
GAZİ ÜNİVERSİTESİ
Gazi
1926

BİRLEŞİK YAPAY SİNİR AĞLARI İLE DİNAMİK SANAL İMALAT HÜCRESİ OLUŞTURMA PROBLEMİ VE BİR UYGULAMA

Burcu ADIGÜZEL

YÜKSEK LİSANS
İMALAT MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2022

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Burcu ADIGÜZEL

15/06/2022

BİRLEŞİK YAPAY SİNİR AĞLARI İLE DİNAMİK SANAL İMALAT HÜCRESİ OLUŞTURMA PROBLEMİ VE BİR UYGULAMA

(Yüksek Lisans Tezi)

Burcu ADIGÜZEL

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2022

ÖZET

Hücresel İmalat, Yalın Üretim yöntemi olarak uzun yıllardır uygulanmaktadır. Ancak hücresel imalat sistemini kurmanın ve yürütmenin zorlukları bulunmaktadır. Bu nedenle yeni yaklaşımlar geliştirilmiştir. Sanal Hücresel İmalat Sistemi (VCMS-Virtual Cellular Manufacturing Systems) ve Dinamik Hücresel İmalat Sistemi (DCMS-Dynamic Cellular Manufacturing Systems) bu yeni yaklaşımlardan ikisidir. Ayrı ayrı uygulanması durumunda her bir sistemin kendi içinde avantajları ve dezavantajları mevcuttur. Günümüzde bu iki sistemin birlikte kullanılmasına yönelik yeni çalışmalar yapılmaktadır. Dinamik Sanal Hücresel İmalat Sistemi (DVCMS-Dynamic Virtual Cellular Manufacturing Systems) bu iki sistemin birlikteliği ile ortaya çıkan yeni bir yöntemdir. Tezgahların fiziksel yerleşimden bağımsız olduğu ve üretim ortamının değişken talepler karşısında yeniden şekillenebildiği bu sistem, imalat hücrelerinin kurulmasında büyük imkânlar sunmaktadır. Bu çalışma kapsamında dinamik sanal imalat hücresi oluşturma problemi ele alınmış ve çözümünde birleşik yapay sinir ağları kullanılmıştır. Yapay sinir ağlarının birleştirilmesinde model karmaşıklığını azaltmak ve genel sistemi daha kolay anlamak ve geliştirmek amacıyla modüler yaklaşım uygulanmıştır. Birleşik ağlara dahil edilen her bir yapay sinir ağı öncesinde farklı ağ mimarileriyle çalıştırılmış, performans karşılaştırması sonucunda en iyi değerleri sunan ağlar seçilerek birleştirilmiştir. Çözüm aracı olarak MATLAB R2021a programı Neural Network (NN) ve App Designer modüllerinin kullanıldığı çalışma temel olarak 3 aşamadan oluşmaktadır. İlk aşamada NN Fitting uygulamasında parça, tezgâh ve operasyon bilgilerini kullanarak parçaları tezgâhlara atayan Çok Katmanlı İleri Beslemeli bir yapay sinir ağı kurulmuştur. İkinci aşamada NN Clustering uygulamasında imalat hücreleri oluşturmak üzere Rekabetçi Yapay Sinir Ağı ile tezgahlar kümelenmiştir. Son aşamada ise denetimli ve denetimsiz iki yapay sinir ağı önce operasyonları tezgâhlara, sonra tezgâhları hücrelere atayacak şekilde App Designer modülünde geliştirilen DVCM App programında birleştirilmiştir. Geliştirilen program uygulama işletmesinin imalat atölyesinde kullanıma sunulmuş, çizelgelemenin kolaylaşmasını ve riskli durumların öngörülebilmesini sağlamıştır.

Bilim Kodu : 91438

Anahtar Kelimeler : Sanal hücresel imalat sistemi, birleşik yapay sinir ağları, talaşlı imalat, imalat hücresi oluşturma, kümeleme

Sayfa Adedi : 83

Danışman : Doç. Dr. Yunus KAYIR

CREATING DYNAMIC VIRTUAL MANUFACTURING CELLS WITH COMBINED ARTIFICIAL NEURAL NETWORKS AND AN APPLICATION

(M. Sc. Thesis)

Burcu ADIGÜZEL

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2022

ABSTRACT

Cellular Manufacturing has been applied as a Lean Manufacturing method for many years, but there are difficulties in establishing and operating the cellular manufacturing system. Therefore, new approaches have been developed. Virtual Cellular Manufacturing System (VCMS) and Dynamic Cellular Manufacturing System (DCMS) are two of these new approaches. If applied separately, each system has its own advantages and disadvantages. Today, new studies are carried out to use these two systems together. Dynamic Virtual Cellular Manufacturing System (DVCMS) is a new method that emerged with the combination of these two systems. This system, where the machines are independent of the physical location and the production environment can be reconfigured in the face of variable demands, offers great possibilities in the establishment of manufacturing cells. In this study, the dynamic virtual manufacturing cell creation problem is discussed and combined artificial neural networks are used in its solution. Modular approach has been applied in combining of artificial neural networks in order to reduce the model complexity and to understand, modify and expand the overall system more easily. Each artificial neural network included in the combined networks was operated with different network architectures beforehand, and the networks offering the best values were selected and combined as a result of the performance comparison. The study, in which the MATLAB R2021a program Neural Network (NN) and App Designer modules are used as a solution tool, basically consists of 3 stages. In the first stage, a Multilayer Feed Forward artificial neural network was established in the NN Fitting application, which assigns the parts to the machines by using the part, machine and operation information. In the second stage, in the NN Clustering application, the machines are clustered with the Competitive Neural Network to form manufacturing cells. In the last stage, two artificial neural networks were combined in the DVCM App program, developed in the App Designer modüle. The developed program has been put into use in the manufacturing workshop of the implementation company and it facilitated scheduling and predicted risky situations.

Science Code : 91438

Key Words : Virtual cellular manufacturing, combined artificial neural networks, cell formation, machining, clustering

Page Number : 83

Supervisor : Doç. Dr. Yunus KAYIR

TEŞEKKÜR

Bu tezin çeşitli aşamalarında yardımlarıyla beni yönlendiren ve değerli tecrübelerinden faydalandığım danışman hocam Sayın Doç. Dr. Yunus KAYIR'a ve yükseköğrenimim süresince yanımda olan kıymetli hocam Sayın Prof. Dr. Abdülkadir GÜLLÜ'ye teşekkürlerimi sunarım. Hayatımın her döneminde her zaman bana destek olan canım ailemi şükranla anıyorum.

İÇİNDEKİLER

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

RESİMLERİN LİSTESİ

Resim **Sayfa**

Resim **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

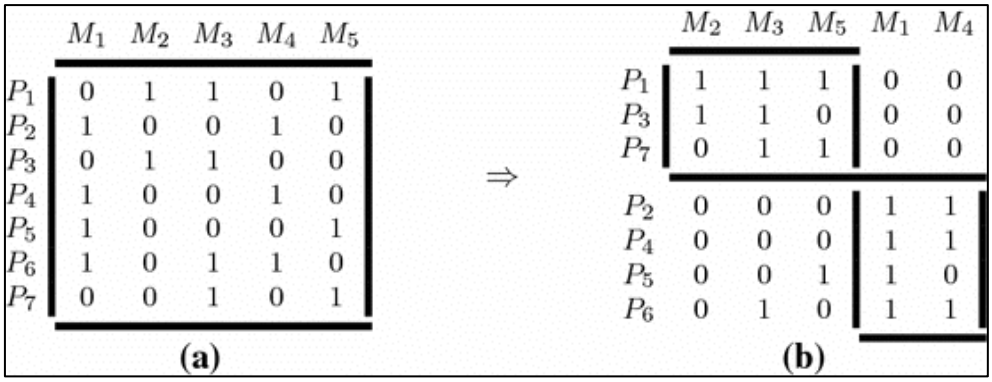
SİMGELER VE KISALTMALAR LİSTESİ

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
ART	Adaptive Resonance Theory (Adaptif Rezonans Teorisi)
CFP	Cell Formation Problem (Hücre Oluşturma Problemi)
CM	Cellular Manufacturing (Hücresel İmalat)
CNN	Competitive Neural Networks (Rekabetçi Sinir Ağları)
DCM	Dynamic Cellular Manufacturing (Dinamik Hücresel İmalat)
DVCM	Dynamic Celluar Virtual Manufacturing (Dinamik Sanal Hücresel İmalat)
GA	Genetik Algoritma
MLF	Multilayer Feed-Forward (Çok Katmanlı İleri Beslemeli)
PFA	Part Flow Analysis (Üretim Akış Analizi)
PSO	Parçacık Sürü Optimizasyonu
SOM	Self Organizing Map (Öz Düzenleyici Harita)
VCM	Virtual Cellular Manufacturing (Sanal Hücresel İmalat)
YSA	Yapay Sinir Ağları

1. GİRİŞ

Hücresel İmalat bir Yalın Üretim yöntemi olarak yaklaşık yarım asırdır uygulanmaktadır. Bu yöntemin temel konusu, imalat hücreleri oluşturmaya yönelik problemleri çözmektir. Literatürde "hücre oluşturma problemi (CFP-Cell Formation Problem)", parça özellikleri, operasyonları ve rotaları ile tezgâh tipleri, kapasiteleri ve diğer kısıtları göz önüne alarak parça aileleri ve tezgâh hücreleri oluşturmak" şeklinde tanımlanmaktadır. Tanımlanan problem, Hücresel İmalatın temel problemidir ve çok kriterli karar verme problemi olarak değerlendirilmektedir [1]. Şekil 1.1'de yer alan örnek bir hücre oluşturma probleminde, 7 parça ve 5 makine için (a), hiçbir kısıt göz önüne alınmadan oluşturulan 2 imalat hücresi için dahi hücre dışında kalan parçalar (b) olabileceği görülmektedir.

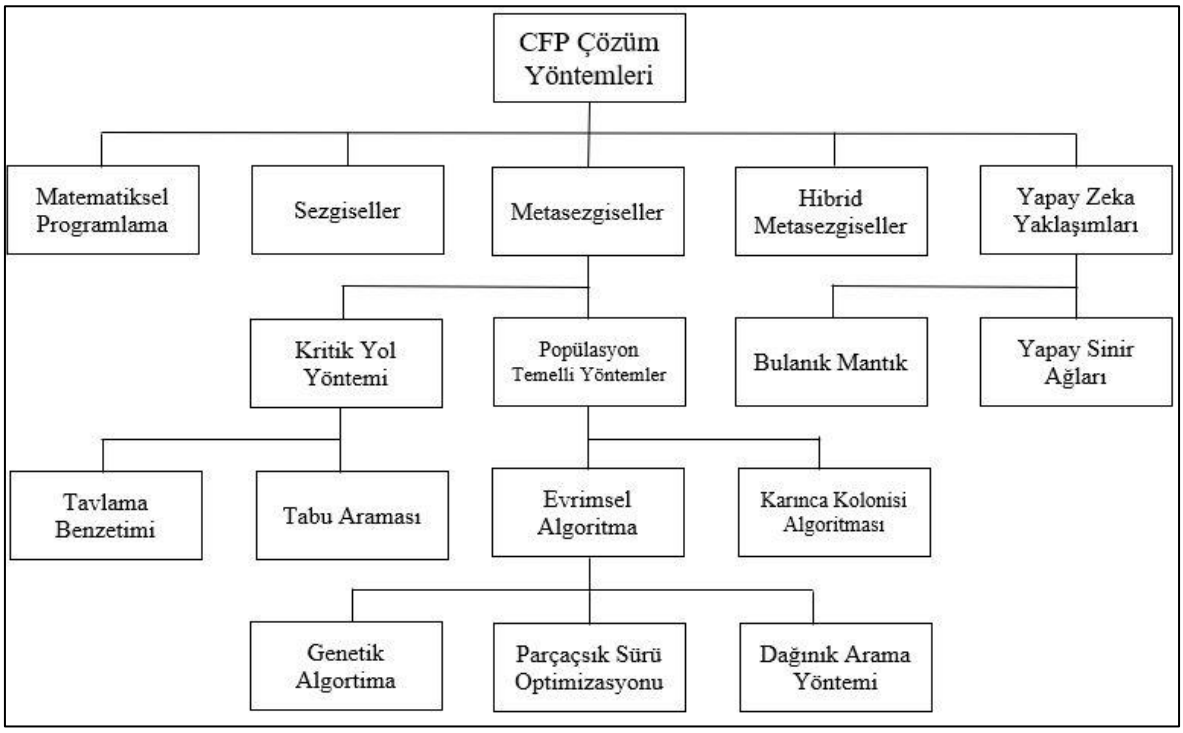


Şekil 1.1. Hücre oluşturma problemi matris görünümü ve çözümü

CFP'nin odaklandığı konular hücre oluşturma, malzeme taşıma, hücre boşlukları, harici elemanlar, darboğaz makineler ve belirsiz taleplerdir. Üzerinde durulan bu konular sebebiyle hücre oluşturma problemini kesin çözüme ulaştıracak bir algoritma henüz bulunamamıştır Problemin zorluğu ve matematiksel algoritmaların yetersiz kalması, çözümü için kullanılan pek çok farklı yaklaşımın doğmasına sebep olmuştur [2]. Bu noktada Uzman Sistemler ve Yapay Sinir Ağları gibi yeni yöntemlere yönelme başlamıştır. Hücre oluşturma probleminin çözümünde genel olarak 5 ana yöntemin kullanıldığı görülmüştür:

1- Matematiksel programlama
2- Sezgiseller
3- Metasezgiseller
4- Hibrid metasezgiseller

5- Yapay zekâ yaklaşımları



Şekil 1.2. Hücre oluşturma probleminde kullanılan yöntemler.

Şekil 1.2'de hücre oluşturma problemin çözümünde kullanılan tüm yöntemler görülmektedir [3]. Literatür taramasında öncelikle klasik hücresel imalat, ardından dinamik ve sanal hücresel imalat sistemlerinde hücre oluşturma problemleri araştırılmıştır. Bu çalışmalarda çözüm yöntemi olarak yapay sinir ağlarını kullanan çalışmalar esas alınmıştır.

Klasik hücresel imalat hücre oluşturma problemini YSA ile ele alan çalışmalar

Literatür taramasının ilk aşamasında klasik hücresel imalat sistemi için hücre oluşturma problemini yapay sinir ağlarıyla ele alan çalışmalar incelenmiştir. Denetimsiz öğrenmeye dayalı yapay sinir ağları ile yapılan bu çalışmalarda SOM (Kohonen Self-Orginizing Maps), ART (Adaptive Resonance Theory), Hopfield, CNN (Competitive Neural Networks) sinir ağları ve bunlardan türetilerek geliştirilmiş yapay sinir ağlarının kullanıldığı görülmüştür.

Venkumar ve Haq, CM hücre yapısındaki harici elemanları en aza indirmeyi hedef alan bir yapılandırılmış ART1 sinir ağı modeli kurmuşlardır. Çalışmada 0-1 parça makine matrisi ART1 sinir ağının girdi veri seti olarak kullanılmıştır. Elde edilen sonuçlar literatürdeki

çalışmalarla karşılaştırılmış ve yapay sinir ağının diğer algoritmalarla eşit veya daha iyi sonuçlar verdiği ortaya koyulmuştur [4]. Mahmoodian vd.'nin Parçacık Sürü Optimizasyonu (PSO) algoritmasını SOM sinir ağı modeli ile destekledikleri çalışmaları, yapay zekayı ve sürü zekasını bir araya getirmektedir. Literatürdeki çalışmalara kıyasla daha iyi sonuç aldıkları çalışmalarını tarımsal üretim faaliyetlerinde bulunan bir firmada gerçek verilerle uygulayarak bir hücresel imalat sistemi kurmuşlardır [5]. Mahmoodian vd.'nin yağtığı bu çalışma, yapay sinir ağları ile optimizasyon algoritmalarının bir arada kullanılması ile hibrid yöntemlere örnek teşkil etmektedir. Liao ve Chen yaptıkları çalışmada parçaların tasarım özelliklerini esas alan girdilerin kullanıldığı bir ART1 sinir ağı modeli geliştirmişlerdir. Kümele yapan bir yapay sinir ağının hata toleransı, öğrenme yeteneği ve hızı ile algoritmalara kıyasla çok daha iyi sonuçlar verebileceğini ortaya koymuşlardır [6]. Benzer şekilde Onwubolu'nun çalışmasında da parçaların tasarım özelliklerini Bilgisayar Destekli Tasarım (CAD)'dan alarak CM hücreleri oluşturan bir SOM sinir ağı modeli geliştirilmiştir. Çalışmada SOM sinir B-rep (sınır gösterimi) halinde parçaları kümelemekte ve buna göre tezgâh, takım ve aparat seçilebilmektedir. Elde edilen verilerin 4 katmalı ileri beslemeli bir yapay sinir ağına sunulmasıyla ise tasarımcıya parçanın üretimi için öngörü sağlanmaktadır [7]. Chattopadhyay vd. SOM sinir ağı modelini kullandıkları çalışmalarında CM için parça-makine hücre formasyonu sağlamışlardır. Çalışma SOM ağı mimarisinin kurulumu ve SOM haritası ölçülerinin belirlenmesi konusunda kriterleri de ortaya koymaktadır [8]. Öztürk vd. hücre oluşturma problemi çözümünde CNN ile elde ettikleri sonuçları genetik algoritma, tabu araması ve tavlama benzetimi yöntemlerinin sonuçlarıyla karşılaştırmış ve 15 örnekten 14'ün de CNN'nin daha iyi sonuç verdigini ortaya koymuşlardır [9]. Venugopal vd. ise hücre oluşturma problemin çözümünde SOM, ART ve CNN ağlarınının uygunluğunu değerlendirmişlerdir. Yapay sinir ağlarının genelleme, uyarlanabilirlik ve doğrusal olmama özellkleri sayesinde hücre oluşturma problemlerinde uygulama açısından pratik olduklarını ifade etmişlerdir [10]. Pandian vd. hücre oluşturma probleminin çözümünde parçaların operasyon sürelerini de göz önüne alarak kurdukları ART sinir ağını, parça işlem sürelerini de dahil ederek geliştirmişlerdir [11]. Benzer şekilde Chen vd. standart ART1 algoritmasını iki şekilde değiştiren alternatif bir algoritma sunmuşlardır. Çalışmalarında orijinal makine-parça matrisinde verilen ikili vektörler yerine iki kutuplu vektörleri kullanan ve performans kriterlerini algoritmaya dahil eden bu yöntemin ART1'den daha iyi sonuçlar verdiğini ortaya koymuşlardır [12]. Guerrero vd. hücre oluşturma probleminin çözümünde iki aşamalı bir yaklaşım uygulamışlardır. Çalışmanın ilk aşamasında parçaları kümeleyen bir SOM ağı kurulmuş, ikinci aşamada doğrusal bir ağ akış modeli ile makineler gruplanmıştır [13].

Kaparthi ve Suresh, Carpenter-Grossberg ağı olarak da bilinen ART sinir ağını kullanarak, büyük veri setlerinin 0-1 matris yapısına getirilmesiyle Üretim Akış Analizi (PFA) yöntemini destekleyen bir kümeleme çalışması gerçekleştirmişlerdir [14]. Safaei vd. (2009), makine kapasitesi ve ürün talebinin "sabit" olarak değerlendirilmemesi gerektiğini ve öngörülemeyen belirsizliklerin zaman içinde hücre yapısını etkileyeceğini öne sürmüşlerdir [15]. Doğru şekilde gruplanmış parçalardan oluşan bir hücrede hazırlık, operasyon ve malzeme taşıma süreleri azalmaktadır [16]. Hücresel imalat uygulamaları yıllar boyunca popülerliğini koruyarak uygulanmaya devam etse de talebin dinamik oluşu ve değişken piyasa koşullarında karşılaşılan belirsizlikler Safaei vd.'nin görüşünü haklı çıkarmıştır.

Dinamik ve sanal hücresel imalat hücre oluşturma problemini YSA ile ele alan çalışmalar

Hücresel imalatın dezavantajları, firmaları ve araştırmacıları yeni yaklaşımlara yönlendirmiş ve bu noktada CM'den yola çıkarak iki farklı üretim felsefesi gündeme gelmiştir: Sanal Hücresel İmalat Sistemi (VCMS-Virtual Cellular Manufacturing Systems) ve Dinamik Hücresel İmalat Sistemi (DCMS-Dynamic Cellular Manufacturing Systems). Hücresel İmalatı tam anlamıyla uygulamayı başarmış olan birinci sınıf üretim firmalarının karşılaştığı yeni sorunlar sebebiyle gündeme gelen bu iki sistem ve türevleri, bilgisayar bilimleri ve bilgi teknolojilerinden faydalanmayı esas alır [17]. Literatür taramasının ikinci aşamasında bu iki sisteme yönelik çalışmalar ele alınmıştır.

Paydar vd. tedarik, üretim ve dağıtım planlama konularını kapsayan dinamik sanal hücre problemi için çift matematiksel model kullanmışlardır. Oluşturdukları model tezgah yüklerini ve müşteri taleplerini de göz önüne almaktadır [18]. Yine Paydar vd., D-VCM problemi için hammadde miktarı gibi bulanık verileri de göz önüne alan bir matematiksel model geliştirilmiştir [19]. Mahdavi vd. her periyotta değişiklik gösteren bir talep yapısı karşısında, makineleri, parçaları ve çalışanları gruplayarak dinamik sanal imalat hücreleri oluşturan bir bulanık hedef programlama üzerine çalışmışlardır. Kurulan modelin en önemli avantajı üretim planlama dönemi boyunca tüm periyotlarda talebi ve parça varyasyonlarını göz önüne almasıdır [20]. Rezazadeh vd. ise yaptıkları çalışmada üretim planlama, dinamik konfigürasyon ve işgücü gereksinimi konularını kapsayan iki bütünleşik VCMS'ni ele alarak PSO algoritması içerisine gömülü bir doğrusal programlama modeli kullanmışlardır [21]. Yine Rezazadeh vd. dinamik koşullar altında sanal imalat hücresi oluşturma problemi çözümü için üretim planlamayı ve iş gücü gereksinimini dikkate alan bir matematiksel model

geliştirmişlerdir [22]. Baykasoğlu ve Görkemli dinamik taleplere cevap veren bir VCM oluşturmak için ajan temelli bir algoritma geliştirmişlerdir. Algoritma önce parça ailelerini oluşturmakta, ardından sanal imalat hücreleri oluşturmakta ve eşzamanlı olarak çizelgelemektedir [23]. Günümüz piyasa koşullarında ürün yaşam döngüsünün çok kısa olduğunu, miktar ve ürün çeşitliliğinin sıklıkla değiştiğini vurgulayan Barve vd., DCMS, VCMS ve D-VCMS gibi hibrid yaklaşımları içeren hücre oluşturma probleminin çözümü için literatürde mevcut yöntemleri sınıflandıran ve karşılaştıran bir çalışma ortaya koymuşlardır. Bu çalışmada hücre oluşturma probleminin çözümünde YSA'nın dinamik ve uyarlanabilir yapısı sebebiyle başarılı olduğunu ancak tüm kısıtları göz önünde bulundurmadığından kapsamlı bir çözüm sunmadığını öne sürmüşlerdir [24].

Literatürün değerlendirilmesi

Çizelge 1.1'de incelenen çalışmalar çizelge halinde görülmektedir. Literatür çalışması sonucunca YSA ile hücre oluşturma probleminin çözümü noktasında çoğunlukla melez yaklaşımlar geliştirildiği görülmüştür. YSA modelleri sıklıkla Genetik Algoritma (GA) ve Parçacık Sürü Algoritması (PSO) gibi optimizasyon yöntemleriyle desteklenmiş ve melez yapılar oluşturulmuştur.

Çizelge 1.1. Literatür özeti

Araştırmacı	Yıl	Problem	Çözüm Yöntemi
Kaparthi ve Suresh	1992	CM	ART yapay sinir ağı
Liao ve Chen	1993	CM	ART yapay sinir ağı
Venugopal	1994	CM	SOM, ART, CNN yapay sinir
Chen vd.	1996	CM	Geliştirilmiş ART yapay sinir ağı
Onwubolu	1999	CM	SOM yapay sinir ağı
Guerrero vd.	2002	CM	SOM yapay sinir ağı ve doğrusal programlama
Venkumar ve Haq	2006	CM	ART yapay sinir ağı
Öztürk vd.	2006	CM	CNN yapay sinir ağı
Pandian vd.	2009	CM	ART yapay sinir ağı
Rezazadeh vd.	2009	VCM	PSO ve doğrusal programlama
Mahdavi vd.	2011	DVCM	Bulanık hedef programlama
Rezazadeh vd.	2011	DVCM	Matematiksel modelleme
Chattopadhyay vd.	2012	CM	SOM yapay sinir ağı
Paydar vd.	2015	DVCM	Matematiksel modelleme
Paydar vd.	2017	DVCM	Matematiksel modelleme
Baykasoğlu vd.	2017	DVCM	Ajan temelli algoritma
Mahmoodian vd.	2017	CM	SOM yapay sinir ağı ve PSO algoritması

İncelenen çalışmalarda, Dinamik Sanal Hücresel İmalat Sistemi'nin (D-VCMS- Dynamic Virtual Cellular Manufacturing Systems), VCMS'nin tezgâhların fiziksel yerleşiminden bağımsız olma avantajı ile DCMS'nin dinamik taleplere karşı yeniden şekillenebilme özelliklerini bir arada barındıran bir yapı olduğu ifade edilmektedir. Literatürde dinamik sanal imalat hücresi oluşturma probleminin çözümünde bulanık hedef programlama, matematiksel modelleme ve ajan temelli algoritma yöntemleri kullanıldığı görülmektedir.

DVCM hücre oluşturma problemi çözümünde YSA'nın tek başına bir çözüm yöntemi olarak ele alınmamış olması, yapılan çalışmanın çözüm yaklaşımı oluşturmaktadır. Yapay sinir ağlarının doğru şekilde kurulması, iyi eğitilmesi ve birleştirilmesi durumunda, literatürde yer alan yöntemlerden daha iyi sonuçlar verdiği bilinmektedir. Hücre oluşturma problemi gibi karmaşık bir problemin çözümünde, kurulmak istenen hücrelerin dinamik ve sanal oluşu için, yapay sinir ağlarının yeni durumlar karşısında sergilediği uyarlanabilme özelliğinden

faydalanılmıştır. Yapay sinir ağlarının gerçek zamanlı olarak öğrenmeye devam edebilmesi, üretim periyodu boyunca dinamik taleplere karşı esneklik sağlamaktadır.

Gerçekleştirilen çalışma

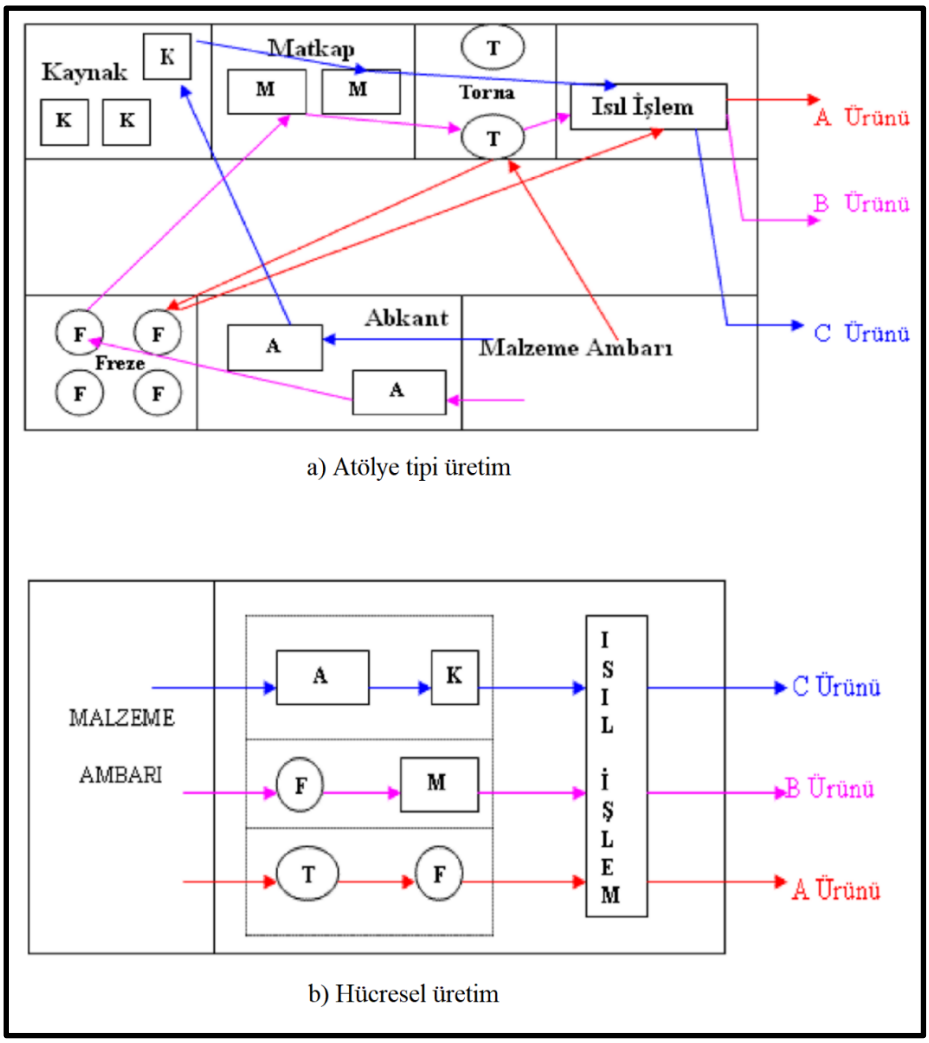
Yapılan çalışmada hücre oluşturma problemi ele alınmış ve problemin çözümünde 'Yapay Sinir Ağları (YSA)'nın denetimli ve denetimsiz öğrenmeye dayalı iki tipi, modüler yaklaşımla birleştirilerek kullanılmıştır. Ağların birleştirilerek kullanılmasıyla, hücre oluşturma problemi için, karmaşıklığı azaltmak ve ağların tek başına ulaşamayacakları sonuca birlikte ulaşmasını sağlamak hedeflenmiştir. Çalışmada hücre oluşturma problemi iki alt probleme bölünmüş, oluşturulan ağların birleştirilmesi ile önce parçalar operasyonlarına bağlı olarak tezgâhlara, ardından tezgâhlar hücrelere atanmıştır. Yapay sinir ağlarının birleştirilmesi noktasında uygulanan modüler yaklaşım, sıklıkla sınıflandırma problemlerinde kullanılan bir yöntem olmakla beraber, literatürde mevcut çalışma kapsamında ele alınan hücre oluşturma (kümeleme) problemine yönelik modüler yaklaşımın kullanıldığı bir çalışmaya rastlanmamıştır. Gelişen teknoloji ile birlikte çalışmalar uygulamalarla desteklenebilir hale gelmiş ve araştırmacılara çözüm için farklı seçenekler kullanabilme imkânı sunmuştur. Bu noktada mevcut çalışmanın talaşlı imalat ile üretim yapan firmalarda uygulanabilir bir alt yapı sağlamış olması da amaçlanmaktadır. MATLAB programı ile oluşturulan yapay sinir ağları yine program dahilinde geliştirilen DVCM App programında birleştirilerek uygulama işletmesinin kullanımına sunulmuştur.

2. HÜCRESEL İMALAT

Değişerek gelişen ekonomik ve çevresel faktörler, yıllar boyunca üretimle uğraşan firmalar için bir sınav niteliğinde olmuştur. Firmalar bu zorlu inovasyon sınavında başarıya ulaşmak için son yüzyılda pek çok farklı yöntem denemişlerdir. 1960'larda Japonya'yı, 1970'lerde Avrupa'yı ve 70'lerden sonra ABD'yi saran Yalın Üretim yaklaşımı, Taiici Ohno'nun Toyota fabrikasının üretim süreçlerinde uyguladığı, adı bu yüzden hala Toyota Üretim Sistemi (TPS) olarak anılan, bir dizi yöntemden oluşur. Temel olarak "israfı önlemek" ve "maliyetleri düşürmek" anlamındaki Yalın Üretim'e, Tam Zamanında Üretim (JIT) hedeflenerek yola çıkılmıştır. JIT, stokta bulundurma maliyetini sıfırlamayı, müşterinin talep ettiği ürünü talep anında üretime almayı hedefleyen bir üretim yöntemidir [25]. Yalın üretimde "taşıma", "bekleme süresi", "stokta bulundurma" gibi ürüne değer katmayan her işlem "israf" olarak ele alınır ve gereksiz bir maliyet olarak görüldüğünden azaltılmalı hatta yok edilmelidir. Literatürde "Yalın Felsefe" olarak da yer alan bu yaklaşım, Kaizen, Poka Yoke, Takım Çalışması, Görsel Kontrol, Hücresel İmalat, Tek Parça Akışı, Kanban Sistemi gibi çok sayıda farklı tekniği kapsamaktadır. Bu tekniklerin tamamının birden uygulanması pratikte zahmetli olsa da bir veya birden çoğunu bir arada uygulamak mümkündür.

