

**T.C.  
GİRESUN ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ÇARPIMSAL NORON MODEL YAPAY SİNİR  
AĞLARI İLE TÜKETİCİ FİYAT İNDEKSİ  
ÖNGÖRÜSÜ**

**YÜKSEK LİSANS TEZİ**

**Adı SOYADI**

**: Recep YILDIRAN**

**Tezin Enstitüye Verildiği Tarih**

**: .../06/2021**

**Enstitü Anabilim Dalı**

**: İstatistik**

**Tez  
Danışmanı**

**: Erol EĞRİOĞLU**

**Haziran 2021  
GİRESUN**

T.C.  
GİRESUN ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**ÇARPIMSAL NORON MODEL YAPAY SİNİR  
AĞLARI İLE TÜKETİCİ FİYAT İNDEKSİ  
ÖNGÖRÜSÜ**

**YÜKSEK LİSANS TEZİ**

**Recep YILDIRAN**

**Enstitü Anabilim Dalı : İstatistik**

**Bu tez 25/05/2021 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.**

**Doç. Dr.  
Nihat TAK  
Jüri Başkanı**

**Doç. Dr.  
Eren BAŞ  
Üye**

**Prof. Dr.  
Erol EĞRİOĞLU  
Üye**

**Doç. Dr.  
Bahadır KOZ  
Enstitü Müdürü**

## BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Recep YILDIRAN

25/05/2021

## TEŐEKKÜR

Tez alıőmamın baőından sonuna kadar her alanında engin bilgisi, birikim ve tecrübeleri ile bana destek olan ok kıymetli hocam Prof. Dr. Erol EĐRİOĐLU'na sonsuz teőekkür ve iten saygılarımı sunarım.

Manevi destekleriyle her zaman yanımda hissettiĐim ok deĐerli dostlarıma gönülden sonsuz teőekkürlerimi ve sevgilerimi sunarım.

Son olarak varlıĐıyla bana gü veren sevgili annem ve aileme en iten teőekkürlerimi ve sevgilerimi sunarım.



## İÇİNDEKİLER

TEŞEKKÜR.....	I
İÇİNDEKİLER .....	II
SİMGELER VE KISALTMALAR LİSTESİ .....	IV
ŞEKİLLER LİSTESİ.....	V
TABLolar LİSTESİ .....	VI
ÖZET.....	VII
SUMMARY.....	VIII
1. GİRİŞ.....	1
2. ENFLASYON NEDİR? .....	4
2.1. Hizmet Ve Mal Sepeti .....	5
2.1.1. Talep Enflasyonu .....	5
2.1.2. Maliyet Enflasyonu .....	6
2.1.3. Para Arzı Enflasyonu.....	6
2.1.4. Enflasyon Beklentisi .....	6
2.2. TEFE TÜFE Nedir .....	7
3. BİYoloJİK VE YAPAY SİNİR AĞLARI .....	8
4. YAPAY SİNİR HÜCRESİ .....	11
5. YAPAY SİNİR AĞI TÜRLERİ.....	13
6. EĞİTİM ALGORİTMALARI .....	16
7. ÇARPIMSAL NÖRON MODEL YAPAY SİNİR AĞLARI.....	18
8. ÇARPIMSAL NÖRON MODEL YAPAY SİNİR AĞLARININ EĞİTİMİ.....	20

8.1. Geri Yayılım Algoritması ile Çarpımsal Nöron Model Yapay Sinir Ağı Eğitimi.....	20
8.2. Parçacık Sürü Optimizasyonu Algoritması ile Çarpımsal Nöron Model Yapay Sinir Ağı Eğitimi.....	22
8.3. Yapay arı kolonisi Optimizasyonu Algoritması ile Çarpımsal Nöron Model Yapay Sinir Ağı Eğitimi.....	26
<b>9. ÇARPIMSAL NÖRON MODEL YSA İLE ZAMAN SERİSİ ÖNGÖRÜ PROBLEMİ İÇİN BİR ALGORITMA.....</b>	<b>31</b>
<b>10. TÜKETİCİ FİYAT ENDEKSİ ZAMAN SERİSİNİN ÇÖZÜMLENMESİ.....</b>	<b>35</b>
<b>11. SONUÇ VE TARTIŞMA.....</b>	<b>44</b>
<b>KAYNAKLAR.....</b>	<b>45</b>
<b>EKLER.....</b>	<b>48</b>
<b>ÖZGEÇMİŞ.....</b>	<b>52</b>

## SİMGELER VE KISALTMALAR LİSTESİ

ARIMA	: Otoregresif Bütünleşik Hareketli Ortalamalar
GSYİH	: Gayri Safi Yurtiçi Hasıla
ÇNM-YSA	: Çarpımsal Nöron Model Yapay Sinir Ağı
DNM-YSA	: Dendrit Nöron Model Yapay Sinir Ağı
MAE	: Ortalama Mutlak Hata
MAPE	: Hatanın Mutlak Yüzdellik Ortalaması
MASE	: Ortalama Mutlak Yüzde Hata
MLP-YSA	: Çok Katmanlı Algılayıcı Yapay Sinir Ağı
MSE	: Ortalama Yüzde Hata
ÜFE	: Üretici Fiyat Endeksi
PSO	: Parçacık Sürü Optimizasyonu
PS-YSA	: Ps-Sigma Yapay Sinir Ağı
SMAPE	: Simetrik Ortalama Mutlak Yüzde Hata
RMSE	: Hata Kareler Ortalaması Karekökü
TEFE	: Toptan Eşya Fiyatlar Endeksi
TÜFE	: Tüketici Fiyat Endeksi
TÜİK	: Türkiye İstatistik Kurumu
YAK	: Yapay arı kolonisi
YSA	: Yapay Sinir Ağları

## ŞEKİLLER LİSTESİ

Şekil 3.1.	: Biyolojik Sinir Hücresi.....	8
Şekil 4.1.	: Yapay Sinir Hücresi.....	11
Şekil 5.1.	: Mimari yapılara göre yapay sinir ağlarının sınıflandırılması.....	13
Şekil 5.2.	: Nöron Sayısına Göre yapay sinir ağı örnekleri.....	13
Şekil 5.3.	: Nöronların bağlantı yönlerine göre yapay sinir ağı örnekleri.....	14
Şekil 5.4.	: Nöronların bağlantı yönlerine göre yapay sinir ağı örnekleri.....	14
Şekil 6.1.	: Yapay Sinir Ağları Eğitim Algoritmaları Sınıflandırması.....	16
Şekil 7.1.	: ÇNM-ANN'nin öngörü problemi için çözüm mimarisi.....	18
Şekil 8.1.	: Geri yayılım öğrenme algoritmasının akış diyagramı.....	22
Şekil 8.2.	: Parçacık sürü optimizasyonu algoritmasının akış diyagramı.....	26
Şekil 8.3.	: Yapay arı kolonisi Algoritmasının Akış diyagramı.....	28
Şekil 9.1.	: ÇNM-YSA Mimarisi.....	33
Şekil 10.1.	: Ocak 2005- Ekim 2020 TUF E Aylık % Değişim Zaman Serisi.....	35
Şekil 10.2.	: TUF E Zaman Serisi İçin Box-Plot Grafiği.....	35
Şekil 10.3.	: Ocak 2005- Ekim 2020 Düzeltilmiş TUF E Aylık % Değişim Zaman Serisi	36
Şekil 10.4.	: TUF E Zaman Serisi İçin Model (10.1)'e göre Birim Kök Testi Sonucu...	37
Şekil 10.5.	: TUF E Zaman Serisi İçin Model 10.2'ye göre Birim Kök Testi Sonucu.....	38
Şekil 10.6.	: TUF E Zaman Serisi İçin Otokorelasyon Grafiği.....	39
Şekil 10.7.	: TUF E Zaman Serisi İçin Kısmi Otokorelasyon Grafiği.....	39
Şekil 10.8.	: R Programından elde edilen ETS yöntemi çıktıları.....	42
Şekil 10.9.	: R Programından elde edilen ARIMA yöntemi çıktıları.....	42

## TABLÖLAR LİSTESİ

<b>Tablo 10.1.</b>	: Test kümesi öngörü sonuçları ve yöntemlerin performansları.....	41
<b>Tablo 10.2.</b>	: Çok adım öngörü performansı karşılaştırması.....	43
<b>Tablo 11.1.</b>	: TÜFE zaman serisi için öngörüler.....	44



## ÖZET

### ÇARPIMSAL NÖRON MODEL YAPAY SİNİR AĞLARI İLE TÜKETİCİ FİYAT İNDEKSİ ÖNGÖRÜSÜ

Enflasyonun öngörülmesi, öngörü yöntemlerinin en önemli uygulamalarından bir tanesidir. Bu tez çalışmasında, Türkiye enflasyon zaman serisi klasik ve makine öğrenmesi yöntemleri ile öngörülmüştür. Özellikle, çarpımsal nöron model yapay sinir ağının enflasyon öngörüsü için performansı klasik ve diğer makine öğrenmesi öngörü yöntemleri ile karşılaştırılmıştır. Çarpımsal nöron modele dayalı yeni bir öngörü algoritması tez kapsamında ortaya koyulmuştur. Gerçekleştirilen uygulamaların sonucunda önerilen öngörü yönteminin enflasyon öngörüsü için en iyi sonuçları ürettiği görülmüştür.

**Anahtar Kelimeler:** Öngörü, enflasyon, yapay sinir ağları, çarpımsal nöron model yapay sinir ağları.

## **SUMMARY**

### **FORECASTING CONSUMER PRICE INDEX WITH MULTIPLICATIVE NEURON MODEL ARTIFICIAL NEURAL NETWORKS**

The forecasting of inflation is one of the most important applications of forecasting methods. In this thesis, Turkey inflation time series is forecasted by using classical and machine learning methods. Especially, the performance of the multiplicative neuron model artificial neural networks for inflations forecasting is compared with classical and other machine learning methods. A new forecasting algorithm that is based on multiplicative neuron model artificial neural networks are proposed in this thesis. As a result of applications, it is concluded that the proposed forecasting algorithm produced the best forecasting results for inflation forecasting.

**Keywords:** Forecasting, Inflation, artificial neural networks, multiplicative neuron model artificial neural networks.

## 1. GİRİŞ

Son yıllarda zaman serisi öngörü probleminin çözümü için makine öğrenmesi teknikleri sıklıkla kullanılmaya başlanmıştır. Yapay sinir ağları özellikle doğrusal olmayan yapıları ve veriye göre şekil almaları sayesinde bir çok zaman serisinin çözümlenmesinde başarılı sonuçlar üretebilmektedir. Literatürde en yaygın kullanılan yapay sinir ağı modeli çok katmanlı algılayıcı olmakla birlikte, son yıllarda uzun kısa dönem hafıza gibi derin yapay sinir ağlarının öngörü problem için başarılı sonuçlar ürettiği görülmektedir. Derin yapay sinir ağları, çok katmanlı algılayıcıya göre daha fazla parametre içeren ancak çok katmanlı algılayıcıya göre daha fazla bilgiyi daha az parametre kullanarak çözümleyebilmektedir. Zaman serisi öngörü literatüründe daha basit ve daha az parametrelili yöntemlerin, genelleştirme yeteneğinin daha iyi olduğu görülmüştür. Yadav ve ark. (2007)'de Derin yapay sinir ağları ve çok katmanlı algılayıcıya göre çok daha az parametre ve nöron ile çalışan çarpımsal nöron model yapay sinir ağını önerilmiştir. Çarpımsal nöron model yapay sinir ağı (ÇNM-YSA) tek nöronlu bir mimariye sahiptir. Çarpımsal nöron model yapay sinir ağının eğitimi için literatürde yoğun çalışmalar gerçekleştirilmiştir. Yeh (2013) çalışmasında yeni bir sürü optimizasyonu yöntemi ile ÇNM-YSA'nın eğitimi gerçekleştirmiştir. Chatterjee ve ark. (2013) ÇNM-YSA'nın eğitimi için geri yayılım öğrenme algoritması ve reel kodlu genetik algoritmaya dayalı bir yaklaşım önermiştir. Wu ve ark. (2013b) çalışmasında enerji tüketimi zaman serisi öngörüsü için ÇNM-YSA'nın online eğitim algoritmasını ortaya koymuşlardır. Cui ve ark. (2015) ÇNM-YSA'nın eğitimi için diferansiyel gelişim ve ateş böceği algoritmalarının melezlenmesi ile bir algoritma önermişlerdir. Samatha (2015) çalışmasında gradiyente dayalı yöntemleri tercih ederken, Bas ve ark. (2016a) ÇNM-YSA için dayanıklı bir eğitim algoritması önermişlerdir. Bas ve ark. (2016) çalışmasında diferansiyel gelişim algoritması ile ÇNM-YSA'nın eğitimini gerçekleştirmişlerdir.

ÇNM-YSA'nın çeşitli modifikasyonları literatürde önerilmiştir. Aladag ve ark. (2013) mevsimsel zaman serilerinin çözümlenmesi için çarpımsal nöron modele dayalı yeni bir yapay sinir ağı önermişlerdir. Wu ve ark. (2013a) ve Wu ve ark.

(2015) Kalman filitresi ve ÇNM-YSA'ya dayalı melez bir yaklaşım ile enerji tüketimi ve rüzgar hızı zaman serilerini çözümlenmişlerdir. Wu ve ark. (2013c) motor sistemlerinin güvenilirliği için iterative doğrusal olmayan filitreler ile ÇNM-YSA'yı kullanmışlardır. Gundogdu ve ark. (2016) çalışmasında Gaussian aktivasyon fonksiyonlu ÇNM-YSA'yı ve onun eğitim algoritmasını önermişlerdir. Yolcu ve ark. (2016) çalışmasında bootstrap yaklaşımlar ile ÇNM-YSA için aralık tahminlerini elde etmişlerdir. Egrioglu ve ark. (2018) çalışmasında geri beslemeli ÇNM-YSA önerilmiştir. Cagcag Yolcu ve ark. (2018) çalışmasında ağırlıkları otoregresif yapıda zamanla değişen ÇNM-YSA yapay sinir ağı önerilmiştir. Corba ve ark. (2019) çalışmasında bir tür geri beslemeli ağ olarak ARCH tipli ÇNM-YSA önerilmiştir. Literatür incelendiğinde ÇNM-YSA ve onun modifikasyonlarının başarılı öngörü sonuçları üreten ve diğer YSA modellerine göre daha basit mimariye sahip oldukları gözlemlenmiştir.

Enflasyon öngörüsü, öngörü yöntemlerinin kullandığı alanlardan bir tanesidir. Türkiye'de enflasyonun öngörüsü için bir çok makale yayınlanmıştır. Köse ve ark. (2000) enflasyon verileri kullanarak VAR modelleri ile yapısal ekonometrik modellerin öngörü başarısını karşılaştırmıştır. Domaç (2003) Türkiye için enflasyonun öngörüsünü kuramsal temellere dayanan modeller kurmuştur. ARIMA modeli, kar marjı modelleri, parasal açık modelleri ve Phillips Eğrisi arasında en iyi modelin Phillips Eğrisi modeli olduğuna karar verilmiştir. Kızılkaya (2007) enflasyonu konu alan başka bir çalışma olup, 1923-2014 tarihleri arasındaki işsizlik oranı 1969-2014 tarihleri arasındaki enflasyon oranları verisi kullanılmıştır. YSA modeli olan ÇKA ve ARIMA modeli olan Box-Jenkins ile yapılan çalışmada bulunan bulguların merkez bankasının tahminlere göre daha doğru sonuçlar olduğu savunulmuştur. Erilli ve ark. (2010) Türkiye enflasyon tahmini için ileri ve geri beslemeli yapay sinir ağlarının bir melez yaklaşımı ile enflasyonu tahmin etmişlerdir. Elde ettikleri sonuçların enflasyon başarı anketinden daha doğru öngörü sonuçları olduğunu saptamışlardır. Liliana (2012) GSYİH öngörüsünde bulunurken YSA modelini ele almışlardır. Ortaya çıkan tahminler hükümetçe yayınlanan tahminlerden başarılı olduğunu göstermiştir.

