



T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



İSTATİSTİK ANABİLİM DALI

PARÇACIK SÜRÜ OPTİMİZASYONUNA DAYALI EN KÜÇÜK BUDANMIŞ
KARELER YÖNTEMİ İLE ÇARPIMSAL NÖRON MODEL İÇİN DAYANIKLI
ÖĞRENME ALGORİTMASI

DOKTORA TEZİ

Özge GÜNDOĞDU
(11210566)

Tezin Savuma Tarihi : 26 Şubat 2016

Tez Danışmanı : Prof. Dr. Erol EĞRİOĞLU

Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü

İstatistik Anabilim Dalında

Özge Gündoğdu Tarafından Hazırlanan

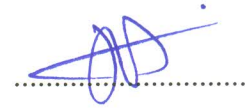
**PARÇACIK SÜRÜ OPTİMİZASYONUNA DAYALI EN KÜÇÜK
BUDANMIŞ KARELER YÖNTEMİ İLE ÇARPIMSAL NÖRON
MODEL İÇİN DAYANIKLI ÖĞRENME ALGORİTMASI**

başlıklı bu çalışma jürimiz tarafından 26/02/2016 tarihinde yapılan sınav ile
DOKTORA tezi olarak kabul edilmiştir.

Başkan : **Doç. Dr. Oğuz KAYNAR**
Cumhuriyet Üniversitesi



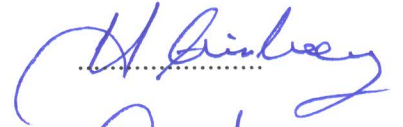
Jüri Üyeleri : **Prof. Dr. Erol EĞRİOĞLU**
Giresun Üniversitesi



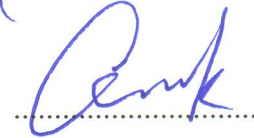
Doç. Dr. Kamil ALAKUŞ
Ondokuz Mayıs Üniversitesi



Yrd. Doç. Dr. Hakan TÜRKAY
Cumhuriyet Üniversitesi



Yrd. Doç. Dr. Cem KOÇAK
Hitit Üniversitesi



..../..../2016

Prof. Dr. Hüseyin DEMİR

Enstitü Müdürü

Canim Anneme,

ÖNSÖZ

İlk olarak; çalışmalarımı yönlendiren, bana duyduğu güven ile çalışmanın her aşamasında bana destek olan, değerli fikirlerini, bilgi birikimini paylaşan ve yardımlarını hiçbir zaman esirgemeyerek bilimsel ve kişisel anlamda yetişme ve gelişmeye katkıda bulunan, her zaman örnek aldığım ve yolundan ilerlediğim sevgili hocam, Sayın Prof. Dr. Erol EĞRİOĞLU'na teşekkürlerimi sunarım.

Tüm hayatım boyunca sonsuz sevgi ve desteklerini hiç esirgemeyen sevgili Babam ve Annem'e hiçbir fedakârlıktan kaçınmadan, tüm sıkıntılarda yanımda olup zorlukların üstesinden gelmemde bana yol gösterdikleri, her konuda bana güvenip destek oldukları, tüm kararlarımda beni cesaretlendirdikleri ve tüm emekleri için sonsuz teşekkür ederim.

Manevi destekleri sayesinde huzur bulmamı sağlayan, sevgi ve dualarını üzerimden hiç eksik etmeden hep yanımda olan sevgili Anneannem ve Dedem'e en içten teşekkürlerimi sunarım.

Birbirimizden ayrı kaldığımız bu uzun süreçte birlikte vakit geçirebilmek için beni sabırla bekleyen, sevgileri ve varlıkları ile beni mutlu eden, hayatımın gülen yüzleri olan başta sevgili ablam Özlem Kunt olmak üzere canım yeğenlerim Mert ve Defne'ye ve desteğiyle hep yanımda olan eniştem Atılğan Kunt'a en içten teşekkürlerimi sunarım.

Ve tabii ki tez dönemi boyunca desteklerini esirgemeyen, dertlerime ortak olup, yeri geldiğinde üzüntülerimi unutturup beni mutlu eden, iyi ki varlar dediğim mesafelerin ayıramadığı sevgili dostlarıma ve yanibaşımda olan sevgili dostlarıma teşekkürü bir borç bilirim.

Şubat 2016

Özge GÜNDOĞDU
Araştırma Görevlisi

İÇİNDEKİLER

Sayfa

| | |
|---|------|
| ÖNSÖZ..... | vii |
| İÇİNDEKİLER..... | ix |
| ÇİZELGELER LİSTESİ..... | xi |
| ŞEKİLLER LİSTESİ..... | xiii |
| KISALTMALAR | xv |
| PARÇACIK SÜRÜ OPTİMİZASYONUNA DAYALI EN KÜÇÜK BUDANMIŞ KARELER YÖNTEMİ İLE ÇARPIMSAL NÖRON MODEL İÇİN DAYANIKLI ÖĞRENME ALGORİTMASI..... | xvii |
| ÖZET..... | xvii |
| ROBUST LEARNING ALGORİTHM FOR MULTIPLICATIVE NEURON MODEL ARTIFICIAL NEURAL NETWORKS WITH LEAST TRIMMED SQUARES BASED ON PARTICLE SWARM OPTIMIZATION | xix |
| ABSTRACT | xix |
| 1. GİRİŞ..... | 1 |
| 2. YAPAY SİNİR AĞLARI..... | 9 |
| 2.1 Çarpımsal Nöron Model Yapay Sinir Ağları | 17 |
| 3. YAPAY SİNİR AĞLARININ EĞİTİMİNDE KULLANILAN ÖĞRENME ALGORİTMALARI | 21 |
| 3.1 Geri Yayılım Öğrenme Algoritması | 21 |
| 3.2 Parçacık Sürü Optimizasyonu Algoritması | 23 |
| 4. ZAMAN SERİLERİNDE AYKIRI DEĞER DURUMUNDA DAYANIKLI TAHMİN EDİCİLER | 29 |
| 4.1 En Küçük Kareler | 30 |
| 4.2 Budanmış Ortalama | 31 |
| 4.3 En Küçük Medyan Kareler | 32 |
| 4.4 En Küçük Budanmış Kareler | 33 |
| 4.5 En Küçük Budanmış Mutlak Değer | 34 |
| 5. BOOTSTRAP YÖNTEMİ..... | 35 |
| 5.1 Bağımlı Veri Yapısı için Bootstrap | 35 |
| 5.1.1 Blok bootstrap | 36 |
| 5.1.1.1 Örtüşmeyen blok bootstrap | 36 |
| 5.1.1.2 Hareketli blok bootstrap..... | 37 |
| 5.1.1.3 Dairesel blok bootstrap | 38 |
| 5.1.1.4 Durağan blok bootstrap..... | 38 |
| 5.1.2 Sieve bootstrap | 38 |
| 6. ÖNERİLEN YÖNTEM VE ALGORİTMASI | 41 |
| 6.1 Önerilen Yöntemin Algoritması..... | 43 |
| 6.2 Önerilen Yöntemin Akış Diyagramı | 47 |
| 6.3 Önerilen Yöntem için Rastgele Hareketli Kayan Blok Bootstrap | 47 |
| 7. UYGULAMA | 51 |
| 7.1 Benzetim Çalışması | 52 |

| | |
|--|------------|
| 7.2 Gerçek Hayat Zaman Serileri ile Çözümleme | 57 |
| 7.2.1. Avustralya bira tüketimi verisi kullanılarak yapılan çözümleme | 59 |
| 7.2.2. Ankara hava kirliliği verisi kullanılarak yapılan çözümleme | 63 |
| 7.3 Rastgele Hareketli Kayan Blok Bootstrap Yöntemi ile Güven Aralığı Tahmini ve Hipotez Testi..... | 66 |
| 7.3.1 Orijinal veri için çözümleme | 67 |
| 7.3.2 Aykırı değer içeren veri için çözümleme | 70 |
| 8. SONUÇ ve ÖNERİLER..... | 73 |
| KAYNAKLAR..... | 75 |
| EKLER | 83 |
| EK 1 | 84 |
| EK 2 | 91 |
| EK 3 | 96 |
| EK 4 | 101 |
| ÖZGEÇMİŞ..... | 103 |

ÇİZELGELER LİSTESİ

| | <u>Sayfa</u> |
|--|--------------|
| Çizelge 6.1. Parçacıklara ait pozisyon değerleri | 43 |
| Çizelge 6.2. Parçacıklara ait hız değerleri | 44 |
| Çizelge 6.3. Parçacıklara ait <i>pbest</i> vektörleri | 45 |
| Çizelge 6.4. Parçacıklara ait <i>gbest</i> vektörleri | 45 |
| Çizelge 7.1. Lilliefors normallik testi sonuçlarına göre kullanılacak test türü | 54 |
| Çizelge 7.2. Ölçütlere göre Kruskal Wallis testi sıra sayıları ortalamaları..... | 55 |
| Çizelge 7.3. Durumlara göre yöntemlerden elde edilen medyan değerleri..... | 56 |
| Çizelge 7.4. Orijinal AUST zaman serisi için tüm yöntemlerden elde edilen ölçütler | 60 |
| Çizelge 7.5. AUST test verisi için tüm durumlarda yöntemlerden elde edilen performans ölçütleri | 61 |
| Çizelge 7.6. Orijinal ANSO zaman serisi için tüm yöntemlerden elde edilen ölçütler | 64 |
| Çizelge 7.7. ANSO test verisi için tüm durumlarda yöntemlerden elde edilen performans ölçütleri | 65 |
| Çizelge 7.8. Orijinal seriye ait gerçek ve öngörü değerleri ile birlikte güven aralığı değerleri..... | 67 |
| Çizelge 7.9. Aykırı değer içeren seriye ait gerçek ve öngörü değerleri ile birlikte güven aralığı değerleri..... | 70 |

ŞEKİLLER LİSTESİ

| | <u>Sayfa</u> |
|--|--------------|
| Şekil 2.1. Genel yapay nöron modeli..... | 10 |
| Şekil 2.2. Çok katmanlı ileri beslemeli yapay sinir ağı | 11 |
| Şekil 2.3. Tek çarpımsal nöron model | 18 |
| Şekil 3.1. Klasik PSO algoritmasının akış diyagramı | 24 |
| Şekil 5.1. Örtüşmeyen blok bootstrap yöntemi için blokların elde edilişi..... | 36 |
| Şekil 5.2. Hareketli blok bootstrap yöntemi için blokların elde edilişi | 37 |
| Şekil 6.1. ÇNM-YSA’da bir parçacığın pozisyonları..... | 43 |
| Şekil 6.2. Önerilen yöntem için akış diyagramı | 47 |
| Şekil 6.3. Rastgele hareketli kayan blok bootstrap yönteminin akış diyagramı | 48 |
| Şekil 7.1. 1956-1994 yılları arasında gözlemlenen AUST zaman serisi | 59 |
| Şekil 7.2. AUST verisi Durum 1 için gözlenen değerler ile elde edilen öngörü değerlerinin birlikte grafiği..... | 61 |
| Şekil 7.3. Tüm durumlar için gözlenen değerler ile elde edilen öngörü değerlerinin birlikte grafikleri | 62 |
| Şekil 7.4. 1994-2006 yılları arasında aylık olarak gözlemlenen ANSO zaman serisi | 63 |
| Şekil 7.5. ANSO verisi Durum 1 için gözlenen değerler ile elde edilen öngörü değerlerinin birlikte grafiği..... | 64 |
| Şekil 7.6. Diğer durumlar için gözlenen değerler ile elde edilen öngörü değerlerinin birlikte grafikleri | 65 |
| Şekil 7.7. Anso zaman serisine ait test kümesi verilerinin grafiği | 66 |
| Şekil 7.8. Öngörüler için histogram grafikleri | 68 |
| Şekil 7.9. Öngörüler için elde edilen güven aralıklarının grafiği..... | 68 |
| Şekil 7.10. Öngörü ve gerçek değerlerin birlikte grafiği | 69 |
| Şekil 7.11. Ağırlıklar için histogram grafikleri | 69 |
| Şekil 7.12. Öngörüler için histogram grafikleri | 71 |
| Şekil 7.13. Öngörüler için elde edilen güven aralıklarının grafiği | 71 |
| Şekil 7.14. Öngörü ve gerçek değerlerin birlikte grafiği | 72 |
| Şekil 7.15. Ağırlıklar için histogram grafikleri | 72 |

KISALTMALAR

| | |
|------------------|---|
| ANN | : Artificial Neural Network |
| ANSO | : Ankara Hava Kirliliđi |
| AUST | : Avustralya Bira Tüketimi |
| ÇKA | : Çok Katmanlı Algılayıcı |
| ÇK-İB-YSA | : Çok Katmanlı İleri Beslemeli Yapay Sinir Ađı |
| ÇNM | : Çarpımsal Nöron Model |
| EKK | : En Küçük Kareler |
| E-YSA | : Elman Tipi Yapay Sinir Ađı |
| GY | : Geri Yayılım |
| GRY-PSO | : Garanti Yakınsamalı Parçacık Sürü Optimizasyonu |
| HBB | : Hareketli Blok Bootstrap |
| HKOK | : Hata Kareler Ortalaması Karekökü |
| LMS | : Least Median Squares |
| LTA | : Least Trimmed Absolute |
| LTS | : Least Trimmed Squares |
| MHYM | : Mutlak Hata Yüzdesi Medyanı |
| OMYH | : Ortalama Mutlak Yüzdellik Hata |
| PSO | : Parçacık Sürü Optimizasyonu |
| RT-YSA | : Radyal Tabanlı Yapay Sinir Ađı |
| SARIMA | : Mevsimsel Otoregresif Hareketli Ortalamalar |
| WMES | : Winters'ın Çarpımsal Üstel Düzleřtirmesi |
| YSA | : Yapay Sinir Ađı |

PARÇACIK SÜRÜ OPTİMİZASYONUNA DAYALI EN KÜÇÜK BUDANMIŞ KARELER YÖNTEMİ İLE ÇARPIMSAL NÖRON MODEL İÇİN DAYANIKLI ÖĞRENME ALGORİTMASI

ÖZET

Zaman serilerinin geçmişteki ve bugünkü bilgilerini kullanarak, gelecek durum ya da koşullar hakkında öngöründe bulunmak birçok bilim dalı için önemlidir. Öngörü yaparken kullanılan analiz yöntemleri ise öngörünün doğruluğu ve performansı üzerinde oldukça etkilidir. Son yıllarda yapay sinir ağları (YSA), üstün öngörü performansı ve geleneksel yöntemlerin aksine, sağlanması zor varsayımlara ihtiyaç duymaması sebebiyle öngörü elde etmede sıklıkla kullanılmaktadır. Literatürde yapısal özellikleriyle farklılaşan birçok YSA türü mevcuttur. Çarpımsal nöron model YSA (ÇNM-YSA), tek nörondan oluşması sebebiyle mimari yapı belirleme problemi içermediğinden kolay kullanım özelliğine ve iyi öngörü gücüne sahiptir. Zaman serilerinde aykırı değerlerin bulunması geleneksel yöntemlerde olduğu gibi YSA'da da öngörü performansını olumsuz etkilemektedir. Literatürde geleneksel yöntemler için önerilen dayanıklı tahmin yöntemleri YSA'ya uyarlanabilmektedir. Çok katmanlı algılayıcı için literatürde dayanıklı mimari yapıya ya da öğrenme algoritmasına sahip birçok dayanıklı yöntem önerilmiştir. Bununla birlikte mimari üstünlüğe sahip ancak çarpımsal birleştirme fonksiyonu kullanması sebebiyle aykırı değerlerden oldukça fazla etkilenen ÇNM-YSA için herhangi bir dayanıklı öğrenme algoritması mevcut değildir.

Bu tez kapsamında ÇNM-YSA için aykırı değer tespiti yapmadan da çalışabilecek dayanıklı bir öğrenme algoritması önerilmiştir. Önerilen yöntemde ağın eğitiminde uygunluk fonksiyonu olarak en küçük budanmış kareler tahmin edicisini temel alan garanti yakınsamalı parçacık sürü optimizasyonu kullanılmıştır. Aynı zamanda ÇNM-YSA'da istatistiksel değerlendirme yapmanın mümkün olduğunu göstermek üzere rastgele hareketli kayan blok bootstrap yöntemi kullanılarak parametreler için hipotez testleri yapılmış ve güven aralıkları tahmin edilmiştir. Önerilen yöntemin, aykırı değerlerin varlığı durumunda öngörü performansını değerlendirmek amacıyla; bir benzetim çalışması yapılmış ve iki farklı gerçek hayat zaman serisinin çözümü yapılarak alternatif yöntemlerin sonuçları ile karşılaştırılmıştır.

Anahtar Kelimeler: Öngörü; Yapay Sinir Ağları; Çarpımsal Nöron Model; Parçacık Sürü Optimizasyonu; Dayanıklı Öğrenme Algoritması; Hareketli Blok Bootstrap.

ROBUST LEARNING ALGORITHM FOR MULTIPLICATIVE NEURON MODEL ARTIFICIAL NEURAL NETWORKS WITH LEAST TRIMMED SQUARES BASED ON PARTICLE SWARM OPTIMIZATION

ABSTRACT

Forecasting about future conditions by using past and present data of time series is an important issue for many disciplines. Analysis methods used for forecasting are very effective on accuracy and performance. In recent year, artificial neural networks (ANNs) are often used in forecasting due to high forecasting performances and they do not need strict assumptions. Many ANNs types used for forecasting are available in the literature. Multiplicative neuron model ANN (MNM-ANN) that does not need to determine architectural structure, has a good forecasting power and ease of use. Existence of outliers in the time series has a negative effect on forecasting performance. Robust methods have been proposed for many traditional methods in the literature, and these methods can be adapted to ANNs. Many robust methods which have robust architecture structure or robust learning algorithm for multilayer perceptron have been proposed in the literature. Because the MNM-ANNs utilize multiplicative aggregation function, it is expected to affect more than other artificial network model. In literature, there is no robust learning algorithm for MNM-ANN.

In this thesis, a robust learning algorithm that ability to work without outliers' detection is proposed for MNM-ANN. Least trimmed squares method based guaranteed convergence particle swarm optimization method is used as fitness function for training of the network in the proposed method. Hypothesis testing and confidence intervals for parameters are estimated by using moving block bootstrap method. Simulation study and analysis for two different real time series are made to evaluate forecasting performance of the proposed method.

Key Words: Forecasting; Artificial Neural Network; Multiplicative Neuron Model; Particle Swarm Optimization; Robust Learning Algorithm; Moving Block Bootstrap.

1. GİRİŞ

Zaman serisi, ilgilenilen bir değişkenin zaman içerisinde değişim gösteren ölçümlerinden oluşan seriler olarak tanımlanabilmektedir. Zaman serileri kullanılarak yapılan analiz ve sonucunda elde edilen öngörü ise çeşitli bilim dalları için geleceğe ilişkin karar verme, politika üretme ve strateji belirlemede büyük önem arz etmektedir. Bu analiz ve elde edilen öngörünün doğruluğu, zaman serisinin yapısal özellikleri göz önünde bulundurularak seçilen uygun analiz yöntemine bağlıdır. Zaman serileri analizinde temel amaç, serinin geçmişte gözlenen ve şimdiki değerleri kullanılarak gelecekteki davranışları hakkında çıkarımlar yapmaktır. Zaman serilerinde, geçmişte elde edilen bilgilerden yararlanarak gelecek olayları ya da koşulları tahmin etmeye öngörü denilmektedir. Öngörü başarısı ise analizin yapıldığı dönemdeki serinin gözlenen özelliklerinin gelecek dönemlerde de büyük değişimlere uğramadan devam etmesine bağlı olmaktadır. Tam olarak doğru öngörüler yapmak mümkün olmamakla birlikte öngörü hataları azaltılarak ve öngörü elde etme süreci hızlandırılarak en iyi ve güvenilir öngörüler elde edilebilmektedir. Öngörü problemi, zaman serileri analizinde önemli bir rol oynamaktadır. Literatürde zaman serisi öngörüsü için kullanılan farklı yöntemler mevcuttur. Son yıllarda öngörü elde etmede geleneksel zaman serisi analiz yöntemlerine iyi bir alternatif olarak yapay sinir ağlarının kullanımı önerilmiştir. Yapay sinir ağları, kullanımının kolaylığı sebebiyle sıklıkla tercih edilmektedir. Geleneksel analiz yöntemleri doğrusal zaman serisi öngörüsünde başarı ile uygulanırken, eğrisel zaman serilerini modelleyemediğinden öngörü için farklı modeller kullanılmaktadır. Bunlar; Koşullu Değişen Varyanslı Otoregresif Model (ARCH) (Engel, 1982), Eşik Otoregresif Modeller (TAR) (Tong ve Lim, 1980) ve Bilinear Modeller (Granger ve Anderson, 1978) gibi eğrisel zaman serisi modelleridir. Bu modeller De Gooijer ve Kumar (1992) çalışmasında özetlenmiştir. Ancak bu eğrisel zaman serisi modelleri sadece belli eğrisel yapılar için kullanıldığından genel olarak yeterli esnekliğe sahip değildir (Zhang ve diğ., 1998). Yapay sinir ağları ise diğer yöntemlerin aksine kullandıkları aktivasyon fonksiyonu sayesinde doğrusal yapıların yanında eğrisel yapıları da modelleyebilme

özelliğine sahiptirler. Bu özellik gerçek hayat zaman serilerinin genellikle doğrusal olmayan yapıları içermesi sebebiyle önem taşımaktadır. Bunların yanında yapay sinir ağları, geleneksel yöntemlerin aksine sağlanması zor varsayımlara (doğrusal olma, hataların normal dağılıma sahip olması, sabit varyanslılık gibi) ihtiyaç duymaması sebebiyle modellemede esneklik sağlamak ve kısıtlamalara sebep olmamaktadır.

Literatürdeki çalışmaların çoğu yapay sinir ağlarının geleneksel yöntemlere göre daha kesin ve daha güvenilir sonuçlar verdiğini göstermektedir. Öngörü probleminin çözümünde birçok yapay sinir ağı türü kullanılmaktadır. Öngörü için yapay sinir ağlarının kullanılması fikri yeni değildir. Literatürdeki ilk uygulama, Hu (1964) tezinde yapay sinir ağları kullanılarak yapılan hava tahminidir. Marquez ve diğ. (1992), Zhang ve diğ. (1998) ve Hippert ve diğ. (2001) çalışmalarında, literatürde öngörü problemi için kullanılan yapay sinir ağı türlerini detaylı bir şekilde vermişlerdir. Literatürde en sık kullanılan yapay sinir ağı türü çok katmanlı algılayıcı ilk olarak Werbos (1974)'te önerilmiş, Rumelhart ve diğ. (1986)'da tekrar gözden geçirilerek günümüzde kullanılmakta olan halini almıştır. Smith ve Gupta (2002)'de zaman serisi öngörüsünde çok katmanlı algılayıcının kullanımı için bir algoritma ortaya koymuşlardır. Bununla birlikte çok katmanlı algılayıcı, Elman tipi geri beslemeli sinir ağları ve radyal temelli yapay sinir ağlarının öngörü problemi için kullanımında araştırmacı tarafından verilecek bazı kararlara ihtiyaç vardır. Bunlardan en önemlisi gizli katman birim sayısına karar verilmesi vasıtasıyla mimari yapının belirlenmesi durumudur. Gizli katman birim sayısının belirlenmesi için literatürde farklı yöntemler kullanılmıştır. Bu yöntemler polinomsal zaman algoritması Roy ve diğ.(1993), kanonik ayrıştırma Wang ve diğ. (1994), ağ bilgi ölçütü Murata ve diğ.(1994), iteratif yapılandırma algoritması Rathbun ve diğ. (1997), budama algoritmaları Lahnarjvi ve diğ. (2002); Reed (1993); Siestema ve Dow (1988), ağırlıklı bilgi kriteri Eğrioğlu ve diğ. (2008) olarak sıralanabilir. Bahsedildiği gibi çok katmanlı algılayıcı sinir ağlarında gizli katman birim sayısının belirlenmesi, araştırmacıya bağlı olmakla birlikte belirli bir ölçüt bulunmamaktadır. Bu nedenle yapay sinir ağlarında bulunması gereken gizli katman birim sayısı her model için farklı olmakta ve en iyi mimari belirlenmeye çalışılmaktadır. Gizli katman birim sayısının belirlenmesi ise yapay sinir ağının öngörü performansını önemli derecede etkileyen bir faktördür.

Son yıllarda zaman serisi öngörü probleminin çözümü için farklı yapay sinir ağı modelleri ve öğrenme algoritmaları önerilmiştir. Bunlardan bir tanesi ilk kez

Yadav ve diğ. (2007)'de önerilen çarpımsal nöron model yapay sinir ağlarıdır. Aladağ ve diğ. (2013) ve Yolcu ve diğ. (2013) çalışmalarında çarpımsal nöron modele dayalı iki farklı yapay sinir ağı türü ortaya koymuşlardır. Eğrioğlu ve diğ. (2013)'te bulanık zaman serisi öngörü algoritmasında çarpımsal nöron model yapay sinir ağlarını kullanmıştır. Çarpımsal nöron model, diğer yapay sinir ağları modellerinden farklı olarak çarpımsal birleştirme fonksiyonu kullanmaktadır. Tek nörondan oluşması sebebiyle diğer yapay sinir ağı türlerine göre daha basit bir ağ yapısına sahiptir. Çarpımsal nöron model yapay sinir ağlarının en önemli avantajlarından bir tanesi tek nörona sahip olması sebebiyle diğer yapay sinir ağı modellerinin aksine gizli katman birim sayısı belirleme problemini içermemesidir. Çarpımsal nöron modelin diğer yapay sinir ağı türlerine göre daha iyi sonuçları ürettiği Yadav ve diğ. (2007)'de gösterilmiştir. Bu yapay sinir ağı türü basit ağ yapısı, kolay kullanım özelliği ve iyi öngörü performansı nedeniyle literatürde sıklıkla kullanılmaya başlanmıştır. Çarpımsal nöron model yapay sinir ağlarının öngörü performansını etkileyen önemli bir faktör de öğrenme algoritmasıdır. Literatürde çok katmanlı ağlar için en sık kullanılan öğrenme algoritmaları geri yayılım algoritması ve Levenberg-Marquardt algoritmasıdır. Çarpımsal nöron model yapay sinir ağlarının eğitimi için ise literatürde farklı yöntemler kullanılmaktadır. Yadav ve diğ. (2007)'de çarpımsal nöron model yapay sinir ağının eğitimi için geri yayılım öğrenme algoritmasını önerirken, Zhao ve Yang (2009) ve Samanta (2011) çalışmalarında parçacık sürü optimizasyonuna dayalı eğitim algoritmasını kullanmışlardır.

Veri setinde bulunan, genel veri yapısının dışında gözlemlenen veriler aykırı değer olarak adlandırılırlar. Zaman serisi veri setinde aykırı değer ya da değerlerin varlığı tüm yöntemler için öngörü doğruluğunu ve güvenilirliğini olumsuz etkileyen bir durumdur. Aykırı değerlerin varlığı durumunda öngörü performansı etkilenmeyen ve tutarlı sonuçlar üreten yöntemler dayanıklı yöntemler olarak adlandırılmaktadır. Dayanıklı yöntemlerin en önemli özelliklerinden birisi de varsayımların sağlanmaması durumuna karşı duyarsız olmalarıdır. Literatürde geleneksel zaman serisi analizi için veri setinin aykırı değer içerdiği durumda kullanılmak üzere Huber (1964; 1965), Hampel (1968) ve Rousseuw (1984) tarafından dört önemli teori ortaya atılmıştır. Dayanıklı regresyon tahmin edicilerinden en sık kullanılanlar L-tahmin ediciler (Lineer Combination of Order Statistics: Sıra istatistiklerinin lineer kombinasyonları), M-tahmin ediciler (Maksimum Likelihood: En çok olabilirlik

tipi), R-tahmin ediciler (Rank Statistics: Rank istatistikleri) ve yüksek bozulma sınırlı tahmin ediciler olan LMS (Least Median Squares: En Küçük Medyan Kareler), LTS (Least Trimmed Squares: En Küçük Budanmış Kareler), LTA (Least Trimmed Absolute Deviations: En Küçük Mutlak Sapmalar) olarak verilebilir. Veri setinde aykırı değer bulunması geleneksel zaman serisi analiz yöntemlerinde olduğu gibi yapay sinir ağlarında da öngörü performansını olumsuz etkileyen bir durumdur. Aykırı değer varlığı yapay sinir ağlarının öğrenme algoritmalarında girdi ağırlıklarının belirlenmesinde de bazı sorunlara yol açmaktadır. Veri setinde aykırı değer bulunduğunda çok katmanlı algılayıcının öngörü performansı ve doğruluğunun bu durumdan olumsuz etkilendiği Hill ve diğ. (1994) ve Zhang ve diğ. (1998) çalışmalarında belirtilmiştir. Bunun üstesinden gelebilmek için, yapay sinir ağlarına dayanıklı tahmin yöntemleri kolaylıkla uyarlanabilmektedir. YSA'larda dayanıklılık kavramı, mimari yapının ve öğrenme algoritmasının dayanıklı olmasına göre farklılık göstermektedir. Borş ve Pitas (1996), Majhi ve diğ. (2012), Aladağ ve diğ. (2013) ve Yolcu ve diğ. (2015) çalışmalarında farklı dayanıklı yapay sinir ağı mimarileri önerilmiştir. Bununla birlikte yapay sinir ağlarında dayanıklı öğrenme algoritmalarının kullanılması yeni olmamakla birlikte farklı ağların eğitimi için birçok dayanıklı öğrenme algoritması önerilmiştir. Bu algoritmaların farklı kullanım özellikleri mevcuttur. Önerilen algoritmaların bir kısmında iki aşamalı olmak üzere; öncelikle veri aykırı değerlerden temizlenip sonrasında ağ eğitilmekte, bir kısmında önce aykırı değer tespiti yapıp daha sonra ağ eğitilmekte ve diğerlerinde ise veri üzerinde herhangi bir işlem yapılmadan öğrenme algoritması ile ağ eğitilmektedir. İlk dayanıklı öğrenme algoritmalarından biri Chen ve Jain (1994)'de önerilen M-tahmin ediciye dayalı olan dayanıklı geri yayılım öğrenme algoritmasıdır. Neubauer (1995) çalışmasında çok katmanlı algılayıcı için, Sanchez (1995)'te ise radyal tabanlı ağlar için dayanıklı bir öğrenme algoritması verilmiştir. Walczak (1996) ve Song ve diğ. (1999)'da çok katmanlı ileri beslemeli sinir ağlarının eğitimi için dayanıklı bir geri yayılım öğrenme algoritması önerilirken, Liano (1996)'da hataların Cauchy dağılımından geldiğini varsayan en küçük ortalama logaritma kareler yöntemi önerilmiştir. Wang ve diğ. (1997)'de kural tabanlı fonksiyon yaklaşımına dayanan bulanık sinir ağları için, Yang ve Chen (1998)'de ise Gaussian tabanlı fonksiyon olan olasılıksal sinir ağlarının eğitimi için maksimum olabilirlik kavramına dayanan dayanıklı yöntemler sunulmuştur. Hampel'in M-tahmin edicisi kullanılarak Lee ve diğ. (1999)'da radyal tabanlı fonksiyon sinir ağlarında aykırı değerlerin etkisini

azaltmak için dayanıklı bir öğrenme algoritması önerilmiştir. Tsai ve Yu (2000) ise ağıın eğitimi esnasında aykırı değerin etkisini azaltmak için bulanık sinir ağlarının dayanıklı bir öğrenme algoritmasını sunmuştur. Chuang ve diğ. (2000)'de aykırı değerlerle karşılaşıldığında geleneksel yöntemlerdeki sorunların üstesinden gelmek için önerilen dayanıklı tavlama geri yayılım öğrenme algoritması, uygun başlangıç ağırlıkları kümesinin seçiminde oldukça önemlidir. Chuang ve diğ. (2002)'de ise hata dağılımının genelleştirilmiş üstel dağılım ailesine ait olduğunu varsayarak adaptif dayanıklı öğrenme algoritmasını önermişlerdir. Allende ve diğ. (2002) genelleştirilmiş maksimum olabilirlik tahmin edicilerini temel alan dayanıklı öğrenme algoritmasını önermişlerdir. İleri beslemeli yapay sinir ağlarının öğrenme süreci verilere dayalı olduğundan modelin parametre tahminlerinin dayanıklı yöntemlerle yapılması gerekmektedir. Bu nedenle Allende ve diğ. (2003)'te Gaussian model altında dayanıklı ve etkili adaptif bir öğrenme algoritması geliştirilmiştir. Aykırı değerlerin sebep olduğu olumsuzluklara karşı çok katmanlı ileri beslemeli sinir ağının eğitim sürecinde Zhao ve diğ. (2004)'te ağırlıklandırılmış hata geri yayılım algoritmasının kullanımı önerilmiştir. Bunun yanında Chuang ve diğ. (2004)'te dayanıklı radyal tabanlı fonksiyon ağı için bir yaklaşım önerilirken, Juanyin ve diğ. (2004)'te sinir ağlarının ensemble'ı için dayanıklı bir öğrenme algoritması önerilmiştir. Espinoza ve diğ. (2005)'te doğrusal olmayan parametrik model tabanlı yeni bir dayanıklı TAO geri yayımlı öğrenme algoritmasını önermişlerdir. Hu ve Sun (2005)'te ise spline temelli sinir ağları için adaptif dayanıklı bir öğrenme algoritması önerilmiştir. Bulanık modellerin adaptif öğrenmesi için Kumar ve diğ. (2006)'da dayanıklı bir algoritma önerilirken, Yamawaki (2006)'da sinir ağları için Krein uzayında Kalman filtresi genişletilerek dayanıklı bir algoritma önermiştir. Chuang ve Jeng (2007) çalışmalarında CPBUM sinir ağları için etki fonksiyonlarına dayalı bir algoritma önermiştir. Rusiecki (2007)'de önerilen en küçük budanmış kareler tahmin edicisi ve Rusiecki (2013)'de önerilen artıkların en küçük budanmış mutlak değer tahmin edicisi kavramlarına dayalı dayanıklı öğrenme algoritmaları kullanılarak, aykırı değerin varlığı durumunda dayanıklılık kavramı için iyi sonuçlar ürettiği gösterilmiştir. Rusiecki (2008)'de minimum kovaryans determinant tahmin edicisini, Rusiecki (2012) ve El Melegy ve diğ.(2009)'da ise en küçük medyan kareler tahmin edicisini temel alan dayanıklı bir öğrenme algoritması sunulmuştur. Radyal tabanlı sinir ağları için Lee ve diğ. (2009)'da M-tahmin edicisine dayalı ve Fu ve diğ. (2009; 2011)'de geri yayımlı dayanıklı öğrenme

algoritmaları önerilmiştir. Fu ve diğ. (2010)'da ise tavlama dayanıklı bulanık sinir ağları kullanılarak kaotik zaman serilerinin tahminine yönelik bir yaklaşım önerilmiştir. Hans ve Udluft (2010) yapay sinir ağlarının ensemble'ı için dayanıklı bir öğrenme algoritması sunmuştur. Ninomiya (2011) çok katmanlı sinir ağlarının eğitimi için Quasi-Newton yöntemini, Sun ve Jin (2011) kanonik korelasyon analizini temel alan ve Man ve diğ. (2001) tek gizli katmana sahip ileri beslemeli sinir ağları için dayanıklı öğrenme algoritmaları önermişlerdir. Beliakov ve diğ. (2011)'de ise aykırı değer varlığında en küçük budanmış kareler tekniği ve ilgili uygunluk kriterleri kullanılarak yapay sinir ağlarının dayanıklılığı için yöntemler geliştirmişlerdir. İleri beslemeli sinir ağlarının öğrenme problemi için Jing (2011)'de yeni bir dayanıklı kontrol yaklaşımı ortaya atılırken, Jing (2012)'de doğrusal matris eşitsizliğine dayalı yeni bir öğrenme algoritması önerilmiştir. Majhi ve diğ. (2011)'de tek katmanlı fonksiyonel bağlantı yapay sinir ağlarının dayanıklılığının belirlenmesi için yeni bir yöntem önerilirken, Majhi ve diğ. (2012)'de Wilcoxon normunun minimizasyonu ile eğitilen çok katmanlı sinir ağı ve maliyet fonksiyonunda Wilcoxon normunu kullanan tek katmanlı wilcoxon fonksiyonel bağlantı yapay sinir ağı olmak üzere iki adet dayanıklı yapay sinir ağı yaklaşımı önerilmiştir. Hsiao ve diğ. (2012) öğrenme algoritması olarak M-tahmin ediciyi temel alan dayanıklı bir geri yayılım öğrenme algoritmasını kullanmışlardır. Ko (2012)'de parametreleri parçacık sürü optimizasyonu ile optimize edilen bulanık sinir ağlarının eğitiminde dayanıklı tavlama algoritmasının kullanımı önerilmiştir. Cao ve diğ. (2015)'te rasgele ağırlıklar ile sinir ağlarının modelleme performansını artırmak için olasılıksal dayanıklı bir öğrenme algoritması önerilmiştir. Vukovic ve diğ. (2015) çalışmasında ileri beslemeli sinir ağları için geleneksel genişletilmiş Kalman filtresine dayalı yeni bir dayanıklı öğrenme algoritması sunulmuştur. Bu çalışmaların tümünde çok katmanlı yapay sinir ağlarının eğitimi için dayanıklı öğrenme algoritmaları önerilmiştir.