Yalın Üretim tekniklerinden birisi olan Hücresel İmalat (CM), Grup Teknolojisi (GT)'nin üretim sahasına uygulanması olarak ifade edilir ve çoğu zaman bu iki ifade bir arada kullanılmaktadır. GT, parçaların benzerliklerinden yararlanarak, tasarım ve üretim süreçlerinde bu parçaları gruplamayı esas alır. Atölye tipi üretimde işlevlerine göre yerleştirilmiş olan tezgâhlar, grup teknolojisinde birbirine benzeyen bu parçaların oluşturduğu parça ailesinin operasyonlarına ve rotasına göre yerleştirilir. Böylece atölye tipi üretimin esnekliği ile yığın üretimin hızını birlikte kullanan melez bir yapı kurulmuş olur [26].

Atölye tipi üretimde birden çok tezgâhta operasyona ihtiyaç duyan bir parça, fonksiyona göre yapılan tezgâh yerleşimden dolayı neredeyse tüm üretim sahasını dolaşmaktadır. Hücresel imalat yapısında ise tezgâhlar parçanın rotasına göre belirli bir hücre içinde yerleştiğinden atölye tipi yerleşimdeki zaman alıcı ve masraflı seyahat ortadan kalkmaktadır. Şekil 2.1'de atölye tipi üretim (a) ve hücresel üretim (b) yapısı görülmektedir.



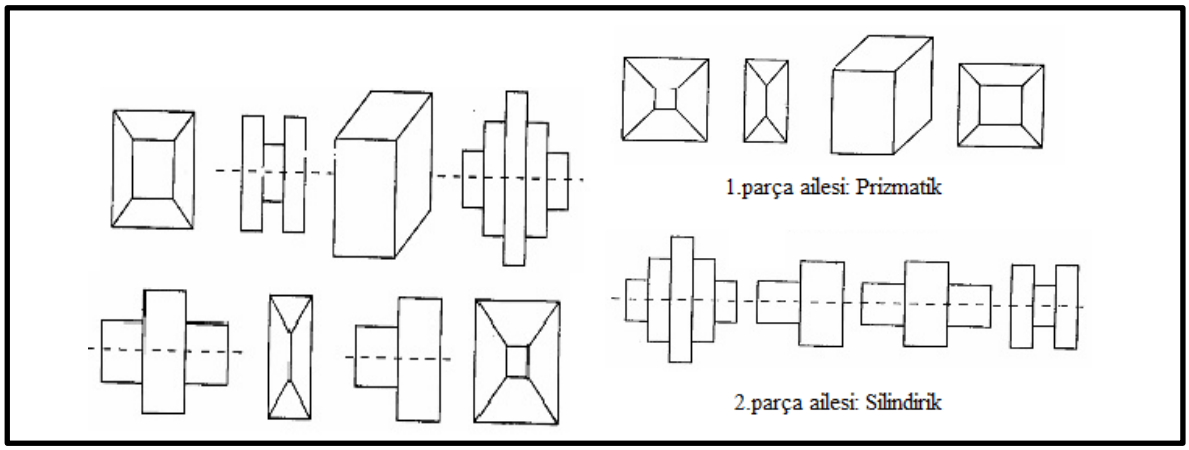
Şekil 2.1. Atölye tipi üretim (a) ve hücresel üretim (b)

2.1. Hücre Oluşturma Yöntemleri

İmalat hücrelerini kurmak Hücresel İmalatın ilk ve en önemli adımıdır. Soleymanpour vd. benzer parçaları ve benzemez makineleri gruplayarak CM hücresi oluşturdukları çalışmalarında, çok kriterli hücresel imalat sistemi tasarı probleminin çözümünde görsel gruplama haricinde iki temel yaklaşım olduğunu ortaya koymuşlardır. Bunlardan ilki parçaları geometrik özelliklerine göre kodlayarak sınıflandırmaya dayanan Parça Kodlama Analizi (PCA)'dir. İkincisi ise benzer parça operasyonlarına göre tezgâhları kümeleyen Üretim Akış Analizi (PFA) yaklaşımıdır [27].

Görsel Gruplama: Görsel gruplama parçaların malzeme, boyut, şekil gibi özelliklerinden yola çıkarak uygulanan bir yöntemdir. Gruplamayı yapacak olan kişinin parçalar hakkında bilgi ve tecrübeye sahip olmaması durumunda uygulanabilir bir yöntem değildir.

Şekil 2.2'de parçaların silindirik ve prizmatik görünümüne göre basitçe iki parça ailesine ayrıldığı görülmektedir [28].



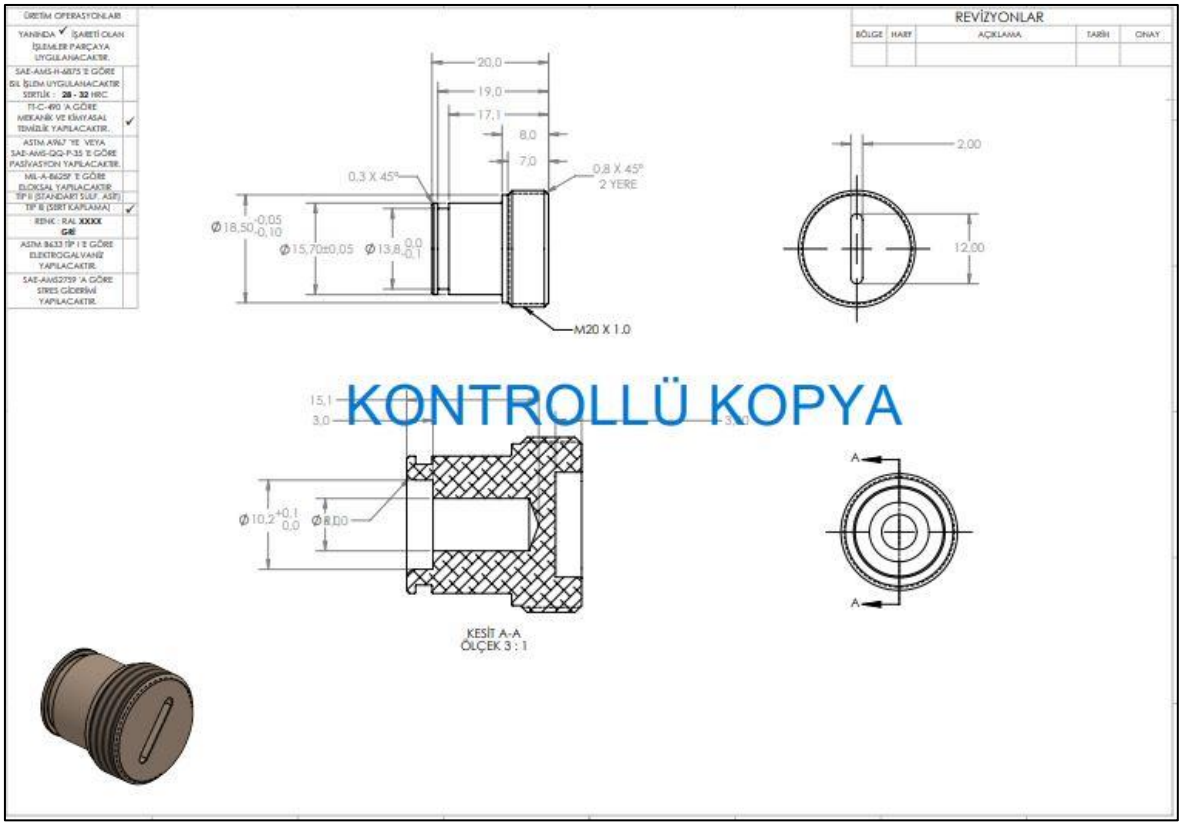
Şekil 2.2. Görsel gruplamayla oluşturulan parça aileleri.

Parça Kodlama Analizi (PCA-Part Coding Analysis): Bilgisayar destekli tasarım adımlarından yola çıkarak oluşturulan bu kodlama sisteminde tasarım aşamaları nümerik ya da nümerik olmayan değerlerle eşleştirilir ve parçanın geometrisini tanımlamak için sınırlı sayıda karakter içeren bir ifade ortaya çıkar. Hiyerarşik kodlama, nitelik kodlama ve hibrid kodlama olmak üzere üç farklı tipte kurulabilen yapı, oluşturulması ve uygulanması açısından zor ve zahmetli olarak nitelendirilmektedir. Şekil 2.3'teki PCA sınıflandırma tablosundan yola çıkarak kübik yapıda, merkez deliği hariç delik bulundurmayan ve helezon dişliye sahip bir parçanın PCA kodunun 32111 olacağı görülmektedir [28].

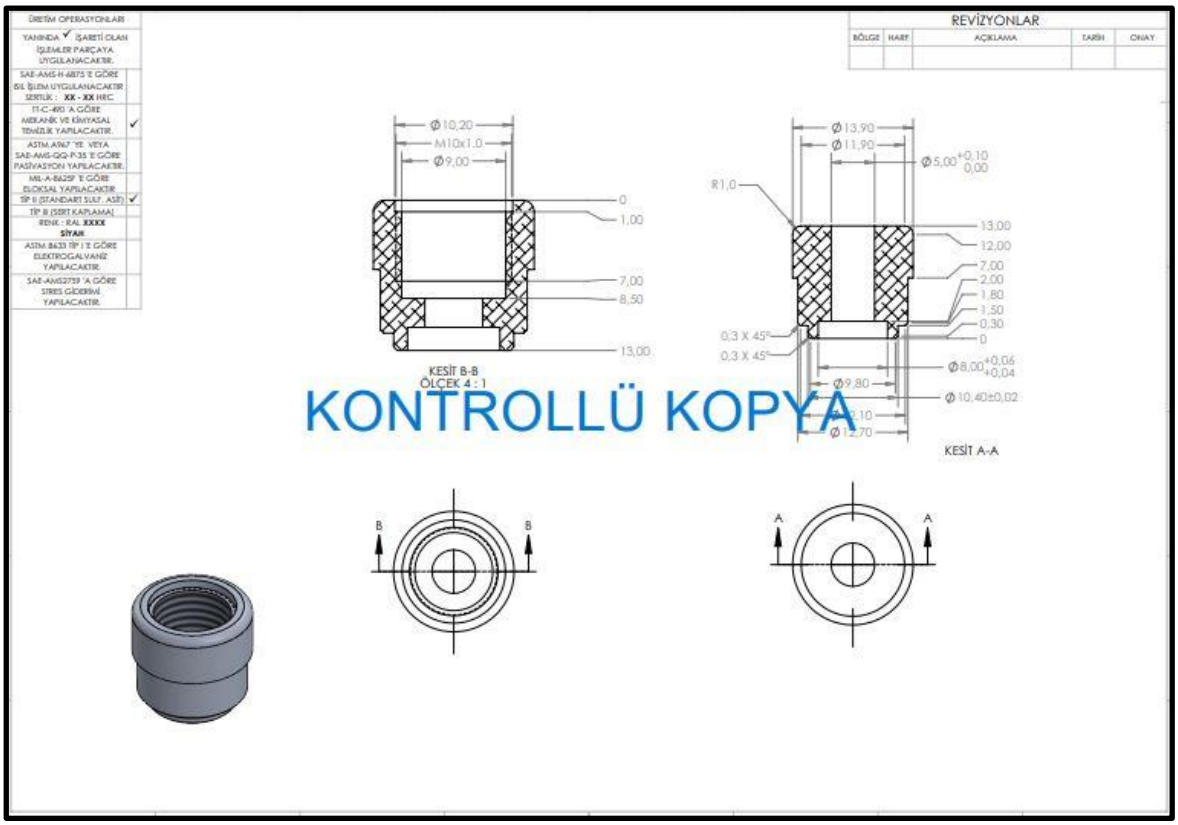
Karakter	Özellik	Karakter seçenekleri			
		1	2	3	4
1	Dış şekil	Toleranslı silindirik	Toleranssız silindirik	Kübik	…
2	İç şekil	Yok	Merkez deliği	Yanal delik	…
3	Delik sayısı	0	1-2	3-5	…
4	Delik çeşidi	Eksenel	Çapraz	Eksenel çapraz	…
5	Dişliler	Helezon	İç destek	Dış destek	…

Şekil 2.3. PCA sınıflandırma ve kodlama tablosu

Parçaların geometrik özellikleri benzer olsa da operasyon açısından aynı grupta olmaları mümkün olmayabilir. Bu yüzden Görsel Gruplama ve PCA yöntemleri etkili sonuçlar verememekte, ancak küçük ölçekli ve manuel çözümü mümkün olan problemlerde uygulanabilmektedir. Resim 2.1'de ve Resim 2.2'de geometrileri benzemesine karşın üretim süreçleri farklı olan parçalar verilmiştir. Parçaların ikisi de silindirik yapı da görülmektedir ve yetkin olmayan bir göz tarafından ilk bakışta torna operasyonu ile ortaya çıkabileceği düşünebilir. Ancak Resim 2.1'de yer alan parçayı elde etmek için yalnızca torna operasyonları yeterli olurken Resim 2.2'deki parçayı elde etmek için hem torna hem freze operasyonlarına gerek duyulmaktadır.



Resim 2.1. Geometrileri benzer üretimleri farklı parçalar-1



Resim 2.2. Geometrileri benzer üretimleri farklı parçalar-2

Üretim Akış Analizi (PFA-Part Flow Analysis): Parçaların bilgisayar destekli tasarım verilerinden ziyade üretim prosesindeki rota bilgilerinden yola çıkarak gruplamaya dayanan bir yöntemdir. En çok kullanılan ve en etkin yöntem olarak bilinen PFA, hücre oluşturma probleminin ilk aşaması olan parça-makine blok matrisinin temel verilerini oluşturmaya dayanır. Örnek bir parça makine matrisi Şekil 2.4'te verilmiştir.

Mevcut çalışmada bu yöntemler arasında en çok tercih edilen ve operasyon bilgilerini de çözüme dahil eden PFA yöntemi kullanılmıştır. Yöntemin detayları ve uygulama adımlarına ilerleyen bölümlerde yer verilmiştir.

Makineler	Parçalar							
	P1	P2	P3	P4	P5	P6	P7	P8
M1	1	1		1				1
M2					1			
M3			1		1			
M4		1				1		
M5	1			1				1
M6			1					
M7		1				1	1	

Şekil 2.4. PFA parça -makine matrisi

2.2. Sanal Hücresel İmalat Sistemi (VCMS)

Firmalar CM'nin pozitif etkilerini muhafaza ederken, negatif yönlerini de azaltmak adına Sanal Hücresel Üretim Sistemine yönelmiştir. Bu sistem yeni siparişler karşısında sistemin fiziksel yapısında hiçbir değişiklik yapmaksızın değişime uyum sağlamayı sağlamaktadır. Daha geniş bir ifadeyle VCMS (Virtual Cellular Manufacturing System), CM'nin üretimi parça aileleri bazında gerçekleştirmesinin avantajını kullanarak bir üretim kontrol mekanizması üzerinden makineler, çalışanlar, taşıma ekipmanları vb. kaynakların sanal bir gruplanmasını yapmaktadır. Bu sistem klasik CM'in hazırlık zamanı etkinliğini ve atölye tipi üretimin çizelgelemedeki esnekliğini eşzamanlı olarak kullanmaktadır.

Ayrıca üretimdeki parça ailelerinin sayısı ve büyüklüğüyle ilgili bir kısıtlaması yoktur. CM'de hücrenin kapasitesi ve yerleşimi sabittir, VCM'de ise hücre yapısı değişen ihtiyaçlara göre yerleşimin sanal olarak yeniden ayarlanabilmesi esnekliğine sahiptir. Her bir makinenin bir parça ailesine ait olduğu klasik CMS yapısının aksine VCM'de ihtiyaç duyulması halinde bir makine birden çok hücrede kullanılabilmektedir ve ortak kullanımdaki tezgâhlar tüm hücreler tarafından kullanılabilir durumdadır [31]. Sanal hücresel imalat sistemi klasik CM'den iki konuda farklılık gösterir: Makinelerin ortak kullanılabilmesi ve makinelerin hücre içindeki fiziksel konumu. Geleneksel hücresel imalat sisteminde parça ailesi, kendisi için tanımlanmış belirli bir hücrede işlem görmek durumundadır.

Hücreler arasında parça değişimi mümkün değildir. Hücreler arası parça değişiminin desteklenmemesi, sıklıkla makine çoğaltılmasına ve düşük makine kullanım oranına sebep olmaktadır. Sanal hücreler arasında makine paylaşımının olması bu sorunları ortadan kaldırmaktadır. Sonuç olarak yeni yatırım maliyeti ve düşük kapasite kullanımından kaynaklanan zarar ortadan kalkmaktadır [32].

Ratchev, sanal imalat hücresi tasarımında dört adımdan oluşan bir metot tanımlamıştır. İlk adım bileşen ihtiyaçlarını belirlemek ve alternatif operasyonlar türetmektir. Daha sonra sanal hücrenin sınırları tanımlanmalı, devamında hangi tezgahlara ihtiyaç duyulacağı belirlenmelidir. Son adım ise sistem değerlemesidir [33]. Pek çok araştırmacı talebin kesin olmadığı veya tahmin edilemez olduğu durumlarda VCMS kullanılmasını tavsiye etmiştir. Tüm bunların yanında VCMS'nin gerçek faydaları ancak dinamik bir yapı kurulduğunda ortaya çıkmaktadır. Dinamik yapıdan bahsedilen bu noktada şu sorular gündeme gelmektedir:

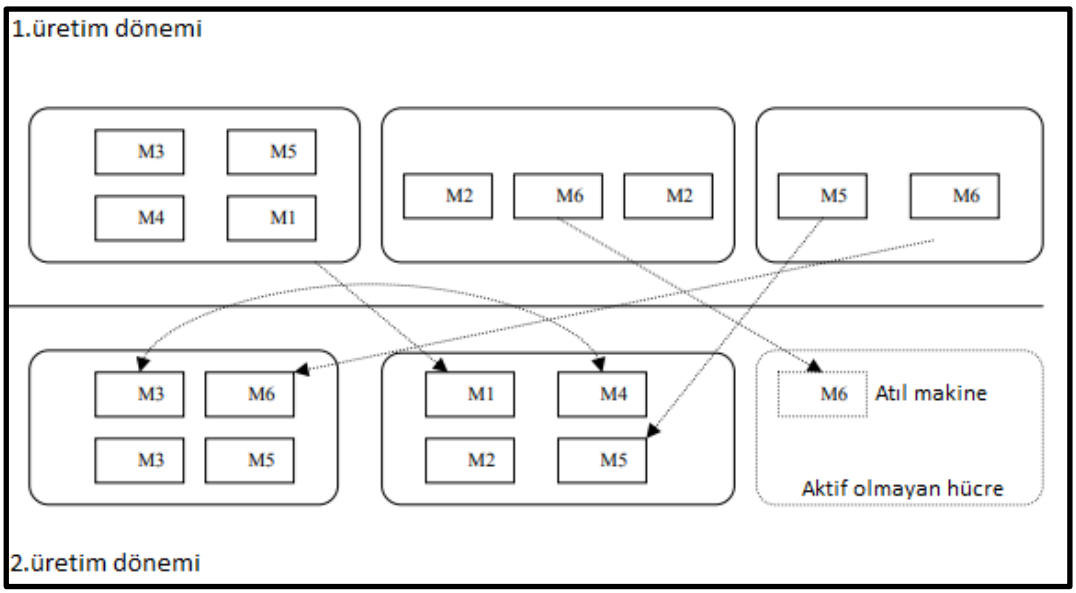
- Hücrelerin yeniden yapılanması hangi sıklıkta gerçekleşecek?
- Belirlenen bir hücre için periyod uzunluğu ne olacak?
- VCMS yeniden yapılandırılırken hücredeki iş/parça hacmi ne olacak?

Bu sorulara verilen cevaplar doğrultusunda sistemin performans kriterlerinin optimizasyonu kolaylaşacaktır [34].

2.3. Dinamik Hücresel İmalat Sistemi (DCMS)

Dinamik hücresel imalat, hücrelerin fiziksel olarak gruplanmasına izin verirken uygun durumlarda hücrenin yeniden yapılanmasına da fırsat veren bir sistemdir. Yeni ürünlerden ötürü talepte değişiklik öngörülen, yani talebin dinamik olduğu durumlarda kullanılması uygun görülen bir yapıdır. DCMS (Dynamic Cellular Manufacturing System), iş istasyonlarının taşınabilir olması durumunda sanal hücreleri fiziksel olarak yeniden konumlandırmaya izin verir [35]. Şekil 2.5'te talepteki değişime göre iki farklı üretim periyodu için yeniden yapılandırılmış dinamik imalat hücreleri görülmektedir. Bu yöntem CM yapısında bir üretim sistemi kurmaya müsait bir firmada ancak ve ancak makine-ekipmanın taşınabilir olması durumunda uygulamaya koyulabilmektedir. Taşınabilir iş

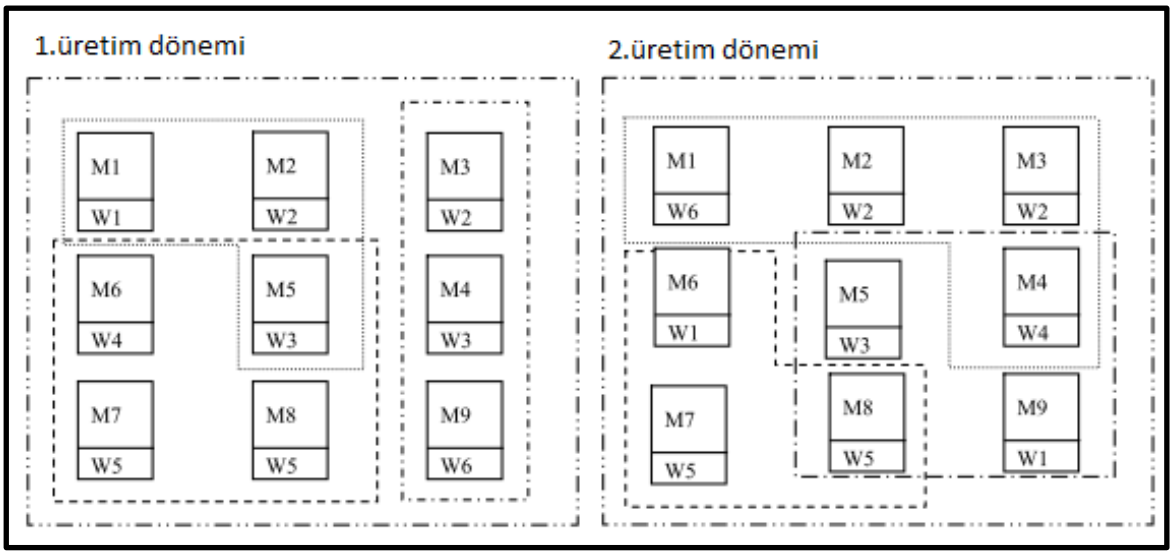
istasyonlarının bulunduğu bir montaj atölyesinde talepteki değişime göre, mevcut bir üretim hücresi diğer hücrelerle iş istasyonu değişimi yapmak suretiyle yeniden tasarlanabilir. Ancak talaşlı imalat yapan ve büyük tezgâhların kullanıldığı bir fabrikada talepteki dalgalanmayı karşılamak adına, hücreleri yeniden oluşturmak için tezgâhların yerini değiştirmek mümkün olmayacaktır. Bu noktada talepteki çeşitlilik ve değişiklik karşısında insan, makine vb. üretim kaynaklarının, fiziksel bir değişime gerek olmaksızın yeniden yapılandırılabildiği "sanal" hücreler, piyasadaki değişikliklere hızlı cevap verebilmenin bir yolu olarak karşımıza çıkmaktadır [36].



Şekil 2.5. Dinamik imalat hücreleri

2.4. Dinamik Sanal Hücresel İmalat Sistemi (D-VCMS)

İmalat hücrelerinin sanal ve dinamik yapısını bir arada barındıran D-VCMS (Dynamic Virtual Cellular Manufacturing System)' de sanal hücreler talep hacmindeki değişime göre belirli bir planlama dönemi boyunca yeni parça aileleri ve makine grupları oluşturarak yeniden yapılanmaktadır. Belirli bir periyotta, herhangi bir parça için operasyon gerektiğinde, bu parça en benzer parça ailesinin bulunduğu hücreye yönlendirilir ve sanal hücredeki makineler bu aile için hazırlanır. Eğer bir periyottan diğerine talep değişirse, sanal hücreye bir başka parça ailesi atanabilir. Özetle dinamik sanal hücresel üretim, belirli bir planlama dönemi boyunca değişen talebe göre parçaları parça ailelerine ve makineleri yeni gruplara atayarak dinamik şekilde çalışan bir sistemdir [37]. Şekil 2.6'da farklı iki periyotta ihtiyaca göre değişen dinamik sanal hücreler görülmektedir.



Şekil 2.6. Dinamik sanal imalat hücreleri

2.5. Hücresel İmalatın Avantajları ve Dezavantajları

Hücresel imalatın avantajları

Hücresel İmalat sisteminin başarısını ölçmek için kullanılan performans kriterlerinin ele alındığı Olorunniwo çalışmasından yola çıkarak sistemin avantajları şu şekilde sıralanabilir [29]:

- Atölye tipi üretimin çeşitliliği ve esnekliği ile seri üretimin hızı ve kalitesi birleşmiş olur.
- Hücre içindeki işlem süreleri, atölye tipi yerleşimdekine göre daha azdır.
- İşlem süreleri kısaldığı için parçalar yarı mamul halinde daha az beklediğinden yarı mamul parça maliyeti düşüktür.
- Her bir hücre belirli bir parça ailesini işlediğinden tezgâhlardaki ayar süresi ve takım maliyeti azdır.
- Çalışma alanından tasarruf sağlar ve malzeme taşıma süreleri daha kısadır.
- Hücreler ve operatörler parça ailelerine odaklı çalıştıklarından ürün kalitesi artar.

Hücresel imalatın dezavantajları

Tüm parçalar mükemmel şekilde sınıflandırılmış ve en yüksek performanslı hücreler oluşturulmuş olsa dahi Hücresel İmalatın bazı dezavantajları bu yapıyı verimsiz hale getirmektedir [30]:

- Parça aileleri ve makine grupları bir kez oluşturulur ve hücreler buna göre belirlenir. Hücreyi ihtiyaca göre yeniden düzenlemek çok zor ve maliyetlidir.
- Bir parça ailesi yalnızca atandığı hücrede işlem görebilir. Parçanın başka bir hücreye geçmesi mümkün değildir. Bu yüzden bazı makinelerden birden fazla sayıda bulundurmak gerekebilir. Bu da yatırım maliyetini artıracaktır.
- Oluşturulan hücrede çalışacak personelin hücredeki tüm makineleri yönetebilecek kabiliyette olması gerekeceğinden nitelikli personel ihtiyacı ve işgücü maliyeti artabilir.
- Hücreler makine duruşlarına karşı çok duyarlıdır ve arıza durumunda bir hücre tamamıyla sistem dışı kalabilir. Bu da önleyici bakım onarım maliyetinin yüksek olması demektir.

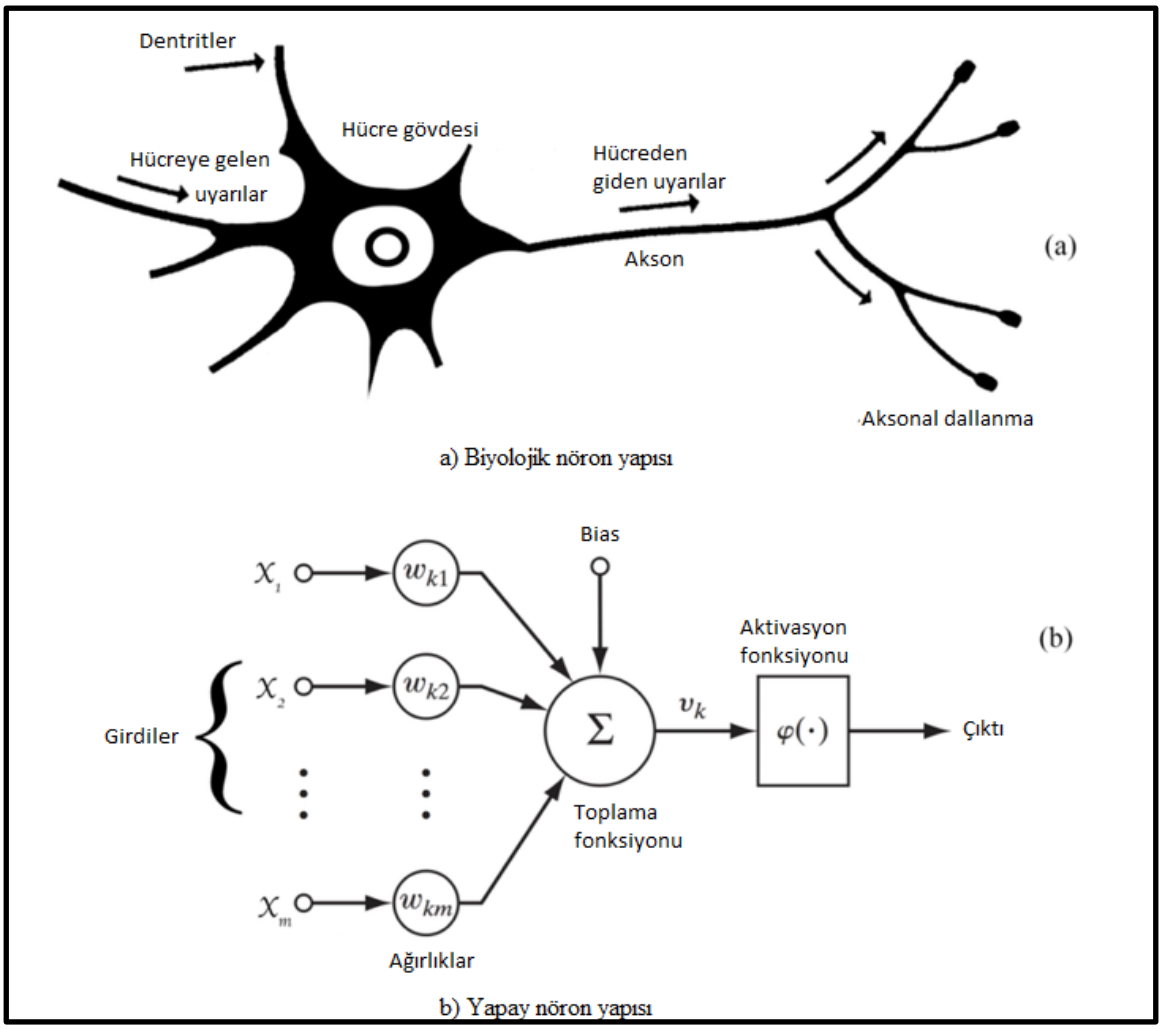
Hücresel İmalatın bahsedilen bu dezavantajlarından dolayı, bu sistemi tümüyle uygulamayı başaran firmalar, karşılaştıkları yeni sorunlar için yeni yöntemler geliştirmek zorunda kalmışlardır. Hücresel İmalatın avantajlarını koruyarak, dezavantajlarını ortadan kaldırmak için çalışmalar yapılmış ve literatürde yerini alan iki sistem ortaya çıkmıştır: Sanal Hücresel İmalat ve Dinamik Hücresel İmalat sistemleri. Mevcut çalışma kapsamında ele alınan Dinamik Sanal Hücresel İmalat sistemi ise bu yeni iki yöntemin avantajlarını bir arada barındırmaktadır.

3. YAPAY SİNİR AĞLARI

Çağımız teknolojisinin geldiği noktada insanoğlu matematiksel formüllerle ifade edilemeyen ve çözümü çok zor olan problemleri zeki sistemler aracılığıyla çözme imkânı bulmuştur. Makinelerin zekâsı olarak adlandırılan "Yapay Zekâ", günümüzde karşılaşılan problemlerin karmaşıklığı ve değişkenliği karşısında, öğrenme, analiz etme, sınıflandırma, optimizasyon, problem çözme ve karar verme becerileri ile hayatımızın her alanında uygulanabilecek çözümler sunmaktadır.

Yapay zekâ biliminin bir alt dalı olarak ele alınan Yapay Sinir Ağları (YSA), "makine öğrenmesi" olarak bilinen ve bilgisayarların olayları öğrenmesini sağlayan yaklaşımlardan biridir [38]. Örneklerden öğrenmeye dayanan bu yaklaşım biyolojik sinir ağı yapısını taklit etmektedir. Nasıl ki insan, tecrübelerine dayanarak yeni kararlar alabiliyorsa, bir yapay sinir ağı da öğrendiklerinden yola çıkarak daha önce karşılaşmadığı durumlar karşısında karar verebilme yeteneğine sahiptir. Bunun için sinir ağındaki her bir sinir hücresi, kendisine sunulan bilgiyi bir diğer sinir hücresine iletir ve bu iletim ağ öğrenene kadar devam eder. Çalışma prensibi basitçe bu şekilde açıklanabilmesine rağmen YSA halen "kara kutu" olarak anılmaktadır. Çünkü bir ağın sonuca ulaşmasında etkili olan tüm bileşenler (girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu, gizli katman sayısı, nöron sayısı, öğrenme algoritması) matematiksel olarak açıklanabilir olsa da bir yapay sinir ağının tüm bunları nasıl gerektiği şekilde bir araya getirerek bir çıktı elde ettiği bilinmemektedir.

