtmektedir. Bu nedenle süreçte uygulanabilecek önleyici kontrol eylemleri hakkında bilgi edinmek açısından, SKK örüntülerini incelemek süreç için kritik öneme sahiptir (Gauri ve Chakraborty, 2008; Pham ve Oztemel, 1993).

BÖLÜM DÖRT
YAPAY SİNİR AĞLARI

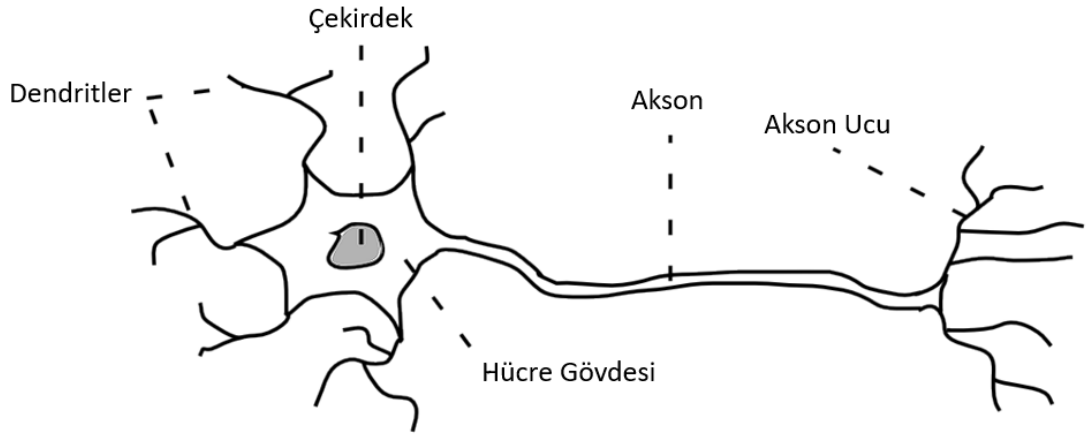
4.1 Yapay Sinir Ağlarına Giriş

YSA, biyolojik sinir hücreleri olan nöronların makineler yardımıyla simüle edilmesiyle birlikte yapay öğrenmenin gerçekleştirilmesini sağlayan bir bilgisayar bilimi alanıdır. Bu alanın ortaya çıkmasında, özellikle bilgisayar bilimleri alanındaki gelişmeler ile birlikte geleneksel programlama yöntemlerinin yetersiz kalması durumu etkili olmuştur. Bu durumdan yola çıkılarak insan beyninin yapısı incelenmeye başlanmıştır. İnsan beynindeki biyolojik sinir hücreleri olan nöronlar ile beynin en temel özelliklerinden biri olan öğrenme özelliği, birlikte birçok kompleks algoritmayı çözebilmemizi sağlamaktadır. İnsan beyninde öğrenme özelliği en temel şekilde, tecrübe kazanarak gerçekleşmektedir. Buradan yola çıkarak araştırmacılar beyinde gerçekleşebilen bu ileri seviye öğrenme özelliğini bilgisayar bilimleri alanına aktarabilmek için beynin matematiksel modellemesini oluşturmuşlardır. Burada amaç, bilgisayarlarında insanlar gibi tecrübe kazanarak zaman içerisinde öğrenmelerini sağlamak ve karşılaşılan problemlerde yardımcı olmalarını sağlamaktır. Böylelikle, makine öğrenmesi alanının en önemli alt dallarından biri olan YSA alanı, 1958 yılında Rosenblatt tarafından, ilk yapay nöronun önerilmesi ile ortaya çıkmıştır. Ardından bilgisayar teknolojisindeki gelişmeler ile birlikte YSA, günümüze kadar birçok problemin çözümünde kullanılmış ve aktif olarak da kullanılmaya devam etmektedir (Graupe, 2007; Öztemel, 2003).

4.2 Biyolojik Sinir Hücresi

Beyin, merkezi sinir sistemindeki en önemli parça olup, işleyişi hücreler arasındaki iletişim yoluyla gerçekleşmektedir. Bir insan beyninde, sinir (nöron) ve glial (glue) şeklinde iki çeşit hücre bulunmaktadır. Bu iki çeşit hücreden, beyinde toplamda yaklaşık olarak 1 trilyon adet bulunmaktadır. Bu hücrelerden de toplamda yaklaşık 100 milyar tanesi sinir hücresi olan nöronlardır. Nöronlar beyin fonksiyonları ve sinir sistemi için ana öğelerdir (Ersoy ve Karal, 2012).

Şekil 4.1’de bir nöronun yapısı verilmiştir.



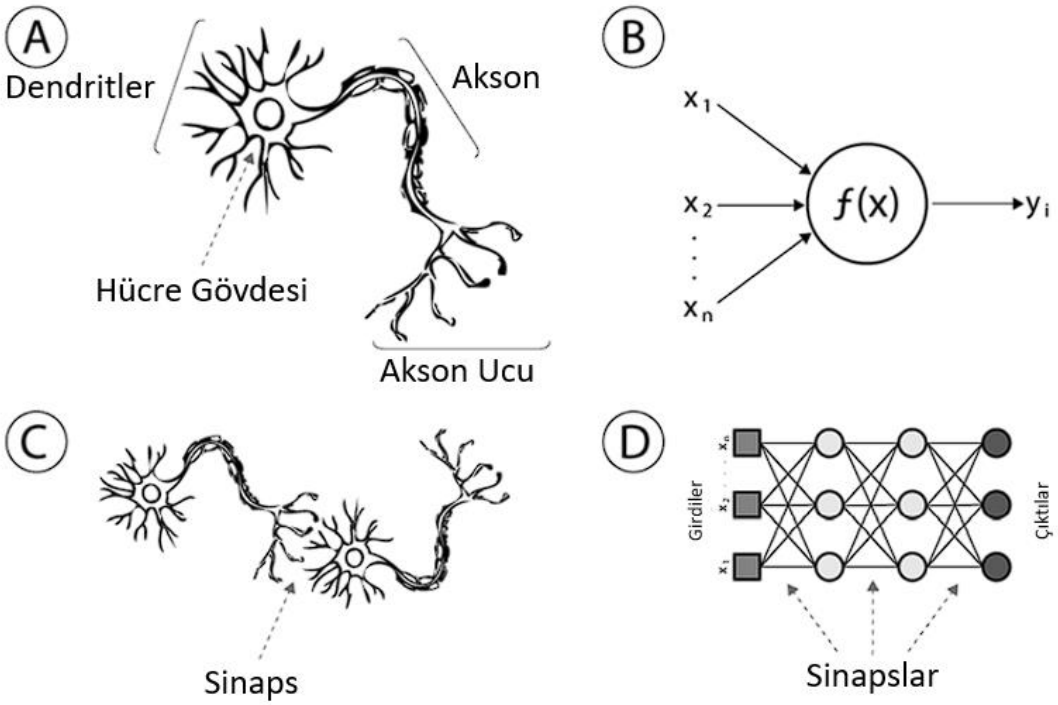
Şekil 4.1 İnsan beyninde bulunan bir nöronun yapısı (Corson ve Aziz-Alaoui, 2009)

Şekil 4.1’de görüldüğü üzere insan beynindeki bir nöron, dendrit, çekirdek, akson ve akson ucu bölgelerinden oluşmaktadır. Nöronlar arasındaki iletişim elektrik sinyalleri ile gerçekleşir. Ayrıca, nöronlar arasındaki iletişim akson ucunda bulunan sinapslar yardımıyla sağlanmaktadır. Dendritler, sinapslardan gelen elektrik sinyallerinin nörona girişini sağlar. Bu sinyaller çekirdekte toplanır ve üretilecek çıktı belirlenir. Ardından üretilecek olan çıktı akson bölgesinden akson ucunda bulunan sinapslara iletilir ve sonrasında sinapslardan elde edilen bu çıktı bir sonraki nörona iletilir. Böylelikle nöronlar arasındaki iletişim gerçekleşir (Ali, 2022; Ersoy ve Karal, 2012).

4.3 Yapay Sinir Hücresi

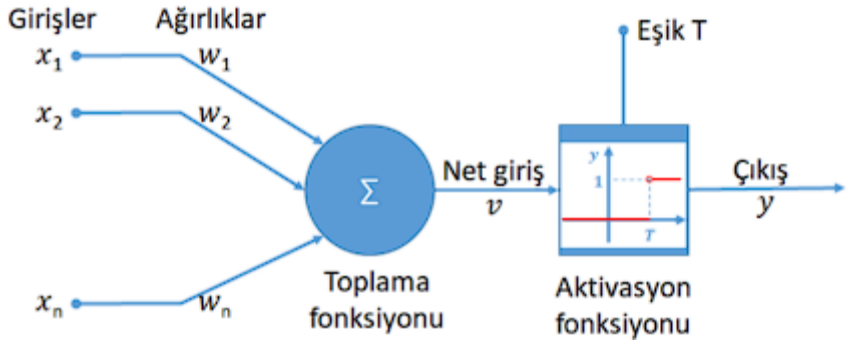
Yapay sinir hücreleri, biyolojik sinir hücrelerinin matematiksel olarak modellenmesi ile oluşturulmuş yapılardır. Biyolojik sinir hücreleri, sinir ağlarının en temel elemanı iken yapay sinir hücreleri ise YSA’nın en temel elemanıdır.

Aşağıda verilen şekilde biyolojik ve yapay sinir hücreleri arasındaki farklar görüntülenebilir.



Şekil 4.2 Biyolojik sinir hücresi; (B) Yapay sinir hücresi; (C) Biyolojik sinaps; (D) YSA'daki sinapslar (Goncalves vd., 2013)

Yapay sinir hücreleri aynı zamanda proses elemanları olarak da adlandırılmaktadır. Şekil 4.2'de verilmiş olan yapay sinir hücresi, biyolojik sinir hücresinde olduğu gibi belirli elemanlardan oluşmaktadır. Bir yapay sinir hücresinde 5 adet eleman bulunmaktadır. Bu elemanlar; girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktılardır (Öztemel, 2003).



Şekil 4.3 Bir yapay sinir hücresinde bulunan elemanların detaylı gösterimi (Şenol, 2022)

YSA elemanlarının, biyolojik sinir sistemindeki karşılıkları Tablo 4.1'de verilmiştir.

Tablo 4.1 YSA elemanlarının biyolojik sinir sistemindeki karşılıkları (Demir, 2018)

Yapay Sinir Ağı Elemanları	**Biyolojik Sinir Sistemi Elemanları**
İşlemci Eleman	Nöron
Toplama Fonksiyonu	Dendrit
Aktivasyon Fonksiyonu	Hücre Gövdesi
Yapay Sinir Ağının Çıktısı	Akson
Ağırlıklar	Sinapslar

Alt başlıklarda yapay sinir hücrelerini oluşturan elemanlar hakkında detaylı bilgi verilecektir. Ayrıca alt başlıklarda genel olarak yapay sinir hücreleri ile öğrenme işlemin nasıl gerçekleştiği hakkında da bilgi verilecektir.

4.3.1 Girdiler

Girdiler, yapay sinir hücresine dışarıdan aktarılan bilgilerdir. YSA'ya gelen bilgiler, öğretilmesi istenen örneğe göre değişiklik gösterir. Bir yapay sinir hücresine direkt olarak dış dünyadan bilgi gelebildiği gibi başka bir yapay sinir hücresinden de bilgi gelebilir (Öztemel, 2003). Öte yandan yapay sinir hücresindeki girdiler, biyolojik sinir hücresinde dendritlere gelen bilgilere karşılık gelmektedir.

4.3.2 Ağırlıklar

Ağırlıklar, yapay sinir hücresine gelen bilginin hücre üzerindeki etkisini ve önemini belirtmektedir. Örneğin Şekil 4.3'te w_2 değeri, x_2 girdisinin ağırlık değerini belirtmektedir. w_2 değeri incelenerek x_2 girdisinin hücreye olan etkisi ve önemi belirlenebilir. Ağırlıkların değerleri hem sabit hem de değişken olabilir. Ayrıca ağırlıklar negatif değerler de alabilir. Bundan dolayı ağırlıkların küçük veya büyük

değerler alması, ilgili girdinin önemsiz olduğu şeklinde yorumlanmamalıdır (Öztemel, 2003).

Öte yandan YSA'da öğrenme işlemi, ağırlıkların ayarlanması ile gerçekleşmektedir. Bundan dolayı ağırlıklar, YSA literatüründe büyük bir öneme sahiptir. Girdisi yapılan bir bilgi ağırlık değeri ile çarpılarak yapay sinir hücresindeki çekirdeğe iletilir. Üretilecek çıktı üzerinde bu yolla girdilerin etkisi ayarlanabilmektedir (Çayıroğlu, 2019).

4.3.3 Toplama Fonksiyonları

Bu fonksiyonlar YSA'ya gelen net bilgiyi (ağırlıklarla çarpılan girdi değerlerini) hesaplar. Çoğunlukla ağırlıklı toplam fonksiyonu kullanılsa da birçok farklı fonksiyon tercih edilebilir.

Tablo 4.2'de toplama fonksiyonlarının yaygın olarak kullanılanları verilmiştir.

Tablo 4.2 Yaygın olarak kullanılan toplama fonksiyonları (Günay, 2018)

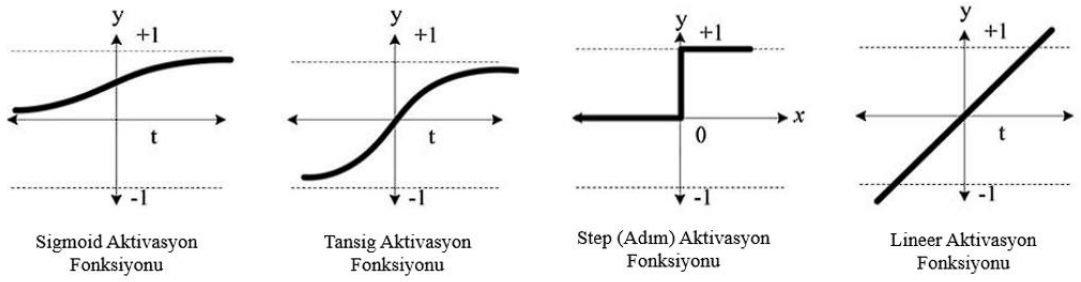
Toplam	$N = \sum_{n=1}^{i} X_n \times W_n$
Çarpım	$N = \prod_{n=1}^{i} X_n \times W_n$
Maksimum	$N = Max(X_n \times W_n)$
Minimum	$N = Min(X_n \times W_n)$
Çoğunluk	$N = \sum_{n=1}^{\text{İ}} Sgn(X_n \times W_n)$
Kümülâtif Toplam	$N = N(eski) + \sum_{n=1}^{\text{İ}} X_n \times W_n$

Tablo 4.2'de verilen toplam fonksiyonlarında ve Şekil 4.3'te de görülen X, ilgili nöron için girdileri, W ise ilgili nöronun kendinden bir önceki nöronlarla yaptığı bağlantılardan gelen ağırlık değerlerini belirtmektedir. Eşitliklerden görüldüğü üzere, bazı durumlarda net ağırlıkların; minimum, maksimum veya işaret fonksiyonuna göre değerleri de alınabilmektedir. Ayrıca kullanılan bazı toplama fonksiyonlarının gelen girdilerin sayısına önem verdiği görülürken bazılarının ise girdilerin değerine önem verdiği görülmektedir. Toplama fonksiyonları, YSA'daki ilgili hücreye göre değişiklik göstermektedir. Bir YSA modelinde, bütün yapay sinir hücreleri aynı toplam fonksiyonundan oluşmayabilir (Çayıroğlu, 2019; Öztemel, 2003).

4.3.4 Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları, yapay sinir hücrelerinde toplama fonksiyonu ile hesaplanan net girdileri işleyerek hücrenin çıktısının belirlenmesini sağlayan fonksiyonlardır. Toplama fonksiyonlarına benzer olarak, aktivasyon fonksiyonları da ilgili yapay sinir hücresine göre değişiklik göstermektedir.

Birçok aktivasyon fonksiyonu bulunmakla birlikte, YSA'da yaygın olarak kullanılan aktivasyon fonksiyonlarının grafikleri Şekil 4.4'te verilmiştir.



Şekil 4.4 YSA'da kullanılan yaygın aktivasyon fonksiyonlarının grafikleri (Ghasemiyeh vd., 2017)

Makine öğrenmesi literatüründe doğrusal modeller için genellikle doğrusal regresyon yöntemleri kullanılmaktadır. Öte yandan YSA ile de doğrusal regresyon yöntemi gerçekleştirilebilmektedir. Şekil 4.4'te verilen lineer aktivasyon fonksiyonuyla birlikte doğrusal regresyon yöntemi, YSA'da uygulanabilmektedir.

Bunun yanı sıra literatür incelendiğinde, doğrusal problemler için genellikle doğrusal regresyon yöntemlerinin kullanıldığı görülmektedir. YSA'nın ise doğrusal olmayan problemler için genellikle kullanıldığı görülmektedir. Bu durum, YSA'nın çoğu makine öğrenmesi yönteminden ayrılmasını sağlayan doğrusal olmama özelliğinin bir sonucudur. Dolayısıyla, belirli istisnalar dışında YSA'da kullanılan aktivasyon fonksiyonları çoğunlukla doğrusal olmayan aktivasyon fonksiyonlarıdır. Ayrıca, geri beslemeli YSA modellerinde aktivasyon fonksiyonlarının türevlenebilir olması gerekmektedir (Çayıroğlu, 2019; Öztemel, 2003).

4.3.5 Hücrenin Çıktısı

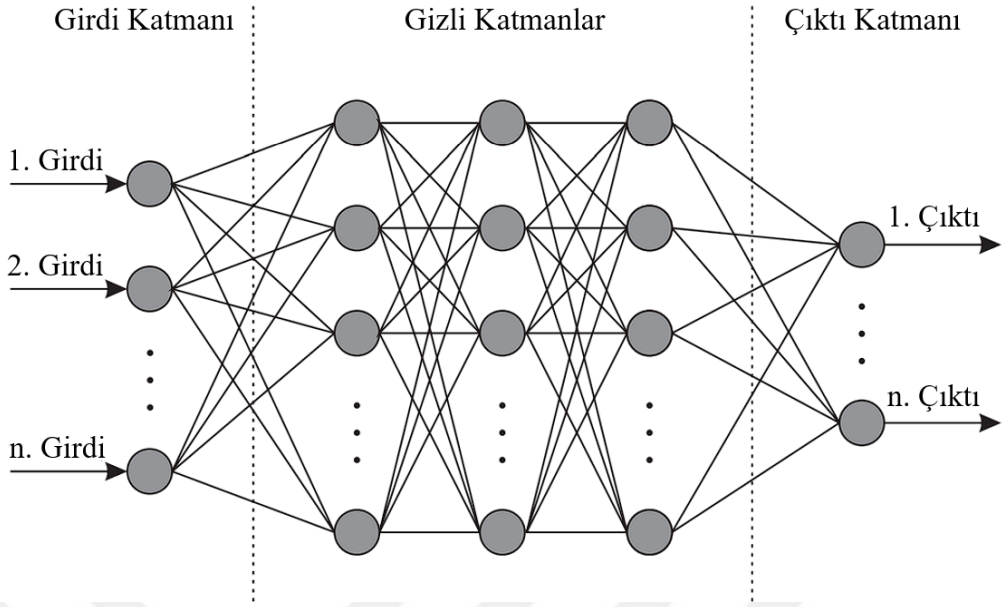
Aktivasyon fonksiyonu ile hesaplanan değer hücrenin çıktısıdır. Genel olarak çıktı değerleri başka bir veya birçok hücreye gönderilebilirken aynı hücre tarafından tekrar girdi olarak da kullanılabilir. Öte yandan eğer ilgili yapay sinir hücresi, YSA'nın çıktı katmanında ise çıktı değeri ilgili yapay sinir hücresinden dış dünyaya gönderilmektedir. Hücrenin çıktıyı nasıl kullanacağı, YSA modeline göre değişiklik göstermektedir. Her hücrenin birden fazla girdisi bulunurken sadece bir çıktısı bulunmaktadır (Çayıroğlu, 2019; Öztemel, 2003).