Literatürde çok katmanlı yapay sinir ağlarının aykırı değere olan duyarlılığını ortadan kaldırmaya yönelik farklı dayanıklı öğrenme algoritmaları geliştirildiği görülmektedir. Bununla birlikte çarpımsal birleştirme fonksiyonu kullanması sebebiyle çok katmanlı yapay sinir ağlarına göre aykırı değerden daha fazla etkilenen çarpımsal nöron model yapay sinir ağı için herhangi bir dayanıklı öğrenme algoritması bulunmamaktadır. Halbuki dayanıklı bir öğrenme algoritmasının kullanılması ile birlikte ÇNM-YSA'nın aykırı değerlere olan duyarlılığı ortadan

kaldırılacak; mimari seçim problemi içermeyen, ağ yapısı sebebiyle kullanımı basit ve öngörü performansı aykırı değer durumunda bile üstün olabilecek bir yapay sinir ağı ortaya konulmuş olacaktır.

Literatürde zaman serisi öngörü problemi için yapay sinir ağları yaklaşımları istatistiksel değerlendirme imkanına sahip değildir. Başka bir ifade ile yapay sinir ağları kullanılarak yapılan analizlerde model girdilerinin katkısının istatistiksel testi ve öngörüler için güven aralıklarının elde edilmesi ile ilgilenilmemiştir. Buna karşın Twari ve diğ. (2010a, 2010b) çalışmalarında bootstrap tekniğini kullanarak yapay sinir ağlarından elde edilen öngörüler için güven aralıkları elde etmiştir. Ancak kullanılan bu bootstrap tekniğinde zaman serisinin ardışık gözlenme durumu göz ardı edildiğinden, zaman serisinin öngörü problemi için uygun bir yöntem olmamaktadır. Literatürde çarpımsal nöron model yapay sinir ağlarından olasılıksal çıkarımların yapıldığı bir çalışma mevcut değildir.

Bu tez kapsamında çarpımsal nöron model yapay sinir ağları için aykırı değer tespiti yapmadan da çalışabilecek (aykırı değer durumunda) dayanıklı bir öğrenme algoritması önerilmiş ve modele ait istatistiksel değerlendirmeler yapılmıştır. Önerilen yöntemde çarpımsal nöron model yapay sinir ağının eğitiminde uygunluk fonksiyonu olarak en küçük budanmış kareler tahmin edicisini temel alan garanti yakınsamalı parçacık sürü optimizasyonu (GRY-PSO) yöntemi kullanılmıştır. Bununla birlikte çarpımsal nöron model yapay sinir ağlarında da istatistiksel değerlendirme yapmanın mümkün olduğunu göstermek üzere hareketli blok bootstrap yöntemi kullanılmıştır.

Bu tez çalışmasının geri kalan kısmı şu şekilde planlanmıştır. Tez çalışmasının ikinci bölümünde, yapay sinir ağlarının genel özelliklerinden bahsedilmiş ve çarpımsal nöron model YSA tanıtılmıştır. Üçüncü bölümde yapay sinir ağlarının eğitiminde kullanılan ve ağın öngörü performansı üzerinde önemli bir etkiye sahip olan bazı öğrenme algoritmaları verilmiştir. Dördüncü bölümde çarpımsal nöron model yapay sinir ağının öğrenme algoritmasına katkıda bulunacak dayanıklı tahmin yöntemlerinin genel özelliklerinden bahsedilmiştir. Beşinci bölümde istatistiksel çıkarım yapmak amacıyla kullanılacak olan bootstrap yöntem tanıtılmıştır. Altıncı bölümde çarpımsal nöron model yapay sinir ağının aykırı değerlere olan duyarlılığını ortadan kaldırmaya yönelik olarak kullanılacak olan önerilen yöntem açıklanmıştır. Yedinci bölümde yapılan benzetim çalışması sonuçları, gerçek hayat zaman serileri ile yapılan analiz sonuçları, bootstrap yöntem yardımı ile elde edilen güven aralığı

değerleri ve hipotez testi sonuçları sunulmuştur. Son bölümde ise bulgular yorumlanarak, gelecek çalışmalara yol gösterecek açıklamalar yapılmıştır.

2. YAPAY SİNİR AĞLARI

Yapay sinir ağı (YSA), geçmişe ait verilerden oluşan zaman serilerinin modellenmesi ve geleceğe yönelik öngörü yapılması işlemlerinde, geleneksel zaman serisi analiz yöntemlerine alternatif olarak günümüzde sıklıkla kullanılan yöntemlerdir. Genel olarak biyolojik sinir ağlarından esinlenilerek oluşturulmuş matematiksel algoritmalar ya da paralel bilgi işleme sistemleri olarak tanımlanabilirler. YSA'lar diğer zaman serisi modelleme yöntemlerine göre daha iyi öngörü sonuçları vermektedir. Bunun nedeni geleneksel yöntemlerdeki varsayımlar sebebiyle oluşan sınırlamaları ortadan kaldırması ve ayrıca zaman serilerindeki hem eğrisel hem de doğrusal yapıyı öğrenebilme ve öğrendiği bilgiyi genelleştirebilme yeteneğine sahip olmasıdır. YSA, gizli katman sayısı ve gizli katman birim sayılarının değiştirilebilmesi nedeniyle veriye çok iyi uyum sağlayabilen veriye dayalı bir yaklaşım ortaya koymaktadır. Bahsedilen özellikler sayesinde YSA'nın karmaşık problemlerde bile basit kullanımı ve sağlanması zor varsayımlara ihtiyaç duymaması ile birlikte iyi öngörü gücüne sahip avantajlı bir yöntem olduğunu söylemek mümkündür. Gerçek hayat zaman serilerinin genellikle karmaşık ve doğrusal olmayan yapısı nedeniyle yapay sinir ağlarının kullanımı günümüzde oldukça yaygınlaşmıştır.

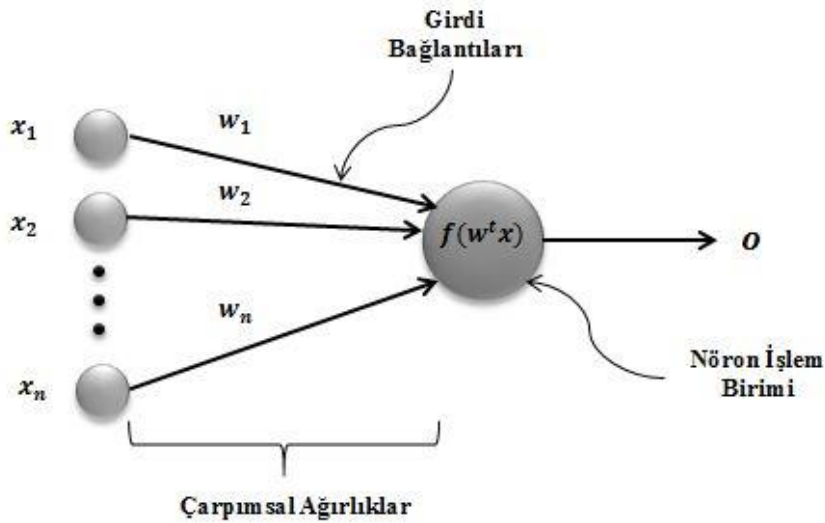
YSA'da öngörünün doğruluğu, zaman serisinin yapısal özellikleri göz önünde bulundurularak seçilen uygun ağ yapısına bağlıdır. Zaman serisi öngörü probleminin çözümünde birçok yapay sinir ağı türü kullanılmaktadır. YSA türlerindeki bu farklılık mimari yapı, öğrenme algoritması ve aktivasyon fonksiyonu gibi özelliklerinden kaynaklanmaktadır.

YSA'lar temel olarak; girdiler, ağırlıklar, birleştirme fonksiyonu, aktivasyon fonksiyonu ve çıktıdan oluşmaktadır. Ağın mimarisi, içerdiği nöronların birbirine bağlantı şekline ve bağlantı yönlerine göre ileri ve geri beslemeli ağlar olmak üzere ikiye ayrılmaktadır. Yapay sinir ağlarını oluşturan ve öngörü performansına etki eden bileşenler ise mimari yapı, birleştirme fonksiyonu, aktivasyon fonksiyonu ve öğrenme algoritması olmak üzere dört başlık altında toplanabilmektedir.

Mimari Yapı: Yapay sinir ağları, birbirlerine bağlı yapay nöronları (düğüm, birimler) içeren bir yapay ağ yapısına sahiptir. Ağ yapısının ve parametre değerlerinin belirlenmesinde belirli bir kural bulunmamaktadır. Bu değerlerin belirlenmesi için belirli bir standart olmamakla birlikte her problem için farklı bir yaklaşım söz konusu olabilmektedir.

YSA mimarisinde temel birim olarak yer alan her bir nöron için, biyolojik sinir ağlarındaki sisteme ve işleyişe benzer bir şekilde; nörona gelen girdi değeri ve bu girdinin karşılığında diğer nöron ya da nöronlara iletilmek üzere hesaplanan bir çıktı değeri mevcuttur. Böylece nörona gelen girdi değeri bir tek nörondan gelen çıktı değeri ya da birden fazla nörondan gelen tüm çıktı değerlerinin toplamı olabilmektedir. Nöron aldığı bu girdi değerini belirlenen bir fonksiyon yardımıyla çıktı değerine dönüştürür ve bu çıktı da diğer nöron ya da nöronlara girdi değeri olarak iletilir. Nöronlar arasındaki bu bilgi iletilmesi işlemine sinyal akışı adı verilir.

Yapay nöron modelinin ilk resmi tanıtımı 1943 yılında McCulloch ve Pitts tarafından verilmiştir. McCulloch-Pitts nöron modeli, zaman serilerinin çözümlemesinde kullanılacak nöron modelinin tanımlanması için bir başlangıç noktası oluşturur. Tanımlanacak yapay nöron modelinde, ağdaki tüm nöronların işlemlerinin eş zamanlı olarak yürütüldüğü varsayılır. Çıktı değerleri, $[0,1]$ aralığında herhangi bir değer alabilir. Ağırlıklar model içinde düzeltilir ve sinyal akışı dışında ağın nöronları arasında hiçbir etkileşim olmaz. Buna göre kullanılacak genel yapay nöron modeli Şekil 2.1’de gösterilmiştir.



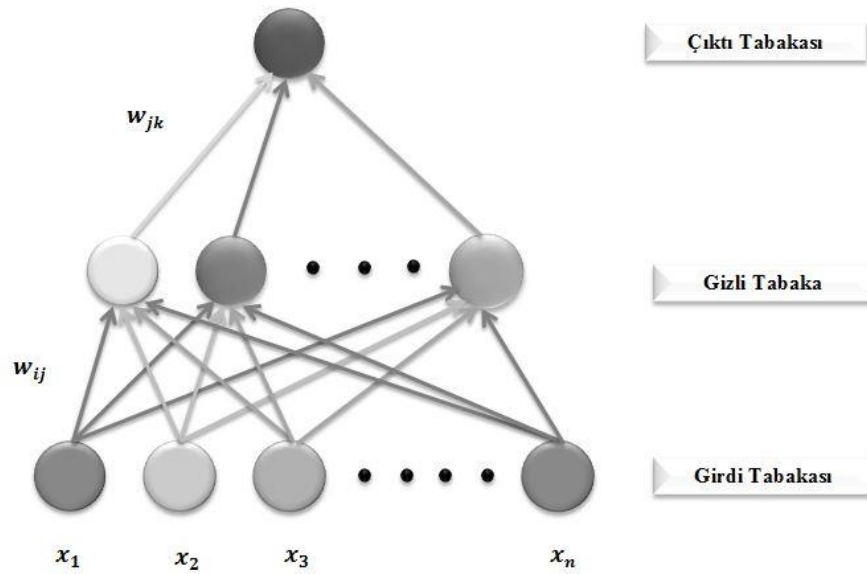
Şekil 2.1. Genel yapay nöron modeli

Her yapay nöron, girdi bağlantılarına sahiptir ve tek çıktılı bir işlem biriminden oluşur. Nöronun çıktı sinyali akışı gibi, x_i nöron girdilerinin sinyal akışı da tek yönlü olarak düşünülür. Eğer ağdaki tüm sinyal akışları ileri yönlü ise, bu ağa *ileri beslemeli yapay sinir ağı* adı verilir. Şekil 2.1’de görülen sembolik gösterim, nöron işlem birimini ve ağırlıkların bir kümesini gösterir. Nöron çıktı değeri;

$$o = f(w^t x) = f\left(\sum_{i=1}^n w_i x_i\right)$$

şeklinde verilir. w , ağırlık vektörü $w = [w_1 \ w_2 \ \dots \ w_n]^t$ olarak tanımlanır ve x girdi vektörü $x = [x_1 \ x_2 \ \dots \ x_n]^t$ şeklinde ifade edilir.

YSA mimarileri genel olarak tek girdi ve tek çıktı katmanından oluşur. Bununla birlikte girdi katmanı, gizli katman ve çıktı katmanı olmak üzere üç katmandan oluşan ağlar da bulunmaktadır. Bu mimariye sahip ağlara çok katmanlı algılayıcı yapay sinir ağları (ÇKA-YSA) denilmektedir. ÇKA-YSA mimarisinde birden çok gizli katman bulunabilir. En basit hali ile çok katmanlı ileri beslemeli bir yapay sinir ağı mimari yapısı Şekil 2.2’de verilmiştir.



Şekil 2.2. Çok katmanlı ileri beslemeli yapay sinir ağı

Şekilde 2.2’de görüldüğü gibi çok katmanlı ileri beslemeli bir yapay sinir ağı mimarisi üç kısımdan oluşur. Bunlar girdi katmanı, gizli katman (ya da katmanlar) ve çıktı katmanıdır. Katmanlar, nöron adı verilen birimlerden oluşmaktadır. Gizli katmanda bulunacak nöron sayısı, girdi ve çıktı katmanındaki nöron sayısından bağımsız olarak belirlenmektedir. Birden fazla gizli katman bulunan ağlarda her bir

katmandaki nöron sayıları birbirinden farklı olabilir. Gizli katman ve bu katmanlarda bulunan nöronların sayısının artması; işlem kargaşası yaratmasına ve hesaplama süresini arttırmasına rağmen, YSA'nın daha karmaşık problemlerin çözümünde de kullanılabilmesini sağlamaktadır. Mimari yapının tam olarak belirlenmesi katmanlar arasındaki ve her bir katmandaki nöron sayısına karar verilmesi, düzenlenmesi ile gerçekleşmektedir.

Yapay sinir ağlarında nöronlar birbirlerine ağırlıklar ile bağlıdır. İleri beslemeli ağlarda bu bağlantılar tek yönlü ve ileri doğrudur. Aynı katmanın birimleri arasında bağlantı yoktur (Günay, 2007). Bir katmanın yalnızca kendinden sonra gelen katmanlar ile bağlantısı bulunmaktadır. YSA'daki veriler sırasıyla girdi katmanına gelir daha sonra gizli katmandan ve çıktı katmanından geçerek ağın çıktısı elde edilir. Geri beslemeli yapay sinir ağlarında ise katmanlar arasındaki bağlantı ileri beslemeli ağlardan farklı olarak kısmen geriye doğrudur. Bir nöronun çıktısı yalnızca kendinden sonra gelen katmana girdi olarak verilmeyip, kendinden önceki katmanda ya da kendi katmanında bulunan herhangi bir nörona da girdi olarak iletilebilir. Geri beslemeli ağlarda çıktı katmanında elde edilen değer de kullanılarak ağın yeni ağırlıkları hesaplanır. Bu tür ağlarda bağlantıların döngüler içermesi sebebiyle, verilen girdiye karşılık yavaş bir şekilde çıktı verilir. Bu nedenle de ağın eğitim süreci ileri beslemeli ağlara göre daha uzun olmaktadır.

Zaman serilerinin gelecek tahmininde kullanılan en yaygın türü, 3 katmanlı ileri beslemeli sinir ağıdır. Tek değişkenli zaman serileri için ağın girdileri; değişkenin geçmiş veya gecikmeli değerlerinden, çıktısı ise elde edilen tahmin değerlerinden oluşur. Yapay sinir ağlarının verilen girdilere karşılık çıktı üretebilmesinin yolu ağın öğrenbilmesidir.

Öğrenme Algoritması: YSA'nın çok sayıdaki veri üzerinde eğitim adı altında işlem yaparak, ağa verilen girdi ve çıktı örnekleri yardımı ile tanımlanması zor fonksiyon kalıplarını öğrenbilme özelliği vardır. YSA'nın veri yapısındaki ilişkiyi öğrenip uygun ağ ağırlıklarını belirleyerek, verilen girdilere karşılık uygun çıktılar üretebilmesi ağın öğrenmesine bağlıdır. Öğrenme işlemi, ağırlıkların önceden belirlenen bir ölçüte göre en iyi değerinin bulunması işlemidir ve bu işlem yapay sinir ağlarının eğitimi olarak isimlendirilir. Yapılan bu işlem bir optimizasyon işlemi olarak düşünülebilir.

Ağın öğrenmesi, eğitim ve test aşaması olmak üzere iki kısma ayrılmaktadır. Eğitim aşamasında belirlenen öğrenme algoritmasına göre ağırlık değerleri hesaplanır ve bu değerler yardımıyla bir çıktı değeri elde edilir. Test aşamasında ise eğitim sırasında belirlenen ağırlık değerleri kullanılarak ağın daha önce görmediği örnekler ağı gösterilir ve bir çıktı üretilir. Elde edilen çıktıdan faydalanılarak ağın performansı test edilir. Genel olarak YSA, model geliştirme ve en iyi ağırlıkların elde edilmesinde eğitim kümesini kullanırken; modelin öngörü yeteneğinin değerlendirilmesi aşamasında ise test kümesini kullanmaktadır.

Ağın eğitiminin ne zaman sonlandırılması (ya da öğrenmenin tamamlanması) gerektiğine dair belirli bir yöntem yoktur. Ancak öğrenmenin tamamlanması hatalar kabul edilir bir düzeyin altına inince ya da belirlenen iterasyon sayısına ulaşıncaya kadar devam eder. Bazı durumlarda ağ hata düzeyinin üstünde bir seviyede takılı kalabilir. Bu gibi durumlarda ağın öğrenmesini iyileştirmek için ağ bileşenlerinde bazı değişiklikler yapılarak ağın yeniden eğitilmesi gerekir. Eğer ağ gereğinden fazla eğitilirse örnekleri öğrenmek yerine ezberler. Bu da ağın diğer örnekler için genelleme yapamamasına ve verilen örneklere karşılık hatalı çıktılar üretmesine neden olur.

Bir ağırlığın yeni ve eski değeri arasındaki fark olarak hesaplanan, ağırlıkların değişim miktarını belirlemek için tanımlanmış kurallar bütününe “öğrenme algoritması” adı verilir. Öğrenme algoritmaları YSA'nın çıktı değeri ile hedef değeri arasındaki farktan elde edilen hata değerine bakılarak parametre değişikliklerini belirlemekte kullanılırlar. Bu algoritmalar, ağına verilen girdi değerlerine karşılık uygun çıktı değerinin üretilmesinde ve ağları hızlandırmada önemli rol oynamaktadır.

YSA'da ağırlıkların belirlenmesinde kullanılan birçok öğrenme algoritması vardır. YSA'ların eğitiminde hatayı en aza indirmek için gradiyent azaltma (gradient descent) tabanlı algoritmalar kullanılmakla birlikte, bu algoritmalarından uygulamalarda en yaygın kullanıma sahip olanı Geri Yayılım (Back Propagation) Algoritmasıdır. Literatürdeki Quasi-Newton ve Levenberg-Marquardt gibi algoritmalar ise ikinci dereceden türev bilgisine dayalı doğrusal olmayan optimizasyon yöntemlerindedir. Bu yöntemler yerel minimuma ulaşabilme ve hızlı yakınsama özellikleri sayesinde YSA için önemli hale gelmişlerdir. Bunun yanı sıra eğitim için kullanılan farklı sezgisel algoritmalar da mevcuttur. Bunlar; Parçacık

Sürü Optimizasyonu, Genetik Algoritma, Yapay Arı Kolonisi Algoritması, Tavlama Benzetimi ve Tabu Arama Algoritması yöntemleridir.

Birleştirme Fonksiyonu: Birleştirme fonksiyonu, bir yapay sinir ağındaki nörona gelen girdilerin ağırlıklandırılarak toplanmasında ve modelin net değerinin hesaplanmasında kullanılır. Birleştirme fonksiyonu için nörona gelen girdilerin değeri ve sayısı bazı durumlarda önemli olmaktadır. Herhangi bir ağ için uygun birleştirme fonksiyonunun belirlenmesinde bir kural bulunmamaktadır. Bunun yanında bir ağa ait her bir nöronun birleştirme fonksiyonu birbirinden farklı olabilir. Ağın mimarisinin belirlenmesi gibi birleştirme fonksiyonunun seçimi de araştırmacıya bağlıdır. YSA için kullanılan farklı türde fonksiyon tipleri mevcuttur. Bunlar; toplam, çarpım, maksimum, minimum, çoğunluk ve kümülatif toplam fonksiyonlarıdır. Bunlardan bazıları aşağıda verilmektedir.

Toplam fonksiyonu: Ağırlık değerleri nörona gelen girdiler ile çarpılır ve bulunan değerler birbirleriyle toplanarak nörona ait *net* değeri hesaplanır.

$$net = \sum_{i=1}^N x_i \times w_i$$

Çarpım fonksiyonu: Ağırlık değerleri nörona gelen girdiler ile çarpılır ve bulunan değerler birbirleriyle çarpılarak nörona ait *net* değeri hesaplanır.

$$net = \prod_{i=1}^N x_i \times w_i$$

Birleştirme fonksiyonu yardımıyla elde edilen *net* değeri, belirlenmiş olan aktivasyon fonksiyonundan geçirilir ve sonucunda nörona ait çıktı değeri hesaplanır.

Aktivasyon Fonksiyonu: Nöron çıktı değerinin hesaplanmasında kullanılan $f(w^t, x)$ fonksiyonu, aktivasyon fonksiyonu olarak adlandırılır. Ağırlık ve girdi vektörünün skaler çarpımı olarak verilen ifade, net aktivasyon değeri olarak adlandırılır.

$$net = w^t x$$

Bu durumda aktivasyon fonksiyonunun tanım kümesi nöron modelinin *net* aktivasyon değerlerinin kümesi olacaktır.

Aktivasyon fonksiyonu, birleştirme fonksiyonundan elde edilen net girdiyi işleyerek nöronun bu girdisine karşılık bir çıktı değeri belirler. Aktivasyon fonksiyonunun seçiminin doğru yapılması ağın performansını önemli derecede

etkileyen bir faktördür. Aktivasyon fonksiyonu girdi ve çıktı birimleri arasındaki eğrisel eşleşmeyi sağlar (Zhang ve diğ., 1998). Günlük hayattaki modellerin çoğu doğrusal yapıya sahip değildir. Doğrusal olmayan bu yapıların modellenmesi yapay sinir ağlarında aktivasyon fonksiyonu yardımı ile yapılmaktadır. YSA'lardaki doğrusal olmama özelliği, aktivasyon fonksiyonunun eğrisel yapıları modelleyebilme özelliğinden kaynaklanmaktadır. Seçilen aktivasyon fonksiyonu doğrusal olmadığında eğim parametresinin belirlenmesi gerekmektedir. Eğim parametresi de optimal sonuca ulaşılmasında önemli rol oynayan bir faktördür. Yapay sinir ağlarının farklı türleri için farklı aktivasyon fonksiyonları kullanılır. YSA'nın kullanım amacına uygun olarak tek veya çift yönlü aktivasyon fonksiyonları kullanılabilir. Bununla birlikte aynı ağın içinde, nöronlarda, farklı aktivasyon fonksiyonları kullanılabilir. Genelde lojistik, tanjant hiperbolik ya da doğrusal olarak seçilir. Aktivasyon fonksiyonu seçilirken türevinin kolay hesaplanabilir olması işlem kolaylığı bakımından önemlidir.

Doğrusal olmayan bir yapıda ve kolay türevlenebilir olması sebebiyle; birçok YSA türünde kullanılan sigmoid aktivasyon fonksiyonu, literatürde yaygın bir kullanıma sahip olan çok katmanlı algılayıcı yapay sinir ağlarında da sıklıkla tercih edilmektedir. Sigmoid aktivasyon fonksiyonunun yapısı aşağıdaki gibidir;

$$f(net) = \frac{1}{1 + e^{-net}}$$

Buraya kadar olan kısımda YSA'ların ağ yapısı, genel özellikleri ve bu özellikleri sayesinde geleneksel yöntemlere sağladıkları üstünlüklerden bahsedilmiştir. Bu özellikler sayesinde YSA'larda üstün öngörü performansına ulaşmaya çalışılmaktadır. Yapay sinir ağlarında öngörü yapılması ise aşağıda aşamalarıyla birlikte verilmektedir.

Yapay Sinir Ağları İle Öngörü

Yapay sinir ağları sınıflama, örüntü tanıma gibi birçok alanda kullanılmakla birlikte, son yıllarda zaman serisi analizi ve öngörüsü için de yoğun bir şekilde kullanılmaktadır. Yapay sinir ağları yöntemiyle zaman serilerinde öngörü elde etmeyi Günay ve diğ. (2007)'de verildiği gibi 7 adımda özetlemek mümkündür.

Algoritma 1. YSA ile Öngörü

Adım 1. Verinin ön işlenmesi

İlk olarak verilerin aktivasyon fonsiyonuna uygun aralığa dönüştürülmesi gerçekleştirilir. x_i , girdi değerlerini göstermek üzere lojistik aktivasyon fonksiyonu kullanılacaksa, girdi değerleri

$$x'_i = \frac{x_i - \text{Min}(x_i)}{\text{Maks}(x_i) - \text{Min}(x_i)}$$

şeklinde $[0,1]$ aralığına dönüştürülebilir.

Adım 2. Veri organizasyonu

Eğitim ve test kümelerinin büyüklüğünün veri kümesinin yüzde kaç olacağına karar verilir.

Genelde verinin %10 veya %20'si, test kümesi olarak kullanılmaktadır.

Adım 3. Modelleme

Girdi sayısı, gizli katman sayısı, gizli katmandaki birim sayısı ve çıktı katmanındaki birim sayısı, bu birimlerde kullanılacak aktivasyon fonksiyonu, öğrenme algoritması ve bu algoritmanın parametreleri ve performans ölçütü belirlenerek, kullanılacak YSA modeli kurulur. Çarpımsal nöron model YSA kullanıldığında gizli katman birim sayısının belirlenmesi gerekmemektedir.

Adım 4. Girdi değerlerinin oluşturulması

YSA'nın girdi değerleri gecikmeli zaman serileridir. X_t , zaman serisi için girdi değerleri oluşturulurken, girdi katmanındaki birim sayısı m ile gösterilsin, m tane gecikmeli zaman serisi $X_{t-1}, X_{t-2}, \dots, X_{t-m}$, şeklinde oluşturulur.

Adım 5. En iyi ağırlık değerlerinin hesaplanması

Eğitim kümesi üzerinden, seçilen öğrenme algoritması ile en iyi ağırlık değerleri bulunur. Elde edilen en iyi ağırlık değerleri kullanılarak, kurulan YSA modelinin çıktı değerleri hesaplanır.

Adım 6. Performans ölçütünün hesaplanması

YSA'nın test kümesi tahminleri elde edilir. Adım 5'te elde edilen çıktı değerlerine ve bu adımda elde edilen değerlere, Adım 1'de uygulanan dönüşümün tersi uygulanır. Bu dönüşüm sonucu elde edilen değerler sırasıyla, eğitim kümesinin tahminlerini ve test kümesinin tahminlerini oluşturur. Test kümesinin tahminleri ile test kümesindeki verilerin arasındaki farka dayalı olarak, seçilen performans ölçütü hesaplanır.

Literatürde en yaygın kullanılan performans ölçütlerinden biri aşağıda formülü verilen hata kareler ortalaması karekök (HKOK) değeridir.

$$HKOK = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}$$

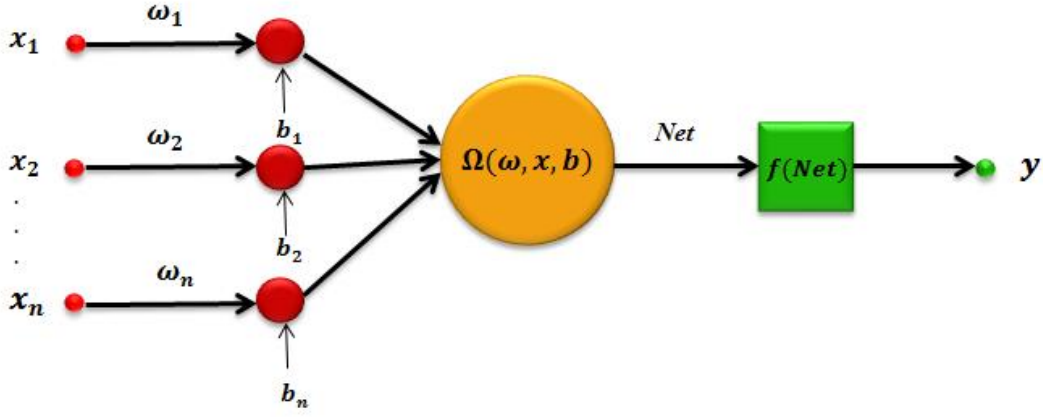
Adım 7. Öngörü

Son olarak, Adım 5'te bulunan en iyi ağırlık değerleri kullanılarak, test kümesinden sonraki zamanlar için yani, gelecek için öngörü değerleri, yine iteratif öngörü ya da doğrudan öngörü yöntemlerinden biri kullanılarak elde edilir (Günay ve diğ., 2007).

Yapay sinir ağlarının geleneksel yöntemlere sağladığı üstünlüğe karşın, çok katmanlı ağlar da mimari yapısı sebebiyle bazı dezavantajlara sahiptir. Ağ mimarisini belirleme tamamen araştırmacının tercihinine bağlıdır. Bu nedenle üstün performans gösterecek ağ mimarisini deneme yanılma yolu ile belirlemek zor ve zaman alıcı olmaktadır. Bu sorunun üstesinden gelmek amacıyla, tek nörondan oluşması sebebiyle, çok katmanlı ağlardaki mimari belirleme problemini içermeyen ve üstün öngörü performansına sahip olan çarpımsal nöron model yapay sinir ağları önerilmiştir.

2.1 Çarpımsal Nöron Model Yapay Sinir Ağları

Çarpımsal nöron model yapay sinir ağı (ÇNM-YSA) ilk olarak Yadav ve diğ. (2007)'de nöronların girdi ve çıktı değerlerinin çarpımsal fonksiyon temel alınarak hesaplandığı bir model olarak önerilmiştir. Çarpımsal nöron modele dayalı yapay sinir ağı diğer yapay sinir ağlarının kullandığı McCulloch ve Pitts nöron modelinden farklı olarak çarpımsal nöron modeli kullanmaktadır. Bu modelde, McCulloch ve Pitts (1943)'in ilk kez önerdiği nöron modelindeki birleştirme fonksiyonu toplamsal yerine çarpımsal olarak değiştirilmiştir. ÇNM-YSA'da gizli katman olmadığından tek katmandan oluşmaktadır ve bu nedenle öngörü performansında önemli bir etkiye sahip olan mimari yapının belirlenmesi sorununu içermemektedir. Bu nedenle ÇNM-YSA'nın kullanımı diğer YSA'lara göre daha basit ve öngörü gücü daha yüksektir. Yadav ve diğ. (2007)'de çarpımsal nöron model YSA'nın literatürde bazı iyi bilinen zaman serileri için çok katmanlı algılayıcıdan daha iyi öngörü performansına sahip olduğu gösterilmiştir. Tek çarpımsal nöron model YSA'nın mimari yapısı Şekil 2.3'te verilmiştir.



Şekil 2.3. Tek çarpımsal nöron model

Bu modelde tek bir nöron vardır ve ileri beslemeli sinir ağından farklı olarak nörona gelen sinyale toplama işlemi değil çarpma işlemi uygulanmaktadır. $\Omega(x, \Theta)$ fonksiyonu ağırlıklandırılmış girdilerin çarpımından oluşur. Burada $\Theta = (w_1, w_2, \dots, w_m, b_1, b_2, \dots, b_m)$ 'dir. Şekil 2.3'te verilen m girdili $(x_i, i = 1, 2, \dots, m)$ çarpımsal nöron model YSA'da $2m$ adet ağırlık bulunmaktadır. Bunlardan m tanesi girdilere $(w_i, i = 1, 2, \dots, m)$ karşılık gelen ağırlıklar ve diğer m tanesi de bu girdilere $(b_i, i = 1, 2, \dots, m)$ karşılık gelen yanlardır. Aktivasyon fonksiyonu aşağıda verilen lojistik fonksiyon olarak alındığını varsayalım.

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.1)$$

Bu durumda nöronun net değeri aşağıdaki gibi elde edilir.

$$net_i = \Omega(x, \Theta) = \prod_{i=1}^m (w_i x_i + b_i), \quad i = 1, 2, \dots, n \quad (2.2)$$

$$y_i = f(net_i), \quad i = 1, 2, \dots, n \quad (2.3)$$

Böylece ağın çıktısı net değerinin aktivasyon fonksiyonundan geçmesiyle $y = f(net)$ olarak elde edilir. Çarpımsal nöron model YSA'da hesaplanacak uygunluk fonksiyonu, tüm öğrenme örnekleri için hesaplanan çıktı değerleri ile hedef değerler arasındaki farktan hesaplanan aşağıda verilen kareler toplamı gibi bir ölçüt olarak alınabilir.

$$HKT = \sum_{i=1}^n (d_i - y_i)^2 \quad (2.4)$$

Burada d_i ve y_i sırasıyla i . öğrenme örneğine karşılık gelen hedef değer ve ağın çıktısını göstermektedir. Çarpımsal nöron model YSA'nın eğitiminde farklı

yöntemler kullanılmaktadır. Yadav ve diğ. (2007)'de çarpımsal nöron model YSA'nın eğitimi için geri yayılım öğrenme algoritmasını önerirken, Zhao ve Yang (2009) ve Samanta (2011) çalışmalarında parçacık sürü optimizasyonu algoritmasını kullanmıştır.

3. YAPAY SİNİR AĞLARININ EĞİTİMİNDE KULLANILAN ÖĞRENME ALGORİTMALARI

YSA'da ağırlıkların en iyi değerlerinin bulunması aşaması olarak belirtilen eğitim aşamasında kullanılan birçok öğrenme algoritması mevcuttur. Öğrenme algoritmasının seçimi, öngörü performansı üzerinde önemli derecede etkisi olan bir faktördür. Bu nedenle araştırmacının probleme uygun öğrenme algoritmasını seçmesi önemli olmaktadır. Literatürdeki uygulamalarda en yaygın kullanıma sahip olan öğrenme algoritmalarından biri yapay sinir ağlarına özgü olarak oluşturulan ve türev bilgisine dayalı geri yayılım algoritması iken eğitim için kullanılan sezgisel algoritmalarından en sık kullanılanlardan bir tanesi ise parçacık sürü optimizasyonu yöntemidir.