Bu çalışmada enflasyon öngörüsü için ÇNM-YSA'ya dayalı bir algoritma önerilmiş ve ÇNM-YSA'nın performansı klasik zaman serileri ve bazı YSA modelleri ile karşılaştırılmıştır.

Tez çalışmasının ikinci bölümünde enflasyon hakkında genel bilgi, üçüncü bölümde biyolojik ve yapay sinir ağları, dördüncü bölümde yapay sinir hücresi, beşinci bölümde yapay sinir ağı türleri, altıncı bölümde yapay sinir ağları için eğitim algoritmaları hakkında özet bilgi sunulmuştur. Yedinci bölümde ÇNM-YSA'lar ve sekizinci bölümde ÇNM-YSA'nın eğitimi için eğitim algoritmaları verilmiştir. Dokuzuncu bölümde ÇNM-YSA'ya dayalı bir öngörü yöntemi algoritması tanımlanmıştır. Onuncu bölümde Türkiye için tüketici fiyat endeksi kullanılarak enflasyon öngörüsü gerçekleştirilmiştir. Tezin son bölümünde ise elde edilen sonuçlar tartışılmıştır.

## 2. ENFLASYON NEDİR?

Enflasyon üretilen malların ve sunulan hizmetlerin fiyatlarındaki artışa verilen isimdir. Başka bir deyişle paranın alım gücünün düşmesi olarak da adlandırılabilir. Örneğin, geçen sene 5 TL den aldığımız herhangi bir ürünün bugün 7 TL olması durumudur. Bu durumda insanlar ürünün fiyatının arttığını düşünürken aslında gerçek paramızın değer kaybettiğidir. Peki, ürün aynı ürün olmasına rağmen piyasada ki fiyatı neden artar? Bu fiyat artışı 2 sebebe dayanır.

- a) Maliyet Artışı: Ürünü üretme maliyetlerinin artmasından dolayı fiyat artışları yaşanmaktadır.
- b) Talep Artışı: Ürüne olan talebin artması nedeniyle fiyat artışları gözlenmektedir.

Ekonomi dediğimiz şey dengeye bağlı bir faktördür ve bir binayı ayakta tutan kolonlara benzer. Eğer kolonlardan biri zarar görürse bina çöker. Bu örnekte ki bina ekonomiye benzer ekonomik kolonları ayakta tutarsa arz ve talep faktörlerinin doğru orantısıdır. Üretim yoksa tüketim yoktur, eğer tüketim yoksa üretim yoktur.

Bu dengeyi sağlamanın asıl temelleriyse tüketicinin ekonomikselse güvenini sağlamaktır. Eğer tüketici üreticinin fiyatlarındaki doğruluğa ve tutarlılığa güvenirse tüketim sağlanır. Bu konuda, para politikalarını belirleyen merkez bankalarının asıl gayesi üretim ve tüketim dengesini ayakta tutabilmek için fiyatlardaki bu istikrarı sağlamalarıdır. İleri gelen İktisat profesörleri ve ekonomistlerin bir ülke için makul buldukları enflasyon oranı %2'dir. Bu sebeple merkez bankaları genel olarak stratejilerini bu yüzdellik dilimi baz alarak belirlerler. Ekonomi seyri dinamiktir bu sebeple sürekli değişimler göstermesi normaldir; ancak bu değişimler minimum düzeyde tutulmalıdır ki sert dalgalanmaların tespit edilerek önüne geçilmesi gerekir. Merkez bankalarının enflasyon konusunda ki bir diğer görevleri özellikle daha başarılı sonuçlar elde etmek adına bu konudaki yetkililerin görüşlerini ve açıklamaları dikkate alarak oluşturdukları öngörülerle enflasyon oranı konusunda beklenti yönetim planlamaları yapmalarıdır. Bu durum ülkenin saygınlığını artıracaktır.

Enflasyon mal ve hizmetlerdeki fiyatların sürekli artış göstermesi olarak tanımlanıp, ekonomide de bir hastalık haline gelmiştir. Öyle ki tüketici bu mal ve hizmetlerdeki harcamalarını azaltmak durumunda kalır. Kısılan taleplerde üretimde düşüşe neden olurken bu düşüş beraberinde istihdamda azalmaya ve işsizliğin artmasına neden olur. Bu noktada da bir iyileşme sağlanmazsa gerekli ekonomi için tehdit olmaya devam eder.

## **2.1. Hizmet ve Mal Sepeti**

Hizmet ve Mal Sepeti endekslerin hesaplanabilmesi için belirlenen bazı mal ve hizmetlerin zaman içindeki (12 ay) değişimini ölçmektedir. Enflasyon sepeti vs. olarak da adlandırılabilir.

Ekonomistler arz ve talep kavramlarını enflasyonun nedenlerinden biri olarak tanımlar. Bir ekonomideki enflasyonu; arz şokları, talep şokları, atalet durumları ve politik dönemler açıklar. İktisadi araştırmalar enflasyonu etkileyen dört ana etken türü üzerinde durmaktadır. Bunları şöyle sıralayabiliriz; talep enflasyonu, maliyet enflasyonu, para arzı ve enflasyon beklentileri.

### **2.1.1. Talep Enflasyonu**

Özellikle işsizlik oranlarında düşüşler ve ekonomi taraflarında toparlanma ve beraberinde büyüme dönemleri tüketim yapanlar tarafından güven oluşturur. Tüketici harcamalarında rahatlamaya gider. Böyle bir noktada talep arzdan daha hızlı bir büyüme kaydeder. Bu teori diğer bir deyişle "az mal, çok para" olarak da özetlenebilir. Söz konusu süreçte taleplerde artış görülmesine rağmen üreticilerin bu talebe oranla aynı hızda üretim yapamaması talep fazlası oluşturur. Kaynaklar üzerinde ki bu baskı sonucu talep enflasyonları görülmektedir.

### **2.1.2. Maliyet Enflasyonu**

Maliyet enflasyonu ilgili ekonomide emtia fiyatlarının sürekli artması veya toplam arzda izlenen seyri sola kaydıran her türlü durum, örneğin savaş, ekonomik küçülme, doğal afetler, girdi fiyatlarında artış vs gibi şartların neden olduğu enflasyon türüdür. Bu şartlar üretim maliyetlerinde artış ve toplam arzda düşüş beraberinde fiyatların genel seviyesinde yükselişe neden olur. Aynı zamanda devlet tarafı yapılan vergilendirmeler vs. gibi durumlarda maliyet enflasyonuna yol açar. Bu enflasyon çeşidi kendi içinde ikiye ayrılır. Bunlardan ilki kar enflasyonu ikincisi ise ücret enflasyonudur. Ücret enflasyonu ücretler seviyesindeki yükselişe rağmen verim seviyesinin durgun olması sebebiyle üretimdeki maliyetlerin yükselişidir.

### **2.1.3. Para Arzı Enflasyonu**

Aşırı arzdan kaynaklanmaktadır. Belirli bir süreçte ilgili ekonomide dolaşımda olan toplam para miktarı, para arzı olarak isimlendirilir. Bir mal veya hizmete çok fazla arz varsa o mal veya hizmetin değeri düşer. Bu durum fiyatlarda yüksekliğe yol açar.

### **2.1.4. Enflasyon Beklentisi**

Toplumdaki üretici ve tüketicilerin ileriki süreçlerde mal ve hizmetlerde fiyat artışları görebilecekleri beklentisiyle ileriye dönük ücret talepleri durumunda mal ve hizmet fiyatlarını yükseltme durumlarıdır. Kira artırımını, alım-satım durumları vb. örnek gösterilebilir. Enflasyon üzerindeki İleriye dönük beklentilerin gerçekleşmesi enflasyonun kendini doğrulaması olarak adlandırılır.

## 2.2. TEFE TÜFE NEDİR?

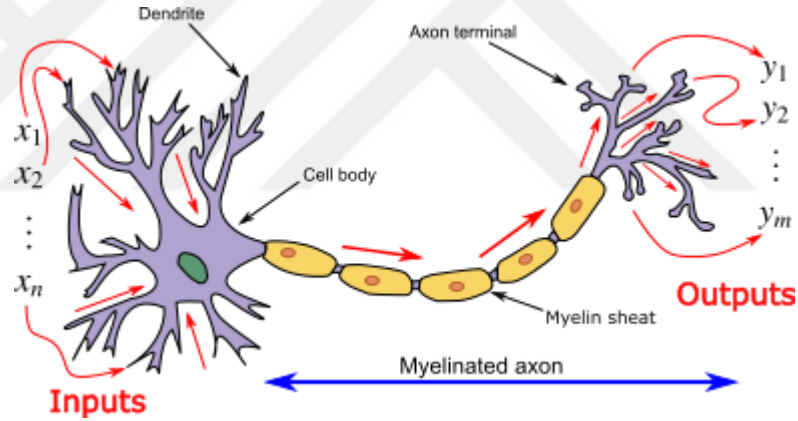
Enflasyon hesaplamalarında farklı fiyat endeksleri kullanılmaktadır. Toptan Eşya Fiyatlar Endeksi (TEFE) ve Tüketici Fiyatlar Endeksi (TÜFE)'deki değişim, enflasyon ölçümünde sıkça kullanılan fiyat endeksleridir. Türkiye İstatistik Kurumu (TÜİK) tarafından hesaplanır.

TEFE, kullandığımız malların tüm ham maddelerini içine almaktadır. Yurtiçi üretilen mal gruplarının oluşan bir sepetin satışına kadar uzanan fiyat değişimlerini kapsamaktadır. Başka bir deyişle toptan satışa kadar sunulan malların fiyat artışlarını gösteren endekstir.

TÜFE, hane halkının tüketimine yönelik daha önce alınmış mal ve hizmetlerin daha sonraki tarihte fiyat karşılaştırılmasına yarar. TÜFE'nin asıl amacı tüketime konu olan mal ve hizmetleri belirli bir tarih içinde fiyatlarının değişimiyle enflasyon oranının hesaplanmasında kullanılmaktadır. Hane halkı, yabancı ziyaretçilerin ve kurumsal nüfusun yani tüm tüketicilerin yaptığı tüm tüketim harcamalarını kapsamaktadır.

### 3. BİYOLOJİK VE YAPAY SİNİR AĞLARI

Yapay sinir ağlarını anlamamanın ilk yolu biyolojik sinir hücrelerini tanımadır. Biyolojik sinir hücrelerinin yapısına bakıldığında dört bölüme ayrılmaktadır. Bu bölümler çekirdek, akson, dendrit ve bağlarından oluşmaktadır. Biyolojik sinir hücrelerinin gövdesine soma denilmektedir. Çekirdek bu bölgede bulunur ve bilginin işleme merkezi olarak görev yapar. Gövdeden çıkan uzantılar dendritler ve aksonlardır. Dendritler sinyalleri almakta, aksonlar ise sinyalleri iletme görevini yapmaktadır. Aksonlar yapı olarak dendritlerden daha kalındır. Bu sebeple aldığı sinyalleri daha hızlı olarak iletmektedir. Başka bir hücreden gelen sinyalleri alma görevinde ise sinapslar etkilidir. Şekil 3.1’de biyolojik sinir hücrelerini gösteren bir şekil verilmiştir.



Şekil 3.1. Biyolojik Sinir Hücresi (“Wikipedia 2021” Biological neuron models)

İnsan beyni incelendiğinde, yaklaşık 100 milyar biyolojik sinir hücrelerinin olduğu düşünülmektedir. Bu hücreler bir biri ile koordineli olarak çalışarak aldıkları bilgiyi işleyip beyin hücrelerine kadar ulaştırırlar ve merkezi sinir sisteminde işlenen bilgiyi tekrar organlar veya duyu sistemlerine iletiler. Yapay sinir hücreleri de biyolojik sinir hücrelerinin yapısını örnek alarak çalışmaktadır.

Beynin üretebilme, oluşturabilme, ezber yapma ve bağıntı kurma gibi yeteneklerinden etkilenen bilim insanları, insan beyninin basit bir matematiksel modelini ortaya çıkarmışlardır. Dolayısıyla yapay sinir ağları biyolojik beyinden

ilham alınarak oluşturulmuştur. Yapay sinir ağları, girdi ve çıktılar arasındaki ilişkileri öğrenebilen bilgisayar yazılımlarıdır.

Yapay sinir ağlarının, biyolojik sinir ağının çalışma prensiplerinden taklit edebildikleri özellikleri aşağıda özetlenmiştir.

- İnsan beyni günlük olaylardan edindiği örnekler sayesinde öğrenme işlemini gerçekleştirmektedir. Bu öğrenme insan beynindeki sinir hücrelerinin arasındaki bağlantıların ayarlanması ile gerçekleştirilmektedir. Yapay sinir ağlarında ise öğrenme yapay sinir hücreleri arasındaki bağlantı ağırlıklarının optimize edilmesi ile gerçekleştirilir.

- İnsan beyni ilk kez karşılaştığı ya da tam olarak bilmediği durumlarda fikir yürütme şeklinde bir mekanizma ile çıkarım yapma, karar verme ya da tahmin yapabilmektedir. Yapay sinir hücresi de benzer şekilde daha önce öğrenmediği örnekler için çıkarım yapabilmektedir. Mevcut örneklerden girdi çıktı ilişkilerini öğrenen yapay sinir ağları, elinde olmayan örnekler için tıpkı insan beyni gibi çıkarım yapmayı sağlayabilmektedir.

- İnsan beyni her gün öğrendiği yeni bilgiler ile bilgi dağarcığını geliştirebilmekte ve daha doğru kararlar verebilmekte ve öngörüler yapabilmektedir. Yapay sinir ağları da benzer şekilde yeni öğrenme örnekleri ile kendini adapte edebilme yeteneğine sahiptir. Yani yapay sinir ağlarının öğrenmesi tıpkı insan beyni gibi ucu açık bir süreçtir.

- İnsan hatalarında ders alabilen bir varlık olup, zaman içinde daha mükemmel doğru bir arayış içindedir. Yapay Sinir Ağlarını öğrenme sürecinde de (geri yayılım öğrenme algoritmasında) örnekleri öğrenmede ortaya çıkan hatalar kullanılmakta yani bir nevi hatalardan dersler çıkarılmaktadır.

- İnsan beyni çıkarım yaptığı durumlarda geçmişte benzer durumlardaki tecrübelerini dikkate alarak çıkarımlarını revize edebilmektedir. Bu durum yapaya

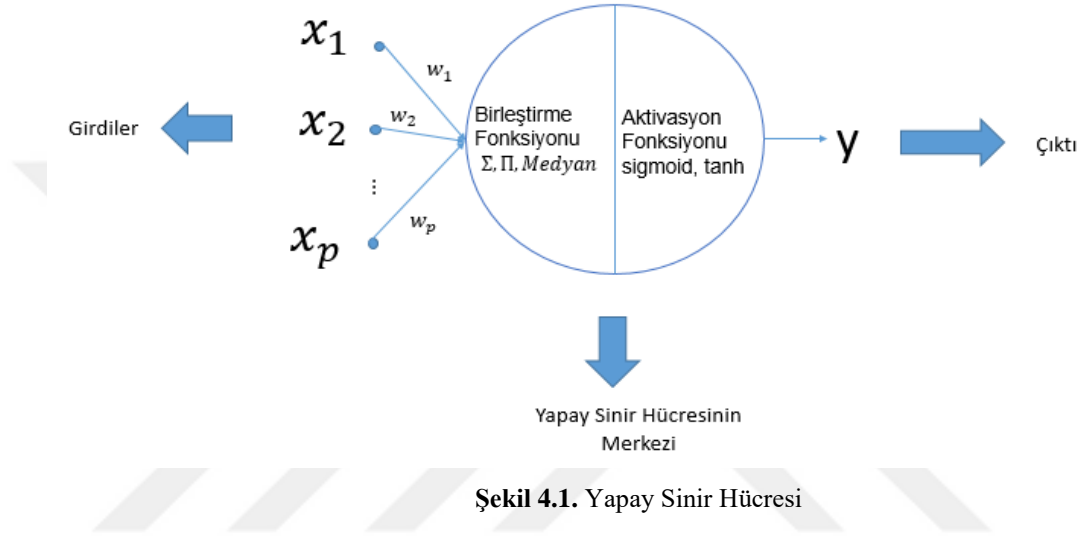
sinir ađlarında geri besleme bađlantıları ile sađlanarak, ıkarım sonuları gemiř kararlar ya da gemiř ıkarımlardaki hatalar ile gncellenmektedir.

