

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AŞIRI ÖĞRENME MAKİNELERİNİN SEYREK GERİ
ÇATMA ALGORİTMALARI İLE OPTİMİZASYONU

Ömer Faruk ALÇİN

Doktora Tezi

Anabilim Dalı: Elektrik Elektronik Mühendisliği

Danışman: Doç. Dr. Melih Cevdet İNCE

Prof. Dr. Abdulkadir ŞENGÜR

ELAZIĞ-2015

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AŞIRI ÖĞRENME MAKİNALARININ SEYREK GERİ ÇATMA ALGORİTMALARI İLE
OPTİMİZASYONU

DOKTORA TEZİ
Ömer Faruk ALÇİN
(11113201)

Tezin Enstitüye Verildiği Tarih : 14.07.2015

Tezin Savunulduğu Tarih : 04.08.2015

Tez Danışmanı : Doç. Dr. Melih Cevdet İNCE (F.Ü.) 
Prof. Dr. Abdulkadir ŞENGÜR (2.Danışman)(F.Ü.) 
Diğer Jüri Üyeleri : Prof. Dr. Hanifi GÜLDEMİR (F.Ü.) 
Doç. Dr. Sami EKİCİ (F.Ü.) 
Doç. Dr. Davut HANBAY (İ.Ü.) 
Yrd. Doç. Dr. Muhammet Fatih TALU (İ.Ü.) 
Yrd. Doç. Dr. Turgay KAYA (F.Ü.) 

AĞUSTOS-2015

ÖNSÖZ

Bu tez çalışması Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı Doktora Programında hazırlanmıştır.

Bana hayatımın her aşamasında maddi manevi desteklerini esirgemeyen aileme ve sevgili eşim Zeynep Mine ALÇİN'e en içten teşekkürlerimi ve şükranlarımı sunarım.

Bu tez çalışmasının her aşamasında her türlü hoşgörü, destek ve bilgisiyle bana katkıda bulunan değerli danışman hocam Doç. Dr. Melih Cevdet İNCE'ye teşekkürlerimi sunarım.

Ayrıca Doktora eğitimine başladığım günden tez çalışmasının hazırlanması ve yazımı süresince karşılaştığım her türlü sorunun çözümü için sürekli destek veren, tecrübelerini benimle paylaşan, bilimsel katkılarıyla bana yardımcı olan ve değer katan danışman hocam Prof. Dr. Abdulkadir ŞENGÜR'e teşekkürü bir borç bilirim.

Doktora eğitimim süresince sürekli desteğini hissettiğim, tez yazım aşamasında bilgi ve görüşlerine başvurduğum saygıdeğer Muzaffer ASLAN'a ve tez yazım aşamasında yardımcı olan Arş. Gör. Ferhat UÇAR'a da ayrıca teşekkür ederim.

Bu doktora tezinin ülkemizde Aşırı Öğrenme Makinaları konusunda çalışma yapacak olanlara katkıda bulunmasını dilerim.

Ömer Faruk ALÇİN
ELAZIĞ- 2015

İÇİNDEKİLER

ÖNSÖZ	I
İÇİNDEKİLER	II
ÖZET	V
SUMMARY	VI
ŞEKİLLER LİSTESİ	VII
TABLolar LİSTESİ	IX
KISALTMALAR LİSTESİ	XI
SİMGELER LİSTESİ	XIII
1. GİRİŞ	1
1.1. Motivasyon ve Literatür	1
1.2. Tezin Amaç ve Kapsamı	5
1.3. Tezin Organizasyonu	6
2. TEORİK ALT YAPI	9
2.1. Moore-Penrose Genelleştirilmiş Ters	9
2.2. Genel Doğrusal Sistemin En Küçük Norm En Küçük Kareler Çözümü	9
2.3. Rastgele TGKİB Ağlar	10
2.4. Eğim-Temelli Öğrenme Algoritması	12
2.5. TGKİB En küçük kareler çözümü	13
2.6. TGKİB ağlar için Öğrenme Algoritması	14
2.7. Optimizasyon	14
2.7.1. Optimizasyon Yöntemlerinin Sınıflandırılması	16
2.8. SGÇ ve Sıkıştırılmalı Algılama	18
2.9. SGÇ Algoritmaları	21
2.9.1. Taban Arayışı Algoritması	21
2.9.2. Yinelemeli Algoritmalar	22
2.9.3. Eşleme Takip Algoritması	23
2.9.4. Dikgen Eşleme Takip Algoritması	24
2.9.5. Kademeli Dikgen Eşleme Takip Algoritması	26
2.9.6. Yinelemeli Sert Eşikleme	27
2.9.7. Sıkıştırılmış Örneklemeli Eşlemeli Takibi Algoritması	29
2.9.8. İleri-Geri Takip Algoritması	30