3.1 Geri Yayılım Öğrenme Algoritması

Geri yayılım algoritmasında (Back Propagation Algorithm) bilgi akışı ileri yönlü iken, hatanın yayılımı geriye doğru olmaktadır. Algoritma hatanın ağ elemanları üzerinde geri yayılımı sebebiyle bu ismi almaktadır. Algoritmanın işleyişi, hesaplanan çıktı ile hedef çıktı arasındaki farkı ifade eden hata değerine göre her katmandaki ağırlıkların güncellenmesi şeklindedir. Hata tüm ağırlıklara dağıtılarak dereceli olarak minimum düzeye indirilmeye çalışılır ve hata değeri belirlenen bir değer altında kalana dek işlemlere devam edilir. Hata değeri kabul edilebilir düzeye geldiğinde uygun ağırlık katsayılarına yaklaşılmış ve ağırlık eğitimi tamamlanmış olur. Eğitime rastgele ağırlıklar ile başlanır. Ağırlık başarısı belirlenen bu ilk ağırlık değerlerinin uygun seçimine bağlıdır.

Geri yayılım öğrenme algoritması, gradiyent tabanlı bir eğitim algoritmasıdır. Ağırlıkların değişimi, hatanın ağırlıklara göre zincir kuralı ile elde edilen kısmi türevlerine göre yapılır. Bu nedenle kullanılan aktivasyon fonksiyonunun türevleri kolay elde edilebilir bir fonksiyon olması gerekmektedir. Yadav ve diğ. (2007)'de lojistik aktivasyon fonksiyonu kullanıldığı durum için ağırlık ve yan değerlerinin

değişimi için gerekli formülleri türetmiştir. Her bir öğrenme örneği için hesaplanacak *net* değerinin formülü (3.1)'de verilmektedir.

$$net_i = \Omega(x, \Theta) = \prod_{i=1}^m (w_i x_i + b_i), \quad i = 1, 2, \dots, n \quad (3.1)$$

Yadav ve diğ. (2007)'de minimum yapılmaya çalışılan fonksiyon (3.2)'de verilmiştir.

$$E = \frac{1}{2n} \sum_{i=1}^n (y_i - d_i)^2 \quad (3.2)$$

Burada d_i , i . öğrenme örneği için hedef değerdir. Ağırlıkların değişimi ise (3.3) ve (3.4)'teki formüllere göre gerçekleştirilir.

$$\Delta w_i = -\eta \frac{dE}{dw_i} = -\eta y(y-d)(1-y) \frac{u}{(w_i x_i + b_i)} x_i \quad w_i = w_i + \Delta w_i \quad (3.3)$$

$$\Delta b_i = -\eta \frac{dE}{db_i} = -\eta y(y-d)(1-y) \frac{u}{(w_i x_i + b_i)} \quad b_i = b_i + \Delta b_i \quad (3.4)$$

Burada η öğrenme parametresi ve $u = \prod_{i=1}^n (w_i x_i + b_i)$ 'dir. Çarpımsal nöron model YSA'nın eğitimi için geri yayılım algoritması aşağıda verilen algoritmadaki gibi uygulanabilir.

Algoritma 1. Geri Yayılım öğrenme algoritması

Adım 1. $\Theta = (w_1, w_2, \dots, w_m, b_1, b_2, \dots, b_m)$ ağırlık ve yanları için rastgele başlangıç değerleri üretilir.

Adım 2. Tüm öğrenme örnekleri için (3.1) formülünden net_i , $i = 1, 2, \dots, n$ değerleri hesaplanır.

Adım 3. Tüm öğrenme örnekleri için ağırlık çıktıları $y_i = f(net_i)$, $i = 1, 2, \dots, n$ hesaplanır.

Adım 4. Bu çıktılarına göre (3.2)'deki formülden E değeri hesaplanır. Eğer $E < \varepsilon$ ise algoritma durdurulur. Aksi takdirde Adım 5'e geçilir. Burada ε değeri 10^{-6} gibi alınabilen hata toleransı değeridir.

Adım 5. (3.3) ve (3.4) formüllerine göre ağırlık ve yan değerleri güncellenir ve Adım 2'ye geri dönlür.

3.2 Parçacık Sürü Optimizasyonu Algoritması

Parçacık Sürü Optimizasyonu (PSO: Particle Swarm Optimization) ilk kez Kennedy ve Eberhart (1995)'te önerilmiştir. Son zamanlarda basit kullanımı, yüksek çözüm kalitesi ve iyi yakınsama özellikleri nedeniyle YSA'ların eğitiminde yaygın olarak kullanılmaktadır. PSO, kuş sürülerinin yiyecek ararken ortaya çıkan hareketlerini değerlendirmek için izlerini süren popülasyon tabanlı bir modern sezgisel arama yöntemidir. Burada rasgele dağılmış parçacıklar, problem uzayındaki çözümlere karşılık gelmektedir. Sürüler olarak adlandırılan bu parçacıklar kişisel pozisyonlarına rağmen, diğer parçacıkların her birinden bilgi toplamaktadırlar. Parçacıklar, kendi deneyimlerini ve komşularının deneyimlerini kullanarak pozisyonlarını güncellerler. Güncelleme durumu parçacıkların hızı olarak isimlendirilir. Her bir çözüme karşılık gelen bir hız vektörü vardır. i . parçacığın pozisyon ve hız vektörleri sırasıyla d -boyutlu arama uzayında $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ ve $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$ ($i = 1, 2, \dots, ps$) olarak gösterilebilir. Burada d problemdeki optimize edilecek değişken sayısıdır ve PSO'nun her iterasyonunda ps adet çözüm bulunmaktadır. Ölçüm fonksiyonunun değerinin temelinde, bir parçacığın önceki pozisyonlarının en iyisi $pbest_i = (p_{i1}, p_{i2}, \dots, p_{id})$ olarak kaydedilir ve gösterilir. Eğer g . parçacık, gruptaki tüm parçacıklar arasında şimdiye kadarki en iyi ise $pbest_g = gbest = (pg_1, pg_2, \dots, pg_d)$ olarak gösterilir. Parçacık, $pbest$ ve $gbest$ 'e ait geçerli hız ve uzaklığı kullanarak pozisyonunu değiştirmeyi dener. Bir sonraki iterasyonda uygunluk ölçümü için her parçacığın hız ve pozisyon değişimi aşağıda verilen eşitlikler kullanılarak hesaplanır. $k + 1$. iterasyondaki parçacığın hızı ve buna bağlı olarak belirlenen parçacığın yeni pozisyonu;

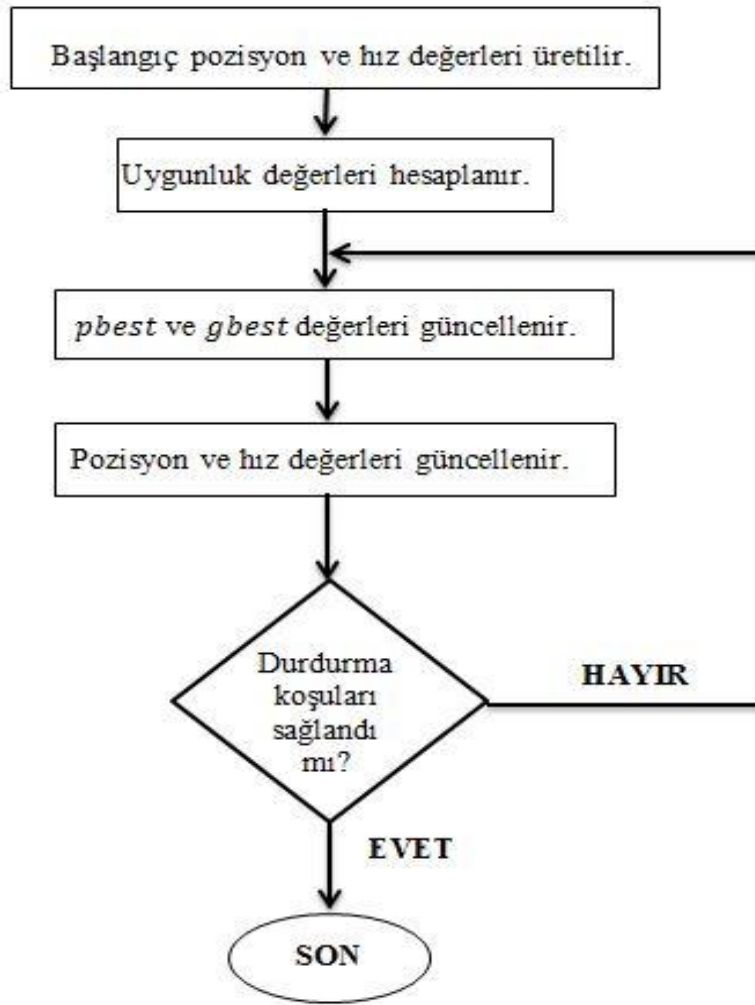
$$v_{id}^{k+1} = c_1 \times rand_1 \times (pbest_{id} - x_{id}) + c_2 \times rand_2 \times (gbest_{gd} - x_{id}), \quad (i = 1, 2, \dots, ps) \quad (3.5)$$

$$x_{id}^{k+1} = x_{id} + v_{id}^{k+1}, \quad (i = 1, 2, \dots, ps) \quad (3.6)$$

şeklindedir. Burada c_1 bilişsel ve c_2 sosyal katsayılar ve $rand_1, rand_2$ ise (0,1) aralığındaki rasgele sayılardır.

İlk iterasyonda $pbest$ değerleri başlangıç pozisyonlarından oluşmaktadır. $gbest$ değeri ise en iyi uygunluk değerine sahip başlangıç çözümü olmaktadır.

Klasik PSO algoritmasının uygulanma aşamasına ait akış diyagramı aşağıda Şekil 3.1'de verildiği gibidir.



Şekil 3.1. Klasik PSO algoritmasının akış diyagramı

Klasik PSO yöntemi yakınsama problemi içermesi sebebiyle, yöntemin yakınsama hızını artırmaya yönelik bazı geliştirmeler yapılmıştır. Yönteme yapılan ilk katkı eylemsizlik ağırlığı parametresinin eklenmesidir. Eylemsizlik ağırlığı w , hız değeri güncellemesine dahil edilirse eşitlik;

$$v_{id}^{k+1} = w \times v_{id}^k + c_1 \times rand_1 \times (pbest_{id} - x_{id}) + c_2 \times rand_2 \times (gbest_{gd} - x_{id}) \quad (3.7)$$

şeklinde olmaktadır. Yöntemin yakınsama hızına katkıda bulunacak geliştirmelerden bir diğeri ise c_1 , c_2 ve w parametre değerlerinin iterasyonlar içerisinde zamanla doğrusal olarak değiştirilmesidir (Chaturvedi ve diğ., 2009).

Büyük bir eylemsizlik ağırlığı iyi küresel aramaya yardımcı olurken, daha küçük değer yerel araştırmaya olanak tanır. Bu nedenle, uygulama ilk aramada daha büyük eylemsizlik ağırlığı faktörü kullanılır ve daha sonraki iterasyonlarda bu değer

kademeli olarak azaltılarak aramaya devam edilir. Zamanla değişen eylemsizlik ağırlığı kavramı aşağıda verildiği gibidir.

$$w = (w_{max} - w_{min}) \times \frac{iter_{max} - iter}{iter_{max}} + w_{min} \quad (3.8)$$

Burada $iter$ şimdiki iterasyon sayısı iken $iter_{max}$ iterasyonların maksimum sayısıdır. Genellikle w değeri 0.4 ile 0.9 aralığında değişir. c_1 , parçacıkları yerel en iyi pozisyona doğru çekerken, c_2 küresel en iyiye doğru çeker. Bilişsel ve sosyal katsayıların değişimi için kullanılan eşitlikler;

$$c_1 = (c_{1f} - c_{1i}) \times \frac{iter}{iter_{max}} + c_{1i} \quad (3.9)$$

$$c_2 = (c_{2f} - c_{2i}) \times \frac{iter}{iter_{max}} + c_{2i} \quad (3.10)$$

olmaktadır. Burada c_{1i} , c_{1f} ve c_{2i} , c_{2f} bilişsel ve sosyal hızlandırma faktörlerinin başlangıç ve final değerleridir. Algoritmada w 'nin değeri doğrusal olarak azaltılırken c_1 ve c_2 değerleri artırılmaktadır. Bununla birlikte algoritmada hızın sınırlandırılması da mümkündür. Her bir parçacığın hızı;

$$-v_{maps} < v_{ij} < v_{maps}$$

ile sınırlandırılarak parçacıkların pozisyonlarının hızlı değişimine izin verilmeyebilir. Literatürdeki uygulamalarda;

$w_{max} = 0.9$ $w_{min} = 0.4$, $c_1 = c_2 = 2$ veya $c_{1i} = c_{2f} = 2.5$, $c_{2i} = c_{1f} = 0.5$ alınması durumunda yöntemin yakınsama hızının artırıldığı sonucuna varılmıştır.

Literatürde PSO'nun birçok versiyonu bulunmakta ve kullanılmaktadır. Bunlardan bir tanesi en iyi parçacığın güncellenmesinin farklı şekilde yapılması ile ortaya çıkan garanti yakınsamalı parçacık sürü optimizasyonu (GRY-PSO: Guaranteed Covergence Particle Swarm Optimization) yöntemidir. Standart PSO algoritmasının adımları aşağıdaki algoritmada olduğu gibi uygulanır.

Algoritma 2. Standart PSO Algoritması

Adım 1. Parçacıkların başlangıç hız ve pozisyon vektörleri rastgele üretilir.

Adım 2. Uygunluk değerleri hesaplanır.

Adım 3. $pbest$ ve $gbest$ güncellenir. Eğer $gbest$ 'in uygunluk değeri belirli bir ε değerinin altındaysa ya da maksimum iterasyon sayısına ulaşılmış ise işlemler durdurulur. Aksi halde Adım 4'e geçilir.

Adım 4. Pozisyonların hız değerleri, pozisyonlar güncellenir ve Adım 2'ye geri dönlür.

$$v_{ij}^{k+1} = w \times v_{ij}^k + c_1 r_1 (pbest_{ij} - x_{ij}) + c_2 r_2 (gbest_j - x_{ij}) \quad (3.11)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (3.12)$$

Burada w eylemsizlik ağırlığı, c_1 bilişsel katsayı ve c_2 ise sosyal katsayıdır. r_1 ve r_2 (0,1) parametrelili düzgün dağılımdan üretilen rastgele sayılardır. $pbest_{ij}$ i . parçacığın en iyi durumu için j . pozisyonundaki değeridir. $gbest_j$ tüm parçacıkların en iyi durumu için j . pozisyonundaki değeridir. x_{ij} , i . parçacığın j . pozisyon değerini, v_{ij} ise i . parçacığın j . hız değerini gösterir. k ise iterasyon numarasını gösteren değişkendir.

Literatürde eylemsizlik ağırlığı, bilişsel ve sosyal katsayıların iterasyonlar içinde sabit alınması yerine değiştirilmesinin sonuçlar üzerinde olumlu etkisi olduğu gösterilmiştir. Ma ve diğ. (2006) çalışmasında bilişsel ve sosyal katsayılar (3.9) ve (3.10) formüllerindeki gibi iterasyonlar içinde doğrusal olarak değiştirilmiştir.

Shi ve Eberhart (1999) çalışmasında ise eylemsizlik ağırlığının değiştirilmesi için (3.13) eşitliğinde verilen formülü kullanmıştır.

$$w = (w_2 - w_1) \frac{\max t - t}{\max t} + w_1 \quad (3.13)$$

Van Der Bergh ve Engelbretch (2002) çalışmasında (3.11) formülünün en iyi parçacık ile kullanılmasında son iki teriminin sıfır olduğunu ve ağırlık değişiminin sadece eylemsizlik ağırlığına bağlı (3.14)'teki gibi değiştiğini ortaya koymuştur.

$$v_{ij}^{k+1} = w * v_{ij}^k \quad (3.14)$$

Bu problemi gidermek için Van Der Bergh ve Engelbretch (2002) çalışmasında en iyi parçacığın hız ve pozisyon güncellenmesinde (3.15) formülünün kullanımını önermiştir.

$$v_{ij}^{k+1} = w * v_{ij}^k - x_{ij}^k + gbest_j + \rho(k) * r_3 \quad (3.15)$$

Burada r_3 , (-1,1) parametrelili düzgün dağılımdan üretilen rastgele sayıdır. $\rho(k)$ ise (3.16) eşitliğine göre her iterasyonda hesaplanır.

$$\rho(k+1) = \begin{cases} 2\rho(k), & s_n > s_c \\ 0.5\rho(k), & f_n > f_c \\ \rho(k), & \text{aksi halde} \end{cases} \quad (3.16)$$

Bu formülde $\rho(0) = 1$ olmakta, s_n başarı sayısı, f_n başarısızlık sayısı ve s_c başarı sayısı için, f_c ise başarısızlık sayısı için üst limittir. Eğer en iyi parçacığın numarası değişmeden, iyileşme sağlar ise başarı sayısı artırılır ve başarısızlık sayısı sıfırlanır, iyileşme sağlanmadıysa başarısızlık sayısı artırılır ve başarı sayısı sıfırlanır. Eğer g_{best} , farklı bir parçacık numarası alırsa, hem başarı hem de başarısızlık sayısı sıfırlanır.

Algoritma 2’de en iyi parçacığın hızı (3.11) formülü yerine (3.15) ile güncellenir ise algoritmaya GRY-PSO yöntemi adı verilir.

Çarpımsal nöron model YSA’nın PSO algoritmaları ile eğitiminde parçacıkların pozisyonları YSA’nın ağırlık ve yanlarını göstermektedir. Çarpımsal YSA’da n adet girdi kullanılırsa, ağırlıklar ve yanların toplam sayısı $2n$ olacağından, parçacıkların pozisyon sayısı da $2n$ olur. PSO’nun uygulanmasında uygunluk fonksiyonu olarak formülü aşağıda verilen *HKOK* değeri alınır.

$$HKOK = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - y_i)^2} \quad (3.17)$$

Burada d_i ve y_i sırasıyla i . öğrenme örneğine karşılık gelen hedef değeri ve ağın çıktısını göstermektedir.

4. ZAMAN SERİLERİNDE AYKIRI DEĞER DURUMUNDA DAYANIKLI TAHMİN EDİCİLER

Regresyon analizinde temel amaç; örneklem regresyon fonksiyonu temel alınarak anakütle regresyon fonksiyonunun tahmin edilmesi ve verilere en uygun regresyon doğrusunun çizilmesidir. Bunun yanısıra mevcut değişkenler arasındaki iktisadi ve matematiksel ilişkiyi modellemek, yorumlamak ve gelecek tahmini yapmak üzere kullanılan bir yöntemdir. Değişkenler arasındaki bu ilişkiyi belirlemek için doğrusal bir model tahmin edilmektedir. Bu modelin parametrelerini belirlemek için ise çeşitli yöntemler bulunmaktadır. Parametreleri tahmin etmek için kullanılan bu yöntemlere tahmin edici denilmektedir. Parametre tahmini için en sık kullanılan tahmin edici, artıkların karelerinin minimize edilmesi esasına dayanan en küçük kareler (EKK) yöntemi olmaktadır. EKK yöntemi kullanımı basit ve kolay uygulanabilir bir yöntem olmasının yanısıra iyi bir performans gösterebilmesi için bazı varsayımlara ihtiyaç duymaktadır. Ancak gerçek hayat zaman serilerinde her zaman bu varsayımlar sağlanmadığından EKK yönteminin uygulanamadığı durumlar bulunmaktadır. Tahmin yapmada varsayımların sağlanmasının yanında veri yapısı da önemli olmaktadır. Veri setinde bulunan ve diğer gözlem değerlerinden çok küçük ya da çok büyük değerler alan gözlemler aykırı değer olarak adlandırılırlar. Aykırı değerler, verilerin özelliklerini yansıtan gerçek değerler olabilecekleri gibi ölçme, kayıt altına alma, aktarma hatalarından meydana gelmektedir. Varsayımların sağlandığı durumda en iyi yöntem EKK olmasına rağmen, veri setinde aykırı değer ya da değerlerin bulunması ya da hata teriminin dağılımının normal dağılıma uymaması durumunda söz konusu varsayımlardan sapmalar meydana geleceğinden, EKK tahmin edicisi güvenilir sonuçlar veremeyecektir. Aykırı değerlerin sebep olduğu bu sorunun üstesinden gelmek üzere, varsayımlardan sapmalara karşı duyarsız olan tahmin ediciler önerilmiştir. Bu tahmin ediciler dayanıklı tahmin ediciler olarak adlandırılmaktadır. Literatürde birçok dayanıklı tahmin edici türü bulunmaktadır. Dayanıklı tahmin edicilerin temel amacı; eldeki veri topluluğuna en uygun yapının belirlenmesinde rol oynayan parametre değerlerinin tahminini elde etmektir.

Dayanıklı regresyon tahmin edicilerinin üç temel tipi L-tipi tahmin ediciler (Lineer Combination of Order Statistics: Sıra istatistiklerinin lineer kombinasyonları), M-tipi tahmin ediciler (Maksimum Likelihood: En çok olabilirlik tipi), R- tipi tahmin ediciler (Rank Statistics: Sıra istatistikleri) ve yüksek bozulma sınırlı tahmin ediciler olarak adlandırılan en küçük medyan kareler tahmin edicisi (LMS: Least Median Squares), en küçük budanmış kareler tahmin edicisi (LTS: Least Trimmed Squares), en küçük budanmış mutlak değer tahmin edicisi (LTA: Least Trimmed Absolute Value), S tahmin ediciler ve MM tahmin edicilerdir. Tüm koşullar dikkate alındığında, herhangi bir dayanıklı regresyon tahmin edicisine ait performansın her zaman diğerlerinden daha iyi olduğu çıkarımı yapılamaz. Bu nedenle parametre tahmininde kullanılacak olan yöntem ya da yöntemlerin ve özelliklerinin iyi bilinmesi önemlidir. Bununla birlikte yöntemlerin aykırı değerlerin etkisi karşısında bir bozulma sınırı vardır.

Aykırı değerlere karşı dayanıklılığının sayısal olarak bir ölçüsü ya da tahmin edicinin güvenilirliğinin bir ölçüsü bozulma sınırı olarak ifade edilir. “Yüksek bozulma sınırına sahip” olarak isimlendirilen tahmin ediciler ile veri setinde hem x hem de y yönündeki aykırı değer ya da değerlerin varlığı durumunda güvenilir parametre tahminleri elde edilir. Diğer dayanıklı tahmin edicileri yüksek bozulma sınırlı tahmin edicilerden ayıran fark ise y yönündeki aykırı değerlere karşı dayanıklı olmaları yanında x yönündeki aykırı değerlerden etkilenebilmeleridir. Yüksek bozulma sınırı ile ifade edilen kavram ise bozulma sınırının asimptotik olarak mümkün olan en iyi değer olan % 50 sınırına ulaşmasıdır. Burada varsayımların sağlanması durumunda en iyi sonucu veren EKK tahmin edicisi yanında yüksek bozulma sınırına sahip tahmin edicilerden olan LMS, LTS, LTA ve Trimmed Mean (Budanmış Ortalama) tahmin edicilerinden bahsedilecektir.

4.1 En Küçük Kareler

Regresyon analizinde, varsayımların sağlanması durumunda iyi sonuçlar vermesi ve kullanım kolaylığı sebebiyle en küçük kareler (EKK) yöntemi yaygın bir kullanıma sahiptir. EKK yönteminin amacı regresyondan elde edilen artık kareler toplamını minimum yaparak doğruya en yakın tahminleri elde etmektir. X 'in bağımsız değişkeni Y 'nin bağımlı değişkeni ifade ettiği bir regresyon modeli;

$$Y_i = \beta_0 + \beta_1 X_{1i} + \varepsilon_i \quad i = 1, 2, \dots, n$$

olarak ifade edilsin. Bu modelden elde edilen tahmin denklemi;

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + e_i \quad i = 1, 2, \dots, n$$

şeklinde verilir. Burada artıklar $e_i = Y_i - \hat{Y}_i$ olmaktadır. Bu durumda $\hat{\beta}$ parametrelerini elde etmek için kullanılan EKK tahmin edicisi;

$$\min_{\hat{\beta}} \sum_{i=1}^n e_i^2$$

olarak verilebilir. Veride aykırı değer ya da değerler bulunması durumunda, EKK yönteminin işleyiş mantığına ters bir durum olarak, artıkların kareleri toplamı artacak ve parametre tahminleri bu durumdan olumsuz olarak etkilenecektir. EKK veri setindeki tek bir aykırı değerden bile etkilenebilir. Bu nedenle aykırı değerlere karşı dayanıklılığın sayısal bir ölçüsü olan bozulma sınırı $1/n$ olarak ifade edilir. Bu nedenle amaç fonksiyonu olarak varsayımların sağlanmadığı durumlara karşı duyarsız olan bir tahmin edici seçilmelidir. Varsayımların sağlanması durumunda regresyon parametrelerinin tahmininde en iyi sonucu veren yöntem EKK olmaktadır.

4.2 Budanmış Ortalama

Budanmış ortalama (Trimmed Mean), L-tipi tahmin edicilerin bir türü olmakla birlikte literatürde sıklıkla kullanılmaktadır. X_1, X_2, \dots, X_n rasgele örnekler olsun ve gözlemler $X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(n)}$ şeklinde artan sırada yazılmış gözlemler olsun. $X_{(i)}$ değeri i . nci sıra istatistiği olarak adlandırılır. γ , $0 \leq \gamma < 0.5$ 'nın budanmak istenilen miktar olarak seçildiğini varsayalım. $g = [\gamma n]$ ve burada $[\gamma n]$, γn 'in en yakın tamsayıya kadar yuvarlanan değeridir. Budanmış ortalama, g 'nin en büyük ve en küçük gözlemleri çıkarılarak ve kalan değerlerin ortalaması alınarak hesaplanır. Budanmış ortalama;

$$\bar{X}_t = \frac{1}{n - 2(\gamma n)} \sum_{i=(g+1)}^{(n-g)} X_i$$

olmak üzere,

$$\bar{X}_t = \frac{X_{g+1} + \dots + X_{n-g}}{n - 2g}$$

eşitliği ile ifade edilir. Özünde ampirik dağılım bulunurken, μ_t tanımlandığında olasılık yoğunluk fonksiyonu budanma yöntemi ile tutarlı bir biçimde budanır. Olasılık yoğunluk fonksiyonu budanmış ise, artık bir olasılık yoğunluk fonksiyonu olarak nitelendirilemez, çünkü eğrinin altındaki alan 1'e eşit değildir, $1 - 2\gamma$ 'ya eşittir. Bu nedenle eğer budanmış olasılık yoğunluk fonksiyonu $1 - 2\gamma$ ' ya göre bölünürse, ortaya çıkan fonksiyon yine olasılık yoğunluk fonksiyonu olacaktır. Burada iki yönlü budama yapılmaktadır. Ancak bazı araştırmacılar γ budama oranından bahsederken, çift yönlü yerine tek yönlü budama yapıldığını varsayarlar. Burada aksi bir durum belirtilmedikçe iki yönlü budama olduğu varsayılmaktadır. Budanmış ortalamaya ait en yaygın kullanımlarından biri olan tanımlama yukarıdaki eşitlikte verilmiştir. Budanmış ortalama için seçilmesi gereken γ değeri, pratikte büyük bir öneme sahiptir ve uygulamalarda genellikle % 20 olarak seçilir (Wilcox, 2005).

4.3 En Küçük Medyan Kareler

En küçük medyan kareler tahmin edicisi (LMS) Hampel (1975)'te önerilen ve Rousseeuw (1984)'te geliştirilen yüksek bozulma sınırına sahip bir tahmin edicidir. LMS'de amaç, EKK yöntemine benzer bir şekilde artıkların kareleri toplamının yerine artıkların karelerine ait medyan değerini minimize etmektir. LMS tahmin edicisi;

$$\min_{\beta} Med(e_i^2)$$

şeklinde verilir. Yöntemin işleyişi veriden elde edilen tüm altkümelere EKK'nın uygulanması ve sonucunda elde edilen hata terimlerinin her birinin medyan değerinin hesaplanması şeklindedir. Hesaplanan bu medyan değerlerinden minimum değere sahip olan altkümenin tahminleri LMS tahmini olarak kabul edilir. Küçük veriler için bu işlemi yapmak mümkün olmasına rağmen büyük veriler için tüm alt kümelerin ele alınmasında fazla sayıda işlem yapılması gerektirdiğinden hesaplama gücü ortaya çıkacaktır.

LMS yöntemi artıkların karelerini minimize edip, geriye kalan veri noktalarını dikkate almadığından bu sürecin tek başına kullanılması önerilmez. Bununla birlikte LMS, tüm mümkün alt kümelerin parametrelerini tahmin edip hata terimlerini değerlendirdiğinden pratik bir hesaplama sahip değildir. Diğer bir

olumsuz özelliği de hata terimlerinin normal dağılıma sahip olduğu durumda bile düşük asimptotik etkinliğe sahip olmasıdır. Bu olumsuz özelliklerine rağmen verideki aykırı değerlere % 50'ye kadar dayanabilmesi nedeniyle yüksek bozulma sınırına sahip bir tahmin edicidir ve aynı zamanda aykırı değerlerin teşhisinde kullanılan bir araçtır.

4.4 En Küçük Budanmış Kareler

Yüksek bozulma sınırına sahip en küçük budanmış kareler tahmin edicisi (LTS) Rousseeuw (1984)'te önerilmiştir. Buradaki budama sözcüğü, veri setindeki şüphelenilen gözlemlerin değişkenler üzerindeki etkisinin kaldırılması anlamında kullanılmaktadır. En küçük budanmış kareler yönteminde artıklara EKK yöntemindekine benzer bir minimizasyon işlemi uygulanmaktadır. Sıralanmış artıkların en büyük değerleri budandıktan sonra yapılan işlem EKK ile aynıdır. $e_t = y_t - \hat{y}_t$ olmak üzere; $e_1^2, e_2^2, \dots, e_n^2$ yöntemden elde edilen artıkların kareleri olsun. Burada ilk olarak hata değerlerinin kareleri küçükten büyüğe doğru sıralanır;

$$e_{(1)}^2 \leq e_{(2)}^2 \leq \dots \leq e_{(n)}^2$$

h terimi toplama dahil edilecek olan artık karelerinin sayısı olmakla birlikte, h değerinin belirlenmesi tahmin performansı açısından önemlidir. Rousseeuw (1984)'te tanımlanan LTS tahmin edicisi;

$$\min_{\hat{\beta}} \sum_{i=1}^h (e_i^2)$$

olarak gösterilir. Burada en büyük değere sahip $(n-h)$ gözlemin budanması ile birlikte geriye kalan h gözleme uygulanan yöntem temelde EKK yöntemi olmaktadır. LTS tahmin edicisinin güvenilirliğinin bir ölçüsü olan bozulma sınırı % 50'dir ve verinin $n/2$ 'sinin aykırı değer olması durumunda bununla başa çıkabileceği anlamına gelmektedir. LTS'nin dayanıklılığı, çoklu regresyon durumunda bağımsız değişkenlerin sayısından etkilenmediğinden güvenilir bir veri tahmin aracıdır. p parametre sayısı olmak üzere $p > 1$ iken $h = [n/2] + 1$ alındığında, LTS tahmin edicisi $([n/2] - p + 2)$ bozulma sınırına sahiptir. Ayrıca $h = [n/2] + [(p+1)/2]$ olduğunda LTS yönteminin bozulma sınırı mümkün olan maksimum değerine ulaşır. Her zaman LTS tahmin edicisi için bir çözüm mevcuttur. Bu tahmin edicinin en büyük dezavantajı ise

işlemlerde artıkların karelerinin sıralanması nedeniyle uzayan işlem süresidir (Rousseuw ve Leroy, 1987).

4.5 En Küçük Budanmış Mutlak Değer

En küçük budanmış mutlak değer tahmin edicisi (LTA), LTS tahmin edicisinin yakın bir türüdür. Buradaki amaç;

$$\min_{\hat{\beta}} \sum_{i=1}^h |e|_{(i)}$$

eşitliğini minimize edecek sabit ve eğim parametrelerini seçmektir. Burada $|e|_{(1)} \leq |e|_{(2)} \leq \dots \leq |e|_{(n)}$ ve $|e|_{(i)}$, i . nci en küçük mutlak artık değeridir. h değeri ise toplama dahil edilen mutlak artık değeridir ve LTS'de tanımlandığı gibidir. LTA tahmin edicisi LMS ve LTS tahmin edicilerinde olduğu gibi yüksek bozulma sınırına yani %50 dayanıklılığa sahiptir. LTA tahmin edicisinin bulunmasında h değerine karar verilmesi önemli bir konudur. LTA tahmin edicisi özellikle büyük veri kümeleri kullanılarak yapılan hesaplamalarda kullanım kolaylığı sebebiyle LMS ve LTS tahmin edicilerine cazip bir alternatiftir (Wilcox, 2005).

5. BOOTSTRAP YÖNTEMİ

Bootstrap yöntemi, küçük bir veri seti kullanılarak bu veri setinden daha büyük bir veri seti elde etmek üzere, çok sayıda yeni veri setinin elde edilmesini sağlayan bir yeniden örnekleme yöntemidir. Başka bir ifade ile anakütleden seçilen bir örneklemden tekrarlı olarak alt örneklemlerin seçilmesi olarak da tanımlanabilir.

Bootstrap yöntemi ile elde edilen bu yeni veri setleri üzerinde güven aralıkları tahmin edilebilmekte ve veri setleri için özel bir dağılım varsayımına ihtiyaç duyulmadığından hipotez testi için de kullanılabilir. İlk olarak Efron (1979) tarafından ortaya atılan yöntem istatistiksel yöntemler yetersiz olduğunda ya da hata terimi ile ilgili parametrik varsayımlar sağlanamadığında bu durumdan etkilenmeden güvenilir tahmin sonuçları üretebilmektedir.

Klasik bootstrap yönteminin kullanımında zaman serisi verisi ele alındığında yöntemin kullanılabilmesi için veri yapısının bağımsız ve özdeş dağılımlı olması gerekmektedir. Bununla birlikte zaman serilerinde veri içi bağımlılık yapısına sahip olan seriler de bulunmaktadır. Bu bağımlılık yapısının yeniden örnekleme ile elde edilen yeni veri setlerine yansıtılmadığı (aktarılamadığı) durumlarda klasik bootstrap yöntemi uygulanamamaktadır. Bu nedenle istatistiksel çıkarımların yapılabilmesi için yöntemde değişiklikler yapılması gerekmektedir. Yapılan bu değişikliklere göre bootstrap yöntemi çeşitlenmekte ve farklı isimler almaktadır.

5.1 Bağımlı Veri Yapısı için Bootstrap

Bootstrap yöntemi, geleneksel yöntemlerin başarısız olduğu durumlarda sunulan bir çözüm yöntemidir. Ancak klasik bootstrap yaklaşımında zaman serilerinde veri içi bağımlılık yapısı tasarlanmamıştır. Yeniden örnekleme esnasında veri setindeki bağımlılık yapısının elde edilen veri setlerinde de tutulması gerekmektedir. Literatürde bağımlı veri yapısı için yeniden örnekleme yöntemleri geliştirilmiştir. Bağımlılık yapısını yansıtabilecek bu yeniden örnekleme yöntemlerinden birisi de blok bootstrap yöntemleridir. Bu yöntemlerin çoğunda, blokları tanımlarken veri

parçalarını dikkate aldığından, bağımlılık yapısı her blok içerisinde tutulabilir. Durağan bir süreçten elde edilen gözlemler;

$$X = \{X_1, X_2, \dots, X_n\}$$

olsun. Blokların uzunluğu ve sayısı sırasıyla $l \in \{1, 2, \dots, n\}$ ve $b \geq 1$ ile simgelensin. Öyle ki $l \times b \leq n$ olsun. n ve m başlangıçtaki veri büyüklüğü ve bootstrap örnek büyüklüğü, $m \leq n$ ve k seçilen blokların sayısı olsun.