Yapay sinir ağları biyolojik sinir sisteminden etkilenerek geliştirilmiştir. Biyolojik sinir hücreleri birbirleriyle sinapslar vasıtası ile iletişim kurarlar. Bir sinir hücresi işlediği bilgileri aksonlar yolu ile diğer hücrelere gönderir. Benzer şekilde yapay sinir hücreleri dışarıdan gelen bilgileri bir toplama fonksiyonu ile toplar ve aktivasyon fonksiyonundan geçirerek çıktıyı üretip ağın bağlantıları üzerinden diğer hücrelere gönderir. Şekil 3.1'de biyolojik bir nöron yapısı (a) ve gerçek bir nöron yapısı (b) birlikte verilmiştir [38].



Şekil 3.1. Biyolojik nöron yapısı (a) ve yapay nöron yapısı (b).

Bir yapay sinir ağının modelini şu bilgiler karakterize etmektedir: Ağın mimarisi, kullanılan toplama fonksiyonu, kullanılan aktivasyon fonksiyonu, öğrenme stratejisi ve öğrenme kuralı. Her bir sinir hücresi bir proses elemanı olarak adlandırılmaktadır ve proses elemanları genellikle girdi katmanı, ara katman ve çıktı katmanı olmak üzere üç katman halinde bir araya gelerek bir ağ oluştururlar. 'Girdi katmanı' sinir ağına bilgilerin sunulduğu yani girdilerin alındığı katmandır. Alınan bilgi, ağırlık ve bias değerleri, toplama fonksiyonu ve aktivasyon fonksiyonu da sürece dahil edilerek ara katmanlarda işlenir. Ağa gelen bilgilerin işlenmesi yani çıktıya dönüştürülmesi işlemi ancak doğru ağırlıkların bulunması ile mümkün olmaktadır. Bu süreç ağın eğitilmesi olarak adlandırılmaktadır. Bir YSA, başlangıçta rastgele atanan ağırlık değerlerini, belirlenen öğrenme algoritmasına göre güncelleyerek en doğru değerleri bulmaya çalışır ve bu işlem tamamlandığında YSA öğrenmiş olur [39].

3.1. Yapay Sinir Ağlarının Özellikleri

Doğrusal olmama, öğrenebilme, genelleme yapabilme ve uyarlanabilme özellikleri sayesinde Yapay Sinir Ağları pek çok farklı tipteki problemin çözümünde kullanılabilmektedir. YSA'nın sahip olduğu temel özelliklere aşağıda kısaca değinilmiştir.

Doğrusal olmama: YSA'nın başlıca özelliği olan doğrusal olmama, sinir ağının kullandığı aktivasyon fonksiyonundan ileri gelmektedir. Sigmoid ve tanjant hiperbolik aktivasyon fonksiyonları yapay sinir ağlarında en çok kullanılanlar aktivasyon fonksiyonlarıdır. Ağın temel proses elemanı olan sinir hücresi doğrusal olmadığından hücrelerin birleşmesinden oluşan sinir ağı da doğrusal değildir. Bu özelliği ile YSA, doğrusal olmayan karmaşık problemlerin çözümünde etkili sonuçlar üretebilmektedir.

Öğrenme: Öğrenme ve öğrendiklerinden yola çıkarak tahminde bulunma YSA'nın temel özelliğidir. Bir sinir ağının öğrenebilmesi için sinir hücreleri arasındaki bağlantıların doğru yapılması ve doğru ağırlıkların bulunması gerekir. Ancak bu bağlantıların nasıl yapılacağı ve ağırlıkların nasıl belirleneceği konusunda kesin yaklaşımlar yoktur. Bu sebeple ağa başlangıçta rastgele verilen ağırlık değerleri ve örneklere karşılık, kendi kendine öğrenmesini sağlayacak şekilde çıktı değerleri sunulur.

Genelleme: YSA, öğrenmeyi tamamladıktan sonra, daha önce karşılaşmadığı girdi değerlerine karşılık çıktı değerleri üretebilme yeteneğine sahiptir. Bu noktada önemli olan yapay sinir ağına eğitim için sunulan girdilerin, problemin tamamı için örnek teşkil edebilecek ve ağın genelleme yapmasını sağlayacak örneklerden seçilmesidir. Parça ölçüsü ve operasyonuna göre uygun tezgâhı öğrenen bir yapay sinir ağı, ilk defa ağa sunulan bir parçanın hangi tezgâhta işlenebileceğini tahmin edebilecektir.

Uyarlanabilirlik: Bir yapay sinir ağı eğitildiği problemdeki değişikliklere uyum sağlayabilmektedir. Problemde karşılaşılan değişikliklere karşın YSA tekrar eğitilebilir. Bu değişikliklerin süreklilik arz etmesi durumunda da YSA gerçek zamanlı olarak eğitime devam edebilmektedir. Güvenli, üretim, sağlık gibi pek çok farklı sektörde kalite kontrol, sinyal işleme, unsur tanıma, denetim gibi anlık durumlar karşısında YSA'nın bu özelliğinden faydalanılmaktadır [40].

3.2. Yapay Sinir Ağlarında Öğrenme Yöntemleri

Yapay Sinir Ağlarında denetimli, denetimsiz ve destekleyici olmak üzere 3 temel öğrenme yöntemi kullanılmaktadır.

Denetimli (Supervised): Ağa belirli giriş değeri için çıkış değerleri verilir. Nöronlar arasındaki ağırlıklar verilen giriş ve çıkış değerlerine göre ayarlanır. Delta öğrenme kuralı ve geri yayılmalı (back-propagation) öğrenme denetimli öğrenmedir. Denetimli öğrenme yönteminin seçilmesi durumunda izlenecek 3 temel adım vardır. İlki uygulama yapılacak problem için verilerin toplanmasıdır. Bu noktada toplanan verilerin problemin tamamını temsil ettiğinden emin olunmalıdır. İkinci olarak toplanan veriler bir yapay sinir ağının anlayabileceği formatta düzenlenmelidir. Üçüncü adımda uygun ağ yapısı kurularak eğitilmeli ve daha önce görmediği örneklerle test edilmelidir. Bu noktadan sonra eğitimi tamamlanmış olan ağ artık aynı şartlar altında karşılaşılan yeni problemler için kullanılabilecektir. Farklı koşullara karşı ağın yeniden eğitilmesi bu 3 adımın tekrarlanması ile mümkün olabilmektedir.

Denetimsiz (Unsupervised): Bu öğrenme yönteminde ağa çıkış değerlerinin verilmesine gerek yoktur. Öğrenme süresince sadece örnek giriş değerleri verilir ve örnekler arasındaki ilişkiler ile sistemin kendi kendisine öğrenmesi sağlanır. Genellikle sınıflandırma, kümeleme, filtreleme gibi amaçlarla kullanılır ve eldeki verilerin benzerlik durumuna göre ağ tarafından gruplanmasının istendiği durumlarda seçilen bir öğrenme yöntemidir. SOM Ağı, Hopfield ağı, Rekabetçi Sinir Ağları (CNN) ve ART ağı denetimsiz öğrenme yöntemini kullanan yapay sinir ağlarıdır.

Destekleyici (Reinforcement): Denetimli öğrenmenin farklı bir türüdür. Çıkış değerlerini ağa vermek yerine, giriş değerlerine göre çıkışı değerlendirerek kademeli olarak öğrenmeye dayanır. Koşul ve kısıtların çok fazla olduğu karmaşık problemler için uygun bir yöntemdir ve ağın öğrenme süreci boyunca karşılaşılan tüm durumlar gözden geçirilerek ilerlenir. Satranç oyunları bu öğrenme yöntemine örnek olarak verilebilir ki çok fazla sayıda durumla karşılaşılan oyun, rakibin her bir hamlesine karşılık bir sonrakine temel oluşturacak şekilde bir öğrenme döngüsüyle ilerlemektedir. Genetik Algoritmalar ve LVQ (Learning Vector Quantizer) ağı destekleyici öğrenme yöntemini kullanmaktadır [41].

3.3. Yapay Sinir Ağı Tipleri

Yapay sinir ağı tiplerini belirleyen karakteristikler ağın mimarisi, ağ içindeki bağlantı şekilleri ve öğrenme yöntemleridir. Bağlantı şekilleri göz önüne alındığında yapay sinir ağlarını 2 grupta sınıflandırmak mümkündür: Veri akışının döngüsel olmadığı ve ileri yönde yapıldığı "İleri Beslemeli Ağlar (Feed-forward Neural Networks)" ve veri akışında geri beslemeler olduğu için döngüsel yapıda olan "Tekrarlayan Sinir Ağları (Recurrent Neural Networks)". Çizelge 3.1'de ise öğrenme yöntemlerine göre denetimli ve denetimsiz öğrenmeye dayalı olarak 2 gruba ayrılan yapay sinir ağları verilmiştir [42].

Çizelge 3.1. Öğrenme yöntemlerine göre YSA tipleri

Yapay Sinir Ağı Tipleri	
Denetimli Öğrenme	Denetimsiz Öğrenme
Perceptron	SOM Ağı (Self Orginizing Maps)
Adaline ağı	Rekabetçi Ağlar (Competitive Networks)
Çok Katmanlı Perceptron (Multilayered Perceptron)	ART Ağı (Adaptive Resonance Theory)
Geri Yayılım Ağları (Back-propagation Networks)	Hopfield Ağı
İleri beslemeli Ağlar (Feed-forward Networks)	Hamming Ağı
Olasılık Tabanlı Ağlar (Probabilistic Networks)	Potts Ağı (Potts Mean Field Annealing)
Boltzman Makinesi (Boltzman Machine)	IAC Ağı (Interactive Activation and Competition Network)
Yüksek Dereceli Ağlar (High Order Networks)	Kaotik Ağlar (Transiently Chaotic Networks)

3.4. Yapay Sinir Ağının Mimarisi

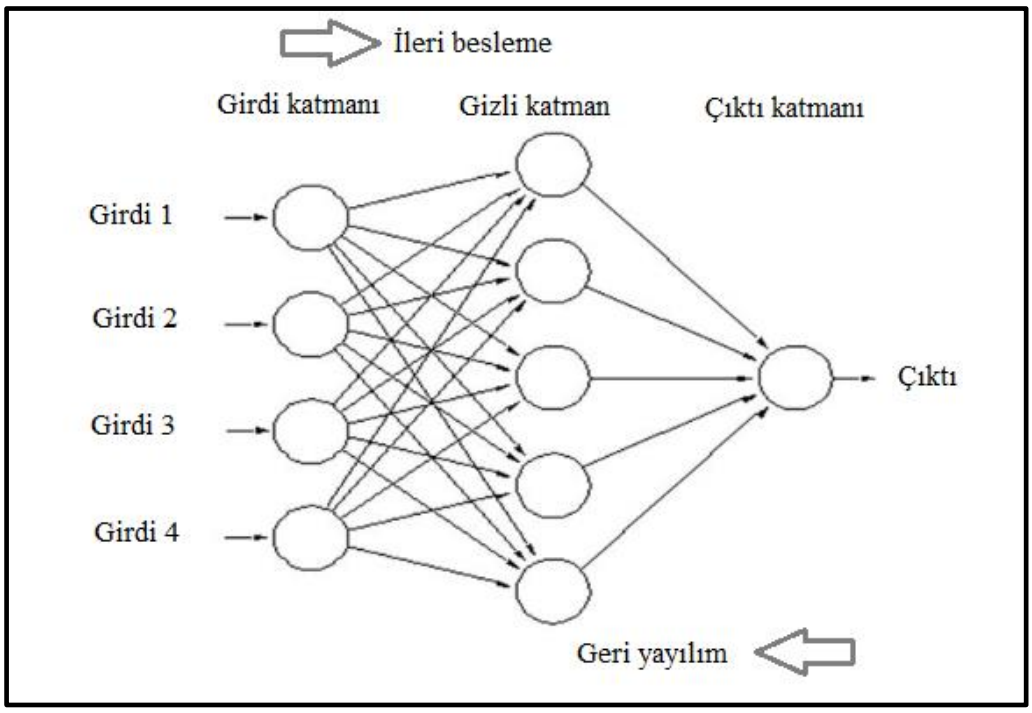
Bir yapay sinir ağı oluşturulduktan sonra, yapılandırılması ve eğitilmesi gerekmektedir. Yapılandırmadan kasıt, örnek verilerin çözmek istediğimiz probleme uygun şekilde ayarlanmasıdır. Eğitim ise ağ içinde değiştirilebilir olan ağırlık ve bias değerleriyle oynayarak ağın öğrenmesini sağlamaktır [43]. Bu noktada karşımıza YSA'nın en temel sorusu çıkmaktadır: Bir yapay sinir ağı hangi algoritma ile eğitilmelidir? Bu sorunun cevabı veri setinin yapısı, ağdaki ağırlıklar, nöron sayıları, ağın kullanım amacı gibi pek çok etkene bağlı olduğundan hangi eğitim algoritmasının en iyi sonucu vereceğini belirlemek çok zordur.

Bir yapay sinir ağı oluşturulurken kullanılacak eğitim fonksiyonu ile birlikte akla gelen bir diğer soru da gizli katman ve nöron sayılarının ne olması gerektiğidir. İleri beslemeli (feed-forward) yapay sinir ağlarında tek gizli katmanın yeterli olduğu kanıtlanmış olmasına rağmen, teoride tek gizli katmanda yer alan nöronlarla çözülebilecek bir problem için pratikte çok daha fazla katmanda yer alan nöronlara gerek duyulabilmektedir [44]. Gizli katman sayısında olduğu gibi, katmanlardaki nöron sayısının ne olması gerektiği konusunda da literatürde farklı yaklaşımlar mevcuttur [45]. En iyi gizli nöron sayısına ulaşmak için ağ mimarisindeki birçok etkeni (örnek veri yapısı, girdi ve çıktı sayısı, eğitim algoritması, aktivasyon fonksiyonu gibi) bir arada değerlendirecek çok karmaşık bir yaklaşım gerekmektedir [46]. Gizli katman nöron sayısını belirlemek için literatürde genel kabul görmüş 3 temel yaklaşım şu şekildedir:

- Gizli nöron sayısı, girdi katmanı ve çıktı katmanı arasında bir değer olmalıdır.
- Gizli nöron sayısı girdi katmanın 2/3'ü değerinde olmalıdır.
- Gizli nöron sayısı girdi katmanının 2 katından az olmalıdır [47].

3.5. Çok Katmanlı İleri Beslemeli Sinir Ağları (MLF-Multilayer Feed-forward Neural Networks)

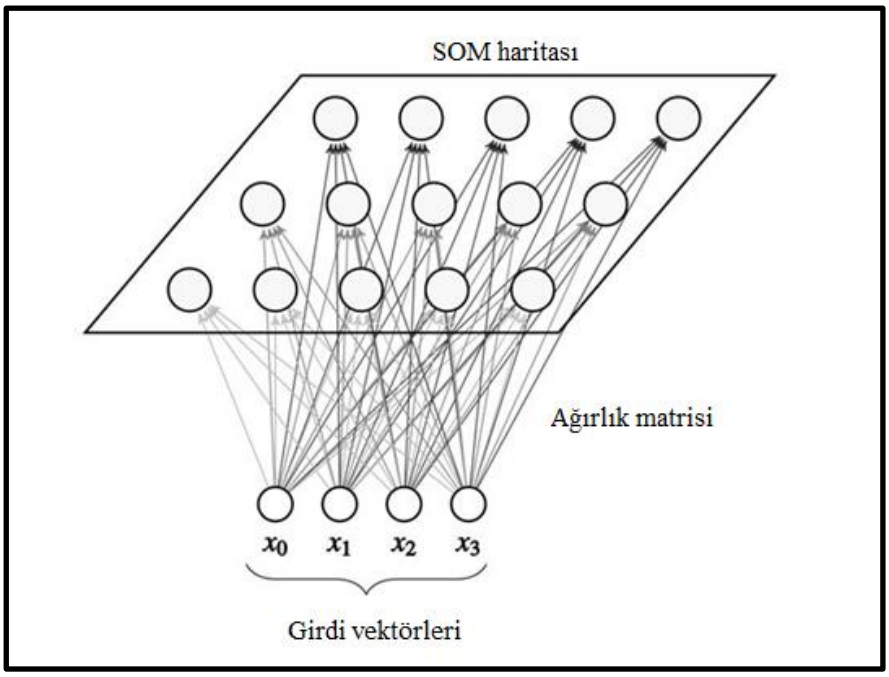
Çok katmanlı ileri beslemeli yapay sinir ağları geri yayılım öğrenme algoritması ile eğitilen ve en yaygın kullanılan sinir ağlarıdır. Çok katmanlı bir ağın mimarisi katmanlara yayılmış nöronlardan oluşur. İlk katman "girdi katmanı", son katman "çıktı katmanı" ve uygulayıcı inisiyatifinde oluşturulan ara katmanlar "gizli katman" olarak adlandırılır. MLF sinir ağı eğitim ve tahmin olarak iki aşamada çalışır. Her iki aşama için ayrı veri setleri gerekmektedir: *Eğitim veri seti* ile eğitilen ağın istenen sonucu verip vermediğini görebilmek için, daha öncesinde hiç karşılaşmadığı örneklerden oluşan *test veri seti* kullanılır. Rastgele seçilen ağırlık değerleriyle başlayan eğitim tekrarlı olarak devam eder. Tüm eğitim veri seti için tek bir tekrar *epoch* olarak ifade edilir. Sinir ağı, başlangıçta rastgele verilen ağırlık değerlerini her bir *epoch'*ta güncelleyerek hatayı minimize etmeye devam eder. Ağın eğitimin tamamlanması için çok fazla sayıda *epoch* gerekmektedir [48]. Bir MLF yapay sinir ağının yapısı Şekil 3.2'de verilmiştir.



Şekil 3.2. MLF ağının yapısı

3.6. SOM (Self Orginizing Maps) Sinir Ağı

1980'lerde Profesör Teuvo Kohonen tarafından ortaya koyulan, Kohonen ağları ya da SOFM (Self Organizing Feature Map) olarak da bilinen SOM sinir ağları, giriş verisini denetimsiz öğrenme yoluyla işleyerek, farklı verileri sınıflandıran bir sinir ağı modelidir. Model sistemin dahili yapısını harici bir harita olarak görüntüleme kabiliyetine sahiptir. Denetimsiz öğrenme yoluyla çalışan bu ağ, verileri kıyaslamak maksadıyla herhangi bir çıkış verisine ihtiyaç duymamaktadır. SOM örüntü tanıma ve kümeleme problemleri için en kullanışlı yöntemdir. Şekil 3.3'te görüldüğü gibi SOM ağı genellikle iki katmandan oluşur. Yapıda giriş modelindeki değer kadar giriş düğümü oluşur. Girdi vektörlerinin oluşturduğu, Kohonen katmanı olarak bilinen giriş katmanı bütünüyle çıkış katmanına bağlıdır ve SOM ağının çekirdeğini oluşturur. Kohonen katmanı biyolojik bir sistemi taklit ederek aldığı giriş verisini ağ boyunca iki boyutlu bir haritaya yaymaktadır [49]. SOM ağındaki öğrenme yöntemi, yalnızca kazanan nöronun ağırlığını değil, aynı zamanda komşuluğundaki nöronların ağırlıkları da değişeceği için diğer rekabetçi tabanlı kümeleme yöntemlerinden farklıdır. Bu ağ yapısında ağırlık değerleri, giriş değerleri arasındaki 2 boyutlu vektörel bir mesafe şeklinde ifade edilir.

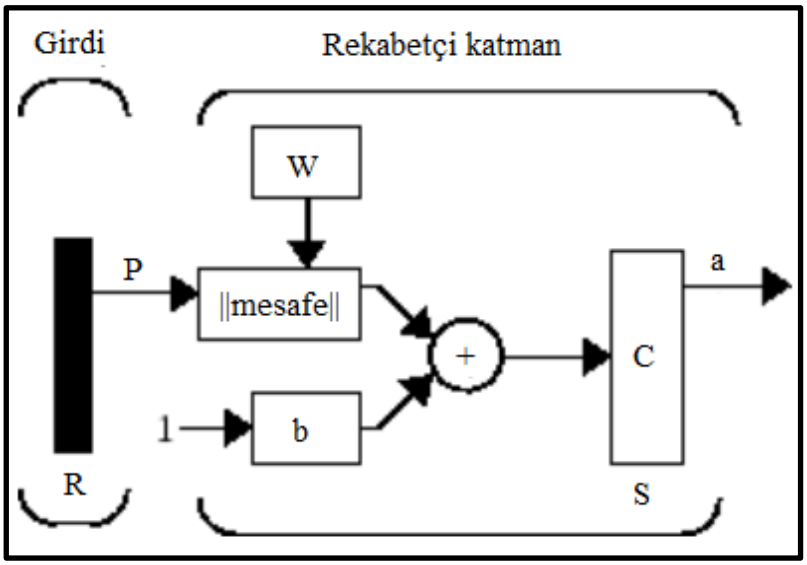


Şekil 3.3. SOM ağı katmanları

3.7. Rekabetçi Sinir Ağları (CNN-Competitive Neural Networks)

Önceki bölümlerde bahsedilen çok katmanlı ileri beslemeleri ağların öğrenme prensibinin aksine rekabetçi öğrenme de eş zamanlı olarak sadece tek bir nöron aktiftir. Çıktı katmanındaki yalnızca bir nöron, yani en güçlü girişe karşılık gelen etkin olacaktır. Kümeleme problemlerinin çözümde kullanılan CNN denetimsiz öğrenme yapısında bir yapay sinir ağıdır ve Winner-take-all (WTA) olarak bilinen 'kazanan hepsini alır' prensibiyle çalışmaktadır. Kazanan hepsini alır, bir katmandaki nöronların aktivasyon için birbirleriyle rekabet ettiği bir hesaplama ilkesidir [50]. Rekabetçi ağlar iki katmanlıdır ve tamamen bağlantılıdır. Genellikle bağlantılar katman içi değil, katmanlar arasıdır. Bağlantılardaki ağırlıklar rastgele olarak ayarlanır ve 0 ila 1 aralığında pozitif değerler seçilir. CNN'nin mimarisi Şekil 3.5'te gösterilmiştir [51]. Bir girdi vektörü olan *P* ağa verildiğinde, girdinin ağırlık vektörel değeri *W* ve bias değeri *b* işleme alınarak, ona en yakın olan ve ağırlık değeri *C* olan vektör kazanan olarak belirlenir. C vektörüne en yakın ağırlık değerine sahip 2. vektör rakip olarak belirlenir. Kazanan vektör ödüllendirilir ve ağırlık vektör değeri W, P vektörüne yakınsayacak şekilde güncellenir. Rakip olarak belirlenen vektör ise cezalandırılır ve ağırlık vektör değeri P girdi vektöründen uzaklaşacak şekilde güncellenir. Bu mekanizma tüm *rakip* vektörler kümeden uzaklaştırılıncaya kadar devam

eder ve kümeler en yakın vektörel ağırlık değerine sahip girdileri içerecek şekilde kendi kendine oluşur [52].



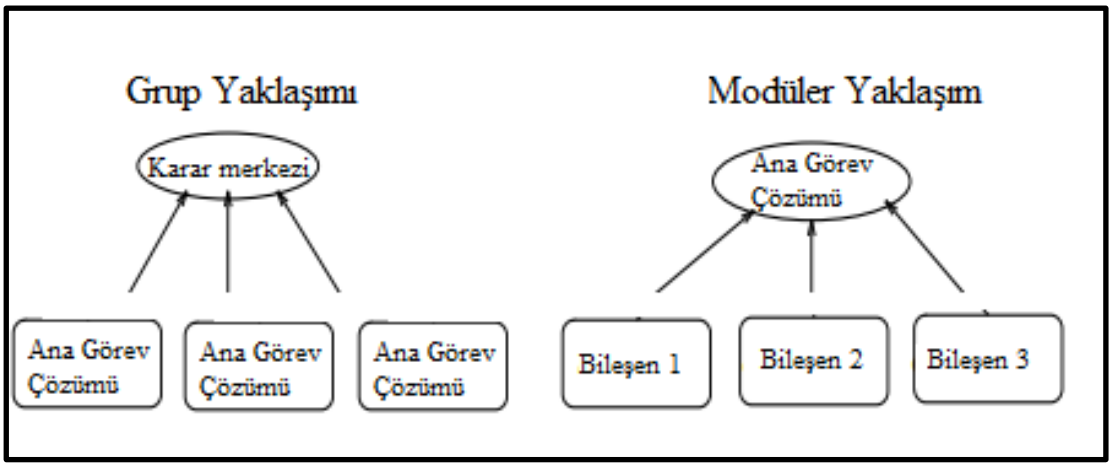
Şekil 3.4. CNN mimarisi

Bazı girdi vektörlerinin kümelere atanamama ihtimalinin olması Rekabetçi Sinir Ağlarının kullanılmasında bir kısıt olarak ifade edilmektedir. Bir başka ifadeyle bazı nöronların başlangıç ağırlık değerleri vektörel olarak girdiden çok uzak mesafede olabilmekte ve eğitim ne kadar devam ederse etsin yarışı kazanamamaktadır. Bunun önüne geçmek için bias değerleri kullanılır ve bu nöronlara yarışmaya katılması için bir şans verilmektedir.

3.8. Birleşik Yapay Sinir Ağları

Yapay sinir ağlarının birleştirilmesi kısaca, belirli sayıda ki ağların aynı görev üzerinde ayrı ayrı eğitilmesi ve sonuçlarının birleştirilmesi şeklinde ifade edilmektedir. Bu yaklaşım Hansen ve Salamon'un çalışması ile ortaya çıkmıştır. Bu yaklaşımda eğitilen ve bir araya getirilen bir çok sinir ağının yaptığı tahminlerin YSA'nın genelleme yeteneğini önemli ölçüde geliştirilebileceğini ine sürmektedirler [53]. Bu yöntem, makine öğrenmesi, yüz tanıma, karakter tanıma, hastalık teşhisi gibi çok sayıda girdi içeren, kompleks sinir ağı yapıları gerektiren problemlerin çözümü için yeni kapılar açmıştır.

Ağların birleştirilmesinde temel olarak iki ana yaklaşım bulunmaktadır [54]. Şekil 3.6'da grup yaklaşımı ve modüler yaklaşımın şematik gösterimi verilmiştir [55].



Şekil 3.5. Modüler ve grup yaklaşımı şematik gösterimi

Birleşik YSA'da grup yaklaşımı

Grup yaklaşımı (ensemble) ağların genelleme yeteneğini artırmak amacıyla kullanılmaktadır. Tahmin amacıyla kullanılan yapay sinir ağlarında, tek bir ağın tahmin değerini 'en iyi' olarak kabul etmek yerine, birden çok ağın ürettiği sonuçların en iyisini bulmak grup yaklaşımı ile mümkündür. Yapay sinir ağlarının grup yaklaşımı ile birleştirilmesinde 5 farklı yöntem uygulanmaktadır [55]:

Örnekleme yöntemi: Bu yöntemde her bir sinir ağı farklı veri setleri ile eğitilmektedir. En sık kullanılan yöntem olan örnekleme yöntemi çapraz doğrulama (cross-validation) ve özyükleme (bootstrapping) olarak adlandırılan yöntemleri kapsamaktadır. Çapraz doğrulama, eğitim ve test için birden çok alt örnek veri setinin kullanıldığı bir yöntemdir [57]. Özyükleme ise bilgisayar destekli istatistiksel bir algoritmadır ve rastgele verilerden oluşan veri seti içindeki bilgiyi ve verinin dağılımını ele alıp gürültüyü azaltmayı amaçlamaktadır [58].

Örnekleme yöntemi sayesinde örneklerin tamamı birden çok ağı eğitmek için kullanılmış olacak, neredeyse hepsi ağları ayrı ayrı test etmiş olacak ve veri seti içinde hata payı azalacağından nihai ağın tahmin yeteneği artmış olacaktır.

Veri setinin bölünmesi yöntemi: Örnekleme yönteminden farklı olarak bu yöntemde eldeki veri setinin tamamı baz alınır ve sonradan birleştirilecek olan ağların eğitimi ve testi için

bölünerek kullanılır. Bu durumda örnek sayısı her bir ağ için azalmış olacağından bileşik ağın performansına negatif etki edebilmektedir.

Uyarlanabilir örnekleme yöntemi: Bu yöntem özellikle görüntü tanıma çalışmalarında kullanılmaktadır. Sinir ağlarının görüntüleri filtreleyerek ayırması sağlanır ve daha sonra bu ağlar diğer ağların öğreticisi olarak kullanılır.

Farklı veri kaynakları yöntemi: Verilerin farklı giriş kaynaklarından toplandığı ve ana problemin farklı dallarını içerdiği bir yöntemdir. Örneğin yakıt enjeksiyon hatalarının belirlenmek istendiği bir birleşik ağ yapısında, ağlardan biri motor silindir basıncı verisini ele alırken, bir diğeri sıcaklık verisini işleyecek ve birleşik ağın tek bir karara varması sağlanacaktır.

Ön işleme yöntemi: Ağların eğitileceği veri setinin bir ön işleme tabi tutulması mümkün olabilmektedir. Özellikle sinyal işlemede kullanılan birleşik ağların eğitimi öncesinde veri seti çeşitli yöntemlerle dallandırılmakta, gürültü eklenmekte ve/veya budanmaktadır. Farklı ağların bu çeşitlendirilmiş veri setleriyle eğitilmesi sonucunda bileşik ağın daha tutarlı sonuçlara ulaşması hedeflenmektedir.

Birleşik YSA'da modüler yaklaşım

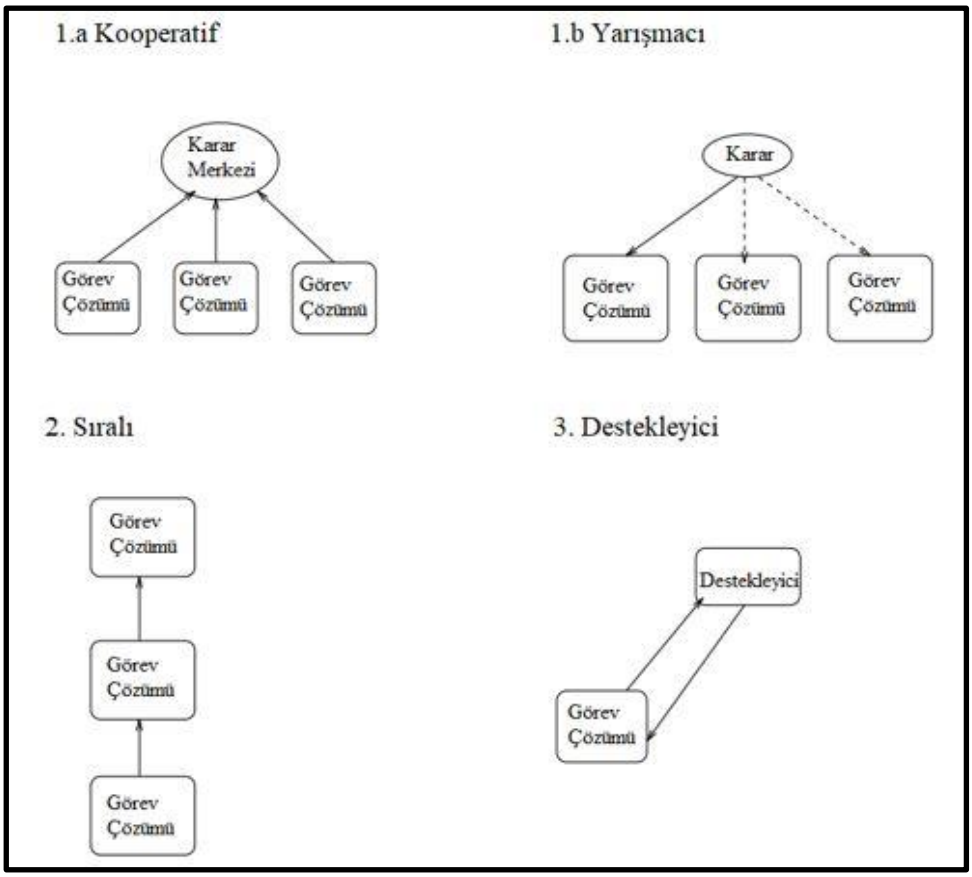
Modüler (modular) yaklaşım, ağların performansının artırılmasını esas alır. Tek bir yapay sinir ağı ile çözülmeye çalışılan bir problem, birden fazla ağın aynı problemin alt problemleri için çalıştırılması durumunda daha iyi sonuçlar elde edilmesini sağlamaktadır. Bazı durumlarda problemi basitleştirmedikçe çözmek mümkün olmamaktadır. 'Böl ve başar' prensibiyle çalışan birleşik yapay sinir ağları için ana problem belirli sayıda alt problemlere bölünmekte ve her bir problem kendisine özel kurulan ve farklı mimarilerin kullanılabildiği yapay sinir ağları ile çözülmektedir. Birleşik ağların kullanılmasının bir başka nedeni de model karmaşıklığını azaltmak ve genel sistemi daha kolay anlamak, değiştirmek ve genişletmektir. Problemin geneli için yeniden bir ağ yapılanmasına girmek yerine, alt problemlerin ağlarına müdahale ederek çözüm bulmak daha pratik bir yöntem olarak görülmektedir. Yapay sinir ağlarının modüler yaklaşımla birleştirilmesinde kullanılan yöntemler (Şekil 3.7) şu şekildedir [55]:

Kooperatif ve Yarışmacı yaklaşımlar yöntemi: Bu iki yöntem yapay sinir ağlarının birleştirmesini temelde aynı şekilde ele alsa da alt problemlerin çözümüne bakış açılarıyla birbirinden ayrılmaktadırlar. Kooperatif yaklaşımda alt problemlere bölünen ana problem için kurulan her bir alt ağın nihai çözüme bir katkısı olduğu varsayılır. Yarışmacı yaklaşımda ise alt problemleri çözmek için kurulan ağlardan amaca en uygun bileşenleri içeren ve ana ağa en çok katkısı olan ağlar seçilmektedir.