- İnsan beyni gereksiz bilgileri ayıklayarak, ıkarım yapması gereken konuda sadece gerekli bilgiler ile alıřabilmektedir. Son zamanlarda nerilen hafıza kapılarına dayalı geri beslemeli ađlar gereksiz bilgilerin dıřlanarak bir đrenmenin gerekleřtirilmesini sađlayabilmektedir.



#### 4. YAPAY SİNİR HÜCRESİ

Biyolojik sinir hücrelerinin çok basit taklitleri olan yapay sinir hücreleri matematiksel modellerden ibaret olmaktadır. Literatürde toplamsal, çarpımsal, medyan, budanmış ortalama yapay nöron modelleri mevcuttur. Şekil 4.1’de yapay sinir hücrelerini genel olarak gösteren bir şekil verilmiştir.



Her yapay sinir hücrelerinin en az bir adet girdisi olmalıdır. Girdilere ilave olarak bazı yapay sinir hücresi modellerinde sabit ve bir değerini alan yan değerleri de bulunmaktadır. Şekil 4.1.’de  $p$  adet girdi  $x_1, x_2, \dots, x_p$  ile gösterilmiştir. Her girdiye karşılık gelen bir ağırlık vardır ve girdilerden gelen bilgiler bu ağırlıkların çarpımı ile yapay sinir hücresi merkezine iletilmektedir. Ağırlıklar  $w_1, w_2, \dots, w_p$  biyolojik sinir hücresindeki gibi girdilerin bağlantılarının güçlendirilip zayıflatılmasına olanak sağlamaktadır. Öğrenme bu ağırlıkların optimize edilmesi ile sağlanmaktadır.

Yapay sinir hücresinin merkezinde birleştirme fonksiyonu ve aktivasyon fonksiyonu vardır. Birleştirme fonksiyonu yapay sinir hücresine gelen bilgilerin bir araya getirilmesi görevini üstlenmektedir. Yapay sinir hücresi toplama, çarpma ve medyan gibi birleştirme fonksiyonları kullanılabilir. Aktivasyon fonksiyonu ise yapay sinir hücresinde birleştirilen bilginin çıktıya dönüştürülmesini sağlayan bir

fonksiyondur. Aktivasyon fonksiyonunun sınırlı bir fonksiyon olarak seçilmesi yapay sinir hücreleri ile yapay sinir ağları oluşturulduğunda oldukça önem taşımaktadır. Böylece yapay sinir ağları fonksiyon yaklaşılabilmek özelliğini kazanmaktadır. Farklı aktivasyon fonksiyonları yazında mevcut olmakla birlikte sigmoid ve tanjant hiperbolik fonksiyonları en sık kullanılanlarıdır. Sonuç olarak yapay sinir hücresi kendisine gelen sinyalleri birleştiren ve dönüştüren bir işlem birimi olarak çalışmaktadır.

Birleştirme fonksiyonunun toplamsal olduğu, aktivasyon fonksiyonunun sigmoid fonksiyon olarak seçildiği durumda yapay sinir hücresinin matematiksel modeli aşağıdaki gibi olur.

$$y = \frac{1}{1 + \exp(-\sum_{i=1}^p w_i x_i)} \quad (4.1)$$

Birleştirme fonksiyonunun medyan olduğu, aktivasyon fonksiyonunun tanjant hiperbolik fonksiyon olarak seçildiği durumda yapay sinir hücresinin matematiksel modeli aşağıdaki gibi olur.

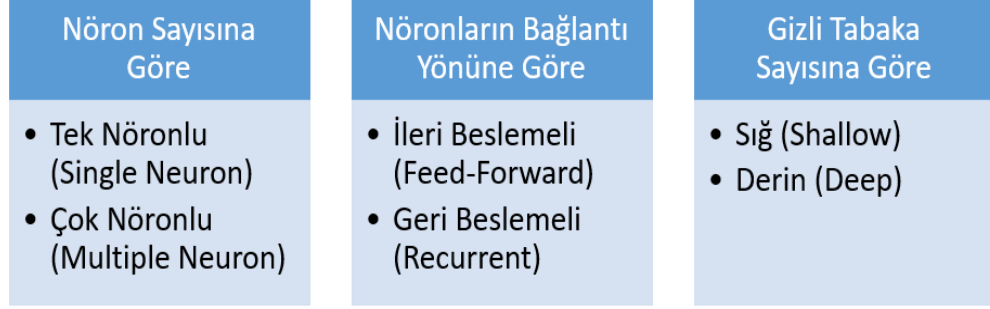
$$y = \tanh(\text{medyan}_i(w_i x_i)) \quad (4.2)$$

Benzer şekilde örnekler artırılabilir. Literatürde sık kullanılan birleştirme fonksiyonlarından biri de çarpımsal birleştirme fonksiyonudur. Birleştirme fonksiyonunun çarpımsal olduğu durumda her ağırlık değerine karşılık bir de yan değeri tanımlanmaktadır. Birleştirme fonksiyonunun çarpımsal olduğu, aktivasyon fonksiyonunun sigmoid fonksiyon olarak seçildiği durumda yapay sinir hücresinin matematiksel modeli aşağıdaki gibi olur.

$$y = \frac{1}{1 + \exp(-\prod_{i=1}^p (w_i x_i + b_i))} \quad (4.3)$$

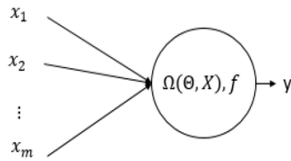
## 5. YAPAY SİNİR AĞI TÜRLERİ

Yapay sinir hücrelerinin birleştirilmesi ile yapay sinir ağları elde edilmektedir. Yapay sinir hücrelerinin birleştirilmesi ile oluşturulan yapılara mimari yapılar adı verilmektedir. Mimari yapılar göre yapay sinir ağlarının farklı sınıflandırmaları mevcuttur. Şekil 5.1’de bu sınıflandırmalar verilmiştir.

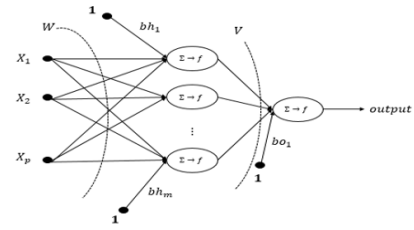


Şekil 5.1 Mimari yapılar göre yapay sinir ağlarının sınıflandırılması

Tüm nöronların bağlantı yönleri ileriye doğru olduğunda yapay sinir ağı ileri beslemeli yapay sinir ağını almaktadır. Gizli tabaka sayısının birden fazla olduğu durumlarda ise yapay sinir ağları derin sinir ağları adını almaktadır. Şekil 5.2, 5.3 ve 5.4’te sırasıyla, nöron sayısı, nöronların bağlantı yönü ve gizli tabaka sayısına göre yapay sinir ağları örnekleri verilmiştir.



Çarpımsal Nöron Model YSA (Multiplicative Neuron Model ANN)

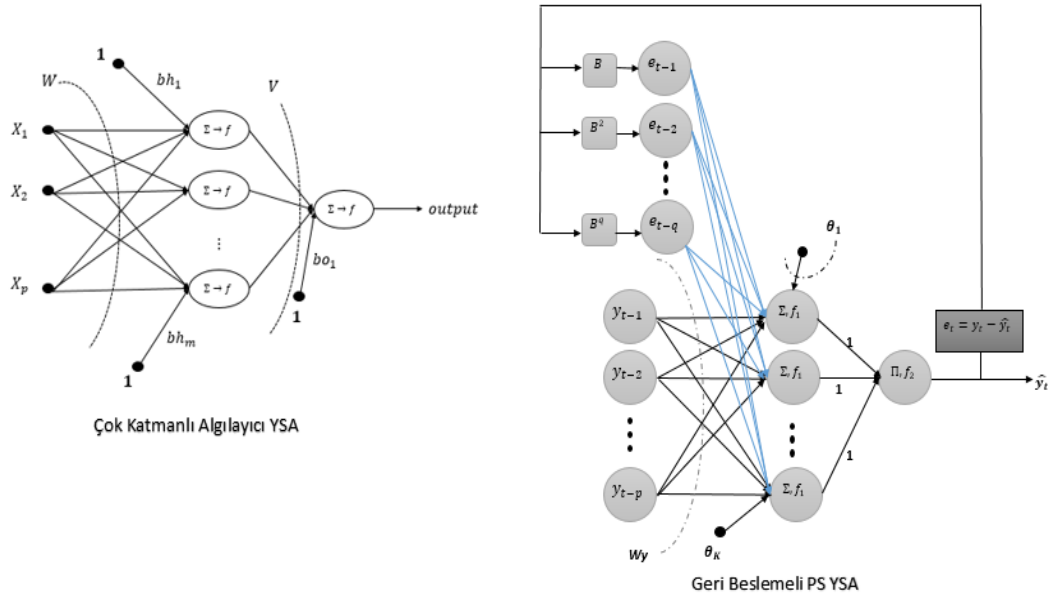


Çok Katmanlı Algılayıcı YSA (Multilayer Perceptron ANN)

Tek Nöronlu YSA’lar

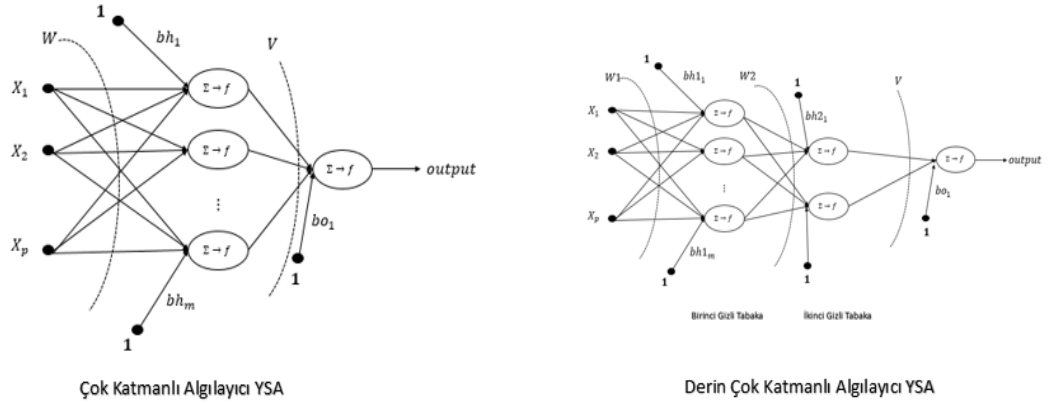
Çok Nöronlu YSA’lar

Şekil 5.2 Nöron Sayısına Göre yapay sinir ağı örnekleri



İleri Beslemeli YSA      Geri Beslemeli YSA

Şekil 5.3 Nöronların bağlantı yönlerine göre yapay sinir ağı örnekleri



Sığ YSA      Derin YSA

Şekil 5.4 Gizli tabaka sayılarına göre yapay sinir ağı örnekleri

Yapay sinir ağları sınıflama, örüntü tanıma, kümeleme, tahmin ve öngörü gibi alanlarda kullanılabilir. Öngörü alanında yapay sinir ağlarının kullanımı son yıllarda oldukça popüler olmuştur. Yapay sinir ağları ile öngörü probleminin çözümünde birçok farklı yapay sinir ağı türleri kullanılmakla birlikte çok katmanlı algılayıcı en popüler yöntemlerden biridir. Bunun dışında son bir iki yıl içinde uzun kısa dönem hafıza derin sinir ağları da etkin şekilde kullanılmaya başlanmıştır.

Zaman serilerinin yapay sinir ağı ile çözümlenmesinde girdilerin farklı seçimleri mevcuttur. Girdiler;

- Öngörülecek zaman serisinin gecikmeli değişkenleri
- Yardımcı zaman serilerinin gecikmeli değişkenleri
- Zamanı gösteren gün ay yıl gibi deterministik bileşenler
- Geri beslemeler

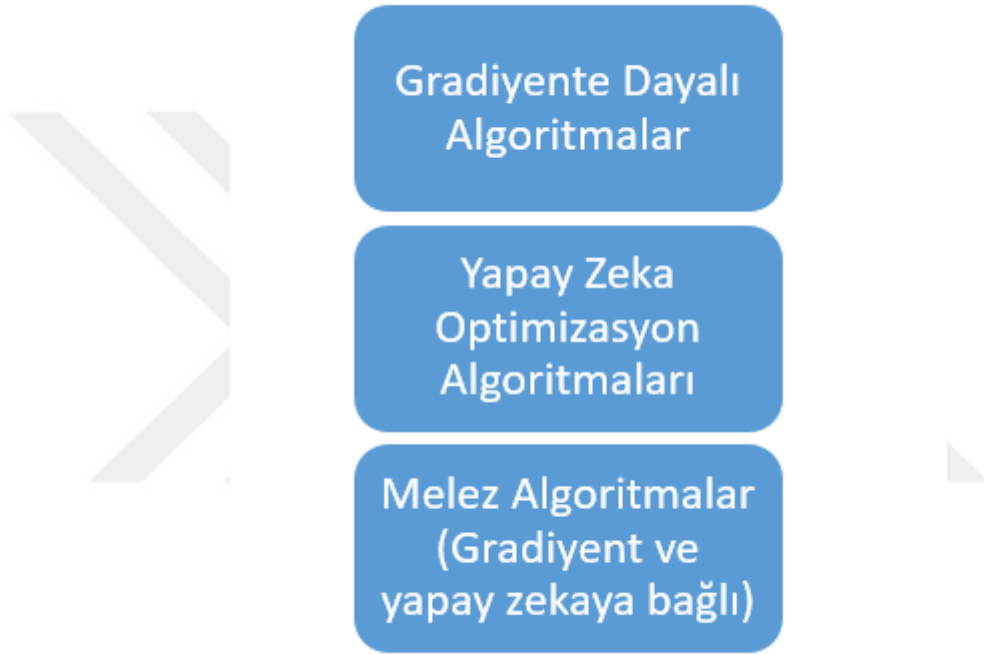
olabilir. Hedef değerler ise genelde öngörüler olmaktadır. Girdiler olarak gecikmeli değişkenler seçildiğinde hangi gecikmeli değişkenlerin girdi olarak alınacağına karar verilmelidir. Zaman serilerinin öngörülmesinde yapay sinir ağını kullanılması durumunda aşağıdaki belirlemelerin yapılması gerekmektedir.

- Girdilerin belirlenmesi: Seçilen girdiler bu problemde şekil değiştirmektedir. Gecikmeli değişkenler girdiler ise gecikmeli değişken seçimi probleminin çözülmesi gerekir.
- Gizli tabaka ve gizli tabakalardaki birim sayılarının belirlenmesi
- Aktivasyon fonksiyonunun seçimi
- Öğrenme algoritması ve algoritmanın parametrelerinin seçimi
- Veri parçalaması yönteminin seçimi
- Verilerin ön işlenmesinde gerekli yöntemlerin seçimi
- Olasılıksal çıkarımlar ve hipotez testleri için yeniden örnekleme yöntemlerinin tercih edilmesi

Belirlemelerin sayısını daha da artırmak mümkündür. Zaman serilerinin yapay sinir ağı ile çözümlenmesinde bu kadar çok belirlemenin yapılması, aslında ciddi bir belirsizliğe neden olmaktadır. Bu nedenle otomatik belirlemelerin yapıldığı yöntemler üzerine literatürde çalışmalar devam etmektedir.