2.9.9.	Dikgen En Küçük Kareler	32
2.9.	Başarım Kriterleri ve Normalizasyon Yöntemleri.....	34
2.9.1.	Ortalama Karesel Hatanın Karekökü.....	34
2.9.2.	Standart Sapma	35
2.9.3.	Korelasyon Katsayısı	35
2.9.4.	Ortalama Mutlak Hata.....	35
2.9.5.	Ortalama Mutlak Yüzde Hata.....	35
2.9.6.	Sıfır Ortalama Ve Birim Varyansa Göre Normalizasyon	36
3.	YİNELEMELİ SERT EŞİKLEME TABANLI AŞIRI ÖĞRENME MAKİNASI.....	37
3.1.	Giriş	37
3.2.	Önerilen YSE-AÖM yöntemi.....	37
3.2.	Deneyisel Çalışmalar	38
4.	REGRESYON PROBLEMLERİ İÇİN DİKGİN EŞLEME TAKİP ALGORİTMASI TABANLI AŞIRI ÖĞRENME MAKİNASI.....	41
4.1.	Giriş.....	41
4.2.	Geliştirilmiş AÖM Yöntemleri.....	41
4.2.1.	Düzenlenmiş AÖM Yöntemleri	41
4.2.2.	EAR-AÖM Yöntemi.....	42
4.2.3.	En Küçük Açısız Regresyon Elastik Net AÖM Yöntemi.....	42
4.2.4.	TDEB-AÖM Yöntemi	43
4.3.	Önerilen DET-AÖM	43
4.4.	Deneyisel Çalışma	44
5.	YFT-SAÖM: YİNELEMELİ FIRSATÇI TAKİP ALGORİTMALARI TABANLI SEYREK AŞIRI ÖĞRENME MAKİNASI.....	48
5.1.	Giriş.....	48
5.2.	Önerilen YFT-SAÖM Yöntemi.....	48
5.3.	Deneyisel Çalışmalar	49
5.3.1.	YFT-AÖM yöntemlerinin Karşılaştırılması.....	51
5.3.2.	DET algoritmasının EMDSO ve EN ile karşılaştırılması	57

6.	İLERİ-GERİ TAKİP ALGORİTMASI TABANLI SEYREK AŞIRI ÖĞRENME MAKİNESİ.....	59
6.1.	Giriş	59
6.2.	Önerilen İGT-AÖM Yöntemi.....	59
6.3.	Deneyisel Çalışmalar	60
6.3.1.	İGT-AÖM ile AÖM Karşılaştırılması	62
6.3.2.	İGT-AÖM Yönteminin EAR, EMDSO ve EN ile Karşılaştırılması	63
7.	ZAMAN SERİSİ TAHMİNİ İÇİN YENİ BİR YAKLAŞIM: DEK-AÖM	66
7.1.	Giriş	66
7.2.	Önerilen DEK-AÖM Yöntemi	66
7.3.	Deneyisel Sonuçlar	67
7.3.1.	DJIA Zaman Serisi.....	68
7.3.2.	McKey-Glass Zaman Serisi	73
7.3.3.	2-B Doğrusal ve Doğrusal Olmayan Zaman Serisi	78
7.3.4.	Ocean Zaman serisi.....	83
7.3.5.	Sunspot zaman serisi.....	88
8.	SONUÇLAR VE DEĞERLENDİRİLMESİ.....	94
8.1.	Sonuçların Değerlendirilmesi.....	94
8.2.	Öneriler	97
	KAYNAKLAR.....	98
	ÖZGEÇMİŞ	107