5.1.1 Blok bootstrap

Bağımlılık yapısına sahip veriler için yeniden örnekleme yöntemleri; örtüşmeyen blok bootstrap (Nonoverlapping Block Bootstrap) Carlstein (1992), hareketli blok bootstrap (Moving Block Bootstrap) Künsch (1989), dairesel blok bootstrap (Circular Block Bootstrap) ve durağan blok bootstrap (Stationary Block Bootstrap) Politis ve Romano (1992; 1994) ve durağan süreçlerin büyük bir sınıfı için Bühlmann (1997)'de sieve bootstrap yöntemi olarak sunulmuştur.

5.1.1.1 Örtüşmeyen blok bootstrap

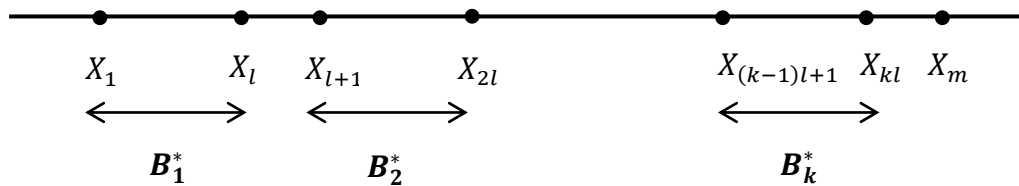
Örtüşmeyen blok bootstrap yöntemi zaman serilerinin ardışık gözlemlerinin b bloğa ayrılmasından oluşmaktadır.

$$B_i = (X_{(i-1)l+1}, \dots, X_{il}) \quad i = 1, \dots, b$$

k tane bloğun rasgele bir örneği olan, $k \geq 1$, B_1^*, \dots, B_k^* örneği $\{B_1, \dots, B_b\}$ 'den yer değiştirme ile seçilmektedir. k bloğun birleşmesi ile oluşan bootstrap örneği ise $m = k \times l$ gözlem ile oluşturulur. Böylece elde edilen örnek;

$$(X_1^*, \dots, X_l^*, \dots, X_{(k-1)l+1}^*, \dots, X_m^*)$$

olmaktadır. Bu bootstrap yöntemi için korelasyon, gözlemlerin blokları içinde güçlü ve bloklar arasında oldukça zayıf olarak düşünülür (Cordeiro ve diğ., 2006). Yöntem yardımı ile blokların elde edilişi Şekil 5.1'de verilmiştir.



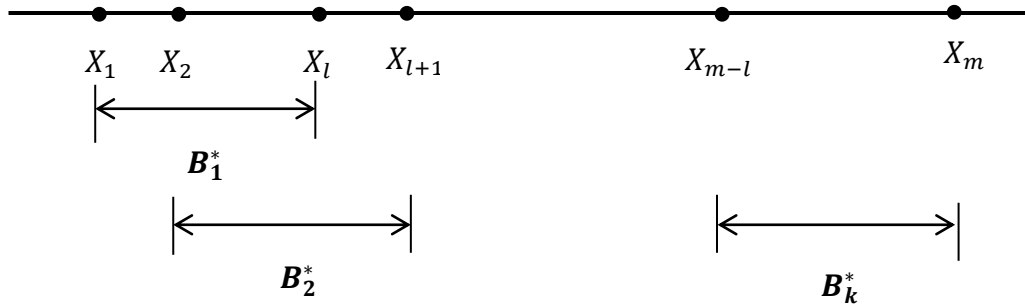
Şekil 5.1. Örtüşmeyen blok bootstrap yöntemi için blokların elde edilişi (Lahiri, 2003)

5.1.1.2 Hareketli blok bootstrap

Hareketli blok bootstrap yöntemi, Künsch (1989) ve Liu ve Singh (1992)'de önerilmiştir. Yöntem temelde bir zaman serisindeki ardışık gözlemlerin bloklarının yeniden örneklenmesi işlemidir. Buna bağlı olarak, orijinal gözlemlerin bağımlılık yapısı her blok içerisinde korunmaktadır.

$$B_i = (X_i, \dots, X_{i+l-1}) \quad 1 \leq i \leq b$$

B_i , X_i ile başlayan l uzunluğundaki bloktur. Burada $b = n - l + 1$ 'dir. Hareketli blok bootstrap örneği $\{B_1, \dots, B_b\}$ koleksiyonundan uygun bir blok sayısının rasgele olarak seçilmesi ile elde edilir. Buna göre, $\{B_1, \dots, B_b\}$ 'den yer değiştirme ile çekilmiş basit rasgele bir örnek B_1^*, \dots, B_k^* olarak gösterilebilir. B_i^* 'da $(X_{(i-1)l+1}^*, \dots, X_{il}^*)$, $i = 1, \dots, k$ ile gösterilen elemanlar, X_1^*, \dots, X_m^* , $m = k \times l$ büyüklüğünde olan hareketli blok bootstrap örneğidir. Efron (1979)'un klasik bootstrap yöntemi, her bloğun tek bir elemandan oluştuğu ($l = 1$) özel durumudur. Hareketli blok bootstrap yöntemi ardışık gözlemlenme durumunu dikkate aldığından, örneklem çekilirken her defasında k kez tek bir gözlemin yeniden örneklenmesi yerine, rasgele yapılan her çekimde ardışık gözlem bloklarında gözlemlerin bağımlılık yapısını da kaybetmeden yeniden örnekleme yapar (Cordeiro ve diğ., 2006). Yöntem yardımı ile blokların elde edilişi Şekil 5.2'de verilmiştir.



Şekil 5.2. Hareketli blok bootstrap yöntemi için blokların elde edilişi (Lahiri, 2003)

5.1.1.3 Dairesel blok bootstrap

Bu bootstrap işlemi önceki diğer yöntemlerin bir uzantısıdır. Fikir, “dairesele olarak tanımlanan” verileri kullanarak, bir daire etrafındaki verileri sarmak ve ek bloklar oluşturmak içindir. Böylece $i > n$ için $X_i = X_{in}$ olarak tanımlanır. Burada $i_n = imodn$ ve $X_0 = X_n$ 'dir. Politis ve Romano (1992)'de önerilen dairesele blok bootstrap yöntemi üst üste ve periyodik olarak blok uzunluğu l 'nin genişletilerek yeniden örneklenmesidir. Dikkat edilmelidir ki her X_i , blokların koleksiyonunda tam olarak l kez görünür ve dairesele blok bootstraptan dolayı eşit olasılıkla bu koleksiyonlardan bloklar yeniden örneklenir. Orijinal gözlemler X_1, \dots, X_n 'den her biri dairesele blok bootstrap altında eşit ağırlık alırlar. Bu özellik dairesele blok bootstrap yöntemini uç etkilerinden muzdarip olan örtüşmeyen blok bootstrap ve hareketli blok bootstrap yöntemlerinden ayırır. Önceki yöntemler gibi X_1^*, \dots, X_m^* gözlemlerinin bir dizisinde l uzunluğunda k blok seçilir ve düzenlenir (Cordeiro ve diğ., 2006).

5.1.1.4 Durağan blok bootstrap

Durağan blok bootstrap sabit bir l uzunluğunda bloklar yerine, rasgele uzunlukta bloklar kullandığı için diğer blok bootstrap yöntemlerinden farklıdır. Politis ve Romano (1994) formülasyonu kullanılarak, bu yöntem şu şekilde tanımlanabilir: X_1^* , $\{X_1, \dots, X_n\}$ 'den rasgele çekilmiş olsun. Yeni gözlem X_2^* 'ın seçimi için, rasgele seçim kolaylaştırılmıştır ve “başarı” olasılığı p olan ikili bir deney yapılır. Eğer ikili deney sonucu “başarı” ise o zaman X_2^* , $\{X_1, \dots, X_n\}$ 'den rasgele olarak tekrar seçilir. Aksi takdirde düzenli aralıklarla genişletilmiş seride X_1^* 'den sonraki gözlem X_2^* olarak belirlenir. Ve benzer şekilde bootstrap örneği n gözlem içerene kadar işleme devam edilir.

Durağan blok bootstrap yönteminin önemli bir özelliği ise $\{X_i^*\}$ $i \in 1, 2, \dots, n$ bootstrap gözlemlerinin durağan olmasıdır ve “durağan” olarak adlandırılmasının sebebi de budur (Cordeiro ve diğ., 2006).

5.1.2 Sieve bootstrap

Bühlmann (1997)'de sieve bootstrap olarak adlandırılan yeni bir bootstrap düzeni önerilmiştir. Bu yöntem önce uygun parametrik modelin bulunması ve elde edilen artıklardan yeniden örnekleme yapılması fikrine dayanır. Ancak model, uyarlamalı

yerine ön eklemeli model dikkate alınarak seçilir. Bu yaklaşım önceki yöntemlerden farklıdır, örnek bootstrap (şartlı olarak) durağanlık ve bağımlılık yapısı sunmaz. Diğer bir farklı özellik, sieve bootstrap örneği önceki yöntemlerde olduğu gibi orijinal verilerin bir altkütmesi değildir. Verilen bir örnek X_1^*, \dots, X_m^* , bir otoregresif süreç tahminidir. Ayrıca artıklar ortalanmış ve bu artıkların ampirik kümülatif dağılım fonksiyonu elde edilmiştir. Bu dağılımdan bağımsız ve özdeş dağılımlı artıkların yeniden örnekleme alınır. Böylece bootstrap hata serileri ve sonrasında bootstrap serileri üretilir. Sieve bootstrap parametrik bir modele dayansa bile temel olarak özünde parametrik olmadığı görülmektedir. Burada AR(p) modeli sadece artıkların serilerini filtrelemek için kullanılır. Zagdanski (1999) sieve bootstrapın farklı uygulamalarını dikkate almıştır. Durağan zaman serilerinin gelecek gözlemleri için güven aralıklarını elde etmiş ve zaman serilerinin gelecek değeri için en iyi doğrusal tahmin edicinin yapısını tanımlamıştır. Sieve bootstrap, gözlemlenen verinin gelecek değerlerinin koşullu dağılımına ait tutarlı tahmin edicilerini elde etmek için ortaya konmuştur (Cordeiro ve diğ., 2006).

6. ÖNERİLEN YÖNTEM VE ALGORİTMASI

Zaman serileri analizinde öngörü, karar verme sürecinin önemli bir unsurudur. İyi öngörü elde etmek ise doğru analiz yöntemini seçmek, öngörü hatasını küçültmek ve süreci hızlandırmak ile mümkün olabilmektedir. Geleneksel zaman serisi analiz yöntemleri ile iyi öngörüler elde edilebilmesine karşın, sağlamaları gereken varsayımlar sebebiyle işlemler karmaşıklaşmakta ve süreç uzamaktadır. Bununla birlikte zaman serileri analizinde sıklıkla kullanılmaya başlanan yapay sinir ağı yöntemleri, iyi öngörü performansları ve basit kullanımları sebebiyle bu yöntemlere tercih edilmektedir. Geleneksel yöntemler yapay sinir ağlarının tersine birçok varsayım gerektirmesi ve bunların çoğunlukla sağlanamaması dolayısıyla iyi öngörü elde etmede yetersiz kalmaktadır. Literatürdeki uygulamalı çalışmalarda da, yapay sinir ağları yaklaşımlarının geleneksel zaman serisi yöntemlerine göre daha iyi sonuçlar verebildiği görülmektedir. Bu nedendir ki son yıllarda zaman serisi öngörü probleminin çözümü için farklı türde yapay sinir ağları önerilmekte ve kullanımları gitgide artmaktadır. Literatüre bakıldığında yapay sinir ağı türlerinden en yaygın kullanıma sahip olanının çok katmanlı algılayıcı yapay sinir ağları (ÇKA-YSA) olduğu görülmektedir.

ÇKA-YSA yaygın kullanımı ve başarılı öngörü sonuçları vermesinin yanı sıra bazı temel problemleri barındırmaktadır. Bunlardan en önemlisi ağı mimari yapı belirleme problemidir. Öngörü için YSA'nın kullanıldığı problemlerde, girdi ve çıktı katmanlarında bulunacak birim sayısı problemin bilinen özellikleri sayesinde belirlenebilir. Ancak gizli katmanda bulunması gereken birim sayısının belirlenmesi araştırmacının vereceği bir karara bağlı olmaktadır. Bu durumda da seçilen gizli katman birim sayısına göre ağı performansında önemli değişiklikler olmaktadır. Bu durumun önüne geçmek amacıyla Yadav ve diğ. (2007) çalışmalarında tek bir nörondan oluşması sebebiyle mimari belirleme problemine sahip olmayan çarpımsal nöron model yapay sinir ağı (ÇNM-YSA)'nı önermişlerdir. Önerilen ÇNM-YSA'nın, ÇKA-YSA'dan farkı tek bir nörona sahip olması (gizli katman içermemesi) ve birleştirme fonksiyonu olarak toplamsal yerine çarpımsal birleştirme fonksiyonu kullanmasıdır. ÇNM-YSA'nın bu özellikleri (farklılıkları) sayesinde diğer YSA

türlerine göre üstün öngörü performansı gösterebildiği yapılan çeşitli çalışmalar yardımı ile kanıtlanmıştır.

İlgilenilen veri setinde aykırı değerlerin var olması, geleneksel zaman serisi analiz yöntemlerinde olduğu gibi YSA modellerinde de öngörü performansını olumsuz olarak etkileyen bir durumdur. ÇNM-YSA, ÇKA-YSA'ya göre aykırı değerlerden daha fazla etkilenmektedir. ÇNM-YSA'daki bu duyarlılığın nedeni, çarpımsal birleştirme fonksiyonu kullanılması ve çarpma işleminin özelliğinden dolayı aykırı değerlerin, ağıın çıktısı ve dolayısı ile ağıın öngörü performansı üzerindeki olumsuz etkisinin oldukça fazla görülmesidir.

Literatürde ÇKA-YSA'nın aykırı değere olan duyarlılığını ortadan kaldırmaya yönelik farklı dayanıklı öğrenme algoritmaları geliştirilmiştir. Ancak ÇNM-YSA aykırı değerden ÇKA-YSA'ya göre daha fazla etkilenmesine karşın herhangi bir dayanıklı öğrenme algoritması bulunmamaktadır. Dayanıklı bir öğrenme algoritmasının kullanılması ile birlikte ÇNM-YSA'nın aykırı değerlere olan duyarlılığı ortadan kaldırılacak; mimari seçim problemi olmayan, kullanımı basit ve öngörü performansı aykırı değer durumunda bile iyi olabilecek bir yapay sinir ağı ortaya konulmuş olacaktır.

Bu tez kapsamında ÇNM-YSA için aykırı değer tespiti yapmadan da çalışabilecek (aykırı değer durumunda) dayanıklı bir öğrenme algoritması önerilmiştir. Önerilen yöntemde ÇNM-YSA'nın eğitiminde uygunluk fonksiyonu olarak budanmış ortalamayı kullanan garanti yakınsamalı parçacık sürü optimizasyonu (GRY-PSO) kullanılmıştır. Esas olarak bu yöntemde regresyon analizinde kullanılan ve yüksek bozulma sınırına sahip dayanıklı bir yöntem olan en küçük budanmış kareler (LTS: Least Trimmed Squares) yaklaşımı ÇNM-YSA'ya uyarlanarak ağıın eğitiminde kullanılmıştır.

Önerilen dayanıklı yöntem sayesinde ÇNM-YSA'nın hem x hem de y yönündeki aykırı değerlerin olumsuz etkilerinden arındırılması sağlanacak ve ağıın öngörü performansı artırılabilecektir. Aynı zamanda yapay sinir ağlarında istatistiksel değerlendirme yapmanın mümkün olduğunu göstermek üzere önerilen yöntem için bootstrap tekniği kullanılarak hipotez testi yapılmıştır.

Önerilen yöntemin algoritması adımlar halinde aşağıda verilmektedir.

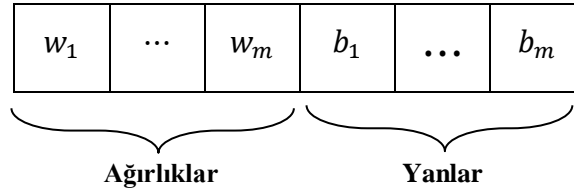
6.1 Önerilen Yöntemin Algoritması

Adım 1. Garanti yakınsamalı parçacık sürü optimizasyonunda (GRY-PSO) kullanılacak olan parametreler ve değerleri belirlenir.

Bunlar; pn parçacık sayısı, c_1 bilişsel katsayı, c_2 sosyal katsayı, w eylemsizlik katsayısı ve s_c başarı sayısı için, f_c ise başarısızlık sayısı için üst limit olarak verilebilir.

Adım 2. Parçacıkların başlangıç hız ve pozisyon vektörleri rastgele üretilir.

Parçacıkların pozisyonları ÇNM-YSA'daki ağırlıklar ve yanlardır. Örneğin m girdinin olduğu ÇNM-YSA'da bir parçacığın pozisyonları Şekil 6.1'de verilmiştir. Şekil 6.1'de görüldüğü gibi m girdinin olduğu bir ÇNM-YSA'da bir parçacığın pozisyon sayısı $2m$ olmaktadır. Parçacığın tüm başlangıç pozisyonları sürekli düzgün dağılımdan $U(0,1)$ aralığından rastgele üretilmektedir. Hızlar ise önceden hızlar için belirlenen $vmaps$ sınır değerine göre $(-vmaps, vmaps)$ aralığından rastgele üretilmektedir.



Şekil 6.1. ÇNM-YSA'da bir parçacığın pozisyonları

$p_{ij}^{(k)}$, k . iterasyonda i . parçacığın j . pozisyon değeri ve $v_{ij}^{(k)}$, k . iterasyonda i . parçacığın j . hız değerini göstermek üzere, k . iterasyondaki sürü pozisyon ve hız değerleri sırasıyla Çizelge 6.1 ve Çizelge 6.2'deki gibi gösterilebilir.

Çizelge 6.1. Parçacıklara ait pozisyon değerleri

| | | | |
|-------------------|-------------------|-----|--------------------|
| $p_{11}^{(k)}$ | $p_{12}^{(k)}$ | ... | $p_{1\ 2m}^{(k)}$ |
| $p_{21}^{(k)}$ | $p_{22}^{(k)}$ | ... | $p_{2\ 2m}^{(k)}$ |
| $p_{pn\ 1}^{(k)}$ | $p_{pn\ 2}^{(k)}$ | ... | $p_{pn\ 2m}^{(k)}$ |

Çizelge 6.2. Parçacıklara ait hız değerleri

| | | | |
|-------------------|-------------------|-----|--------------------|
| $v_{11}^{(k)}$ | $v_{12}^{(k)}$ | ... | $v_{1\ 2m}^{(k)}$ |
| $v_{21}^{(k)}$ | $v_{22}^{(k)}$ | ... | $v_{2\ 2m}^{(k)}$ |
| $v_{pn\ 1}^{(k)}$ | $v_{pn\ 2}^{(k)}$ | ... | $v_{pn\ 2m}^{(k)}$ |

Adım 3. Her bir parçacık için uygunluk fonksiyonu değeri hesaplanır.

İterasyon sayısı k olduğunda, i . parçacık için uygunluk fonksiyonu değerinin hesaplanması Adım 3.1 ile Adım 3.3 arasında verilmiştir.

Adım 3.1. k . iterasyonda i . parçacığın $2 \times m$ adet pozisyonundan çarpımsal nöron modelin ağırlık ve yanları (6.1) ve (6.2) eşitliklerindeki gibi elde edilmektedir.

$$w_1 = p_{i1}^{(k)}, w_2 = p_{i2}^{(k)}, \dots, w_m = p_{im}^{(k)} \quad (6.1)$$

$$b_1 = p_{i\ m+1}^{(k)}, b_2 = p_{i\ m+2}^{(k)}, \dots, b_m = p_{i\ 2m}^{(k)} \quad (6.2)$$

Adım 3.2. Ağırlık ve yan değerlerini kullanarak ÇNM-YSA'nın her bir öğrenme örneği için net değerleri (6.3) eşitliğindeki gibi hesaplanır.

$$net_t = \prod_{j=1}^m w_j x_{t-j} + b_j, \quad t = 1, \dots, n \quad (6.3)$$

Burada n adet öğrenme örneği bulunmaktadır. Zaman serisinin N adet gözlemi varsa $n = N - m$ olmaktadır. Ağırlık çıktıları ise (6.4) eşitliğindeki gibi hesaplanmaktadır.

$$\hat{y}_t = \frac{1}{1 + \exp(-net_t)}, \quad t = 1, 2, \dots, n \quad (6.4)$$

Adım 3.3. Her bir parçacık için uygunluk fonksiyonu değeri hesaplanır.

Uygunluk fonksiyonu olarak ÇNM-YSA'dan öğrenme örnekleri için hesaplanan çıktılar kullanılarak (6.5) eşitliğinde verildiği gibi bir ölçüt hesaplanır.

$e_t = y_t - \hat{y}_t$ olmak üzere; $e_1^2, e_2^2, \dots, e_n^2$ yöntemden elde edilen artıkların kareleri olsun. Artık kareler $e_{(1)}^2 \leq e_{(2)}^2 \leq \dots \leq e_{(bm)}^2 \leq e_{(m+1)}^2 \leq \dots \leq e_{(n)}^2$ şeklinde sıralanmış olarak verilsin. Burada \hat{y}_t ÇNM-YSA'nın çıktısı, y_t hedef değeri, n öğrenme örneklerinin sayısı ve α - budama oranı olmak üzere; toplama dahil edilen artıkların karelerinin sayısı $bm = n - \lceil n\alpha \rceil$ olmaktadır.

Bu durumda j . nci öğrenme örneği için uygunluk fonksiyonu değeri;

$$f_{uygunluk_j} = \frac{1}{bm} \sum_{i=1}^{bm} e_{(i)}^2, \quad j = 1, \dots, n \quad (6.5)$$

şeklinde hesaplanmaktadır.

Budama yapılırken; artıkların kareleri arasında bir sıralama yapıldığından ve küçük artık değerlerinin kareleri sifıra yaklaşacağından, yalnızca sıralamanın üst kısmından budama yapılmaktadır. Bu nedenle yaptığımız işlem esasen tek taraflı budama olmaktadır.

Adım 4. $pbest$ ve $gbest$ değerleri oluşturulur ya da güncellenir.

$pbest$ ilk iterasyonda başlangıç pozisyonlarının kendisidir. İlk iterasyondan sonra her bir parçacığın en iyi durumunun saklandığı vektörlerden oluşmaktadır. Bir parçacığın pozisyon değerleri güncellendikten sonra daha iyi bir uygunluk değerine sahip olursa (uygunluk değeri daha küçük olur ise) bu parçacığın $pbest$ 'teki değerleri güncellenen pozisyon değerleri ile değiştirilir. $pbest_{ij}$, i . parçacığın en iyi durumu için j . pozisyondaki değeridir. $gbest_j$ tüm parçacıkların en iyi durumu için j . pozisyondaki değeridir.

Eğer $gbest$ 'in uygunluk değeri önceden belirlenen bir küçük pozitif sayı olan ϵ değerinin altındaysa ya da maksimum iterasyon sayısına ulaşılmış ise işlemler durdurulur. Aksi halde Adım 5'e geçilir. Eğer algoritma durdurulur ise optimal ağırlık ve yan değerleri $gbest$ 'ten alınır.

$pbest$ ve $gbest$ vektörleri Çizelge 6.3 ve Çizelge 6.4'teki gibi gösterilir.

Çizelge 6.3. Parçacıklara ait $pbest$ vektörleri

| | | | |
|---------------------|---------------------|-----|----------------------|
| $pbest_{11}^{(k)}$ | $pbest_{12}^{(k)}$ | ... | $pbest_{12m}^{(k)}$ |
| $pbest_{21}^{(k)}$ | $pbest_{22}^{(k)}$ | ... | $pbest_{22m}^{(k)}$ |
| $pbest_{pn1}^{(k)}$ | $pbest_{pn2}^{(k)}$ | ... | $pbest_{pn2m}^{(k)}$ |

Çizelge 6.4. Parçacıklara ait $gbest$ vektörleri

| | | | |
|-----------------|-----------------|-----|--------------------|
| $gbest_1^{(k)}$ | $gbest_2^{(k)}$ | ... | $gbest_{2m}^{(k)}$ |
|-----------------|-----------------|-----|--------------------|

Adım 5. Pozisyonların hız değerleri ve pozisyonlar sırasıyla (6.7) ve (6.8) eşitliklerinde verilen formüllere göre güncellenirken $gbest$ 'e karşılık gelen parçacığın pozisyon değerleri (6.6) ve (6.8) eşitliğine göre güncellenir ve Adım 3'e geri dönlür.

$$v_{ij}^{(k+1)} = w * v_{ij}^{(k)} - p_{ij}^{(k)} + gbest_j^{(k)} + \rho(k) * r_3 \quad (6.6)$$

$$v_{ij}^{(k+1)} = w \times v_{ij}^{(k)} + c_1 r_1 (pbest_{ij} - p_{ij}) + c_2 r_2 (gbest_{ij} - p_{ij}) \quad (6.7)$$

$$p_{ij}^{(k+1)} = p_{ij}^{(k)} + v_{ij}^{(k+1)} \quad (6.8)$$

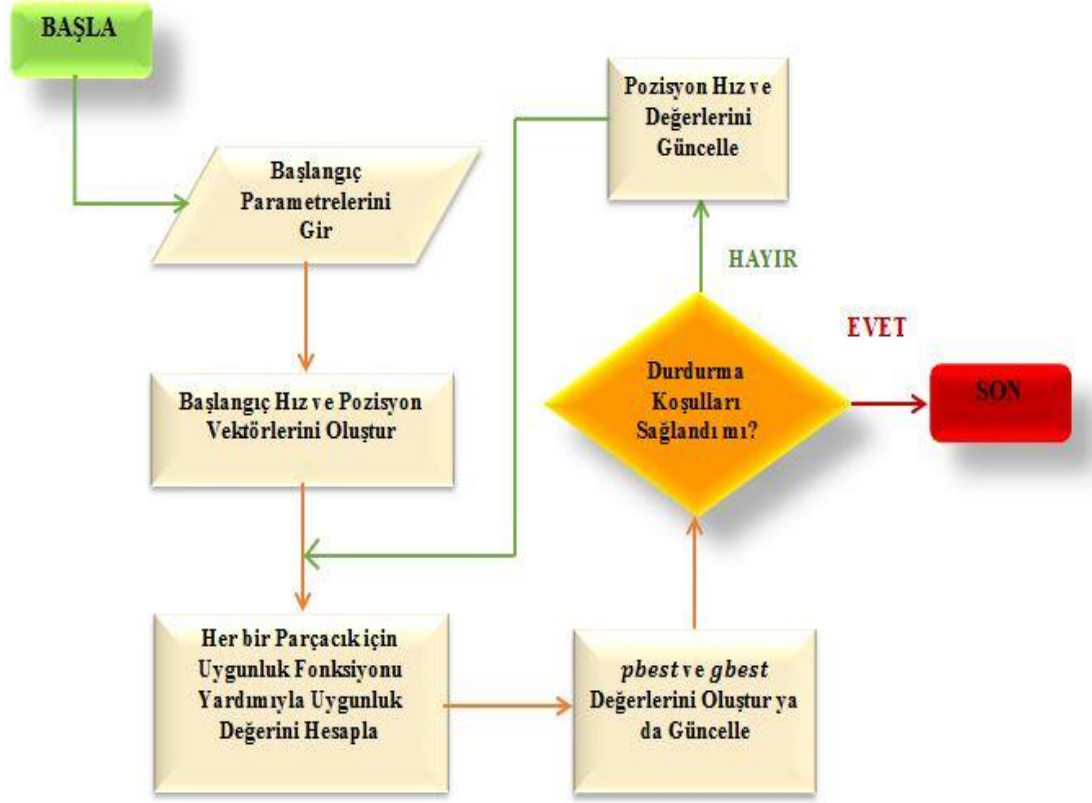
Burada r_1 ve r_2 (0,1) parametrelili düzgün dağılımdan üretilen rastgele sayılar ve r_3 , (-1,1) parametrelili düzgün dağılımdan üretilen rastgele sayıdır. $\rho(k)$ ise (6.9) eşitliğine göre her iterasyonda hesaplanır.

$$\rho(k+1) = \begin{cases} 2\rho(k) , & s_n > s_c \\ 0.5\rho(k) , & f_n > f_c \\ \rho(k) , & \text{aksi halde} \end{cases} \quad (6.9)$$

bu formülde $\rho(0) = 1$ olmakta, s_n başarı sayısı, f_n başarısızlık sayısı ve s_c başarı sayısı için, f_c ise başarısızlık sayısı için üst limittir. Eğer en iyi parçacığın numarası değişmeden, iyileşme sağlar ise başarı sayısı artırılır ve başarısızlık sayısı sıfırlanır, iyileşme sağlanmadıysa başarısızlık sayısı artırılır ve başarı sayısı sıfırlanır. Eğer $gbest$, farklı bir parçacık numarası alırsa, hem başarı hem de başarısızlık sayısı sıfırlanır.

6.2 Önerilen Yöntemin Akış Diyagramı

Önerilen yöntemin işlem adımları Şekil 6.2’de akış diyagramında verildiği gibidir.



Şekil 6.2. Önerilen yöntem için akış diyagramı

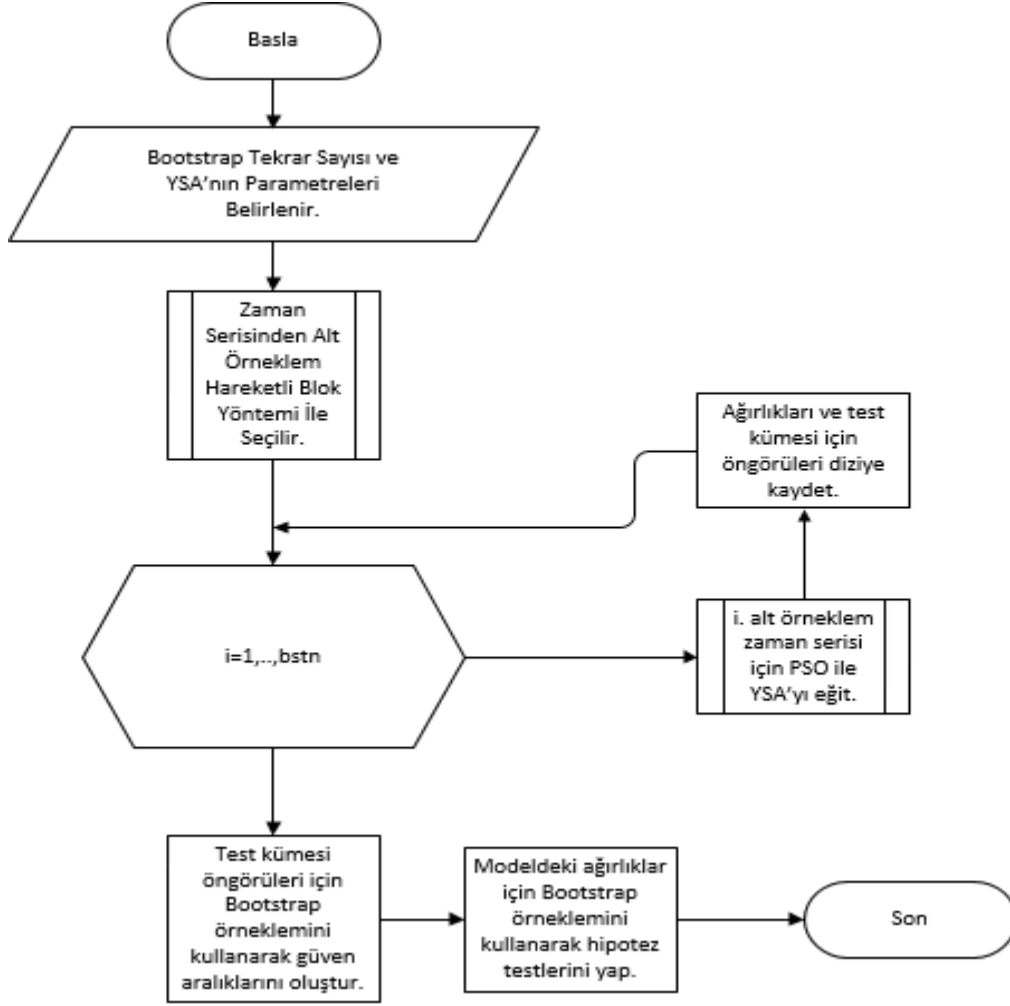
6.3 Önerilen Yöntem için Rastgele Hareketli Kayan Blok Bootstrap

Tiwari ve diğ. (2010a, 2010b) çalışmalarında bootstrap yöntemleri kullanarak YSA’dan elde edilen öngörüler için güven aralıkları elde etmeyi başarmıştır. Ancak bu çalışmalarda zaman serisinin ardışık gözlemlenme durumunu göz ardı eden bootstrap yöntem kullanılmış ve model girdilerinin istatistiksel olarak anlamlı katkılarının olup olmadığı konusunda herhangi bir araştırma yapılmamıştır.

Tiwari ve diğ. (2010a, 2010b) çalışmaları yapay sinir ağlarından olasılıksal çıkarımlar yapmanın mümkün olduğunu göstermektedir.

Tezin bu kısmında çarpımsal nöron model yapay sinir ağında öngörüler için güven aralıklarının elde edilmesi ve modeldeki parametreler olan ağırlıklar için hipotez testlerinin yapılması rastgele hareketli kayan blok bootstrap yöntemi ile

gerçekleştirilmiştir. Önerilen yöntem için rastgele hareketli kayan blok bootstrap'a ait akış diyagramı Şekil 6.3'te verilmiştir.



Şekil 6.3. Rastgele hareketli kayan blok bootstrap yönteminin akış diyagramı

Akış diyagramında bir bütün olarak verilen yöntemin alt işlemleri aşağıda açıklanmaya çalışılmıştır. İlk olarak zaman serisinden rastgele hareketli kayan blok bootstrap yöntemi ile nasıl alt örneklem zaman serilerinin oluşturulduğunu açıklayalım.

Zaman serisinin gözlem sayısı n , test kümesi uzunluğu ise n_{test} olsun. Alt örneklem blok olarak ilk $n - n_{test}$ gözlemden seçilir. Seçim işlemi için ilk olarak alt örneklem uzunluğu (n_{alt}), $U(a, b)$ dağılımından rastgele seçilir. Burada,

$$a = \lfloor (n - n_{test}) \times 0.5 \rfloor, \quad b = \lfloor (n - n_{test}) \times 0.75 \rfloor$$

olmaktadır. Alt örneklem uzunluğu seçildikten sonra alt örneklem başlangıç noktası (n_0), $U(c, d)$ aralığından rastgele olarak seçilen sayının tam kısmına bir eklenerek

elde edilir. Burada, $c = 1$ ve $d = n - ntest - nalt$ olmaktadır. Böylece x_t zaman serisinden alt örneklem zaman serisi

$$x_t^{alt} = x_{t+n0-1}, t = 1, 2, \dots, nalt$$

olarak elde edilir. Elde edilen her bir alt zaman serisi için algoritması yukarıda verilen önerilen yöntem uygulanarak YSA eğitilir.