## 6. EĞİTİM ALGORİTMALARI

Yapay sinir ağının eğitimi, öğrenme örneklerindeki hedeflenen çıktıları girdilerin doğrusal olmayan kombinasyonları olarak üretecek ağırlık değerlerinin belirlenmesi işlemidir. Yapay sinir ağlarının eğitiminde optimizasyon algoritmaları kullanılmaktadır. Bu algoritmalar aşağıda verilen Şekil 6.1'deki gibi sınıflandırılabilir.



Şekil 6.1 Yapay Sinir Ağları Eğitim Algoritmaları Sınıflandırması

Gradyente dayalı algoritmalar yapay sinir ağlarına özgü algoritmalar olup, en bilineni geri yayılım öğrenme algoritmasıdır. Bu algortmada her bir ağırlık gradyent vektörünün ters işaretine göre değiştirilir. Optimizasyondaki adım uzunluğu, geri yayılım öğrenme algoritmasında öğrenme parametresi olarak isimlendirilmektedir. Geri yayılım öğrenme algoritması yakınsama hızı düşük bir algoritma olduğundan, literatürde çeşitli modifikasyonları önerilmiştir. Bunlar öğrenme parametresinin adaptasyonu ve momentum teriminin eklenmesi işlemleri ile ortaya koyulan öğrenme algoritmalarıdır. Yakınsama hızının artırılması için kullanılan diğer algoritmalar ise Gradyent yanında Hessian matrisini de kullanan ikinci türeveye dayalı Newton

yöntemi benzeri algoritmalarıdır. En sık kullanılan bu tür algoritma ise Levenberg Marquardt algoritmasıdır.

Yapay zeka optimizasyon algoritmalarından parçacık sürü optimizasyonu, genetik algoritma, yapay arı kolonisi algoritması gibi algoritmalar literatürde yapay sinir ağlarının eğitim algoritması olarak uyarlanarak kullanılmışlardır. Yapay zeka optimizasyon algoritmalarının öğrenme algoritması olarak kullanılması durumunda aşağıdaki iki avantaj söz konusudur.

- Türevlere ihtiyaç duyulmamaktadır.
- Yerel optimum tuzaklarına yakalanma olasılığı azalmaktadır.

Melez algoritmalar ise yerel optimum problemine takılma olasılığı az olan ve araştırılan arama bölgesinde minimum noktaya ulaşmayı garantileyen algoritmalarıdır. Genel olarak yapay sinir ağının ağırlıklarını ( $w$ ) ile gösterirsek eğitim algoritması aşağıdaki gibi bir hata fonksiyonunun ağırlıklara göre minimizasyonudur.

$$E = \sum_{i=1}^n (y_i - hedef_i)^2 \quad (6.1)$$

Burada  $n$  öğrenme örneklerinin sayısını,  $y_i$  ağın  $i$ 'nci çıktısını ve  $hedef_i$  ise  $i$ 'nci hedef değeri göstermektedir. Eğitim algoritması aşağıdaki minimizasyon problemine indirgenir.

$$\min_w E \{ w | x_1, x_2, \dots, x_n, hedef_1, hedef_2, \dots, hedef_n \} \quad (6.2)$$

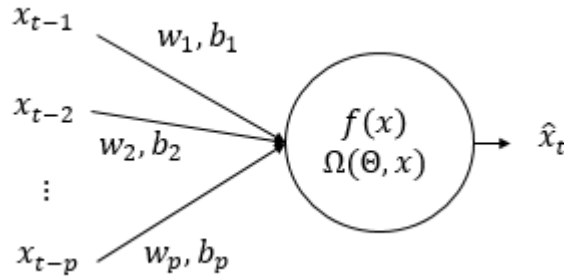
Optimizasyon algoritmalarında her bir iterasyonda ağırlıklar aşağıdaki gibi hesaplanan miktar ile değiştirilir.

$$w^{(t+1)} = w^{(t)} + \Delta w \quad (6.3)$$

$\Delta w$  değişim miktarı Gradyente dayalı algoritmalarda, gradiyentin bir fonksiyonu olup, yapay zeka optimizasyon algoritmalarında akıllı stratejiler ile belirlenmektedirler.

## 7. ÇARPIMSAL NÖRON MODEL YAPAY SİNİR AĞLARI

Çarpımsal nöron model yapay sinir ağı (ÇNM-YSA) ilk kez Yadav ve ark. (2007) çalışmasında ortaya koyulmuştur. ÇNM-YSA mimari yapısında sadece bir adet nöron bulundurmaktadır. Diğer yapay sinir ağı türleri ile karşılaştırıldığında daha az parametre ile etkili sonuçlar üretebilen bir yapay sinir ağıdır. ÇNM-YSA tek hücre birimin nöron modeli olarak toplamsal ya da Mc-Culloch&Pitts nöron model yerine birleştirme fonksiyonunun çarpma işlemine dayalı olduğu çarpımsal nöron modeli kullanılmaktadır. ÇNM-YSA sınıflama, örüntü tanıma, tahmin ya da öngörü probleminin çözümü için literatürde başarılı sonuçlar üretmiştir. Genel olarak, öngörü probleminin çözümünde ÇNM-YSA'nın girdileri gecikmeli değişkenler olarak alınmaktadır. ÇNM-YSA'nın mimarisi Şekil 7.1'de verilmiştir.



Şekil 7.1 ÇNM-YSA'nın öngörü problemi için çözüm mimarisi

Şekilden görüldüğü gibi, ÇNM-YSA'nın girdileri  $p$  adet gecikmeli değişkendir. Ağırlıklar  $W = (w_1, w_2, \dots, w_p)$  vektörü ve yan değerleri  $B' = (b_1, b_2, \dots, b_p)$  vektörü ile gösterilebilir. Ağın hedef değerleri ise  $n$  gözlemlili bir zaman serisi için  $x_t, t = p + 1, p + 2, \dots, n$  olmaktadır. ÇNM-YSA'nın çıktıları öğrenme örnekleri üzerinden hesaplanır. Girdiler ve hedef değerler öğrenme örneklerini oluşturur. Girdi matrisi ( $X$ ), hedef değer matrisi ( $Y$ ) ve öğrenme örnekleri matrisi ( $LS$ ) olmak üzere aşağıdaki eşitliklerde verilmiştir.

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_p \\ x_2 & x_3 & \cdots & x_{p+1} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n-p} & x_{n-p+1} & \cdots & x_{n-1} \end{bmatrix} \quad (7.1)$$

$$Y' = [x_{p+1} \quad x_{p+2} \quad \cdots \quad x_n] \quad (7.2)$$

$$LS = \begin{bmatrix} x_1 & x_2 & \cdots & x_p & x_{p+1} \\ x_2 & x_3 & \cdots & x_{p+1} & x_{p+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n-p} & x_{n-p+1} & \cdots & x_{n-1} & x_n \end{bmatrix} \quad (7.3)$$

LS matrisinin satırları öğrenme örnekleridir. ÇNM-YSA'nın çıktıları aşağıdaki formüller ile hesaplanır.

$$WX = [wx_1 \quad wx_2 \quad \cdots \quad wx_p] = W \otimes X \quad (7.4)$$

$$NET = \sum_{i=1}^p wx_i + B \quad (7.5)$$

Burada  $\otimes$  elemanter çarpımı göstermekte olup NET ise  $n - p$  bir vektör olmaktadır. Çıktı vektörü ise  $Output' = (\hat{x}_{p+1} \quad \hat{x}_{p+2} \quad \cdots \quad \hat{x}_n)$  olup aşağıdaki gibi hesaplanmaktadır.

$$Output = \frac{1}{1 + \exp(-NET)} \quad (7.6)$$

## 8. ÇARPIMSAL NÖRON MODEL YAPAY SİNİR AĞLARININ EĞİTİMİ

Çarpımsal nöron model yapay sinir ağlarının eğitiminde geri yayılım öğrenme algoritması, parçacık sürü optimizasyonu, diferansiyel gelişim algoritması ve yapay arı kolonisi algoritmaları literatürde kullanılmaktadır. Çarpımsal nöron model yapay sinir ağının geri yayılım öğrenme algoritması ile eğitimi ilk kez Yadav ve ark. (2007)'de kullanılmıştır. Zhao ve Yang (2009) parçacık sürü optimizasyonu ile çarpımsal nöron modeli eğitmişlerdir. Yeh (2013) çarpımsal nöron modelin eğitiminde de kullanılan yeni bir sürü optimizasyonu yöntemi önermiştir. Cui ve ark. (2015) ateş böceği ve diferansiyel gelişim algoritmasının melez bir yaklaşımını çarpımsal nöron modelin eğitimi için önermişlerdir. Samantha (2015) Gradyente dayalı online bir eğitim algoritmasını çarpımsal nöron model ile doğrusal olmayan dinamik bir sistemin kontrolü için önermişlerdir. Bas (2016) çarpımsal nöron model yapay sinir ağının eğitimi için diferansiyel gelişim algoritmasına dayalı bir eğitim algoritması önermiştir. Bas ve ark. (2016b) çarpımsal nöron modelin eğitimi için yarasa algoritmasına dayalı bir eğitim algoritması kullanmışlardır. Bas ve ark. (2016b) çarpımsal nöron model yapay sinir ağının eğitimi için M tahmin edicisine dayalı sağlam bir öğrenme algoritması önermiştir.

Bu çalışmada, geri yayılım öğrenme algoritması, parçacık sürü optimizasyonu ve yapay arı kolonisi algoritmaları ile çarpımsal nöron model yapay sinir ağının eğitimin nasıl gerçekleştirileceği algoritmalar ve akış diyagramları ile detaylı bir şekilde verilmiştir.

### 8.1. Geri Yayılım Algoritması Ile Çarpımsal Nöron Model Yapay Sinir Ağı Eğitimi

Geri yayılım öğrenme algoritması belirlenen karesel hata fonksiyonunun türevlerine dayalı olarak çalışmaktadır. Hata fonksiyonu (8.1) eşitliğinde verildiği gibi tüm öğrenme örneklerinden hesaplanan hata kareler ortalaması değeridir.

$$E = \frac{1}{2n} \sum_{i=1}^n (\hat{x}_i - d_i)^2 \quad (8.1)$$

Burada  $d_i$  ve  $\hat{x}_i$ , sırasıyla  $i$ . öğrenme örneği için hedef değer ve ağıın çıktısı olmaktadır. Ağıın eğitimi için hata fonksiyonunu hem ağırlıklar hem de yanlara göre türevlerine ihtiyaç vardır ve bu türevler aşağıdaki gibi kolaylıkla elde edilebilir.

$$\frac{\partial E}{\partial w_k} = \sum_{i=1}^n y_i (y_i - d_i) (1 - y_i) \frac{\prod_{j=1}^m (w_j x_{kj} + b_j)}{(w_k x_k + b_k)} x_k \quad (8.2)$$

$$\frac{\partial E}{\partial b_k} = \sum_{i=1}^n y_i (y_i - d_i) (1 - y_i) \frac{\prod_{j=1}^m (w_j x_{kj} + b_j)}{(w_k x_k + b_k)} \quad (8.3)$$

Ağırlıkların ve yanların değişimi hata fonksiyonuna göre elde edilen türevlerin  $\lambda$  katı ve ters işaretlisi büyüklüğünde gerçekleştirilir. Burada  $\lambda$  öğrenme parametresi olup tüm ağırlık ve yanlar için eşit olarak alınmaktadır. Buna göre ağırlık ve yan değişimleri (8.4) ve (8.5) eşitlikleri ile gerçekleştirilir.

$$\Delta w_k = -\lambda \frac{\partial E}{\partial w_k} \quad (8.4)$$

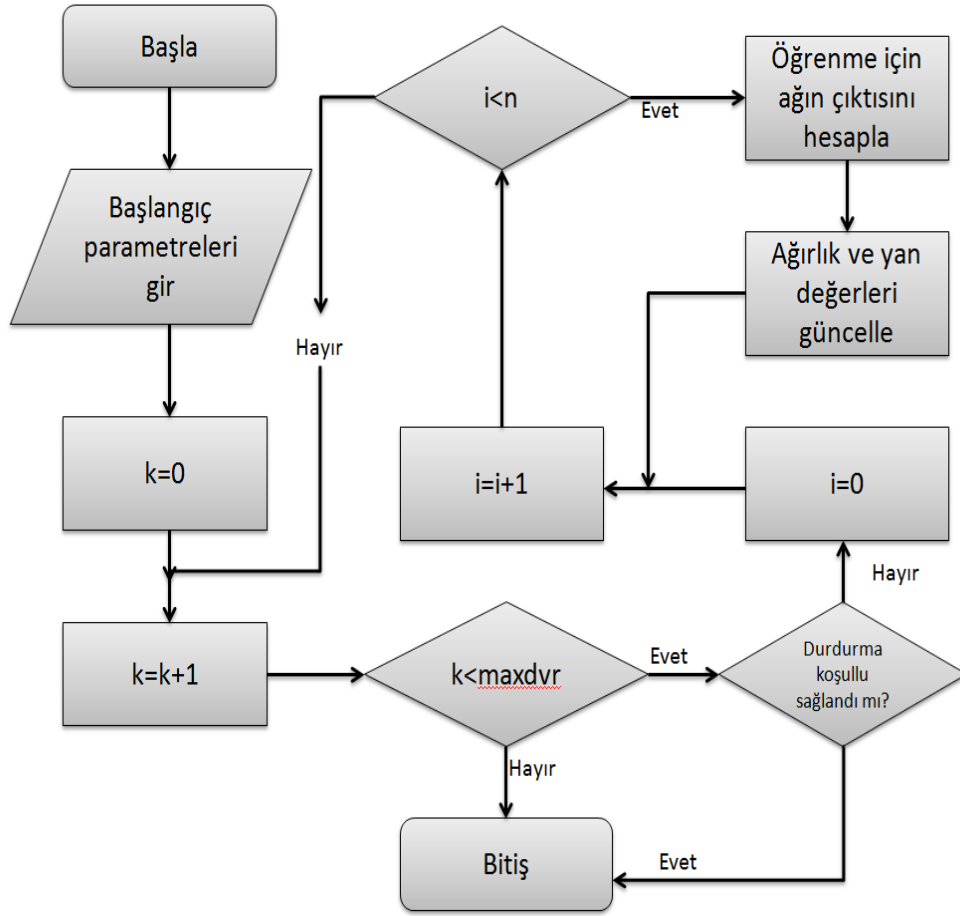
$$\Delta b_k = -\lambda \frac{\partial E}{\partial b_k} \quad (8.5)$$

Geri yayılım öğrenme algoritmasında her bir iterasyon sonunda güncel ağırlıklar ise (8.6) ve (8.7) formülleri ile hesaplanır.

$$w_k = w_k + \Delta w_k \quad (8.6)$$

$$b_k = b_k + \Delta b_k \quad (8.7)$$

Geri yayılım öğrenme algoritmasının akış diyagramı Şekil 8.1'de verilmiştir. Bu akış diyagramı standart geri yayılım öğrenme algoritmasına göre tasarlanmıştır.



Şekil 8.1 Geri yayılım öğrenme algoritmasının akış diyagramı (Egrioglu vd., 2019)

## 8.2. Parçacık Sürü Optimizasyonu Algoritması Ile Çarpımsal Nöron Model Yapay Sinir Ağı Eğitimi

Parçacık sürü optimizasyonu genellikle kuş ve balık sürülerinin davranış ve hareketlerini gözlemleyen bir algoritma yapısıdır. Kennedy ve Eberhart (1995)'de parçacık sürü optimizasyonu (PSO) önermiştir. PSO en çok kullanılan kuş sürülerinin davranışlarını, çevreye adapte olmalarını zengin yiyecek kaynaklarını bulma ve tuzaklardan kaçabilme gibi davranışlarını baz almıştır. Parçacık sürü optimizasyonda parçacıklar en iyi sonucu takip etme eğilimi göstermektedir. Bundan dolayı tüm parçacıklar kümeler halinde en iyi sonucu bulmaya çalışmaktadır. Kullanımı basit ve yüksek çözüm kümesi nedeniyle yaygın uygulama alanı bulmaktadır.