Gelecek başlıklarda biyolojik sinir hücrelerinden bahsedilmeyecektir. Bundan dolayı gelecek başlıklarda yapay sinir hücresi ifadesi yerine YSA literatüründe de yaygın olarak kullanılan nöron ifadesi kullanılacaktır.

4.4 Yapay Sinir Ağlarının Yapısı

Nöronlar gruplanarak YSA'yı oluşturmaktadır. Bu gruplanma ile ayrıca YSA'da bulunan katmanlar oluşmaktadır. YSA'da üç tip katman bulunmaktadır. Bu katmanlar; girdi katmanı, gizli katman ve çıktı katmanı şeklindedir.

Şekil 4.5'te bir YSA modelinin yapısı verilmiştir.



Şekil 4.5 Üç gizli katmana sahip bir çok katmanlı YSA modelinin yapısı (Bre vd., 2018)

Şekil 4.5 incelendiğinde çember biçimindeki şekiller nöronları belirtmektedir. Nöronların gruplar halinde katmanları oluşturduğu şekilden de görülmektedir. Katmanlar ile ilgili olarak (Çuhadar ve Kayacan, 2005; Uztemur, 2019):

- **Girdi Katmanı:** YSA'da ilk katman girdi katmanı olarak adlandırılır. Bu katman dış ortamdan gelen verileri gizli katmana gönderir. Girdi katmanı, problem üzerinde etkisi bulunan değişkenlerin sayısı kadar nöron içerir. Bu katmanda nöronlar tarafından herhangi bir veri işlenmesi gerçekleştirilmez. Ayrıca, YSA'da giriş katmanından yalnızca bir adet bulunmaktadır.
- **Gizli (Ara) Katman:** Girdi katmanından gelen verilerin işlendiği katmandır. Bundan dolayı, YSA'da öğrenmenin gerçekleştiği asıl katman gizli katmanlardır. Gizli katman adeti, YSA'da bir veya birden fazla olabilir fakat adet arttıkça YSA modelinin öğrenme süresi ve karmaşıklığı da artmaktadır. Ayrıca bazı YSA modellerinde gizli katman bulunmadan da modelin eğitimi gerçekleştirilebilir.
- **Çıktı Katmanı:** Son katman olan çıktı katmanı ise gizli katmanda işlenen verilerden elde edilen çıktıyı dış ortama iletir. Girdi katmanında olduğu gibi çıktı katmanından da YSA'da bir adet bulunmaktadır.

Şekil 4.5'te görüldüğü üzere her nöron, bir sonraki katmanda bulunan diğer bütün nöronlara bağlanır. Bu bağlantılar, ağırlık katsayıları ile temsil edilmektedir. Nöronlar arasındaki bağlantılar ve kendilerine gelen bilgileri işleme yöntemleri, problem için uygulanan YSA modelinin çeşidine göre farklı şekillerde gerçekleşebilir (Hameed, 2017).

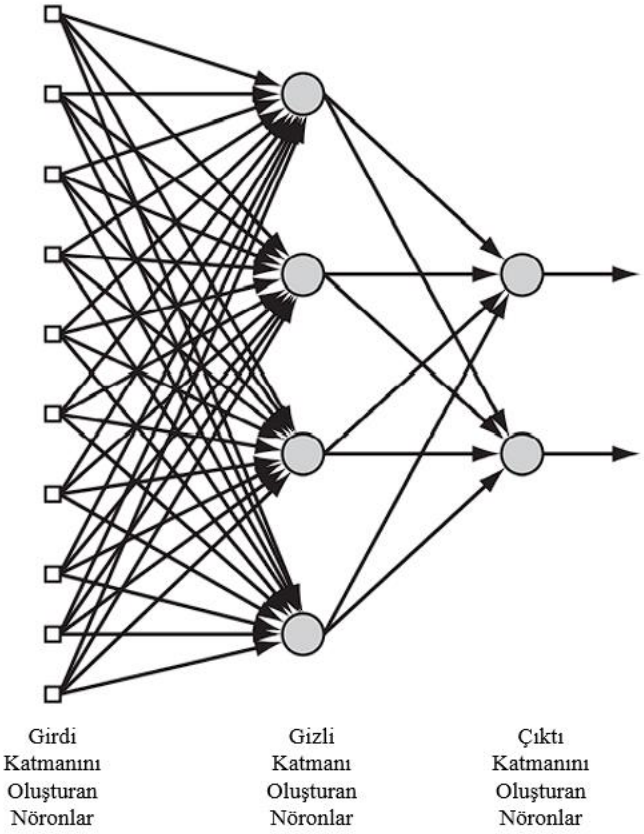
4.5 Yapay Sinir Ağı Mimarileri

Nöronların birbirleri ile bağlanma şekli YSA mimarisini (veya topolojisini) oluşturur. Nöronlar arasındaki ara bağlantıların farklı şekillerde gerçekleşebilmesi birçok olası mimariyi ortaya çıkarmaktadır. Bu olası mimarilerden genel olarak en yaygın olanları, ileri beslemeli ve geri beslemeli mimarileridir. Bundan dolayı, YSA'nın yaygın olarak bu iki mimariye göre sınıflandırılması gerçekleştirilmektedir (Suzuki, 2011).

4.5.1 İleri Beslemeli Yapay Sinir Ağları

Bu tip mimariye sahip YSA modellerinde her bir katmanda bulunan nöronlar, bir sonraki katmanda bulunan diğer nöronlar ile bağlantıya sahiptir. Böylelikle, bir katmandan elde edilen çıktılar bir sonraki katmanın girdilerini oluşturmaktadır. Son olarak çıktı katmanından elde edilen çıktı veya çıktılar ise YSA'nın girdi katmanına gelen girdilere göre oluşturduğu yanıtı temsil etmektedir (Haykin, 2008).

Şekil 4.6'da örnek bir ileri beslemeli mimariye sahip YSA modelinin yapısı verilmiştir.



Şekil 4.6 Örnek bir ileri beslemeli YSA mimarisi (Haykin, 2008)

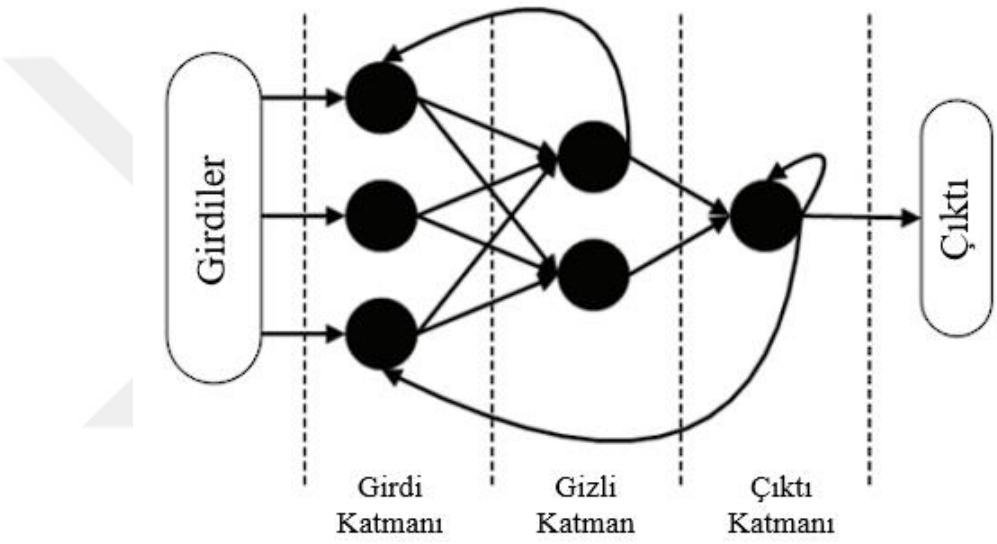
Şekil 4.6'da verilen YSA mimarisi örneğinde verilmiş olan ileri beslemeli YSA modeli tek bir gizli katman içermektedir. Bu YSA modelini 10-4-2 ağı olarak adlandırabiliriz. Çünkü girdi, gizli ve çıktı katmanlarında sırasıyla 10, 4 ve 2 nöron bulunmaktadır. Bunu daha da genelleştirebiliriz. Örneğin, iki gizli katmanı bulunan bir ileri beslemeli YSA modelinde; girdi katmanında, birinci gizli katmanında, ikinci gizli katmanında ve son olarak çıktı katmanında sırasıyla m, h_1, h_2 ve q adet nöron bulunsun. Dolayısıyla bu ileri beslemeli YSA modelini $m - h_1 - h_2 - q$ ağı olarak adlandırabiliriz. Öte yandan, tek katmanlı YSA'da ileri beslemeli YSA modellerindendir (Haykin, 2008).

4.5.2 Geri Beslemeli Yapay Sinir Ağları

Geri beslemeli ağlar, ileri beslemeli ağlardan farklı olarak en az bir adet geri besleme döngüsüne sahiptir. Geri besleme döngüsü, ağdaki bir nörondan elde edilen

çıktının tekrar aynı katmanda veya kendinden önce gelen herhangi bir katmanda bulunan nöronlar için girdi olarak kullanılmasını sağlayan bağlantıdır. Geri besleme döngüsü içeren bir ağ için girdiler sadece tek bir yönde iletilmez, ayrıca geriye doğru da iletilir. Öte yandan, geri besleme döngülerinde herhangi bir sınırlama bulunmamaktadır ve bir nöron, kendisiyle de geri besleme döngüsü oluşturabilir (Haykin, 2008; Suzuki, 2011).

Şekil 4.7'de örnek bir geri beslemeli YSA modelinin yapısı verilmiştir.

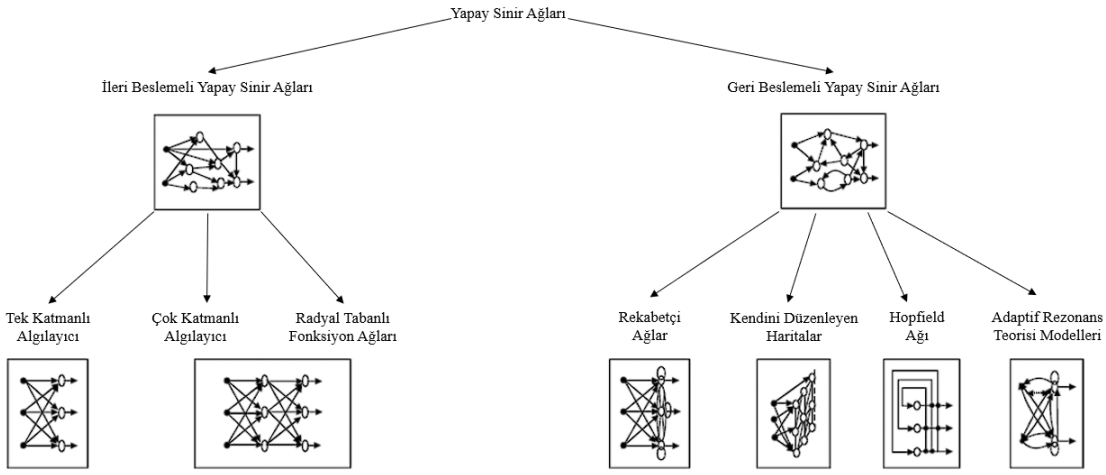


Şekil 4.7 Örnek bir geri beslemeli YSA mimarisi (Suzuki, 2011)

4.6 Yapay Sinir Ağı Modelleri

Genel olarak YSA'nın temel yapısı, nöronlar ve nöronlar arasında bulunan bağlantılar ile aktivasyon fonksiyonlarına dayandığından, YSA modelleri ve mimarileri arasında benzerlikler bulunmaktadır. Farklı YSA modellerinin oluşmasındaki en önemli unsur, öğrenme kurallarıdır. Çünkü çeşitli öğrenme kuralları, ilgili ağ modelleri için mimari yapıyı da değiştirmektedir (Anderson ve Mcneill, 1992).

Şekil 4.8'de geri beslemeli ve ileri beslemeli YSA mimarilerine göre yaygın olarak kullanılan YSA modelleri verilmiştir.



Şekil 4.8 Mimarilerine göre temel YSA modelleri (Verma ve Kumar Singh, 2015)

Özellikle son yıllarda, teknoloji ve büyük veri alanlarındaki gelişmeler ile birlikte yapay zekâ yöntemlerinin kullanımında büyük bir artış yaşanmıştır. Bundan kaynaklı olarak YSA literatürü incelendiğinde, son yıllarda birçok yeni YSA modelinin literatüre eklendiği görülmektedir. Dolayısıyla Şekil 4.8'de en temel YSA modelleri verilmiştir ve verilmiş olan bu ağ modellerinin kullanımı problemden probleme göre değişiklik göstermektedir.

Şekil 4.8 incelendiğinde bu tezin kapsamında gerçekleştirilecek olan çalışmada örüntü tanıma için kullanılacak olan ÇKA modeli görülmektedir. ÖVK modeli ise kendi kendini düzenleyen haritalar (self organizing maps) modelinin öncüsüdür ve her iki model de Teuvo Kohonen tarafından geliştirilmiştir (Wikipedia, 2023).

4.7 Yapay Sinir Ağlarında Öğrenme ve Test

Nöronlar arasındaki bağlantılarda yer alan ağırlıkların ayarlanması ile YSA'da öğrenme işlemi gerçekleşmektedir. Eğitimin başlangıcında ağırlık değerleri rastgele olarak belirlenir ve ardından ağa girdisi yapılan örneklere göre ağırlık değerlerinin ayarlanması gerçekleştirilir. Örneklerin ağa sunulması işlemi, ilgili YSA'nın eğitimi için belirlenen iterasyon sayısı kadar devam eder. Burada amaç hedef çıktıya göre en ideal ağırlık değerlerine ulaşmaktır. İdeal ağırlık değerlerine ulaşıldığında ilgili YSA

artık kendisine sunulan örnek verilere göre genelleme yapabilme özelliği kazanır. İşte bu durum YSA'da öğrenme olarak tanımlanır. Öte yandan ideal ağırlık değerleri belirlenirken ilgili YSA modeline göre farklı öğrenme algoritmaları da kullanılmaktadır (Çok katmanlı algılayıcı modeli ile geri yayılım algoritmasının kullanılması gibi.) (Öztemel, 2003).

YSA'da öğrenme işleminin gerçekleşmesinin ardından elde edilen ağırlık değerleri için doğruluğun ölçülebilmesi açısından test işlemi gerçekleştirilir. Bu işlem için YSA'ya, eğitim sırasında girişi yapılan örnekler dışında hiç görmediği örnekler gösterilir. Gösterilen bu yeni örneklere göre YSA'dan yeni çıktılar elde edilir. Elde edilen bu çıktılar, test işlemi için ağa gösterilen örnek girdilerin gerçek çıktı değerleri olan hedef çıktı değerleri ile karşılaştırılır. Bu karşılaştırma sonucunda belirlenen performans metriğine göre elde edilen hata düzeyi incelenerek ağın öğrenmesinin ne kadar doğrulukla gerçekleştiği hakkında önemli bilgiler elde edilir (Öztemel, 2003).

YSA literatürü incelendiğinde; gizli katman sayısı, gizli katmanlarda bulunan nöron sayısı ve diğer ağ parametrelerini net olarak belirleyecek net bir yöntem bulunmamaktadır. Bundan dolayı YSA'da, ilgili model için çoğunlukla farklı ağ parametreleri kullanılarak ve gizli katmandaki nöron sayıları değiştirilerek birden fazla kez eğitim gerçekleştirilir. Böylelikle deneme yanılma yoluyla ilgili problem için en ideal parametreleri içeren YSA modelinin belirlenmesi sağlanır.

Test veri seti, ağa eğitim sırasında sunulmayan gözlemleri ifade eder ve bu gözlemler eğitimin ardından modelin doğruluğunu test etmek için kullanılır. Bundan dolayı modelin doğruluğu açısından test veri setinin elde edilebilmesi bir makine öğrenmesi modeli için büyük önem taşımaktadır. Öte yandan makine öğrenmesi literatüründe, ilgili problem için kullanılacak olan veri seti sınırlı sayıda gözleme sahipse, eğitim ve test işlemleri için veri setinin belirli oranlarda bölünmesi gerçekleştirilmektedir. Veri setinde bulunan toplam gözlem sayısının, eğitim ve test veri setleri için bölünme işlemi gerçekleştirilirken yaygın olarak kullanılan oran %80 eğitim ve %20 test şeklindedir. Bu oran toplam gözlem sayısına göre %70 – %30, %60 – %40 ve bazen %50 – %50 şeklinde de değişebilmektedir (Joseph, 2022).

Genel olarak makine öğrenmesi literatüründe ilgilenilen probleme göre (tahmin, sınıflandırma, kümeleme vb.) modelin hata düzeyi elde edilirken birçok farklı performans metriği kullanılmaktadır. Bu tezin kapsamında gerçekleştirilecek olan çalışmanın problemi, istatistiksel örüntü tanımadır. Temelinde örüntü tanıma, örüntülerin sınıflandırılması anlamına gelmektedir ve bundan dolayı çoğu örüntü tanıma problemi için sınıflandırma performans metrikleri kullanılabilmektedir. Sınıflandırma türündeki problemler için kullanılan bu performans metrikleri; doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1 (F1 score) şeklindedir. Bu performans metriklerinin hesaplanabilmesi için ise hata matrisine ihtiyaç duyulmaktadır.

4.7.1 Hata Matrisi

Örnek bir hata matrisi Tablo 4.3'te verilmektedir.

Tablo 4.3 İkili sınıflandırma problemleri için örnek bir hata matrisi örneği

		Tahmin Edilen Sınıflar	
Gerçek Sınıflar	**SINIFLAR**	**Pozitif**	**Negatif**
	Pozitif	Gerçek Pozitifler (GP)	Yanlış Negatifler (YN)
	Negatif	Yanlış Pozitifler (YP)	Gerçek Negatifler (GN)

Tablo 4.3'te $2x2$ boyutundaki pozitif ve negatif sınıflarına sahip bir ikili sınıflandırma hata matrisi verilmiştir. Bu matristeki sınıfların kesişimde bulunan elemanlar ile ilgili sınıflandırıcı modeli için doğru ve yanlış sınıflandırma değerlendirilebilir. Matristeki bu elemanlar şu şekilde tanımlanır (Sokolova ve Lapalme, 2009):

- **Gerçek Pozitif (GP):** Doğru tahmin edilen pozitif gözlemlerin sayısıdır.
- **Yanlış Pozitif (YP):** Yanlış tahmin edilen pozitif gözlemlerin sayısıdır.
- **Gerçek Negatif (GN):** Doğru tahmin edilen negatif gözlemlerin sayısıdır.
- **Yanlış Negatif (YN):** Yanlış tahmin edilen negatif gözlemlerin sayısıdır.