ÖZET

Son zamanlarda, Aşırı Öğrenme Makinaları (AÖM) makina öğrenmesi alanında ilgi duyulan bir konu haline gelmiştir. AÖM, Tek Gizli Katmanlı İleri Beslemeli (TGKİB) ağlar için önerilmiş yeni bir öğrenme algoritmasıdır. AÖM öğrenme algoritması iyi genelleme performansı, aşırı hızlı öğrenme yeteneği ve düşük işlem karmaşıklığı gibi avantajlara sahiptir. Bu avantajlarının yanı sıra, AÖM'nin bazı eksiklikleri vardır. Öncelikle, AÖM çıkış ağırlıklarının hesaplanmasında en küçük kareler minimizasyonu kullanıldığından dolayı ezberleme problemi ile karşı karşıyadır. Bir diğer sakınca ise AÖM'nin başarımının gizli katman nöron sayısına bağlı olmasıdır. Ayrıca gizli katman nöron sayısı eğitim veri setinden büyük olması durumunda en küçük kareler çözümünden dolayı AÖM'de tekillik problemi ortaya çıkabilir ve çözüm kararsız bir davranış göstermektedir.

Bu tez çalışmasında, AÖM çıkış ağırlıkları seyrek kabul edilerek, çıkış ağırlıklarının Yinelemeli Fırsatçı Takip (YFT) algoritmaları kullanılarak hesaplatılması amaçlanmıştır. Araştırılan YFT algoritmaları aşağıda belirtilmiştir;

1. Yinelemeli Sert Eşikleme (YSE), Dikgen Eşleme Takip (DET), Sıkıştırılmalı Örneklemeli Eşleme Takip (SÖET) ve Kademeli Dikgen Eşleme Takip (KDET)
2. İleri Geri Takip (İGT)
3. Dikgen En küçük Kareler (DEK).

Önerilen YFT tabanlı AÖM yöntemleri regresyon (grup 1), sınıflandırma (grup 2) ve zaman serisi (grup 3) problemlerine uygulanmıştır. Deneysel çalışmalar, önerilen yöntemlerin tekillik ve ezberleme problemlerine karşı dayanıklı bir AÖM mimarisi elde edildiğini göstermektedir. Ayrıca deneysel sonuçlar en uygun gizli katman nöron sayısı elde edildiğini göstermektedir.

Anahtar Kelimeler: Tek-gizli katmanlı ileri beslemeli ağlar, aşırı öğrenme makinaları, seyreklik, seyrek geri çatma, yinelemeli fırsatçı takip algoritmaları.

SUMMARY

OPTIMIZATION OF EXTREME LEARNING MACHINE WITH SPARSE RECOVERY ALGORITHMS

Recently, the Extreme Learning Machine (ELM) becomes an interesting topic in machine learning area. The ELM has been proposed as a new learning algorithm for Single-Hidden Layer Feed forward Networks (SLFNs). The ELM structure has several advantageous such as good generalization performance, extremely fast learning ability and low computational process. Besides this advantageous, the ELM structure has some drawbacks. Firstly, the ELM encounters over-fitting problems because of using a least squares minimization in calculation of the output weights. Another drawback is that performance of the ELM depends on the number of hidden neurons. On the other hand the ELM may encounter the singularity problem, and its solution may become unstable, when the hidden nodes are greater than the training data.

In this thesis, the output weights, which are considered sparse, have been computed by using Greedy Pursuit (GP) algorithms. The investigated GP algorithms are given as following;

1. Iterative Hard Thresholding (IHT), Orthogonal Matching Pursuit (OMP), Compressive Sampling Matching Pursuit (CoSaMP) and Stagewise Orthogonal Matching Pursuit (StOMP);
2. Forward-Backward Pursuit (FBP);
3. Orthogonal Least-Squares (OLS).

The proposed GP based ELM methods have been applied to regression (group 1), classification (group 2) and time series prediction (group 3) problems. The experimental results show that a robust ELM architecture which is getting over the singularity and over fitting has been obtained by using the proposed methods. The result also show that the number of hidden neurons has been obtained.

Keywords: Single-Layer feedforward network, extreme learning machine, sparsity, sparse recovery, greedy pursuit algorithms.