Çarpımsal nöron model yapay sinir ağlarının eğitiminde PSO yoğun bir şekilde kullanılmaktadır. Parçacık sürü optimizasyonda çözümü bulmak adına her bir pozisyona parçacık denir. Parçacığın pozisyonu rastgele olarak üretilir ve aşağıda gösterilen  $P_k$  vektöründe saklanır.

$$P_k = \{p_{k,1}, p_{k,2}, \dots, p_{k,d}\}, \quad k = 1, 2, \dots, pn \quad (8.8)$$

Burada  $p_{k,i}$  ( $i = 1, 2, \dots, d$ ) sırasıyla  $k$  parçacığının  $i$ . pozisyonunu,  $pn$  parçacık  $d$  parçacıkların pozisyonlarını gösterir. Parçacıklar çarpımsal nöron modelin ağırlıkları ve yan değeri olacaktır.

Bir parçacığın pozisyon sayısı,  $\Theta = (w_1, w_2, \dots, w_m, b_1, b_2, \dots, b_m)$  vektörünün eleman sayısı kadar yani  $2m$ 'dir. Her bir parçacığın birinci pozisyon değeri  $w_1$  ağırlığının sayısal değeri, ikinci pozisyonu  $w_2$  ağırlığının sayısal değerine ve diğerleri de benzer şekilde ağırlık ve yanların sayısal değerlerine karşılık gelir.

Her bir parçacığın pozisyonu için rastgele bir hız değeri belirlenir ve  $V_k$  vektöründe saklanır.

$$V_k = \{v_{k,1}, \dots, v_{k,d}\}, \quad k = 1, 2, \dots, pn \quad (8.9)$$

Her bir parçacık için amaç fonksiyonu değeri hesaplanarak bu değerlere dayalı olarak  $Pbest$  ve  $Gbest$  belirlenir.

$$Pbest_k = \{pb_{k,1}, \dots, pb_{k,d}\}, \quad k = 1, 2, \dots, pn \quad (8.10)$$

$$Gbest = \{P_{g,1}, \dots, P_{g,d}\} \quad (8.11)$$

Her parçacık kendi konum ve hızlarına sahiptir. Parçacıklar en iyi konumlarını  $Pbest$  ve  $Gbest$  iki konumda muhafaza eder.  $Pbest_k$   $k$ . parçacığın

geçerli iterasyona kadar bulunduğu en iyi pozisyonlarının vektörü ve  $Gbest$  ise sürünün geçerli iterasyona kadar bulunduğu en iyi pozisyonunun vektörüdür. Bu adımda amaç fonksiyonu olarak hedef değerler ve ağırlık çıktıları arasında hesaplanan bir hata kareler ortalaması kullanılabilir.

Hız güncellenme denklemlerinde hızlanma kat sayısı olan parametreler  $c_1$  ve  $c_2$ ; sırasıyla bilişsel ve sosyal katsayılarıdır.  $c_1$  bilişsel ivme kat sayısı parçacıkların kendi hafızalarına göre arama alanındaki hareketini,  $c_2$  ise sürünün hafızası ile  $Gbest$ 'e yaklaşma hareketini kontrol eder.  $w$  ise eylemsizlik ağırlığı parametresi olsun.  $(c_{1i}, c_{1f})$ ,  $(c_{2i}, c_{2f})$ , ve  $(w_1, w_2)$  değerleri parametreler için başlangıç ve bitiş değerleri olsun. Her bir iterasyonda bu parametreler (8.12), (8.13) ve (8.14) eşitlikleri ile hesaplanır.

$$c_1(t) = (c_{1f} - c_{1i}) \frac{t}{maxt} + c_{1i} \quad (8.12)$$

$$c_2(t) = (c_{2f} - c_{2i}) \frac{t}{maxt} + c_{2i} \quad (8.13)$$

$$w(t) = (w_2 - w_1) \frac{maxt-t}{maxt} + w_1 \quad (8.14)$$

Burada  $maxt$  ve  $t$  sırasıyla maksimum iterasyon sayısı ve geçerli iterasyon sayısını gösterir. Hız ve pozisyon değerleri (8.15) ve (8.16) eşitlikleri ile güncellenir.

$$v_{i,j}^{t+1} = [w(t) \times v_{i,j}^t + c_1(t) \times rand_1 \times (pb_{i,j} - p_{i,j}) + c_2(t) \times rand_2 \times (p_{g,j} - p_{i,j})] \quad (8.15)$$

$$p_{i,j}^{t+1} = p_{i,j} + v_{i,j}^{t+1} \quad (8.16)$$

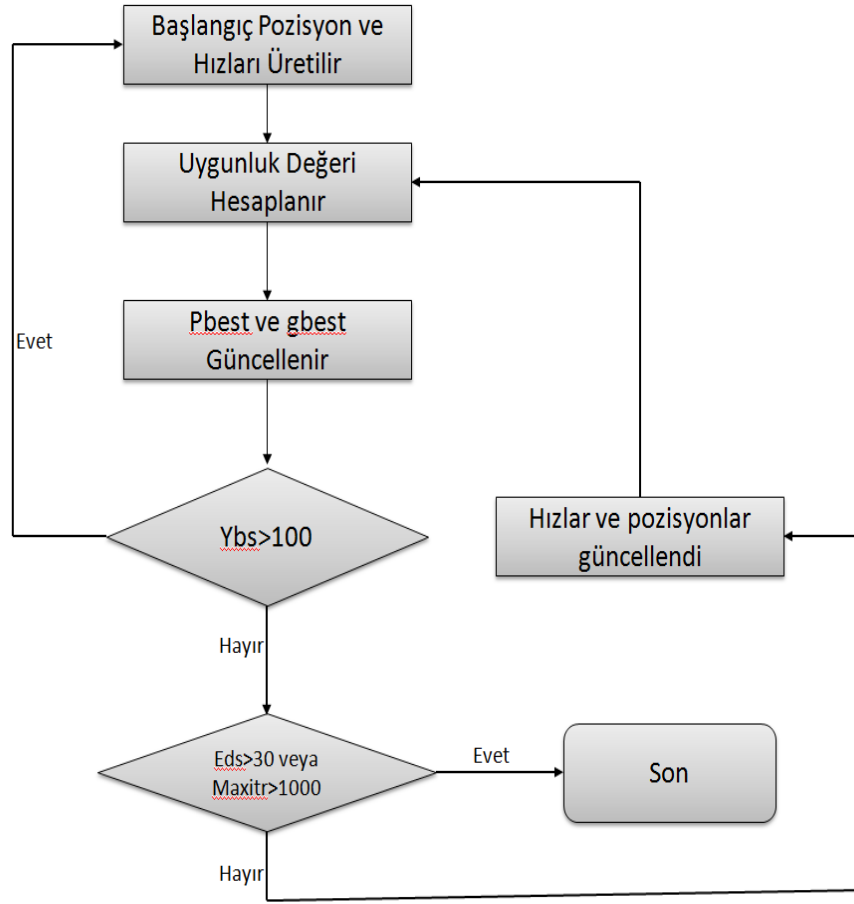
$rand_1$  ve  $rand_2$   $[0, 1]$  parametrelili düzgün dağılımdan üretilen rastgele sayılardır. Algoritma (8.10)- (8.16) formülleri arasında önceden belirlenen maksimum iterasyon sayısına kadar çalıştırılır.

Parçacık sürü optimizasyonu ile çarpımsal nöron model yapay sinir ağının eğitiminde erken durdurma koşulu ve yeniden başlama stratejisi ağın ezberlemeden eğitiminin tamamlanmasına yardımcı olmaktadır.

Erken durdurma koşulu, parçacık sürü optimizasyonunun ardışık iterasyonlarında hata düzeyinde iyileşme sağlanıp sağlanmamasına bağlı olarak oluşturulur. Erken durdurma için erken durdurma sayacı “Eds” algoritma içinde tutulur. Ardışık iterasyonlarda hata seviyesinde iyileşme sağlandı ise sayaç sıfırlanır. Aksi halde başlangıçta sıfır olan sayacın değeri bir artırılır. Her iterasyon sonunda sayacın değeri kontrol edilir. Eğer sayaç değeri 30’dan büyük ise parçacık sürü optimizasyonu durdurulur. Böylece ağın ezberlemesinin önüne geçilmiş olur. Algoritmanın durdurulması için bir diğer koşul ise iterasyon sayacı “Itr” değerinin maksimum iterasyon sayısının 1000 değerini geçmesidir.

Yeniden başlama stratejisi ise parçacık sürü optimizasyonuna arama uzayında farklı noktaların keşfedilmesi ve yerel optimum tuzağından kurtulma şansı tanır. Bu stratejide her 100 iterasyonda bir parçacıkların tüm pozisyon ve hız değerleri rastgele üretilirken pbest ve gbest ise değiştirilmez ve sabit bırakılır. Bu strateji için algoritmada yeniden başlama sayacı “Ybs” oluşturulur ve her iterasyonda bu sayaç artırılır. Sayaç değeri 100 geçtiğinde ise sıfırlanır.

Parçacık sürü optimizasyonu algoritmasının akış diyagramı Şekil 8.2’da verilmiştir. Bu akış diyagramı standart parçacık sürü optimizasyonu algoritmasına göre tasarlanmıştır.



Şekil 8.2 Parçacık sürü optimizasyonu algoritmasının akış diyagramı (Egriöğlü vd., 2019)

### 8.3. Yağay Arı kolonisi Optimizasyonu Algoritması İle Çarpımsal Nöron Model Yapay Sinir Ağı Eğitimi

Çarpımsal nöron model yapay sinir ağının eğitiminde kullanılabilecek bir diğer yöntem ise yapay arı kolonisi algoritmasıdır. Yapay arı kolonisi algoritmasının uygulanmasında kaynakların pozisyonları, çarpımsal nöron modelin ağırlıkları ve yan değeri olacaktır. Uygunluk fonksiyonu ise yine eğitim kümesi için hesaplanan hata kareler ortalaması değeridir.

Yapay arı kolonisi algoritması (YAK) Karaboğa (2005) tarafından önerilmiş ve arıların yiyecek aramalarını esas alan bir yapay zeka optimizasyon algoritmasıdır. YAK algoritması bal arısı sürülerinin karakterlerini, davranışlarını ve besin ararken kullandıkları yöntemlerden oluşturulmuştur. Arıların besin bulmada göstermiş olduğu hareketler optimizasyon problemlerinin çözümünde kullanılmıştır. Sürü

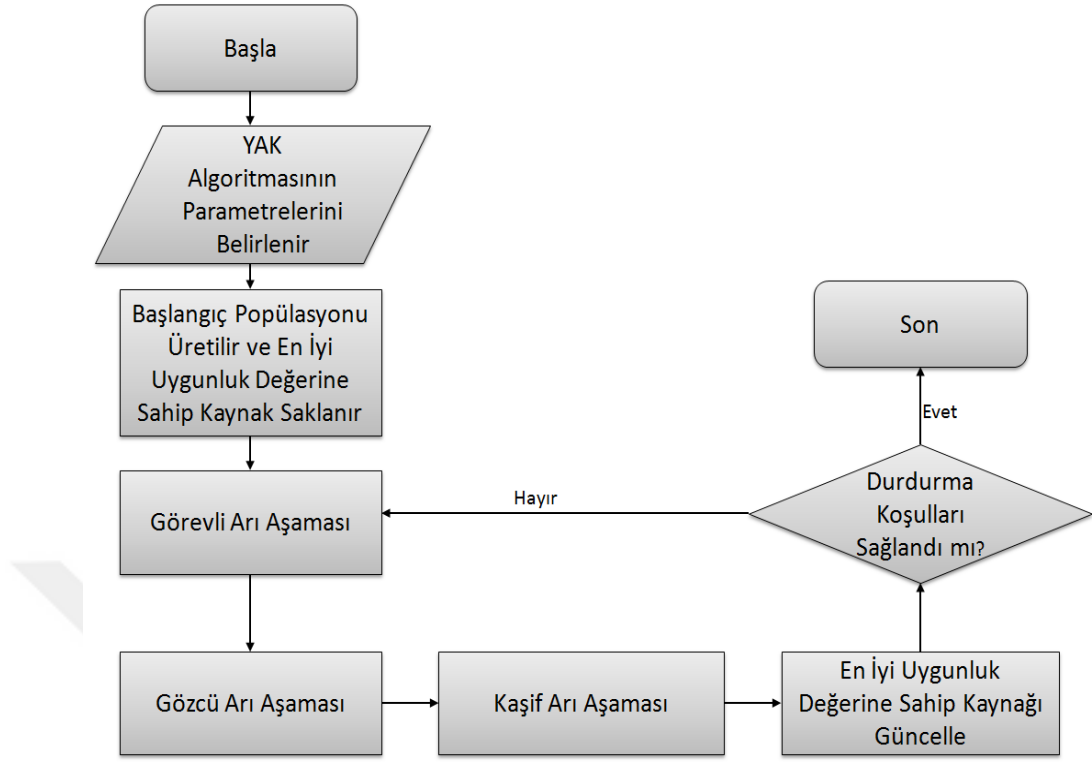
zekâsı içinde yaşayan arılar içgüdüsel olarak bir birleriyle uyum ve kurallar çerçevesinde hareket etmekte olup, her arı belirli bir görevi yürütmektedir. Balın getirilmesi, yiyeceklerin depolanması, iletişim ve besin arama gibi işler belirli sosyal bir düzende yürütülmektedir. YAK algoritmasında görevli arı, gözcü arı ve kâşif arı olmak üzere üç farklı arı türü görev almaktadır. YAK algoritmasında, her bir besin kaynağı arama alanındaki bir aday çözüm olarak tanımlanır ve başlangıçta besin kaynağı sayısı, kovadaki görevli arı sayısına eşittir. Her bir besin kaynağının kalitesi, ilgili kaynağın amaç fonksiyonu (uygunluk) değeri ile belirlenir.

Görevli arılar, keşfedilen kaynaklardan kovana besin taşımaktan sorumludurlar. Ayrıca kaynağın yerini ve kalitesini kovadaki diğer arılarla paylaşırlar. Görevli arılar buldukları kaynakların komşu pozisyonları yardımı ile yeni kaynaklar bulurlar ve iki kaynak arasında en iyi kaynağı hafızasında tutar.

Gözcü arılar yuvada bekler ve görevli arılardan aldığı bilgiler ile yeni besin kaynaklarına yönelir. Yeni kaynaklardan elde edilen bilgilerin uygunluk değeri hesaplanır ve diğer gözcü arılarla paylaşılır. Gözcü arılarda her bir uygunluk değerini kullanarak en iyi olasılıklı kaynakları seçerler.

Kâşif arılar belirli denemelerden sonra tükenen kaynakları popülasyondan çıkarırlar ve rastgele olarak yeniden kaynaklara yönelirler. Yeni üretilen kaynaklar popülasyona eklenir.

Yapay arı kolonisi algoritmasının akış diyagramı Şekil 8.3’de verilmiştir. Bu akış diyagramı standart yapay arı kolonisi algoritmasına göre tasarlanmıştır.



Şekil 8.3 Yapay arı kolonisi Algoritmasının Akış Diyagramı (Egrioğlu vd., 2019)

YAK algoritmasının aşağıda verilen parametreleri belirlenir. Bu parametreler:

Kaynak Sayısı ( $SN$ ): Bu değer 30 ile 50 arasında bir değer olarak seçilebilir.

Problemin boyutu ( $D$ ): Yapay sinir ağının girdi sayısına göre belirlenir. Boyut girdi sayısının iki katıdır.

Gözcü Arı Sayısı ( $NOB$ ): 30 ile 50 arasında değişen bir sayı olarak seçilir.

Limit değeri ( $LIMIT$ ): Bu parametre tükenen kaynakların belirlenmesi için kullanılır. Genel olarak kaynak sayısına göre belirlenmekle birlikte bu çalışmada 200 olarak tercih edilmiştir.

Maksimum iterasyon sayısı ( $MAXITR$ ): 50, 100, 500 veya 1000 olarak seçilebilir.

Erken durdurma için limit değeri ( $ANFS$ ): 5, 6, 10, 30 veya 50 olarak seçilebilir. Eğer peşpeşe  $ANFS$  iterasyon en iyi kaynağa iyileşme sağlanamamışsa algoritma durdurulur.