Sıralı yaklaşım yöntemi: Ağların birleştirilmesinde girdi ve çıktı verilerini esas alan bu yaklaşımda, ana problemin alt problemlerinden birinin çıktısı bir diğerinin ya da ana ağın girdisi olarak kullanılabilmektedir. Yalnızca girdi/çıktı değerleriyle kalmayıp gizli katmanların paylaşımı ya da ağlar arasına ek gizli katmanlar koyarak ta ağları sıralı olarak birleştirmek mümkündür.

Destekleyici yaklaşım yöntemi: Ana problemin çözümü için kurulan ağın performansını etkileyen çeşitli parametrelerin bir diğer ağ tarafından belirlenebilmektedir. Ağ performansını doğrudan destekleyen ağların birleştirilmesi şeklinde ifade edilen bu yöntemde, ana ağın hatasını tahmin etmek için de bir başka sinir ağı destek verebilmektedir.

Yapay sinir ağlarının birleştirilmesi ile her zaman mükemmel sonuçların elde edileceği düşünülmemelidir. Özellikle ağların birleştirilmesi çoğu zaman tekil olarak kullanılmasından daha zahmetli ve riskli adımlar içermektedir. Özellikle problemlerin alt parçalara ayrılması ve uygulanacak olan birleştirme yönteminin seçilmesi bileşik ağın performansını doğrudan etkilemektedir. Problemin karmaşık olduğu, alt bileşenlerin değişmesi ile tüm ağın etkilenebileceği, verinin gürültü içerdiği ve görüntü/sinyal işleme gibi süreklilik arz eden problemlerde ağların birleştirilerek kullanılması uygun görülmektedir [56].



Şekil 3.6. YSA modüler yaklaşım yöntemleri

Yapay sinir ağlarının birleştirilmesinde kullanılan modüler yaklaşım ve grup yaklaşımı için, YSA'da kullanılan *sınıflandırma* ve *kümeleme* terimlerinin de açıklanması gerekmektedir: *Sınıflandırma*, denetimli öğrenme yöntemini esas alan, girdilere karşılık tahmini çıktıların bilindiği, yani girdilerin belirli sınıflara ayrıldığı bir yöntemdir. Bir örnekle açıklamak gerekirse, literatürde sık karşılaşılan İris Çiçeği Türü probleminde görseller setosa, versicolor, virginica olarak bilinen 3 çiçek türü altında sınıflandırılmıştır. Bu durumda ağların birleştirilmesinde grup yaklaşımı uygulanarak, farklı ağların tahminlerinin en iyisi şeklinde bir sonuçla çiçek türünü belirlemek mümkündür.

Kümeleme ise denetimsiz öğrenme yöntemine dayanan, girdilere karşılık çıktıların olmadığı, hatta kaç küme oluşacağının bile bilinmediği bir yöntemdir. Belirli özelliklere sahip veri grubunun kendi içerisinde ayrıştırılması ve birbirine en yakın değerlerin aynı kümede bir araya getirilmesi şeklinde ifade edilmektedir. Bu durum grup yaklaşımının denetimsiz öğrenmeye dayalı problemlerde uygulanmasını zorlaştırmaktadır. Kümeleme yapan iki ayrı ağın üç farklı küme oluşturduğunu varsayarak ortak bir karar vermelerinin beklenmesi

durumu ancak ve ancak ağların aynı girdiyi aynı kümeye atmış olması durumunda mümkün olabilecektir. Literatürde yer alan kaynaklarda ‘oy birliği’ yöntemiyle çalışan grup yaklaşımının denetimli öğrenme ağları için etkili bir yöntem olduğu ancak denetimsiz ağlar için gelecekte araştırılması gereken bir konu olduğuna değinilmiştir [59].

3.9. Yapay Sinir Ağlarının Avantajları ve Dezavantajları

Yapay sinir ağları bilgiyi ağın tümünde saklamaktadırlar. Klasik programlama algoritmalarında bir veri tabanında tutulan ham bilgi, yapay sinir ağlarında ağın tümüne yayılmış durumdadır. Bu sebeple bilginin bir kısmı kaybolsa dahi bu durum ağın görevini yerine getirmesini ve genel performansını etkilememektedir. Bir sinir ağı diğer programlama yapılarından farklı olarak hızla değil kademeli olarak zaman içinde bozulma göstermektedir. Ağı oluşturan hücrelerden bir veya birden fazlasının işlevini yitirmesi durumunda dahi ağ bir çıktı üretme yeteneğine sahiptir. Bu özellik yapay sinir ağlarının hata toleransının yüksek olması şeklinde ifade edilmektedir. Yapay sinir ağları yayılmış bir hafızaya sahiptirler. Bir sinir ağının öğrenebilmesi için, ağın beklenen çıktıya ulaşabileceği örneklerle eğitilmesi gerekmektedir. Ağ bir kez öğrendikten sonra bu bilgiyi hafızasında tutabilmekte ve hiç karşılaşmadığı durumlar karşısında bile bir tahminde bulunabilmektedir. Yapay sinir ağından beklenen en iyi sonucun alınması için ağın mimarisini oluşturan bileşenlerin bir araya getirilme yöntemiyle ilgili kesin kurallar yoktur. Ağ mimarisinin oluşturulması uygulayıcının tecrübesine bağlıdır ve deneme yanılma yöntemine dayanmaktadır. Bu durum sinir ağlarına karşı şüphe oluşturmakta ve her zaman daha iyi bir sonucun olabileceği izlenimi bırakmaktadır. Kara kutu olarak anılması her ne kadar yapay sinir ağlarının dezavantajı olarak görülse de bu durum yapay sinir ağlarının yaygın olarak kullanılmasının önüne geçmemektedir ve yapılan yeni araştırmaların yakın gelecekte bu kara kutuyu açması ön görülmektedir. Benzer şekilde ağların donanımsal gereksinimi olması dezavantaj olarak görülse de gelişen teknoloji ile paralel işlemcilerin eş zamanlı olarak görev aldığı, büyük veriler içeren yapay sinir ağlarının beklenen performanslarda çalışabildiği görülebilmektedir. Yapay sinir ağları tekil durumda veya bir araya getirilerek aynı anda birden fazla görevi yerine getirebilmektedir. Makine öğrenmesinin temelini oluşturan yapay sinir ağları, öğrenmeye devam edebilme özelliği sayesinde, tüm dezavantajları bilinmesine rağmen, çağımız teknolojisinin temel taşı olarak görülmektedir [60].

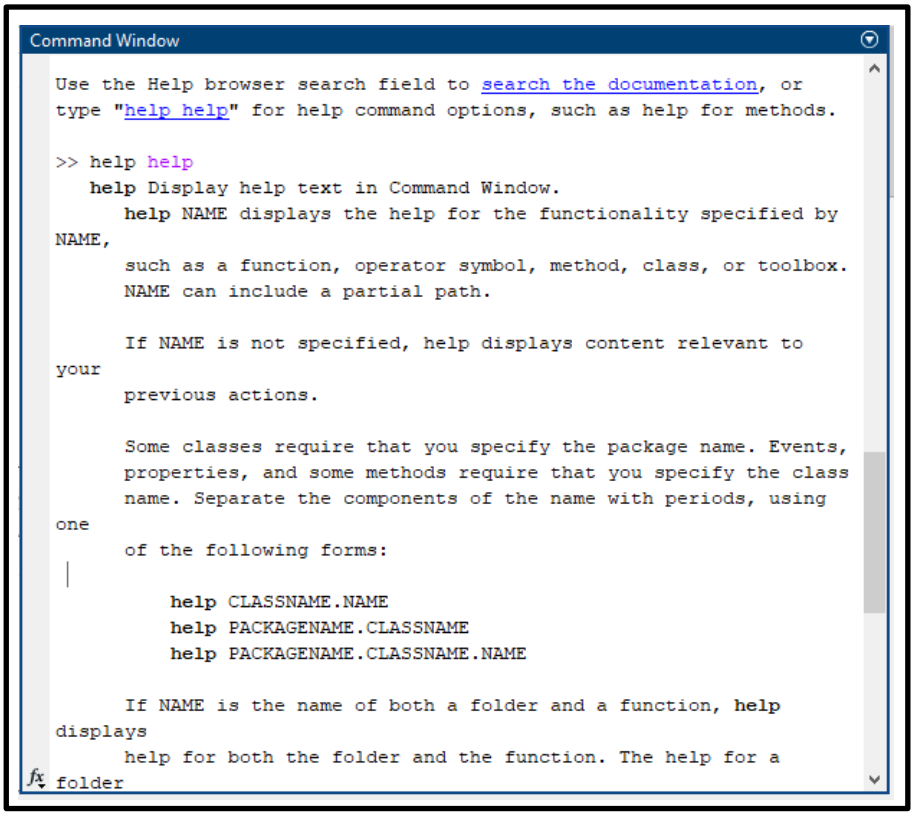
4. MATLAB

4.1. MATLAB ile Yapay Sinir Ağları

Yapay Sinir Ağları, MathWorks firması tarafından 1992'de MATLAB 4.0 yazılımına dahil edilen bir modül olarak (Neural Network Toolbox) sunulmuştur. Zaman içerisinde MATLAB ANN (Artificial Neural Network) modülü pratik şekilde uygulanabilir hale gelmiş; doğrusal olmayan ve büyük veriler içeren karmaşık problemlerin çözümü için akademide ve endüstride geniş kullanım alanı bulmuştur [61].

Fitting (nftool), Clustering (nctool), Pattern Recognition (nprtool) ve Time Series (ntstool) uygulamalarını içeren MATLAB Neural Network modülü bir yapay sinir ağı oluşturmak için matematiksel formülleri literatürde tanımlanmış tüm gerekli bileşenleri kullanıcıya sunmaktadır. Kullanım kolaylığı sayesinde, programın yönergelerini izleyerek bir yapay sinir ağı zahmetsizce oluşturabilmekte, oluşan ağın kod yapısı kaydedilerek güncellenebilmekte ya da uygulayıcının belirlediği özelliklerde bir sinir ağı oluşturmak için kodlama ekranı kullanılarak kod yazılabilmektedir. En önemlileri Havacılık, Otomotiv, Sağlık, Medikal, Bankacılık, Savunma Sanayii, Üretim olmak üzere pek çok sektörde yapay sinir ağı uygulamaları ve MATLAB programı modülleri kullanılmaktadır. Üretim sektöründe süreç kontrolü, ürün tasarım ve analizleri, gerçek zamanlı parça tanımlama, makine bakım analizleri, dinamik modelleme, gerçek zamanlı kalite kontrol başta olmak üzere geniş kullanım alanına sahip olan MATLAB, çağımız rekabet ortamında kullanıcılarına açık ara fark yaratma imkânı sunmaktadır. MATLAB programlama dili alt yapısı ve birçok farklı alanda kullanıma uygun modülleri bir arada bulundurması sayesinde tüm uygulamaları senkronize halde çalıştırmayı sağlamaktadır. Bu çalışmada olduğu gibi Neural Network modülünde yapılan çalışmalar App Designer modülünde kullanılabilmekte, Deep Learning modülüne geçilerek yapay zekâ ile çalışan bir sistem yapısı kurulabilmektedir. Veri işleme çalışmalarında programlama dillerinin kullanımı kaçınılmazdır. Bu durum uygulayıcıları zorlamakta, pratik şekilde çözülebilecek problemleri kodlama gereğiyle karmaşık hale getirebilmektedir. Bu noktada MATLAB'ın en büyük avantajlarından birisi kodlama bilgisine sahip olmayan uygulayıcılara dahi yönlendirmeler ile programı kullanabilme becerisi kazandırmasıdır.

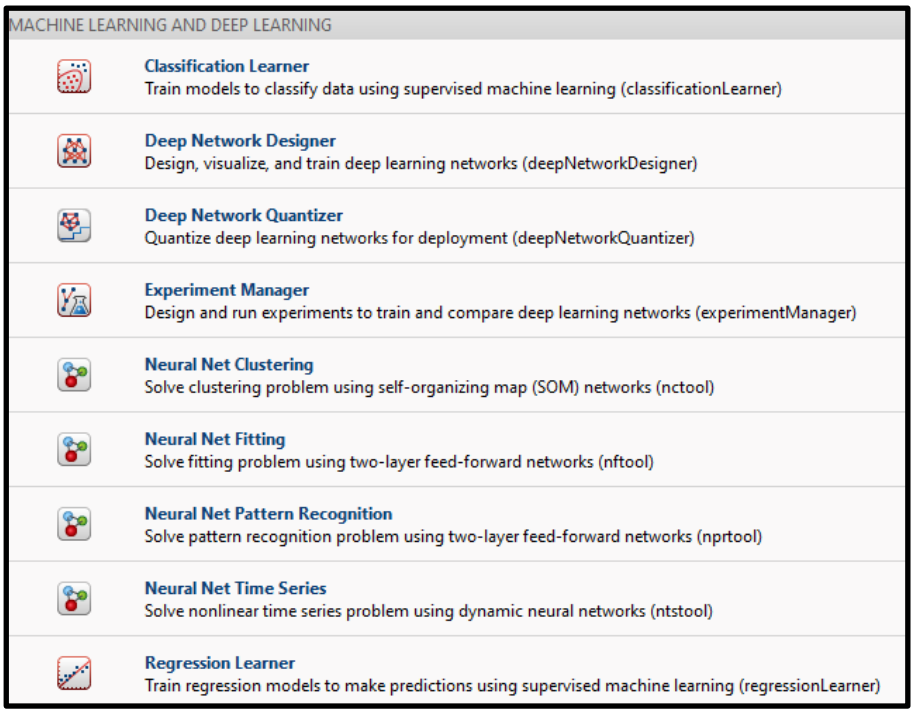
Resim 4.1'de MATLAB Command Window ekranında yardım talep edilen *help* başlığının yazılmasıyla hem ekran üzerinden hem de çevrimiçi kaynaklarda yardım konularına ulaşılabilmektedir.



Resim 4.1. MATLAB yardım komutu çalıştırma

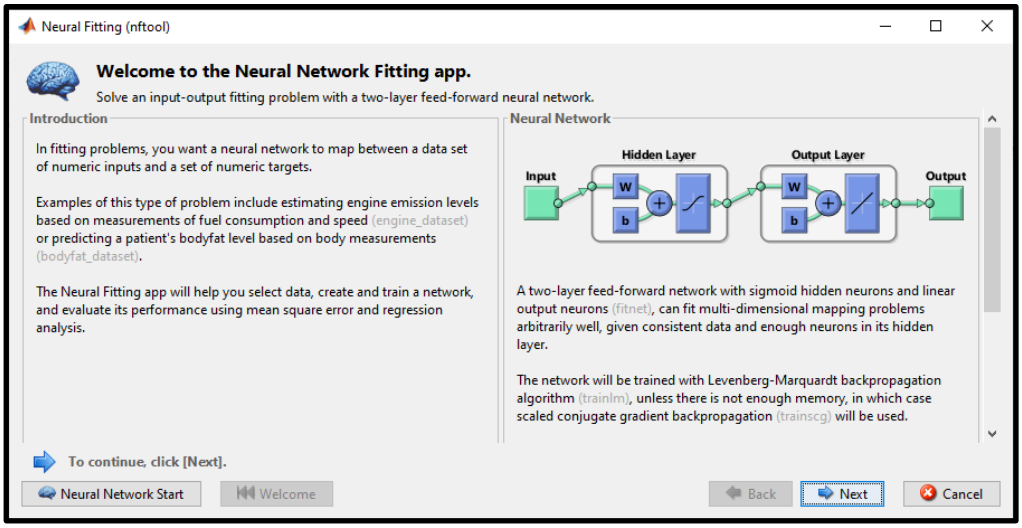
MATLAB Yapay Sinir Ağları modülünün içerdiği bazı önemli özellikler aşağıda listelenmiştir.

- Derin öğrenme araçları, sinir ağları ve autoencoder uygulamaları
- Paralele işleme ve GPU desteği
- Çok katmanlı, radial basis, LVQ, RNN (Recurrent Neural Network) ağları gibi denetimli öğrenme algoritmaları
- SOM (Self-Orginizing Map) ve CNN (Competitive Neural Network) ağları gibi denetimsiz öğrenme algoritmaları
- Veri işleme, örüntü sınıflandırma ve kümeleme uygulamaları
- Kontrol sistemi uygulamaları ve yapay sinir ağları için Simulink desteği



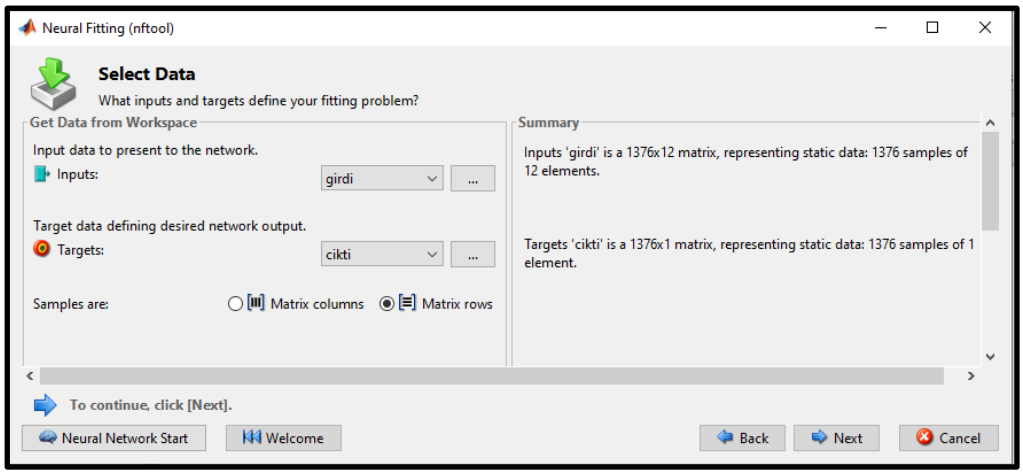
Resim 4.2. MATLAB makine öğrenmesi ve derin öğrenme uygulamaları

Makine Öğrenmesi ve Derin Öğrenme (Machine Learning and Deep Learning) modülünde yer alan uygulamalar Resim 4.2'de yer almaktadır [62]. Neural Net Fitting ve Neural Net Clustering uygulamalarına bu ana ekrandan ulaşılabileceği gibi Command Window ekranında *nftool* / *nctool* yazılarak ta ulaşılabilmektedir. Yönergeler takip edilerek, kod yazmaya gerek kalmaksızın basit olarak bir yapay sinir ağı kurulabilmektedir. İlgili ekran Resim 4.3'te verilmiştir.



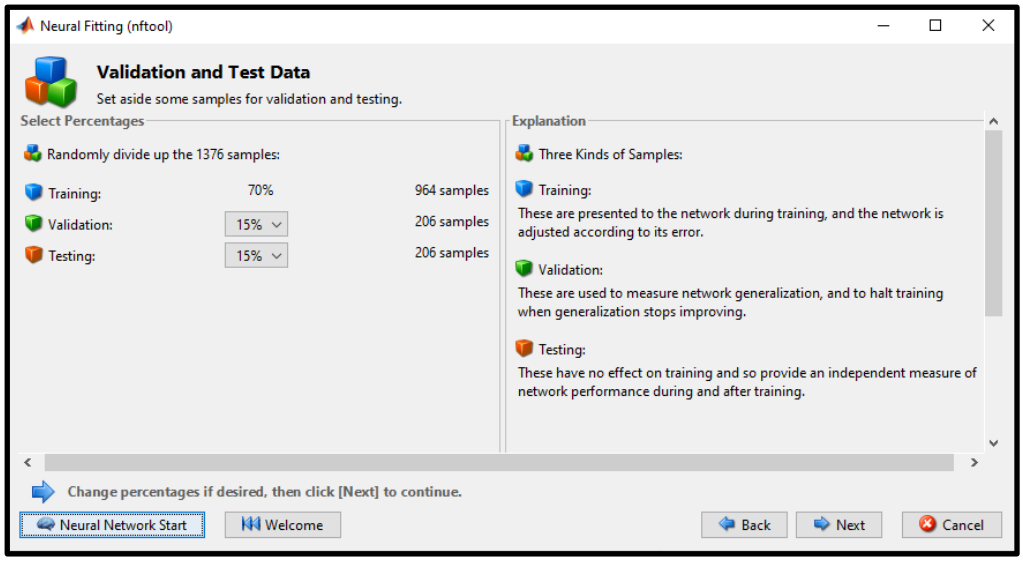
Resim 4.3. MATLAB nftool başlangıç ekranı

Yönergeleri takip ederek ilerlenir ve ağın veri dosyaları Inputs/Targets bölümlerinden seçilir. Resim 4.4'te yer alan Veri Seçme ekranında *girdi* ve *cikti* olarak adlandırılan veri dosyalarının seçili olduğu görülmektedir. Seçilen veri dosyası sayısal değer içeren verilerden oluşmalı ve .xls formatında olmalıdır.



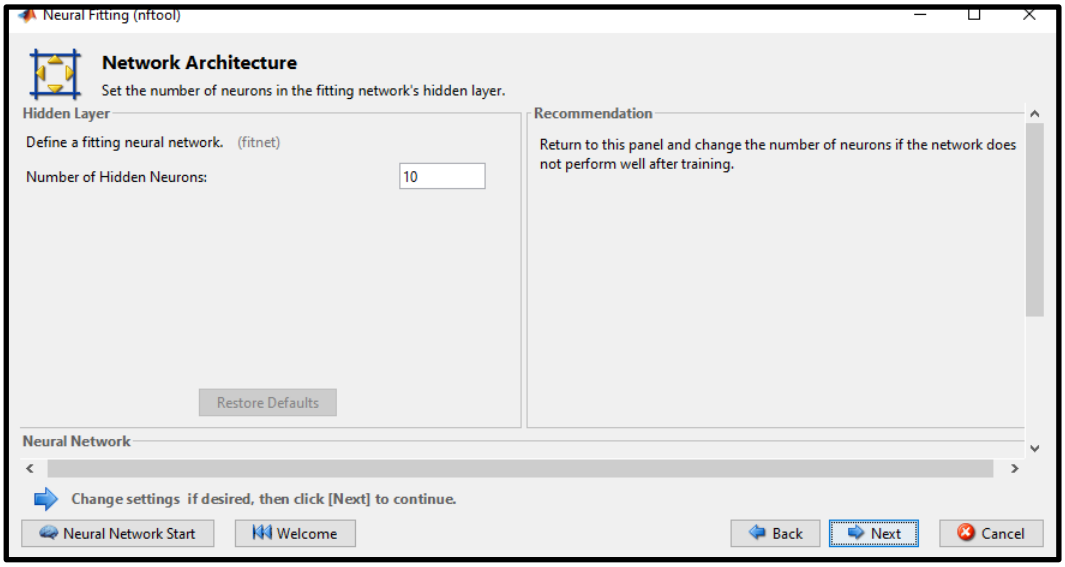
Resim 4.4. MATLAB nftool veri giriş ekranı

Verilerin seçilmesinden sonra Resim 4.5'teki Validasyon ve Test Verisi ekranında görüldüğü gibi eğitim, validasyon ve test için kullanılacak olan verilerin oranı yüzdesel olarak manuel ayarlanabilmektedir. Literatürde yer alan %70 eğitim veri seti oranı varsayılan olarak gelmektedir.



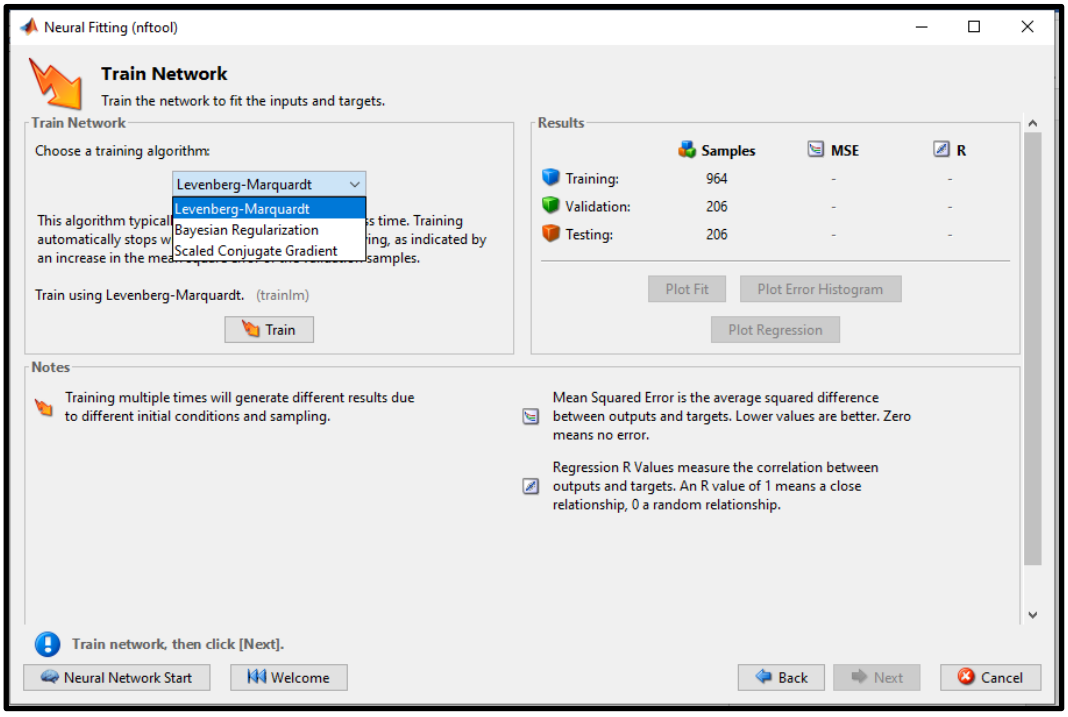
Resim 4.5. MATLAB nftool validasyon ve test verisi ekranı

Resim 4.6 Ağ Mimarisi ekranında görülen gizli katman nöron sayısı varsayılan olarak 10 görülmektedir. Literatürde yer alan yaklaşımlar esas alınarak, uygulayıcının inisiyatifinde yine manuel olarak değiştirilebilmektedir.



Resim 4.6. MATLAB nftool ağ mimarisi ekranı

MATLAB *nftool*'da kullanıcıya varsayılan olarak sunulan eğitim algoritmaları Ağın Eğitimi ekranı olan Resim 4.7'de görülmektedir.



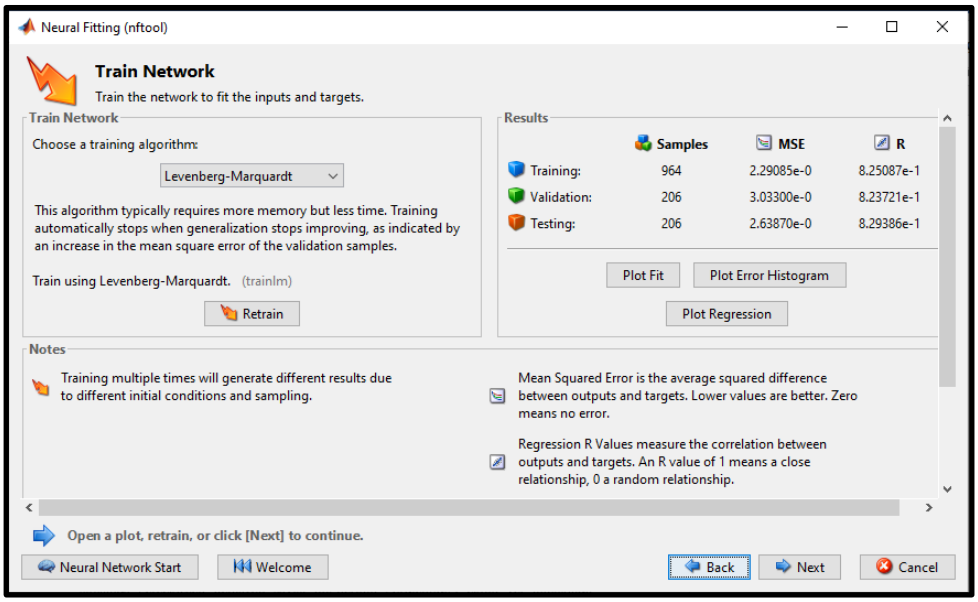
Resim 4.7. MATLAB nftool ağın eğitimi ekranı

MATLAB Neural Network modülünde yer alan diğer eğitim fonksiyonları, kullandıkları algoritmalar ve öncelikli kullanım alanları Çizelge 4.1'de verilmiştir.

Çizelge 4.1. MATLAB eğitim fonksiyonları ve öncelikli kullanım alanları

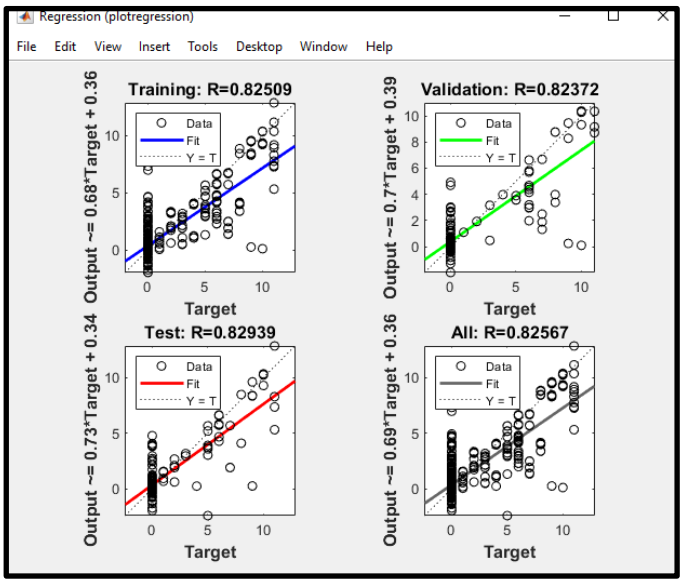
Eğitim Fonksiyonu	Algoritması	Öncelikli Kullanım Amacı
trainlm	Levenberg-Marquardt	Regresyon
trainbfg	BFGS Quasi-Newton	Regresyon
trainbr	Bayesian Regularization	Örüntü Tanıma
trainrp	Resilient Backpropagation	Örüntü Tanıma
trainscg	Scaled Conjugate Gradient	Örüntü Tanıma
traincgb	Conjugate Gradient with Powell/Beale Restarts	Regresyon/Örüntü Tanıma
traincgf	Fletcher-Powell Conjugate Gradient	Regresyon/Örüntü Tanıma
traincgp	Polak-Ribiére Conjugate Gradient	Regresyon/Örüntü Tanıma
trainoss	One Step Secant	Regresyon/Örüntü Tanıma
traingdx	Variable Learning Rate Gradient Descent	Regresyon/Örüntü Tanıma
traingdm	Gradient Descent with Momentum	Regresyon/Örüntü Tanıma
traingd	Gradient Descent	Regresyon/Örüntü Tanıma

Train tuşuna basılmasıyla eğitim başlayacak ve sonuçlanmasıyla MSE ve R değerleri Resim 4.8'de verildiği şekilde ekranda görülecektir. *Hata histogramı* ve Regresyon sonuçlarını *Plot* seçenekleriyle grafiksel olarak görüntülemek mümkündür.



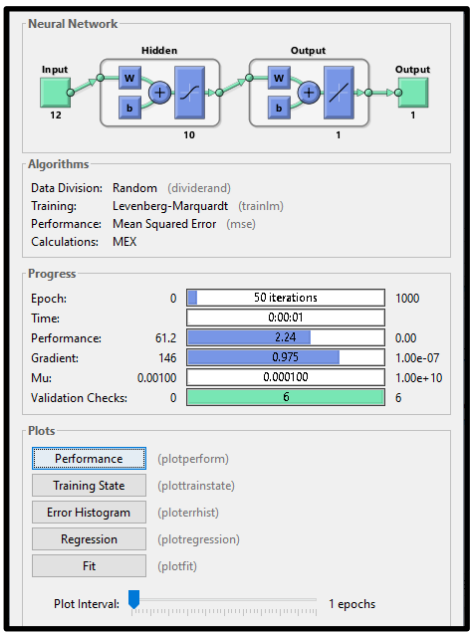
Resim 4.8. MATLAB nftool ağın eğitimi sonuç ekranı

Yapılan örnek çalışmaya ait regresyon grafikleri Resim 4.9'da görülmektedir.