Akış diyagramında verilen test kümesi verileri için güven aralıklarının oluşturulması test kümesi verileri için elde edilen örneklemelere dayalı olarak gerçekleştirilir. Her bir alt örneklem zaman serisi için eğitilen ağ kullanılarak hesaplanan öngörü değerleri iki boyutlu bir diziye kaydedilir.

$$\{\hat{Y}_{i,j}\}, \quad i = 1, 2, \dots, bstn; j = 1, 2, \dots, ntest$$

Burada $bstn$ bootstrap tekrarların sayısını göstermektedir. Matrisin her bir sütunu her bir test kümesi için elde edilen öngörünün örneklemine göstermektedir. Güven aralıkları yüzdeler kullanılarak aşağıdaki gibi hesaplanır.

$$P\left(Yüzdelik(\{\hat{Y}_{.,j}\}, 0.025) \leq Y_{.,j} \leq Yüzdelik(\{\hat{Y}_{.,j}\}, 0.975)\right) = 0.95, j = 1, 2, \dots, ntest$$

YSA modelindeki m adet ağırlığın her biri için bootstrap örneklemi kullanılarak hipotez testleri gerçekleştirilir. Ağırlıklar için bootstrap örneklemi;

$$\{\widehat{W}_{i,j}\}, \quad i = 1, 2, \dots, bstn; j = 1, 2, \dots, m$$

şeklinde gösterilebilir. Hipotez testi için ilk olarak her bir ağırlığın örneğine normallik testi uygulanır. Normal dağılım koşulu sağlanır ise tek örneklem t -testi ile aksi halde işaret (signtest) testi ile dağılımın konum parametresinin sıfır olup olmadığı her bir ağırlık için test edilir. Hipotezler $i = 1, 2, \dots, m$ olmak üzere,

$$H_0: w_i = 0$$

$$H_1: w_i \neq 0$$

şeklinde olmaktadır.

7. UYGULAMA

Tezin uygulama kısmında; benzetim çalışması ve gerçek hayat zaman serilerinin çözümlenmesi yoluyla aykırı değer durumunda öngörü performanslarındaki değişimin görülmesi ve önerilen yöntem ile alternatif yöntemlerin karşılaştırılması amaçlanmıştır. Uygulamanın ilk aşamasında öncelikle bir benzetim çalışması yapılmıştır. Bu benzetim çalışması yardımıyla önerilen yöntem ve diğer alternatif yöntemlerin aykırı değer durumundaki öngörü performansları karşılaştırılmıştır. Ardından önerilen yöntem, gerçek hayat zaman serisi verilerine uygulandığında öngörü performansını ölçmek ve alternatif yöntemlerin öngörü performansları ile karşılaştırılmak amacıyla; literatürde sıkça kullanılan AUST (Avustralya Bira Tüketimi) ve ANSO (Ankara Hava Kirliliği) verileri çözümlenerek sonuçlar karşılaştırılmıştır. Son bölümde ise ağırlıklarına ait hipotez testlerinin yapılması ve ağırdan elde edilen öngörülere ait güven aralıklarının oluşturulması amacıyla hareketli blok bootstrap yöntemi kullanılarak ANSO verilerine ait bir uygulama yapılmıştır.

Öngörü performanslarının başarısı, serilerin gerçek değeri ile öngörü değerinin kullanıldığı çeşitli kriterler yardımıyla karşılaştırılmaktadır. Bu amaçla kullanılacak olan kriterler hata kareler ortalaması karekökü (HKOK), ortalama mutlak yüzdelik hata (OMYH), mutlak hata yüzdesi medyanı (MHYM) olarak belirlenmiştir. Kullanılacak olan kriterlerin formülleri sırasıyla eşitlik (7.1) eşitlik (7.2) ve eşitlik (7.3)'te verilmiştir.

$$HKOK = \sqrt{\frac{\sum_{t=1}^n (y_t - \hat{y}_t)^2}{n}} \quad (7.1)$$

$$OMYH = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (7.2)$$

$$MHYM = \text{medyan} \left(\frac{|y_t - \hat{y}_t|}{y_t} \right), \quad t = 1, 2, \dots, n \quad (7.3)$$

Burada n , y_t ve \hat{y}_t sırasıyla; öğrenme örneği sayısı, gözlenen değer ve öngörü değerini göstermektedir.

7.1 Benzetim Çalışması

Bu bölümde öncelikle ÇNM-YSA için simülasyonla veri üretilmiş ardından önerilen yöntem ve alternatif yöntemler için bu verilerle çözümlenmeler yapılmıştır. Yapılan bu çalışmada önerilen yöntemin aykırı değer varlığı durumundaki performansı, hem test hem de eğitim verisi için HKOK, OMYH ve MHYM ölçütleri bakımından geri yayımlı öğrenme algoritması (GY), garanti yakınsamalı parçacık sürü optimizasyonu (GRY-PSO) ve standart parçacık sürü optimizasyonu yöntemleri (PSO) ile eğitilmiş olan ÇNM-YSA'larla çözümlenip sonuçlar karşılaştırılmıştır. Veri üretim algoritması aşağıda verildiği gibidir.

Algoritma1. ÇNM-YSA için Veri Üretimi Algoritması

Adım 1. $U(0, 0.01)$ parametrelili düzgün dağılımdan rasgele olarak m adet başlangıç değeri (x_0) üretilir.

Adım 2. $U(0,1)$ parametrelili düzgün dağılımdan rasgele olarak m adet ağırlık ve m adet yan değeri üretilir.

Adım 3. Zaman serisinin n adet değeri eşitlik (7.4) yardımıyla ardışık olarak hesaplanır. x_t serisinin ilk m tane değeri ise Adım 1'de başlangıç değeri olarak üretilen x_0 'dır.

$$x_t = \prod_{j=1}^m (x_{t-j} \times w_j + b_j) , t = m + 1, \dots, n + m \quad (7.4)$$

Burada w_j ve b_j sırasıyla ÇNM-YSA'nın ağırlık ve yan değerlerini göstermektedir.

Adım 4. x_t zaman serisinin başlangıç değeri olan ilk m gözlemi atılarak, zaman serisinin n gözleme sahip olması sağlanır.

Adım 5. Hata serisi için, $N(0, \sigma^2)$ dağılımından rasgele olarak n adet gözlem üretilir ve ε_t serisi elde edilir.

Adım 6. y_t zaman serisi, x_t serisine ε_t serisi eklenerek elde edilir.

$$y_t = x_t + \varepsilon_t$$

Adım 7. y_t zaman serisinin maksimum değeri (*maksy*) belirlenir. Aykırı değer sayısı (*naykr*),

$$naykr = \lceil n \times y\ddot{u}zde \rceil$$

formülü ile hesaplanır. Burada *yüzde*, veri setinin kirletilme yüzdesidir. Verinin ilk $(n - ntest)$ gözlemi eğitim kümesi ve son $ntest$ gözlemi test kümesi olarak kullanılır. Eğitim kümesine eklenecek olan *naykr* adet aykırı değer ise, verinin ilk

$(n - n_{test})$ gözlemi içerisinde rasgele olarak seçilir. Bunun için $(n - n_{test})$ parametrelili kesikli düzgün dağılımdan n_{aykr} adet $r_i, i = 1, 2, \dots, n_{aykr}$ rasgele sayısı üretilir. y_t zaman serisinde seçilen gözlemlerin yerine $(maksy \times kat)$ değeri konulur. Burada kat , veriye aykırı değer olarak enjekte edilecek değerin veri setinin en büyük değerinin kaç katı olduğunu ifade etmektedir.

$$y_t(r_i) = maksy \times kat, \quad i = 1, 2, \dots, n_{aykr}$$

Sonuçta *yüzde* oranında kirletilmiş, n_{aykr} adet aykırı değer içeren ve çarpımsal modele uygun bir y_t zaman serisi elde edilmiş olur.

ÇNM-YSA için veri üretilirken kullanılan parametrelerin her birinin aldığı farklı değerler için çözümlenmeler yapılmıştır. Bu parametreler ve değerleri;

n : Üretilen veri setine ait gözlem sayısı ($n=100$ ve 500).

σ^2 : Hata varyansı ($\sigma^2= 1$ ve 10).

kat : Veri setine eklenecek olan aykırı değerin büyüklüğü ($kat=$ veriye ait maksimum değerin 5 ve 10 katı).

yüzde : Veri setinin kirletilme yüzdesi ($yüzde= \%5$ ve $\%10$).

tryüzde : Veri setinin budanma yüzdesi ($tryüzde= \%20$ ve $\%30$)

olarak değiştirilerek 32 farklı durum için çözümlenme yapılmıştır. Her bir durum için çözümlenme yapılmadan önce yukarıda algoritması verilen benzetim süreci ile 100 farklı veri üretilmiştir.

Önerilen yöntem haricinde; ÇNM-YSA-GY, ÇNM-YSA-GRYPSO ve ÇNM-YSA-PSO yöntemleri ile çözümlenme yapılmıştır. Çözümlenmeler sonucunda belirtilen yöntemlere ait hesaplanan ölçütlerin her birinin aşağıda belirtilen eğitim ve test kümeleri için ortalamaları hesaplanmıştır.

| | HKOK ölçütü değeri | OMYH ölçütü değeri | MHYM ölçütü değeri |
|--------------------|---------------------------|---------------------------|---------------------------|
| Eğitim kümesi için | HKOK _{eğitim} | OMYH _{eğitim} | MHYM _{eğitim} |
| Test kümesi için | HKOK _{test} | OMYH _{test} | MHYM _{test} |

İşlemler 100 kez tekrarlanmış ve tüm hata ölçütleri için her bir yönteme ait 100 farklı ortalama değer hesaplanmış ve bu değerler arasında istatistiksel olarak anlamlı bir farklılık olup olmadığı belirlenmiştir. Bunu yaparken öncelikle normallik testi sonuçlarına göre dört yöntemden elde edilen hata ölçütlerine parametrik ya da parametrik olmayan testlerden hangisinin uygulanacağına karar verilmiştir. Çizelge

7.1’de belirtilen tüm hata ölçütleri için yapılan Lilliefors normallik testi (p değerleri) sonuçları ve kullanılan testlerin türleri ve sonuçları verilmiştir.

Çizelge 7.1. Lilliefors normallik testi sonuçlarına göre kullanılacak test türü

| | Test Türü | HKOK | OMYH | MHYM | HKOK | OMYH | MHYM |
|-------|------------------|--------|--------|----------|--------|--------|--------|
| | | test | Test | test | egt | egt | egt |
| Durum | | KW | KW | F | KW | KW | KW |
| 1 | Test İstatistiği | 296.06 | 226.49 | 7798 | 374.06 | 237.20 | 351.39 |
| | p değeri | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Durum | | KW | KW | F | KW | KW | KW |
| 2 | Test İstatistiği | 354.77 | 297.23 | 31972.48 | 373.01 | 180.32 | 372.01 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum | | KW | KW | KW | KW | KW | KW |
| 3 | Test İstatistiği | 350.16 | 332.57 | 345.03 | 374.06 | 86.25 | 343.20 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

Yöntemlerden elde edilen hata ölçütlerine normallik testi uygulanmıştır. EK 1’de verilen normallik testi sonuçlarına göre kullanılacak olan test türüne karar verilmiştir. Test sonuçları incelendiğinde p değerinin 0.05’ten küçük olduğu görülmüştür. Bu durumda normal dağılıma sahip olmayan ölçütler için parametrik olmayan testlerden yararlanılmıştır. Yöntemlerin karşılaştırılmasında tüm grupların normal dağılımlı olduğu ölçütler için ANOVA (F) testi, diğer ölçütler açısından yöntemlerin karşılaştırılmasında ise Kruskal-Wallis testi kullanılmıştır. Kullanılan testler ve test istatistiği değerleri Çizelge 7.1’de durumlara göre ayrı ayrı verilmiştir. Diğer durumlar için test sonuçları ve p değerleri ise EK 1’de ayrıntılı olarak verilmektedir.

Kruskal-Wallis ve F testi için grupların ortalamaları arasındaki farkın istatistiksel olarak anlamlı olmadığı hipotezi test edilmiştir. Buna göre Çizelge 7.1 incelendiğinde; Kruskal-Wallis ve F testine ait tüm p değerlerinin 0.05’ten küçük olduğu görülmektedir. Buradan hareketle belirlenen ölçütler bakımından tüm gruplar arasında istatistiksel olarak anlamlı bir farklılık olduğu söylenir.

Kruskal-Wallis testi sonucunda elde edilen sıra sayıları ortalaması değerleri de incelendiğinde yöntemler arasındaki farklılık açıkça görülmektedir. Yöntemlere ait sıra sayıları ortalaması değerleri aşağıda Çizelge 7.2’de verilmektedir.

Çizelge 7.2. Ölçütlere göre Kruskal Wallis testi sıra sayıları ortalamaları

| | Önerilen Yöntem | ÇNM-GY | ÇNM-PSO | ÇNM-GRYPSO | |
|----------------|-----------------|---------|---------|------------|---------|
| Durum 1 | HKOKtest | 53.260 | 323.660 | 177.640 | 247.440 |
| | OMYHtest | 50.500 | 255.41 | 237.120 | 258.970 |
| | MHYMtest | 1.157 | 1.665 | 1.624 | 1.638 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 217.940 | 274.400 | 259.160 |
| | MHYMegt | 50.500 | 350.350 | 168.840 | 232.310 |
| Durum 2 | HKOKtest | 50.500 | 163.430 | 239.040 | 349.030 |
| | OMYHtest | 50.500 | 326.160 | 236.580 | 188.760 |
| | MHYMtest | 1.107 | 2.390 | 2.153 | 2.203 |
| | HKOKegt | 350.500 | 250.500 | 51.210 | 149.790 |
| | OMYHegt | 98.030 | 205.850 | 182.510 | 315.610 |
| | MHYMegt | 50.500 | 350.500 | 151.890 | 249.110 |

Çizelge 7.2’den görüldüğü üzere tüm durumlarda HKOKegt ölçütü için sıra sayıları ortalaması değerlerine bakıldığında, aykırı değer bulunması sebebiyle beklenildiği gibi en yüksek değere sahip yöntemin önerilen yöntem olduğu görülmektedir. Bununla birlikte önerilen yöntemin HKOKegt ölçütü haricindeki tüm ölçütlerde sıra sayıları ortalaması için en küçük değeri aldığı görülmüştür. Çizelge incelendiğinde tüm durum ve ölçütler için önerilen yöntemin; aykırı değer olması durumunda diğer yöntemlere göre oldukça iyi sonuç verdiği görülmektedir. Diğer durumlar için hesaplanan değerlere EK 2’de yer verilmektedir.

Kruskal-Wallis testi sonucunda tüm ölçütlere göre gruplar arasında istatistiksel olarak anlamlı bir farklılık olduğu belirlenmiştir. Bu sebeple hangi gruplar arasında farklılık olduğunu belirlemek amacıyla çoklu karşılaştırma testleri yapılmıştır. Farklılığın hangi gruplar arasında olduğunu belirlemek amacıyla Wilcoxon işaret testi kullanılmıştır. Medyanların eşitliği hipotezi test edildiğinde tüm p değerlerinin 0.05’ten küçük olduğu görülmüştür. Bu durumda medyanlar eşit değildir ve tüm ikili gruplar arasında farklılık olduğu söylenir. Durumlar için; ölçütlere göre yöntemlerden elde edilen medyan değerleri Çizelge 7.3’te verilmektedir.

Çizelge 7.3. Durumlara göre yöntemlerden elde edilen medyan değerleri

| | Önerilen Yöntem | ÇNM-GY | ÇNM-PSO | ÇNM-GRYPSO | |
|----------------|-----------------|--------|---------|------------|--------|
| Durum 1 | HKOKtest | 1.140 | 1.264 | 1.2437 | 1.2502 |
| | OMYHtest | 3.126 | 6.203 | 6.1298 | 6.2047 |
| | MHYMtest | 1.155 | 1.6612 | 1.6242 | 1.6398 |
| | HKOKegt | 3.665 | 3.3667 | 3.3116 | 3.3206 |
| | OMYHegt | 4.287 | 6.094 | 6.2074 | 6.1758 |
| | MHYMegt | 1.045 | 1.4884 | 1.4051 | 1.4196 |
| Durum 2 | HKOKtest | 1.263 | 1.831 | 1.865 | 1.929 |
| | OMYHtest | 3.143 | 7.976 | 7.746 | 7.616 |
| | MHYMtest | 1.104 | 2.382 | 2.153 | 2.202 |
| | HKOKegt | 6.946 | 6.250 | 6.136 | 6.143 |
| | OMYHegt | 8.364 | 13.787 | 13.637 | 15.126 |
| | MHYMegt | 1.052 | 2.263 | 2.059 | 2.114 |

Çizelge 7.3'ten görüldüğü gibi tüm ölçütler için önerilen yöntemin medyan değerinin diğer yöntemlerle karşılaştırıldığında daha küçük olduğu görülmektedir. Buradan hareketle önerilen yöntemin aykırı değer durumunda diğer yöntemlere göre daha iyi sonuçlar verdiği söylenebilir. Diğer durumlar için yöntemlerin medyan değerleri ise EK 3'te verilmektedir.

Tablonun tümüne bakıldığında, parametre değerlerindeki değişimin sonuçlar üzerindeki etkisini yorumlayacak olursak;

* Tüm parametreler sabit tutulup, veri setine eklenecek aykırı değer in büyüklüğü olan “kat” parametresinin değeri 5'ten 10'a çıkarıldığında;

- Önerilen yöntem haricindeki tüm yöntemlerin test kümesi ölçütlerine ait medyan değerlerinde artış meydana gelmiştir.

- Bununla birlikte önerilen yöntem ve diğer yöntemlerde; aykırı değer in, ağı n eğitim kümesine eklenmesinden dolayı eğitim kümesi ile ilgili olan ölçütlerin medyan değerlerinde yaklaşık 2 katına kadar artış gözlemlenmektedir.

* Tüm parametreler sabit tutulup, veri setinin kirletilme yüzdesi olan “yüzde” parametresinin değeri %5'ten %10'a çıkarıldığında;

- Önerilen yönteme ait tüm ölçütlerin medyan değerlerinde küçük miktarda değişimler olurken, diğer yöntemlere ait tüm ölçütlerin medyan değerlerinde yaklaşık 2 kata kadar artış gözlenmiştir.

* Bir diğerinde tüm parametreler sabit tutulup, hata varyansı “ σ^2 ” parametresinin değeri 1'den 10'a çıkarıldığında;

- Önerilen yöntem ve diğer yöntemlere ait eğitim ve test kümelerinin HKOK ölçütünün medyan değerlerinde yaklaşık 10 kata kadar bir artış gözlemlenirken,

- Önerilen yöntem ve diğer yöntemlere ait diğer tüm ölçütlerin medyan değerlerinde ise azalma meydana gelmiştir.

* Son olarak diğer tüm parametreler sabit tutulup, veri setinin budanma yüzdesi olan “tryüzde” parametresinin değeri %20’den %30’a çıkarıldığında;

- Önerilen yöntem haricindeki tüm yöntemlerin test kümesinin HKOK ölçütüne ait medyan değerleri 3 katına çıkarken,

- Tüm yöntemlerin test kümesinin OMYH ölçütüne ait medyan değerlerinde azalma meydana gelmiş,

- Bununla birlikte tüm yöntemlerin diğer ölçütlerinin medyan değerlerinde kayda değer bir değişim görülmemiştir.

* Aynı şekilde diğer tüm parametreler sabit tutulup, veri setine ait gözlem sayısı olan “n” parametresinin değerinin 100’den 500’e çıkarıldığında;

- Önerilen yönteme ait test kümesinin OMYH ölçütünün medyan değerlerinde kayda değer bir artış gözlemlenmezken, diğer yöntemlerin bu ölçütüne ait medyan değerlerinde yaklaşık 1.5 kata kadar bir artış gözlenmiştir.

- Bunun yanında önerilen yöntemin eğitim kümesine ait OMYH ölçütünün medyan değerinde azalma meydana gelirken, diğer tüm yöntemlerin bu değerinde artış meydana gelmiştir.

- Tüm yöntemlere ait diğer ölçütlerin medyan değerlerinde ise kayda değer bir değişim görülmemiştir.

7.2 Gerçek Hayat Zaman Serileri ile Çözümleme

Uygulamanın bu bölümünde literatürde sıklıkla kullanılan Avustralya bira tüketimi (AUST) ve Ankara hava kirliliği (ANSO) zaman serisi verileri önerilen yöntem ile çözümlenmiş ve elde edilen sonuçlar literatürdeki bazı yöntemlerin sonuçları ile birlikte değerlendirilmiştir. Önerilen yöntemin ve diğer yöntemlerin öngörü performanslarının karşılaştırılmasında ise HKOK ve OMYH ölçütleri kullanılmıştır.

Önerilen yöntemin orijinal serideki öngörü performansının karşılaştırılması amacıyla literatürde sıklıkla kullanılan bazı yöntemlerle de çözümleme yapılmıştır.

Bu yöntemler;

ÇNM-YSA-GY: Geri yayımlı öğrenme algoritmasına dayalı çarpımsal nöron model yapay sinir ağı

ÇNM-YSA-PSO: Parçacık sürü optimizasyonuna dayalı çarpımsal nöron model yapay sinir ağı

ÇK-YSA-PSO: Parçacık sürü optimizasyonuna dayalı çok katmanlı yapay sinir ağı

ÇK-İB-YSA: Çok katmanlı ileri beslemeli yapay sinir ağı

WMES: Winters'ın çarpımsal üstel düzleştirilmesi

SARIMA: Mevsimsel otoregresif hareketli ortalamalar

RT-YSA: Radyal tabanlı yapay sinir ağı

E-YSA: Elman tipi yapay sinir ağı

Öncelikle verinin orijinal hali ile çözümleme yapılmış, ardından yöntemin aykırı değer durumundaki performansını görmek üzere iki farklı durumda daha çözümleme yapılmıştır. Bunu yaparken de veri setinin en yüksek değerinin 5 ve 10 katı eğitim verisine aykırı değer olarak enjekte edilerek çözümleme yapılmıştır. Çözümlemelerde yapılacak olan değişikliklere ait bilgiler durum adı altında aşağıda açıklanmaktadır.

Durum 1. Orijinal veri için çözümleme;

İlk olarak yöntemin performansını görmek üzere orijinal zaman serisi ile çözümleme yapılmıştır.

Durum 2. Orijinal verinin 5 kat aykırı değeri içerdiği durum için çözümleme;

Aykırı değer durumunda öngörü performansını incelemek amacıyla; veri setine ait bir veya birden fazla gözlem değeri, veri setinin en büyük değerinin 5 katı olan değer ile değiştirilmiştir.

Durum 2.1. İlk olarak eğitim kümesindeki 15. gözlem değeri, veri setinin en büyük değerinin 5 katı olarak değiştirilmiştir.

Durum 2.2. Bu durumda eğitim kümesindeki 15. ve 120. gözlem değeri, veri setinin en büyük değerinin 5 katı olarak değiştirilmiştir.

Durum 2.3. Bu durumda ise eğitim kümesindeki 15., 75. ve 120. gözlem değeri, veri setinin en büyük değerinin 5 katı olarak değiştirilmiştir.

Durum 3. Orijinal verinin 10 kat aykırı değeri içerdiği durum için çözümleme;

Aykırı değer durumunda öngörü performansını incelemek amacıyla; veri setine ait bir veya birden fazla gözlem değeri, veri setinin en büyük değerinin 10 katı olan değer ile değiştirilmiştir.

Durum 3.1. İlk olarak eğitim kümesindeki 15. gözlem değeri, veri setinin en büyük değerinin 10 katı olarak değiştirilmiştir.

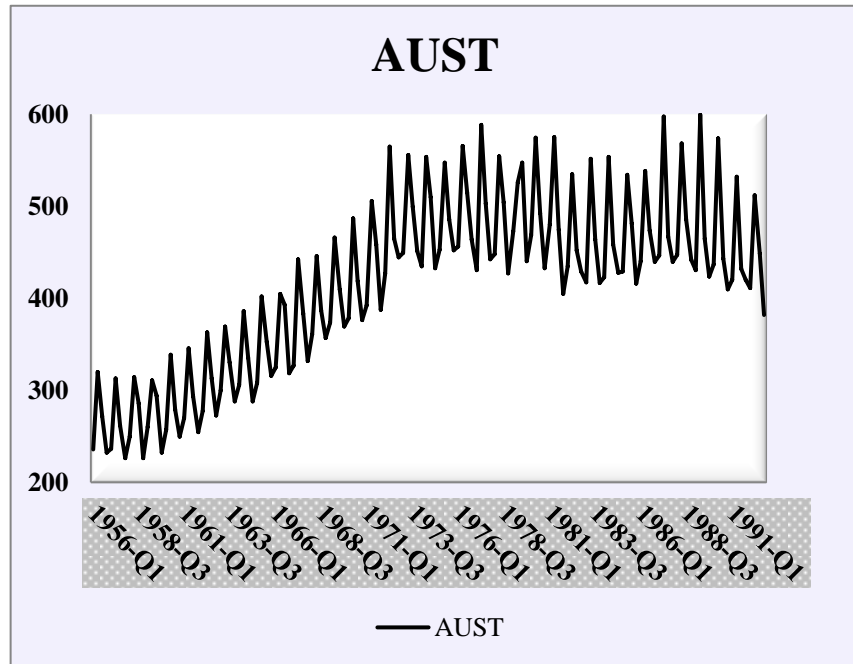
Durum 3.2. Bu durumda eğitim kümesindeki 15. ve 120. gözlem değeri, veri setinin en büyük değerinin 10 katı olarak değiştirilmiştir.

Durum 3.3. Bu durumda ise eğitim kümesindeki 15., 75. ve 120. gözlem değeri, veri setinin en büyük değerinin 10 katı olarak değiştirilmiştir.

Yukarıda belirtilen her bir durum için AUST ve ANSO serilerine ait yapılan analiz sonuçları ve yorumları ile birlikte öngörü grafikleri aşağıda verilmektedir.

7.2.1. Avustralya bira tüketimi verisi kullanılarak yapılan çözümleme

1956-1994 yılları arasında çeyreklik olarak gözlemlenen ve 148 gözleme sahip olan AUST zaman serisi verileri önerilen yöntem ile çözümlenmiştir. Trend ve mevsimsel bileşene sahip olan AUST zaman serisinin grafiği Şekil 7.1’de verildiği gibidir.



Şekil 7.1. 1956-1994 yılları arasında gözlemlenen AUST zaman serisi

Zaman serisinin son 16 gözlemi test verisi olarak ayrılmış, geriye kalan 132 veri eğitim verisi olarak kullanılmıştır. Verinin çözümlenmesinde ÇNM-YSA'nın girdi sayısı (m) 2 ile 8 arasında değiştirilmiştir. Parçacık sayısı ise 30 olarak alınmıştır. Ayrıca her çözümlemede yineleme sayısı 100 olarak alınmıştır.

Yukarıda belirtilen her durum için elde edilen sonuçlar, yorumları ve öngörü grafikleri aşağıda verilmektedir.

Durum 1. Orijinal veri için çözümlenme

Önerilen yönteme ait en iyi sonuç girdi sayısı $m=2$ olduğu durumda elde edilmiştir. Önerilen yöntem ve diğer yöntemlerden elde edilen sonuçların en iyi durumları için test kümesine ait HKOK ve OMYH değerleri Çizelge 7.4'te verilmiştir (Gundogdu ve diğ., 2015).

Çizelge 7.4. Orijinal AUST zaman serisi için tüm yöntemlerden elde edilen ölçütler

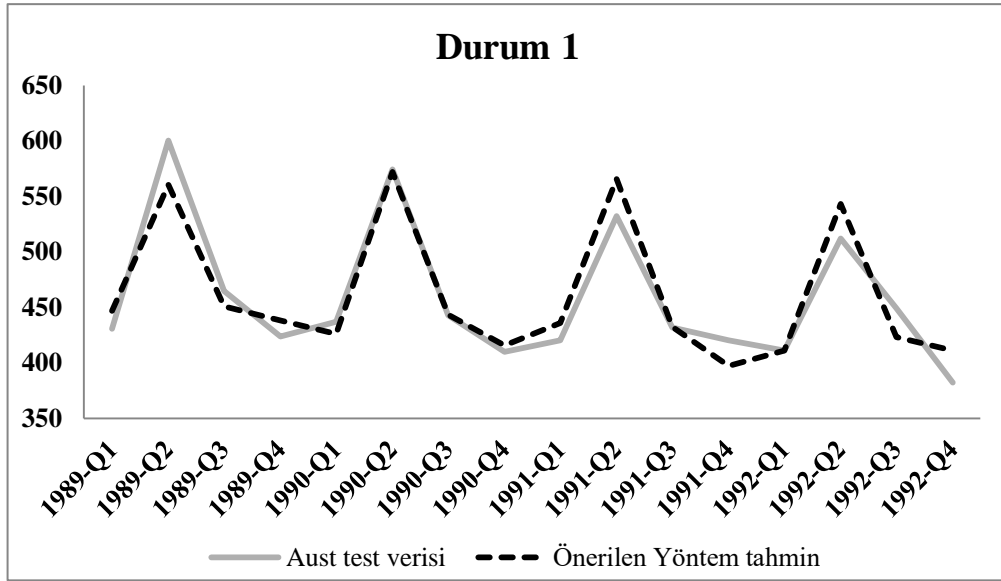
| Yöntem | HKOK | OMYH |
|------------------------|----------------|---------------|
| ÇNM-YSA-GY | 74.2551 | 0.0983 |
| WMES | 53.3295 | 0.1072 |
| SARIMA | 47.0367 | 0.0949 |
| ÇK-YSA-PSO | 44.7780 | 0.0856 |
| RT-YSA | 41.7000 | 0.0686 |
| ÇNM-YSA-PSO | 26.7831 | 0.0498 |
| ÇK-İB-YSA | 24.1052 | 0.0476 |
| E-YSA | 22.6581 | 0.0436 |
| Önerilen Yöntem | 20.5351 | 0.0384 |

Çizelge 7.4 incelendiğinde önerilen yöntemin hem HKOK hem de OMYH ölçütleri bakımından, tüm yöntemler içerisinde en iyi performansa sahip olduğu görülmektedir.

Önerilen yöntemde kullanılan ağa ait ağırlık ve yan değerleri ise aşağıda verilmektedir.

| | $i=1$ | $i=2$ |
|-------|--------|---------|
| w_i | 0.7194 | 0.3820 |
| b_i | 7.0188 | -0.1993 |

Elde edilen öngörü değerlerinin, gerçek değerlere uygunluğunu gözlemlemek amacıyla; gerçek değerler ve öngörü değerlerinin birlikte grafiği çizilmiş ve Şekil 7.2'de verilmiştir.



Şekil 7.2. AUST verisi Durum 1 için gözlenen değerler ile elde edilen öngörü değerlerinin birlikte grafiği

Şekil 7.2 incelendiğinde öngörü değerlerinin, test verisinin gözlem değerleri ile uyumlu olduğu görülmektedir.

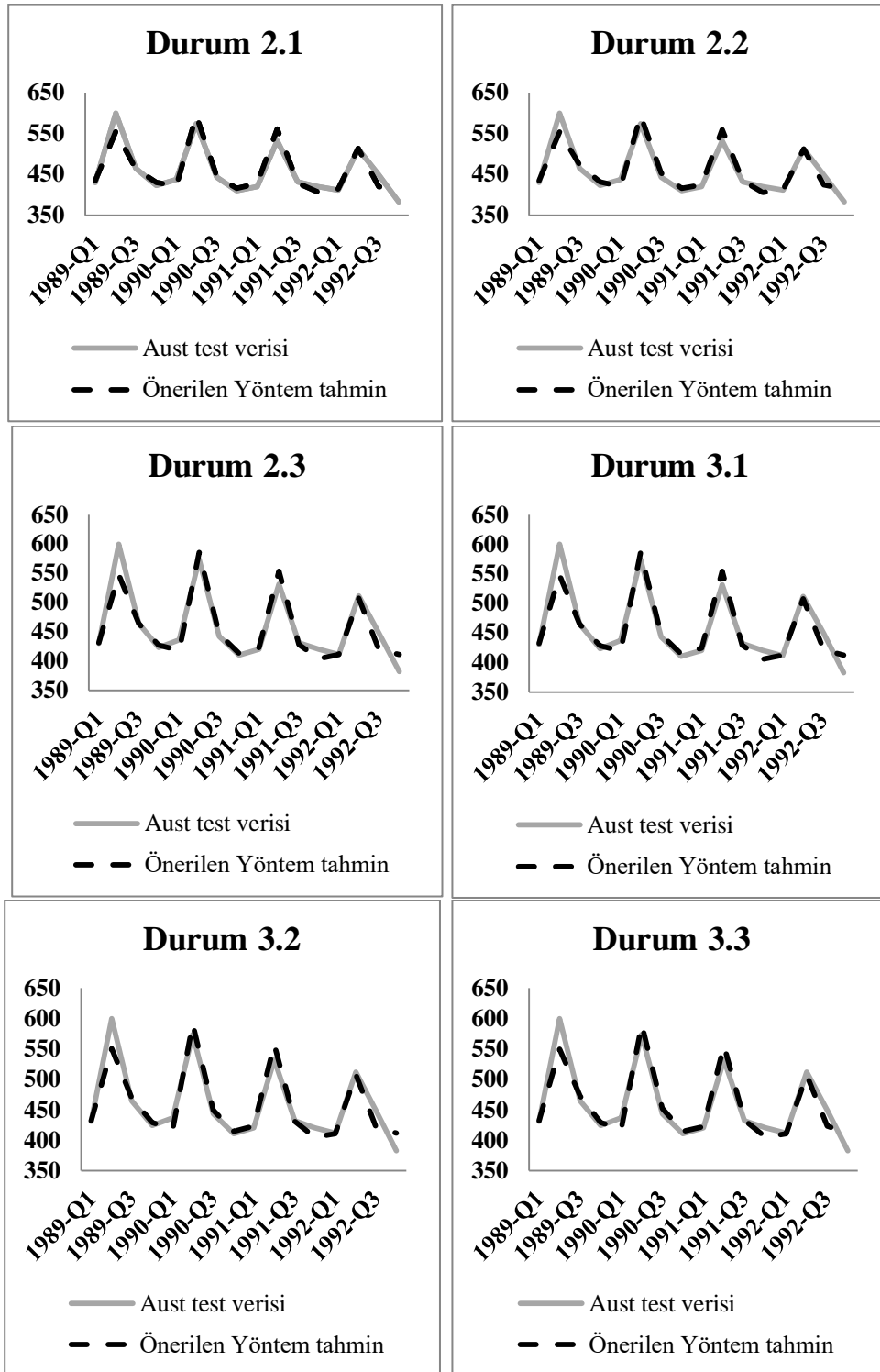
Diğer tüm durumlar için önerilen yöntem ve alternatif yöntemlerden elde edilen performans ölçütlerinin değerleri Çizelge 7.5'te birlikte verilmektedir. Bu durumlara ait parametre değerleri ise EK 4' te verilmiştir.

Çizelge 7.5. AUST test verisi için tüm durumlarda yöntemlerden elde edilen performans ölçütleri

| | | AUST | | | |
|---------------------|-----------|------|------------|-------------|-----------------|
| | | HKOK | ÇNM-YSA-GY | ÇNM-YSA-PSO | Önerilen Yöntem |
| 5 Kat Aykırı Değer | Durum 2.1 | HKOK | 176.5654 | 70.4099 | 18.8972 |
| | | OMYH | 0.3827 | 0.0863 | 0.0288 |
| | Durum 2.2 | HKOK | 284.1976 | 79.0784 | 18.8850 |
| | | OMYH | 0.6320 | 0.1140 | 0.0300 |
| | Durum 2.3 | HKOK | 328.3296 | 67.8867 | 18.8894 |
| | | OMYH | 0.7255 | 0.1351 | 0.0265 |
| 10 Kat Aykırı Değer | Durum 3.1 | HKOK | 188.1847 | 70.1409 | 18.8741 |
| | | OMYH | 0.4107 | 0.1012 | 0.0269 |
| | Durum 3.2 | HKOK | 224.7867 | 73.3807 | 18.8436 |
| | | OMYH | 0.4939 | 0.1530 | 0.0276 |
| | Durum 3.3 | HKOK | 244.2329 | 104.9389 | 18.9083 |
| | | OMYH | 0.5347 | 0.2233 | 0.0282 |

Çizelge 7.5 incelendiğinde önerilen yöntemin diğer yöntemlere göre her iki ölçüt bakımından da tüm durumlarda en iyi sonucu verdiği görülmektedir. Bunun

yanında test verisine ait gözlenen değerler ve elde edilen öngörü değerlerinin durumlara göre ayrı ayrı grafikleri Şekil 7.3’te verilmektedir.