YAK algoritmasında gerçekleştirilen önemli adımlardan bir tanesi başlangıç kaynaklarının üretilmesidir. Başlangıç kaynak değerleri (0,1) parametrelili düzgün dağılımdan üretilir.

$$x_{ij} \sim \text{Uniform}(0,1), i = 1,2, \dots, SN; j = 1,2, \dots, D \quad (8.17)$$

Her bir kaynak için aşağıdaki gibi başarısızlık indeksi sıfırlanır.

$$failure_i = 0; i = 1,2, \dots, SN \quad (8.18)$$

YAK algoritmasında her bir kaynak için uygunluk fonksiyonu değerleri ( $f_i; i = 1,2, \dots, SN$ ) hesaplanarak en iyi kaynak hafızaya alınır. Uygunluk fonksiyonu olarak hata kareler ortalaması kullanılmıştır.

$$HKO = \frac{1}{ntrain-p} \sum_{i=p+1}^{ntrain} (X_t - \hat{X}_t)^2 \quad (8.19)$$

İşçi arı aşamasında her bir kaynağa bir işçi arı görevlendirilir. Her görevli olunan kaynak için bir komşu kaynak rastgele seçilir.  $i1$  ve  $i2$  sırasıyla bu kaynakların indisleri olsun. Aşağıdaki işlemler her bir görevli arı için uygulanır.

- Rastgele bir  $i3$  pozisyonu seç.
- $(-1,1)$  aralığından  $\emptyset$  sayısını rastgele üret
- Aşağıdaki formül ile yeni pozisyon değeri bul.

$$x_{i1,i3}^{new} = x_{i1,i3} + \emptyset(x_{i1,i3} - x_{i2,i3}) \quad (8.20)$$

Yeni pozisyon değerini kullanarak yeni bir kaynak üretilir. Yeni kaynağın uygunluk değeri ( $f_{new}$ ) hesaplanır. Görevli olunan  $i1$  kaynağının uygunluk değeri ise ( $f_{i1}$ ) olmak üzere aşağıdaki işlemler yapılır.

Eğer  $f_{new} \leq f_{i1}$  yeni kaynak kabul edilir ve başarısızlık indeksi sıfırlanır  $failure_{i1} = 0$ . Aksi halde yeni kaynak red edilir ve başarısızlık indeksi artırılır  $failure_{i1} = failure_{i1} + 1$ .

Gözcü arı aşamasında gözcü arılar uygunluk değerlerine göre hesaplanan olasılıklara göre kaynaklara görevlendirilir.

$$P_i = \frac{1/f_i}{\sum_{t=1}^{SN} 1/f_t} \quad (8.21)$$

Her bir gözcü arı için aşağıdaki işlemler gerçekleştirilir.

- $i1$  kaynağı olasılıklara göre rastgele seçilir.
- $i2$  komşu kaynağı seçilir.
- Rastgele bir  $i3$  pozisyonu seçilir.
- $(-1,1)$  aralığından  $\emptyset$  sayısını rastgele üretilir.

- (8.21) formülü ile yeni pozisyon değeri bulunur.

Yeni pozisyon değerini kullanarak yeni bir kaynak üretilir. Yeni kaynağın uygunluk değeri ( $f_{new}$ ) hesaplanır. Görevli olunan  $i_1$  kaynağının uygunluk değeri ise ( $f_{i_1}$ ) olmak üzere aşağıdaki işlemler yapılır.

Eğer  $f_{new} \leq f_{i_1}$  yeni kaynak kabul edilir ve başarısızlık indeksi sıfırlanır ( $failure_{i_1} = 0$ ). Aksi halde yeni kaynak red edilir ve başarısızlık indeksi artırılır ( $failure_{i_1} = failure_{i_1} + 1$ .)

YAK algoritmasında her bir iterasyonda en iyi kaynak güncellenir. Eğer en iyi kaynak değışti ise erken durdurma sayacı sıfırlanır ( $NFS = 0$ ) aksi halde sayaç değeri artırılır ( $NFS = NFS + 1$ ).

Kaşif arı aşamasında tüm kaynaklar için bir kaşif arı görevlendirilir. Eğer  $failure_i > LIMIT$  ;  $i = 1, 2, \dots, SN$  ise kaynak atılarak yeni bir kaynak rastgele üretilir.

Her iterasyon sonunda durdurma koşulları kontrol edilir. Eğer  $NFS > ANFS$  veya maksimum iterasyon sayısına ulaşılmış ise algoritma durdurulur. Aksi halde işlemlere devam edilir.

## 9. ÇARPIMSAL NÖRON MODEL YSA İLE ZAMAN SERİSİ ÖNGÖRÜ PROBLEMİ İÇİN BİR ALGORITMA

Zaman serilerinde öngörü elde etmek için yapay sinir ağlarının kullanılması verinin ön işlenmesi ve dönüştürülmesi, verinin parçalanması, yapay sinir ağının eğitimi ve mimari seçimi, öngörülerin elde edilmesi adımlarını kapsamaktadır. Bu tez çalışmasında çarpımsal nöron model yapay sinir ağı ile zaman serisi öngörüsü için aşağıda verilen algoritma tasarlanmıştır.

Zaman serisinin gözlemleri eğitim ve test verisi olarak 2'ye ayrılır. Zaman serisinin sondan %10'luk kısmı test verisi olarak kullanılır. Zaman serisini ilk baştan %90'luk kısmı ise eğitim verisi olarak kullanılır. Eğitim verisi zaman serisinin durağanlaştırma, normalleştirme gibi ön işlemleri yanı sıra, ÇNM-YSA'nın girdilerinin oluşturulması için de kullanılır.

Zaman serisindeki durağan dışılık araştırılır. Bu aşamada zaman serisi olarak eğitim verisi kullanılır. Zaman serisinin, verinin genel karakteristiği ile uyumsuz olan aykırı değerler içermesinin ÇKA'nın öngörü performansını olumsuz etkilediği Hill ve ark. (1994) ve Zhang ve ark. (1998) tarafından ortaya konan çalışmalarda belirtilmiştir. Bununla birlikte, klasik YSA yaklaşımlarının veri setinde aykırı değer bulunduğu durumlarda bazı problemlere sahip olduğu Chuang ve ark. (2002) çalışmalarında gösterilmiştir.

İlk olarak zaman serisindeki mevsimsellik tespit edilir. Bunun için aşağıda verilen koşul kontrol edilir.

$$|ACF_s| > 1.645 \sqrt{\frac{1+2(ACF_1 + \sum_{i=2}^{s-1} ACF_i^2)}{n}} \quad (9.1)$$

Burada  $ACF_s$  eğitim verisi için  $s$  dereceden otokorelasyon katsayısını gösterir. Eğer koşul sağlanır ise zaman serisinin  $m$  periyodu ile mevsimsel olduğu sonucuna varılır. İkinci olarak, zaman serisindeki trend ya da birim kökler belirlenir. Bunun için birim kök testleri uygulanır. Zaman serisinin mevsimsellik içermesi durumunda

mevsimsel fark parametresi  $D = 1$  olarak alınır aksi halde  $D = 0$  alınır. Zaman serisi birim kök içermesi durumunda  $d = 1$  olarak alınır, aksi halde  $d = 0$  alınır.

Zaman serisi eğitim ve test kümesi ayırımı olmaksızın tüm gözlemleri içerecek şekilde belirlenen fark parametrelerine göre aşağıdaki gibi durağan hale getirilir. Orijinal zaman serisi  $x_t$  olmak üzere, durağan zaman serisi  $y_t$  ile gösterilsin.

$$z_t = (1 - B)^d (1 - B^s)^D x_t \quad (9.2)$$

Durağan ( $z_t$ ) eğitimi verisi olan zaman serisinin kısmi otokorelasyon katsayıları hesaplanır. Güven sınırları dışında kalan kısmi otokorelasyonlar belirlenir.

$$PACF_i > \frac{2}{\sqrt{n}} \text{ veya } PACF_i < -\frac{2}{\sqrt{n}}, i=1,2,\dots,n/4 \quad (9.3)$$

(9.3) koşulunu sağlayan gecikme sayıları ( $i$ ) belirlenir.

Durağan zaman serisi aşağıda verilen dönüşüm ile eğitim verisi için hesaplanan minimum ve maksimum değerler kullanılarak normalleştirilir.

$$y_t = \frac{z_t - \min(z_t)}{\max(z_t) - \min(z_t)} \quad (9.4)$$

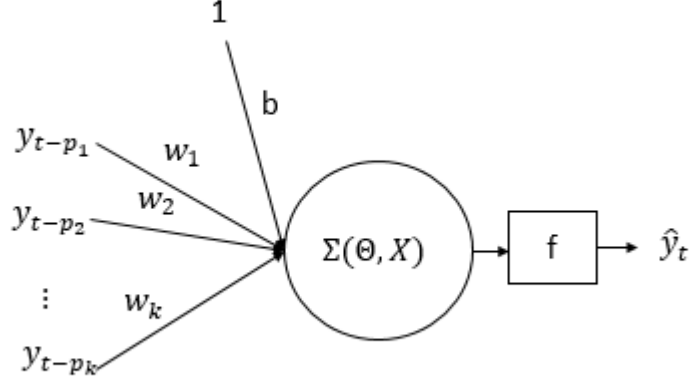
ÇNM-YSA'nın girdileri oluşturulur.

ÇNM-YSA'nın girdileri (9.3) koşulunu sağlayan gecikme sayılarıdır. YSA'nın girdileri olan gecikmeli değişkenler aşağıdaki vektördeki gibi gösterilir.

$$Girdiler = [y_{t-p_1}, y_{t-p_2}, \dots, y_{t-p_k}] \quad (9.5)$$

(9.4)'de durağan zaman serisinin kısmi otokorelasyonlarının  $k(k \geq 1)$  tanesinin (9.3) koşulunu sağladığı varsayılmaktadır. Eğer hiçbir kısmi otokorelasyon katsayısı (9.3) eşitliğini sağlamıyor ise yani  $k = 0$  ise ÇNM-YSA'nın çalışabilmesi eğer seri mevsimsellik içeriyorsa ((9.1) koşulunu sağlıyor ise) için  $k = 1$  ve  $p_1 = s$  olarak,

mevsimsellik içermiyorsa  $k = 1$  ve  $p_1 = 1$  olarak alınır. ÇNM-YSA girdileri ile Şekil 9.1 deki gibi gösterilir.



Şekil 9.1. ÇNM-YSA Mimarisi

Şekil 9.1 de gösterilen ÇNM-YSA eğitim verileri ile eğitilerek optimum ağırlık ve yan değerleri elde edilir.

Optimum ağırlık ve yan değerleri elde edilen ÇNM-YSA ile test kümesi için öngörüler elde edilir. Öngörülerin elde edilmesinde iki strateji uygulanır.

- 1) Bir adım öngörü performansının araştırılması için eğitim kümesi dışında kalan geçmiş değerler gerçek değerler alınarak öngörüler hesaplanır.
- 2) Çok adım öngörü performansının araştırılması için eğitim kümesi dışındaki geçmiş değerler için tahmin değerleri kullanılır.

Elde edilen öngörüler normalleştirme ve durağanlaştırmada kullanılan parametrelere göre orijinal ölçeğine getirilir. Normalleştirme işlemi (9.6) eşitliği ile ters dönüştürülür iken  $d = 0, D = 1$  ve  $s = 12$  için fark alma işlemi (9.7) le geri dönüştürülür. Fark alma parametrelerinin farklı değeri için (9.7) formülü de değiştirilerek uygulanır.

$$\hat{z}_t = \hat{y}_t(\max(z_t) - \min(z_t)) + \min(z_t) \quad (9.6)$$

$$\hat{x}_t = \hat{z}_t + x_{t-12} \quad (9.7)$$

Hesaplanan öngörülerin performansının ölçülmesi için aşağıda verilen farklı hata ölçütleri hesaplanır.

$$RMSE = \sqrt{\frac{1}{ntest} \sum_{i=1}^{ntest} (x_t - \hat{x}_t)^2} \quad (9.8)$$

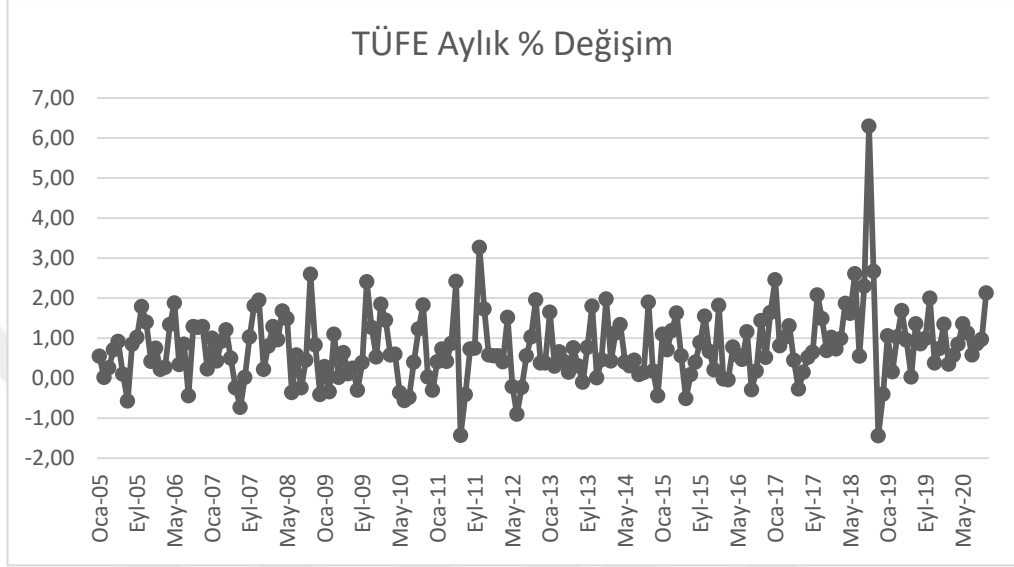
$$MAPE = \frac{100}{ntest} \sum_{i=1}^{ntest} \left| \frac{x_t - \hat{x}_t}{x_t} \right| \quad (9.9)$$

$$SMAPE = 200 \frac{1}{ntest} \sum_{i=1}^{ntest} \frac{|x_t - \hat{x}_t|}{|x_t| + |\hat{x}_t|} \quad (9.10)$$

$$MASE = \frac{1}{ntest} \sum_{i=1}^{ntest} \left| \frac{x_t - \hat{x}_t}{\frac{1}{n-1} \sum_{i=2}^n |x_t - x_{t-1}|} \right| \quad (9.11)$$

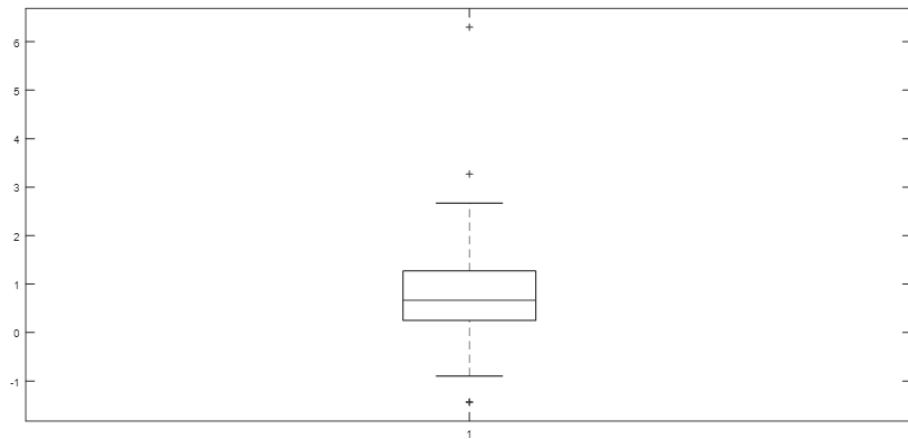
## 10. TÜKETİCİ FİYAT ENDEKSİ ZAMAN SERİSİNİN ÇÖZÜMLENMESİ

Uygulamada 2003 yılını baz alan (2003=100) Ocak 2005 ile Ekim 2020 tarihleri arasındaki tüketici fiyat endeksi aylık değişim yüzdesi zaman serisi kullanılmıştır. Zaman serisinin grafiği Şekil 10.1 de verilmiştir.