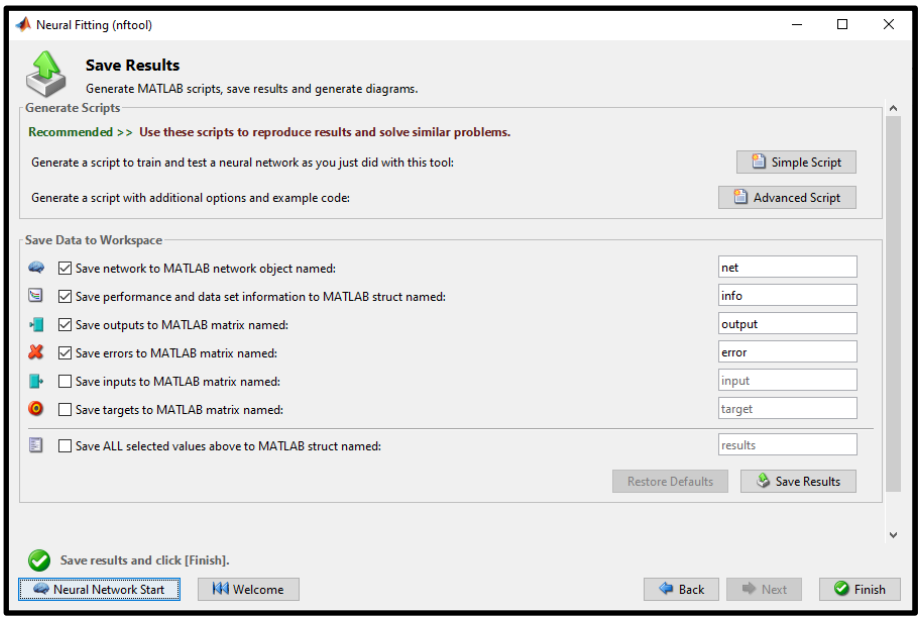


Resim 4.9. MATLAB nftool regresyon grafik sonuçları

Resim 4.10 ağın eğitiminin gerçekleştiği ekranı göstermektedir. Ekranın en üst kısmında kurulan ağın şematik görünümü yer almaktadır. Varsayılan olarak seçilen veri seti dağılım oranı *Random* olarak görülmektedir. Ardından sırasıyla ağın eğitim algoritması (*trainlm*) ve performans ölçüm değeri (*mse*) yer almaktadır, Ağın eğitimi sırasında gerçekleşen iterasyon sayısı ve eğitim süresi bilgileri de bu ekranda görülebilmektedir.

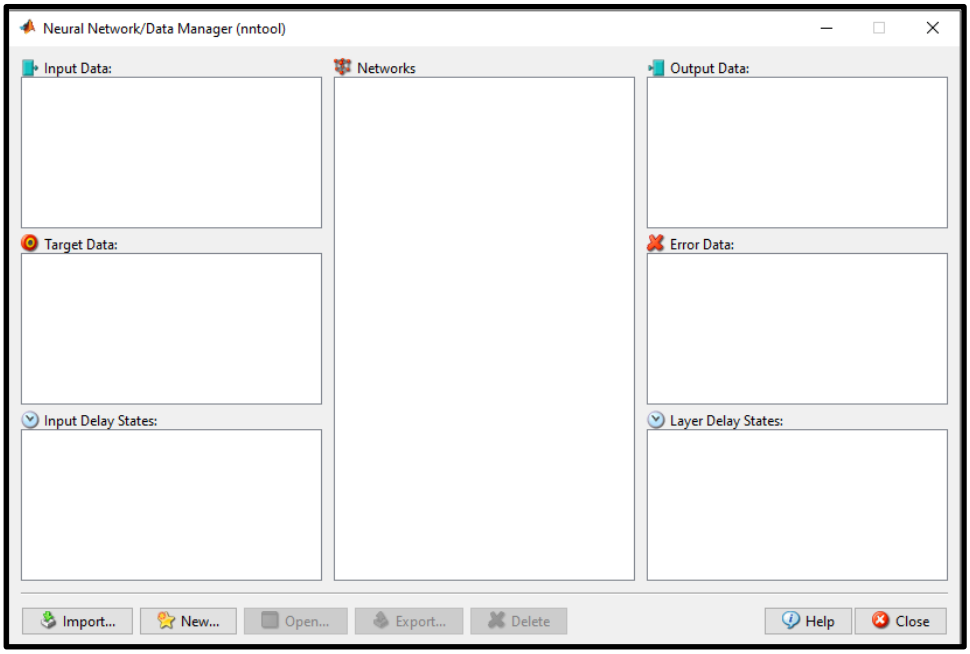


Resim 4.10. MATLAB nftool ağın eğitimi ekranı



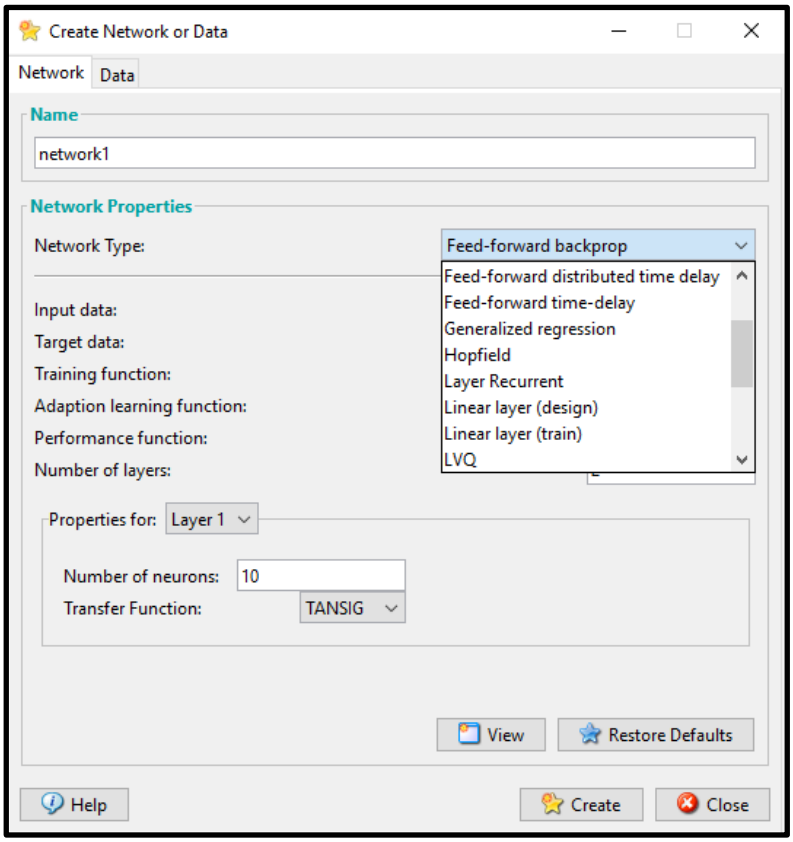
Resim 4.11. MATLAB nftool çalışmanın kaydedilmesi ekranı

Resim 4.11'de görüldüğü gibi çalışmanın sonuçları kaydedilebilmektedir. Program tarafından otomatik olarak oluşturulan kodlara ulaşılmakta ve gerekli görülen düzenlemeler yapılabilmektedir. Ayrıca Neural Net Fitting uygulamasına MATLAB tarafından yeni sürümlerde uygulamadan kaldırılacağı bildirilen Resim 4.12'de görülen Neural Network/Data Manager (*nntool*) uygulamasına Command Window üzerinden *nntool* komutu ile ulaşılabilmektedir.



Resim 4.12. MATLAB nntool ekranı

Bu ekranda Input/Target data verileri seçildikten sonra *New* butonu ile sinir ağının mimarisinin belirleneceği Resim 4.13'te verilen ekrana geçilmektedir. *nftool'*dan farklı olarak bu ekran ve sonrasında ağın ağırlık ve bias değerleri görülüp ayarlanabilmekte, katmanların nöron sayıları ve transfer fonksiyonları ayrı ayrı seçilebilmekte, ağın tüm parametre değerleri değiştirilebilmektedir [63].

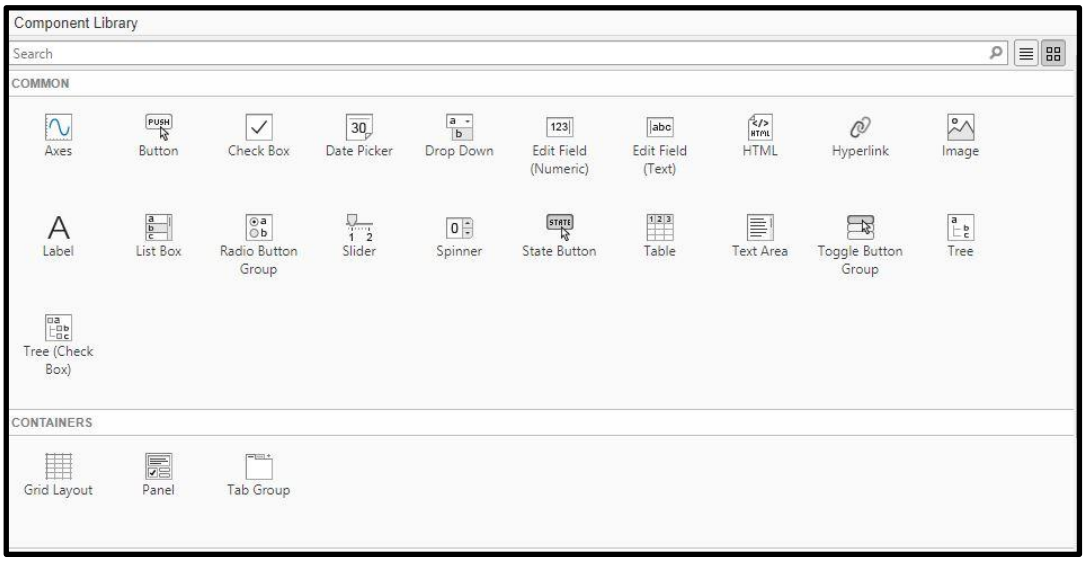


Resim 4.13. MATLAB nntool ağ mimarisi ekranı

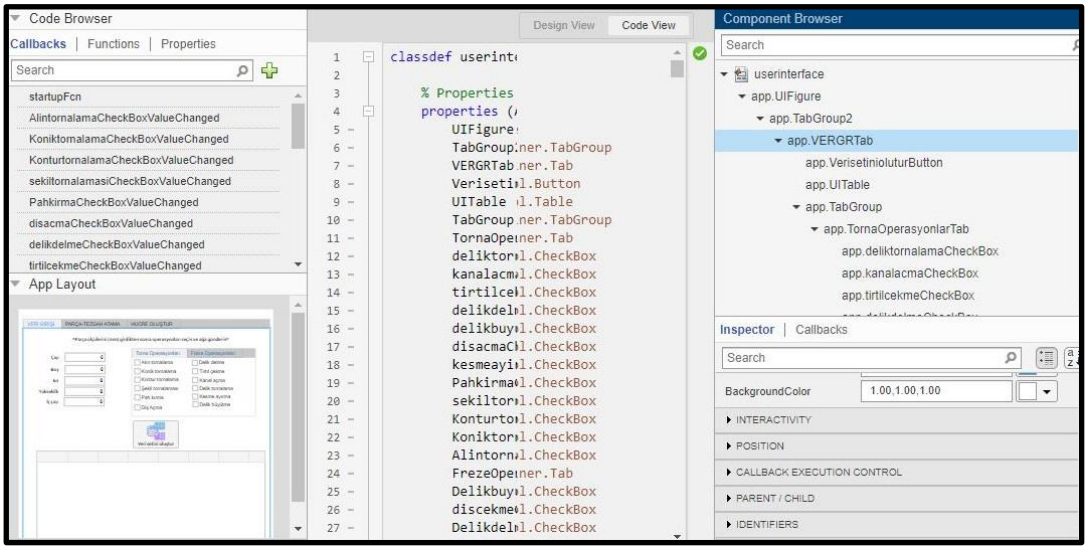
4.2. MATLAB App Designer

MATLAB App Designer, profesyonel bir yazılım geliştiricisi olmanıza gerek kalmadan profesyonel uygulamalar oluşturmaya imkân vermektedir. Sürükle bırak özelliğiyle pratik olarak kullanılabilen görsel bileşenleri sayesinde Grafik kullanıcı ara yüzü (GUI) tasarımını düzenlenmekte ve entegre olarak çalışan kod ekranı ile geliştirilebilmektedir. Oluşturulan programlar MARLAB Drive ile paylaşılabilmekte, istenirse masaüstü ya da web uygulaması şeklinde MATLAB Complier ile yayınlanabilmektedir. Program dahilinde çalışan *Code Analyzer* sayesinde uyarı ve hata mesajlarını takip ederek, uygulama için yazılan kodları düzeltmek mümkündür. Resim 4.14'te bir kısmı görünen bileşen kütüphanesinde buton,

seçim kutusu, ağaç, açılır menü gibi standart bileşenlerin yanında kalibre düğmesi, kontrol düğmesi, kontrol ışığı, şalter gibi bileşenlerle aygıt kontrolleri görselleştirmek mümkündür.



Resim 4.14. MATLAB App Designer bileşen kütüphanesi



Resim 4.15. App Designer kod penceresi görünümü

App Designer detaylı ve anlaşılır kullanım kılavuzu sayesinde uygulayıcıya kolaylık sağlamaktadır. Mevcut çalışmada geliştirilen programın App Designer kod penceresi görünümü Resim 4.15'te verilmiştir. Aynı ekranda tasarım görünümü, her bir bileşen için eklenen kodlar, bileşenlerin özelliklerinin değiştirilebildiği pencere ve tasarımın genel kod şablonu görülebilmektedir [64].

5. YÖNTEM VE ARAÇLAR

Dinamik sanal imalat hücreleri oluşturmak için operasyon bazlı tezgâh atama ve kümeleme yapan birleşik yapay sinir ağlarının kurulduğu bu çalışmada, çözüm aracı olarak MATLAB R2021a programı Neural Network (NN) ve App Designer modülleri kullanılmıştır. İlk olarak ele alınan problem için varsayımlar ve kısıtlar belirlenmiştir. Ardından yapay sinir ağlarının eğitiminde kullanılmak üzere veri seti hazırlanmıştır. Birinci aşamada NN Fitting uygulamasında ileri beslemeli geri yayılımlı bir yapay sinir ağı kurulmuştur. Bu yapay sinir ağı parça, tezgâh ve operasyon bilgilerini kullanarak parçaları tezgâhlara atamaktadır. Bu ağdan elde edilen sonuçlar, NN Clustering uygulamasında kurulan rekabetçi yapay sinir ağı nın girdisi olarak kullanılmak üzere revize edilmiştir. İkinci aşamada kurulan rekabetçi yapay sinir ağı imalat hücreleri oluşturmaktadır. Son aşamada ise birleşik ağların kurulmasında modüler yaklaşım esas alınarak; önce operasyonları tezgâhlara, sonra tezgâhları hücrelere atayan farklı tiplerdeki bu iki yapay sinir ağı, App Designer modülünde geliştirilen özel bir programda birleştirilmiştir. Geliştirilen program verilerin toplandığı işletmede denenmiş ve bir örnek uygulama yapılmıştır.

5.1. Varsayımlar ve Kısıtlar

Bu çalışmada kullanılan tüm veriler medikal sektöründe üretim yapan orta-büyük ölçekli bir firmada toplanmıştır. İşletmenin talaşlı imalat biriminin mevcut yerleşim düzeni sabit konumludur ve ana üretim planı 2 aylık periyotlar halinde hazırlanmaktadır Literatürdeki kaynaklar esas alınarak bu çalışma kapsamındaki varsayımlar ve kısıtlar aşağıda belirtilmiştir [11-65]:

Tezgâhlar: İşletmede kullanılan tüm tezgâhlar ve teknik özellikleri (eksen, takım tutucu, takım sayısı, işleme kapasitesi vd.) bilinmektedir. Bir üretim periyodu boyunca tüm parçaların operasyonları için tüm tezgâhların süre bazında çalışma kapasiteleri yeterlidir. Tüm parçaların operasyon tipleri için makinelerin kabiliyetleri yeterli ve belirlidir. İşletmedeki tezgâhların yerleşim planı sabittir ve bir planlama dönemi boyunca değişmesi mümkün değildir. Çizelge 5.1’de tezgâh bilgileri görülmektedir.

Operatörler: Operatör sayısı her bir makinedeki her operasyon için yeterlidir ve herhangi bir işe atanmış operatör süreci yürütebilecek beceriye sahiptir.

Çizelge 5.1. Tezgâh bilgileri

Tezgâh Kodları		Tezgâh Bilgileri	Eksen Ölçüleri (mm)				
İşletme	YSA	Marka Model / Özellik	X	Y	Z	C	B
T1	501	Mıcrocut Mcv 2412 / Cnc Dik İşlem Merkezi (3 Eksen) 400x600 Mm 20 Kva	610	305	520	0	0
T2	502	Mıcrocut Mcv 2418 / Cnc Dik İşlem Merkezi (3 Eksen) 400x600 Mm 20 Kva	610	455	510	0	0
T3	503	Mıcrocut Mcv 2418 / Cnc Dik İşlem Merkezi (4 Eksen) 400x600 Mm 20 Kva	610	455	510	0	0
T4	504	Yang Sl200 / Cnc Torna 8" 240x445 Mm 12 Kw	480	240	520	0	0
T5	505	Yang Sl12g / Cnc Torna 6" Ø150x155 Mm 9 Kw	330	135	150	0	0
T6	506	Haas Ds-30ssy / Cnc Torna 8" C Eksen 800x50,8 Mm 380 V	318	51	584	0	0
T7	507	Dmg Ctx Gıldemeıster 500 / Cnc Torna 1xø30 Mm 25 Kva	210	80	525	0	0
T8	508	Dmg50 Ecomıll 50 / Cnc Dik İşleme Merkezi Ø500 Mm 17 Kva	500	450	400	360	115
T9	509	Dmg Morı Mılltap 700 / 5 Eksen İmalat Tezgâhı Yüksek Devirli 25 Kva	700	420	380	200	200
T10	510	Dmg Morı Mılltap 700 / 5 Eksen İmalat Tezgâhı Yüksek Hassasiyetli 19 Kva	700	420	380	200	200
T11	511	Mekay Mekamat-5 / Otomat Torna Ø32x150 Mm 5,5 Kw	440	200	130	0	0

Parçalar: Talaşlı imalat biriminde üretilen parça çeşidi ve sayısı belirlidir. Her bir parça için üretim maliyeti bilinmektedir ancak herhangi bir periyotta o parçaya olan talep dinamiktir. Her bir parça için bağlama aparatı, mengene, fikstür mevcut ve belirlidir.

Operasyonlar: Her bir parçanın operasyonu bir veya birden fazla tezgâhta gerçekleşebilir. Her bir parçanın her bir operasyonu için hazırlık ve sökme-takma zamanları bilinmektedir ancak operasyon süresinden küçük olduğu müddetçe göz ardı edilebilir. Çalışmada ele alınan operasyonlar ve YSA için belirlenmiş kodlar Çizelge 5.2'de verilmiştir.

Taşıma: Makineler arasındaki mesafe ve malzeme taşıma süreleri belirlidir. Eğer gerekliyse parçanın aynı makineye geri dönmesine izin verilir. Mevcut bir otomatik taşıma sistemi

yoktur ve taşıma işlemi klasik yöntemlerle yapılmaktadır. Parçanın taşıma süresi operasyon süresinden kısaysa göz ardı edilebilir.
Hücreler: Sanal hücreler her bir parçanın operasyonu baz alınarak kurulacaktır. Üretim sona erdiğinde hücre dağılabilir veya hücredeki bir tezgâh başka bir hücreye geçebilir. Operatör ise her aşamada hücre içinde ve hücreler arasında yer değiştirebilir.

Talep: Dinamiktir, bir üretim dönemi boyunca talepte değişkenlikler olabilmektedir. Sistem bu değişkenliğe cevap verebilecek şekilde tasarlanmıştır.

Çizelge 5.2. Operasyonlar ve YSA kodları

Torna operasyonları ve YSA kodları		Freze operasyonları ve YSA kodları	
Alın tornalama	10001	Silindirik frezeleme	20001
Konik tornalama	10002	Kanal açma	20002
Kontur tornalama	10003	Yan frezeleme	20003
Şekil tornalaması	10004	Çifte frezeleme	20004
Pah kırma	10005	Biçimlendirme	20005
Kesme (ayırma)	10006	Geleneksel alın frezeleme	20006
Diş açma	10007	Kısmi alın frezeleme	20007
Delik büyütme	10008	Uç frezeleme	20008
Delik delme	10009	Profil frezeleme	20009
Tırtıl çekme	10010	Cep frezeleme	20010
Kanal açma	10011	Yüzey şekillendirme	20011
Delik tornalama	10012	Delik delme	20012
		Diş çekme	20013
		Delik büyütme	20014

5.2. Veri Setinin Hazırlanması

Yapay sinir ağlarının sadece sayısal değerlerle çalışıyor olması, veri setinin hazırlanması sırasında, sayısal olmayan değerlere sayısal değerler atanmasını gerektirmektedir. Girdi sayısı ve çeşidinin fazla olduğu veri setlerinde kodlama sisteminin belirli bir düzende ilerlemesi, atanan sayısal değerlerin birbiri ile çakışmasını önlemektedir. Çizelge 5.3'te görüldüğü gibi çalışmada sırası ile torna operasyonları 1, freze operasyonları 2, malzeme bilgileri 4, tezgâh bilgileri 5 numara altında gerekli bilgilere göre değer alarak kodlanmıştır.

Çalışma kapsamında uygulama işletmesinin talaşlı imalat atölyesinde en çok üretilen ve 26 farklı torna ve freze operasyonu için örnek teşkil edebilecek 34 parça ele alınmıştır. Parçaların tamamı için teknik resimler üzerinde CAM verilerinden yola çıkarak tüm operasyonlar belirlenmiştir. YSA'nın örneklerden öğrenme özelliği sebebiyle, ağın eğitilmesi için kullanılan veri setinde tüm operasyonların yer almasına ve her bir örnekten farklı ölçülerde parçalar olmasına dikkat edilmiştir. Uygulama firmasının gizlilik koşulları sebebiyle, ölçüleri ve operasyonları belirlenerek yapay sinir ağının eğitim veri setinde kullanılan 34 parçadan sadece 5 tanesinin teknik resmi örnek olarak eklerde verilmiştir.

Çizelge 5.3. Veri setinde yer alan örnekler

Malzeme Bilgileri					Operasyon Bilgileri	Tezgah Bilgileri						YSA çıktısı
Çap	Boy	En	Yükseklik	İç Çap	Operasyon Kodu	Tezgah Kodu	X	Y	Z	B	C	Tezgah kodu
41014	42014	43000	44000	45000	10003	11	511440	511200	511130	511000	511000	11
41014	42014	43000	44000	45000	10003	10	510700	510420	510380	510200	510200	0
41014	42014	43000	44000	45000	10003	9	509700	509420	509380	509200	509200	0
41014	42014	43000	44000	45000	10003	8	508500	508450	508400	508360	508115	0
41014	42014	43000	44000	45000	10003	7	507210	507080	507525	507000	507000	7
41014	42014	43000	44000	45000	10003	6	506318	506051	506584	506000	506000	6
41014	42014	43000	44000	45000	10003	5	505330	505135	505150	505000	505000	5
41014	42014	43000	44000	45000	10003	4	504480	504240	504520	504000	504000	4
41014	42014	43000	44000	45000	10003	3	503610	503455	503510	503000	503000	0
41014	42014	43000	44000	45000	10003	2	502610	502455	502510	502000	502000	0
41014	42014	43000	44000	45000	10003	1	501610	501305	501520	501000	501000	0
41030	42060	43000	44000	45000	10002	11	511440	511200	511130	511000	511000	11
41030	42060	43000	44000	45000	10002	10	510700	510420	510380	510200	510200	0
41030	42060	43000	44000	45000	10002	9	509700	509420	509380	509200	509200	0
41030	42060	43000	44000	45000	10002	8	508500	508450	508400	508360	508115	0
41030	42060	43000	44000	45000	10002	7	507210	507080	507525	507000	507000	7
41030	42060	43000	44000	45000	10002	6	506318	506051	506584	506000	506000	6
41030	42060	43000	44000	45000	10002	5	505330	505135	505150	505000	505000	5
41030	42060	43000	44000	45000	10002	4	504480	504240	504520	504000	504000	4
41030	42060	43000	44000	45000	10002	3	503610	503455	503510	503000	503000	0

Toplanan verilerle oluşan 2027 satırlık veri setinde yer alan örneklerden bir kısmı Çizelge 5.3'te görülmektedir. Veri setinin sütunlarında yer alan bilgiler ve bulunma gerekçeleri şu şekildedir:

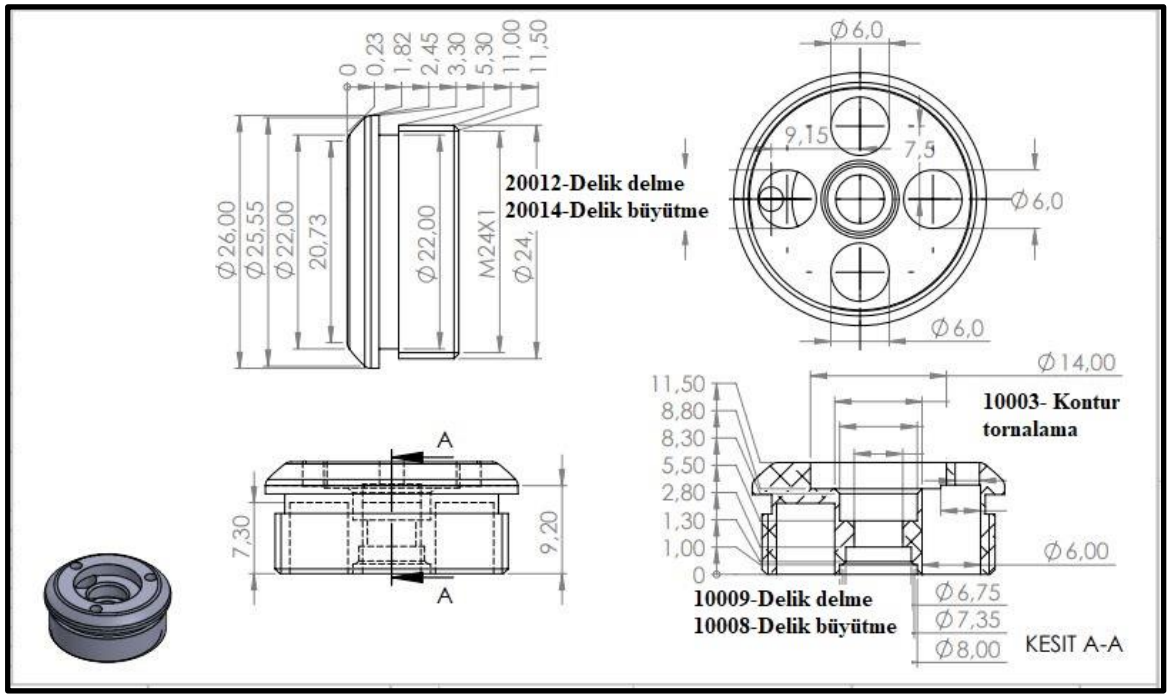
Parça ölçüleri: Kurulan ağ parça ölçüsü bilgisi ile tetiklenmektedir. Veri setinde parçaların çap, boy, en, yükseklik ve iç çap ölçüsü mm cinsinden verilmiştir.

Parçanın operasyonları: Ağın temel hedefi parçanın operasyonlarının gerçekleştirilebileceği uygun tezgâhı bulmaktır. Yapay sinir ağlarının dezavantajlarından olan sadece nümerik verilerle çalışması durumu dikkate alınarak alın tornalama, delik büyütme vb. sayı içermeyen ifadeler için Çizelge 5.2'de verilen sayısal değerler atanmıştır.

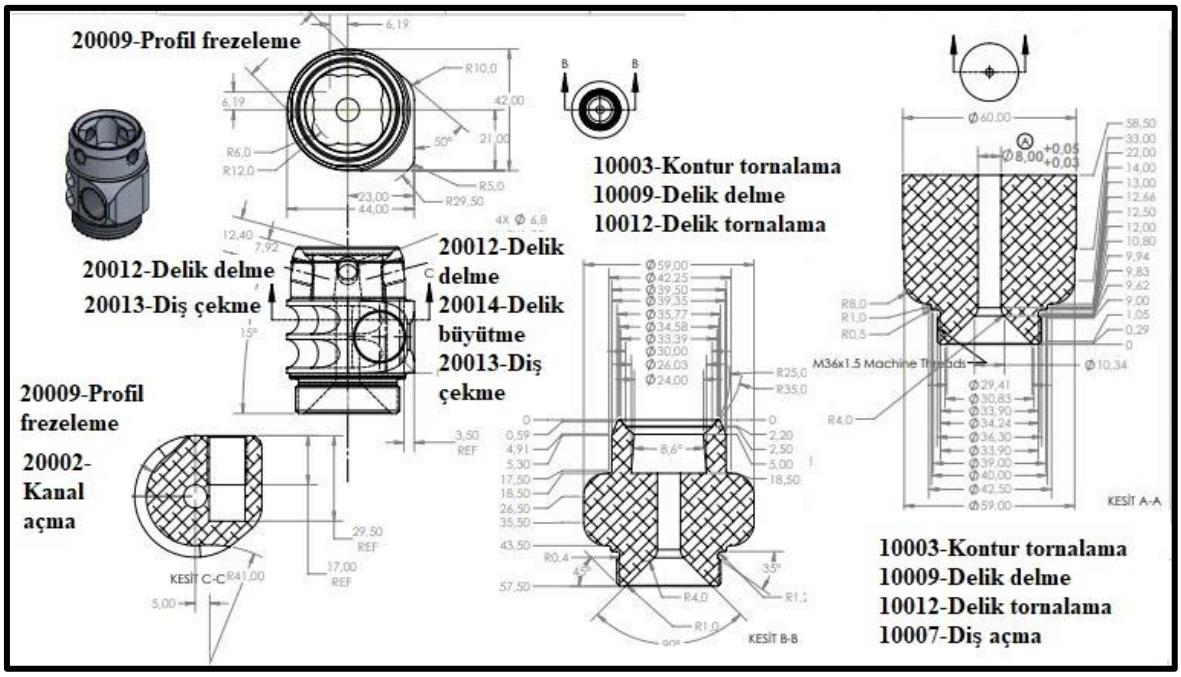
Resim 5.1 ve Resim 5.2'de verilen iki örnek parçaya ait teknik resimler dikkate alınarak, parçanın operasyonları için belirlenmiş YSA operasyon kodları görülmektedir.

Tezgâh bilgileri: Bu noktada yine YSA'nın bir dezavantajı olan problemin ağa gösterimi konusu dikkate alınmıştır. Girilen parça ölçüleri ve operasyon bilgisine karşılık atölyede mevcut 11 tezgâh ve her birinin eksen ölçüleri, tüm alternatifleri ağa göstermek üzere veri setinde yer almaktadır.

YSA çıktısı: 13.sutunda yer alan tezgâh kodu bilgileri, ağın tahmin etmesi beklenen çıktı olan, verilen parça ölçüleri, parçanın operasyonları ve atölyede mevcut tüm tezgâhlara göre parçanın işlenebileceği tezgâhı göstermektedir. Sütunda yer alan 0 değeri parçanın o tezgâhta işlenmediğini ifade etmektedir.



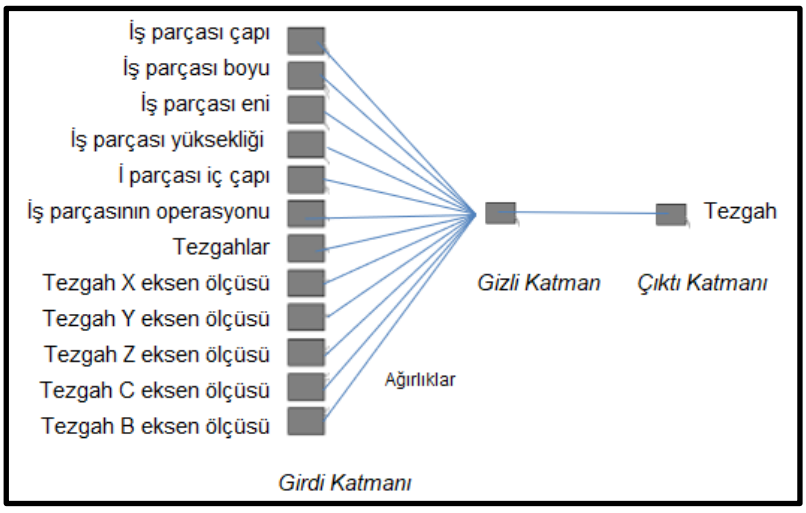
Resim 5.1. Parça 1 teknik resmi, operasyon kodları ve tanımları



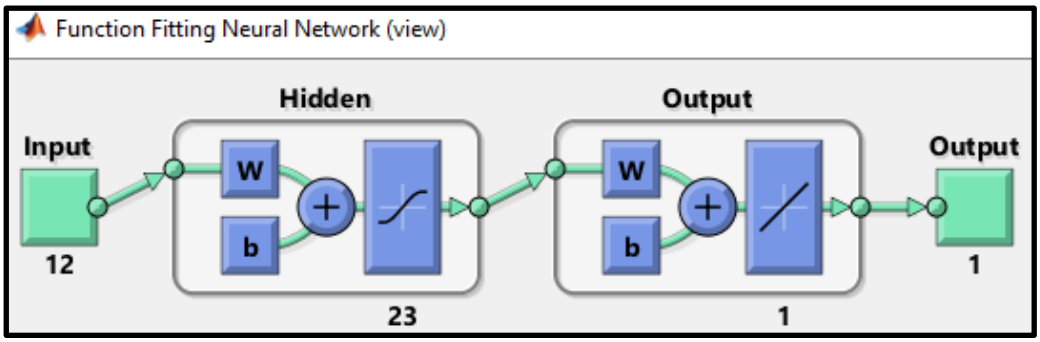
Resim 5.2. Parça 2 teknik resmi, operasyon kodları ve tanımları

5.3. Aşama 1- Operasyon Bazlı Tezgâh Atama Problemi

Problemin çözümü için tek katmanlı ileri beslemeli geri yayılımlı bir yapay sinir ağı kurulmuş ve veri setindeki örnekler ağa 12 girdi, 1 çıktı şeklinde sunulmuştur. Oluşturulan yapay sinir ağından beklenti, verilen örneklerden yola çıkarak, iş parçasının hangi tezgâhta işlenebileceğini tahmin etmesidir. Şekil 5.1'de kurulan YSA'nın yapısı ve Şekil 5.2'de MATLAB ağ görüntüsü verilmiştir.