ŞEKİLLER LİSTESİ

Şekil 2.1.	TGKİB ağ mimarisi	10
Şekil 2.2.	Optimizasyonu akış şeması	15
Şekil 2.3.	Optimizasyon yöntemlerinin sınıflandırılması	17
Şekil 2.4.	Seyrek bir x işareti	19
Şekil 2.5.	Ölçüm işareti	20
Şekil 2.6.	Elde edilmiş işaret ile gerçek işaretin karşılaştırılması	22
Şekil 2.7.	x işaretinin ET algoritması ile elde edilmesi	24
Şekil 2.8.	x işaretinin DET algoritması ile elde edilmesi	25
Şekil 2.9.	x işaretinin KDET algoritması ile elde edilmesi	27
Şekil 2.10.	x işaretinin YSE algoritması ile elde edilmesi	28
Şekil 2.11.	x işaretinin SÖET algoritması ile elde edilmesi	30
Şekil 2.12.	x işaretinin İGT algoritması ile geri çatılması	32
Şekil 2.13.	x işaretinin DEK algoritması ile elde edilmesi	34
Şekil 3.1.	Önerilen YSE-AÖM yöntemi akış şeması	38
Şekil 5.1.	Gizli katman nöron sayısı artışına bağlı DET ve AÖM'nin ortalama OKHK değişimi	57
Şekil 6.1.	Önerilen İGT-AÖM yöntemi akış şeması	60
Şekil 7.1.	Önerilen DEK-AÖM yöntemi	67
Şekil 7.2.	DJIA finansal zaman serisi	68
Şekil 7.3.	DJIA zaman serisi AÖM tahmin eğrisi	69
Şekil 7.4.	DJIA zaman serisi AÖM tahmin hatası eğrisi	70
Şekil 7.5.	DJIA zaman serisi DEK-AÖM tahmin eğrisi	71
Şekil 7.6.	DJIA zaman serisi DEK-AÖM tahmin hatası eğrisi	72
Şekil 7.7.	Mackey-Glass zaman serisi	73
Şekil 7.8.	Mackey-Glass zaman serisi AÖM tahmin eğrisi	74
Şekil 7.9.	Mackey-Glass zaman serisi AÖM tahmin hatası	75
Şekil 7.10.	Mackey-Glass zaman serisi DEK-AÖM tahmin eğrisi	76
Şekil 7.11.	Mackey-Glass zaman serisi DEK-AÖM tahmin hatası eğrisi	77
Şekil 7.12.	2-B DDO zaman serisi	78
Şekil 7.13.	2-B DDO zaman serisi AÖM tahmin eğrisi	80
Şekil 7.14.	2-B DDO zaman serisi AÖM tahmin eğrisi	81
Şekil 7.15.	2-B DDO zaman serisi DEK-AÖM tahmin eğrisi	82
Şekil 7.16.	2-B DDO zaman serisi DEK-AÖM tahmin hatası	83
Şekil 7.17.	Ocean zaman serisi	84

Şekil 7.18. Ocean zaman serisi zaman serisi AÖM tahmin eğrisi	85
Şekil 7.19. Ocean zaman serisi zaman serisi AÖM tahmin hata eğrisi.....	86
Şekil 7.20. Ocean zaman serisi DEK-AÖM tahmin eğrisi	87
Şekil 7.21. Ocean zaman serisi DEK-AÖM tahmin hatası eğrisi	88
Şekil 7.22. Sunspot zaman serisi.....	89
Şekil 7.23. Sunspot zaman serisi AÖM tahmin eğrisi.....	90
Şekil 7.24. Sunspot zaman serisi AÖM tahmin hatası eğrisi.....	91
Şekil 7.25. Sunspot zaman serisi DEK-AÖM tahmin eğrisi	92
Şekil 7.26. Sunspot zaman serisi DEK-AÖM tahmin hatası eğrisi	93