Bu elemanların değerleri ile sınıflandırma problemleri için yaygın olarak kullanılan performans metrikleri olan doğruluk, kesinlik, duyarlılık ve F1 metriklerinin hesaplanması gerçekleştirilmektedir. İlgili hesaplamalar için kullanılan eşitlikler Tablo 4.4'te özetlenmiştir:

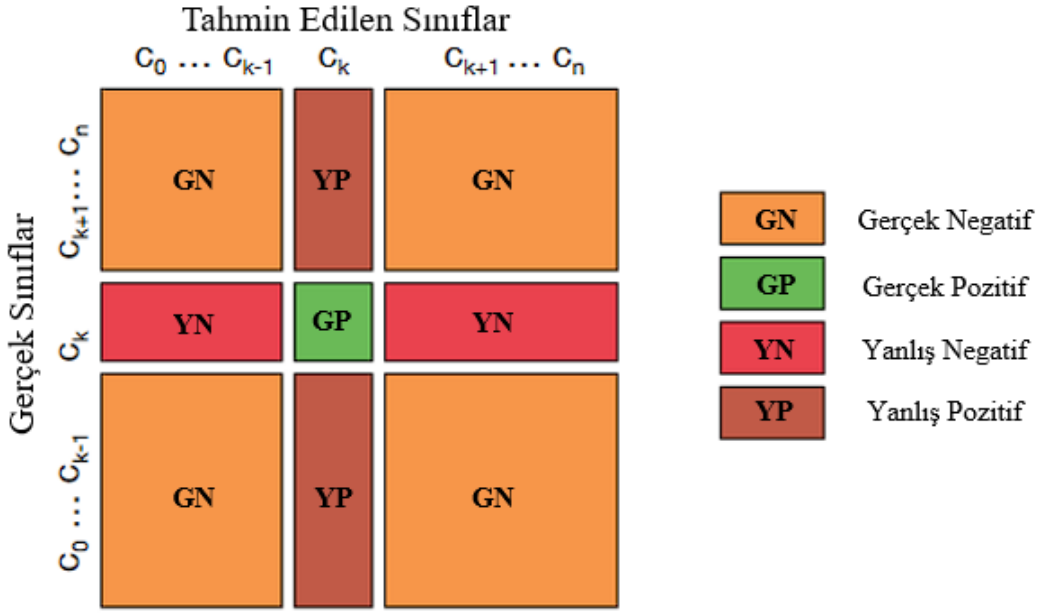
Tablo 4.4 Sınıflandırma problemleri için yaygın olarak kullanılan performans metrikleri (Lever vd., 2016; Lipton vd., 2014)

Doğruluk	$\frac{GP + GN}{GP + GN + YN + YP}$	Bir sınıflandırıcı modeli için sınıflandırma doğruluğunun oranını belirtir.
Kesinlik	$\frac{GP}{GP + YP}$	İlgili sınıflandırıcı modelinin, pozitif gözlemleri yanlış pozitif tanıma gerçekleştirmeden, ne kadar iyi sınıflandırabildiğinin oranını belirtir.
Duyarlılık	$\frac{GP}{GP + YN}$	Tüm pozitif gözlemlerin ilgili sınıflandırıcı modeli tarafından ne kadar iyi sınıflandırıldığının oranıdır.
F1	$2\,x\,\frac{Kesinlik\ x\ Duyarlılık}{Kesinlik + Duyarlılık} = \frac{2GP}{2GP + YP + YN}$	Bu metrik, kesinlik ve duyarlılık değerlerinin harmonik ortalamasından oluşmaktadır. Bundan dolayı model için önemli bir orandır ve bazı durumlarda doğruluk yerine de kullanılmaktadır.

Tablo 4.4'te verilen metriklerden özellikle doğruluk metriği, sınıflandırma problemlerindeki en temel performans metriğidir. Öte yandan bazı sınıflandırma

problemlerinde eğer ilgili gözlemler için elde edilen sınıflar dengesiz bir dağılıma sahipse sadece doğruluk metriği yeterli olmamaktadır. Bu gibi durumlarda kesinlik, duyarlılık ve F1 değerleri ile modelin performansının ölçülmesi mutlaka gerçekleştirilmelidir. Özellikle F1 metriği, bu gibi durumlarda doğruluk değeri yerine de kullanılmaktadır ve hem kesinlik hem de duyarlılık değerlerinin harmonik ortalaması ile önemli bir performans ölçümü sağlamaktadır. Tablo 4.4'te belirtilen bu metrikler ilgili sınıflandırıcı modeli için ne kadar yüksek oranlara sahipse model o kadar tutarlı ve etkili sınıflandırma işlemi gerçekleştirmektedir.

Öte yandan çoklu bir sınıflandırma örneğinde N sınıf sayısı olmak üzere, NxN boyutunda bir hata matrisi elde edilmektedir. Bu tip bir hata matrisinin yapısının anlaşılabilmesi için Şekil 4.9 incelenebilir.



Şekil 4.9 Çoklu sınıflandırma için hata matrisi yapısı (Krüger, 2016)

İkili sınıflandırma problemlerinden farklı olarak çoklu sınıflandırma problemlerinde, iki sınıftan fazla sınıfın sınıflandırılması işlemi amaçlanmaktadır. Şekil 4.9'da n adet sınıf içeren bir çoklu sınıflandırma problemi için hata matrisi yapısı verilmiştir. C_k sınıfına göre şekil incelendiğinde, ilgili matriste yeşil hücre, C_k sınıfına ait olan ve ilgili sınıflandırıcı tarafından da C_k sınıfında sınıflandırılan gözlemlerin

sayısını belirtmektedir. Bu gözlemler C_k sınıfı için doğru sınıflandırılan gözlemlerdir ve gerçek pozitif olarak belirtilirler. Turuncu hücreler, C_k sınıfına ait olmadığı doğru şekilde sınıflandırılmış olan ve gerçekte $C_0 \dots C_{k-1}$, $C_{k+1} \dots C_n$ sınıflarında bulunan gözlemlerdir. Bu gözlemler gerçek negatif olarak belirtilirler. Kahverengi hücreler, C_k sınıfına ait olmayan ($C_0 \dots C_{k-1}$, $C_{k+1} \dots C_n$ sınıflarına ait olan) fakat yanlış bir şekilde C_k sınıfına ait olarak sınıflandırılmış olan gözlemlerdir ve yanlış pozitif olarak belirtilirler. Son olarak, kırmızı hücreler, C_k sınıfına ait olan fakat bu sınıfa ait değil şeklinde, $C_0 \dots C_{k-1}$, $C_{k+1} \dots C_n$ sınıflarında yanlış sınıflandırılan gözlemlerdir ve bu gözlemler yanlış negatif olarak belirtilirler.

Sınıf sayısının artış göstermesinden kaynaklı olarak çoklu sınıflandırma problemlerini çözmek amacıyla oluşturulan sınıflandırıcı modelleri için Tablo 4.4'te verilen performans metriklerinin ilgili sınıf bazında olacak şekilde problemin türüne göre mikro, makro veya ağırlıklı ortalamaları alınmaktadır (Krüger, 2016; Sokolova ve Lapalme, 2009).

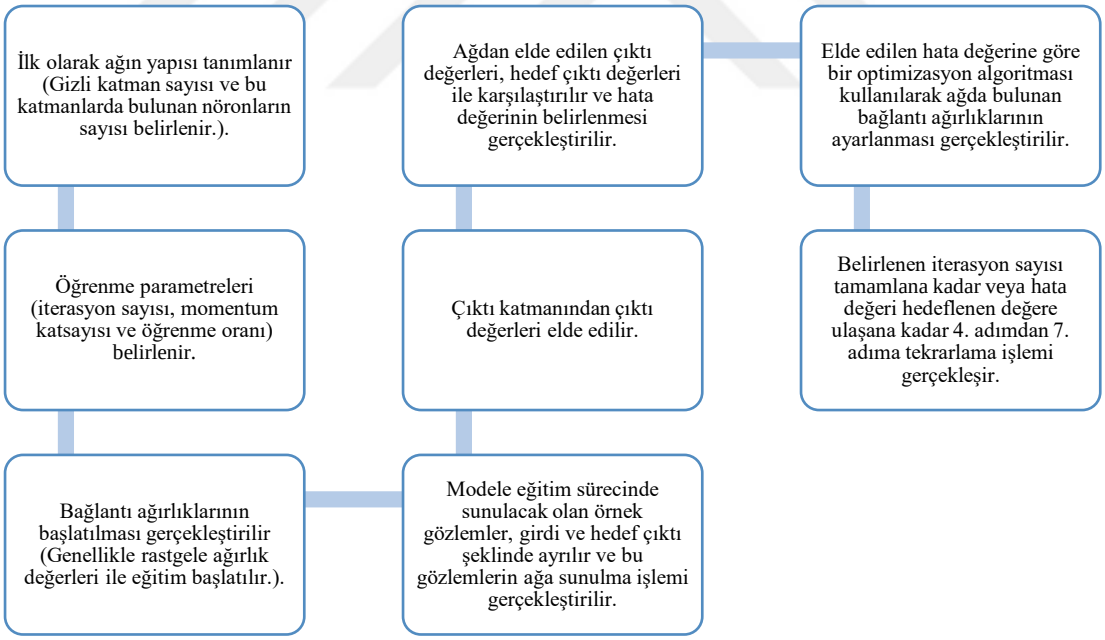
Bu tezin kapsamında yapılan çalışmada, ilgili YSA modellerinden elde edilen en iyi örüntü tanıyıcı modelleri için makro ortalama kullanılarak doğruluk, kesinlik, duyarlılık ve F1 metriklerinin genel değerleri hesaplanacaktır. Makro ortalama kullanılmasının nedeni olarak, kullanılacak olan veri setlerinde bulunan örüntü sayılarının birbirine eşit olması verilebilir. Çünkü makro ortalama tüm örüntülere, örüntülerin sayıları arasındaki farkı gözetmeksizin eşit yaklaşırken, mikro ortalama büyük sayıya sahip örüntüleri destekler. Ayrıca bu tezin kapsamında yapılan çalışmada eğitim ve test veri setlerinde bulunan örüntü sayıları eşit dağılacak şekilde oluşturulacağından dolayı ilgili metriklerin makro ortalama değerleri ile aritmetik ortalama değerleri birbirine eşittir. Bundan dolayı gerçekleştirilecek uygulama kapsamında elde edilecek olan genel sınıflandırıcı performans metrikleri için kullanılacak olan eşitliklerden uygulama bölümünde bahsedilecektir.

4.8 Çok Katmanlı Algılayıcı (Multilayer Perceptron) Modeli

Bir ÇKA modeli, girdi ve çıktı katmanları arasında bir veya daha fazla gizli katmana sahip en temel ileri beslemeli YSA modelidir. Bundan dolayı bu YSA modelinde, çoğu ileri beslemeli YSA'da olduğu gibi nöronlar yardımıyla girdi katmanından çıktı katmanına doğru tek yönde bilgilerin akışı gerçekleşmektedir (Abhishek, 2020).

Şekil 4.5'te üç gizli katmandan oluşan bir ÇKA modeli verilmiştir. Şekilden de görüldüğü üzere herhangi bir katmanda bulunan bir nöronun kendisinden sonra gelen katmanda bulunan bütün nöronlara bağlandığı görülmektedir. Bu durum ağın tamamen bağlı olduğunun bir göstergesidir. Yani bir ÇKA modeli aynı zamanda tamamen bağlı bir sinir ağı modelidir (Haykin, 2008).

Bir ÇKA modeli eğitilirken yaygın olarak Şekil 4.10'da verilen diyagramdaki adımlar izlenir:



Şekil 4.10 ÇKA modeli eğitimi sırasında izlenen adımların diyagramı (Pham ve Oztemel, 1993)

ÇKA modeli birçok öğrenme algoritmasıyla birlikte kullanılabilir. Fakat bu YSA modeli ile en yaygın kullanılan öğrenme algoritması geri yayılım algoritmasıdır.

Literatür incelendiğinde ÇKA ve geri yayılım algoritmasının birçok önemli problemin çözümünde kullanıldığı ve aktif olarak da kullanılmaya devam edildiği görülmektedir.

4.8.1 Geri Yayılım Algoritması

Geri yayılım algoritması (backpropagation algorithm), kullanıldığı YSA modellerinde ağırlıkların ayarlanması için kullanılan bir öğrenme algoritmasıdır. 1970'lerin başında bağımsız kaynaklarca geliştirilmeye başlanmıştır. Bu bağımsız kaynaklardan olan Rumelhart, Hinton ve Williams tarafından 1986 yılında kullanımı önerilmiştir. Ardından bu öğrenme algoritması ile birçok farklı ağ mimarisi oluşturulmuştur. Bu ağ mimarilerinden en etkili ve yaygın olarak kullanılanı ise ÇKA ve geri yayılım algoritması birleşimidir. Bundan dolayı çoğu farklı problemde ve uygulamada bu birleşim aktif olarak kullanılmaktadır (Anderson ve Mcneill, 1992; Graupe, 2007).

Geri yayılım algoritması, ilgili ağın çıktısından elde edilen hatanın geriye doğru yayılmasını gerçekleştirerek hatanın azalmasını sağlar. k. iterasyonda i ve j nöronları arasındaki bağlantıda bulunan ağırlık değerinin değişimi $\Delta w_{ji}(k)$ olmak üzere, $\Delta w_{ji}(k)$ aşağıdaki eşitlik ile elde edilir (Sağıroğlu vd., 2000):

$$\Delta w_{ji}(k) = -\alpha \frac{\partial E}{\partial w_{ji}(k)} + \mu \Delta w_{ji}(k-1) \tag{4.1}$$

Yukarıda verilen eşitlikte α öğrenme oranını, μ momentum katsayısını ve $\Delta w_{ji}(k-1)$ bir önceki iterasyonda i ve j nöronları arasındaki ağırlığın değişimini ifade etmektedir. Bir ÇKA modelini geri yayılım algoritması ile eğitmek, ilgili ağ modeline sunulacak olan bütün örnek gözlemlerin (eğitim için kullanılacak olan gözlemler), belirli sayıda gözlem içeren gruplar şeklinde (girdi ve hedef çıktı grupları) her iterasyonda sırayla sunulması anlamına gelmektedir. Bu sunulma işlemi sırasında ÇKA modelinden elde edilen çıktı ile hedef çıktı arasındaki hata değerleri, ağırlıkları ayarlamak için ağ üzerinden geri yayılır. Bu durum ilgili ÇKA modeli için belirlenen iterasyon sayısı bitene kadar veya hedeflenen ağırlık değerlerine ulaşılana kadar

devam eder. Ayrıca bir eğitim iterasyonu, ilgili gözlem grubunun ağa sunulmasından ve ağırlıkların bu gözlem grubuna göre güncellenmesinden sonra tamamlanmaktadır (Sağıroğlu vd., 2000).

4.9 Öğrenmeli Vektör Kuantizasyon (Learning Vector Quantization) Modeli

ÖVK modeli 1984 yılında Teuvo Kohonen tarafından kullanımı önerilmiş olan bir ileri beslemeli YSA modelidir. ÖVK modeli yapısı gereği genel olarak sınıflandırma problemlerinde kullanılmaktadır. Temel mantığı, bir n boyutlu girdi vektörünü, elde hazır bulunan bir referans vektör setine haritalandırılmasını (uydurulmasını) sağlamaktır. Burada haritalandırmak ile anlatılmak istenilen, ÖVK modelinde gerçekleşen öğrenme ile girdi vektörünün, hangi referans vektör tarafından temsil edileceğinin belirlenmesidir (Çetin ve Temurtaş, 2018; Öztemel, 2003).

ÖVK modeli rekabetçi öğrenme algoritmaları arasında yerini almaktadır. Bundan dolayı k – ortalamalar (k – means) kümeleme yöntemi ve kendi kendini düzenleyen haritalar gibi diğer rekabetçi öğrenme algoritmaları ile yakından ilgilidir. Rekabetçi öğrenme algoritmalarında gerçekleşen eğitim aşamasında, yalnızca belirli komşulukların veya öğelerin güncellenmesinin gerçekleştiği, kazanan her şeyi alır öğrenme kuralı veya bu öğrenme kuralına dayalı varyantlar kullanılır (Gauri, 2010; Nova ve Estévez, 2014).

ÇKA modelinden farklı olarak ÖVK modelindeki gizli katman, Kohonen katmanı (veya rekabetçi katman) olarak adlandırılmaktadır. Kohonen katmanı, kazanan her şeyi alır öğrenme kuralının uygulandığı katmandır. ÖVK modeli, k – en yakın komşu (k – nearest neighbors) algoritmasını temel alır ve vektör kuantizasyon işlemi gerçekleştirilerek girdi vektörlerinin önceden belirlenmiş sayıda referans vektöre göre sınıflandırılmasını gerçekleştirir (Pham ve Oztemel, 1994).

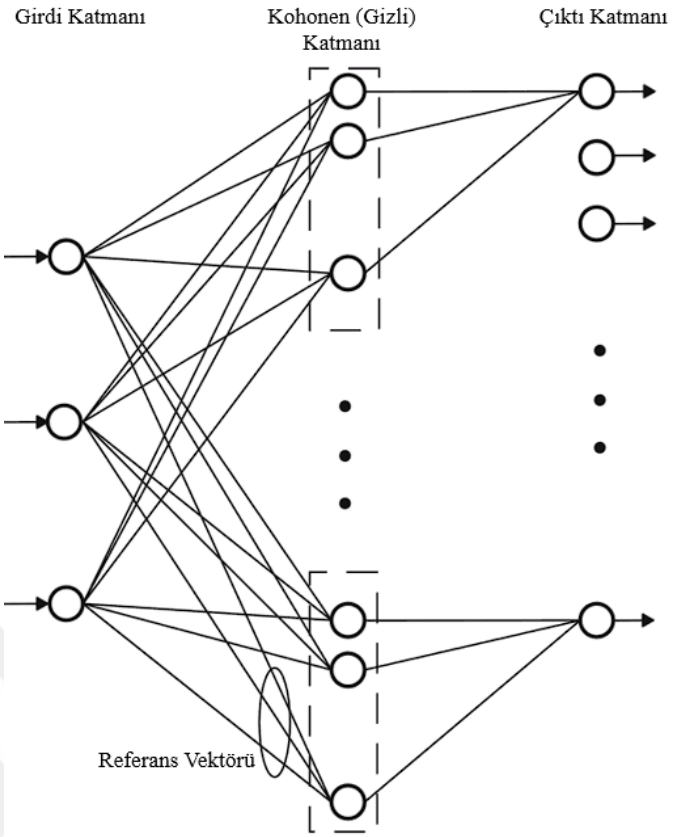
Referans vektörlerinin sayısı, Kohonen katmanında belirlenen nöron sayısı kadardır. Sınıflandırma işlemi, girdi vektörü ve referans vektörleri arasındaki öklid

uzaklıklarının belirlenmesi ile gerçekleştirilir. Öklid uzaklıklarını hesaplama işlemi aşağıda verilen eşitlik ile gerçekleştirilir (Çetin ve Temurtaş, 2018; Öztemel, 2003):

$$d_i = \sqrt{\sum_j (R_{ij} - x_j)^2} \tag{4.2}$$

Bu eşitlikte; x_j, girdi vektörünün j. iterasyondaki değerini belirtir. R_{ij}, i. nöronun temsil ettiği referans vektörünün j. iterasyondaki değerini belirtir. d_i ise i. nöronun temsil ettiği referans vektörü ve girdi vektörü arasındaki uzaklığı temsil eder (Çetin ve Temurtaş, 2018). Girdi vektörü ve referans vektörleri ile elde edilen öklid uzaklıklarına göre ilgili girdi vektörü için hangi referans vektörü ile en yakın uzaklığa sahip olduğu belirlenir. En yakın uzaklığı sağlayan referans vektörünü temsil eden nöron Kohonen katmanında kazanan pozisyonunu alır. Böylelikle ilgili nöronun ağırlık değerinin güncellenmesi gerçekleştirilir. Bu şekilde Kohonen katmanı ve girdi katmanı arasındaki ağırlıkların ayarlanması gerçekleştirilir (Öztemel, 2003).