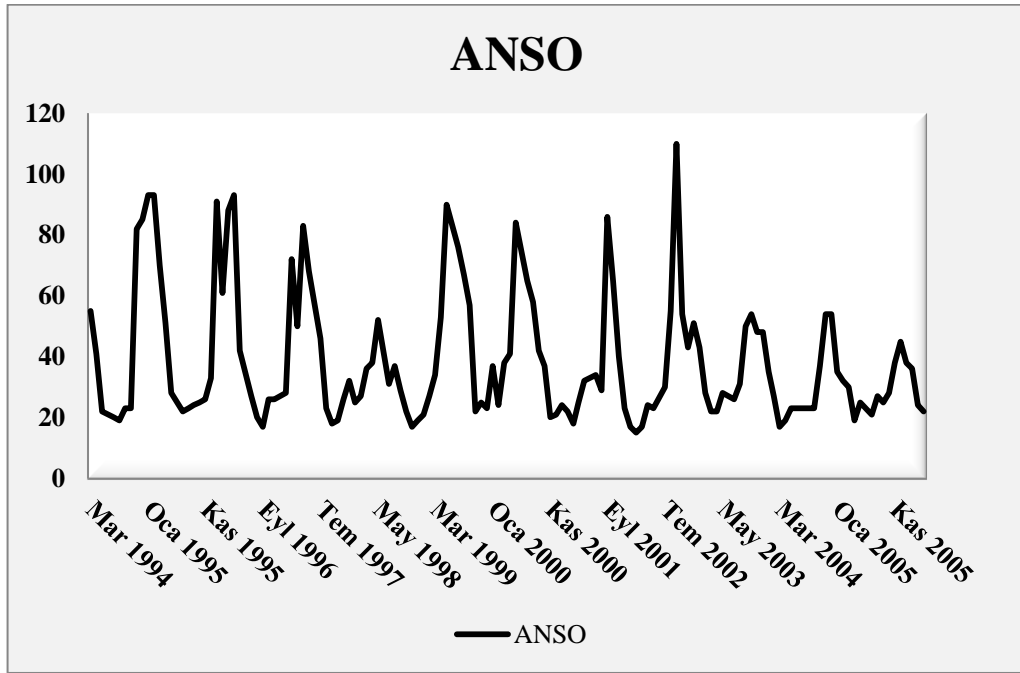


Şekil 7.3. Tüm durumlar için gözlenen değerler ile elde edilen öngörü değerlerinin birlikte grafikleri

Şekil 7.3 incelendiğinde tüm diğer durumlar için öngörü değerlerinin, test verisinin gözlem değerleri ile uyumlu olduğu görülmektedir. Böylece önerilen yöntemin öngörü performansının iyi olduğu söylenebilir.

7.2.2. Ankara hava kirliliği verisi kullanılarak yapılan çözümleme

1994-2006 yılları arasında aylık olarak gözlemlenen ve 134 gözleme sahip olan ANSO zaman serisi verileri önerilen yöntem ile çözümlenmiştir. Trend ve mevsimsel bileşene sahip olan ANSO zaman serisinin grafiği Şekil 7.4'te verildiği gibidir.



Şekil 7.4. 1994-2006 yılları arasında aylık olarak gözlemlenen ANSO zaman serisi

Zaman serisinin son 10 gözlemi test verisi olarak ayrılmış, geriye kalan 124 veri eğitim verisi olarak kullanılmıştır. Verinin çözümlenmesinde ÇNM-YSA'nın girdi sayısı (m) 2 ile 12 arasında değiştirilmiştir. Parçacık sayısı ise 30 olarak alınmıştır. Ayrıca her çözümlemede yineleme sayısı 100 olarak alınmıştır.

Yukarıda belirtilmiş olan durumlar için elde edilen sonuçlar, yorumları ve tahmin grafikleri aşağıda verilmektedir.

Durum 1. Orijinal veri için çözümleme

Önerilen yönteme ait en iyi sonuç girdi sayısı $m=4$ olduğu durumda elde edilmiştir. Önerilen yöntem ve diğer yöntemlerden elde edilen sonuçların en iyi durumları için test kümesine ait HKOK ve OMYH değerleri Çizelge 7.6'da verilmiştir (Gundogdu ve diğ., 2015).

Çizelge 7.6. Orijinal ANSO zaman serisi için tüm yöntemlerden elde edilen ölçütler

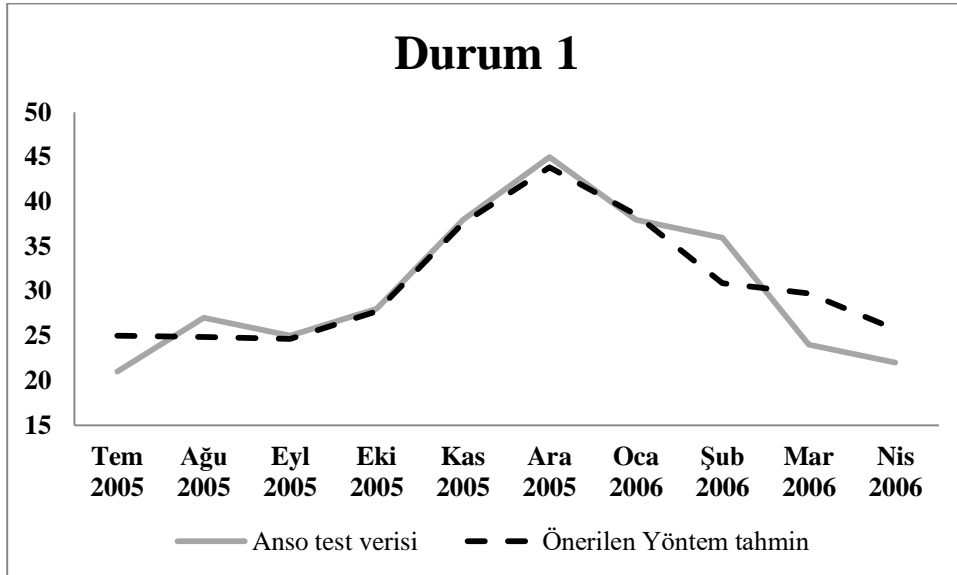
| Yöntem | HKOK | OMYH |
|------------------------|---------------|---------------|
| ÇNM-YSA-GY | 33.0310 | 1.1883 |
| RTF-YSA | 10.3189 | 0.3248 |
| SARIMA | 9.6249 | 0.2336 |
| WMES | 7.1062 | 0.2204 |
| ÇK-PSO | 6.7440 | 0.1904 |
| ÇNM-YSA-PSO | 6.3404 | 0.1822 |
| ÇK-İB-YSA | 3.7402 | 0.0995 |
| E-YSA | 3.6718 | 0.9900 |
| Önerilen Yöntem | 3.0902 | 0.0894 |

Çizelge 7.6 incelendiğinde önerilen yöntemin hem HKOK hem de OMYH ölçütleri bakımından, tüm yöntemler içerisinde en iyi performansa sahip olduğu görülmektedir.

Önerilen yöntemde kullanılan ağa ait ağırlık ve yan değerleri ise aşağıda verilmektedir.

| | $i=1$ | $i=2$ | $i=3$ | $i=4$ |
|-------|---------|--------|---------|---------|
| w_i | -0.1813 | 0.6252 | -0.5440 | 1.2893 |
| b_i | 0.9491 | 4.2469 | 0.5254 | -1.2234 |

Elde edilen öngörü değerlerinin, gerçek değerlere uygunluğunu gözlemlemek amacıyla; gerçek değerler ve öngörü değerlerinin birlikte grafiği çizilmiş ve Şekil 7.5'te verilmiştir.



Şekil 7.5. ANSO verisi Durum 1 için gözlenen değerler ile elde edilen öngörü değerlerinin birlikte grafiği

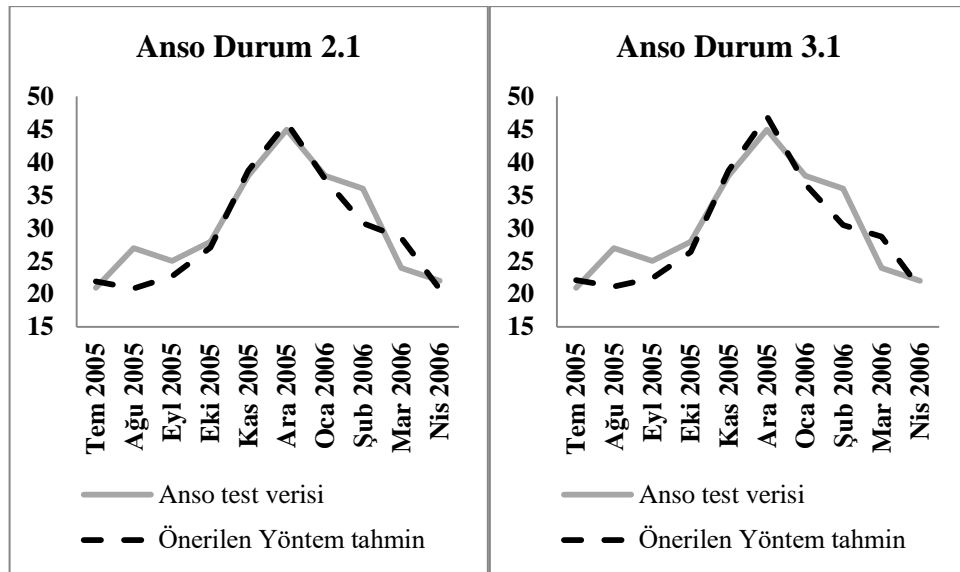
Şekil 7.5 incelendiğinde öngörü değerlerinin, test verisinin gözlem değerleri ile uyumlu olduğu görülmektedir.

Diğer durumlar için önerilen yöntem ve alternatif yöntemlerden elde edilen performans ölçütlerinin değerleri Çizelge 7.7’de birlikte verilmektedir. Bu durumlara ait parametre değerleri ise EK 5’ te verilmiştir.

Çizelge 7.7. ANSO test verisi için tüm durumlarda yöntemlerden elde edilen performans ölçütleri

| | | ANSO | | |
|----------------------------|-------------------|------------|-------------|-----------------|
| | | ÇNM-YSA-GY | ÇNM-YSA-PSO | Önerilen Yöntem |
| 5 Kat Aykırı Değer | Durum HKOK | 252.2203 | 20.6752 | 3.0826 |
| | 2.1 OMYH | 8.8910 | 0.6634 | 0.0840 |
| 10 Kat Aykırı Değer | Durum HKOK | 527.1575 | 25.6036 | 3.2522 |
| | 3.1 OMYH | 18.5194 | 0.9035 | 0.0933 |

Çizelge 7.7 incelendiğinde önerilen yöntemin diğer yöntemlere göre her iki ölçüt bakımından da tüm durumlarda en iyi sonucu verdiği görülmektedir. Bunun yanında test verisine ait gözlenen değerler ve elde edilen öngörü değerlerinin durumlara göre ayrı ayrı grafikleri Şekil 7.6’da verilmektedir.



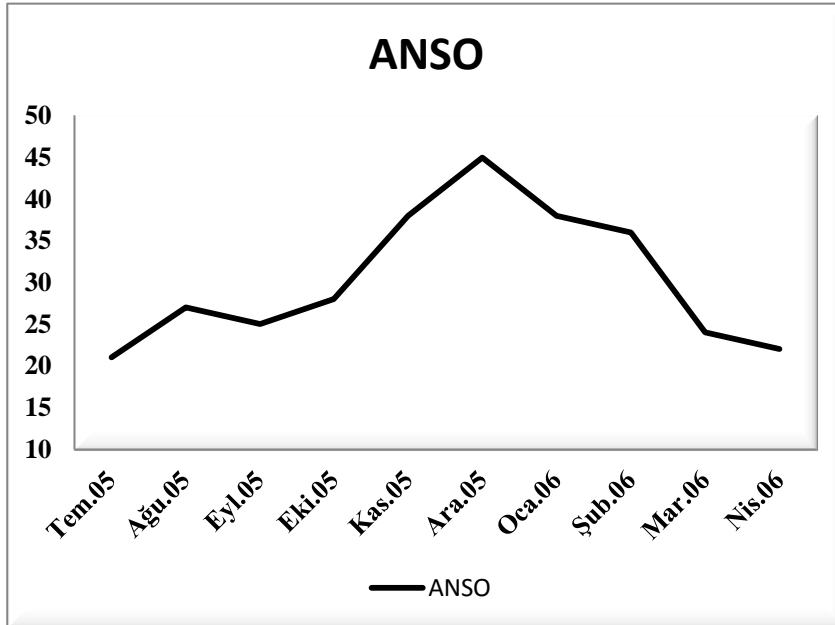
Şekil 7.6. Diğer durumlar için gözlenen değerler ile elde edilen öngörü değerlerinin birlikte grafikleri

Şekil 7.6 incelendiğinde diğer durumlar için öngörü değerlerinin, test verisinin gözlem değerleri ile uyumlu olduğu görülmektedir.

7.3 Rastgele Hareketli Kayan Blok Bootstrap Yöntemi ile Güven Aralığı Tahmini ve Hipotez Testleri

Yapay sinir ağları ile nokta tahmini yapmak mümkünken, hipotez testi yapılamadığı için istatistiksel değerlendirme de yapılamamaktadır. Ancak YSA ile bootstrap yöntemler birlikte kullanıldığında hem güven aralığı tahmini hem de hipotez testleri birlikte yapılabilmektedir. Bu sayede YSA'larda istatistiksel değerlendirme yapmak mümkün olmaktadır.

Tezin bu kısmında çarpımsal nöron model yapay sinir ağında öngörüler için güven aralıklarının elde edilmesi ve modeldeki parametreler olan ağırlıklar için hipotez testlerinin yapılması, zaman serilerinin ardışık gözlenme durumunu da göz önünde bulunduran hareketli blok bootstrap yöntem ile gerçekleştirilmiştir. Çözümlemede logaritmik dönüşüm yapılmış ANSO orijinal zaman serisi ve aykırı değer eklenmiş ANSO zaman serisi verileri kullanılmıştır. Zaman serisinin son 10 gözlemi test verisi olarak ayrılmış, geriye kalan 124 veri eğitim verisi olarak kullanılmıştır. Trend ve mevsimsel bileşene sahip olan ANSO zaman serisinin test kümesi verilerine ait grafik Şekil 7.7'de verildiği gibidir.



Şekil 7.7. Anso zaman serisine ait test kümesi verilerinin grafiği

7.3.1 Orijinal veri için çözümlenme

İlk olarak ANSO orijinal verilerinin logaritmik dönüşümlü hali ile eğitilen YSA'dan elde edilen öngörüler için güven aralığı tahmini yapılmıştır. Kullanılan parametreler aşağıda verilmektedir.

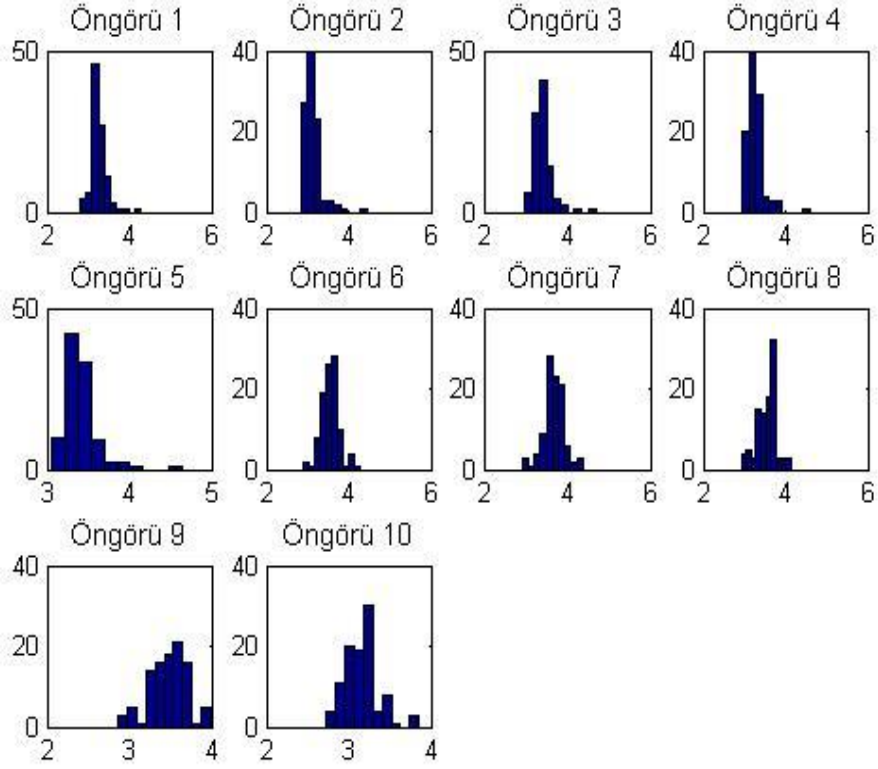
| | | | |
|------------|------|-----------|-------|
| n | :134 | pn | :30 |
| $n_{eğt}$ | :124 | m_0 | :2-8 |
| n_{test} | :10 | $maxt$ | :100 |
| $bstn$ | :100 | $yuzdetr$ | :0.20 |

Zaman serisine ait gerçek ve öngörü değerleri, güven aralığı değerleri birlikte aşağıda Çizelge 7.8'de verilmektedir.

Çizelge 7.8. Orijinal seriye ait gerçek ve öngörü değerleri ile birlikte güven aralığı değerleri

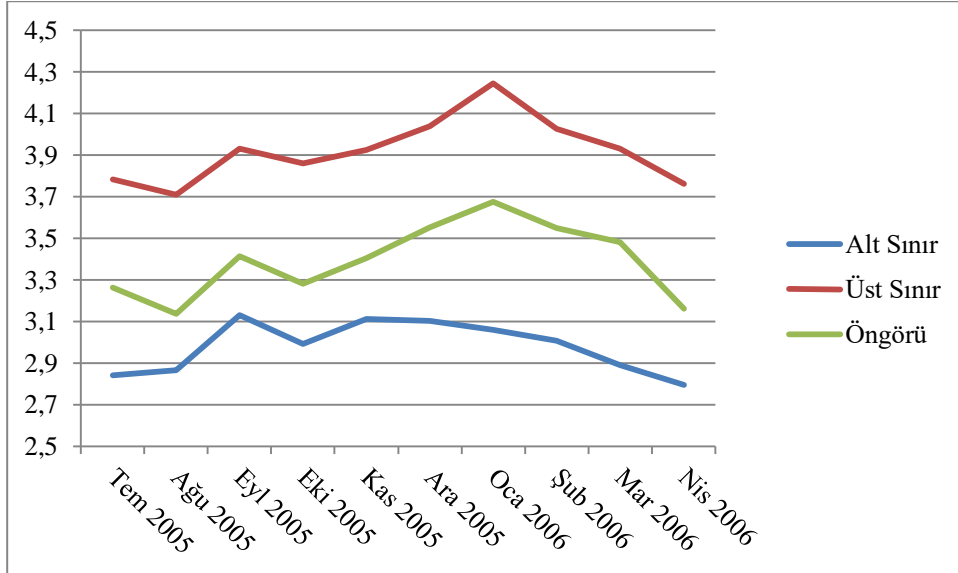
| Tarih | Alt Sınır | Üst Sınır | Öngörü | Gerçek |
|--------------|-----------|-----------|--------|--------|
| Temmuz 2005 | 2.8401 | 3.7836 | 3.2629 | 3.0445 |
| Ağustos 2005 | 2.8651 | 3.7082 | 3.1373 | 3.2958 |
| Eylül 2005 | 3.1306 | 3.9309 | 3.4131 | 3.2189 |
| Ekim 2005 | 2.9907 | 3.8599 | 3.2805 | 3.3322 |
| Kasım 2005 | 3.1123 | 3.9254 | 3.4038 | 3.6376 |
| Aralık 2005 | 3.1040 | 4.0393 | 3.5522 | 3.8067 |
| Ocak 2005 | 3.0584 | 4.2457 | 3.6749 | 3.6376 |
| Şubat 2005 | 3.0078 | 4.0280 | 3.5484 | 3.5835 |
| Mart 2005 | 2.8899 | 3.9326 | 3.4817 | 3.1781 |
| Nisan 2005 | 2.7962 | 3.7605 | 3.1603 | 3.0910 |

Çizelge 7.8'de görüldüğü gibi tüm öngörü değerleri ilgili güven aralığı içerisinde yer almaktadır. Öngörüler için elde edilen histogram grafikleri ise Şekil 7.8'de verilmektedir.



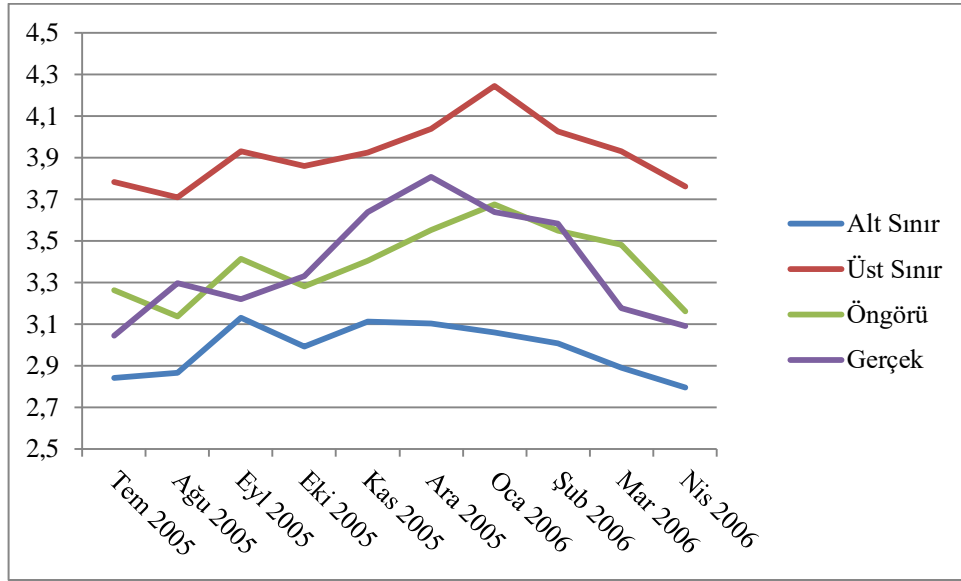
Şekil 7.8. Öngörüler için histogram grafikleri

Öngörüler için elde edilen güven aralıklarının grafiği aşağıda Şekil 7.9’da verilmektedir.



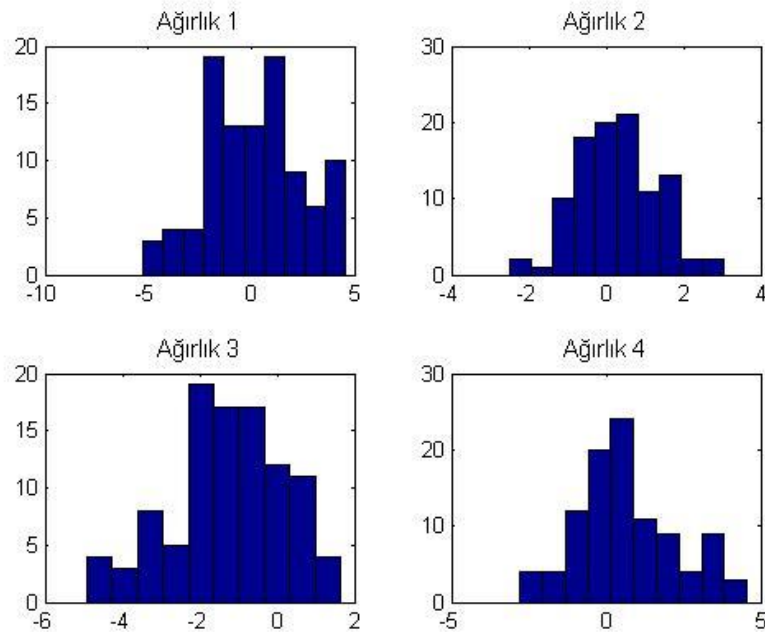
Şekil 7.9. Öngörüler için elde edilen güven aralıklarının grafiği

Grafikten görüldüğü üzere güven aralıkları öngörü değerlerini tam olarak kapsamaktadır. ANSO verisi için elde edilen öngörüler ve gerçek değerlerin birlikte grafiği ise Şekil 7.10'da verilmektedir.



Şekil 7.10. Öngörü ve gerçek değerlerin birlikte grafiği

Ağırlıklar için elde edilen histogram grafikleri Şekil 7.11'de verilmektedir.



Şekil 7.11. Ağırlıklar için histogram grafikleri

Parametrelerin dağılımı normal dağılım olmadığından; parametrik olmayan işaret testi ile ağırlıkların medyanının sıfıra eşit olduğu yokluk hipotezi test edilmiş,

1. ve 2. ağırlıklara ait p değerleri 0.05'ten büyük çıkarken, 3. ve 4. ağırlıklara ait p değerleri 0.05'ten küçük çıktığı görülmüştür.

7.3.2 Aykırı değer içeren veri için çözümleme

Veri setinde aykırı değer olması durumunda çözümleme yapılabildiğini göstermek için; ANSO verilerine 5 kat aykırı değerli bir veri eklenerek logaritması alınmış, elde edilen yeni veri seti ile YSA eğitilmiştir. Elde edilen öngörüler için güven aralığı tahmini yapılmıştır. Kullanılan parametreler aşağıda verilmektedir.

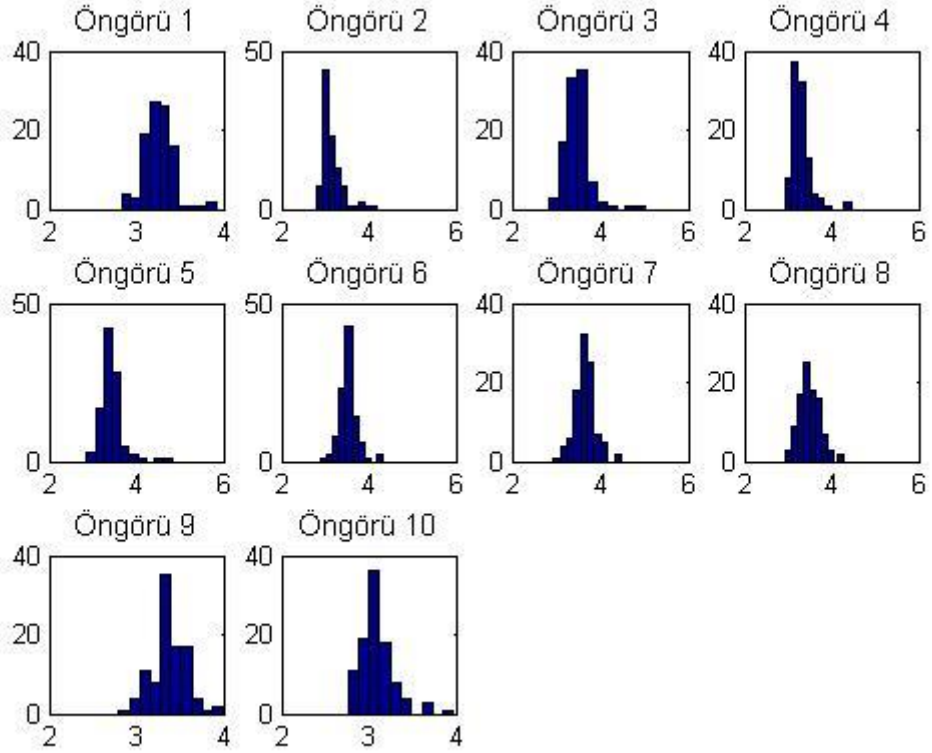
| | | | |
|------------|------|-----------|-------|
| n | :134 | pn | :30 |
| $n_{eğt}$ | :124 | m_0 | :2-8 |
| n_{test} | :10 | $maxt$ | :100 |
| $bstn$ | :100 | $yuzdetr$ | :0.20 |

Aykırı değer içeren zaman serisine ait gerçek ve öngörü değerleri, güven aralığı değerleri birlikte aşağıda Çizelge 7.9'da verilmektedir.

Çizelge 7.9. Aykırı değer içeren seriye ait gerçek ve öngörü değerleri ile birlikte güven aralığı değerleri

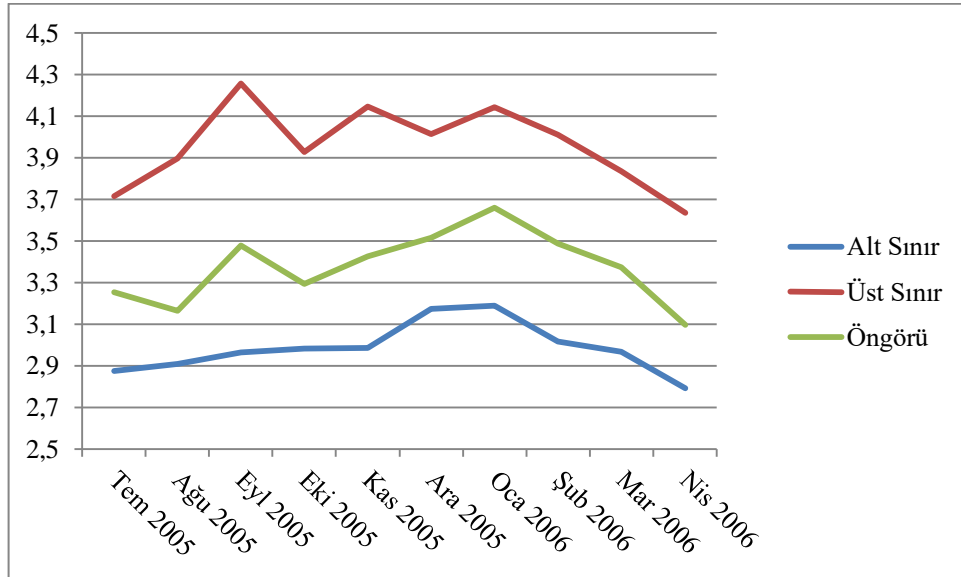
| Tarih | Alt Sınır | Üst Sınır | Öngörü | Gerçek |
|--------------|-----------|-----------|--------|--------|
| Temmuz 2005 | 2.8759 | 3.7154 | 3.2533 | 3.0445 |
| Ağustos 2005 | 2.9078 | 3.8974 | 3.1638 | 3.2958 |
| Eylül 2005 | 2.9644 | 4.2578 | 3.4773 | 3.2189 |
| Ekim 2005 | 2.9840 | 3.9282 | 3.2925 | 3.3322 |
| Kasım 2005 | 2.9869 | 4.1459 | 3.4276 | 3.6376 |
| Aralık 2005 | 3.1722 | 4.0142 | 3.5157 | 3.8067 |
| Ocak 2005 | 3.1902 | 4.1439 | 3.6608 | 3.6376 |
| Şubat 2005 | 3.0174 | 4.0114 | 3.4876 | 3.5835 |
| Mart 2005 | 2.9667 | 3.8343 | 3.3724 | 3.1781 |
| Nisan 2005 | 2.7910 | 3.6344 | 3.0958 | 3.0910 |

Çizelge 7.9'da görüldüğü gibi tüm öngörü değerleri ilgili güven aralığı içerisinde yer almaktadır. Öngörüler için elde edilen histogram grafikleri ise Şekil 7.12'de verilmektedir.



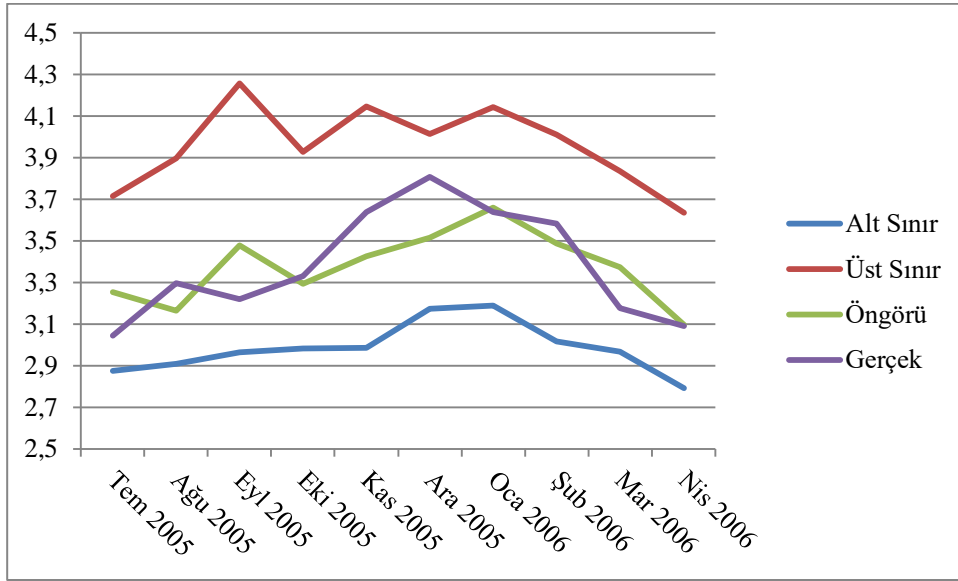
Şekil 7.12. Öngörüler için histogram grafikleri

Öngörüler için elde edilen güven aralıklarının grafiği aşağıda Şekil 7.13'te verilmektedir.



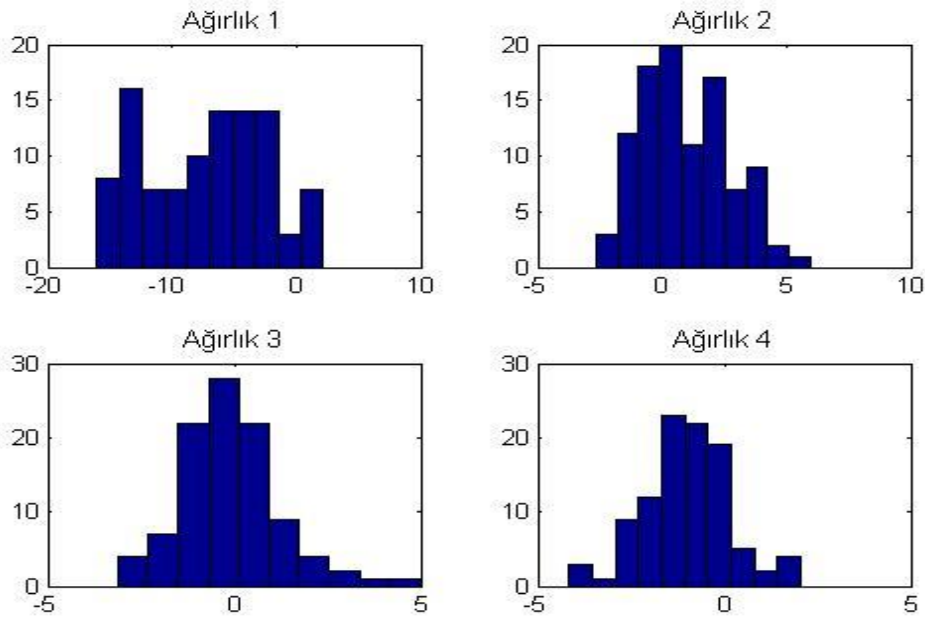
Şekil 7.13. Öngörüler için elde edilen güven aralıklarının grafiği

Grafikten görüldüğü üzere güven aralıkları öngörü değerlerini tam olarak kapsamaktadır. Aykırı değer içeren ANSO verisi için elde edilen öngörüler ve gerçek değerlerin birlikte grafiği ise Şekil 7.14’de verilmektedir.



Şekil 7.14. Öngörü ve gerçek değerlerin birlikte grafiği

Ağırlıklar için elde edilen histogram grafikleri Şekil 7.15’te verilmektedir.



Şekil 7.15. Ağırlıklar için histogram grafikleri

Parametrelerin dağılımı normal dağılım olmadığından; parametrik olmayan işaret testi ile ağırlıkların medyanının sıfıra eşit olduğu yokluk hipotezi test edilmiş, 3. ağırlığa ait p değeri 0.05’ten büyük çıkarken, 1., 2. ve 4. ağırlıklara ait p değerleri 0.05’ten küçük çıkmıştır.