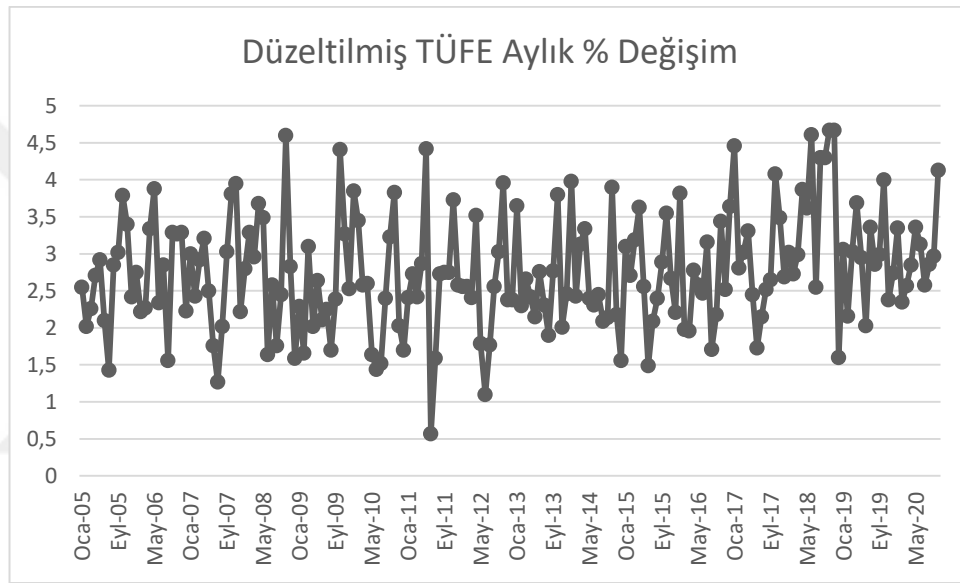
Şekil 10.1. Ocak 2005- Ekim 2020 TÜFE Aylık % Değişim Zaman Serisi

Zaman serisinin grafiği incelendiğinde, herhangi bir trend veya mevsimsellik ile ilgili dalgalanmaya rastlanmamıştır. Ancak seride aykırı değerler olabileceği değerlendirildiğinden Box-plot grafiği elde edilerek Şekil 10.2 de verilmiştir.



Şekil 10.2. TÜFE Zaman Serisi İçin Box-Plot Grafiği

Box-Plot grafiđi incelendiđinde 82'nci (Ekim 2011), 165'nci (Eylül 2018) ve 167'nci (Kasım 2018) gözlemlerin aykırı deđer olarak deđerlendirilebileceđi görülmüştür. Bu gözlemler silinerek, bu gözlemlerin yerine seri açık bir trend ve mevsimsellik içermediđinden bir önceki gözlemlerin deđerleri eklenerek TÜFE zaman serisi düzeltilmiştür. Ayrıca serideki negatif deđerlerin dışlanması için serideki tüm gözlemlere serinin minimum deđer -1.43'den mutlak deđerce daha büyük olan 2'deđer eklenmiştür. Düzeltilmişt TÜFE zaman serisinin grafiđi Şekil 10.3'de verilmiştür.



Şekil 10.3. Ocak 2005- Ekim 2020 Düzeltilmiş TÜFE Aylık % Deđişim Zaman Serisi

Zaman serisinin son 12 gözlemi test verisi olarak ilk 178 gözlem ise eğitim verisi olarak ayrılmıştır. Eğitim verisinin durađanlığı ve otokorelasyonları aştıđında incelenmiştür.

Zaman serisinin durađanlığının araştırılması için R programında "urca" kütüphanesi kullanılarak Dickey ve Fuller (1981) de geliştirilen birim kök testi aştıđında verilen iki modele göre sırasıyla gerçekleştirilerek test sonuçları Şekil 10.4 ve 10.5 verilmiştür.

$$\Delta x_t = \gamma_0 x_{t-1} + \sum_{k=1}^m \gamma_k \Delta x_{t-k} + \varepsilon_t \quad (10.1)$$

$$\Delta x_t = \beta_0 + \gamma_0 x_{t-1} + \sum_{k=1}^m \gamma_k \Delta x_{t-k} + \varepsilon_t \quad (10.2)$$

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####

Test regression none

Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
    Min       1Q   Median       3Q      Max
-3.1991 -0.4768  0.1390  0.7130  2.4573

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
z.lag.1      -0.03631    0.02488  -1.460   0.146
z.diff.lag  -0.31638    0.07229  -4.377 2.07e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9266 on 174 degrees of freedom
Multiple R-squared:  0.1223, Adjusted R-squared:  0.1123
F-statistic: 12.13 on 2 and 174 DF, p-value: 1.172e-05

Value of test-statistic is: -1.4596

Critical values for test statistics:
    1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Şekil 10.4. TÜFE Zaman Serisi İçin Model (10.1)'e göre Birim Kök Testi Sonucu

```

#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####

Test regression drift

Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)

Residuals:
    Min       1Q   Median       3Q      Max
-2.56883 -0.49335 -0.09899  0.46867  1.82555

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.35663    0.26742   8.812 1.26e-15 ***
z.lag.1     -0.85567    0.09526  -8.983 4.42e-16 ***
z.diff.lag   0.09308    0.07607   1.224  0.223
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.772 on 173 degrees of freedom
Multiple R-squared:  0.3942, Adjusted R-squared:  0.3872
F-statistic: 56.28 on 2 and 173 DF, p-value: < 2.2e-16

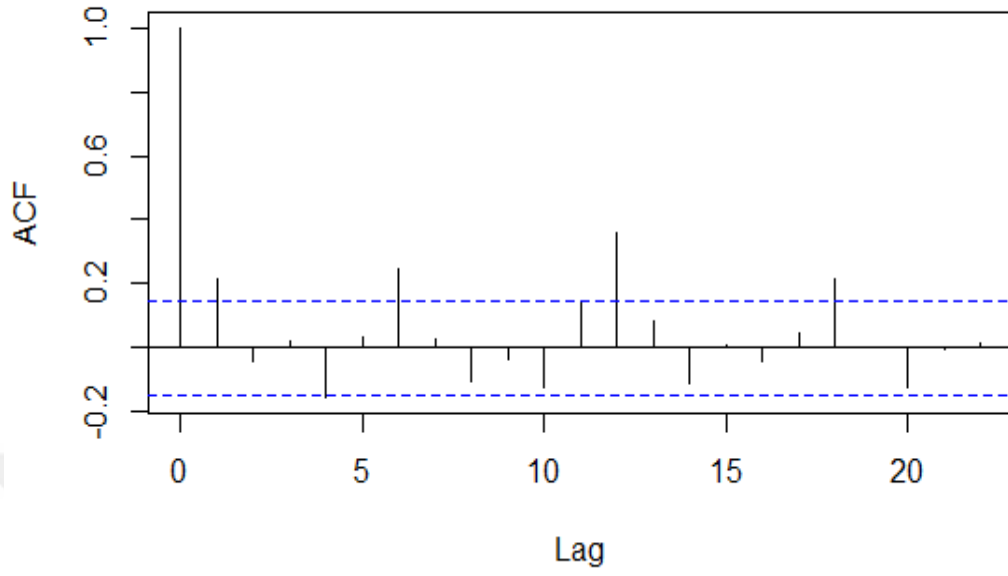
Value of test-statistic is: -8.9825 40.3644

Critical values for test statistics:
      1pct  5pct 10pct
tau2 -3.46 -2.88 -2.57
phi1  6.52  4.63  3.81

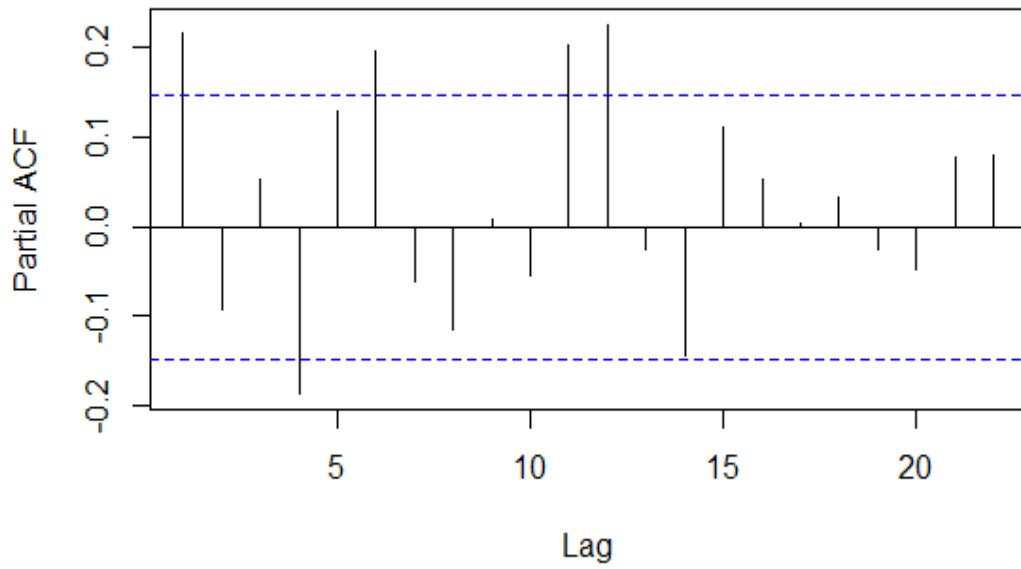
```

Şekil 10.5. TÜFE Zaman Serisi İçin Model 10.2'ye göre Birim Kök Testi Sonucu

(10.1) ve (10.2) modeline göre uygulanan ADF testlerinden sırasıyla 0.1123 ve 0.3872'lik düzeltilmiş çoklu belirtme katsayısı değerleri elde edilmiştir. Bu durumda ikinci modele göre elde edilen test sonuçları dikkate alınmıştır. ADF test sonucuna göre test istatistiğinin değer -8.9825 olarak elde edilmiş ve krtiki değerler olan -3.46, -2.88 ve -2.57 değerlerine göre daha küçük olduğundan seride birim kök olmadığı serinin durağan olduğu sonucuna varılmıştır. Serinin otokorelasyon ve kısmi otokorelasyon grafikleri elde edilerek Şekil 10.6 ve 10.7'de verilmiştir.



Şekil 10.6. TÜFE Zaman Serisi İçin Otokorelasyon Grafiği



Şekil 10.7. TÜFE Zaman Serisi İçin Kısmi Otokorelasyon Grafiği

Grafiksel incelemeye ek olarak,  $1.645 \sqrt{\frac{1+2(ACF_1+\sum_{i=2}^{s-1} ACF_i^2)}{n}} = 3.9243$  ,  $ACF_{12} = 0.3587$

olduğundan seride mevsimsellik olmadığı sonucuna varılmıştır. Güven sınırları dışında kalan kısmi otokorelasyonlar 1'nci, 4'üncü, 6'ncı, 11'nci ve 12'nci olarak gözükmektedir. Böylece çarpımsal nöron model için gecikmeli değişkenler aşağıdaki gibi belirlenmiştir.

$$p_1 = 1, p_2 = 4, p_3 = 6, p_4 = 11, p_5 = 12$$

Belirlenmiş gecikmeli değişkenler ise eğitimi verisi üzerinden ÇNM eğitilerek test kümesi için öngörüler elde edilmiştir. ÇNM'in test kümesi için performansı ARIMA, Holt üstel düzeltirme, çok katmanlı algılayıcı (MLP-YSA), dendritik nöron model YSA (DNM-YSA), Pi-sigma YSA (PS-YSA) yöntemleri ile karşılaştırılmıştır. Test kümesi için öngörü performansı ve sonuçlar Tablo 10.1'de verilmiştir. Tablo 10.1'e göre ÇNM-YSA yöntemi RMSE ölçütüne göre en iyi ikinci yöntem olmakla birlikte, MAPE, SMAPE ve MASE ölçütlerine göre en iyi sonucu üretmiştir. Bunun yanında ÇNM-YSA'nın MASE değeri 0,4211 olarak elde edildiğinden naive metoda görede oldukça başarılı öngörü sonucu ürettiği anlaşılmaktadır.

**Tablo 10.1.** Test kümesi öngörü sonuçları ve yöntemlerin performansları

<b>Gerçek</b>	<b>ARIMA</b>	<b>HOLT</b>	<b>ÇKA</b>	<b>DNM-YSA</b>	<b>PS-YSA</b>	<b>ÇNM-YSA</b>
<b>2,3800</b>	3,0402	3,4549	2,9574	2,9542	2,8742	2,8656
<b>2,7400</b>	2,5885	3,4584	2,6036	2,6965	2,3742	2,6956
<b>3,3500</b>	2,7870	3,4644	2,8633	2,8102	2,9628	2,9345
<b>2,3500</b>	2,8877	3,4781	2,9301	2,8959	2,9771	2,6982
<b>2,5700</b>	2,6185	3,4747	2,6651	2,7002	2,5148	2,7628
<b>2,8500</b>	2,7381	3,4715	2,8271	2,7749	2,9357	2,8180
<b>3,3600</b>	2,7773	3,4701	2,8754	2,8270	2,9691	2,7575
<b>3,1300</b>	2,8925	3,4751	2,9289	2,8950	2,9804	3,1583
<b>2,5800</b>	2,8081	3,4758	2,8910	2,8562	2,8951	2,8704
<b>2,8600</b>	2,6942	3,4665	2,7778	2,7623	2,7496	2,7750
<b>2,9700</b>	2,7905	3,4591	2,8768	2,8286	2,9699	2,8131
<b>4,1300</b>	2,7939	3,4517	2,8842	2,8416	2,9592	2,8575
<b>RMSE</b>	<b>0,5301</b>	<b>0,7166</b>	<b>0,4894</b>	<b>0,5040</b>	<b>0,4645</b>	<b>0,4716</b>
<b>MAPE</b>	<b>13,0610</b>	<b>23,5777</b>	<b>11,8362</b>	<b>12,1840</b>	<b>11,5293</b>	<b>10,5855</b>
<b>SMAPE</b>	<b>3,3529</b>	<b>5,1506</b>	<b>2,9999</b>	<b>3,1163</b>	<b>2,9123</b>	<b>2,7426</b>
<b>MASE</b>	<b>0,5114</b>	<b>0,8186</b>	<b>0,4597</b>	<b>0,4771</b>	<b>0,4422</b>	<b>0,4211</b>

Test kümesi performansı bir adım ileri öngörü doğruluğu üzerinden karşılaştırılmıştır. İkinci olarak ÇNM-YSA'nın öngörü performansı test ötesi veriler için çok adım ileri öngörü açısından karşılaştırılmıştır. Karşılaştırma sonuçları Tablo 10.2'de verilmiştir. Çok adım ileri öngörü performansı için ETS, ARIMA, sNAIVE, STL+ARIMA, STL+ETS yöntemleri R paket programında uygulanmıştır. ETS ve ARIMA yönteminin uygulanmasından elde edilen R çıktıları Şekil 10.8 ve 10.9'da verilmiştir.

```

ETS(A,N,A)
Call:
ets(y = TUFE)

Smoothing parameters:
  alpha = 0.0627
  gamma = 1e-04

Initial states:
  l = 2.6526
  s = -0.4452 0.2806 1.1144 0.0927 -0.3991 -0.5246
      -0.6817 0.0875 0.3505 -0.0331 -0.2646 0.4226

sigma: 0.612

      AIC      AICc      BIC
825.8274 828.5860 874.5327

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE
MASE      ACF1
Training set 0.03348013 0.5890587 0.4639593 -4.317244 19.03816 0.688
7803 0.144236

```

Şekil 10.8. R Programından elde edilen ETS yöntemi çıktıları

```

Series: TUFE
ARIMA(4,1,1)(2,0,0)[12]

Coefficients:
      ar1      ar2      ar3      ar4      ma1      sar1      sar2
      0.1788 -0.0446 0.0822 -0.1447 -0.9802 0.2275 0.2830
s.e. 0.0737 0.0780 0.0746 0.0745 0.0223 0.0734 0.0737

sigma^2 estimated as 0.4649: log likelihood=-195.03
AIC=406.07 AICc=406.87 BIC=432

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      M
ASE      ACF1
Training set 0.0451947 0.6673252 0.5266692 -5.560234 21.83699 0.7818
775 0.001377689

```

Şekil 10.9. R Programından elde edilen ARIMA yöntemi çıktıları

Kasım-Şubat ayları arası gerçekleşen TÜFE değerleri ve sırasıyla ETS, ARIMA, sNAIVE, STL+ARIMA, STL+ETS, PS-YSA, DNM-YSA yöntemlerinin çok adım öngörülleri ve bu öngörüler için hesaplanan RMSE, MAPE, SMAPE ve MASE ölçütleri değerleri Tablo 10.2’de verilmiştir. Tablo 10.2’den görüldüğü gibi en iyi çok adım test ötesi öngörü performansı tüm ölçütlere göre ÇNM-YSA ile elde edilmiştir.