Şekil 4.11'de bir ÖVK modeli örneği verilmiştir:



Şekil 4.11 Örnek bir ÖVK ağ modelinin yapısı (Pelàez ve Castillo Piqueira, 2008)

Şekil 4.11 incelendiğinde Kohonen katmanı ile çıktı katmanı arasındaki bağlantının kısmen olduğu görülmektedir. Ayrıca Şekil 4.11'den de görüldüğü üzere her iterasyonda referans vektörlerinin girdi vektörlerine yakınlaşması sağlanmaktadır. Bu yakınlaşma ilgili iterasyonda kazanan nöronun temsil ettiği referans vektörünün, ilgili girdi vektörüne yakınlaşması ile gerçekleştirilmektedir. Öte yandan ilgili iterasyonda kaybeden nöronların temsil ettiği referans vektörleri ise ilgili girdi vektöründen uzaklaştırılmaktadır. Referans vektörlerinin yakınlaştırılmasını ve uzaklaştırılmasını sağlayan eşitlikler sırasıyla aşağıdaki gibidir (Çetin ve Temurtaş, 2018; Öztemel, 2003):

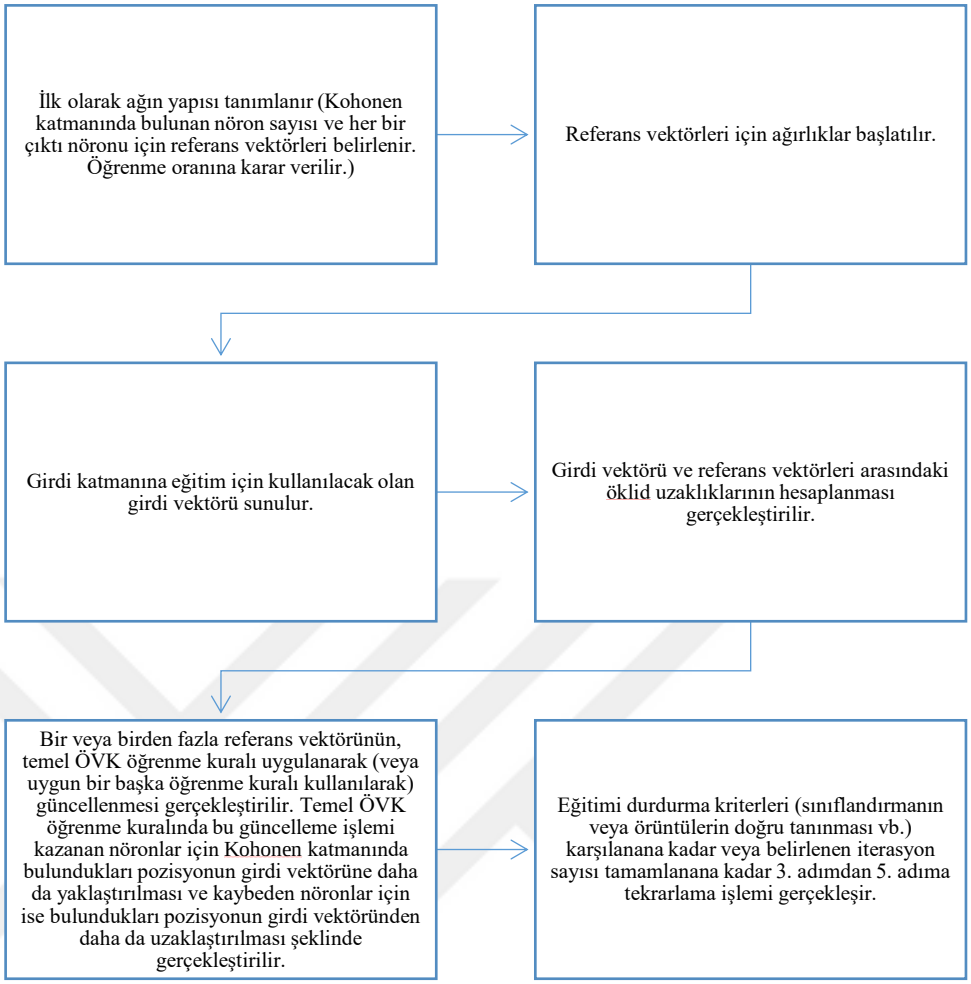
$$R_{yi} = R_{ei} + \lambda(R_{ei} - x_i) \tag{4.3}$$

$$R_{yj} = R_{ej} + \lambda(R_{ej} - x_j) \tag{4.4}$$

Verilen eşitliklerde λ öğrenme oranını temsil etmektedir. $i.$ ve $j.$ iterasyonlar birbirinden bağımsız olmak üzere; R_{yi} ve R_{yj} sırasıyla $i.$ ve $j.$ iterasyonlardaki referans vektörlerinin yeni değerlerini, R_{ei} ve R_{ej} sırasıyla $i.$ ve $j.$ iterasyonlardaki referans vektörlerinin eski değerlerini, x_i ve x_j ise sırasıyla $i.$ ve $j.$ iterasyonlardaki girdi vektörlerini temsil etmektedir (Çetin ve Temurtaş, 2018).

ÖVK modeli için çıktı ve Kohonen katmanları arasındaki ağırlık değerleri sabit olarak 1'dir. Ayrıca çıktı ve Kohonen katmanlarında bulunan nöronların çıktı değerleri de 1 veya 0 olacak şekilde ikili (binary) değerler alır. Kohonen katmanında kazanan nöron çıktı olarak 1 değerini ve katmanda bulunan diğer nöronlar (kaybeden nöronlar) ise çıktı olarak 0 değerini verir. Ardından Kohonen katmanından 1 değerinin yollandığı çıktı katmanındaki ilgili nöron, ağ çıktısı olarak 1 değerini ve çıktı katmanındaki diğer nöronlar ise 0 değerini verir. Böylelikle ilgili girdi vektörüne göre ÖVK modelinde ağırlıkların ayarlanması gerçekleştirilmiş olur (Öztemel, 2003).

Bir ÖVK modeli eğitilirken ÇKA modeline benzer bir şekilde, Şekil 4.12'de verilen diyagramdaki adımlar yaygın olarak izlenmektedir.



Şekil 4.12 ÖVK modeli eğitimi sırasında izlenen adımların diyagramı (Pham ve Oztemel, 1993)

BÖLÜM BEŞ
UYGULAMA

5.1 Çalışmanın Tanıtılması

Bu çalışmada, 3. Bölümde detaylıca bahsedilmiş olan $\overline{X}$ kontrol kartlarında ortaya çıkabilecek olan örüntülerin ÇKA ve ÖVK modelleri ile en iyi şekilde tanınmasını sağlayan YSA modelinin belirlenmesi amaçlanmaktadır. Çoğu çalışmadan farklı olarak bu çalışmada, Şekil 3.7'de verilmiş olan 7 temel örüntünün yanı sıra bu 7 temel örüntüden türetilmiş olan 9 adet eş zamanlı örüntünün de ilgili YSA modeli tarafından tanınması amaçlanmıştır.

Çalışma tümüyle MATLAB® ortamında gerçekleştirilmiş ve simülasyon yoluyla elde edilen süreçler aracılığıyla ilgili örüntüler elde edilmiştir. Öte yandan simülasyon için gerekli olan parametreler gerçek bir süreç için belirlenip değiştirilerek, gerçek bir süreç için de aynı çalışma uygulanabilmektedir. Bundan dolayı, bu tez kapsamında yapılacak olan çalışmadan yola çıkarak herhangi bir süreç için KKÖT sistemi oluşturulabilir.

5.2 Verilerin Elde Edilmesi

Temelde bir KKÖT sisteminin oluşturulabilmesi için örnek örüntülerin gerçek bir süreçten elde edilmesi gerekmektedir. Ancak, iyi bir KKÖT sistemi çok sayıda örüntü gerektirmektedir. Bu örüntülerin süreçten toplanması ise ilgili süreç için çok fazla zaman ve maliyet anlamına gelmektedir. Bu sorunun çözümü için araştırmacılar literatürde simülasyon tekniğini önermektedir. Simülasyon tekniği yardımıyla ilgili örüntülerin elde edilmesini sağlayan süreçler simüle edilebilmektedir (Gauri ve Chakraborty, 2008). Bundan dolayı bu çalışmada da hem eğitim hem de test veri setlerinin oluşturulması için simülasyon tekniği kullanılacaktır.

Bu çalışmada hem temel örüntüler için süreçler hem de eş zamanlı örüntüler için süreçler simüle edilecektir. İlk olarak temel örüntüler için süreçler sırasıyla aşağıda verilen Eşitlik (5.1) ve Tablo 5.1 yardımıyla simüle edilmektedir (Hong vd., 2019):

$$y_t = \mu + r_t\sigma + d(t) \tag{5.1}$$

Tablo 5.1 Temel yedi kontrol kartı örüntüsünü oluşturan süreçler için simülasyon formülleri (Hong vd., 2019)

Örüntü Tipi	**Örüntü Parametreleri**	**Parametre Değerleri**	**Örüntü Formülü ($d(t)$)**
NOR	Ortalama (μ) Standart Sapma (σ)	80 5	-
TEÇ	Periyodik Değişim Değeri (α) Periyod (T)	$1{,}5\sigma \leq \alpha \leq 2{,}5\sigma$ 8	$\alpha * \sin(2\pi t/T)$
SİS	Sistematik Hareket Değeri (u)	$1\sigma \leq u \leq 3\sigma$	$u(-1)^t$
ART	Trend Eğimi (g)	$0{,}05\sigma \leq g \leq 0{,}1\sigma$	$t * g$
AZT	Trend Eğimi (g)	$-0{,}1\sigma \leq g \leq -0{,}05\sigma$	$-t * g$
YAK	Kayma Pozisyonu (k) Kayma Miktarı (s)	$k = 1, k = 0$ $1{,}5\sigma \leq s \leq 2{,}5\sigma$	$k * s$
AAK	Kayma Pozisyonu (k) Kayma Miktarı (s)	$k = 1, k = 0$ $-2{,}5\sigma \leq s \leq -1{,}5\sigma$	$-k * s$

Hem Eşitlik (5.1) hem de Tablo 5.1'de verilen parametreler arasındaki en önemli parametreler, standart sapma (σ) ve ortalama (μ) parametreleridir. Bu parametreler

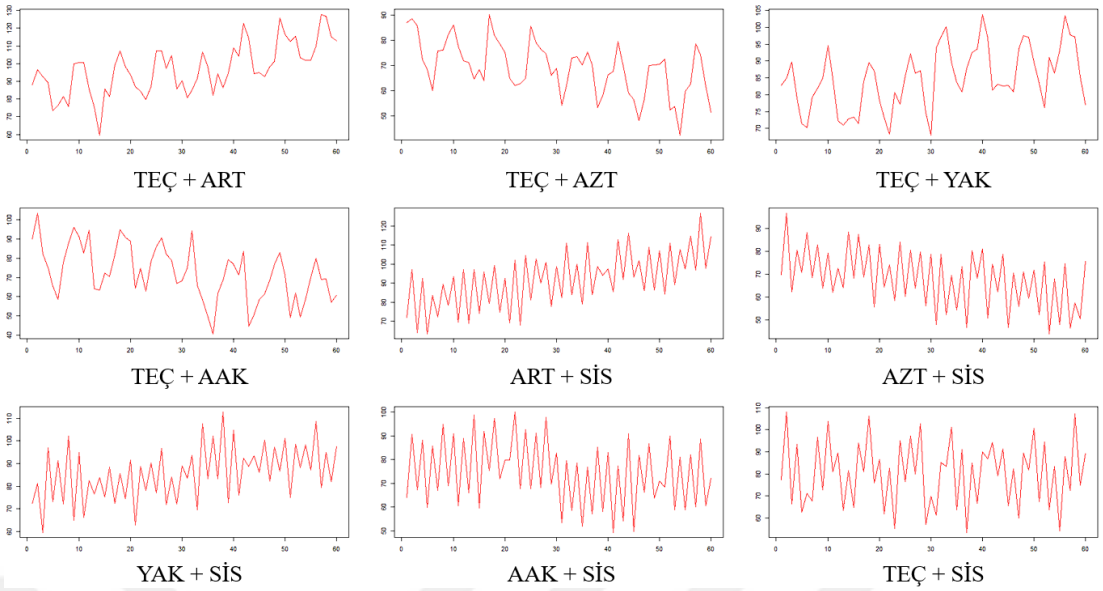
ilgilenilen süreç için değişiklik gösteren en temel değerlerdir. Bundan dolayı herhangi bir gerçek süreç için simülasyon yöntemi ile örüntü elde edilmek isteniyorsa ilk olarak ilgili sürece göre bu parametrelerin çok hassas bir şekilde belirlenmesi gerekmektedir. Literatürde yaygın olarak kullanıldığı için bu çalışmada da σ değeri 80 ve μ değeri 5 olarak belirlenmiştir.

Eşitlik (5.1) ve Tablo 5.1 incelendiğinde; $d(t)$, oluşturulacak olan örüntü için ilgili sürecin simüle edilmesini sağlayan formülü temsil etmektedir. Bundan dolayı literatürde örüntü formülü olarak isimlendirilmektedir. $t(= 1, \dots, n)$, bir örüntü için oluşturulan gözlem sayısını, r_t standart normal dağılımla oluşturulan rassal sayıyı ve y_t, t'inci gözlem verisinin değerini belirtmektedir. k parametresi ise ilgili örüntüde bulunan kayma pozisyonundan önceki gözlemler için 0 değerini ve kayma pozisyonundan sonraki gözlemler için 1 değerini almaktadır. Diğer parametreler ise verilen aralıklarda rastgele değerler almaktadır.

Öte yandan, eş zamanlı örüntüler oluşturulurken Eşitlik (5.2)'den faydalanılır (Hong vd., 2019).

$$y_t = \mu + r_t\sigma + d_1(t) + d_2(t) \tag{5.2}$$

Eşitlik (5.2)'de, $d_1(t)$ ve $d_2(t)$ birleştirilecek olan iki örüntüyü oluşturan süreçlerin formüllerini temsil eder. Örneğin, eğer tekrarlayan çevrim ve yukarı ani kayma örüntüleri ile türetilmiş bir eş zamanlı örüntü elde edilmek isteniyorsa; $d_1(t)$ için tekrarlayan çevrim ve $d_2(t)$ için yukarı ani kayma örüntülerini oluşturan süreçlerin formüllerinin ilgili parametrelerle birlikte kullanılması gerekmektedir. Şekil 5.1'de verilen 9 adet eş zamanlı örüntü Eşitlik (5.2) yoluyla elde edilmektedir.



Şekil 5.1 Eş zamanlı kontrol kartı örüntüleri

Bu tez kapsamında gerçekleştirilecek olan çalışma için eğitim veri seti, 16 örüntünün her biri için 1500 örnek olmak üzere toplam 24000 örnek ve 60 gözlem verisinden ($24000x60$) oluşmaktadır. Test veri seti ise 16 örüntünün her biri için 375 örnek olmak üzere toplam 6000 örnek ve 60 gözlem verisinden ($6000x60$) oluşmaktadır. Eğitim ve test veri setleri, toplamda 30000 örnek hedef alınarak ve örnek sayıları %80 – %20 oranlarında olacak şekilde ayrı ayrı simüle edilmiştir. Her iki veri seti de MATLAB® ortamında, yukarıda verilen formüllerin kodlanmasıyla simüle edilmiştir. Toplamda 16 örüntü için süreç simülasyonu gerçekleştirilerek ilgili veri setlerinin oluşturulması sağlanmıştır.

Öte yandan ilgili örüntüler için hedef çıktı veri seti, Tablo 5.2'de verilen hedef çıktılara göre oluşturulmuştur.

Tablo 5.2 İlgili örüntülerin hedef çıktıları

Örüntü Sınıfı	Örüntü Kodu	Çıktılar															
1	NOR	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	TEÇ	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	SİS	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	ART	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
5	AZT	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	YAK	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
7	AAK	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
8	TEÇ + ART	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
9	TEÇ + AZT	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
10	TEÇ + YAK	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
11	TEÇ + AAK	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
12	ART + SİS	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
13	AZT + SİS	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
14	YAK + SİS	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
15	AAK + SİS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
16	TEÇ + SİS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Son olarak eğitim sürecinin daha hızlı ve etkili sonuç verebilmesi için girdi veri setlerine (hem eğitim hem de test veri setleri için), makine öğrenmesi literatüründe sıkça kullanılan normalizasyon ve ölçekleme işlemleri gerçekleştirilmiştir. Eğitim ve test için kullanılacak olan girdi veri setleri [-1,1] aralığında aşağıda verilen eşitliğe göre normalize edilmiştir (Gauri, 2010):

$$s_j = [2(f_j - f_{min})/(f_{max} - f_{min}) - 1] \quad (5.3)$$

Eşitlik incelendiğinde; s_j ve f_j, j. gözlem için sırasıyla ölçeklenmiş ve gerçek değerlerdir. Bu çalışmada veri setlerinde 60 gözlem verisi bulunduğu için $j = 1,2,\dots,60$ şeklindedir. f_{max} ve f_{min} değerleri ise ilgili örnek için sırasıyla maksimum ve minimum değerleri belirtmektedir (Gauri, 2010).

5.3 Çok Katmanlı Algılayıcı Modelinin Yapısı ve Eğitimi

ÇKA modeliyle oluşturulmuş olan çalışmada; girdi katmanında, veri setinde 60 gözlem verisi bulunduğu için 60 nöron bulunmaktadır. Çıktı katmanında ise 16 örüntünün tanınması gerçekleştirileceği için her biri bir örüntüyü temsil eden toplam 16 nöron bulunmaktadır. Gizli katmanda bulunan ideal nöron sayısının belirlenebilmesi için gizli katmandaki nöron sayısı 29 nörondan 36 nörona doğru değiştirilerek ÇKA modelleri oluşturulmuş ve sonuçlar elde edilmiştir.

ÇKA modeli ile gerçekleştirilen çalışmada MATLAB® ortamında bulunan üç tip geriye yayılımlı öğrenme algoritması kullanılmıştır. Bu algoritmalar; Levenberg – Marquardt (trainlm), bayes düzenlileştirme (trainbr), momentum ve adaptif öğrenme oranı ile gradyan iniş (traingdx) şeklindedir (Demuth ve Beale, 1992; Gauri ve Chakraborty, 2008). MATLAB® ortamında ÇKA modeli için birçok eğitim algoritması bulunmaktadır. Fakat KKÖT literatürü incelendiğinde; traingdx, trainlm ve trainbr eğitim algoritmaları ile yapılan çalışmalar daha ön plana çıkmaktadır. Bundan dolayı, bu tezin kapsamında yapılan çalışma için de ÇKA modeli ile bu üç eğitim algoritması kullanılmıştır.

Genel olarak YSA modelleri eğitilirken bazı öğrenme parametrelerinin belirlenmesi gerekmektedir. Bu çalışmada da ÇKA modeli için bazı öğrenme parametreleri KKÖT literatürü aracılığıyla belirlenmiştir. Bu parametrelerin modelin öğrenmesindeki etkisi büyüktür. Bu çalışma için ÇKA modeli ile gerçekleştirilen eğitimlerde kullanılan ideal öğrenme parametreleri şu şekildedir (Gauri ve Chakraborty, 2008):

- **Maksimum epoch (iterasyon) sayısı:** 2500
- **Maksimum hata oranı:** 500

- **Öğrenme oranı:** 0,1
- **Aktivasyon fonksiyonları:** Hem gizli katman hem de çıktı katmanı için aktivasyon fonksiyonu, Şekil 4.4'te de verilen hiperbolik tanjant (tansig) aktivasyon fonksiyonu olarak belirlenmiştir.
- **Eğitim algoritması:** Sırasıyla traingdx, trainlm ve trainbr olarak belirlenerek sonuçlar elde edilmiştir (Trainlm algoritması, MATLAB® ortamında ÇKA modeli için varsayılan eğitim algoritmasıdır).

İlgili eğitim algoritmalarına göre eğitimi gerçekleştirilen ÇKA modellerinden elde edilen sonuçlar sırasıyla Tablo 5.3, 5.4 ve 5.5'te verilmiştir. Ayrıca MATLAB® ortamında ÇKA modeli ile gerçekleştirilen eğitimler için oluşturulan kodlar EK 4 'te verilmektedir.