8. SONUÇ ve ÖNERİLER

Bu tez kapsamında zaman serilerinden öngörü elde etmede; yapay sinir ağları yönteminin geleneksel zaman serisi yöntemlerine tercih edilme sebeplerinden ve gösterdiği üstün performanstan söz edilmiştir. Bununla birlikte zaman serilerinde aykırı değerler bulunduğu anda öngörü performansının kötüleşmesi sebebiyle uygulanacak çözüm yöntemleri araştırılmıştır. Bunun için de regresyonda kullanılan dayanıklı yöntemlerden yola çıkılarak, yapay sinir ağlarının performansında önemli bir etkisi olan eğitim algoritmasında geliştirmeler yapılmıştır. Elde edilen yeni eğitim algoritması sayesinde aykırı değer durumunda bile iyi öngörü performansına sahip olabilecek bir yapay sinir ağı elde edilmiştir.

Yöntemi desteklemek amacı ile bir simülasyon çalışması yapılmış, üretilen farklı veriler yardımıyla parametre değerleri de değiştirilerek elde edilen 32 farklı durum için önerilen yöntem ve diğer üç yönteme ait performans karşılaştırması yapılmıştır. Gerçek hayat zaman serilerinde yöntemin öngörü performansını test etmek için ise iki farklı veri seti kullanılarak çözümlenmeler yapılmıştır. Bununla birlikte YSA'larda yalnızca nokta tahmini yapılabilirken, bootstrap yöntemlerle birlikte kullanıldığında güven aralığı tahmini ve hipotez testi yapılarak istatistiksel değerlendirmenin de mümkün olduğu gerçek hayat zaman serisi verileri kullanılarak gösterilmiştir.

Genel olarak YSA'lar öngörü elde etmede kullanım kolaylığı ve yüksek öngörü doğruluğu performansı sebebiyle geleneksel yöntemlere üstünlük sağlarken, veri setinde aykırı değer ya da değerler bulunması durumunda öngörü performansının en az diğer yöntemler kadar kötü etkilendiği görülmüştür. Tezde YSA türleri içerisinde iyi öngörü performansına sahip olduğu literatürdeki çalışmalarla desteklenen çarpımsal nöron model yapay sinir ağları kullanılmış, fakat kullandığı çarpımsal birleştirme fonksiyonu sebebiyle aykırı değerlerden diğer YSA türlerine göre daha çok etkilendiği görülmüştür. Bu durumun önüne geçmek için YSA'nın eğitiminde kullanılan parçacık sürü optimizasyonu eğitim algoritmasında dayanıklı yöntemlerden esinlenilerek farklı amaç fonksiyonları tercih edilmesi yoluna gidilmiştir. Önerilen yöntem sayesinde aykırı değer tespiti yapmadan da

çalıřabilecek ve aykırı deęer bulunması durumunda bile iyi öngörü sonuçları elde edecek, kullanımı kolay bir dayanıklı eęitim algoritması ve YSA türü ortaya konmuřtur.

Verilen tüm durumlar için elde edilen hata deęerlerine bakıldıęında veri setinde aykırı deęer varlıęına bakılmaksızın önerilen yöntem sonuçlarının dięer yöntemlere nazaran çok daha iyi sonuçlar verdięi görölmüřtür. Yapılan benzetim çalıřması ve gerçek hayat zaman serisi çözümlmelerinde hesaplanan hata ölçütleri yardımıyla önerilen yöntemin tüm durumlardaki iyi performansı desteklenmiřtir. Ayrıca kullanılan hareketli blok bootstrap yöntem yardımı ile YSA'da öngörülere ait güven aralıkları elde edilmiř ve istatistiksel deęerlendirme yapılabilmiiřtir. Böylelikle önerilen yöntemin olasılıksal çıkarımlar elde edebilmek için farklı bir kullanımı sunulmuřtur.

Bundan sonraki çalıřmalarda önerilen yönteme benzer řekilde farklı YSA türleri için dayanıklı eęitim algoritmalarının türetilmesi, en küçük budanmıř kareler yerine dięer dayanıklı istatistikleri kullanan YSA eęitim algoritmaları üzerinde çalıřılacaktır. Ayrıca farklı bootstrap yöntemler ile olasılıksal çıkarımların elde edilmesi de dięer bir gelecek planıdır.

KAYNAKLAR

- Aladag C.H., Yolcu U., Egrioglu E., 2013. Robust multilayer neural network based on median neuron model, *Neural Computing and Applications*.
- Aladag C. H., Yolcu U., Egrioglu E., 2013. A new multiplicative seasonal neural network model based on particle swarm optimization. *Neural Processing Letters*, 37(3), 251-262.
- Aladag C. H., Turksen I. B., Dalar A. Z., Egrioglu E., Yolcu U., 2014. Application of type-1 fuzzy functions approach for time series forecasting, *Turkish Journal of Fuzzy Systems*, 1(5), 1-9.
- Allende H., Moraga C., Salas R., 2002. Robust estimator for the learning process in neural networks applied in time series, *Artificial Neural Networks — ICANN 2002 Lecture Notes in Computer Science*, Volume 2415, pp 1080-1086.
- Allende H., Salas R., Moraga C., 2003. A Robust and Effective Learning Algorithm for Feedforward Neural Networks Based on the Influence Function, *Pattern Recognition and Image Analysis Lecture Notes in Computer Science*, Volume 2652, pp28-36.
- Alpaslan F., Egrioglu E., Aladag C. H., Ilter D., Dalar, A. Z., 2014. Comparison of single multiplicative neuron artificial neural network models using abc and bp training algorithms, *Anadolu University Journal of Science and Technology*.
- Beliakov G., Kelarev A., Yearwood J., 2011. Robust artificial neural networks and outlier detection, *Technical report*, arXiv:1110.0169v1 [math.OC] , 2 October.
- Bühlmann P., 1997. Sieve *Bootstrap for Time series*, *Bernoulli.*, 3, 123-148.
- Broş A.G., Pitas I., 1996. Median Radial Basis Function Neural Network, *IEEE Transaction on Neural Networks*, 7(6), 1351-1364.
- Cao F., Ye H., Wang D., 2015. A probabilistic learning algorithm for robust modeling using neural networks with random weights, *Information Sciences*, 313, 62–78.
- Carlstein E., 1992. Resampling techniques for stationary time-series: some recent developments, *New Directions in time series Analysis.*, 75–85.
- Chaturvedi K. T., Pandit M., Srivastava L., 2009. Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch, *Electrical Power and Energy Systems*, 31, 249–257.
- Chen D. S., Jain R. C., 1994. A robust back propagation learning algorithm for function approximation, *IEEE Transactions on Neural Networks*, Vol.5, No. 3, May.
- Chuang C. C., Hsiao C. C., Su, S. F., 2000. The annealing robust backpropagation learning algorithm, *IEEE Trans. Neural Networks* 11(5), 1067-1077.

- Chuang C.-C., Su S.-F., Hsiao C.-C., 2000. The annealing robust backpropagation (ARBP) learning algorithm, *IEEE Transactions on Neural networks*, Vol. 11, No. 5, September.
- Chuang C., Su S., Hsiao C., 2000. The annealing robust backpropagation (ARBP) learning algorithm, *IEEE Trans Neural Netw*, 11:1067–1076.
- Chuang C.-C., Jeng J.-T., Hsiao C.-C., 2002. Constructing an influence function of robust learning algorithm based on error distributions for the neural networks, *IEEE Transactions on Neural Networks*.
- Chuang C.-C., Jeng J.-T., Lin P.-T., 2004. Annealing robust radial basis function networks for function approximation with outliers, *Neurocomputing*, 56, 123 – 139.
- Chuang C.-C., Jeng J.-T., 2007. CPBUM neural networks for modeling with outliers and noise, *Applied Soft Computing*, 7, 957–967.
- Cordeiro C., Neves, M., 2006. The bootstrap methodology in time series forecasting, *Proceedings of CompStat2006* (J. Black and A. White, Eds.), Springer Verlag, 1067-1073.
- De Gooijer J.G., Kumar, K., 1992. Some recent developments in non-linear time series modelling, testing, and forecasting, *International Journal of Forecasting*, 8, 135–156.
- Efron B., 1979. Bootstrap methods: another look at the Jackknife, the annals of statistics., 7, 1–26.
- Efron B., Tibshirani R., 1988. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy, *Statist. Sci.*, vol. 1, pp. 54-77.
- Eğrioğlu E., Aladağ Ç.H. ve Günay S., 2008. A new model selection strategy in artificial neural network, *Applied Mathematics and Computation* 195, 591-597.
- Eğrioğlu E., Aladağ Ç. H., Yolcu U., Çorba B. Ş., Çağcağ O., 2013. Fuzzy time series method based on multiplicative neuron model and membership values, *American Journal of Intelligent Systems*, 3(1), 33-39.
- El-Melegy M. T., Essai M. H., Ali A. A., 2009. Robust training of artificial feedforward neural networks, *Foundations of Computational, Intelligence Volume 1 Studies in Computational Intelligence*, Volume 201, pp 217-242.
- Engle R.F., 1982. Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation, *Econometrica*, 50, 987–1008.
- Espinoza A. V. P., Ordieres M. J. B., Martinez P. F. J., Gonzalez M. A., 2005. TAO-robust backpropagation learning algorithm, *Neural Networks*, 18, 191–204.
- Fu Y.-Y., Wu C.-J., Jeng J.-T., 2009. Chia-Nan Ko, Identification of MIMO systems using radial basis function networks with hybrid learning algorithm, *Applied Mathematics and Computation*, 213, 184–196.
- Fu Y.-Y., Jeng J.-T., Ko C.-N., 2010. ARFNNs with SVR for prediction of chaotic time series with outliers, *Expert Systems with Applications*, 37, 4441-4451.
- Fu Y.-Y., Wu C.-J., Ko C.-N., Jeng J.-T., 2011. Radial basis function networks with hybrid learning for system identification with outliers, *Applied Soft Computing*, 11, 3083–3092.

- Granger C.W.J., Anderson, A.P., 1978. An Introduction to Bilinear Time Series Models, Vandenhoeck and Ruprecht, Göttingen.
- Gundogdu O, Egrioglu E., Aladag C. H., Yolcu U., 2015. Multiplicative neuron model artificial neural network based on gauss activation function, *Neural Computing and Application*, 1(1), 1-15.
- Günay S., Eğrioğlu E., Aladağ Ç.H., 2007. Tek değişkenli zaman serileri analizine giriş, *Hacettepe Üniversitesi Yayınları*, 230s, Ankara.
- Hampel F. R. (1975). Beyond location parameters: robust concepts and methods, *Bulletin of the ISI*, 46, 375–391.
- Hampel F. R., 1968. Contributions to the theory of robust estimation, Unpublished Ph.D. dissertation, University of California, Berkeley.
- Hampel F.R., Ronchetti E.M., Rousseeuw P.J., Stahel W.A., 1986. Robust statistics the approach based on influence functions, *Wiley*, New York.
- Hans A., Udluft S., 2010. Ensembles of neural networks for robust reinforcement learning, *Ninth International Conference on Machine Learning and Applications*.
- 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.
- Hippert H. S., Pedreira C. E., Souza R. C. 2001. Neural networks for short term load forecasting: a review and evaluation, *IEEE Transaction on Power Systems*, 16(1), 44-55.
- Hsiao C.-C., Chuang C.-C., Jeng J.-T., 2012. Robust back propagation learning algorithm based on near sets, *International Conference on System Science and Engineering*, Dalian, China June 30-July 2.
- Hu M.J.C., 1964. Application of the adaline system to weather forecasting. Master Thesis, Technical Report 6775-1, *Stanford Electronic Laboratories*, Stanford, CA, June.
- Hu L., Sun Z., 2005. A robust learning algorithm for feedforward neural networks with adaptive spline activation function, *Advances in Neural Networks – ISNN 2005 Lecture Notes in Computer Science*, Volume 3496, pp 566-571.
- Huber P. J., 1981. *Robust statistics*, Wiley, New York.
- Huber P. J., 1964. Robust estimation of a location parameter, *Annals of mathematical statistics*, 35, 73-101.
- Itamiya K., Sawada M., Kikuta D., Tod K., 2013. A hybrid measurement system of three dimensional coordinates by combination of a multi-link manipulator and particle swarm optimization techniques, *American Journal of Intelligent Systems*, 3(2), 51-56.
- Jing X., 2011. An H_∞ control approach to robust learning of feedforward neural networks, *Neural Networks*, 24, 759–766.
- Jing X., 2012. Robust adaptive learning of feedforward neural networks via LMI optimizations, *Neural Networks*, 31, 33–45.
- Juanyin Q., Wei W., Pan W., 2004. Robust learning of neural networks ensemble for modeling, *Proceedings of the Sm World Congress on Intelligent Control and Automation*, Hangzhou, P.R. China, June 15-19.

- Kennedy J., Eberhart R., 1995. Particle swarm optimization, *In Proceedings of IEEE International Conference on Neural Networks*, 4, 1942–1948, Piscataway, NJ, USA.
- Ko C.-N., 2012. WSVR-based fuzzy neural network with annealing robust algorithm for system identification, *Journal of the Franklin Institute*, 349, 1758–1780.
- Kumar M., Stoll R., Stoll N., 2006. Deterministic approach to robust adaptive learning of fuzzy models, *IEEE Transactions on Systems, Man, and Cybernetics-Part b: Cybernetics*, Vol. 36, No. 4, August.
- Künsch H., 1989. The Jackknife and the bootstrap for general stationary observations, *The Annals of Statistics.*, 17, No.3, 1217–1241.
- Lahiri S. N., 2003. Resampling methods for dependent data, *Springer*, 374p., NewYork.
- Lahnajarvi J. J. T., Lehtokangas M. I. ve Saarinen J. P. P., 2002. Evaluation of constructive neural networks with cascaded architectures. *Neurocomputing* 48, 573-607.
- Lee C. C., Chung P.C., Tsai, J.R., Chang C.I., 1999. Robust radial basis function neural networks, *IEEE Transaction on systems, Man and Cybernetics: Part B: Cybernetics* 29(6), 674-685.
- Lee C.-C., Tsai C.-L., Chiang Y.-C., 2009. Federal funds rate prediction: a comparison between the robust rbf neural network and economic models, *Journal of Information Science and Engineering*, 25,763-778.
- Liano K., 1996. Robust error measure for supervised neural network learning with outliers, *IEEE Transactions on Neural Networks*, Vol. 7, No. 1, January.
- Liu R. Y. ve Singh K., 1992. Moving blocks jackknife and bootstrap capture weak dependence, *Exploring the Limits of Bootstrap.*, Wiley New York, 225–248.
- Ma Y., Jiang C., Hou Z. ve Wang C., 2006. The formulation of the optimal strategies for the electricity producers based on the particle swarm optimization algorithm, *IEEE Trans Power Syst*, 21(4), 1663–71.
- Majhi B., Panda G., 2011. Robust identification of nonlinear complex systems using low complexity ANN and particle swarm optimization technique, *Expert Systems with Applications*, 38, 321–333.
- Majhi B., Rout M., Ritanjali M., Panda G., Fleming P.J., 2012. New robust forecasting models for exchange rates prediction, *Expert Systems with Applications*, 39, 12658-12670.
- Man Z., Lee K., Wang D., Cao Z., Miao C., 2011. A new robust training algorithm for a class of single-hidden layer feedforward neural networks, *Neurocomputing*, 74, 2491–2501.
- Marquez L., Hill, T., O'Connor, M., Remus, W., 1992. Neural network models for forecast a review. In: *IEEE proceedings of the 25th Hawaii International Conference on System Sciences*, 4, pp. 494–498.
- McCulloch W. S., ve Pitts W., 1943. A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 5, 115–133.

- Murata N., Yoshizawa S. ve Amari S., 1994. Network information criterion-determining the number of hidden units for an artificial neural network model, *IEEE Transacion on Neural Networks* 5, 865-872.
- Neubauer A., 1995. Robust learning algorithms for multi-layer perceptrons with discretized synaptic weights, *IEEE International Conference*, Dept. Of Commun. Eng., Duisburg Gerhard-Mercator-Univ., Germany.
- Ninomiya H., 2011. Robust training of multilayer neural networks using parameterized online quasi-newton algorithm, *10th International Conference on Machine Learning and Applications*.
- Politis D.N. ve Romano J.P., 1992. A circular block-resampling procedure for stationary data, *Exploring the Limits of Bootstrap.*, Wiley New York, 263–270.
- Politis D.N. ve Romano J.P., 1994. The stationary bootstrap, *Journal of the American Statistical Association.*, 89, No. 428, 1303–1313.
- Rathbun T.F., Rogers S.K., DeSimio M.P. ve Oxley M.E., 1997. MLP iterative construction algorithm, *Neurocomputing*, 17 (3-4), 195-216.
- Reed R., 1993. Pruning algorithms a survey, *IEEE Transactions on Neural Networks* 4, 740-747.
- Rousseeuw P.J., 1984. Least median of squares regression, *J Am Stat Assoc*, 79:871–880.
- Rousseeuw P. J., Leroy, A. M., 1987. *Robust regression & outlier detection*. New York: Wiley.
- Rousseeuw P. J., Van Driessen K., 2006. Computing LTS regression for large data sets, *Data Mining and Knowledge Discovery*, 12:29-45.
- Roy A., Kim L.S. ve Mukhopadhyay S., 1993. A polynomial time algorithm for the construction and training of a class of multilayer perceptrons, *Neural Networks*, 6, 535-545.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by backpropagating errors, *Nature*, 323 (6188), 533-536.
- Rusiecki A., 2007. Robust LTS backpropagation learning algorithm, *In Computational and ambient intelligence*, pp. 102–109, Berlin, Heidelberg: Springer.
- Rusiecki A., 2008. Robust MCD-based backpropagation learning algorithm. In: Rutkowski et al. (eds) *ICAISC LNCS (LNAI)*, vol 5097., pp 154–163, Springer, New York.
- Rusiecki A., 2012. Robust learning algorithm based on iterative least median of squares, *Neural Process Lettters*, 36:145–160.
- Rusiecki A., 2013. Robust learning algorithm based on LTA estimator, *Neurocomputing*, 120, 624–632.
- Samanta B., 2011. Prediction of chaotic time series using computational intelligence, *Expert Systems with Applications*, 38(9), 11406-11411.
- Sanchez D. V. A., 1995. Robustization of a learning method for RBF networks, *Neurocomputing*, 9: 85–94.

- Sharda R., 1994. Neural networks for the MS/OR analyst: An application bibliography, *Interfaces*, 24 (2), 116–130.
- Shi Y. ve Eberhart R.C., 1999. Empirical study of particle swarm optimization, *Proc IEEE Int Congr Evol Comput*, 3,101–6.
- Siestema J. ve Dow R., 1988. Neural net pruning – why and how? in: *Proceedings of the IEEE International Conference on Neural Networks 1*, 325-333.
- Smith K. A., Gupta J. N. D., 2002. Neural networks in business: techniques and applications, Idea Group Publishing, Hershey, p. 271.
- Song Q., Xiao J., Soh Y. C., 1999. Robust backpropagation training algorithm for multilayered neural tracking controller, *IEEE Transactions on Neural Networks*, Vol. 10, No. 5, September.
- Sun S., Jin F., 2011. Robust co-training, *Int J Pattern Recognit Artif Intell*, 25(7):1113–1126.
- Tiwari M. K., Chattejee C., 2010a. Uncertainty assessment and ensemble flood forecasting using bootstrap based artificial neural networks, *Journal of Hydrology*, 382, 20-33.
- Tiwari M. K., Chattejee C., 2010b. Development of an accurate and reliable hourly flood forecasting model using wavelet-bootstrap-ANN (WBANN) hybrid approach, *Journal of Hydrology*, 384, 458-470.
- Tong H., Lim, K.S., 1980. Threshold autoregressive, limit cycles and cyclical data, *Journal of the Royal Statistical Society Series, B* 42 (3), 245–292.
- Tsai H.-H., Yu P.-T. 2000. On the optimal design of fuzzy neural networks with robust learning for function approximation, *IEEE transaction on systems, man and cybernetics – Part B: Cybernetics*, 30(1), 217–223.
- Walczak B., 1996. Neural networks with robust backpropagation learning algorithm, *Analytica Chimica Acta* ,322, 21-29.
- Wang Z., Massimo C.D., Tham M.T. ve Morris A.J., 1994. A procedure for determining the topology of multilayer feedforward neural networks, *Neural Networks* 7, 291-300.
- Wang W.-Y., Lee T.-T., Liu C.-L., Wang C.-H., 1997. Function approximation using fuzzy neural networks with robust learning algorithm, *IEEE Transactions on Systems, Man, and Cybernetics-Part b: Cybernetics*, Vol. 27, No. 4, August.
- Werbos P. J., 1974. Beyond regression: new tools for prediction and analysis in the behavioral sciences, Ph. D. Thesis, Harvard University, Cambridge.
- Wilcox Rand R., 2005. Introduction to robust estimation and hypothesis testing, Second Edition, pp. 427-430, Elsevier, United States of America.
- Van Den Bergh F. ve Engelbrech A. P., 2002. A new locally convergent particle swarm optimizer, *Proceedings of IEEE Conference on Systems, Man and Cybernetics*, Hammamet, Tunisia.
- Vuković N., Miljković Z., 2015. Robust sequential learning of feedforward neural networks in the presence of heavy-tailed noise, *Neural Networks*, 63, 31–47.
- Yadav R. N., Kalra P. K., John J., 2007. Time series prediction with single multiplicative neuron model, *Applied Soft Computing*, 7, 1157-1163.

- Yang Z. R., Chen S., 1998. Robust maximum likelihood training of heteroscedastic probabilistic neural networks, *Neural Networks*, 11, 739–747.
- Yamawaki S., 2006. The Study of the Robust Learning Algorithm for Neural Networks, Knowledge-Based Intelligent Information and Engineering Systems, *Lecture Notes in Computer Science* Volume 4252, pp 866-870.
- Yolcu U., Aladag C. H., Egrioglu E., 2013. A new linear & nonlinear artificial neural network model for time series forecasting, *Decision Support System*, 54, 1340-1347.
- Yolcu U., Baş E., Egrioglu E., Aladag C. H., 2015. A new multilayer feedforward network based on trimmed mean neuron model, *Neural Network World* 6/15, 587-602.
- Yu Z.-G., Song S.-M., Duan G.-R., Pei R., 2006. robust adaptive neural networks with an online learning technique for robot control, *Advances in Neural Networks - ISNN 2006 Lecture Notes in Computer Science*, Volume 3972, pp 1153-1159.
- Zhang G., Patuwo B. E., Hu Y. M., 1998. Forecasting with artificial neural networks: The state of the art, *International Journal of Forecasting*, 14, 35-62.
- Zhao W., Chen D., Hu S., 2004. Detection of outlier and a robust BP algorithm against outlier, *Computers and Chemical Engineering*, 28, 1403–1408.
- Zhao L., Yang Y., 2009. PSO-based single multiplicative neuron model for time series prediction, *Expert Systems with Applications*, 36, 2805-2812.

EKLER

EK 1: Tüm Durumlar için Lilliefors Normallik Testi Sonuçları ve Kullanılacak Test Türü

EK 2: Tüm Durumlar için Ölçütlere Göre Kruskal Wallis Testi Sıra Sayıları Ortalamaları

EK 3: Tüm Durumlara Göre Yöntemlerden Elde Edilen Medyan Değerleri

EK 4: AUST Zaman Serisine Ait Durumların Tümü İçin Parametre Değerleri

EK 5: ANSO Zaman Serisine Ait Durumların Tümü İçin Parametre Değerleri

EK 1

| | | HKOKtest | OMYHtest | MHYMtest | HKOKegt | OMYHegt | MHYMegt |
|----------------|------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Durum 1 | Önerilen Yöntem | 0.185 | 0.422 | 0.500 | 0.500 | 0.500 | 0.079 |
| | ÇNM-GY | 0.002 | 0.015 | 0.235 | 0.001 | 0.009 | 0.001 |
| | ÇNM-PSO | 0.032 | 0.009 | 0.205 | 0.001 | 0.001 | 0.288 |
| | ÇNM-GYPSO | 0.494 | 0.500 | 0.168 | 0.500 | 0.053 | 0.500 |
| | Test Türü | KW | KW | F | KW | KW | KW |
| | Test İstatistiği | 296.06 | 226.49 | 7798 | 374.06 | 237.20 | 351.39 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 2 | Önerilen Yöntem | 0.035 | 0.001 | 0.187 | 0.500 | 0.001 | 0.197 |
| | ÇNM-GY | 0.001 | 0.003 | 0.070 | 0.024 | 0.355 | 0.001 |
| | ÇNM-PSO | 0.044 | 0.033 | 0.204 | 0.102 | 0.001 | 0.396 |
| | ÇNM-GYPSO | 0.469 | 0.500 | 0.342 | 0.500 | 0.167 | 0.500 |
| | Test Türü | KW | KW | F | KW | KW | KW |
| | Test İstatistiği | 354.77 | 297.23 | 31972.48 | 373.01 | 180.32 | 372.01 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 3 | Önerilen Yöntem | 0.213 | 0.005 | 0.004 | 0.500 | 0.001 | 0.001 |
| | ÇNM-GY | 0.500 | 0.002 | 0.500 | 0.038 | 0.001 | 0.009 |
| | ÇNM-PSO | 0.002 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.246 | 0.227 | 0.491 | 0.073 | 0.111 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 350.16 | 332.57 | 345.03 | 374.06 | 86.25 | 343.20 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 4 | Önerilen Yöntem | 0.149 | 0.001 | 0.002 | 0.451 | 0.010 | 0.001 |
| | ÇNM-GY | 0.010 | 0.003 | 0.061 | 0.001 | 0.001 | 0.001 |
| | ÇNM-PSO | 0.235 | 0.206 | 0.088 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.009 | 0.500 | 0.103 | 0.500 | 0.045 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 239.21 | 335.87 | 320.51 | 374.06 | 110.65 | 336.14 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 5 | Önerilen Yöntem | 0.268 | 0.500 | 0.034 | 0.500 | 0.001 | 0.500 |
| | ÇNM-GY | 0.001 | 0.001 | 0.002 | 0.001 | 0.001 | 0.002 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.183 | 0.500 | 0.410 | 0.500 | 0.113 | 0.225 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 353.75 | 258.83 | 334.82 | 374.06 | 342.86 | 353.65 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

| | | | | | | | |
|-----------------|------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Durum 6 | Önerilen Yöntem | 0.065 | 0.034 | 0.318 | 0.407 | 0.001 | 0.470 |
| | ÇNM-GY | 0.027 | 0.002 | 0.047 | 0.001 | 0.001 | 0.006 |
| | ÇNM-PSO | 0.140 | 0.001 | 0.001 | 0.001 | 0.001 | 0.358 |
| | ÇNM-GYPSO | 0.259 | 0.500 | 0.500 | 0.459 | 0.020 | 0.025 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 317.85 | 270.84 | 300.23 | 374.06 | 108.36 | 330.27 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 7 | Önerilen Yöntem | 0.113 | 0.500 | 0.210 | 0.017 | 0.001 | 0.500 |
| | ÇNM-GY | 0.057 | 0.221 | 0.350 | 0.006 | 0.500 | 0.062 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.324 | 0.376 | 0.099 | 0.500 | 0.500 | 0.475 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 343.39 | 305.35 | 337.89 | 374.06 | 235.99 | 242.08 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 8 | Önerilen Yöntem | 0.062 | 0.321 | 0.080 | 0.304 | 0.002 | 0.253 |
| | ÇNM-GY | 0.069 | 0.500 | 0.361 | 0.001 | 0.291 | 0.001 |
| | ÇNM-PSO | 0.013 | 0.001 | 0.002 | 0.001 | 0.001 | 0.004 |
| | ÇNM-GYPSO | 0.500 | 0.500 | 0.340 | 0.252 | 0.500 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 340.12 | 326.53 | 337.15 | 374.06 | 247.47 | 271.18 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 9 | Önerilen Yöntem | 0.011 | 0.007 | 0.129 | 0.500 | 0.108 | 0.300 |
| | ÇNM-GY | 0.001 | 0.001 | 0.005 | 0.001 | 0.150 | 0.001 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.500 | 0.001 | 0.500 | 0.500 |
| | ÇNM-GYPSO | 0.274 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 |
| | Test Türü | KW | KW | KW | KW | F | KW |
| | Test İstatistiği | 366.46 | 223.34 | 241.25 | 364.18 | 2254.61 | 356.15 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 10 | Önerilen Yöntem | 0.065 | 0.116 | 0.042 | 0.226 | 0.436 | 0.500 |
| | ÇNM-GY | 0.002 | 0.001 | 0.028 | 0.001 | 0.263 | 0.001 |
| | ÇNM-PSO | 0.105 | 0.001 | 0.043 | 0.001 | 0.001 | 0.074 |
| | ÇNM-GYPSO | 0.284 | 0.166 | 0.500 | 0.312 | 0.286 | 0.017 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 291.11 | 227.17 | 278.71 | 374.06 | 228.44 | 354.42 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

| | | | | | | | |
|-----------------|------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Durum 11 | Önerilen Yöntem | 0.179 | 0.002 | 0.500 | 0.268 | 0.001 | 0.500 |
| | ÇNM-GY | 0.058 | 0.001 | 0.059 | 0.001 | 0.065 | 0.058 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.500 | 0.500 | 0.150 | 0.066 | 0.153 | 0.113 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 341.82 | 298.76 | 339.58 | 374.06 | 299.26 | 335.99 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 12 | Önerilen Yöntem | 0.500 | 0.500 | 0.303 | 0.500 | 0.001 | 0.500 |
| | ÇNM-GY | 0.061 | 0.008 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-PSO | 0.011 | 0.001 | 0.232 | 0.001 | 0.001 | 0.048 |
| | ÇNM-GYPSO | 0.500 | 0.170 | 0.500 | 0.302 | 0.137 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 237.80 | 335.02 | 319.69 | 374.06 | 57.38 | 335.16 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 13 | Önerilen Yöntem | 0.500 | 0.436 | 0.236 | 0.235 | 0.001 | 0.196 |
| | ÇNM-GY | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.002 |
| | ÇNM-GYPSO | 0.409 | 0.500 | 0.500 | 0.156 | 0.500 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 350.64 | 253.62 | 332.60 | 374.06 | 338.61 | 345.20 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 14 | Önerilen Yöntem | 0.008 | 0.389 | 0.345 | 0.281 | 0.104 | 0.500 |
| | ÇNM-GY | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-PSO | 0.052 | 0.001 | 0.001 | 0.001 | 0.001 | 0.048 |
| | ÇNM-GYPSO | 0.197 | 0.500 | 0.500 | 0.492 | 0.500 | 0.485 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 335.87 | 259.56 | 341.93 | 374.06 | 264.29 | 335.47 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 15 | Önerilen Yöntem | 0.015 | 0.304 | 0.400 | 0.147 | 0.001 | 0.500 |
| | ÇNM-GY | 0.081 | 0.056 | 0.006 | 0.003 | 0.099 | 0.292 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.346 | 0.276 | 0.500 | 0.272 | 0.500 | 0.356 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 350.03 | 289.57 | 340.86 | 374.06 | 291.39 | 278.04 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

| | | | | | | | |
|-----------------|------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Durum 16 | Önerilen Yöntem | 0.255 | 0.133 | 0.230 | 0.500 | 0.001 | 0.500 |
| | ÇNM-GY | 0.105 | 0.001 | 0.244 | 0.001 | 0.500 | 0.003 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.141 | 0.500 | 0.077 | 0.001 | 0.270 | 0.079 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 333.09 | 273.78 | 323.93 | 374.06 | 230.40 | 277.88 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 17 | Önerilen Yöntem | 0.500 | 0.001 | 0.500 | 0.500 | 0.001 | 0.183 |
| | ÇNM-GY | 0.018 | 0.109 | 0.049 | 0.001 | 0.001 | 0.004 |
| | ÇNM-PSO | 0.104 | 0.001 | 0.001 | 0.001 | 0.001 | 0.010 |
| | ÇNM-GYPSO | 0.500 | 0.500 | 0.500 | 0.014 | 0.500 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 314.53 | 158.64 | 241.07 | 374.06 | 190.37 | 353.04 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 18 | Önerilen Yöntem | 0.064 | 0.428 | 0.500 | 0.500 | 0.161 | 0.312 |
| | ÇNM-GY | 0.001 | 0.500 | 0.001 | 0.001 | 0.002 | 0.001 |
| | ÇNM-PSO | 0.005 | 0.001 | 0.426 | 0.001 | 0.154 | 0.264 |
| | ÇNM-GYPSO | 0.334 | 0.500 | 0.303 | 0.500 | 0.500 | 0.097 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 270 | 300.13 | 319.44 | 373.34 | 281.24 | 371.45 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 19 | Önerilen Yöntem | 0.359 | 0.213 | 0.418 | 0.273 | 0.001 | 0.003 |
| | ÇNM-GY | 0.003 | 0.120 | 0.146 | 0.031 | 0.007 | 0.105 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.500 | 0.500 | 0.124 | 0.340 | 0.282 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 355.07 | 229.52 | 338.88 | 374.06 | 211.40 | 353.43 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 20 | Önerilen Yöntem | 0.359 | 0.213 | 0.418 | 0.273 | 0.001 | 0.003 |
| | ÇNM-GY | 0.003 | 0.120 | 0.146 | 0.031 | 0.007 | 0.105 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.500 | 0.500 | 0.124 | 0.340 | 0.282 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 355.07 | 229.52 | 338.88 | 374.06 | 211.40 | 353.43 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

| | | | | | | | |
|-----------------|------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Durum 21 | Önerilen Yöntem | 0.001 | 0.071 | 0.500 | 0.118 | 0.237 | 0.500 |
| | ÇNM-GY | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.008 |
| | ÇNM-GYPSO | 0.247 | 0.500 | 0.500 | 0.500 | 0.500 | 0.228 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 355.31 | 261.88 | 333.18 | 374.06 | 301.64 | 349.25 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 22 | Önerilen Yöntem | 0.003 | 0.023 | 0.078 | 0.500 | 0.206 | 0.500 |
| | ÇNM-GY | 0.001 | 0.001 | 0.001 | 0.001 | 0.393 | 0.001 |
| | ÇNM-PSO | 0.137 | 0.001 | 0.115 | 0.001 | 0.001 | 0.456 |
| | ÇNM-GYPSO | 0.500 | 0.500 | 0.323 | 0.295 | 0.430 | 0.144 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 343.69 | 328.09 | 337.18 | 374.06 | 344.47 | 345.30 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 23 | Önerilen Yöntem | 0.018 | 0.224 | 0.500 | 0.500 | 0.500 | 0.372 |
| | ÇNM-GY | 0.500 | 0.256 | 0.500 | 0.082 | 0.053 | 0.500 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.500 | 0.500 | 0.078 | 0.209 | 0.500 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 351.09 | 322.38 | 341.30 | 374.06 | 330.47 | 319.47 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 24 | Önerilen Yöntem | 0.077 | 0.022 | 0.049 | 0.279 | 0.001 | 0.237 |
| | ÇNM-GY | 0.004 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.003 | 0.001 | 0.001 | 0.104 |
| | ÇNM-GYPSO | 0.500 | 0.500 | 0.500 | 0.500 | 0.254 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 337.99 | 324.68 | 329.50 | 374.06 | 302.70 | 248.98 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 25 | Önerilen Yöntem | 0.030 | 0.500 | 0.500 | 0.397 | 0.370 | 0.086 |
| | ÇNM-GY | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-PSO | 0.123 | 0.001 | 0.332 | 0.001 | 0.001 | 0.213 |
| | ÇNM-GYPSO | 0.500 | 0.301 | 0.500 | 0.500 | 0.241 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 307.65 | 336.37 | 274.70 | 374.06 | 273.51 | 354.33 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

| | | | | | | | |
|-----------------|------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Durum 26 | Önerilen Yöntem | 0.041 | 0.019 | 0.294 | 0.089 | 0.337 | 0.340 |
| | ÇNM-GY | 0.004 | 0.162 | 0.070 | 0.001 | 0.001 | 0.001 |
| | ÇNM-PSO | 0.261 | 0.001 | 0.419 | 0.001 | 0.036 | 0.016 |
| | ÇNM-GYPSO | 0.500 | 0.500 | 0.500 | 0.405 | 0.500 | 0.500 |
| | Test Türü | KW | KW | F | KW | KW | KW |
| | Test İstatistiği | 258.21 | 299.04 | 390004.90 | 373.71 | 274.19 | 371.55 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 27 | Önerilen Yöntem | 0.276 | 0.221 | 0.500 | 0.500 | 0.001 | 0.012 |
| | ÇNM-GY | 0.001 | 0.185 | 0.005 | 0.052 | 0.022 | 0.004 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.500 | 0.039 | 0.500 | 0.010 | 0.308 | 0.114 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 345.35 | 230.47 | 340.98 | 374.06 | 236.47 | 350.12 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 28 | Önerilen Yöntem | 0.054 | 0.500 | 0.122 | 0.087 | 0.001 | 0.500 |
| | ÇNM-GY | 0.149 | 0.288 | 0.017 | 0.001 | 0.031 | 0.003 |
| | ÇNM-PSO | 0.500 | 0.001 | 0.006 | 0.001 | 0.001 | 0.500 |
| | ÇNM-GYPSO | 0.500 | 0.214 | 0.500 | 0.314 | 0.271 | 0.500 |
| | Test Türü | F | KW | KW | KW | KW | KW |
| | Test İstatistiği | 33092.33 | 242.54 | 326.42 | 374.06 | 202.80 | 335.83 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 29 | Önerilen Yöntem | 0.500 | 0.001 | 0.500 | 0.216 | 0.321 | 0.121 |
| | ÇNM-GY | 0.001 | 0.150 | 0.001 | 0.001 | 0.027 | 0.001 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.003 |
| | ÇNM-GYPSO | 0.044 | 0.069 | 0.209 | 0.500 | 0.070 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 353.04 | 226.49 | 320.99 | 374.06 | 310.96 | 354.78 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 30 | Önerilen Yöntem | 0.287 | 0.001 | 0.440 | 0.434 | 0.051 | 0.322 |
| | ÇNM-GY | 0.001 | 0.001 | 0.001 | 0.001 | 0.479 | 0.001 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.014 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.500 | 0.006 | 0.500 | 0.334 | 0.500 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 334.06 | 227.12 | 327.57 | 374.06 | 253.72 | 342.35 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

| | | | | | | | |
|---------------------|------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Durum 31 | Önerilen Yöntem | 0.500 | 0.132 | 0.500 | 0.216 | 0.001 | 0.396 |
| | ÇNM-GY | 0.390 | 0.001 | 0.500 | 0.002 | 0.041 | 0.039 |
| | ÇNM-PSO | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.005 |
| | ÇNM-GYPSO | 0.261 | 0.112 | 0.427 | 0.500 | 0.002 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 348.98 | 336.68 | 345.50 | 374.06 | 243.88 | 327.36 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Durum 32 | Önerilen Yöntem | 0.195 | 0.500 | 0.426 | 0.169 | 0.001 | 0.168 |
| | ÇNM-GY | 0.061 | 0.001 | 0.052 | 0.001 | 0.080 | 0.001 |
| | ÇNM-PSO | 0.002 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| | ÇNM-GYPSO | 0.500 | 0.218 | 0.341 | 0.260 | 0.500 | 0.500 |
| | Test Türü | KW | KW | KW | KW | KW | KW |
| | Test İstatistiği | 337.62 | 285.84 | 335.86 | 374.06 | 241.24 | 247.65 |
| | p değeri | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