Şekil 5.1. Kurulan ağın yapısı



Şekil 5.2. Ağın MATLAB şematik görünümü

Çalışmada ağın eğitimi için yapılan denemelerde kullanılan 5 farklı eğitim algoritması şu şekildedir:

1- Levenberg-Marquardt (*trainlm*)
2- BFGS Quasi-Newton (*trainbfg*)
3- Resilient Backpropagation (*trainrp*)
4- Scaled Conjugate Gradient (*trainscg*)
5- Fletcher-Powell Conjugate Gradient (*traincgf*)

Nöron sayıları önceki bölümlerde bahsedilen 3 temel yaklaşıma dayanarak 9, 12 ve 23 olarak belirlenmiştir. Yapılan deneme sayısı ve eşik değerin üstündeki gizli katman nöron sayısı uygulayıcının inisiyatifine bağlı olarak artırılmış ve farklı zamanlarda toplam 172 deneme yapılarak eğitim ve validasyon için korelasyon katsayı değerleri (R) kayıt altına alınmıştır. Korelasyon katsayısı değeri -1 ile +1 arasında değişmekte ve ağın tahmin ettiği çıktı ile gerçek çıktıların ne kadar iyi eşleştiğini ifade etmektedir. Değerin +1'e yakın olması ağ performansının yüksek olduğunu göstermektedir.

Çizelge 5.4. Elde edilen en iyi sonuçlar

Eğitim Fonksiyonu	Gizli Katman Nöron Sayısı	Eğitim R değeri	Validasyon R değeri
trainlm	9	0,85	0,89
trainlm	12	0,90	0,89
trainlm	*23*	*0,92*	*0,91*
trainlm	35	0,88	0,88
trainbfg	9	0,79	0,80
trainbfg	12	0,82	0,81
trainbfg	23	0,87	0,85
trainbfg	35	0,83	0,89
trainrp	9	0,65	0,66
trainrp	12	0,70	0,68
trainrp	23	0,59	0,68
trainrp	35	0,63	0,79
trainscg	9	0,67	0,66
trainscg	12	0,74	0,77
trainscg	23	0,77	0,78
trainscg	35	0,79	0,81
traincgf	9	0,53	0,55
traincgf	12	0,67	0,60
traincgf	23	0,6	0,69
traincgf	35	0,68	0,75

Her bir eğitim fonksiyonu için 4 farklı nöron sayısı (9, 12, 23, 35) ile 7 tekrarlı deneme yapılmıştır. Eğitim fonksiyonlarından *trainlm* hariç tümünde en yüksek ağ performansı, en yüksek nöron sayısı olan 35'te elde edilmiştir. *Trainlm* eğitim fonksiyonu, 12 girdi için eşik değer olan 24 nöron sayısının aşıldığı denemelerde R=0,91 değerini yakalayamamıştır. En iyi performansın *trainlm* eğitim fonksiyonunda elde edildiği, ardından sırasıyla *trainbfg*, *trainscg*, *trainrp* ve *traincgf* eğitim fonksiyonlarının geldiği görülmektedir. Elde edilen sonuçlar Çizelge 6.2'de verilmiştir. Çizelge 5.4'te *trainlm* eğitim fonksiyonu için elde edilen sonuçlar verilmiştir. Tekrarlı denemelerle performansın arttığı tek eğitim fonksiyonu olan *trainlm* ile yapılan denemelerde, çalışmanın en iyi performansı olan R=0,91 değerine, 23 nöron sayısında ve 7. tekrarda ulaşılmıştır.

Çizelge 5.5. Trainlm eğitim fonksiyonu ile elde edilen sonuçlar

Gizli Katman Nöron Sayısı	Eğitim R degeri	Validasyon R degeri	Gizli Katman Nöron Sayısı	Eğitim R degeri	Validasyon R degeri
	0,86	0,84		0,91	0,80
	0,82	0,82		0,87	0,87
	0,81	0,80		0,89	0,87
9	0,89	0,83	23	0,91	0,82
	0,82	0,55		0,89	0,87
	0,85	0,89		0,90	0,89
	0,71	0,62		0,92	0,91
	0,87	0,88		0,88	0,77
	0,90	0,89		0,88	0,88
	0,78	0,71		0,90	0,87
12	0,88	0,84	35	0,84	0,82
	0,43	0,38		0,86	0,81
	0,79	0,85		0,89	0,72
	0,87	0,78		0,91	0,87

Mevcut çalışmanın ilk alt problemi olan operasyon bazlı tezgâh atama probleminin çözümü için kurulan nihai ağın parametreleri Çizelge 5.5'te verilmiştir. Ağın öğrenme katsayısı, katman sayısı, nöron sayısı gibi parametre değerlerinin kullanıcı inisiyatifinde, deneme yanılma yolu ile belirlenebilmesi YSA'nın dezavantajları arasında yer almaktadır. Bu durum göz önüne alınarak denemeler sonucunda elde edilen en iyi değerde ağın eğitimi tamamlanmış ve eğitilen ağ sonraki aşamada kullanılmak üzere *.mat* formatında kaydedilmiştir.

Çizelge 5.6. Ağ parametreleri

MLF Yapay Sinir Ağı Parametreleri	
Eğitim algoritması	Levenberg-Marquardt backpropagation (trainlm)
LR (Lerarning rate)	0,3
Eğitim veri seti oranı	70%
Validasyon veri seti oranı	15%
Test veri seti oranı	15%
Hata fonksiyou (Error function)	Ortalama Kare Hata (MSE)
Maksimum epoch sayısı	1000
Katmanlar	12 girdi, 1 gizli, 1 çıktı
Gizli katman nöron sayısı	23

5.4. Aşama 2- Veri Setinin Kümeleme Ağlarına Göre Revizyonu

Kohonen MATLAB SOM sinir ağı uygulamaları üzerine yaptığı çalışmasında kümeleme çalışmalarının çoğu zaman nesnelerin bir grup özelliğinin belirlenmesi şeklinde yapıldığını ifade etmiştir. Bu özellikler istatistiksel ya da bilimsel çalışmalarda sayısal değerler olarak karşımıza çıkmaktadır. Ancak sosyal bilimlerde bu özellikler sözel olarak ifade edilmektedir ve sinir ağlarıyla kümele yapabilmek için bu ifade sayısallaştırılmalıdır. Örneğin "X kişisi kısa saçlıdır" ifadesinin doğru olduğunu bildirmek için 1, yanlış olduğunu bildirmek için 0 değeri kullanılır. Kohonen mevcut çalışmaya benzer nitelikte bir örnekte, 13 hayvan için 11 özelliği değişken olarak ele almıştır. SOM ağına uygun şekilde 0-1 ikili değerleriyle oluşturduğu matriste 13 satır hayvanları (güvercin, tavuk, ördek, şahin, kartal, kurt, köpek, tilki, aslan, kedi, kaplan, inek), 11 sütun ise satırlarla eşleşen ve doğru olması durumunda 1, yanlış olması durumunda 0 değerini alan değişkenleri (küçük, orta, büyük, 2 bacaklı, 4 bacaklı, toynaklı, yeleli, avcı, koşucu, uçan, yüzen) ifade etmektedir.

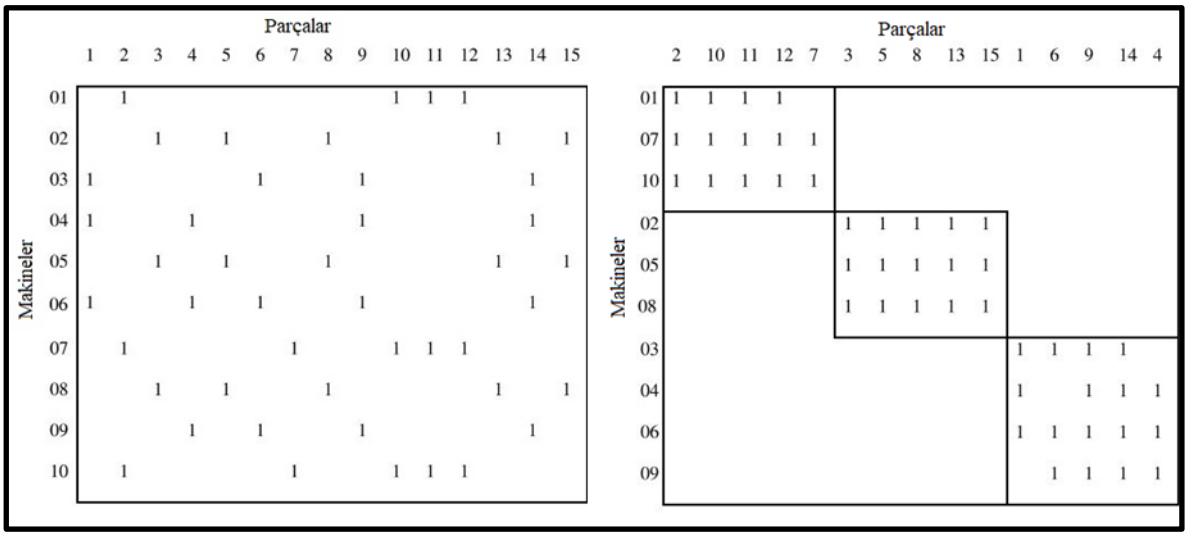
Bu verilere göre oluşturulan X matrisi şu şekildedir [66]:

X = [1 0 0 1 0 0 0 0 0 1 0
1 0 0 1 0 0 0 0 0 0 0
1 0 0 1 0 0 0 0 0 1 1
1 0 0 1 0 0 0 1 0 1 0
0 1 0 1 0 0 0 1 0 1 0
0 1 0 0 1 0 0 1 0 0 0
0 1 0 0 1 0 0 0 1 0 0
0 1 0 0 1 0 1 1 1 0 0
1 0 0 0 1 0 0 1 0 0 0
0 0 1 0 1 0 0 1 1 0 0
0 0 1 0 1 0 1 1 1 0 0
0 0 1 0 1 1 1 0 1 0 0
0 0 1 0 1 1 0 0 0 0 0]

Mevcut çalışmanın 2.bölümünde bahsedilen ve hücre oluşturma yöntemlerinden PFA'nın temelini oluşturan parça-makine matrisi $\{a_{ij}\}$, *m* satırda makineler ve *n* sütunda parçalardan oluşmaktadır. Matriste yer alan bir a_{ij} elemanının değeri için şu koşul geçerlidir:

$$aij = \begin{cases} 1, eğer\ j\ parçası\ için\ i\ nolu\ tezgah\ kullanılacaksa \\ 0, \quad diğer\ durumlar \end{cases} \tag{5.1}$$

Şekil 5.3'de 10 makine ve 15 parça için oluşturulmuş örnek bir parça-makine matrisi ve hücre oluşturma problemi çözümü görülmektedir [67].



Şekil 5.3. Örnek parça-makine matrisi ve problem çözümü

Kohonen'in SOM ağı uygulamasından yola çıkarak mevcut çalışmanın 1.aşamasında tezgâh atama probleminin çözümü için hazırlanmış Çizelge 4.3'te yer alan veri seti, Çizelge 5.6'da görüldüğü şekilde 0-1 ikili matrise çevrilmiştir.

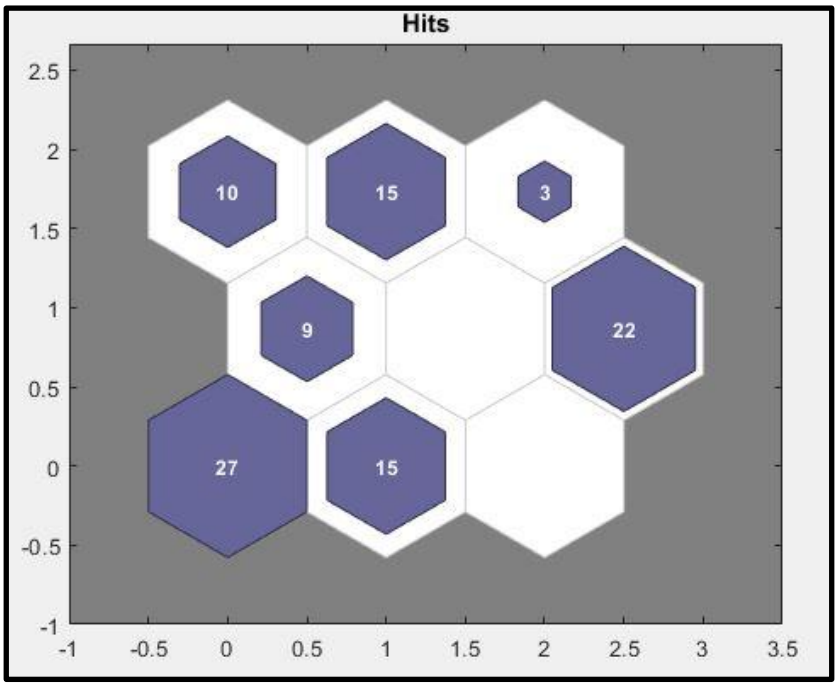
Çizelge 5.7. SOM matris yapısına uygun veri seti

		Tezgahlar										
Parça	Operasyon	501	502	503	504	505	506	507	508	509	510	511
4	10003	0	0	0	0	1	0	0	0	0	0	1
4	10009	0	0	0	0	1	0	0	0	0	0	1
4	10008	0	0	0	0	1	0	0	0	0	0	1
4	10007	0	0	0	0	1	0	0	0	0	0	1
4	20009	0	0	0	0	1	0	0	0	0	0	1
4	10006	1	1	1	0	0	0	0	0	0	0	1
5	10003	0	0	0	0	1	0	0	0	0	0	1
5	10009	0	0	0	0	1	0	0	0	0	0	1
5	10008	0	0	0	0	1	0	0	0	0	0	1
5	10007	0	0	0	0	1	0	0	0	0	0	1
5	20009	0	0	0	0	1	0	0	0	0	0	1
5	10006	1	1	1	0	0	0	0	0	0	0	1
6	20006	1	1	1	0	0	0	0	0	0	0	0
6	20012	1	1	1	0	0	0	0	0	0	0	0
8	20003	0	1	1	0	0	0	0	1	0	0	0
8	20009	0	1	1	0	0	0	0	1	0	0	0
9	20003	0	1	1	0	0	0	0	1	0	0	0
9	20009	0	1	1	0	0	0	0	1	0	0	0

Problemin 2.aşamasında kullanılmak üzere hazırlanan bu matriste satırlar parçaları ve operasyonlarını, sütunlar tezgâhları ifade etmektedir. 1.aşamada ağın çıktısı olarak bulunan 13. satır değerleri yeni matriste 0-1 değerlerini alarak operasyonun ilgili tezgâhta yapılıp yapılamayacağını göstermektedir. Örnek olarak 1.aşamadaki ağın sonucuna göre 4. parçanın 20009 kodlu operasyonu 5 ve 11 numaralı tezgahlarda işlenebilmektedir. Bu bilgiye dayanarak yeni matriste yalnızca 5 ve 11 numaralı tezgahlar 1 değerini alırken diğer tezgahlar 0 değerini almıştır.

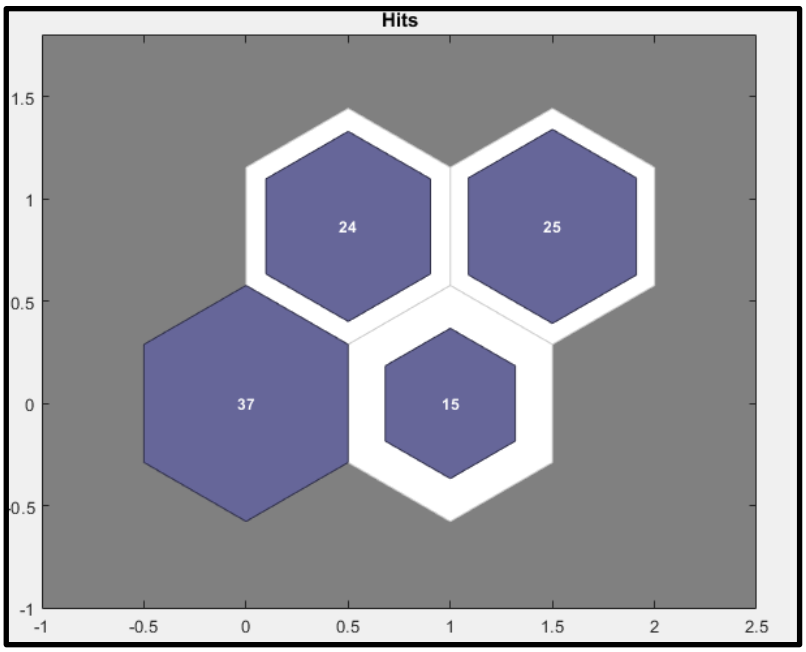
5.5. Aşama 3- İmalat Hücrelerinin Oluşturulması

SOM ağ yapısının kurulması deneme yanılma yöntemine dayanmaktadır. Başlangıçta kaç küme seçileceğini belirlemek mümkün olmadığından rastgele bir sayı ile başlanır ve kümelerin kalitesine göre sayı artırılıp azaltılmaktadır [66]. Boş hücre oluşmaması, her bir tezgâhın mutlaka bir hücreye atanmasa dahi hiçbir tezgâhın bulunmadığı bir hücrenin var olmaması hücre oluşturma probleminin temel kurallarından birisi olarak vurgulanmaktadır. SOM ağı için M nöron sayısı ve N gözlem sayısı değeri olmak üzere, başlangıç nöron sayısını bulmak için literatürde yer alan $M \approx 5\sqrt{N}$ denkleminden yola çıkılmıştır [68].



Şekil 5.4. SOM örnek veri dağılımı

Yapılan denemelerde nöron sayısının 2'den fazla olması durumunda giriş vektörlerine karşılık gereğinden fazla çıkış vektörü olduğu, yani bazı hücrelerin boş kaldığı görülmüştür. Bu şartlar altında mevcut çalışmada yapılan deneme yanılmalar sonucunda en fazla hücre sayısının 4 olması gerektiği belirlenmiştir. Şekil 5.4'te yer alan SOM haritasında 34 parça için 101 giriş verisi için 2'den fazla nöron seçilmesi durumunda bazı hücrelerin boş kalacağı görülmektedir.



Şekil 5.5. SOM ağı giriş verisi dağılımı

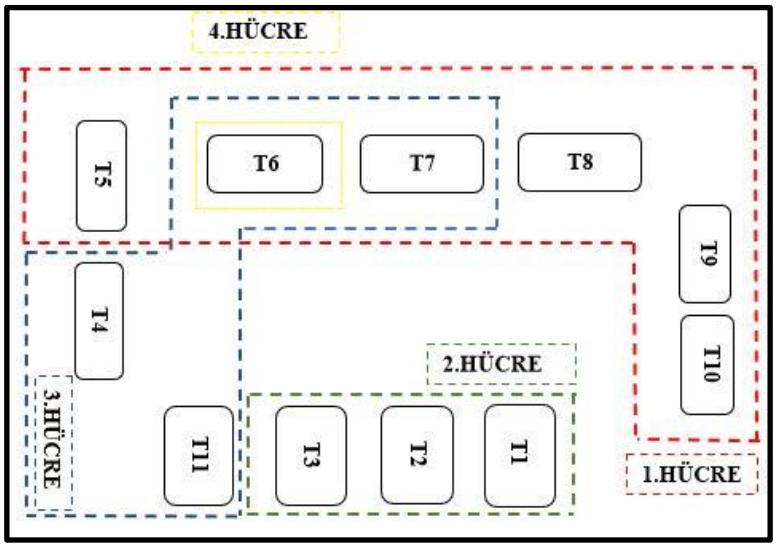
Şekil 5.5'te bu çalışmada yapılan MATLAB SOM ağı kümelemesi sonucunda oluşan haritada, her bir nörona karşılık gelen 101 adet giriş verisinin dağılımı görülmektedir. SOM sinir ağları görüntü tanıma, ses tanıma, hastalık teşhisi, müşteri analizi, metin ve sembol tanıma gibi büyük veri setlerinin belirli özellikler doğrultusunda sınıflandırılması ve ayrıştırılmasına ihtiyaç duyulan pek çok farklı alanda yapay zekâ uygulamalarıyla karşımıza çıkmaktadır.

Çizelge 5.6'da verilen matris yapısındaki veri setini girdi olarak kullanan, en fazla hücre sayısını 4 olarak ele alan ve parça operasyonları bazında tezgâhları hücrelere atayan iki farklı kümeleme ağı çalıştırılmıştır. SOM ve CNN ağlarının çıktı değerleri ham veri olarak Çizelge 5.7'de görülmektedir. Çizelge üzerinden örnekle SOM ağının 5.parçanın 10007 operasyonuna dayanarak 5 ve 11 numaralı tezgâhları 3. hücreye atadığı görülürken, CNN ağı aynı tezgâhları 2. hücreye atamıştır.

Çizelge 5.8. SOM ve CNN ağlarının çıktı değerleri

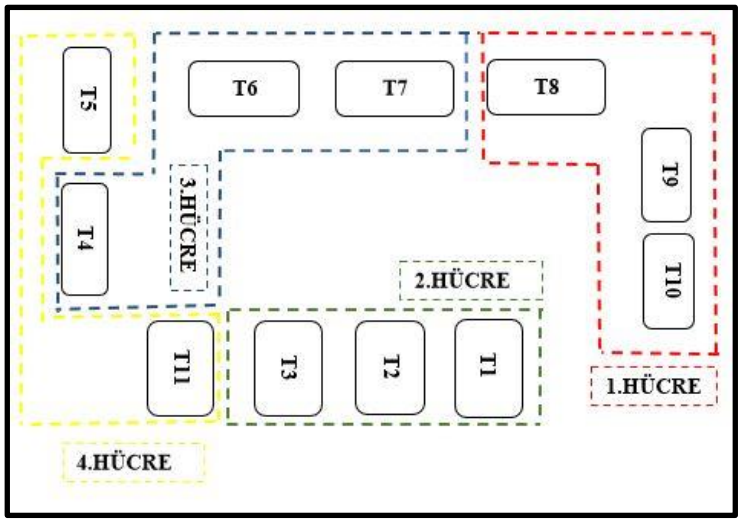
		Tezgahlar											YSA hücreleri	
Parça	Operasyon	1	2	3	4	5	6	7	8	9	10	11	SOM	CNN
4	10003	0	0	0	0	1	0	0	0	0	0	1	2	2
4	10009	0	0	0	0	1	0	0	0	0	0	1	2	2
4	10008	0	0	0	0	1	0	0	0	0	0	1	2	2
4	10007	0	0	0	0	1	0	0	0	0	0	1	2	2
4	20009	0	0	0	0	1	0	0	0	0	0	1	2	2
4	10006	1	1	1	0	0	0	0	0	0	0	1	4	1
5	10003	0	0	0	0	1	0	0	0	0	0	1	2	2
5	10009	0	0	0	0	1	0	0	0	0	0	1	2	2
5	10008	0	0	0	0	1	0	0	0	0	0	1	2	2
5	10007	0	0	0	0	1	0	0	0	0	0	1	2	2
5	20009	0	0	0	0	1	0	0	0	0	0	1	2	2
5	10006	1	1	1	0	0	0	0	0	0	0	1	4	1
6	20006	1	1	1	0	0	0	0	0	0	0	0	4	1
6	20012	1	1	1	0	0	0	0	0	0	0	0	4	1
8	20003	0	1	1	0	0	0	0	1	0	0	0	4	1
8	20009	0	1	1	0	0	0	0	1	0	0	0	4	1
9	20003	0	1	1	0	0	0	0	1	0	0	0	4	1
9	20009	0	1	1	0	0	0	0	1	0	0	0	4	1
10	10003	0	0	0	1	0	1	1	0	0	0	0	1	4
10	10007	0	0	0	0	1	1	1	0	0	0	0	3	2
12	10004	0	0	0	1	1	0	0	0	0	0	1	2	2
12	10006	0	0	0	1	1	0	0	0	0	0	0	2	2
13	10003	0	0	0	1	0	1	1	0	0	0	0	1	4
13	20006	1	1	1	0	0	0	0	0	0	0	0	4	1
15	10003	0	0	0	0	1	0	0	0	0	0	1	2	2
15	10006	0	0	0	0	1	0	0	0	0	0	1	2	2
17	10003	0	0	0	1	0	1	1	0	0	0	1	1	4

Şekil 5.6'da işletmenin mevcut sabit tezgâh yerleşimi üzerinde SOM sinir ağı ile oluşturulan sanal dinamik imalat hücreleri görülmektedir. Nöron sayısı 2 olarak ele alınan çalışmada ağın en fazla 4 hücre oluşturması istenmiştir ve bu kriterler altında SOM ağının 4.hücre olarak 6 numaralı tezgâhı seçtiği ve bu tezgâhın halihazırda 3.hücreye dahil olduğu görülmektedir.



Şekil 5.6. SOM ağı D-VCM hücreleri

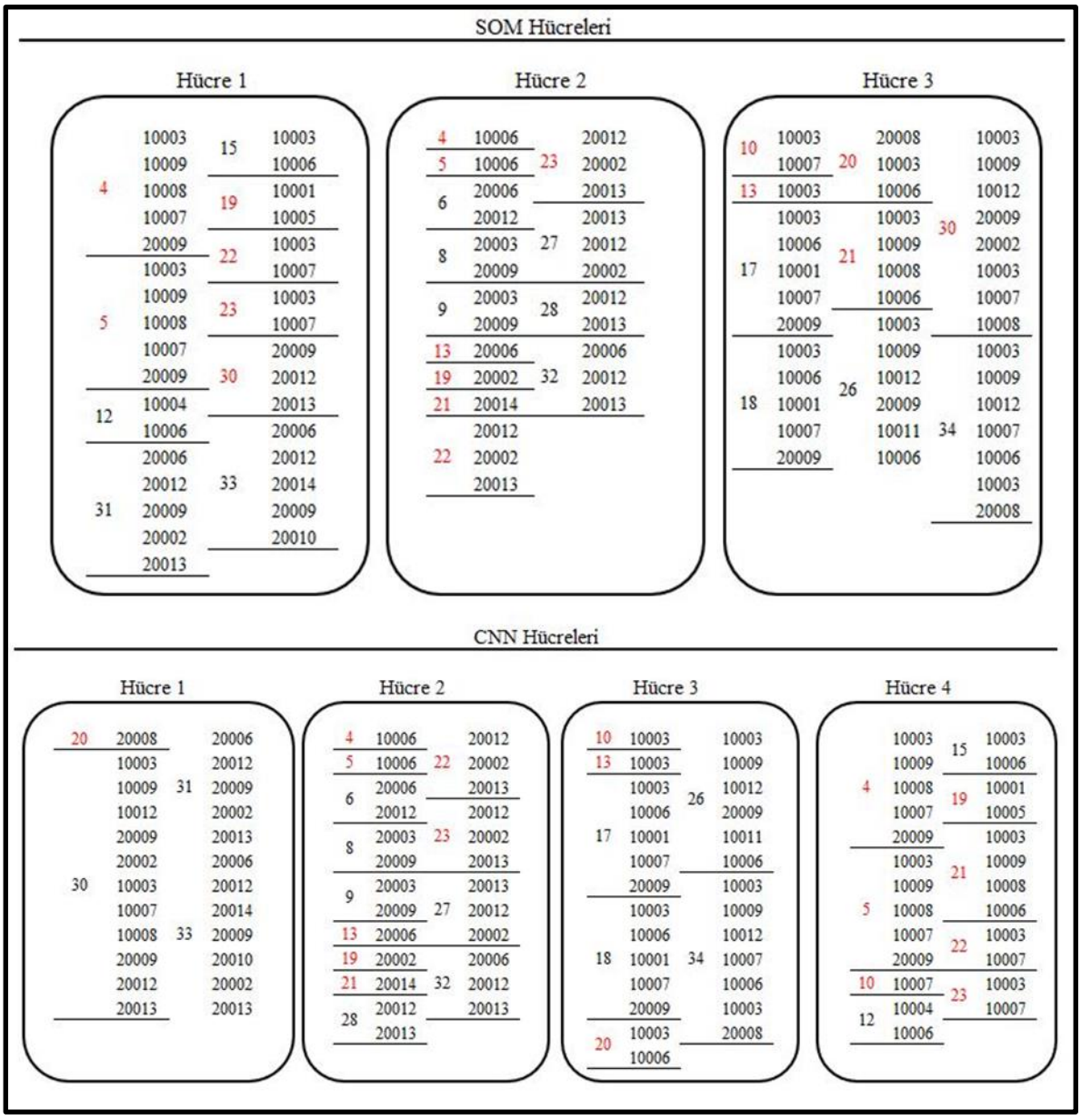
Önceki bölümlerde bahsedildiği gibi sanal imalat hücreleri arasında makine paylaşımı mümkün olabilmektedir. Ancak paylaşılan tezgâh aynı hücreye dahil olduğundan 6 numaralı tezgâhı bağımsız değerlendirmek mümkün olmamıştır. Bu sebeple CNN ağı sonuçları ile yapılan karşılaştırmada SOM ağı için 3 imalat hücresi esas alınmış, 6 numaralı tezgâhın yükü ve parça gezinmesi verileri 3.hücre dahilinde değerlendirilmiştir.



Şekil 5.7. CNN ağı D-VCM hücreleri

Şekil 5.7'de CNN ağlarının oluşturduğu sanal dinamik imalat hücreleri yine işletmenin mevcut sabit tezgâh yerleşimi üzerinde verilmiştir. SOM hücre yapılanmasına karşılık, tezgâh dağılımı açısından daha düzenli bir kümelenme oluştuğu görülmekle birlikte iki ağın belirlediği dinamik sanal imalat hücresi yapılarının karşılaştırması, hücrelere atanan

parça/operasyon sayısına bağlı olarak tezgâh yükleri ve hücreler arası parça gezinmesi kriterleri esas alınarak yapılmıştır.



Şekil 5.8. SOM ve CNN hücreleri parça/operasyon dağılımı

Şekil 5.8'de SOM ve CNN ağlarının oluşturduğu imalat hücrelerinde yer alan parçalar ve operasyonları hücre bazında görülmektedir. Kırmızı işaretli parçalar tüm operasyonlarını aynı hücre içerisinde tamamlayamayan yani hücreler arası gezinen parçalardır ve CNN ağı ile oluşan hücrelerde gezinen parça sayısı 9 iken, SOM ağı ile 3 imalat hücresi belirlenmiş olmasına rağmen gezinen parça sayısı 8'dir. CNN hücrelerinde gezinen 9 parçanın 7'si tek,

2’si 2 operasyon için hücre dışına çıkmaktadır ve uygulayıcı inisiyatifinde yapılacak müdahale ile gezinmenin en aza indirilmesi mümkün olacaktır.

Oluşan hücreler tezgahların operasyon bazında dolulukları açısından incelendiğinde CNN hücrelerinin tezgâh yükü açısından SOM hücrelerine göre daha orantılı bir dağılıma sahip olduğu görülmektedir. Çizelge 5.8’de her bir hücre için tezgahlar ve ilgili operasyon sayıları verilmiştir. Kümeleme ağlarının sonuçları değerlendirildiğinde SOM ağının daha az hücre belirlemesine rağmen oluşturduğu hücreler arasında parça gezinmesinin CNN hücrelerindekine yakın olduğu; CNN ağıyla elde edilen hücrelerde tezgâh dağılımının daha düzgün olduğu; operasyon bazında hücrelerdeki tezgâh yüklerinin SOM sinir ağıyla elde edilen sonuçlara göre CNN ağında daha orantılı olduğu görülmektedir.