Tablo 5.3 ÇKA modeli ve traingdx eğitim algoritması ile elde edilen sonuçlar

Gizli katmandaki nöron sayısı	Traingdx eğitim algoritması ile eğitilen ÇKA model adı	Örüntü tanıma yüzdesi (Test) (%)	Örüntü tanıma yüzdesi (Eğitim) (%)
29	TRAINGDX – 1	91,3	91,6
30	TRAINGDX – 2	96,2	96,7
31	TRAINGDX – 3	90,8	91,4
32	TRAINGDX – 4	91,2	91,5
33	TRAINGDX – 5	96,1	96,4
34	TRAINGDX – 6	91,3	91,4
35	TRAINGDX – 7	96,4	96,7
36	TRAINGDX – 8	95,3	95,9
	ORTALAMA	93,6	94,0

Tablo 5.4 ÇKA modeli ve trainlm eğitim algoritması ile elde edilen sonuçlar

Gizli katmandaki nöron sayısı	Trainlm eğitim algoritması ile eğitilen ÇKA model adı	Örüntü tanıma yüzdesi (Test) (%)	Örüntü tanıma yüzdesi (Eğitim) (%)
29	TRAINLM – 1	91,7	92,1
30	TRAINLM – 2	96,7	96,7
31	TRAINLM – 3	95,2	95,6
32	TRAINLM – 4	95,0	95,7
33	TRAINLM – 5	95,1	95,5
34	TRAINLM – 6	97,1	97,3
35	TRAINLM – 7	95,7	96,2
36	TRAINLM – 8	89,8	90,8
	ORTALAMA	94,5	95,0

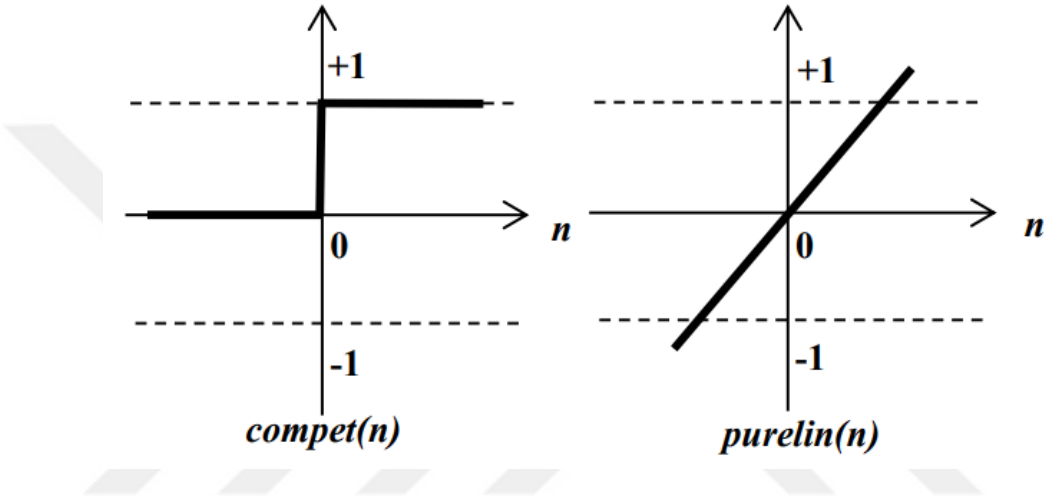
Tablo 5.5 ÇKA modeli ve trainbr eğitim algoritması ile elde edilen sonuçlar

Gizli katmandaki nöron sayısı	Trainbr eğitim algoritması ile eğitilen ÇKA model adı	Örüntü tanıma yüzdesi (Test) (%)	Örüntü tanıma yüzdesi (Eğitim) (%)
29	TRAINBR – 1	91,8	92,4
30	TRAINBR – 2	96,7	97,2
31	TRAINBR – 3	98,9	99,6
32	TRAINBR – 4	91,5	92,3
33	TRAINBR – 5	92,5	92,9
34	TRAINBR – 6	91,5	92,2
35	TRAINBR – 7	97,9	98,8
36	TRAINBR – 8	97,8	98,0
	ORTALAMA	94,8	95,4

5.4 Öğrenmeli Vektör Kuantizasyon Modelinin Yapısı ve Eğitimi

ÖVK modelinde de ÇKA modeline benzer şekilde girdi katmanında 60 nöron ve çıktı katmanında 16 nöron bulunmaktadır. Kohonen katmanındaki ideal nöron sayısının belirlenebilmesi için yine ÇKA modelindeki gizli katmana benzer olarak, Kohonen katmanındaki nöron sayısı 29 nörondan 36 nörona doğru değiştirilerek ÖVK modelleri eğitilmiş ve ardından sonuçlar elde edilmiştir.

Öte yandan, ÖVK modeli için de yine bazı öğrenme parametrelerinin belirlenmesi gerekmektedir. MATLAB® ortamında ÖVK modeli için Şekil 4.4'te verilen aktivasyon fonksiyonları yerine, Kohonen katmanı için compet ve çıktı katmanı için purelin aktivasyon fonksiyonları kullanılmaktadır (Avgikou vd., 2006). Bu iki aktivasyon fonksiyonu MATLAB® ortamında ÖVK modeli için varsayılan olarak kullanılan aktivasyon fonksiyonlarıdır. Bu aktivasyon fonksiyonlarının grafikleri Şekil 5.2'de verilmektedir.



Şekil 5.2 ÖVK modeli için kullanılan aktivasyon fonksiyonlarının grafikleri (Koltan Yılmaz, 2014)

Ayrıca öğrenme oranı ÖVK modelinde ÇKA modelinden farklı olarak 0,01 olarak belirlenmiştir. ÖVK modeli için KKÖT literatürü incelendiğinde, 6 veya 9 adet temel örüntünün tanınması için gerçekleştirilen çalışmalarda yaygın olarak öğrenme oranının 0,1 olarak belirlendiği görülmektedir. Fakat bu çalışmada, tanınması amaçlanan örüntü sayısı 16 olduğu için ÖVK modelinde 0,1 öğrenme oranı, ağırlıkların örüntü sayısına göre göre daha büyük bir şekilde güncellenmesini sağlamaktadır. Dolayısıyla bu durum, modelin öğreniminin yetersiz bir şekilde gerçekleşmesine neden olmaktadır. Bundan dolayı, 0,1 öğrenme oranı ile de ÖVK modellerinin eğitimi denenmiş fakat herhangi bir örüntü tanıma yüzdesi elde edilememiştir. Öte yandan, ÖVK modelinde kazanan her şeyi alır öğrenme kuralı uygulandığı için ÇKA modelinden farklı olarak sadece kazanan nöronların ağırlıklarının güncellenmesi gerçekleştirilir. Bundan dolayı, ÇKA modeline kıyasla ağırlık güncellemelerinin sayısı ÖVK modelinde daha azdır. Dolayısıyla ÇKA

modellerinin eğitimi için 2500 olarak belirlenen epoch sayısı, ÖVK modellerinin eğitimi için 100 olarak belirlenmiştir. ÖVK modelinin eğitimi için kullanılacak olan öğrenme parametreleri şu şekilde özetlenebilir:

- **Maksimum epoch (iterasyon) sayısı:** 100
- **Öğrenme oranı:** 0,01
- **Aktivasyon fonksiyonları:** Gizli katman için compet ve çıktı katmanı için purelin aktivasyon fonksiyonları belirlenmiştir.
- **Eğitim algoritması:** MATLAB® ortamında ÖVK modeli için varsayılan eğitim algoritması olan learnlv1 eğitim algoritması kullanılmıştır.

ÖVK modellerinin eğitimlerinden elde edilen sonuçlar Tablo 5.6'da verilmiştir. Ayrıca ÖVK modeli ile gerçekleştirilen eğitimler için MATLAB® ortamında oluşturulan kodlar EK 5'te verilmektedir.

Tablo 5.6 ÖVK modeli ile elde edilen sonuçlar

Kohonen katmanındaki nöron sayısı	**Epoch sayısına göre eğitilen ÖVK modelinin adı**	**Örüntü tanıma yüzdesi (Test) (%)**	**Örüntü tanıma yüzdesi (Eğitim) (%)**
29	ÖVK – 1	90,9	91,0
30	ÖVK – 2	85,3	85,3
31	ÖVK – 3	84,9	85,2
32	ÖVK – 4	84,6	85,0
33	ÖVK – 5	90,6	91,0
34	ÖVK – 6	84,9	85,4
35	ÖVK – 7	85,8	85,9
36	ÖVK – 8	85,2	85,4
	ORTALAMA	86,5	86,8

BÖLÜM ALTI
SONUÇLAR

Şekil 5.1'de verilen eş zamanlı örüntüler belirlenirken Hong vd. (2019) gerçekleştirdiği çalışmadan faydalanılmıştır. Hong vd. (2019), derin öğrenme yöntemlerinden biri olan evrişimli sinir ağı (convolutional neural network) ve kendi kendini düzenleyen haritalar modellerini kullanarak 7 temel ve 9 eş zamanlı olmak üzere toplamda 16 kontrol kartı örüntüsünün tanınmasını gerçekleştirmişlerdir. Ardından aşağı-yukarı artan trend ve aşağı-yukarı ani kayma örüntüleri ile 4 eş zamanlı kontrol kartı örüntüsü daha türetmişlerdir. Türetilen bu 4 eş zamanlı kontrol kartı örüntüsünü, 16 örüntüye dahil ederek toplam örüntü sayısını arttırmışlardır. Böylelikle, 7 temel ve 13 eş zamanlı olmak üzere toplamda 20 kontrol kartı örüntüsünün tanınmasını sağlamışlardır. Her iki örüntü tanıma çalışmasında da evrişimli sinir ağı modelleri ile en iyi sonucu elde etmişlerdir.

Bu tez çalışmasında da Hong vd. (2019) gerçekleştirdiği çalışmada kullanılan 16 kontrol kartı örüntüsü kullanılmıştır. Hong vd. (2019) gerçekleştirdiği çalışmada yer alan 20 kontrol kartı örüntüsü ile de bu çalışma gerçekleştirilmiş fakat ÖVK modeli ile verimli bir sonuç elde edilememiştir. İlgili eğitim algoritmaları ile eğitilen ÇKA modelleri için elde edilen sonuçlar ise Tablo 6.1'de verilmiştir. Ayrıca 20 kontrol kartı örüntüsünü tanımak için eğitilen YSA modellerinde, tez kapsamında gerçekleştirilen çalışma ile aynı eğitim parametreleri kullanılmıştır. Bu tezin kapsamının ÖVK modelini içermesinden dolayı 20 kontrol kartı örüntüsü için sadece sonuçlar paylaşılacak olup gerçekleştirilen çalışma detaylandırılmayacaktır. Dolayısıyla 16 kontrol kartı örüntüsü ile çalışmaya devam edilecektir.

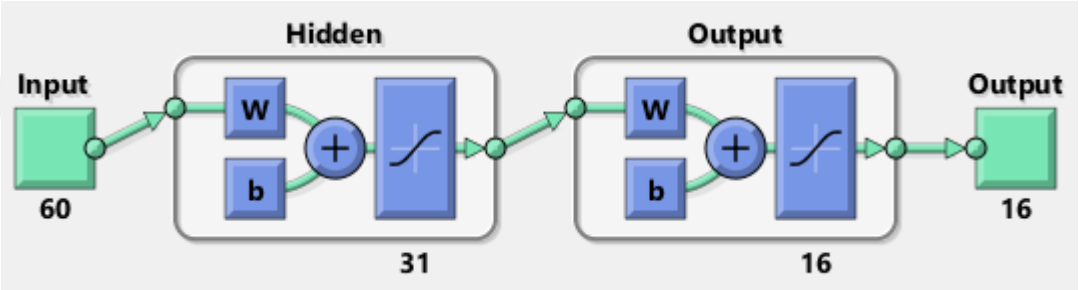
Tablo 6.1 7 temel ve 13 eş zamanlı olmak üzere 20 adet kontrol kartı örüntüsü için eğitilen ÇKA modellerinden elde edilen sonuçlar

20 adet kontrol kartı örüntüsü için simülasyon yoluyla oluşturulan veri setlerinin boyutları	**Eğitim veri seti:** 20 örüntünün her biri için 2500 örnek olmak üzere toplam 50000 örnek ve 60 gözlem verisinden (50000x60) oluşmaktadır.		**Test veri seti:** 20 örüntünün her biri için 625 örnek olmak üzere toplam 12500 örnek ve 60 gözlem verisinden (12500x60) oluşmaktadır.			
ÇKA modeli için MATLAB® ortamındaki ilgili eğitim algoritması	**TRAINGDX**		**TRAINLM**		**TRAINBR**	
Gizli katmandaki nöron sayısı	**Örüntü tanıma yüzdesi (Test) (%)**	**Örüntü tanıma yüzdesi (Eğitim) (%)**	**Örüntü tanıma yüzdesi (Test) (%)**	**Örüntü tanıma yüzdesi (Eğitim) (%)**	**Örüntü tanıma yüzdesi (Test) (%)**	**Örüntü tanıma yüzdesi (Eğitim) (%)**
29	88,5	88,7	88,2	88,5	92,5	92,9
30	87,5	87,9	88,5	89,0	90,8	91,2
31	89,1	89,3	89,8	90,0	92,1	92,7
32	88,6	88,7	90,2	90,6	94,1	94,5
33	84,6	84,8	90,0	89,9	90,5	90,7
34	88,9	89,0	92,7	93,2	91,5	92,0
35	85,3	85,5	88,1	88,4	90,9	91,3
36	85,2	85,2	86,5	86,6	92,4	93,0
ORTALAMA	87,2	87,4	89,3	89,5	91,9	92,3

16 kontrol kartı örüntüsü için ÇKA ve ÖVK modellerinden elde edilen sonuçlar karşılaştırılırken hem test hem de eğitim veri setleri ile elde edilmiş olan örüntü tanıma yüzdeleri incelenmiştir. Ayrıca ÇKA ve ÖVK ağları ile elde edilen en iyi modeller için hata matrisi grafikleri kullanılarak; duyarlılık, kesinlik ve F1 performans metrikleri hesaplanarak incelenmiştir. Performans metriklerinin hesaplanması, hem MATLAB® ortamında ağların eğitiminin ardından gerçekleştirilmiş olup hem de R programlama dili kullanılarak ayrıca gerçekleştirilmiştir.

Buradan yola çıkarak, ÇKA modeli için Tablo 5.3, 5.4 ve 5.5 incelendiğinde; Traingdx, trainlm ve trainbr eğitim algoritmalarına sahip modellerden, test ve eğitim

veri setleriyle elde edilen en iyi örüntü tanıma oranlarına sahip ağ modellerinin sırasıyla TRAINGDX – 7, TRAINLM – 6 ve TRAINBR – 3 isimli modeller olduğu görülmektedir. Bu ağ modelleri incelendiğinde, özellikle trainbr eğitim algoritması ile eğitilen ve gizli katmanında 31 nöron bulunduran TRAINBR – 3 isimli modelin, test veri seti ile %98,9 ve eğitim veri seti ile %99,6 örüntü tanıma oranlarını elde ettiği görülmektedir. Bu oranlar, ÖVK modelleri de dahil olmak üzere bu çalışmada elde edilen en yüksek örüntü tanıma oranlarıdır. Şekil 6.1'de MATLAB® ortamından elde edilen, TRAINBR – 3 modeli için ağ mimarisi diyagramı verilmiştir.



Şekil 6.1 TRAINBR – 3 modeli için MATLAB® ortamından elde edilen ağ mimarisi diyagramı

Şekil 6.1'de verilen diyagram incelendiğinde, TRAINBR – 3 modelinin gizli ve çıktı katmanlarında bulunan hiperbolik tanjant (tansig) aktivasyon fonksiyonlarının grafikleri görülmektedir.

MATLAB® ortamında elde edilen hata matrisi grafikleri için genel olarak, satırlar gerçek örüntü çıktılarını ve sütunlar ise hedef örüntü çıktılarını belirtmektedir. Gerçek pozitifler (köşegen matris) yeşil renkle ifade edilirken, yanlış pozitifler veya yanlış negatifler kırmızı renk ile ifade edilir. Hata matrisi grafiğinde, son satır ve son sütunun kesiştiği hücrede ilgili model için örüntü tanıma oranı (doğruluk performans metriği) değeri bulunmaktadır. Öte yandan, hata matrisi grafiğinde en sondaki sütunda bulunan oranlar her örüntü için duyarlılık performans metriğinin değerlerini belirtmektedir. En sondaki satırda belirtilen oranlar ise her örüntü için kesinlik performans metriğinin değerlerini vermektedir (The MathWorks Inc., 2023).

İlgili veri setine göre (test veya eğitim) modelin genel duyarlılık performans metriği, her örüntü için verilmiş olan ve hata matrisinin son sütununda bulunan duyarlılık oranlarının aritmetik ortalaması alınarak elde edilmiştir:

$$Duyarlılık_{Test} = \frac{Duyarlılık_{Test_1} + \cdots + Duyarlılık_{Test_{16}}}{16} \quad (6.1)$$

$$Duyarlılık_{Eğitim} = \frac{Duyarlılık_{Eğitim_1} + \cdots + Duyarlılık_{Eğitim_{16}}}{16} \quad (6.2)$$

Genel duyarlılık metriğine benzer olarak genel kesinlik metriğinin değeri de en alt satırda bulunan kesinlik oranlarının aritmetik ortalaması alınarak elde edilmiştir:

$$Kesinlik_{Test} = \frac{Kesinlik_{Test_1} + \cdots + Kesinlik_{Test_{16}}}{16} \quad (6.3)$$

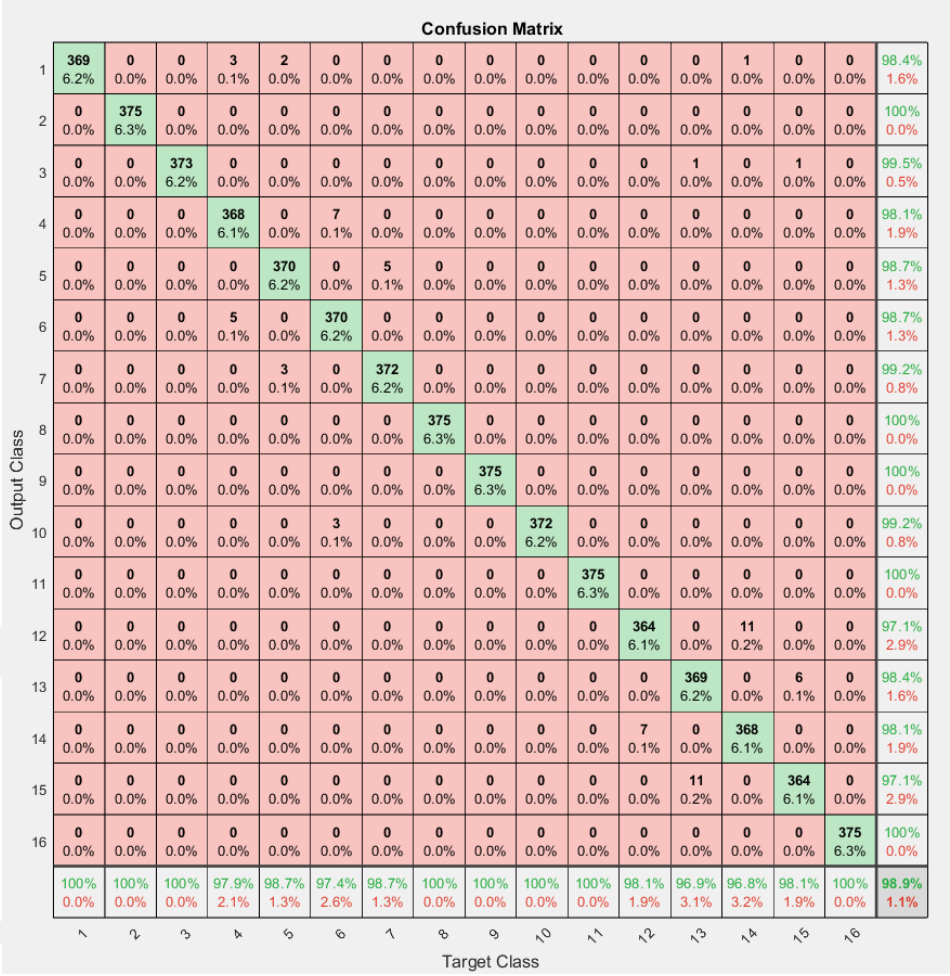
$$Kesinlik_{Eğitim} = \frac{Kesinlik_{Eğitim_1} + \cdots + Kesinlik_{Eğitim_{16}}}{16} \quad (6.4)$$