EK 2

| | Önerilen Yöntem | ÇNM-GY | ÇNM-PSO | ÇNM-GRYPSO | |
|----------------|------------------------|---------------|----------------|-------------------|---------|
| Durum 1 | HKOKtest | 53.260 | 323.660 | 177.640 | 247.440 |
| | OMYHtest | 50.500 | 255.410 | 237.120 | 258.970 |
| | MHYMtest | 1.157 | 1.665 | 1.624 | 1.638 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 217.940 | 274.400 | 259.160 |
| | MHYMegt | 50.500 | 350.350 | 168.840 | 232.310 |
| Durum 2 | HKOKtest | 50.500 | 163.430 | 239.040 | 349.030 |
| | OMYHtest | 50.500 | 326.160 | 236.580 | 188.760 |
| | MHYMtest | 1.107 | 2.390 | 2.153 | 2.203 |
| | HKOKegt | 350.500 | 250.500 | 51.210 | 149.790 |
| | OMYHegt | 98.030 | 205.850 | 182.510 | 315.610 |
| | MHYMegt | 50.500 | 350.500 | 151.890 | 249.110 |
| Durum 3 | HKOKtest | 50.500 | 346.550 | 163.960 | 240.990 |
| | OMYHtest | 50.500 | 348.100 | 210.770 | 192.630 |
| | MHYMtest | 50.500 | 349.510 | 174.400 | 227.590 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 112.680 | 254.240 | 230.720 | 204.360 |
| | MHYMegt | 50.500 | 350.370 | 179.200 | 221.930 |
| Durum 4 | HKOKtest | 50.500 | 279.510 | 217.120 | 254.870 |
| | OMYHtest | 50.500 | 349.720 | 208.900 | 192.880 |
| | MHYMtest | 50.500 | 341.710 | 218.370 | 191.420 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 103.530 | 269.770 | 224.480 | 204.220 |
| | MHYMegt | 50.500 | 350.250 | 198.990 | 202.260 |
| Durum 5 | HKOKtest | 50.500 | 349.430 | 164.950 | 237.120 |
| | OMYHtest | 50.500 | 304.380 | 212.540 | 234.580 |
| | MHYMtest | 50.500 | 348 | 186.030 | 217.470 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 348.600 | 175.050 | 227.850 |
| | MHYMegt | 50.500 | 350.050 | 166.040 | 235.410 |
| Durum 6 | HKOKtest | 50.500 | 336.850 | 181.820 | 232.830 |
| | OMYHtest | 50.500 | 314.760 | 216.350 | 220.390 |
| | MHYMtest | 50.500 | 331.080 | 224.190 | 196.230 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 131.800 | 149.540 | 262.300 | 258.360 |
| | MHYMegt | 50.500 | 347.550 | 199.010 | 204.940 |

| | | | | | |
|-----------------|----------|---------|---------|---------|---------|
| Durum 7 | HKOKtest | 50.500 | 348.660 | 174.520 | 228.320 |
| | OMYHtest | 50.500 | 334.750 | 199.200 | 217.550 |
| | MHYMtest | 50.500 | 347.020 | 177.840 | 226.640 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 282.520 | 236.160 | 232.820 |
| | MHYMegt | 50.500 | 273.180 | 267.320 | 211 |
| Durum 8 | HKOKtest | 50.500 | 349.950 | 183.060 | 218.490 |
| | OMYHtest | 50.500 | 345.800 | 199.500 | 206.200 |
| | MHYMtest | 50.500 | 349.550 | 187.760 | 214.190 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 295.780 | 229.090 | 226.630 |
| | MHYMegt | 50.500 | 214.810 | 314.910 | 221.780 |
| Durum 9 | HKOKtest | 50.500 | 150.510 | 255.850 | 345.140 |
| | OMYHtest | 53.490 | 224.390 | 255.310 | 268.810 |
| | MHYMtest | 50.500 | 266.790 | 211.950 | 272.760 |
| | HKOKegt | 350.500 | 250.500 | 57.610 | 143.390 |
| | OMYHegt | 4.020 | 6.742 | 6.373 | 6.487 |
| | MHYMegt | 50.500 | 350.260 | 164.030 | 237.210 |
| Durum 10 | HKOKtest | 50.500 | 320.050 | 186.950 | 244.500 |
| | OMYHtest | 50.500 | 266.080 | 243.320 | 242.100 |
| | MHYMtest | 50.500 | 308.230 | 188.060 | 255.210 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 231.880 | 257.140 | 262.480 |
| | MHYMegt | 50.500 | 350.170 | 165.500 | 235.830 |
| Durum 11 | HKOKtest | 50.500 | 349.190 | 177.910 | 224.400 |
| | OMYHtest | 50.500 | 331.340 | 201.950 | 218.210 |
| | MHYMtest | 50.500 | 349.690 | 183.130 | 218.680 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 55.870 | 337.740 | 193.690 | 214.700 |
| | MHYMegt | 50.500 | 349.430 | 190.310 | 211.760 |
| Durum 12 | HKOKtest | 50.500 | 269.520 | 216.060 | 265.920 |
| | OMYHtest | 50.500 | 349.740 | 202.920 | 198.840 |
| | MHYMtest | 50.500 | 342.580 | 207.070 | 201.850 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 137.500 | 261.120 | 205.110 | 198.270 |
| | MHYMegt | 50.500 | 349.700 | 196.490 | 205.310 |
| Durum 13 | HKOKtest | 50.500 | 349.080 | 167.360 | 235.060 |
| | OMYHtest | 50.500 | 301.460 | 223.510 | 226.530 |
| | MHYMtest | 50.500 | 348.170 | 193.040 | 210.290 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 349.180 | 183.080 | 219.240 |
| | MHYMegt | 50.500 | 350.120 | 175.640 | 225.740 |

| | | | | | |
|---------------------|----------|---------|---------|---------|---------|
| Durum 14 | HKOKtest | 50.500 | 349.980 | 195.770 | 205.750 |
| | OMYHtest | 50.500 | 301.540 | 205.140 | 244.820 |
| | MHYMtest | 50.500 | 350.020 | 221.330 | 180.150 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 306.580 | 205 | 239.920 |
| | MHYMegt | 50.500 | 349.970 | 200.670 | 200.860 |
| Durum 15 | HKOKtest | 50.500 | 350.480 | 170.560 | 230.460 |
| | OMYHtest | 50.500 | 325.500 | 201.430 | 224.570 |
| | MHYMtest | 50.500 | 350.470 | 183.620 | 217.410 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 327.250 | 204.590 | 219.660 |
| | MHYMegt | 50.500 | 309.080 | 252.970 | 189.450 |
| Durum 16 | HKOKtest | 50.500 | 345.340 | 180.370 | 225.790 |
| | OMYHtest | 50.500 | 312.900 | 199.870 | 238.730 |
| | MHYMtest | 50.500 | 344.060 | 194.550 | 212.890 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 236.190 | 273.300 | 242.010 |
| | MHYMegt | 50.500 | 224 | 318.930 | 208.570 |
| Durum 17 | HKOKtest | 50.680 | 335.240 | 182.270 | 233.810 |
| | OMYHtest | 78.160 | 269.630 | 222.450 | 231.760 |
| | MHYMtest | 50.500 | 287.240 | 222.170 | 242.090 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 112.080 | 330.010 | 177.680 | 182.230 |
| | MHYMegt | 50.500 | 350.500 | 167.410 | 233.590 |
| Durum 18 | HKOKtest | 50.500 | 207.300 | 231.540 | 312.660 |
| | OMYHtest | 50.500 | 332.220 | 202.580 | 216.700 |
| | MHYMtest | 50.500 | 342.110 | 212.120 | 197.270 |
| | HKOKegt | 350.500 | 250.500 | 50.990 | 150.010 |
| | OMYHegt | 50.500 | 321.620 | 213.230 | 216.650 |
| | MHYMegt | 50.500 | 350.500 | 152.280 | 248.720 |
| Durum 19 | HKOKtest | 50.500 | 350.350 | 165.180 | 235.970 |
| | OMYHtest | 50.500 | 269.540 | 232.770 | 249.190 |
| | MHYMtest | 50.500 | 350.090 | 186.220 | 215.190 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 59.050 | 215.450 | 267.230 | 260.270 |
| | MHYMegt | 50.500 | 350.490 | 167 | 234.010 |
| Durum 20 | HKOKtest | 50.500 | 350.350 | 165.180 | 235.970 |
| | OMYHtest | 50.500 | 269.540 | 232.770 | 249.190 |
| | MHYMtest | 50.500 | 350.090 | 186.220 | 215.190 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 59.050 | 215.450 | 267.230 | 260.270 |
| | MHYMegt | 50.500 | 350.490 | 167 | 234.010 |

| | | | | | |
|---------------------|----------|---------|---------|---------|---------|
| Durum 21 | HKOKtest | 50.500 | 349.610 | 163.800 | 238.090 |
| | OMYHtest | 50.500 | 300.710 | 200.660 | 250.130 |
| | MHYMtest | 50.500 | 347.900 | 189.460 | 214.140 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 250.500 |
| | OMYHegt | 50.500 | 333.010 | 201.900 | 216.590 |
| | MHYMegt | 50.500 | 350.100 | 170.680 | 230.720 |
| Durum 22 | HKOKtest | 50.500 | 350.090 | 177.660 | 223.750 |
| | OMYHtest | 50.500 | 346.590 | 200.940 | 203.970 |
| | MHYMtest | 50.500 | 350.500 | 190.600 | 210.850 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 349.120 | 174.220 | 228.160 |
| | MHYMegt | 50.500 | 350.310 | 175.980 | 225.210 |
| Durum 23 | HKOKtest | 50.500 | 350.500 | 169.440 | 231.560 |
| | OMYHtest | 50.500 | 341.710 | 187.400 | 222.390 |
| | MHYMtest | 50.500 | 350.500 | 182.890 | 218.110 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 347.410 | 195.530 | 208.560 |
| | MHYMegt | 50.500 | 339.570 | 226 | 185.930 |
| Durum 24 | HKOKtest | 50.500 | 348.080 | 180.550 | 222.870 |
| | OMYHtest | 50.500 | 342.900 | 187.090 | 221.510 |
| | MHYMtest | 50.500 | 346.400 | 191.420 | 213.680 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 334.010 | 208.820 | 208.670 |
| | MHYMegt | 50.500 | 237.140 | 295.990 | 218.370 |
| Durum 25 | HKOKtest | 50.500 | 330.810 | 183.440 | 237.250 |
| | OMYHtest | 50.500 | 348.190 | 183.670 | 219.640 |
| | MHYMtest | 50.500 | 314.540 | 201.640 | 235.320 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 316.510 | 214.130 | 220.860 |
| | MHYMegt | 50.500 | 350.500 | 166.130 | 234.870 |
| Durum 26 | HKOKtest | 50.500 | 217.780 | 228.730 | 304.990 |
| | OMYHtest | 50.500 | 331.880 | 205.440 | 214.180 |
| | MHYMtest | 1.232 | 2.842 | 2.745 | 2.734 |
| | HKOKegt | 350.500 | 250.500 | 50.740 | 150.260 |
| | OMYHegt | 50.500 | 316.200 | 208.300 | 227 |
| | MHYMegt | 50.500 | 350.500 | 152.210 | 248.790 |
| Durum 27 | HKOKtest | 50.500 | 349.870 | 174.830 | 226.800 |
| | OMYHtest | 50.500 | 258.470 | 227.670 | 265.360 |
| | MHYMtest | 50.500 | 350.450 | 183.320 | 217.730 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 53.570 | 207.060 | 262.770 | 278.600 |
| | MHYMegt | 50.500 | 350.460 | 170.430 | 230.610 |

| | | | | | |
|---------------------|----------|---------|---------|---------|---------|
| Durum 28 | HKOKtest | 12.936 | 30.305 | 299.003 | 30.006 |
| | OMYHtest | 50.500 | 212.210 | 259.150 | 280.140 |
| | MHYMtest | 50.500 | 344.240 | 218.640 | 188.620 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 70.180 | 190.270 | 279.230 | 262.320 |
| | MHYMegt | 50.500 | 347.930 | 183.750 | 219.820 |
| Durum 29 | HKOKtest | 50.500 | 348.540 | 164.250 | 238.710 |
| | OMYHtest | 50.500 | 262.010 | 250.880 | 238.610 |
| | MHYMtest | 50.500 | 338.570 | 181.260 | 231.670 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 336.460 | 192.010 | 223.030 |
| | MHYMegt | 50.500 | 350.360 | 165.470 | 235.670 |
| Durum 30 | HKOKtest | 50.500 | 347.180 | 184.390 | 219.930 |
| | OMYHtest | 50.500 | 241.700 | 265.900 | 243.900 |
| | MHYMtest | 50.500 | 346.280 | 199.110 | 206.110 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 294.770 | 206.300 | 250.430 |
| | MHYMegt | 50.500 | 349.030 | 176.760 | 225.710 |
| Durum 31 | HKOKtest | 50.500 | 350.500 | 171.800 | 229.200 |
| | OMYHtest | 50.500 | 349.270 | 187.520 | 214.710 |
| | MHYMtest | 50.500 | 350.500 | 176.190 | 224.810 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 291.930 | 226.340 | 233.230 |
| | MHYMegt | 50.500 | 340.550 | 233.700 | 177.250 |
| Durum 32 | HKOKtest | 50.500 | 349.420 | 186.030 | 216.050 |
| | OMYHtest | 50.500 | 323.120 | 202 | 226.380 |
| | MHYMtest | 50.500 | 349.120 | 188.850 | 213.530 |
| | HKOKegt | 350.500 | 250.500 | 50.500 | 150.500 |
| | OMYHegt | 50.500 | 288.760 | 236.390 | 226.350 |
| | MHYMegt | 50.500 | 242.590 | 293.240 | 215.670 |

EK 3

| | Önerilen Yöntem | ÇNM-GY | ÇNM-PSO | ÇNM-GRYPSO | |
|----------------|-----------------|--------|---------|------------|--------|
| Durum 1 | HKOKtest | 1.140 | 1.264 | 1.244 | 1.250 |
| | OMYHtest | 3.126 | 6.203 | 6.130 | 6.205 |
| | MHYMtest | 1.155 | 1.661 | 1.624 | 1.640 |
| | HKOKegt | 3.665 | 3.367 | 3.312 | 3.321 |
| | OMYHegt | 4.287 | 6.094 | 6.207 | 6.176 |
| | MHYMegt | 1.045 | 1.488 | 1.405 | 1.420 |
| Durum 2 | HKOKtest | 1.263 | 1.831 | 1.865 | 1.929 |
| | OMYHtest | 3.143 | 7.976 | 7.746 | 7.616 |
| | MHYMtest | 1.104 | 2.382 | 2.153 | 2.202 |
| | HKOKegt | 6.946 | 6.250 | 6.136 | 6.143 |
| | OMYHegt | 8.364 | 13.787 | 13.637 | 15.126 |
| | MHYMegt | 1.052 | 2.263 | 20.586 | 2.114 |
| Durum 3 | HKOKtest | 11.838 | 1.861 | 17.933 | 1.809 |
| | OMYHtest | 42.803 | 12.893 | 119.874 | 11.902 |
| | MHYMtest | 11.793 | 27.875 | 26.432 | 2.666 |
| | HKOKegt | 48.501 | 43.724 | 43.021 | 4.311 |
| | OMYHegt | 43.497 | 92.696 | 90.665 | 8.932 |
| | MHYMegt | 10.325 | 20.526 | 19.578 | 1.968 |
| Durum 4 | HKOKtest | 1.422 | 3.268 | 3.251 | 3.259 |
| | OMYHtest | 4.747 | 24.393 | 22.314 | 22.189 |
| | MHYMtest | 1.190 | 4.998 | 4.873 | 4.854 |
| | HKOKegt | 9.847 | 8.659 | 8.536 | 8.551 |
| | OMYHegt | 7.059 | 17.179 | 16.427 | 16.265 |
| | MHYMegt | 1.063 | 3.718 | 3.583 | 3.586 |
| Durum 5 | HKOKtest | 1.063 | 1.413 | 1.362 | 1.372 |
| | OMYHtest | 3.300 | 8.680 | 8.461 | 8.492 |
| | MHYMtest | 0.990 | 1.776 | 1.706 | 1.712 |
| | HKOKegt | 4.090 | 3.900 | 3.858 | 3.868 |
| | OMYHegt | 3.881 | 7.870 | 7.399 | 7.465 |
| | MHYMegt | 0.993 | 1.626 | 1.546 | 1.558 |
| Durum 6 | HKOKtest | 1.112 | 2.086 | 2.021 | 2.030 |
| | OMYHtest | 2.854 | 11.986 | 11.599 | 11.601 |
| | MHYMtest | 1.024 | 2.962 | 2.875 | 2.865 |
| | HKOKegt | 7.970 | 7.408 | 7.352 | 7.368 |
| | OMYHegt | 8.010 | 18.828 | 22.830 | 22.813 |
| | MHYMegt | 1.024 | 2.675 | 2.576 | 2.576 |

| | | | | | |
|-----------------|----------|--------|--------|--------|--------|
| Durum 7 | HKOKtest | 1.091 | 2.185 | 2.070 | 2.085 |
| | OMYHtest | 3.222 | 14.898 | 14.089 | 14.147 |
| | MHYMtest | 0.999 | 2.943 | 2.795 | 2.809 |
| | HKOKegt | 5.633 | 5.288 | 5.235 | 5.246 |
| | OMYHegt | 4.150 | 13.233 | 13.058 | 13.023 |
| | MHYMegt | 0.978 | 2.309 | 2.302 | 2.291 |
| Durum 8 | HKOKtest | 1.120 | 3.732 | 3.573 | 3.590 |
| | OMYHtest | 3.143 | 28.543 | 27.035 | 27.065 |
| | MHYMtest | 0.997 | 5.103 | 4.895 | 4.907 |
| | HKOKegt | 11.032 | 10.114 | 10.035 | 10.055 |
| | OMYHegt | 6.687 | 26.459 | 26.028 | 26.014 |
| | MHYMegt | 1.002 | 4.093 | 4.174 | 4.124 |
| Durum 9 | HKOKtest | 1.108 | 3.121 | 3.195 | 3.207 |
| | OMYHtest | 2.918 | 4.690 | 4.770 | 4.782 |
| | MHYMtest | 1.141 | 1.485 | 1.475 | 1.487 |
| | HKOKegt | 3.625 | 3.260 | 3.167 | 3.172 |
| | OMYHegt | 4.074 | 6.727 | 6.372 | 6.476 |
| | MHYMegt | 1.008 | 1.484 | 1.395 | 1.416 |
| Durum 10 | HKOKtest | 1.110 | 1.263 | 1.244 | 1.249 |
| | OMYHtest | 2.924 | 6.202 | 6.127 | 6.120 |
| | MHYMtest | 1.133 | 1.661 | 1.624 | 1.639 |
| | HKOKegt | 3.624 | 3.368 | 3.312 | 3.321 |
| | OMYHegt | 3.942 | 6.158 | 6.207 | 6.215 |
| | MHYMegt | 1.009 | 1.490 | 1.405 | 1.419 |
| Durum 11 | HKOKtest | 1.169 | 1.806 | 1.732 | 1.742 |
| | OMYHtest | 2.287 | 6.584 | 6.391 | 6.402 |
| | MHYMtest | 1.085 | 2.455 | 2.333 | 2.344 |
| | HKOKegt | 4.714 | 4.354 | 4.288 | 4.297 |
| | OMYHegt | 7.784 | 10.444 | 9.833 | 9.875 |
| | MHYMegt | 0.986 | 2.002 | 1.916 | 1.921 |
| Durum 12 | HKOKtest | 1.343 | 3.269 | 3.252 | 3.267 |
| | OMYHtest | 4.173 | 24.475 | 22.319 | 22.280 |
| | MHYMtest | 1.134 | 4.986 | 4.876 | 4.868 |
| | HKOKegt | 9.711 | 8.662 | 8.536 | 8.550 |
| | OMYHegt | 5.751 | 17.385 | 16.429 | 16.398 |
| | MHYMegt | 1.017 | 3.731 | 3.585 | 3.589 |
| Durum 13 | HKOKtest | 1.048 | 1.408 | 1.362 | 1.373 |
| | OMYHtest | 3.132 | 8.704 | 8.461 | 8.460 |
| | MHYMtest | 0.968 | 1.770 | 1.706 | 1.711 |
| | HKOKegt | 4.053 | 3.898 | 3.858 | 3.869 |
| | OMYHegt | 3.487 | 7.850 | 7.399 | 7.446 |
| | MHYMegt | 0.971 | 1.619 | 1.547 | 1.558 |

| | | | | | |
|-----------------|----------|--------|--------|--------|--------|
| Durum 14 | HKOKtest | 1.100 | 2.180 | 2.097 | 2.099 |
| | OMYHtest | 7.677 | 43.780 | 40.960 | 42.083 |
| | MHYMtest | 0.920 | 2.812 | 2.718 | 2.699 |
| | HKOKegt | 8.235 | 7.737 | 7.679 | 7.695 |
| | OMYHegt | 5.294 | 24.694 | 23.068 | 23.456 |
| | MHYMegt | 0.924 | 2.522 | 2.431 | 2.431 |
| Durum 15 | HKOKtest | 1.045 | 2.127 | 1.993 | 2.006 |
| | OMYHtest | 4.549 | 24.626 | 21.711 | 21.965 |
| | MHYMtest | 0.960 | 2.971 | 2.781 | 2.795 |
| | HKOKegt | 5.410 | 5.125 | 5.067 | 5.077 |
| | OMYHegt | 3.597 | 12.965 | 12.177 | 12.213 |
| | MHYMegt | 0.955 | 2.312 | 2.279 | 2.263 |
| Durum 16 | HKOKtest | 1.142 | 3.891 | 3.776 | 3.800 |
| | OMYHtest | 3.068 | 27.669 | 26.874 | 27.049 |
| | MHYMtest | 0.982 | 5.294 | 5.159 | 5.173 |
| | HKOKegt | 11.367 | 10.526 | 10.447 | 10.467 |
| | OMYHegt | 6.107 | 24.821 | 25.037 | 24.860 |
| | MHYMegt | 0.980 | 4.247 | 4.290 | 4.244 |
| Durum 17 | HKOKtest | 5.655 | 6.174 | 6.063 | 6.087 |
| | OMYHtest | 5.183 | 15.759 | 14.803 | 14.923 |
| | MHYMtest | 1.257 | 1.721 | 1.702 | 1.706 |
| | HKOKegt | 17.169 | 15.806 | 15.549 | 15.591 |
| | OMYHegt | 6.976 | 15.377 | 10.937 | 10.816 |
| | MHYMegt | 1.123 | 1.581 | 1.483 | 1.496 |
| Durum 18 | HKOKtest | 12.351 | 17.861 | 17.964 | 18.114 |
| | OMYHtest | 2.818 | 7.415 | 7.122 | 7.148 |
| | MHYMtest | 1.265 | 2.829 | 2.748 | 2.740 |
| | HKOKegt | 66.931 | 60.247 | 59.360 | 59.449 |
| | OMYHegt | 6.067 | 15.756 | 14.986 | 14.946 |
| | MHYMegt | 1.145 | 2.446 | 2.244 | 2.292 |
| Durum 19 | HKOKtest | 11.599 | 17.680 | 16.974 | 17.091 |
| | OMYHtest | 6.250 | 23.287 | 22.641 | 22.857 |
| | MHYMtest | 1.229 | 2.709 | 2.543 | 2.552 |
| | HKOKegt | 44.767 | 40.556 | 39.848 | 39.933 |
| | OMYHegt | 7.140 | 13.598 | 14.440 | 14.402 |
| | MHYMegt | 1.099 | 2.138 | 2.010 | 2.028 |
| Durum 20 | HKOKtest | 11.599 | 17.680 | 16.974 | 17.091 |
| | OMYHtest | 6.250 | 23.287 | 22.641 | 22.857 |
| | MHYMtest | 1.229 | 2.709 | 2.543 | 2.552 |
| | HKOKegt | 44.767 | 40.556 | 39.848 | 39.933 |
| | OMYHegt | 7.140 | 13.598 | 14.440 | 14.402 |
| | MHYMegt | 1.099 | 2.138 | 2.010 | 2.028 |

| | | | | | |
|-----------------|----------|---------|--------|--------|--------|
| Durum 21 | HKOKtest | 10.624 | 13.752 | 13.251 | 13.364 |
| | OMYHtest | 2.599 | 7.646 | 7.203 | 7.370 |
| | MHYMtest | 1.093 | 1.812 | 1.747 | 1.752 |
| | HKOKegt | 38.565 | 36.823 | 36.412 | 36.517 |
| | OMYHegt | 4.288 | 8.892 | 8.474 | 8.492 |
| | MHYMegt | 1.078 | 1.704 | 1.627 | 1.640 |
| Durum 22 | HKOKtest | 10.959 | 20.308 | 19.501 | 19.627 |
| | OMYHtest | 2.461 | 11.527 | 10.946 | 10.943 |
| | MHYMtest | 1.106 | 2.940 | 2.813 | 2.820 |
| | HKOKegt | 76.476 | 71.773 | 71.220 | 71.383 |
| | OMYHegt | 4.545 | 13.323 | 12.570 | 12.701 |
| | MHYMegt | 10.841 | 2.692 | 2.579 | 2.596 |
| Durum 23 | HKOKtest | 10.607 | 20.273 | 18.895 | 19.012 |
| | OMYHtest | 2.945 | 14.668 | 13.634 | 13.767 |
| | MHYMtest | 1.113 | 2.943 | 2.729 | 2.739 |
| | HKOKegt | 50.760 | 47.844 | 47.297 | 47.397 |
| | OMYHegt | 3.386 | 11.597 | 11.013 | 11.045 |
| | MHYMegt | 1.055 | 2.403 | 2.344 | 2.334 |
| Durum 24 | HKOKtest | 11.555 | 36.031 | 34.864 | 35.018 |
| | OMYHtest | 2.843 | 26.364 | 25.292 | 25.476 |
| | MHYMtest | 1.120 | 5.251 | 5.104 | 5.118 |
| | HKOKegt | 105.031 | 96.567 | 95.824 | 96.017 |
| | OMYHegt | 5.953 | 23.670 | 23.019 | 22.977 |
| | MHYMegt | 1.082 | 4.318 | 4.351 | 4.321 |
| Durum 25 | HKOKtest | 11.066 | 12.297 | 12.086 | 12.132 |
| | OMYHtest | 2.629 | 4.025 | 3.709 | 3.768 |
| | MHYMtest | 1.232 | 1.697 | 1.661 | 1.666 |
| | HKOKegt | 33.709 | 31.354 | 30.848 | 30.932 |
| | OMYHegt | 4.402 | 8.296 | 8.076 | 8.054 |
| | MHYMegt | 1.092 | 1.572 | 1.477 | 1.490 |
| Durum 26 | HKOKtest | 11.768 | 17.901 | 17.960 | 18.128 |
| | OMYHtest | 2.603 | 7.421 | 7.124 | 7.133 |
| | MHYMtest | 1.232 | 2.835 | 2.746 | 2.729 |
| | HKOKegt | 66.147 | 60.255 | 59.355 | 59.447 |
| | OMYHegt | 5.624 | 15.596 | 14.982 | 15.028 |
| | MHYMegt | 1.107 | 2.448 | 2.243 | 2.297 |
| Durum 27 | HKOKtest | 11.147 | 17.753 | 16.973 | 17.082 |
| | OMYHtest | 5.809 | 23.114 | 22.642 | 23.107 |
| | MHYMtest | 1.190 | 2.712 | 2.543 | 2.556 |
| | HKOKegt | 44.247 | 40.545 | 39.848 | 39.931 |
| | OMYHegt | 5.817 | 13.435 | 14.439 | 14.533 |
| | MHYMegt | 1.056 | 2.128 | 2.010 | 2.027 |

| | | | | | |
|---------------------|----------|--------|--------|--------|--------|
| Durum 28 | HKOKtest | 13.086 | 30.307 | 29.905 | 29.985 |
| | OMYHtest | 6.124 | 43.355 | 44.799 | 45.214 |
| | MHYMtest | 1.184 | 4.870 | 4.722 | 4.697 |
| | HKOKegt | 88.504 | 79.248 | 78.099 | 78.242 |
| | OMYHegt | 8.877 | 25.661 | 27.730 | 27.470 |
| | MHYMegt | 1.076 | 3.715 | 3.592 | 3.609 |
| Durum 29 | HKOKtest | 10.251 | 12.881 | 12.458 | 12.576 |
| | OMYHtest | 3.280 | 13.769 | 13.330 | 13.158 |
| | MHYMtest | 1.098 | 1.754 | 1.682 | 1.697 |
| | HKOKegt | 36.742 | 35.489 | 35.089 | 35.195 |
| | OMYHegt | 3.735 | 7.995 | 7.591 | 7.657 |
| | MHYMegt | 1.061 | 1.635 | 1.559 | 1.576 |
| Durum 30 | HKOKtest | 10.936 | 19.892 | 19.189 | 19.282 |
| | OMYHtest | 2.711 | 10.999 | 11.128 | 11.095 |
| | MHYMtest | 1.083 | 2.938 | 2.839 | 2.843 |
| | HKOKegt | 73.287 | 69.541 | 68.995 | 69.153 |
| | OMYHegt | 6.334 | 20.987 | 20.280 | 20.546 |
| | MHYMegt | 1.073 | 2.655 | 2.543 | 2.566 |
| Durum 31 | HKOKtest | 10.383 | 20.239 | 18.766 | 18.873 |
| | OMYHtest | 2.488 | 15.166 | 13.887 | 14.024 |
| | MHYMtest | 1.077 | 2.975 | 2.750 | 2.764 |
| | HKOKegt | 50.870 | 48.204 | 47.648 | 47.754 |
| | OMYHegt | 3.102 | 11.865 | 11.648 | 11.654 |
| | MHYMegt | 1.034 | 2.402 | 2.360 | 2.346 |
| Durum 32 | HKOKtest | 10.756 | 34.447 | 32.839 | 33.012 |
| | OMYHtest | 2.515 | 25.066 | 24.143 | 24.269 |
| | MHYMtest | 1.102 | 5.296 | 5.059 | 5.083 |
| | HKOKegt | 99.999 | 93.149 | 92.361 | 92.553 |
| | OMYHegt | 4.369 | 22.212 | 22.000 | 21.953 |
| | MHYMegt | 1.059 | 4.187 | 4.208 | 4.183 |

EK 4

| | | AUST | | |
|--------------------------------|------------------|-------------------|------------|------------|
| | | | <i>i=1</i> | <i>i=2</i> |
| Orjinal | Durum 1 | $\frac{w_i}{b_i}$ | 0.7194 | 0.3820 |
| | | | 7.0188 | -0.1993 |
| 5 Kat Aykırı Değer | Durum 2.1 | $\frac{w_i}{b_i}$ | 0.5256 | 3.5883 |
| | | | 3.1633 | -1.0699 |
| | Durum 2.2 | $\frac{w_i}{b_i}$ | -0.5628 | 4.5208 |
| | | | 2.5958 | -1.3449 |
| | Durum 2.3 | $\frac{w_i}{b_i}$ | 0.3385 | -1.3945 |
| | | | -8.1477 | 0.4222 |
| 10 Kat Aykırı Değer | Durum 3.1 | $\frac{w_i}{b_i}$ | -0.0643 | 29.4390 |
| | | | 0.7596 | -5.4828 |
| | Durum 3.2 | $\frac{w_i}{b_i}$ | -0.3441 | 16.1999 |
| | | | 1.4229 | -2.9603 |
| | Durum 3.3 | $\frac{w_i}{b_i}$ | 1.4348 | -6.3612 |
| | | | -3.6271 | 1.1669 |

EK 5

| | | ANSO | | | | |
|----------------------------|------------------|-------------------|------------|------------|------------|------------|
| | | | <i>i=1</i> | <i>i=2</i> | <i>i=3</i> | <i>i=4</i> |
| Orijinal | Durum 1 | $\frac{w_i}{b_i}$ | -0.1813 | 0.6252 | -0.5440 | 1.2893 |
| | | | 0.9491 | 4.2469 | 0.5254 | -1.2234 |
| 5 Kat Aykırı Değer | Durum 2.1 | $\frac{w_i}{b_i}$ | -1.6739 | -0.7277 | 8.7635 | -0.1480 |
| | | | 0.3588 | 2.8517 | -2.0933 | 2.3752 |
| 10 Kat Aykırı Değer | Durum 3.1 | $\frac{w_i}{b_i}$ | -8.8146 | 0.5646 | 3.2638 | -0.1858 |
| | | | 2.0427 | 1.9246 | -0.7290 | 1.7472 |

ÖZGEÇMİŞ

Adı Soyadı : Özge GÜNDOĞDU

Doğum Yeri ve Tarihi : Sandıklı/AFYON 10.05.1988

Adres : Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Fakültesi
Ekonometri Bölümü Kampüs/SİVAS

E-Posta : ogundogdu@cumhuriyet.edu.tr

Lisans : Ondokuz Mayıs Üniversitesi /İstatistik

Yüksek Lisans : Süleyman Demirel Üniversitesi /Ekonometri

Mesleki Deneyim ve Ödüller : Araştırma Görevlisi