Çizelge 5.9. Operasyon bazında hücrelerdeki tezgâh yükleri

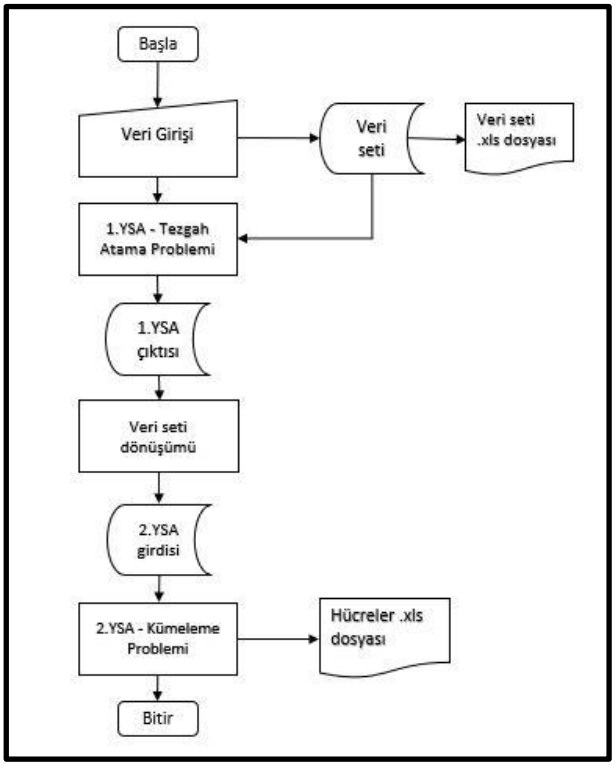
	SOM		CNN	
	Tezgah	Doluluk	Tezgah	Doluluk
HÜCRE 1	5-8-9-10	33 operasyon (10 parça)	8-9-10	24 operasyon (4 parça)
HÜCRE 2	1-2-3	25 operasyon (13 parça)	1-2-3	25 operasyon (13 parça)
HÜCRE 3	4-6-7-11	41 operasyon (9 parça)	4-6-7	27 operasyon (7 parça)
HÜCRE 4	-	-	5-11	27 operasyon (9 parça)

Çalışma kapsamında ele alınan problemin bileşik ağlarla çözümünde, alt problemlerden 2.si olan hücre oluşturma probleminin çözümü için, yukarı bahsedilen değerlendirmelerden yola çıkarak, CNN yapay sinir ağı seçilmiş ve *.mat* formatında kaydedilmiştir.

Kümeleme ağları denetimsiz öğrenmeye dayalı olarak çalışmakla birlikte ağın ağırlıklarını (vektörel uzaklık) ayarlayabilmesi için eğitilmesi gerekmektedir. Mevcut çalışmada bir üretim periyodu boyunca üretilecek 24 parça için hücreler oluşturulmuş ve ağın hafızasında kaydedilmiştir. Bu noktadan sonra ağ yeni gelecek parçaları hafızasında tuttuğu ağırlık değerlerine göre uygun hücrelere atayacak ve/veya yeni hücre yapılanmaları oluşturacaktır. Bu durum dinamik talep karşısında sanal hücrelerin yeniden yapılanabilir olmasıyla tam olarak örtüşmektedir.

5.6. Aşama 4- Yapay Sinir Ağlarının Birleştirilmesi

Ağların birleştirilmesi ile ilgili yapılan literatür araştırmalarına ve yöntemlerin uygulanabilirliğine dayanarak mevcut çalışmada, hangi parçanın hangi tezgâhta işleneceği, bu tezgâhın hangi kümede yer alacağı ve küme sayısı problemin başında belirsiz olduğundan ağların birleştirilmesi modüler yaklaşım ile gerçekleştirilmiştir. Çalışmada ele alınan problemin daha önceki bölümlerde ifade edilen karakteristikleri sebebiyle, yapay sinir ağlarının birleştirilmesi noktasında seçilen yöntem, Bölüm 3.8'de yer verilen Modüler Yaklaşım olmuştur. Hücre oluşturma problemi gerek problemin zorluğu ve karmaşıklığı gerekse kurulacak yapay sinir ağının yönetimi açısından karmaşık bir problem olarak değerlendirilmektedir. Bu durum da Modüler Yaklaşımın bir alt yöntemi olan Sıralı Yaklaşım uygulanmıştır. Problem alt parçalara bölünerek çözülmüş ve bir ağın çıktısı diğer ağın girdisi olarak kullanılmıştır. Ağların kurulumu ve birleştirilmesi Şekil 5.9'da yer alan akış diyagramı ile gösterilmiştir. Veri girişi ile başlayan programda, veri seti sonraki aşamada kullanılmak üzere .xls formatında kaydedilmektedir. 1.yapay sinir ağının çalışması sonrasında elde edilen sonuç, 2. yapay sinir ağının girdisi olacak şekilde dönüştürülmektedir. 2.yapay sinir ağı, 1.ağın çıktısını kullanmakta ve sonuca ulaşmaktadır.



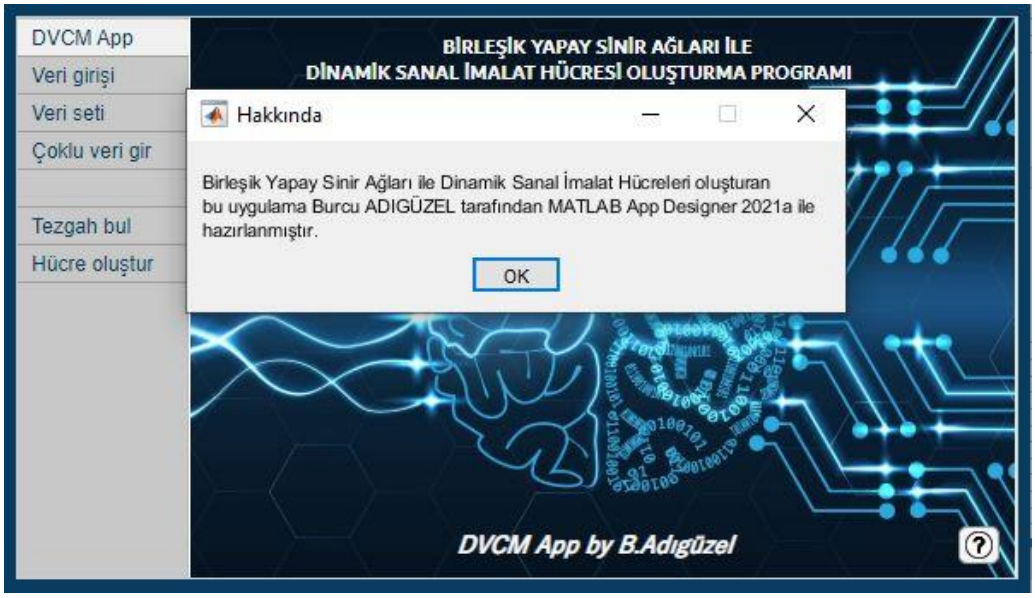
Şekil 5.9. Ağların birleştirilmesi akış diyagramı

6. GELİŞTİRİLEN PROGRAM- DVCM APP VE ÖRNEK UYGULAMA

Birleşik yapay sinir ağları kullanılarak dinamik sanal imalat hücrelerinin oluşturulduğu DVCM App programı, MATLAB 2021a programı App Designer uygulaması ile geliştirilmiştir. *Veri girişi*, *Tezgâh bul* (tezgâh atama alt problem çözümü) ve *Hücre oluştur* (hücre oluşturma alt problem çözümü) sekmelerinde mevcut çalışma dahilinde yapılan çalışmalarla oluşturulmuş veri setleri ve .mat formatında kayıtlı eğitilmiş ağlar kullanılmıştır.

Programın ara yüz tasarımı tamamen özgün olup App Designer kütüphanesi bileşenleri ve web kaynaklı görsellerle hazırlanmıştır. Hakkında, Veri Girişi, Veri seti, Çoklu veri gir, Tezgâh bul ve Hücre oluştur olmak üzere 6 sayfanın bulunduğu program kullanıcıya kolaylık sağlamak amacıyla her sayfada yönlendirmeler içermektedir. Resim 6.1'de programın ana sayfası görülmektedir.

Resim 6.1. DVCM App ana sayfa görünümü

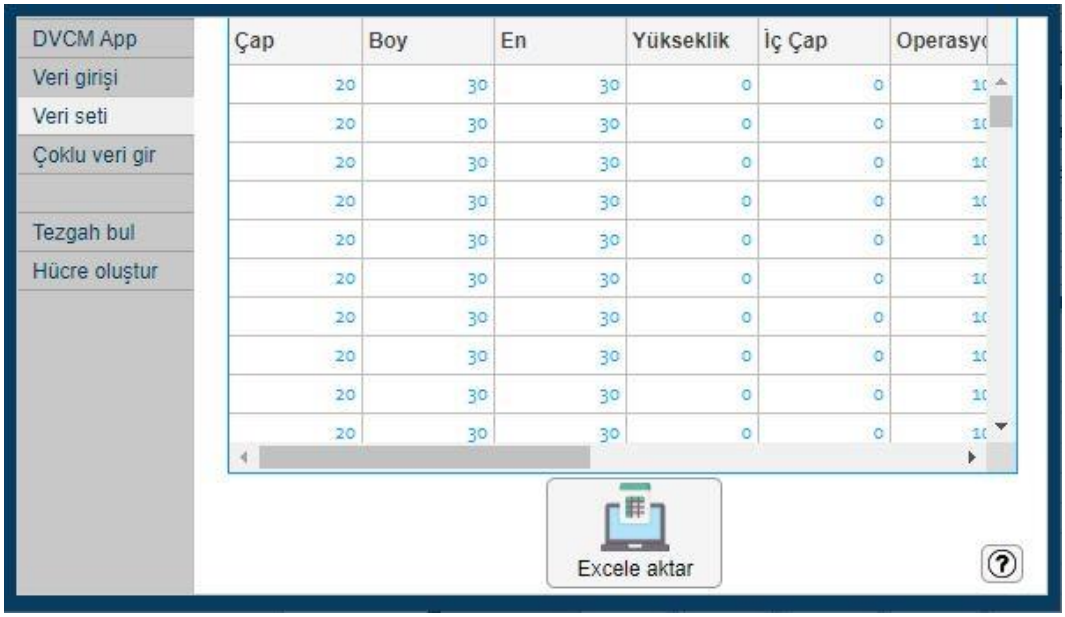


Resim 6.2. DVCM App bilgilendirme sayfası

Resim 6.3. DVCM App veri girişi sayfası görünümü

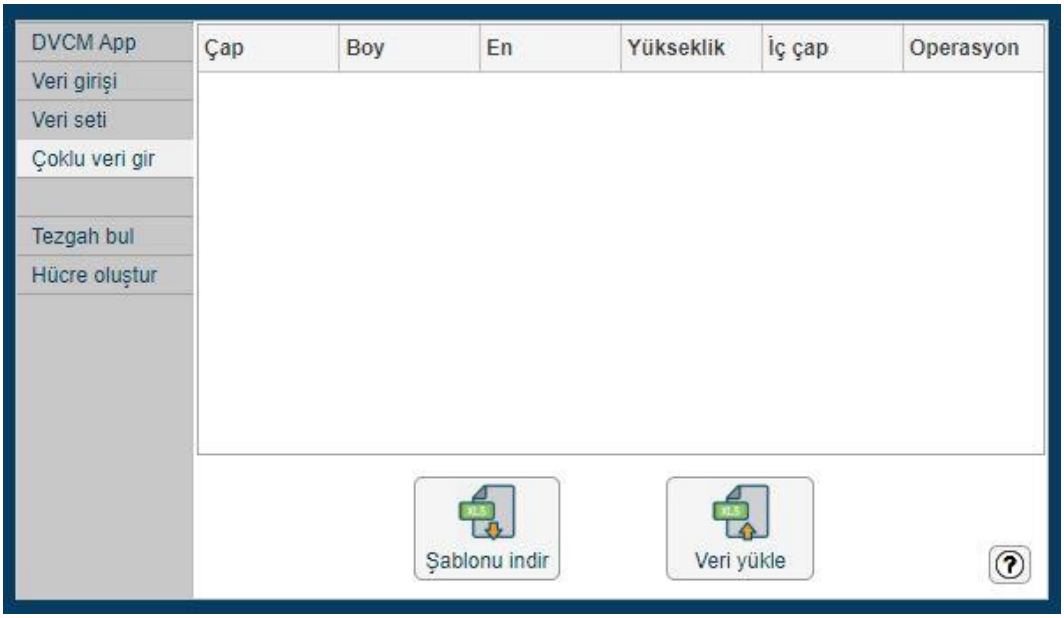
Resim 6.3'te görülen veri girişi sayfasında öncelikle kullanıcının üretime dahil etmek istediği parça ölçülerini girmesi ve parça için gerekli olan torna/freze operasyonlarını seçmesi beklenmektedir. Parça ölçüleri değerleri mm cinsinden kabul edilmektedir ve mümkün olabilecek tüm operasyonlar checkbox ile seçilebilmektedir. Bu bilgilerin girilmesi ve *veri setini oluştur* butonuna basılmasıyla, mevcut çalışmanın 1. aşamasında tezgâh atama problemi için kurulan ve eğitilen, ileri beslemeli geri yayılımlı yapay sinir ağının çalışması için gerekli olan veri seti otomatik olarak oluşmaktadır. Girilen bilgilerin

tamamı program dahilinde yapay sinir ağının yapısına uygun şekilde kodlanıp çoğaltılmakta ve Resim 6.4'te yer alan Veri seti sayfasında uygulayıcıya gösterilmektedir. Standart bir şablon halinde program tarafından hazırlanan bu veri seti dosyası sadece kullanıcının görüntülemesi amacıyla *Excele aktar* butonuna basılmasıyla .xls formatında indirilebilmektedir.



Resim 6.4. DVCM App veri seti sayfası görünümü

Halihazırda oluşturulmuş imalat hücreleri ve bu hücrelere dağıtılmış parçaları temel alarak, dinamik bir talep karşısında üretime dahil edilmesi gereken ilave parçalar olması durumu için tasarlanmış olan *Veri girişi* sayfasına ek olarak, Resim 6.5'te görülen *Çoklu veri gir* sayfasında kullanıcının bir üretim periyodu içindeki tüm parçaları ağa tek seferde sunmak istemesi halinde kullanabileceği .xls formatında hazır bir şablon bulunmaktadır. Dosya program üzerinden indirilebilmektedir. Kullanıcı verileri şablona kaydedildikten sonra, *Şablonu yükle* butonuna basılıp Windows dosya seçim ekranından ilgili dosyanın seçmesiyle, dosyanın içerdiği veriler (parçanın çap, boy, en, yükseklik, iç çap ölçüleri mm cinsinden ve gerekli operasyonlar YSA kodları) Şekil 'deki haliyle ekranda görünecektir.



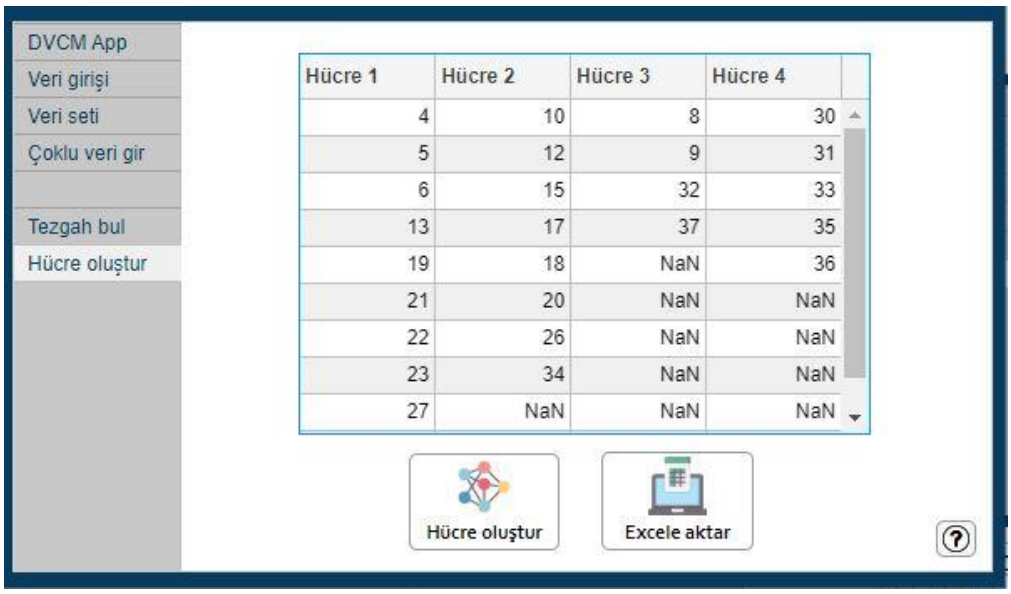
Resim 6.5. DVCM App çoklu veri giriş sayfası görünümü

Resim 6.6'da görülen *Tezgâh bul* sayfası uygulamanın yapıldığı işletmeye uygun olarak tasarlanmıştır. İşletmede çalışma kapsamında ele alınan 11 tezgâhın görsellerinin yer aldığı bu sayfa, mevcut çalışmanın operasyon bazlı tezgâh atama alt problemi için kurulan yapay sinir ağının çalıştırıldığı kısımdır. Veri girişin ardından *Ağı çalıştır* butonuna basılmasıyla program veri setini ağa göndermekte ve Bölüm 5.3 Aşama 1'in sonunda *.mat* formatında kaydedilmiş sinir ağı çalışmaktadır. Yapay sinir ağının tahmin etmesi beklenen çıktı değeri yani operasyon bazında parçanın işlenebileceği tezgâhlar Resim 6.6'daki gibi kullanıcıya sunulmaktadır.

Programın bu aşamasında kullanıcı ilerlemek yani imalat hücreleri oluşturmak zorunda değildir. *Tezgâh bul* sayfası talaşlı imalat atölyesinde üretime dahil edilecek tüm parçaların mevcut operasyonlarına bağlı olarak uygun tezgâhları bulmak için kullanılabilmektedir. Resim 6.6'da 1'den 11'e kadar tüm tezgahlar içinde parçanın işlenebilmesi için en uygun olan 11 numaralı tezgâh 1.sırada olmak üzere sırasıyla 6., 7. ve 5. tezgahlar alternatif olarak sunulmaktadır.

Resim 6.6. DVCM App tezgâh bul sayfası görünümü

Parçalara uygun tezgahların bulunmasıyla eşzamanlı olarak tahmin veri seti, program dahilinde arka planda, kümelemeye uygun olacak şekilde 0-1 matrise dönüştürülmekte ve yeni veri seti dosyası 2.alt problem olan hücre oluşturma probleminin çözümü için kurulan CNN ağının girdi seti olarak kaydedilmektedir. CNN ağı belirli bir periyod için geçerli üretim planı dahilinde oluşturduğu kümelenme yapısına yeni eklenen parçaları dahil ederek, imalat hücrelerine atanmış parçaları Resim 6.7'deki gibi göstermektedir. *Excele aktar* butonuna basılmasıyla oluşan hücrelerde hangi tezgahların yer aldığı ve parçaların hangi operasyonlarının bu tezgahlarda yapılacağı da görülebilmektedir.



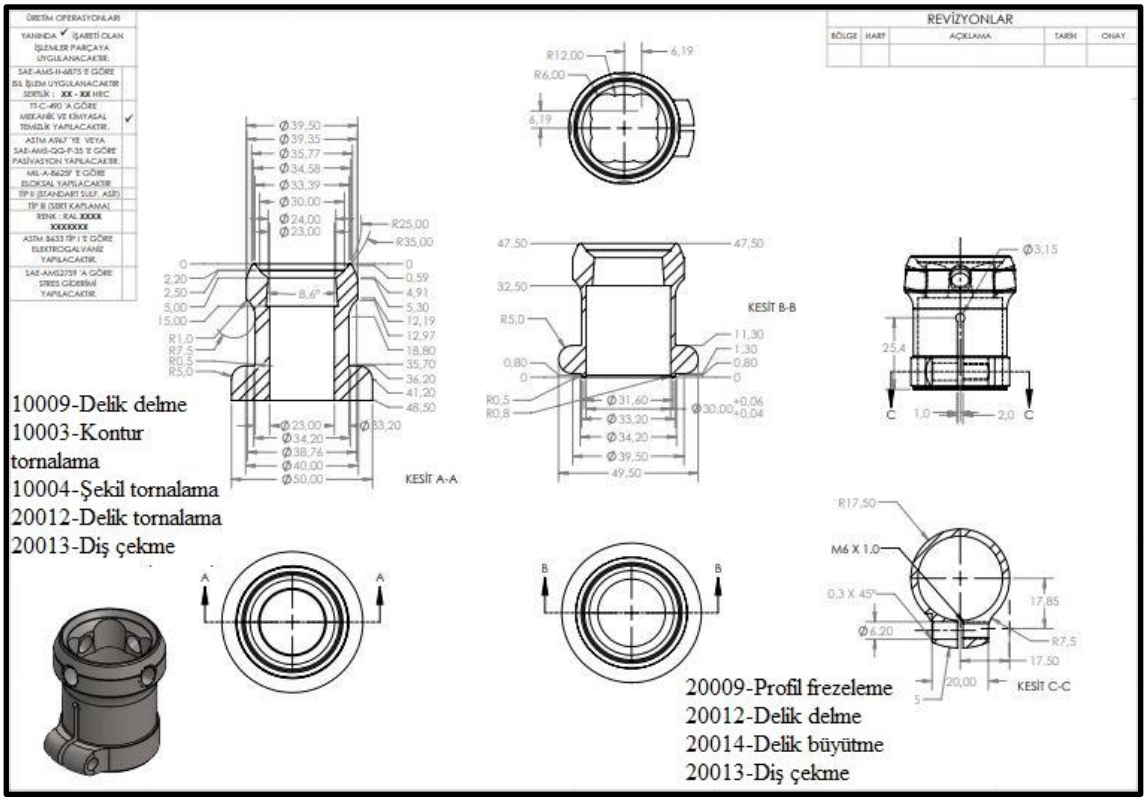
Resim 6.7. DVCM App hücre oluştur sayfası görünümü

MATLAB, gerçek veya karmaşık sayı olmayan değerleri, NaN adı verilen ve “Not a Number” yani “Sayı değil” anlamına gelen özel bir değerle temsil eder. Çalışmada programın sonuçları yazdırdığı .xls formatındaki dosyadaki boş hücreler Resim 6.7’de NaN değeri ile gösterilmektedir. Örnek olarak 3. imalat hücresine 8, 9, 32 ve 37 numaralı parçaların atandığı görülmektedir.

Örnek uygulama

Program uygulama işletmesinin mevcut dönemdeki 2 aylık üretim planı dahilindeki 26 parça için çalıştırılmış ve Resim 6.7’deki hücreler oluşturulmuştur. Her bir parça ilgili hücrede kendi numarasıyla görülmektedir. Örnek olarak 1.hücrede yer alan parçalar 4, 5, 6, 13, 19, 21, 22, 23 ve 27 numaralı parçalardır.

Mevcut üretim planına 40 numaralı yeni bir parça dahil edilmek istenmektedir. Resim 6.8’de 40 numaralı yeni parça ve operasyonlarının YSA kodları görülmektedir. Yapılan çalışmanın aşamaları ve programın çalışma mantığı esas alınarak uygulanan adımlar aşağıda sıralanmıştır:

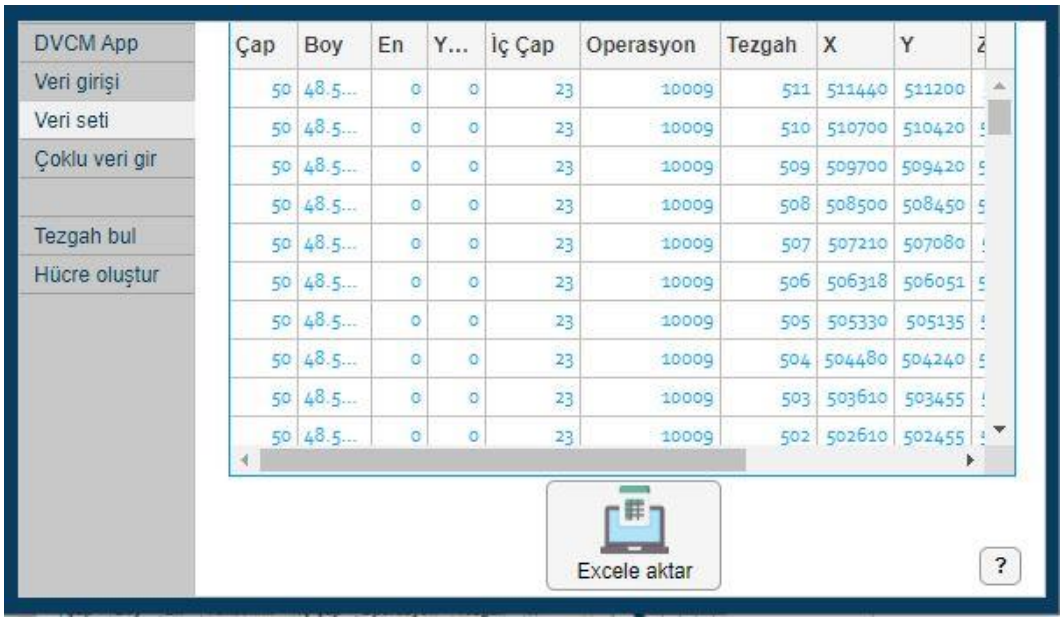


Resim 6.8. 40 numaralı parça ve operasyon YSA kodları

1.adım: Sisteme dahil edilmek istenen parçanın teknik resmi üzerinde talaşlı imalat operasyonları belirlenir. Resim 6.9'da görüldüğü gibi programın veri girişi ekranında parça ölçüleri girilir ve operasyonlar seçilir.

Resim 6.9. 40 numaralı parça için veri girişi

2.adım: Programın veri giriş ekranından parça ölçüleri ve belirlenen operasyonlar girilerek veri seti oluşturulur. Oluşturulan veri seti Resim 6.9'da görülmektedir.

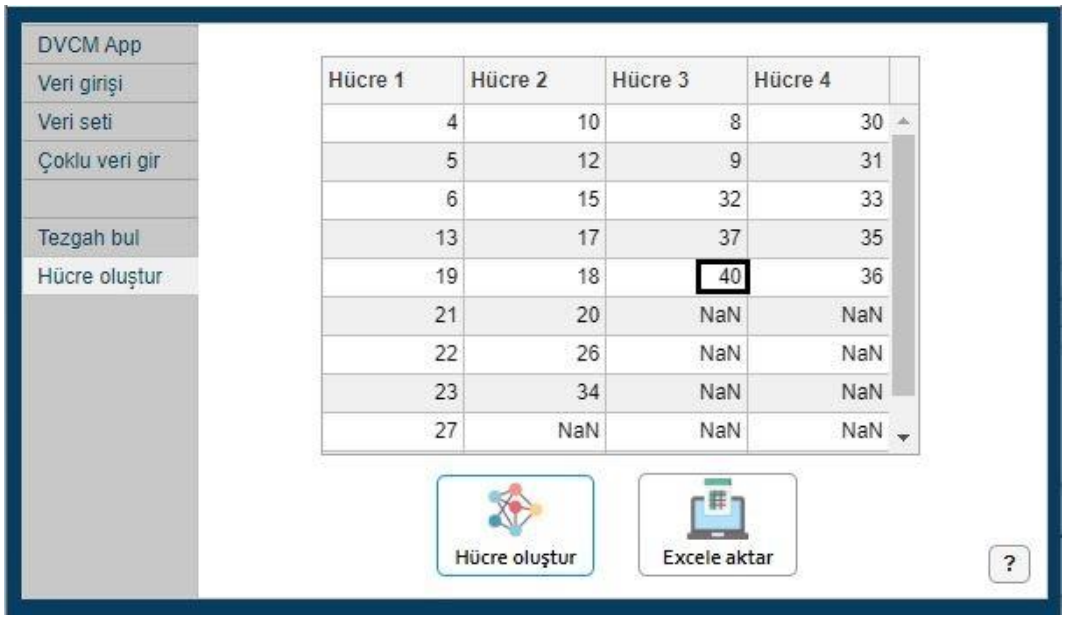


Resim 6.10. 40 numaralı parça için oluşturulan veri seti

3.adım: Programın *Tezgâh Bul* sekmesinde *Ağı Çalıştır* butonuna basılmasıyla 1.yapay sinir ağı çalıştırılır ve 40 numaralı yeni parçanın işlenebileceği tezgahlar bulunur. Resim 6.10'da görüldüğü gibi yeni parçamız 11,6 ve 7 numaralı tezgahlarda işlenebilmektedir.

Resim 6.11. 40 numaralı parçanın işlenebileceği tezgahlar

4.adım: Programın Hücre oluştur sayfasına geçilir ve 2.yapay sinir ağı çalıştırılır. Resim 6.11'de görüldüğü gibi program yeni parçayı mevcut 26 parça için oluşturduğu hücre yapılanmasını bozmadan, 3.imalat hücresine dahil etmiştir.



Resim 6.12. 40 numaralı parçanın atandığı imalat hücresi

7. SONUÇ VE ÖNERİLER

Dinamik Sanal Hücresel İmalat Sistemi kurmak amacıyla imalat hücrelerinin oluşturulduğu bu çalışmada, literatürde yapılan çalışmalardan farklı olarak çözüm yönteminde sadece yapay sinir ağları kullanılmıştır. Yapay sinir ağlarının kullanılması noktasında ise modüler yaklaşım uygulanmış ve birleşik yapay sinir ağları ile çözüme ulaşılmıştır.

Çalışma yapay sinir ağı algoritmalarının performanslarını aynı örnekler üzerinde deneme ve en uygun ağ performansını bulma fırsatı sunmuştur. Yapay sinir ağları ile yapılan çalışmalarda büyük veri setleriyle farklı eğitim algoritmalarının denenmesi sonucunda, eğitim algoritmasının karakteristiğine göre farklı sonuçlar elde edilebilmektedir. Genel olarak sinir ağının çok fazla sayıda ağırlık değeri içerdiği tahmin temelli çalışmalarda denetimli öğrenmeye dayalı ileri beslemeli sinir ağlarında Levenberg-Marquardt (*trainlm*) algoritmasının en iyi sonuçları verdiği görülmüştür. Çalışmanın ilk aşamasında yapılan denemeler de bu durumu destekler bir sonuç ortaya koymuştur ve geliştirilen kullanıcı ara yüzünde bu algoritma ile eğitilmiş olan sinir ağı kullanılmıştır.

Denetimsiz ağlarla yapılan çalışmalarda SOM sinir ağının kullanımı temel yaklaşım olarak ele alınmaktadır ve bu ağdan ve/veya bu ağın geliştirilmesiyle ortaya çıkan algoritmalar mevcuttur. Veri setinin 0-1 değişkenlerden ya da tamamen sayısal değerlerden oluşması durumunda kümeleme ağlarının performansları da değişebilmektedir. Mevcut çalışma kapsamında hücre oluşturma problemi çözümünde kümeleme ağlarıyla yapılan çalışmada rekabetçi algoritmanın Kohonen algoritmasına istinaden daha iyi sonuçlar ürettiği görülmüş ve benzer şekilde geliştirilen uygulamada bu algoritma ile çalışan CNN yapay sinir ağı kullanılmıştır.

Yapay sinir ağlarının dezavantajlarından biri olan yalnızca nümerik verilerle çalışması ve ağa verilerin sunuluş şeklinin ağ performansını doğrudan etkilemesi mevcut çalışmada da karşılaşılan bir dezavantaj olmuştur. Çalışma kapsamında kurulan ağların eğitiminde kullanılabilecek hazır veri setleri mevcut olmadığından, yapay sinir ağlarıyla yapılacak özgül çalışmalarda veri toplama ve derlemenin önemi bu çalışma kapsamında ortaya koyulan bir başka noktadır.

Ağların birleştirilmesi noktasında, karmaşık bir problem olarak ele alınan hücre oluşturma probleminde, parça-operasyon-tezgâh-hücre temel taşlarının problemin yapısına uygun şekilde alt problemlere ayrıştırılması çözüme ulaşmakta kolaylık sağlamıştır.

Çözüm için iki kademeli bir yaklaşım uygulanması ile ağın değiştirilmesi ve geliştirilmesi noktasında avantaj sağlanmış ve veri seti dahil tüm çözüm kurgusunu en baştan tasarlamak yerine, alt ağlara ve veri setlerine müdahale edilerek hatalar azaltılmıştır.

Kullanılan basılı ve dijital kaynakların büyük çoğunluğunun Türkçe dışındaki dillerde olması sebebiyle bu çalışmanın yapay zekâ alanındaki yeni araştırmacılara örnek olması ve bu alanda ülkemizde yapılan çalışmaların artırılması hedeflenmiştir.

Yapılan çalışmanın tekrarlanabilir, uygulanabilir ve geliştirilebilir olması dikkat çekilmek istenen bir başka noktadır. Geliştirilen uygulama işletmenin Üretim Planlama ve Çizelgeleme birimi tarafından mevcut üretim dönemi için kullanılmaya başlamıştır. Talebin dinamik olduğu işletmede program sayesinde tezgâh çizelgelemenin kolaylaştığı ve tezgâh yükleme için riskli görülen durumlarda daha rahat karar verilebildiği şeklinde geri dönüşler alınmıştır. Gelecek üretim periyotlarında bu durumun etkilerinin teslimat sürelerinde kısalma ve müşteri memnuniyetinin artması şeklinde yorumlanması beklenmektedir.