Genel F1 metriği için ise ilk olarak her örüntü açısından hata matrisi grafiğinde verilen kesinlik ve duyarlılık metriklerinin, Tablo 4.4'te verilen F1 metriği formülüne göre hesaplanması gerçekleştirilmiştir. Ardından her örüntü için elde edilen F1 oranlarının aritmetik ortalaması alınarak genel F1 değeri elde edilmiştir:

$$F1_{Test} = \frac{\left(2x\frac{Duyarlılık_{Test_1}\ x\ Kesinlik_{Test_1}}{Duyarlılık_{Test_1} + Kesinlik_{Test_1}}\right) + \cdots + \left(2x\frac{Duyarlılık_{Test_{16}}\ x\ Kesinlik_{Test_{16}}}{Duyarlılık_{Test_{16}} + Kesinlik_{Test_{16}}}\right)}{16} \quad (6.5)$$

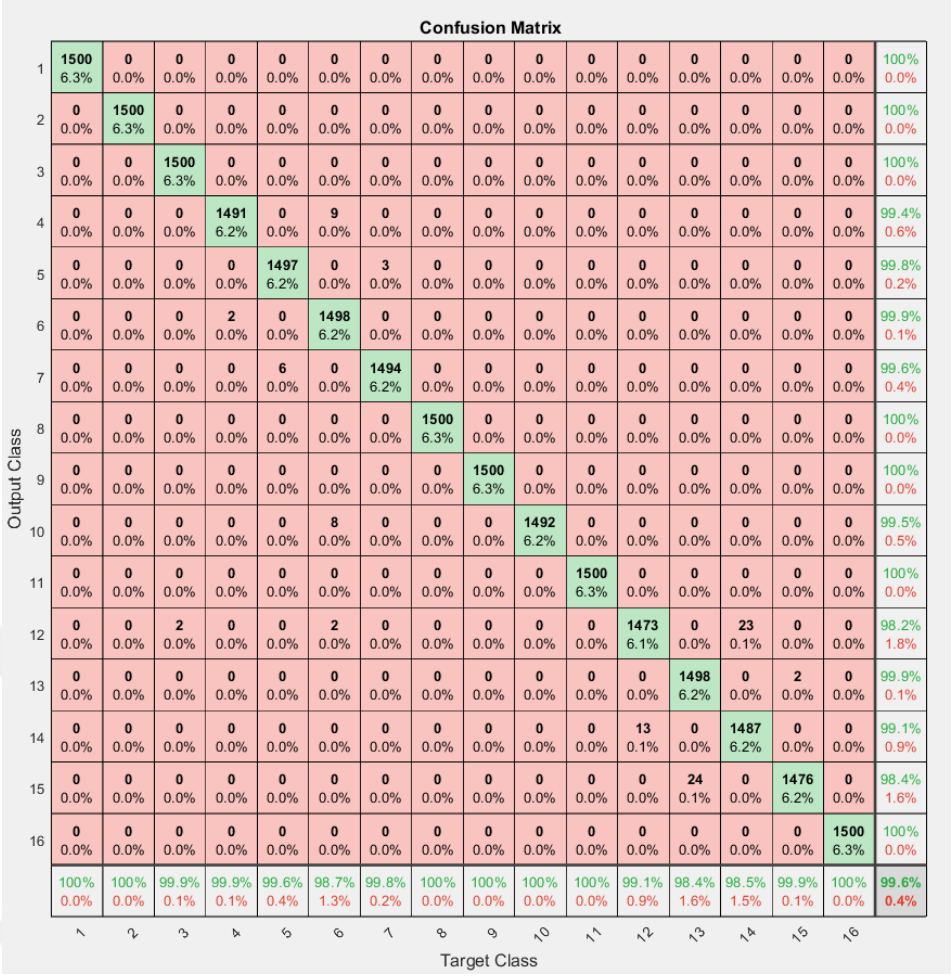
$$F1_{Eğitim} = \frac{\left(2x\frac{Duyarlılık_{Eğitim_1}\ x\ Kesinlik_{Eğitim_1}}{Duyarlılık_{Eğitim_1} + Kesinlik_{Eğitim_1}}\right) + \cdots + \left(2x\frac{Duyarlılık_{Eğitim_{16}}\ x\ Kesinlik_{Eğitim_{16}}}{Duyarlılık_{Eğitim_{16}} + Kesinlik_{Eğitim_{16}}}\right)}{16} \quad (6.6)$$

Şekil 6.2 ve 6.3'te TRAINBR – 3 modeli ile sırasıyla test ve eğitim veri setleri kullanılarak elde edilen hata matrisi grafikleri verilmiştir.



Şekil 6.2 Test veri seti için MATLAB® ortamında TRAINBR – 3 modeli ile elde edilen hata matrisi grafiği (Toplam 6000 örnek, her örüntü için 375 örnek bulunmaktadır.)

Şekil 6.2 incelendiğinde, TRAINBR – 3 modelinin daha önce görmediği gözlemleri içeren test veri seti ile elde edilen hata matrisinin grafiği görülmektedir. Hata matrisi grafiğinden görüldüğü üzere bu model için Tablo 5.5'te de verilmiş olan %98,9 örüntü tanıma oranı, son satır ve son sütunun kesiştiği hücrede bulunmaktadır. Bu model için test veri seti ile genel duyarlılık performans metriği ($Duyarlılık_{Test}$) %98,90 olarak elde edilmiştir. Duyarlılık metriğine benzer olarak genel kesinlik metriğinin ($Kesinlik_{Test}$) değeri de %98,91 olarak elde edilmiştir. Son olarak genel F1 metriği ($F1_{Test}$) %98,90 olarak hesaplanmıştır.



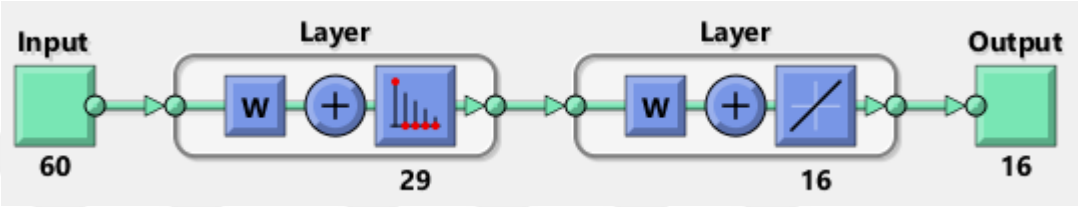
Şekil 6.3 Eğitim veri seti için MATLAB® ortamında TRAINBR – 3 modeli ile elde edilen hata matrisi grafiği (Toplam 24000 örnek, her örüntü için 1500 örnek bulunmaktadır.)

Şekil 6.3'te verilen hata matrisi grafiğinden yola çıkarak, eğitim veri seti ile elde edilen örüntü tahminleri ve hedef örüntü çıktılarına göre TRAINBR – 3 modeli için sırasıyla örüntü tanıma oranı, $Duyarlılık_{Eğitim}$, $Kesinlik_{Eğitim}$ ve $F1Skoru_{Eğitim}$ metriklerinin hepsi %99,6 olarak hesaplanmıştır.

TRAINBR – 3 YSA modeli için hem eğitim hem de test veri setlerine göre elde edilen performans metriklerinin tutarlı olduğu görülmektedir. Test veri setine göre elde edilen genel performans metriği oranları birbirlerine çok yakındır. Bunun yanı sıra eğitim veri seti ile elde edilen genel performans metriği oranlarının birbirine eşit olması bu modelin tüm örüntüler için etkili ve tutarlı tahminler yapabildiği anlamına gelmektedir. Ayrıca, bu model için elde edilen bütün genel performans metriği

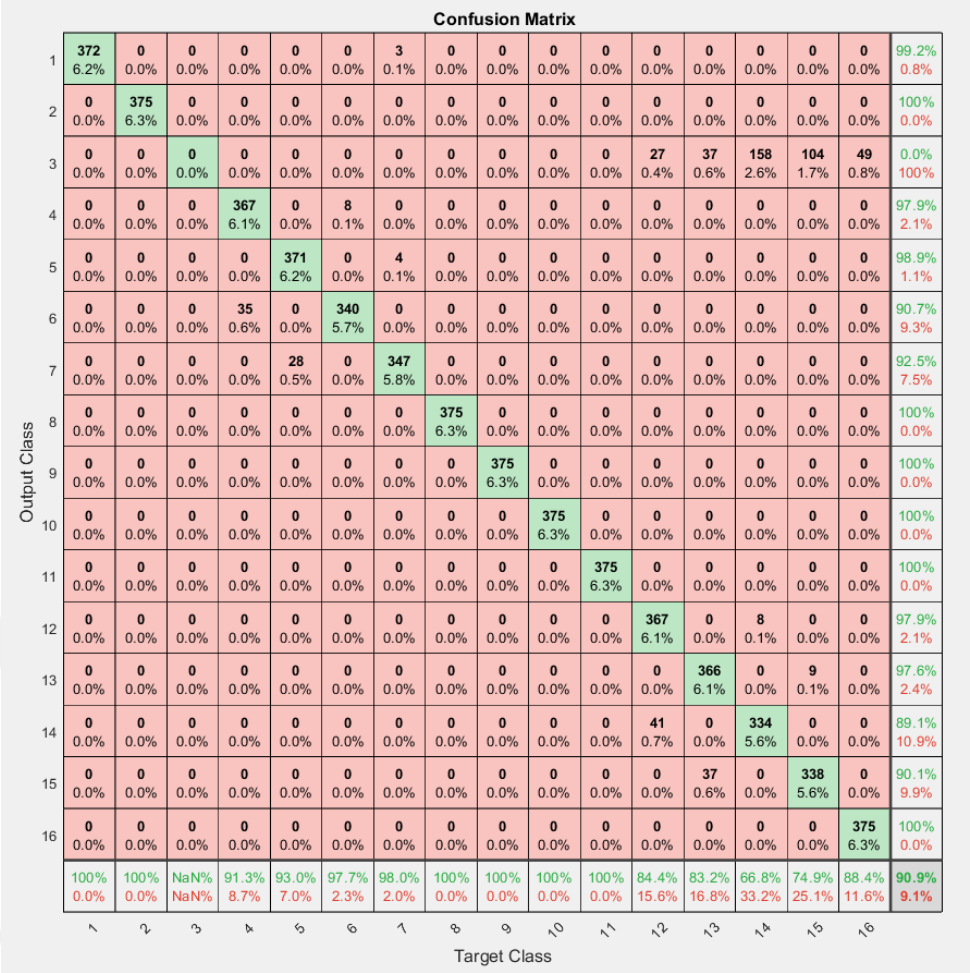
değerleri gayet yüksek değerlerdir ve bu durum modelin çok iyi bir örüntü tanıma performansına sahip olduğunun bir göstergesidir.

Öte yandan ÖVK modeli için Tablo 5.6 incelendiğinde ise en iyi örüntü tanıma oranını veren ağ modelinin, Kohonen katmanında 29 nöron bulunduran ÖVK – 1 isimli model olduğu görülmektedir. Şekil 6.4'te ÖVK – 1 modeli için MATLAB® ortamından elde edilen ağ mimarisi diyagramı verilmiştir.



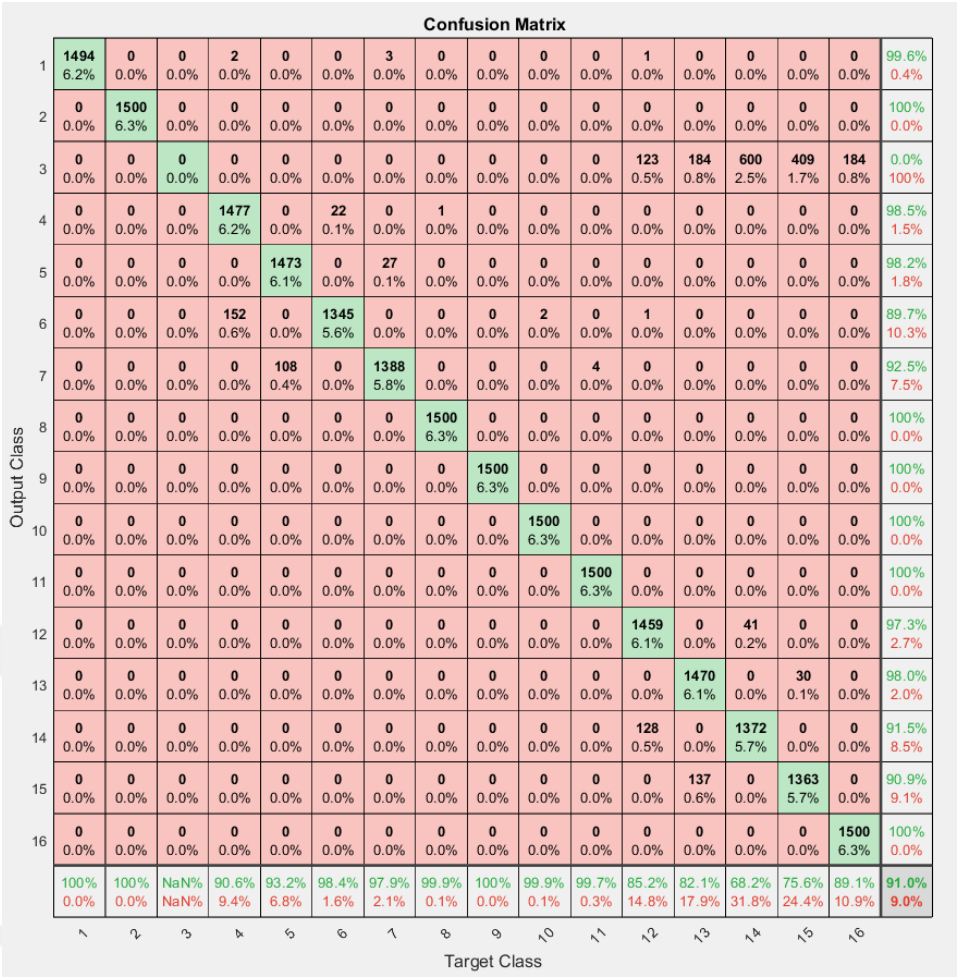
Şekil 6.4 ÖVK – 1 modeli için MATLAB® ortamından elde edilen ağ mimarisi diyagramı

Şekil 6.4'te verilen diyagram incelendiğinde, 29 nöronun bulunduğu katman olan Kohonen katmanında kullanılan compet aktivasyon fonksiyonu ve çıktı katmanında kullanılan purelin aktivasyon fonksiyonunun grafikleri görülmektedir. Şekil 6.5 ve 6.6'da ise ÖVK – 1 modeli ile sırasıyla test ve eğitim veri setleri kullanılarak elde edilen hata matrisi grafikleri verilmiştir.



Şekil 6.5 Test veri seti için MATLAB® ortamında ÖVK – 1 modeli ile elde edilen hata matrisi grafiği (Toplam 6000 örnek, her örüntü için 375 örnek bulunmaktadır.)

Test veri seti için elde edilen hata matrisi grafiğinden ilgili performans metrikleri; $Duyarlılık_{Test} = \%90{,}87$, $Kesinlik_{Test} = \%91{,}85$ ve $F1Skoru_{Test} = \%94{,}11$ şeklinde elde edilmiştir. Bu hesaplamalar sırasında model tarafından 3. örüntü kapsamında herhangi bir doğru örüntü tanıma işlemi gerçekleştirilemediği için duyarlılık ve kesinlik değerleri 3. örüntü için hesaplamalar sırasında 0 olarak belirlenmiştir. Öte yandan hata matrisinde yine 3. örüntü, herhangi bir gerçek ve yanlış pozitif değere sahip olmadığı için 3. örüntünün kesinlik değerinde $\frac{0}{0}$ belirsizliği ortaya çıkmaktadır ve bundan dolayı kesinlik değeri 3. örüntü için hata matrisi grafiğinde tanımsız olarak ifade edilen "NaN" ile belirtilmektedir.



Şekil 6.6 Eğitim veri seti için MATLAB® ortamında ÖVK – 1 modeli ile elde edilen hata matrisi grafiği (Toplam 24000 örnek, her örüntü için 1500 örnek bulunmaktadır.)

Test veri setinde görülen, ÖVK – 1 modelinin 3. örüntüyü tanıyamama durumu, eğitim veri seti ile elde edilen hata matrisinde de mevcuttur. Yine 3. örüntü için duyarlılık ve kesinlik değerleri 0 olarak belirlenerek ilgili performans metrikleri; $Duyarlılık_{Eğitim} = \%91$ $Kesinlik_{Eğitim} = \%91{,}98$ ve $F1Skoru_{Eğitim} = \%94{,}24$ şeklinde elde edilmiştir.

ÖVK – 1 modeli için hem eğitim hem de test veri setleriyle elde edilen ilgili performans metrikleri tutarlıdır. Fakat hata matrisi grafikleri her iki veri seti için de incelendiğinde, bu modelin 3. örüntü olan sistematik örüntüye sahip örnekleri; 12., 13., 14., 15. veya 16. örüntüler olarak tanıdığı görülmektedir. Tablo 5.2'den bu örüntüler incelendiğinde hepsinin eş zamanlı örüntüler oldukları ve yine hepsinin sistematik

örüntüyü içerdiği görülmektedir. Dolayısıyla ilgili performans metrikleri ve hata matrisi grafiği incelendiğinde, ÖVK – 1 modelinin; sistematik örüntü dışındaki örüntüleri düzgün bir şekilde tanıyabildiği söylenebilir.

Tablo 5.3, 5.4, 5.5 ve 5.6'da verilmiş olan ortalama örüntü oranları ile Tablo 6.2 oluşturulmuştur.

Tablo 6.2 Eğitim algoritmalarına göre test ve eğitim veri setleri için elde edilen ortalama örüntü tanıma oranları

Yapay Sinir Ağı Modeli	**Eğitim Algoritması**	**Test Veri Seti için Ortalama Örüntü Tanıma Oranı (%)**	**Eğitim Veri Seti için Ortalama Örüntü Tanıma Oranı (%)**
Çok Katmanlı Algılayıcı	TRAINBR	94,8	95,4
	TRAINLM	94,5	95,0
	TRAINGDX	93,6	94,0
Öğrenmeli Vektör Kuantizasyon	LEARNLV1	86,5	86,8

Tablo 6.2'den ÇKA modelleri için verilmiş olan ortalama örüntü tanıma yüzdeleri incelendiğinde; trainbr eğitim algoritması kullanılarak eğitilen ÇKA modellerinin, test veri seti için ortalama %94,8 ve eğitim veri seti için ortalama %95,4 olacak şekilde ortalama örüntü tanıma oranlarına sahip olduğu görülmektedir. Bu çalışma için bu değerler en yüksek ortalama örüntü tanıma oranlarıdır. Bundan dolayı trainbr eğitim algoritmasıyla eğitimi gerçekleştirilen ÇKA modellerinin bu çalışma kapsamında en iyi YSA modelleri olduğu söylenebilir. İkinci YSA modelleri ise test ve eğitim veri setleri için sırasıyla %94,5 – %95 örüntü tanıma ortalamalarına sahip olan trainlm eğitim algoritması ile eğitimi gerçekleştirilen ÇKA modelleridir. Trainlm eğitim algoritması, MATLAB® ortamında ÇKA ağ modeli için kullanılan varsayılan eğitim algoritmasıdır. Bundan dolayı literatürde özellikle temel örüntülerin tanınmasını

içeren çalışmalarda çoğunlukla ön plana çıkmaktadır. Öte yandan bu çalışma için trainlm ve trainbr eğitim algoritmaları ile birbirine çok yakın ortalama örüntü tanıma oranları elde edilmiştir. Dolayısıyla bu durum, trainlm eğitim algoritmasının eş zamanlı örüntüler için de iyi bir örüntü tanıma performansı sunduğunun bir göstergesidir. ÇKA modelleri için son olarak, traingdx eğitim algoritması ile eğitilen ÇKA modellerinden elde edilen ortalama örüntü tanıma oranlarının %93,6 – %94 olduğu görülmektedir. Traingdx eğitim algoritması ile eğitilen ÇKA modellerinin de genel olarak iyi bir performans sunduğu söylenebilir fakat trainbr ve trainlm eğitim algoritmalarına göre daha düşük örüntü tanıma oranlarına sahip olduğu görülmektedir. Öte yandan eğitimler sırasında, traingdx eğitim algoritması ile eğitilen ÇKA modellerinin, eğitim zamanı açısından en kısa sürede eğitilen modeller olduğu görülmüştür.