**Tablo 10.2.** Çok adım öngörü performansı karşılaştırması

Aylar	Gerçekleşen	ETS	ARIMA	sNAIVE	STL+ARIMA	STL+ETS	PS-ANN	DNM-ANN	ÇNM
Kasım	2,3000	1,3320	1,4236	0,3800	1,6339	1,6342	1,2511	1,1045	1,6025
Aralık	1,2500	0,6062	0,5607	0,7400	0,6308	0,6312	0,7038	1,0722	1,2731
Ocak	1,6800	1,4740	1,1704	1,3500	0,9481	0,9484	0,7675	1,0095	0,8397
Şubat	0,9100	0,7868	0,5547	0,3500	1,4320	1,4324	0,5081	0,6031	0,6795
	RMSE	0,5935	0,6382	1,0451	0,6394	0,6392	0,7734	0,7079	0,5582
	MAPE	29,8478	40,6564	51,3649	44,8562	44,8515	46,9452	34,9596	26,8804
	SMAPE	9,3910	12,9675	19,0757	12,4990	12,4942	15,3896	10,9981	8,3299
	MASE	0,6470	0,8102	1,1067	0,8464	0,8462	0,9698	0,7836	0,5971

## 11. SONUÇ VE TARTIŞMA

Zaman serisi öngörü probleminin çözümü için yapay sinir ağları başarılı şekilde uygulanmaktadır. Bu çalışmada çarpımsal nöron model yapay sinir ağı ile 2003 yılını baz alan (2003=100) Ocak 2005 ile Ekim 2020 tarihleri arasındaki tüketici fiyat endeksi aylık değişim yüzdesi zaman serisi çözümlenerek, yöntemin bir adım ve çok adım öngörü performansı literatürde iyi bilinen klasik öngörü yöntemleri ve bazı makine öğrenmesi yöntemleri ile karşılaştırılmıştır. Zaman serisi öngörü probleminin ÇNM-YSA ile çözümü için bir algoritma önerilmiştir. Önerilen algoritma zaman serisinin otomatik bir şekilde çözümünü elde etmeye varmaktadır. Algoritmanın performansı TÜFE zaman serisi için kullanılan diğer klasik ve makine öğrenmesi öngörü yöntemlerine göre hem bir adım ileri hem de test verisi ötesinde çok adım ileri öngörü açısından üstün olduğu gözlemlenmiştir. TÜFE zaman serisi için elde edilen 2020 yılının son iki ayın ve 2021 yılının ilk 10 ayı için test verisi ötesinde elde edilen öngörüler Tablo 11.1’de verilmiştir.

**Tablo 11.1.** TÜFE zaman serisi için öngörüler

<b>Kasım-2020</b>	<b>Aralık-2020</b>	<b>Ocak-2021</b>	<b>Şubat-2021</b>	<b>Mart-2021</b>	<b>Nisan-2021</b>
1,6025	1,2731	0,8397	0,6795	0,7492	1,0701
<b>Mayıs-2021</b>	<b>Haziran-2021</b>	<b>Temmuz-2021</b>	<b>Ağustos-2021</b>	<b>Eylül-2021</b>	<b>Ekim-2021</b>
1,2332	1,1616	0,9722	0,8294	0,8420	0,9328

Gelecek çalışmalarda önerilen algoritmanın diğer ekonomik göstergeler içinde uygulanması planlanmaktadır. Ayrıca ÇNM-YSA ile çok adım ileri öngörüler için aralık tahminlerinin elde edilmesi planlanmaktadır.

## KAYNAKLAR

Bas, E. (2016). The training of multiplicative neuron model based artificial neural networks with differential evolution algorithm for forecasting. *Journal of Artificial Intelligence and Soft Computing Research*, 6(1), 5-11.

Bas, E., Uslu, V.R., & Egrioglu, E. (2016a). Robust learning algorithm for multiplicative neuron model artificial neural networks. *Expert Systems with Applications*, 56, 80–88.

Bas, E., Yolcu, U., Egrioglu, E., Cagcag Yolcu. O., & Dalar, A.Z. (2016b). Single multiplicative neuron model artificial neuron network trained by bat algorithm for time series forecasting. *American Journal of Intelligent Systems*, 6(3), 74-77.

Cagcag Yolcu O, Bas E, Egrioglu E, Yolcu U. (2018). Single multiplicative neuron model artificial neural network with autoregressive coefficient for time series modelling. *Neural Process Letters*. 47: 1133–1147.

Chatterjee S., Singh J.B., Nigam S., Upadhyaya L.N. (2013) A Study of a Single Multiplicative Neuron (SMN) Model for Software Reliability Prediction. In: Jordanov I., Jain L.C. (eds) *Innovations in Intelligent Machines -3. Studies in Computational Intelligence*, vol 442, 89-102, Springer, Berlin, Heidelberg.

Chuang, Chen-Chia & Su, Shun-Feng & Jeng, Jin-Tsong & Hsiao, Chih-Ching. (2002). Robust support vector regression networks for function approximation with outliers. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*. 13. 1322-30. 10.1109/TNN.2002.804227.

Corba B.S., Egrioglu E, Dalar AZ. (2020). AR–ARCH type artificial neural network for forecasting. *Neural Processing Letters*. 51:819-836.

Cui, H., Feng, J., Guo, J., & Wang, T. (2015). A novel single multiplicative neuron model trained by an improved glowworm swarm optimization algorithm for time series prediction. *Knowledge-Based Systems*, 88, 195–209.

Domaç, İ., (2003), “Explaining And Forecasting Inflation In Turkey”, TCMB Working Papers.

<http://unpan1.un.org/intradoc/groups/public/documents/apcity/unpan018318.pdf>

Erilli, N.A., Eğrioglu, E., Yolcu, U., Aladağ, Ç.H. ve Uslu, V.R. (2010). Türkiye’de Enflasyonun İleri ve Geri Beslemeli Yapay Sinir Ağlarının Melez Yaklaşımı İle Öngörüsü”, *Doğuş Üniversitesi Dergisi*, 11(1):42-55.

Eğrioglu E., Yolcu U., Baş E. (2019). *Yapay Sinir Ağları (Öngörü ve Tahmin Uygulamaları)*. Nobel Akademik Yayıncılık

Hill T, Marquez, L, O’Connor M., Remus W. (1994). Artificial neural networks for forecasting and decision making. *International Journal of Forecasting* 10:5–15.

Köse, N., N. Uçar ve S. Aksoy (2000), “Alternatif Modellerin Enflasyon Öngörü Performansı: Türkiye Örneği”, *Gazi Üniversitesi İktisadi Ve İdari Bilimler Fakültesi Dergisi*, 2(3), 37-52.

Liliana T. A. N. (2012). Artificial Neural Network Application in Gross Domestic Product Forecasting an Indonesia Case, *Journal of Theoretical and Applied Information Technology*, Vol:45, No:2, pp.410-415.

Kızılkaya O. (2017) Türkiye’nin Enflasyon ve İşsizlik Oranının Yapay Sinir Ağları ve Box-jenkins Yöntemi İle Tahmini, *Social Sciences Studies Journal*, Cilt:3, Sayı:12.

Samanta B. (2015). Single multiplicative neuron model as an alternative to multi-layer perceptron neural network. *Neural, Parallel, and Scientific Computations*, 23, 367-376.

Wu X, Mao J, Du Z, Chang Y. (2013). Online training algorithms based single multiplicative neuron model for energy consumption forecasting. *Energy*. 59: 126-132.

Yadav RN, Kalra PK, John J. (2007). Time series prediction with single multiplicative neuron model. *Applied Soft Computing*. 7: 1157-1163.

Yeh, W.C. (2013). New parameter-free simplified swarm optimization for artificial neural network training and its application in the prediction of time series. *IEEE Transactions on Neural Networks and Learning Systems*, 24 (4), 661-665.

Yolcu U, Jin Y, Egrioglu E. (2016). An ensemble of single multiplicative neuron models for probabilistic prediction. 2016 IEEE Symposium Series on Computational Intelligence (SSCI), 8 Pages, Athens, Greece.

Zhang G., Patuwo B.Y., Hu M. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14 (1):35-62.

Zhao L, Yang Y (2009) PSO-based single multiplicative neuron model for time series prediction. *Expert Systems with Applications* 36:2805-2812

Elektronik Kaynaklar

Türkiye Cumhuriyeti Merkez Bankası (nd), Tüketici fiyat endeksi 2003/100

<https://www.tcmb.gov.tr/wps/wcm/connect/TR/TCMB+TR/Main+Menu/Istatistikler/Enflasyon+Verileri/Tuketici+Fiyatlari>

Wikipedia (2021), Biological neuron models.

[https://en.wikipedia.org/wiki/Biological\\_neuron\\_model](https://en.wikipedia.org/wiki/Biological_neuron_model)

## EKLER

### Program:

```
function
[xong, rmsetest, msebest, pgbest, rmse, yhatg, yhattestg, rn, map
etest, ST, itn]=smnpsotestSTATong(xt, miM, ntest, nf)
m=length(miM);
ps=30; vmaps=1; w1=0.4; w2=0.9; c1i=2; c2i=1; c1f=1; c2f=2; itr=1
000;
x0=xt;
n0=length(xt);
xtest=x0((n0-ntest+1):n0);
xt=x0(1:(n0-ntest));
x=(xt-min(xt))/(max(xt)-min(xt));
saat=clock;
rand('seed', saat(6)*10000000);
rn=rand('seed');
p=2*m;
%p parametre sayısı
n=length(x);
A=unifrnd(0,1,ps,p);
V=unifrnd(-vmaps,vmaps,ps,p);
%Pso paramterleri program hızlı çalışması için başta
oluşturuluyor;
w=zeros(1,itr);
c1=zeros(1,itr);
c2=zeros(1,itr);
for k=1:ps
    ws=A(k,1:m);
    b=A(k,(m+1):2*m);
    yhat=multifSTAT(x,ws,b,miM);
    nh=length(yhat);
    for i=1:nh
        hata(i)=(yhat(i)-x(n-nh+i))^2;
    end
    mse(k)=mean(hata);
end
%en iyi parçacığı saklıyor.
MSEegt=min(mse);
for i=1:ps
    if MSEegt==mse(i)
        dd=i;
        break
    end
end
pgbest=A(dd,:);
msebest=mse(dd);
```

```

pid=A;
msepid=mse;
i22=0;
kk=0;msebesteski=msebest;
for i1=1:itr
    i22=i22+1;
    % Güncelleme
    w(i1)=(w1-w2)*((itr-i1)/itr)+w2;
    c1(i1)=(c1f-c1i)*(i1/itr)+c1i;
    c2(i1)=(c2f-c2i)*(i1/itr)+c2i;
    for i2=1:ps
        for i3=1:p

V(i2,i3)=V(i2,i3)*w(i1)+c1(i1)*unifrnd(0,1)*(pid(i2,i3)-
A(i2,i3))+c2(i1)*unifrnd(0,1)*(pgbest(i3)-A(i2,i3));
            V(i2,i3)=min(vmaps,max(-vmaps,V(i2,i3)));
            A(i2,i3)=A(i2,i3)+V(i2,i3);
        end
    end
    if i22>=20
        A=unifrnd(0,1,ps,p);
        V=unifrnd(-vmaps,vmaps,ps,p);
        i22=0;
    end
    for k=1:ps
        ws=A(k,1:m);
        b=A(k,(m+1):2*m);
        yhat=multifSTAT(x,ws,b,miM);
        nh=length(yhat);
        for i=1:nh
            hata(i)=(yhat(i)-x(n-nh+i))^2;
        end
        mse(k)=mean(hata);
    end
    %en iyi parçacığı saklıyor.
    MSEegt=min(mse);
    for i=1:ps
        if MSEegt==mse(i)
            dd=i;
            break
        end
    end
    if MSEegt<msebest
        pgbest=A(dd,:);
        msebest=mse(dd);
    %
        [i1,msebest]
    end
    if abs(msebest-msebesteski)>10^-6
        kk=0;

```

```

else
    kk=kk+1;
end
if kk>200
    break
end
msebesteski=msebest;
for j1=1:ps
    if mse(j1)<=msepid(j1)
        pid(j1,:)=A(j1,:);
        msepid(j1)=mse(j1);
    end
end
end
end
itn=il;
ws=pgbest(1:m);
b=pgbest((m+1):2*m);
yhat=multifSTAT(x,ws,b,miM);
nh=length(yhat);
yhatg=(yhat)*(max(xt)-min(xt))+min(xt);
for i=1:nh
    hata(i)=(yhatg(i)-xt(n-nh+i))^2;
    hata2(i)=abs((yhatg(i)-xt(n-nh+i))/xt(n-nh+i));
end
rmse=mean(hata)^0.5;
mape=mean(hata2);
x=(x0-min(x0))/(max(x0)-min(x0));
yhattum=multifSTAT(x,ws,b,miM);
nt=length(yhattum);
yhatterest=yhattum((nt-ntest+1):nt);
yhatterestg=(yhatterest)*(max(xt)-min(xt))+min(xt);
for i=1:ntest
    hata3(i)=(xtest(i)-yhatterestg(i))^2;
    hata4(i)=abs((xtest(i)-yhatterestg(i))/xtest(i));
end
rmsetest=(mean(hata3))^0.5;
mapetest=mean(hata4);
for i=1:nf
    yhatterestg=multifSTAT(x,ws,b,miM-1);
    xong(i)=yhatterestg(end)*(max(xt)-min(xt))+min(xt);
    x=[x;yhatterestg(end)];
end
xy=(x0-min(xt))/(max(xt)-min(xt));
for jj=1:m
    yhattumT(:,jj)=multifSTATSig(xy(1:end-ntest),ws,b,miM,jj);
end
xt1=xy(max(miM)+1:end-ntest);
XX=[ones(size(yhattumT,1),1),yhattumT];

```

```
[b,bint]=regress(xt1,XX);
for i=1:m
    if sign(bint(i+1,1))*sign(bint(i+1,2))=1
        ST(i)=1;
    else
        ST(i)=0;
    end
end
end
end
```



## ÖZGEÇMİŞ

Ortaokul ve liseyi Sümer Lisesi'nde tamamladı. 2010 yılında İnönü Üniversitesi Turizm ve Otel İşletmeciliği Bölümü'nden, 2012 yılında Anadolu Üniversitesi İktisat Fakültesi İktisat Bölümü'nden mezun oldu. 2017 yılında Giresun Üniversitesinde Pedagojik Formasyon eğitimi tamamladı ve 2020 yılında Eskişehir Atatürk Üniversitesi Açık Öğretim Fakültesi Laborant ve Veteriner Sağlık bölümünden mezun oldu. 2012 yılında Giresun Üniversitesi nde göreve başladı ve aktif olarak Giresun Üniversitesi Sağlık Uygulama ve Araştırma Merkezi Müdürlüğünde Bilgisayar İşletmeni olarak çalışmaktadır.