Bu çalışma sonucunda elde edilen bilgi ve tecrübe ile bu alanda çalışma yapacak olan araştırmacılara bazı öneriler sunulmuştur:

- Geliştirilen sistemde operasyon sırası da yapay sinir ağına kısıt olarak verilebilir. Bu sayede bir sonraki adımda operasyon süreleri de ağa öğretilerek planlama periyodunu da ağın tahmin etmesi sağlanabilecektir.
- Yapay sinir ağına veri seti içinde mm cinsinden ölçü şeklinde verilen parça boyutlarının teknik resimlerden doğrudan alınması sağlanabilir. Bu noktada çalışmanın birleşik ağlarla yapılmasının avantajı kullanılarak, teknik resimler için görüntü tanıma (pattern recognition) yapan bir sinir ağı kurulup birleşik ağlara eklenmesi mümkün olacaktır.
- Tezgâh takım tutucu bilgilerinin parça bazında kesici takım ihtiyacı ile eşleştirilmesi ile oluşturulacak yeni bir veri setinin birleşik ağlara dahil edilmesiyle, tezgâh atama probleminin sonuçlarına maliyet tahminleri de eklenebilecektir.

KAYNAKLAR

1. Shanker, R., Vrat, P. (1999). Some design issues in cellular manufacturing using the fuzzy programming approach. *International Journal of Production Research*, 37(11), 2545-2563.

2. Delgoshaei, A., Ali, A. (2019). Evolution of clustering techniques in designing cellular manufacturing systems: A state-of-art review. *International Journal of Industrial Engineering Computations*, 10(2), 177-198.

3. Papaioannou, G., Wilson, J. M. (2010). The evolution of cell formation problem methodologies based on recent studies (1997–2008): Review and directions for future research. *European Journal of Operational Research*, 206(3), 509-521.

4. Venkumar, P., Noorul Haq, A. (2006). Fractional cell formation in group technology using modified ART1 neural networks. *The International Journal of Advanced Manufacturing Technology*, 28(7), 761-765.

5. Mahmoodian, V., Jabbarzadeh, A., Rezazadeh, H., Barzinpour, F. (2019). A Novel İntelligent Particle Swarm Optimization Algorithm for Solving Cell Formation Problem. *Neural Computing and Applications*, 31(2), 801-815.

6. Liao, T. W., Chen, L. (1993). An evaluation of ART1 neural models for GT part family and machine cell forming. *Journal of Manufacturing Systems*, 12(4), 282-290.

7. Onwubolu, G. C. (1999). Design of parts for cellular manufacturing using neural network-based approach. *Journal of Intelligent Manufacturing*, 10(3), 251-265.

8. Chattopadhyay, M., Dan, P. K., Mazumdar, S. (2012). Application of Visual Clustering Properties of Self Organizing Map in Machine–Part Cell Formation. *Applied Soft Computing*, 12(2), 600-610.

9. Ozturk, G., Ozturk, Z. K., Islier, A. A. (2006). A Comparison of Competitive Neural Network with Other AI Techniques in Manufacturing Cell Formation. In *International Conference on Natural Computation*, Springer, Berlin, Heidelberg, 575-583.

10. Venugopal, V., Narendran, T. T. (1994). Machine-cell formation through neural network models. *The International Journal of Production Research*, 32(9), 2105-2116.

11. Pandian, R. S., Mahapatra, S. S. (2009). Manufacturing Cell Formation with Production Data Using Neural Networks. *Computers & Industrial Engineering*, 56(4), 1340-1347.

12. Chen, D. S., Chen, H. C., Park, J. M. (1996). An improved art neural net for machine cell formation. *Journal of Materials Processing Technology*, 61(1-2), 1-6.

13. Guerrero, F., Lozano, S., Smith, K. A., Canca, D., Kwok, T. (2002). Manufacturing Cell Formation Using a New Self-Organizing Neural Network. *Computers and Industrial Engineering*, 42(2-4), 377-382.

14. Kaparthi, S., Suresh, N. C. (1992). Machine-component cell formation in group technology: A neural network approach. *International Journal of Production Research*, 30(6), 1353-1367.

15. Safaei, N., Tavakkoli-Moghaddam, R. (2009). An extended fuzzy parametric programming-based approach for designing cellular manufacturing systems under uncertainty and dynamic conditions. *International Journal of Computer Integrated Manufacturing*, 22(6), 538-548.

16. Delgoshaei, A., Ali, A. (2019). Review evolution of cellular manufacturing system's approaches: human resource planning method. *Journal of Project Management*, 4(1), 31-42.

17. Drolet, J., Abdulnour, G., Rheault, M. (1996). The Cellular Manufacturing Evolution. *Computers and Industrial Engineering*, 31(1-2), 139-142.

18. Paydar, M. M., Saidi-Mehrabad, M. (2015). Revised multi-choice goal programming for integrated supply chain design and dynamic virtual cell formation with fuzzy parameters. *International Journal of Computer Integrated Manufacturing*, 28(3), 251-265.

19. Paydar, M. M., Saidi-Mehrabad, M. (2017). A hybrid genetic algorithm for dynamic virtual cellular manufacturing with supplier selection. *The International Journal of Advanced Manufacturing Technology*, 92(5), 3001-3017.

20. Mahdavi, I., Aalaei, A., Paydar, M. M., Solimanpur, M. (2011). Multi-objective cell formation and production planning in dynamic virtual cellular manufacturing systems. *International Journal of Production Research*, 49(21), 6517-6537.

21. Rezazadeh, H., Ghazanfari, M., Sadjadi, S. J., Aryanezhad, M., Makui, A. (2009). Linear programming embedded particle swarm optimization for solving an extended model of dynamic virtual cellular manufacturing systems. *Journal of Applied Research and Technology*, 7(1), 83-108.

22. Rezazadeh, H., Mahini, R., Zarei, M. (2011). Solving a Dynamic Virtual Cell Formation Problem by Linear Programming Embedded Particle Swarm Optimization Algorithm. *Applied Soft Computing*, 11(3), 3160-3169.

23. Baykasoglu, A., Gorkemli, L. (2017). Dynamic virtual cellular manufacturing through agent-based modelling. *International Journal of Computer Integrated Manufacturing*, 30(6), 564-579.

24. Barve, S. B., Khodke, P. M., Sathe, P. P. (2011). Research issues in cellular manufacturing system. *International Journal of Applied Engineering Research*, 6(3), 291-302.

25. Şeker, A. (2016). Yalın üretim sisteminde kanban, tek parça akışı ve U tipi yerleştirme sistemleri. *The Journal of Academic Social Science Studies*, 50, 449-470.

26. Papaioannou, G., Wilson, J. M. (2010). The evolution of cell formation problem methodologies based on recent studies (1997–2008): Review and directions for future research. *European Journal of Operational Research*, 206(3), 509-521.

27. Soleymanpour, M., Vrat, P., Shankar, R. (2002). A transiently chaotic neural network approach to the design of cellular manufacturing. *International Journal of Production Research*, 40(10), 2225-2244.

28. İnternet: KCG Collage of Technology (2022), *Unit III Cellular Manufacturing*. Web: https://kcgcollege.ac.in/pdf/mech/study%20materials/ME%206703/Unit%20III-min.pdf, Son Erişim Tarihi: 20.03.2022.

29. Olorunniwo, F. O. (1997). A framework for measuring success of cellular manufacturing implementation. *International Journal of Production Research*, 35*(11), 3043-3062.*

30. İnternet: Sarker, B. R. (1999). Grouping efficiency measures in cellular manufacturing: a survey and critical review. *International Journal of Production Research*, 37(2), 285-314.

31. Hamedi, M., Ismail, N., Esmaeilian, G., Ariffin, M. (2012). Virtual cellular manufacturing system based on resource element approach and analyzing its performance over different basic layouts. *International Journal of Industrial Engineering Computations*, 3(2), 265-276.

32. Ko, K. C., Egbelu, P. J. (2003). Virtual cell formation. *International Journal of Production Research*, 41(11), 2365-2389.

33. Ratchev, S. M. (2001). Concurrent Process and Facility Prototyping for Formation of Virtual Manufacturing Cells. *Integrated Manufacturing Systems*.

34. Das, G. (2018). *Design and Operational Analysis of Virtual Cellular Manufacturing Systems.* Yayınlanmamış Doktora Tezi, Dayalbagh Educational Institute, Agra, 2-3.

35. Balakrishnan, J., Cheng, C. H. (2005). Dynamic cellular manufacturing under multiperiod planning horizons. *Journal of Manufacturing Technology Management*.

36. McLean, C. R., Bloom, H. M., Hopp, T. H. (1983). The Virtual Manufacturing Cell. In *Information Control Problems in Manufacturing Technology*,1982, 207-215.

37. Balakrishnan, J. Cheng, C. H. (2007). Multi-period planning and uncertainty issues in cellular manufacturing: a review and future directions. *European Journal of Operational Research*, 177(1), 281-309.

38. Akgün, E., Demir, M. (2018). Modeling course achievements of elementary education teacher candidates with artificial neural networks. *International Journal of Assessment Tools in Education*, 5(3), 491-509.

39. Öztemel, E. (2012). *Yapay Sinir Ağları*. İstanbul: Papatya Yayıncılık, 13-19.

40. İnternet: Sönmez, F. (2022). *Nöral Sistemlere Giriş Ders Notu*. Web: http://www.ferdisonmez.com/dersler/13/1-NORALSISTEMLEREGIRIS.DersNotu-PDF.pdf, Son Erişim Tarihi: 19.03.2022.

41. Krenker, A., Bešter, J., Kos, A. (2011). Introduction to the Artificial Neural Networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications.* InTech, 1-18.

42. Özkan, Ö. (2010). *Group Technology and Cellular Manufacturing with Artificial Neural Networks*. Yayınlanmamış Doktora Tezi, Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü, İzmir, 9-12.

43. Beale, M. H., Hagan, M. T., Demuth, H. B. (2018). Deep Learning Toolbox User's Guide. Maryland, USA. *The Mathworks Inc.*

44. Bebis, G., Georgiopoulos, M. (1994). Feed-Forward Neural Networks. *IEEE Potentials*, 13(4), 27-31.

45. Xu, S., Chen, L. (2008, June). *A Novel Approach for Determining the Optimal Number of Hidden Layer Neurons for FNN's and its Application in Data Mining.* 5th International Conference on Information Technology and Applications, Queensland, Australia.

46. İnternet: Sarle, W. S. (2002). Neural Networks FAQs.(Dal Newsgroup Usenet Comp. ai. Neuralnet) Web: ftp://ftp. sas. com/pub/neural. *FAQ. html. zip*, Son Erişim Tarihi: 14.04.2022.

47. Panchal, G., Ganatra, A., Kosta, Y. P., Panchal, D. (2011). Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers. *International Journal of Computer Theory and Engineering*, 3(2), 332-337.

48. Svozil, D., Kvasnicka, V., Pospichal, J. (1997). Introduction to Multi-Layer Feed-Forward Neural Networks. *Chemometrics and Intelligent Laboratory Systems*, 39(1), 43-62.

49. Pai, P. F., Lee, E. S. (2001). Parts Clustering by Self-Organizing Map Neural Network in a Fuzzy Environment. *Computers & Mathematics with Applications*, 42(1-2), 179-188.

50. Meyer-Baese, A., Thummler, V., Theis, F. (2006, July). *Stability Analysis of an Unsupervised Competitive Neural Network.* In The 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, Canada, 1025-1028.

51. İnternet: MathWorks Support Documentation Cluster with A Competitive Neural Network: Web: https://www.mathworks.com/help/deeplearning/ug/cluster-with-a-competitive-neural-network.html, Son Erişim Tarihi: 24.04.2022.

52. Castellano, G., Fanelli, A. M., Roselli, T. (2001, July). *Mining Categories of Learners by a Competitive Neural Network*. In IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222,*)* Italy, Bari, 945-950.

53. Hansen, L. K., Salamon, P. (1990). Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), 993-1001.

54. Zhou, Z. H., Wu, J., Tang, W. (2002). Ensembling Neural Networks: Many Could be Better than All. *Artificial Intelligence*, 137(1-2), 239-263.

55. Sharkey, A. J. C. (1996). On Combining Artificial Neural Nets. *Connection Science*, 8(3-4), 299-314.

56. Sharkey, A. J., Sharkey, N. E. (1997). Combining Diverse Neural Nets. *The Knowledge Engineering Review*, 12(3), 231-247.

57. Zhang, G., Hu, M. Y., Patuwo, B. E., Indro, D. C. (1999). Artificial neural networks in bankruptcy prediction: general framework and cross-validation analysis. *European Journal of Operational Research*, 116(1), 16-32.

58. Allende, H., Ñanculef, R., Salas, R. (2004, April). *Robust Bootstrapping Neural Networks*. In Mexican International Conference on Artificial Intelligence, Springer, Berlin, Heidelberg, 813-822.

59. Zhou, Z. H., Tang, W. (2006). Clusterer Ensemble. *Knowledge-Based Systems*, 19(1), 77-83.

60. Mijwel, M. M. (2018). Artificial Neural Networks Advantages and Disadvantages. *Researchgate.*

61. Anifowose, F., Labadin, J., Abdulraheem, A. (2017). In *Artificial Intelligence: Concepts, Methodologies, Tools, and Applications*. IGI Global, USA, 325-356.

62. İnternet: MathWorks Add-On Explorer. Web: https://www.mathworks.com/products/matlab/add-on-explorer.html, Son Erişim Tarihi: 04.04.2022.

63. İnternet: Beale, H., Hagan, T., Demuth, B. (2021). *MATLAB R2021a Deep Learning Toolbox User's Guide*. The MathWorks, Inc.

64. İnternet: *MATLAB R2022a App Building,* (2022). The MathWorks, Inc.

65. Liu, C., Wang, J., Zhou, M. (2018). Reconfiguration of Virtual Cellular Manufacturing Systems Via İmproved İmperialist Competitive Approach. *IEEE Transactions on Automation Science and Engineering*, 16(3), 1301-1314.

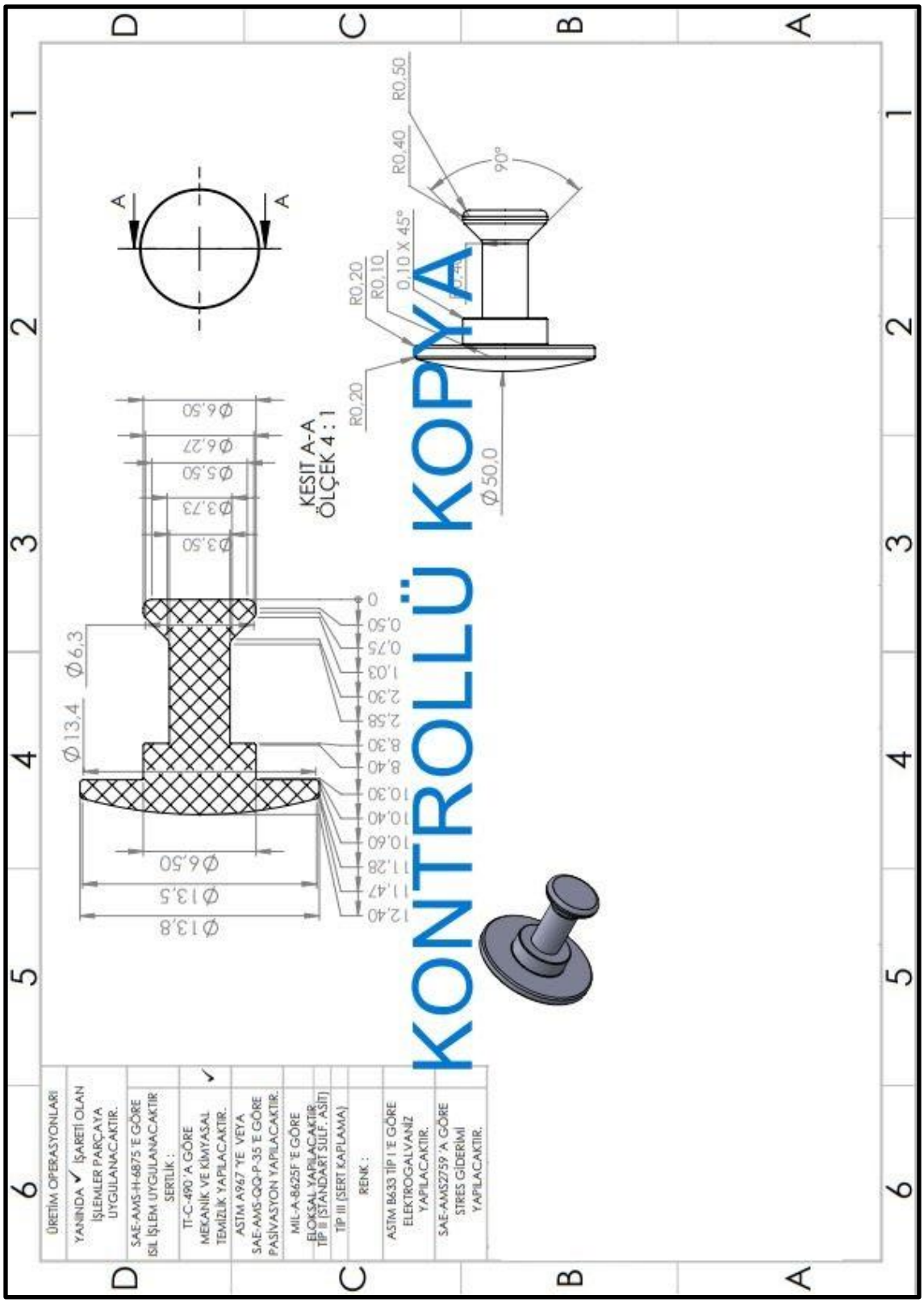
66. Kohonen, T. (2014). MATLAB İmplementations and Applications of the Self-Organizing Map. *Unigrafia Oy, Helsinki, Finland*, *2*.

67. Malakooti, B., Yang, Z. (2002). Multiple Criteria Approach and Generation of Efficient Alternatives for Machine-Part Family Formation in Group Technology. *IIE Transactions*, 34(9), 837-846.

68. Tian, J., Azarian, M. H., Pecht, M. (2014). *Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm*. In PHM Society European Conference, Maryland, USA, 2(1), 210-215.

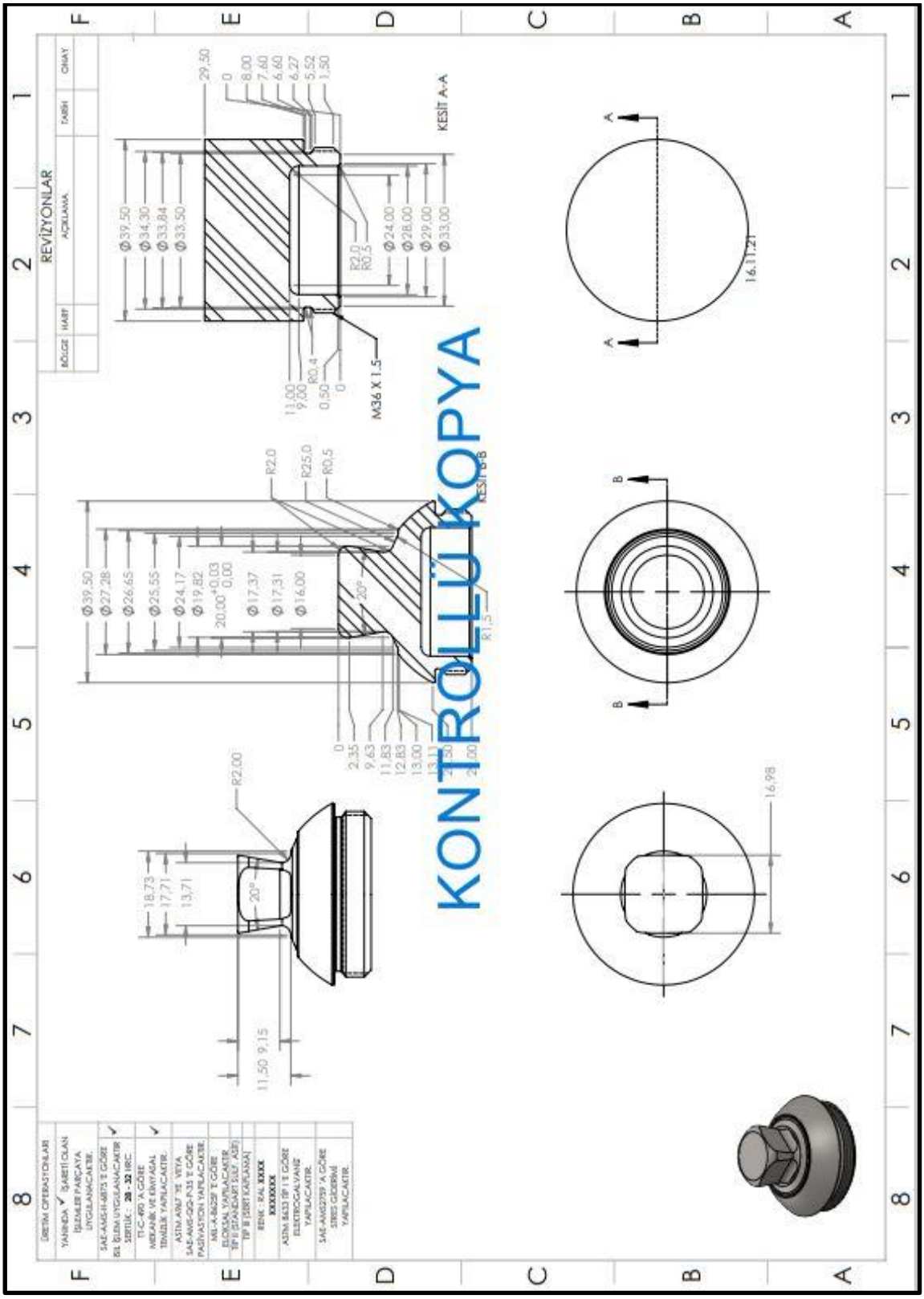
EKLER

EK-1. Parça 1 teknik resmi



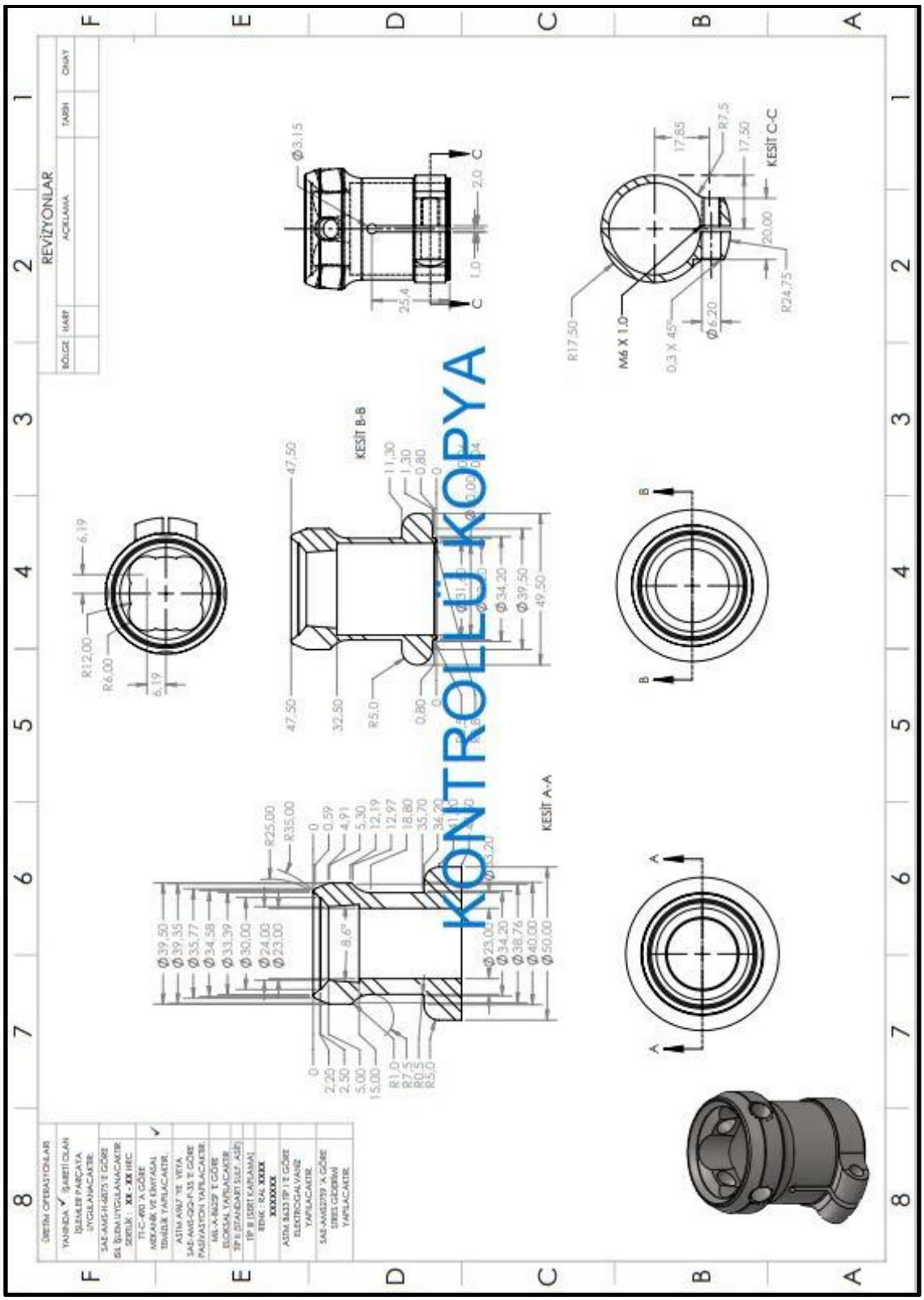
Resim 1.1. Parça 1 teknik resmi

EK-2. Parça 2 teknik resmi



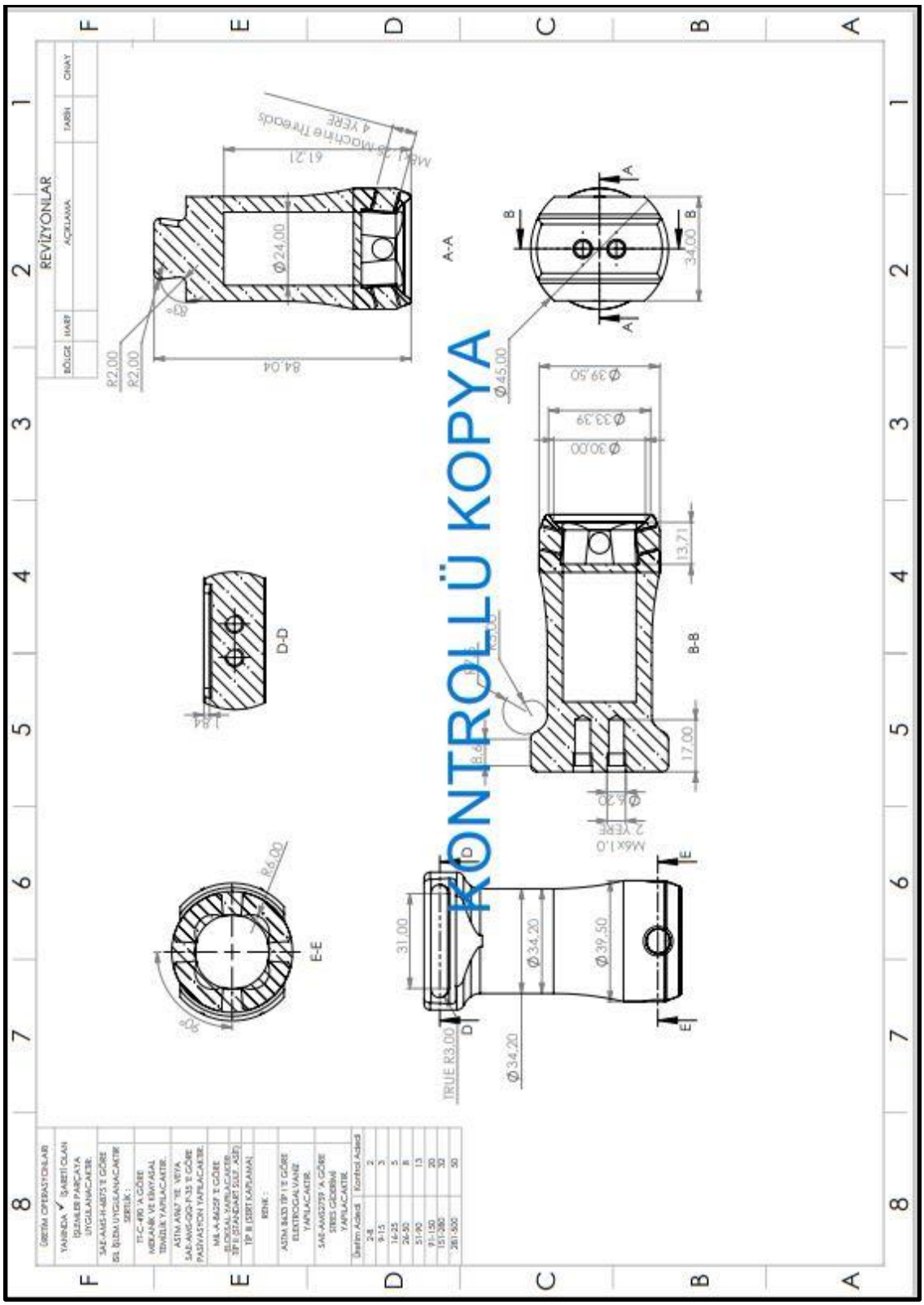
Resim 2.1. Parça 2 teknik resmi

EK-3. Parça 3 teknik resmi



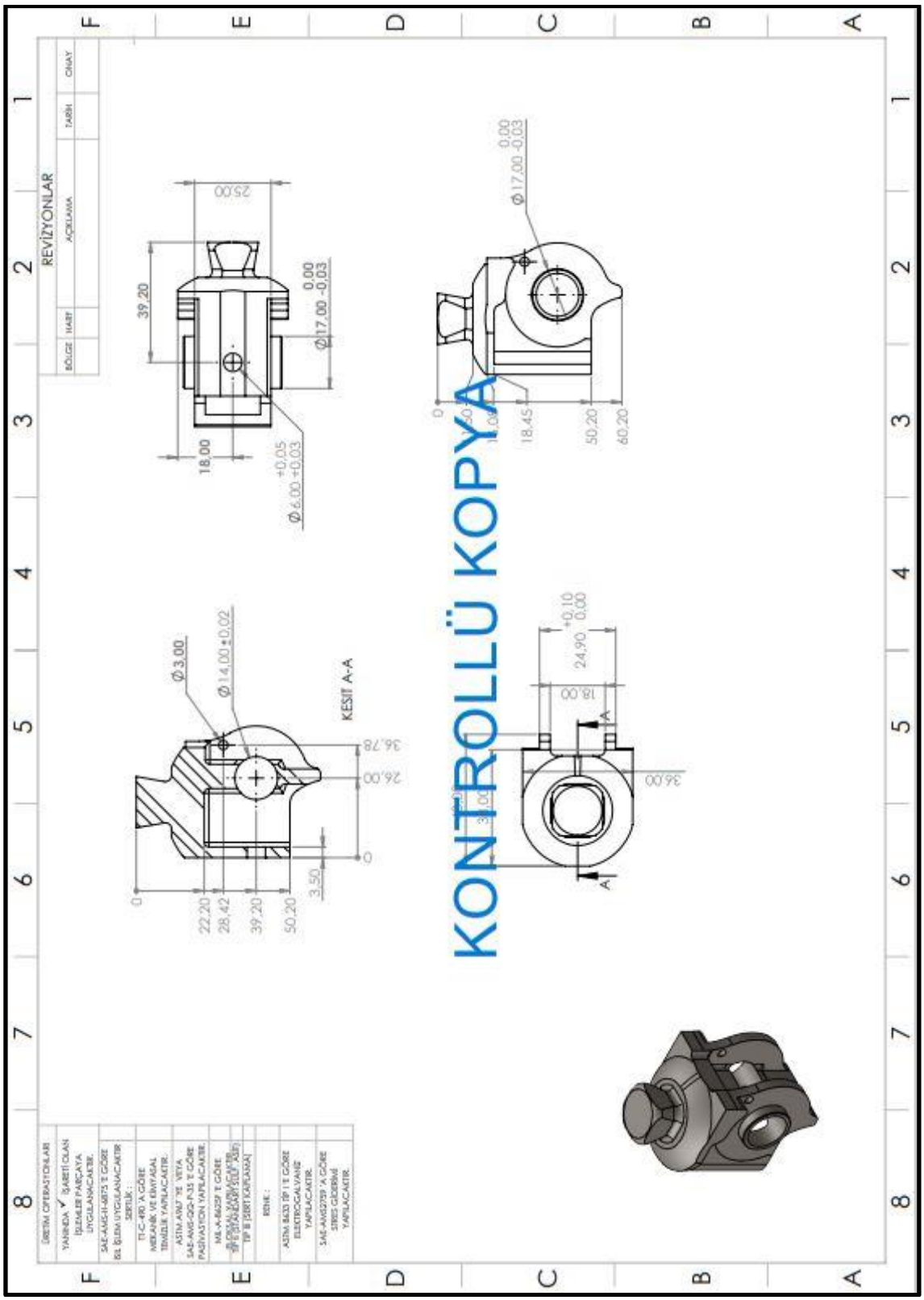
Resim 3.1. Parça 3 teknik resmi

EK-4. Parça 4 teknik resmi



Resim 4.1. Parça 4 teknik resmi

EK-5. Parça 5 teknik resmi



Resim 5.1. Parça 5 teknik resmi

GAZİ GELECEKTİR...