Son olarak yine Tablo 6.2 incelendiğinde, ÖVK modellerinin %86,5 – %86,8 ortalama örüntü tanıma oranlarına sahip olduğu görülmektedir. ÇKA modellerine göre bu oranların düşük olmasındaki nedenin anlaşılabilmesi için ÖVK modellerinden elde edilen bütün hata matrisleri ayrıca incelenmiştir. Bu hata matrislerinden elde edilen sonuçlarda, ÖVK modellerinin en az bir örüntüyü ve en fazla üç örüntüyü tanıyamadığı görülmüştür. Özellikle ÖVK modellerinden elde edilen hata matrislerinin çoğunda, 3. örüntü olan sistematik örüntünün tanınamadığı görülmüştür. Bazı modeller için ise sistematik örüntüye ek olarak sistematik ve yukarı – aşağı ani kayma örüntülerinden türetilmiş olan 14. ve 15. eş zamanlı örüntülerinin de tanınamadığı görülmüştür. Bu durum ÖVK modellerinin sistematik örüntü oluşturan süreçlerde, ortalamada gerçekleşen küçük dalgalanmaları tespit edemediğinin bir göstergesidir. Dolayısıyla ÖVK modelleri, bu çalışma kapsamında en düşük örüntü tanıma performansına sahiptir. Ayrıca eğitimler sırasında, ÖVK modellerinin eğitim sürelerinin ÇKA modellerinin eğitim sürelerine göre daha uzun sürdüğü görülmüştür. KKÖT literatüründe yapılan çalışmalar incelendiğinde, ÖVK modelinin özellikle temel örüntüleri tanımada çok iyi bir performans gösterdiği ve çalışmalarda sıkça kullanıldığı görülmektedir. Bundan dolayı bu çalışma kapsamında ayrıca ÖVK modellerinin 150 ve 200 epoch değerleri ile de eğitimleri gerçekleştirilmiştir. Fakat 100 epoch değeri ile eğitimi gerçekleştirilen modellere göre daha iyi bir sonuç elde

edilemediği için çalışma kapsamında bu eğitimlerden elde edilen sonuçlar detaylandırılmamıştır.

Bu tez kapsamında yapılan çalışmada ÇKA modelinin, temel örüntülere ek olarak eş zamanlı örüntüleri de başarıyla tanıdığı görülmektedir. Özellikle, trainbr eğitim algoritması ile birlikte ÇKA modelinin KKÖT açısından çok iyi bir performans sunduğu söylenebilir. Hem elde edilen örüntü tanıma oranlarına göre hem de performans metriği değerlerinin tutarlı ve çok iyi sonuçlar vermesi sebebiyle bu çalışma için en iyi YSA modelinin TRAINBR – 3 isimli ağ modeli olduğu söylenebilir. ÖVK modeli için ise kabul edilebilir düzeyde örüntü tanıma oranları ve genel performans metriği değerleri elde edilmiştir. Öte yandan ÖVK modeli için kullanılan parametreler (epoch sayısı hariç), bu çalışma kapsamında MATLAB® ortamındaki varsayılan parametreler olarak belirlenmiştir. Dolayısıyla bu parametreler, eş zamanlı örüntülere göre detaylı bir iyileştirmeyle birlikte kullanılabilirse, ÖVK modeli için de daha iyi örüntü tanıma oranları elde edilebilir. Sonuç olarak, bu çalışma kapsamında trainbr eğitim algoritması ile eğitilmiş ve gizli katmanında 31 nöron bulunan bir ÇKA modeli, 7 temel kontrol kartı örüntüsüne ek olarak 9 eş zamanlı kontrol kartı örüntüsü için de en iyi örüntü tanıma performansını sunmaktadır.

KAYNAKLAR

Abhishek, L. (2020). Optical Character Recognition using Ensemble of SVM, MLP and Extra Trees Classifier. *2020 International Conference for Emerging Technology (INCET)*, 1-4. https://doi.org/10.1109/INCET49848.2020.9154050

Akın, H. B. (2017). Kalite kontrol tekniklerinin uygulanmasında bilgisayar programlarının kullanılması. *Anadolu Bil Meslek Yüksekokulu Dergisi*, *47*, 1-19.

Ali, A. (2022). *Making And Prioritizing Managerial Decisions Using Fuzzy Logic And Artificial Neural Networks* [Yüksek Lisans Tezi]. İstanbul Sabahattin Zaim Üniversitesi Lisansüstü Eğitim Enstitüsü.

Anderson, D., & Mcneill, G. (1992). *Artificial Neural Networks Technology*.

Aslam, M., Saghir, A., & Ahmad, L. (2021). *Introduction to Statistical Process Control*. Wiley.

ASTM Committee E-11 on Quality and Statistics. (1990). *Manual on Presentation of Data and Control Chart Analysis* (7. Basım). ASTM.

Atalay, O., & Kılıç, Ö. (2015). Balık Kılçığı Yöntemi ile Mobil Vinç Kazası Olası Nedenlerinin İncelenmesi. *Çukurova University Journal of the Faculty of Engineering and Architecture*, *30*(1), 73-78.

Avgikou, A., Vosniakos, G.-C., & Benardos, P. (2006). *SPC pattern recognition using LVQ neural nets*. https://www.researchgate.net/publication/263156840

Awadalla, M. H. A., & Sadek, M. A. (2012). Spiking Neural Network - Based Control Chart Pattern Recognition. *Alexandria Engineering Journal*, *51*(1), 27-35. https://doi.org/10.1016/j.aej.2012.07.004

Best, M., & Neuhauser, D. (2006). Walter A Shewhart, 1924, and the Hawthorne factory. *Quality & safety in health care*, *15*(2), 142-143. https://doi.org/10.1136/qshc.2006.018093

Bre, F., Gimenez, J. M., & Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, *158*, 1429-1441. https://doi.org/10.1016/j.enbuild.2017.11.045

Cheng, C.-S. (1997). A neural network approach for the analysis of control chart patterns. *International Journal of Production Research*, *35*(3), 667-697. https://doi.org/10.1080/002075497195650

Corson, N., & Aziz-Alaoui, M. A. (2009). *Asymptotic Dynamics Of The Slow-Fast Hindmarsh-Rose Neuronal System*.

Çayıroğlu, İ. (2019). *İleri Algoritma Analizi*.

Çetin, O., & Temurtaş, F. (2018). Öğrenmeli Vektör Kuantalama ile Beyin Bilgisayar Arayüzü Üzerine Bir Çalışma. *Sakarya University Journal of Computer and Information Sciences*, *1*(2), 1-7.

Çuhadar, M., & Kayacan, C. (2005). Yapay Sinir Ağları Kullanılarak Konaklama İşletmelerinde Doluluk Oranı Tahmini: Türkiye'deki Konaklama İşletmeleri Üzerine Bir Deneme. *Anatolia: Turizm Araştırmaları Dergisi*, *16*(1), 24-30.

Değerli, Z. (2006). *Toplam Kalite Yönetiminde İstatistiksel Süreç Kontrolünün Önemi ve Bir İşletme Uygulaması* [Yüksek Lisans Tezi]. Akdeniz Üniversitesi Sosyal Bilimler Enstitüsü.

Demir, T. (2018). *Yapay Sinir Ağları Tabanli Hibrit Modellerle Yük Tahmini* [Yüksek Lisans Tezi]. Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü.

Demir, Y. (2008). *Shewhart, CUSUM ve EWMA Kontrol Grafiklerinin Makine Sanayine Uygulanması* [Yüksek Lisans Tezi]. Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Zootekni Anabilim Dalı.

Demir, Y., & Mirtagioğlu, H. (2016). Shewhart CUSUM ve EWMA Kontrol Grafiklerinin Makine Sanayine Uygulanması. *YYU TAR BİL DERG*, *26*(2), 254-265.

Demuth, H., & Beale, M. (1992). *Neural Network Toolbox For Use with MATLAB User's Guide*. www.mathworks.com

Dengizler Kayaalp, İ. (2007). *Konfeksiyon İşletmelerinde Kalitenin İyileştirilmesi Amacıyla İstatistiksel Kalite Kontrol Yöntemlerinin Kullanılması Üzerine Bir Araştırma* [Doktora Tezi]. Ege Üniversitesi Fen Bilimleri Enstitüsü.

Ersoy, E., & Karal, Ö. (2012). Yapay Sinir Ağları ve İnsan Beyni. *İnsan ve Toplum Bilimleri Araştırma Dergisi*, *1*(2).

Ertuğrul, İ., & Karakaşoğlu, N. (2006). Kalite Kontrolde Örneklem Büyüklüğünün Değişken Olması Durumunda p Kontrol Şemalarının Oluşturulması. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, *5*(10), 65-80.

Gauri, S. K. (2010). Control chart pattern recognition using feature-based learning vector quantization. *The International Journal of Advanced Manufacturing Technology*, *48*(9-12), 1061-1073. https://doi.org/10.1007/s00170-009-2354-7

Gauri, S. K., & Chakraborty, S. (2008). Improved recognition of control chart patterns using artificial neural networks. *The International Journal of Advanced Manufacturing Technology*, *36*(11-12), 1191-1201. https://doi.org/10.1007/s00170-006-0925-4

Ghasemiyeh, R., Moghdani, R., & Sana, S. S. (2017). A Hybrid Artificial Neural Network with Metaheuristic Algorithms for Predicting Stock Price. *Cybernetics and Systems*, *48*(4), 365-392. https://doi.org/10.1080/01969722.2017.1285162

Goncalves, V., Maria, K., & da Silv, A. B. F. (2013). Applications of Artificial Neural Networks in Chemical Problems. Içinde *Artificial Neural Networks - Architectures and Applications*. InTech. https://doi.org/10.5772/51275

Graupe, D. (2007). *Principles Of Artificial Neural Networks* (2. Basım).

Guh, R. S., & Tannock, J. D. T. (1999). A Neural Network Approach to Characterize Pattern Parameters in Process Control Charts. *Journal of Intelligent Manufacturing*, *10*, 449-462.

Günay, M. (2018). *Makine Öğrenmesiyle Müşteri Kayıplarının Tahmini* [Yüksek Lisans Tezi]. İstanbul Üniversitesi Fen Bilimleri Enstitüsü.

Hameed, A. A. H. (2017). *Robust Adaptive Learning Approach of Artificial Neural Networks* [Doktora Tezi]. Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

Haykin, S. (2008). *Neural Networks and Learning Machines* (3. Basım). Pearson Prentice Hall.

Hong, Z., Li, Y., & Zeng, Z. (2019). Convolutional Neural Network for Control Chart Patterns Recognition. *Proceedings of the 3rd International Conference on Computer Science and Application Engineering*, 1-9. https://doi.org/10.1145/3331453.3360974

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining*, *15*(4), 531-538. https://doi.org/10.1002/sam.11583

Juran, J. M., & Godfrey, A. B. (1998). *Juran's Quality Handbook* (5. Basım). McGraw-Hill.

Kaya, İ., Oktay, S., & Engin, O. (2005). Kalite Kontrol Problemlerinin Çözümünde Yapay Sinir Ağlarinin Kullanimi. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *21*(1-2), 92-107.

Kaymakamtorunları Deniz, F. (2019). *Performance Comparison of Nonparametric CUSUM and EWMA Control Charts Under Mean Shifts* [Yüksek Lisans Tezi]. Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü.

Koltan Yılmaz, Ş. (2014). *Doğrusal Vektör Kuantizasyon Modeli Kullanılarak Yapay Sinir Ağlarıyla Kontrol Şemalarında Örüntü Tanıma: Hazır Beton Üreten Bir İşletmede Uygulama* [Doktora Tezi]. İnönü Üniversitesi Sosyal Bilimler Enstitüsü.

Krüger, F. (2016). *Activity, Context, and Plan Recognition with Computational Causal Behaviour Models* [Doktora Tezi, University of Rostock]. https://www.researchgate.net/publication/314116591

Kurt, R. (2018). *İstatistiksel Süreç Kontrolünde Shewhart, CUSUM ve EWMA Kontrol Kartları ile Yapay Sinir Ağlarının Bütünleşik Kullanımı: Bir Orman Endüstri İşletmesinde Uygulama* [Doktora Tezi]. Bartın Üniversitesi Fen Bilimleri Enstitüsü.

Lever, J., Krzywinski, M., & Altman, N. (2016). Points of significance: Classification evaluation. *Nature Methods*, *13*(8), 603-604.

Lipton, Z. C., Elkan, C., & Narayanaswamy, B. (2014). *Thresholding Classifiers to Maximize F1 Score*. http://arxiv.org/abs/1402.1892

Magar, V. M., & Shinde, V. B. (2014). Application of 7 Quality Control (7 QC) Tools for Continuous Improvement of Manufacturing Processes. *International Journal of Engineering Research and General Science*, *2*(4). www.ijergs.org

Masood, I., & Hassan, A. (2008). Application of Full Factorial Experiment in Designing an ANN-based Control Chart Pattern Recognizer. *Proceedings of the International Graduate on Engineering and Science (IGCES'08)*.

Megahed, F. M., & Jones-Farmer, L. A. (2015). Statistical Perspectives on "Big Data". Içinde *Frontiers in Statistical Quality Control 11* (C. 11, ss. 29-47). Springer International Publishing. https://doi.org/10.1007/978-3-319-12355-4_3

Montgomery, D. C. (2013). *Introduction to Statistical Quality Control* (7. Basım). Wiley.

Nova, D., & Estévez, P. A. (2014). A review of learning vector quantization classifiers. Içinde *Neural Computing and Applications* (C. 25, Sayı 3-4, ss. 511-524). Springer London. https://doi.org/10.1007/s00521-013-1535-3

Olgun, M. O. (2011). *Kalite Kontrol Grafiklerinde Örüntü Tanıma* [Yüksek Lisans Tezi]. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü.

Oztemel, E., & Gursev, S. (2020). Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, *31*(1), 127-182. https://doi.org/10.1007/s10845-018-1433-8

Özcan, T. (2020). *İstatistiksel Kalite Kontrolü*. http://auzefkitap.istanbul.edu.tr/kitap/endustrimuhlt_ue/istatistikselkalitekontrolu.pdf

Öztemel, E. (2003). *Yapay Sinir Ağları*. Papatya Yayıncılık.

Pelàez, J. R., & Castillo Piqueira, J. R. (2008). Biological clues for up-to-date artificial neurons. Içinde *Computational Intelligence: for Engineering and Manufacturing* (ss. 131-146). Springer US. https://doi.org/10.1007/0-387-37452-3_6

Pham, D. T., & Oztemel, E. (1993). Control Chart Pattern Recognition Using Combinations of Multi-Layer Perceptrons and Learning-Vector-Quantization Neural Networks. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, *207*(2), 113-118. https://doi.org/10.1243/PIME_PROC_1993_207_325_02

Pham, D. T., & Oztemel, E. (1994). Control chart pattern recognition using learning vector quantization networks. *International Journal of Production Research*, *32*(3), 721-729. https://doi.org/10.1080/00207549408956963

Pham, D. T., & Wani, M. A. (1997). Feature-based control chart pattern recognition. *International Journal of Production Research*, *35*(7), 1875-1890. https://doi.org/10.1080/002075497194967

Qiu, P. (2014). *Introduction to Statistical Process Control*. Chapman & Hall/CRC.

Qiu, P. (2020). Big Data? Statistical Process Control Can Help! *American Statistician*, *74*(4), 329-344. https://doi.org/10.1080/00031305.2019.1700163

Reis, M. S., & Gins, G. (2017). Industrial Process Monitoring in the Big Data/Industry 4.0 Era: from Detection, to Diagnosis, to Prognosis. *Processes*, *5*(3). https://doi.org/10.3390/pr5030035

Sağıroğlu, Ş., Beşdok, E., & Erler, M. (2000). Control Chart Pattern Recognition Using Artificial Neural Networks. *Turkish Journal of Electrical Engineering*, *8*(2), 137-147.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427-437. https://doi.org/10.1016/j.ipm.2009.03.002

Suzuki, K. (2011). *Artificial Neural Networks - Methodological Advances and Biomedical Applications*. InTech.

Swamidass, P. M. (2000). *Encyclopedia of Production and Manufacturing Management*. Kluwer Academic Publishers.

Şenol, D. (2022). *Yapay Sinir Ağlari ile Kripto Para Tahmini* [Yüksek Lisans Tezi]. Sakarya Üniversitesi Fen Bilimleri Enstitüsü.

Temelatma, B. (2021). *İlaç Sektöründe Ürün Değerlendirmelerinde Kontrol Grafiklerinin ve Proses Yeterlilik Analizlerinin Kullanılması* [Yüksek Lisans Tezi]. İstanbul Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü.

The MathWorks Inc. (2023). *Plot classification confusion matrix - MATLAB plotconfusion*. https://www.mathworks.com/help/deeplearning/ref/plotconfusion.html

Uztemur, M. (2019). *Yapay Sinir Ağlari ile Optik Karakter Tanıma*. İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü.

Verma, K., & Kumar Singh, P. (2015). An Insight to Soft Computing based Defect Prediction Techniques in Software. *International Journal of Modern Education and Computer Science*, *7*(9), 52-58. https://doi.org/10.5815/ijmecs.2015.09.07

Vining, G. (2009). Technical Advice: Phase I and Phase II Control Charts. *Quality Engineering*, *21*(4), 478-479. https://doi.org/10.1080/08982110903185736

Western Electric Company. (1956). *Statistical Quality Control Handbook* (1. Basım). Western Electric Co.

Wicks, A. M., & Roethlein, C. J. (2009). A Satisfaction-Based Definition of Quality. Içinde *Journal of Business & Economic Studies* (C. 15, Sayı 1).

Wikipedia. (2023). *Learning vector quantization*. https://en.wikipedia.org/wiki/Learning_vector_quantization

Woodall, W. H., & Montgomery, D. C. (1999). Research Issues and Ideas in Statistical Process Control. *Journal of Quality Technology*, *31*(4), 376-386. https://doi.org/10.1080/00224065.1999.11979944

Yıldırım, S. (2006). *Toplam Kalite Yönetiminde İstatistiksel Süreç Kontrolü'nün Önemi ve Bir Demirdöküm İşletmesinde Yapılan Uygulama* [Yüksek Lisans Tezi]. Eskişehir Osmangazi Üniversitesi Sosyal Bilimler Enstitüsü.

Yılmaz, B. (2019). *Karma Proses Kontrol Grafikleri* [Yüksek Lisans Tezi]. Gazi Üniversitesi Fen Bilimleri Enstitüsü.

Yılmaz, Ş. (2009). *Yüksek Kaliteli Süreçlerde Uygulanabilir Kontrol Grafikleri ve Ekonomik Tasarımına Bir Yaklaşım* [Yüksek Lisans Tezi]. Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü.

EKLER

EK 1: $\overline{X}$ kontrol kartı için ilgili katsayılar ve örneklem boyutuna göre değerleri (Demir, 2008)

n	d_2	d_3	D_1	D_2	D_3	D_4
2	1.128	0.853	0	3.686	0	3.267
3	1.693	0.888	0	4.358	0	2.575
4	2.059	0.880	0	4.698	0	2.282
5	2.326	0.864	0	4.918	0	2.115
6	2.534	0.848	0	5.078	0	2.004
7	2.704	0.833	0.205	5.203	0.076	1.924
8	2.847	0.820	0.387	5.307	0.136	1.864
9	2.970	0.808	0.546	5.391	0.184	1.816
10	3.078	0.797	0.687	5.469	0.223	1.777
11	3.173	0.787	0.812	5.534	0.256	1.744
12	3.258	0.778	0.924	5.592	0.284	1.716
13	3.336	0.770	1.026	5.646	0.308	1.692
14	3.407	0.762	1.121	5.693	0.329	1.671
15	3.472	0.755	1.207	5.737	0.348	1.652
16	3.532	0.749	1.285	5.779	0.364	1.636
17	3.588	0.743	1.359	5.817	0.379	1.621
18	3.640	0.738	1.426	5.854	0.392	1.608
19	3.689	0.733	1.490	5.888	0.404	1.596
20	3.735	0.729	1.548	5.922	0.414	1.586
21	3.778	0.724	1.606	5.950	0.425	1.575
22	3.819	0.720	1.659	5.979	0.434	1.566
23	3.858	0.716	1.710	6.006	0.443	1.555
24	3.895	0.712	1.759	6.031	0.452	1.548
25	3.931	0.709	1.804	6.058	0.459	1.541
>25	$\overline{R}/\sigma$	σ_R/σ	$d_2 - 3d_3$	$d_2 + 3d_3$	$1 - 3d_3/d$	$1 + 3d_3/d$

EK 2: *R* kontrol kartı için ilgili katsayılar ve örneklem boyutuna göre değerleri (Demir, 2008)

n	d_2	d_3	D_1	D_2	D_3	D_4
2	1.128	0.853	0	3.686	0	3.267
3	1.693	0.888	0	4.358	0	2.575
4	2.059	0.880	0	4.698	0	2.282
5	2.326	0.864	0	4.918	0	2.115
6	2.534	0.848	0	5.078	0	2.004
7	2.704	0.833	0.205	5.203	0.076	1.924
8	2.847	0.820	0.387	5.307	0.136	1.864
9	2.970	0.808	0.546	5.391	0.184	1.816
10	3.078	0.797	0.687	5.469	0.223	1.777
11	3.173	0.787	0.812	5.534	0.256	1.744
12	3.258	0.778	0.924	5.592	0.284	1.716
13	3.336	0.770	1.026	5.646	0.308	1.692
14	3.407	0.762	1.121	5.693	0.329	1.671
15	3.472	0.755	1.207	5.737	0.348	1.652
16	3.532	0.749	1.285	5.779	0.364	1.636
17	3.588	0.743	1.359	5.817	0.379	1.621
18	3.640	0.738	1.426	5.854	0.392	1.608
19	3.689	0.733	1.490	5.888	0.404	1.596
20	3.735	0.729	1.548	5.922	0.414	1.586
21	3.778	0.724	1.606	5.950	0.425	1.575
22	3.819	0.720	1.659	5.979	0.434	1.566
23	3.858	0.716	1.710	6.006	0.443	1.555
24	3.895	0.712	1.759	6.031	0.452	1.548
25	3.931	0.709	1.804	6.058	0.459	1.541
>25	$\bar{R}/\sigma$	σ_R/σ	$d_2 - 3d_3$	$d_2 + 3d_3$	$1 - 3d_3/d$	$1 + 3d_3/d$

EK 3: *S* kontrol kartı için ilgili katsayılar ve örneklem boyutuna göre değerleri (Demir, 2008)

n	c_4	B_3	B_4	B_5	B_6
2	0.7979	0	3.267	0	2.606
3	0.8862	0	2.568	0	2.276
4	0.9213	0	2.266	0	2.088
5	0.9400	0	2.089	0	1.964
6	0.9515	0.030	1.970	0.029	1.876
7	0.9594	0.118	1.882	0.113	1.806
8	0.9650	0.185	1.815	0.179	1.751
9	0.9693	0.239	1.761	0.232	1.707
10	0.9727	0.284	1.716	0.276	1.669
11	0.9754	0.321	1.679	0.313	1.637
12	0.9776	0.354	1.646	0.346	1.610
13	0.9794	0.382	1.618	0.374	1.585
14	0.9810	0.406	1.594	0.399	1.563
15	0.9823	0.428	1.572	0.421	1.544
16	0.9835	0.448	1.552	0.440	1.526
17	0.9845	0.466	1.534	0.458	1.511
18	0.9854	0.482	1.518	0.475	1.496
19	0.9862	0.497	1.503	0.490	1.483
20	0.9869	0.510	1.490	0.504	1.470
21	0.9876	0.523	1.477	0.516	1.459
22	0.9882	0.534	1.466	0.528	1.448
23	0.9887	0.545	1.455	0.539	1.438
24	0.9892	0.555	1.445	0.549	1.429
25	0.9896	0.565	1.435	0.559	1.420
>25	$\frac{4(n-1)}{4n-3}$	$1-\frac{3}{c_4\sqrt{2n-2}}$	$1+\frac{3}{c_4\sqrt{2n-2}}$	$c_4-\frac{3}{c_4\sqrt{2n-2}}$	$c_4+\frac{3}{c_4\sqrt{2n-2}}$

EK 4: ÇKA modeli ile gerçekleştirilen eğitimler için kullanılan MATLAB® kodları

```
% İlgili kodlara github üzerinden de ulaşılabilir:
% https://github.com/canerkonuk/ControlChartPatternRecognitionWithANN

clc;
clear;
clear all;
close all;


% Eğitim ve test için gerekli olan girdi ve hedef veri setlerinin okunması
input=xlsread('train_24000x60');
target=xlsread('train_output_labels');
testInputs=xlsread('test_6000x60');
testTargets=xlsread('test_output_labels');


% İlgili veri setlerinin ağa girdisi yapılabilmesi için gerçekleştirilen
% transpoz işlemi
input=input';
target=target';
testInputs=testInputs';
testTargets=testTargets';


% İlgili girdi veri setlerinin -1 ve 1 aralığında normalize edilip
% ölçeklendirilmesi
input=normalize(input, 'range', [-1 1]);
testInputs=normalize(testInputs, 'range', [-1 1]);


trainInputs = input;
trainTargets = target;


%Eğitim algoritmasının belirlenmesi
%'trainlm' - 'trangdx' - 'trainbr' eğitim algoritmaları kullanıldı.
trainFcn = 'trainbr';


for i=29:36

% Modelde bulunan gizli katmandaki nöron sayısının ve eğitim fonksiyonunun
% belirlenmesi
hiddenLayerSize = i;
net = patternnet(hiddenLayerSize, trainFcn);


% Tanımlanan çok katmanlı algılayıcı modeline girdilerin ve çıktıların
% eklenmesi
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};


% Eğitim, test ve validasyon için verilerin bölünmesi
% (Test verisi ayrıca simule edildiği için burada herhangi bir bölme işlemi
% gerçekleştirilmedi.)
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 100/100;
net.divideParam.valRatio = 0/100;
net.divideParam.testRatio = 0/100;


% Eğitim parametrelerinin ağa tanıtılması
net.trainParam.show=25;
net.trainParam.lr=0.1; % learning rate - öğrenme oranı
net.trainParam.epochs=2500;
net.trainParam.goal=0.01;
net.trainParam.max_fail=500; % maximum fail - maksimum hata
```

```
net.trainParam.mc=0.5; % momentum constant - momentum katsayısı
net.trainParam.min_grad=0;


% Performans fonksiyonunun belirlenmesi işlemi
net.performFcn = 'mse';


% Sırasıyla girdi ve çıktı katmanlarında kullanılacak olan aktivasyon
% fonksiyonları
net.layers{1}.transferFcn='tansig';
net.layers{2}.transferFcn='tansig';


% YSA için kullanılabilen plot fonksiyonları
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotconfusion', 'plotroc'};


% Ağın eğitimi
[net,tr] = train(net,trainInputs,trainTargets);


% Elde edilen çıktılar ile ilgili metriklerin test veri seti ile elde
% edilmesi
testPredictedTargets = net(testInputs);
trainPredictedTargets = net(trainInputs);


% Ağın mimarisinin görüntülenmesi
view(net)


% Öğrenme algoritmasına göre modelin eğitim süresi değiştiği için modelleri
% kayıt işlemini gerçekleştiriyorum.
% Böylelikle eğitilen modeller sonradan kullanılabilir.
fname = sprintf("cka_model_%d_noron_"+trainFcn, i);
save(string(fname), 'net')



% Test ve eğitim veriseti için ilgili hata matrislerinin oluşturulması ve
% kayıt edilmesi
plotconfusion(testPredictedTargets, testTargets)
saveas(gcf,"cka_model_"+i+"_noron_test_confusion_matrix_"+trainFcn)

plotconfusion(trainPredictedTargets, trainTargets)
saveas(gcf,"cka_model_"+i+"_noron_train_confusion_matrix_"+trainFcn)


% Test veri seti için performans metriklerinin hesaplanması
[~, testPredictedLabels] = max(testPredictedTargets);
[~, testTrueLabels] = max(testTargets);
testNumCorrect = sum(testPredictedLabels == testTrueLabels);
testNumTotal = numel(testTrueLabels);


% Test veri setine göre örüntü tanıma oranı
testAccuracy = testNumCorrect / testNumTotal;


% Test veri setine göre kesinlik, duyarlılık ve f1 skoru değerlerinin
% hesaplanması:
testNumClasses = size(testTargets, 1);
testPrecision = zeros(testNumClasses, 1);
testRecall = zeros(testNumClasses, 1);
testF1Score = zeros(testNumClasses, 1);
```

```
for a = 1:testNumClasses
    testTruePositive = sum(testPredictedLabels == a & testTrueLabels == a);
    testFalsePositive = sum(testPredictedLabels == a & testTrueLabels ~= a);
    testFalseNegative = sum(testPredictedLabels ~= a & testTrueLabels == a);

    testPrecision(a) = testTruePositive / (testTruePositive + testFalsePositive);
    testRecall(a) = testTruePositive / (testTruePositive + testFalseNegative);
    testF1Score(a) = 2 * testPrecision(a) * testRecall(a) / (testPrecision(a) +
testRecall(a));
end


% Test veri setine göre genel performans metriklerinin hesaplanması
testMacroPrecision = mean(testPrecision, "omitnan");
testMacroRecall = mean(testRecall, "omitnan");
testMacroF1Score = mean(testF1Score, "omitnan");


%#####################################################################


% Eğitim veri seti için performans metriklerinin hesaplanması
[~, trainPredictedLabels] = max(trainPredictedTargets);
[~, trainTrueLabels] = max(trainTargets);
trainNumCorrect = sum(trainPredictedLabels == trainTrueLabels);
trainNumTotal = numel(trainTrueLabels);


% Eğitim veri setine göre örüntü tanıma oranı
trainAccuracy = trainNumCorrect / trainNumTotal;


% Eğitim veri setine göre kesinlik, duyarlılık ve f1 skoru değerlerinin
% hesaplanması:
trainNumClasses = size(trainTargets, 1);
trainPrecision = zeros(trainNumClasses, 1);
trainRecall = zeros(trainNumClasses, 1);
trainF1Score = zeros(trainNumClasses, 1);

for b = 1:testNumClasses
    trainTruePositive = sum(trainPredictedLabels == b & trainTrueLabels == b);
    trainFalsePositive = sum(trainPredictedLabels == b & trainTrueLabels ~= b);
    trainFalseNegative = sum(trainPredictedLabels ~= b & trainTrueLabels == b);

    trainPrecision(b) = trainTruePositive / (trainTruePositive + trainFalsePositive);
    trainRecall(b) = trainTruePositive / (trainTruePositive + trainFalseNegative);
    trainF1Score(b) = 2 * trainPrecision(b) * trainRecall(b) / (trainPrecision(b) +
trainRecall(b));
end


% Eğitim veri setine göre genel performans metriklerinin hesaplanması
trainMacroPrecision = mean(trainPrecision, "omitnan");
trainMacroRecall = mean(trainRecall, "omitnan");
trainMacroF1Score = mean(trainF1Score, "omitnan");


% Elde edilen sonuçların sonradan incelenebilmesi için nöron sayısı ve
% eğitim fonksiyonunun ismine göre oluşturulan bir .txt dosyasına
% yazdırılması
fname = sprintf("MLPTrainAndTestMetrics%dNeuron"+trainFcn+".txt", i);
fid = fopen(fname,'wt');

fprintf(fid,'Test veri seti için:');
fprintf(fid,'Model Adi: cka_model_%.2f%%\n_noron', i);
fprintf(fid,'Accuracy: %.2f%%\n', testAccuracy * 100);
fprintf(fid,'Precision: %.2f%%\n', testMacroPrecision * 100);
fprintf(fid,'Recall: %.2f%%\n', testMacroRecall * 100);
fprintf(fid,'F1 Score: %.2f%%\n', testMacroF1Score * 100);
fprintf(fid,'---------------------------------------');
fprintf(fid,'Eğitim veri seti için:');
```

```
fprintf(fid,'Model Adi: cka_model_%.2f%%\n_noron', i);
fprintf(fid,'Accuracy: %.2f%%\n', trainAccuracy * 100);
fprintf(fid,'Precision: %.2f%%\n', trainMacroPrecision * 100);
fprintf(fid,'Recall: %.2f%%\n', trainMacroRecall * 100);
fprintf(fid,'F1 Score: %.2f%%\n', trainMacroF1Score * 100);
fprintf(fid,'====================================');

fclose(fid);

end
```

EK 5: ÖVK modeli ile gerçekleştirilen eğitimler için kullanılan MATLAB® kodları

```
% İlgili kodlara github üzerinden de ulaşılabilir:
% https://github.com/canerkonuk/ControlChartPatternRecognitionWithANN

clc;
clear;
clear all;
close all;


% Eğitim ve test için gerekli olan girdi ve hedef veri setlerinin okunması
input=xlsread('train_24000x60');
target=xlsread('train_output_labels');
testInputs=xlsread('test_6000x60');
testTargets=xlsread('test_output_labels');


% İlgili veri setlerinin ağa girdisi yapılabilmesi için gerçekleştirilen
% transpoz işlemi
input=input';
target=target';
testInputs=testInputs';
testTargets=testTargets';


% İlgili girdi veri setlerinin -1 ve 1 aralığında normalize edilip
% ölçeklendirilmesi
input=normalize(input, 'range', [-1 1]);
testInputs=normalize(testInputs, 'range', [-1 1]);


trainInputs = input;
trainTargets = target;


net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 100/100;
net.divideParam.valRatio = 0/100;
net.divideParam.testRatio = 0/100;


for i=29:36

% Modelde bulunan Kohonen katmanındaki nöron sayısının ve ilgili
% ağ parametrelerinin belirlenmesi
kohonenLayerSize = i;
net = lvqnet(kohonenLayerSize,0.01); %0.01 learning rate - öğrenme oranı
net.trainParam.epochs = 100;
net.performFcn = 'mse'; %performance function - performans fonksiyonu


% Ağın eğitimi
net = train(net,trainInputs,trainTargets);


% Elde edilen çıktılar ile ilgili metriklerin test veri seti ile elde
% edilmesi
testPredictedTargets = net(testInputs);
trainPredictedTargets = net(trainInputs);


% Ağın mimarisinin görüntülenmesi
view(net)


% Modelin eğitim süresi uzun sürebildiği için kayıt işlemini
% gerçekleştiriyorum. Böylelikle eğitilen modeller sonradan kullanılabilir.
fname = sprintf("lvq_model_%d_noron_100", i);
save(string(fname), 'net')
```

```
% Test ve eğitim veriseti için ilgili hata matrislerinin oluşturulması ve
% kayıt edilmesi
plotconfusion(testPredictedTargets, testTargets)
saveas(gcf,"lvq_model_"+i+"_noron_test_confusion_matrix_100")

plotconfusion(trainPredictedTargets, trainTargets)
saveas(gcf,"lvq_model_"+i+"_noron_train_confusion_matrix_100")


% Test veri seti için performans metriklerinin hesaplanması
[~, testPredictedLabels] = max(testPredictedTargets);
[~, testTrueLabels] = max(testTargets);
testNumCorrect = sum(testPredictedLabels == testTrueLabels);
testNumTotal = numel(testTrueLabels);


% Test veri setine göre örüntü tanıma oranı
testAccuracy = testNumCorrect / testNumTotal;


% Test veri setine göre kesinlik, duyarlılık ve f1 skoru değerlerinin
% hesaplanması:
testNumClasses = size(testTargets, 1);
testPrecision = zeros(testNumClasses, 1);
testRecall = zeros(testNumClasses, 1);
testF1Score = zeros(testNumClasses, 1);


for a = 1:testNumClasses
    testTruePositive = sum(testPredictedLabels == a & testTrueLabels == a);
    testFalsePositive = sum(testPredictedLabels == a & testTrueLabels ~= a);
    testFalseNegative = sum(testPredictedLabels ~= a & testTrueLabels == a);

    testPrecision(a) = testTruePositive / (testTruePositive + testFalsePositive);
    testRecall(a) = testTruePositive / (testTruePositive + testFalseNegative);
    testF1Score(a) = 2 * testPrecision(a) * testRecall(a) / (testPrecision(a) +
testRecall(a));
end


% Test veri setine göre genel performans metriklerinin hesaplanması
testMacroPrecision = mean(testPrecision, "omitnan");
testMacroRecall = mean(testRecall, "omitnan");
testMacroF1Score = mean(testF1Score, "omitnan");


%#################################################################


% Eğitim veri seti için performans metriklerinin hesaplanması
[~, trainPredictedLabels] = max(trainPredictedTargets);
[~, trainTrueLabels] = max(trainTargets);
trainNumCorrect = sum(trainPredictedLabels == trainTrueLabels);
trainNumTotal = numel(trainTrueLabels);


% Eğitim veri setine göre örüntü tanıma oranı
trainAccuracy = trainNumCorrect / trainNumTotal;


% Eğitim veri setine göre kesinlik, duyarlılık ve f1 skoru değerlerinin
% hesaplanması:
trainNumClasses = size(trainTargets, 1);
trainPrecision = zeros(trainNumClasses, 1);
trainRecall = zeros(trainNumClasses, 1);
trainF1Score = zeros(trainNumClasses, 1);
```

```
for b = 1:testNumClasses
    trainTruePositive = sum(trainPredictedLabels == b & trainTrueLabels == b);
    trainFalsePositive = sum(trainPredictedLabels == b & trainTrueLabels ~= b);
    trainFalseNegative = sum(trainPredictedLabels ~= b & trainTrueLabels == b);

    trainPrecision(b) = trainTruePositive / (trainTruePositive + trainFalsePositive);
    trainRecall(b) = trainTruePositive / (trainTruePositive + trainFalseNegative);
    trainF1Score(b) = 2 * trainPrecision(b) * trainRecall(b) / (trainPrecision(b) +
trainRecall(b));
end


% Eğitim veri setine göre genel performans metriklerinin hesaplanması
trainMacroPrecision = mean(trainPrecision, "omitnan");
trainMacroRecall = mean(trainRecall, "omitnan");
trainMacroF1Score = mean(trainF1Score, "omitnan");


% Elde edilen sonuçların sonradan incelenebilmesi için nöron sayısı ve
% eğitim fonksiyonunun ismine göre oluşturulan bir .txt dosyasına
% yazdırılması
fname = sprintf("LVQTrainAndTestMetrics%dNeuron100.txt", i);
fid = fopen(fname,'wt');

fprintf(fid,'Test veri seti için:');
fprintf(fid,'Model Adi: lvq_model_%.2f%%\n_noron', i);
fprintf(fid,'Accuracy: %.2f%%\n', testAccuracy * 100);
fprintf(fid,'Precision: %.2f%%\n', testMacroPrecision * 100);
fprintf(fid,'Recall: %.2f%%\n', testMacroRecall * 100);
fprintf(fid,'F1 Score: %.2f%%\n', testMacroF1Score * 100);
fprintf(fid,'--------------------------------------');
fprintf(fid,'Eğitim veri seti için:');
fprintf(fid,'Model Adi: lvq_model_%.2f%%\n_noron', i);
fprintf(fid,'Accuracy: %.2f%%\n', trainAccuracy * 100);
fprintf(fid,'Precision: %.2f%%\n', trainMacroPrecision * 100);
fprintf(fid,'Recall: %.2f%%\n', trainMacroRecall * 100);
fprintf(fid,'F1 Score: %.2f%%\n', trainMacroF1Score * 100);
fprintf(fid,'======================================');

fclose(fid);

end
```