TABLÖLER LİSTESİ

Tablo 2.1.	DET Algoritması.....	25
Tablo 2.2.	KDET algoritması	26
Tablo 2.3.	YSE Algoritması	28
Tablo 2.4.	SÖET algoritması.....	29
Tablo 2.5.	İleri-Geri takip algoritması	31
Tablo 2.6.	DEK algoritması.	33
Tablo 3.1.	Kullanılan veri setleri	38
Tablo 3.2.	YSE-AÖM sonuçları	39
Tablo 3.3.	AÖM ile YSE-AÖM yönteminin karşılaştırılması	40
Tablo 4.1.	Kullanılan veri setleri hakkında özet bilgiler.....	44
Tablo 4.2.	Veri setleri için kullanılan artım ve aralık miktarları.....	45
Tablo 4.3.	DET-AÖM yöntemi sonuçları	46
Tablo 4.4.	DET-AÖM ile YSA, AÖM, EN ve EMDSO ile karşılaştırması	47
Tablo 5.1.	Önerilen YFT-SAÖM yöntemi	50
Tablo 5.2.	Deneylerde kullanılan veri setleri hakkında bilgi.	49
Tablo 5.3.	Veri setleri için kullanılan artım ve aralık miktarları.....	50
Tablo 5.4.	YFT-AÖM yöntemlerinin karşılaştırılması	51
Tablo 5.5.	YFT-SAÖM (OMP) yöntemi ile AÖM'nin karşılaştırma sonuçları.....	52
Tablo 5.6.	Seçilmiş veri setleri hakkında bilgi	58
Tablo 5.7.	YFT-AÖM (DET) yöntemi ile EN ve EMDSO yöntemlerinin karşılaştırma sonuçları.....	58
Tablo 6.1.	Kullanılan veri setlerinin özellikleri.....	61
Tablo 6.2.	İGT-AÖM yöntemi sonuçları (Results of FBP-ELM method).....	62
Tablo 6.3.	İGT-AÖM ile AÖM karşılaştırması.....	63
Tablo 6.4.	İGT-AÖM ile EAR, EMDSO ve Elastik Net karşılaştırma sonuçları	65
Tablo 7.1.	AÖM DJIA zaman serisi Sonuçları.....	69
Tablo 7.2.	DEK-AÖM DJIA zaman serisi sonuçları	70
Tablo 7.3.	DJIA zaman serisi için AÖM ve DEK-AÖM karşılaştırması	71
Tablo 7.4.	Mckey-Glass zaman serisi AÖM sonuçları	74
Tablo 7.5.	Mackey-Glass zaman serisi DEK-AÖM sonuçları	76
Tablo 7.6.	Mackey-Glass zaman serisi için AÖM ve DEK-AÖM karşılaştırması.....	77

Tablo 7.7.	2-B doğrusal ve durağan olmayan zaman serisi AÖM sonuçları	79
Tablo 7.8.	2-B DDO zaman serisi DEK-AÖM sonuçları	80
Tablo 7.9.	2-B DDO zaman serisi için AÖM ve DEK-AÖM karşılaştırması.....	82
Tablo 7.10.	Ocean zaman serisi AÖM sonuçları.....	84
Tablo 7.11.	Ocean zaman serisi DEK-AÖM sonuçları.....	85
Tablo 7.12.	Ocean zaman serisi için AÖM ve DEK-AÖM karşılaştırması	87
Tablo 7.13.	Sunspot zaman serisi AÖM sonuçları	89
Tablo 7.14.	Sunspot zaman serisi DEK-AÖM sonuçları	91
Tablo 7.15.	Sunspot zaman serisi için AÖM ve DEK-AÖM karşılaştırması	92

KISALTMALAR LİSTESİ

2-B DDO	: 2-boyutlu doğrusal ve durağan olmayan
AÖM	: Aşırı Öğrenme Makinası
DDO	: Doğrusal ve Durağan Olmayan
DEK	: Dikgen En küçük Kareler
DET	: Dikgen Eşleme Takip
DJIA	: Dow Jones Borsa Endeksi
DVM	: Destek Vektör Makinaları
EAR	: En küçük açısız regresyon
EB-AÖM	: En uygun Budamalı Aşırı Öğrenme Makinası
EMDSO	: En küçük Mutlak Daralma ve Seçme Operatörü
EN	: Elastik Net (Elastic Net)
ET	: Eşleme Takip
İBYSA	: İleri Beslemeli Yapay Sinir Ağları
İGT	: İleri-Geri Takip
KDET	: Kademeli Eşleme Takip
KİO	: Kısıtlı İzometri Özelliğine
OKHK	: Ortalama Karesel Hatanın Karekökü
OMH	: Ortalama mutlak hata
OMYH	: Ortalama mutlak yüzde hata
SA	: Sıkıştırılmış Algılama
SAÖM	: Seyrek Aşırı Öğrenme Makinası
SEN	: Saf Elastik Net
SGÇ	: Seyrek Geri Çatma
SOBV	: Sıfır ortalama ve birim varyans
SÖET	: Sıkıştırılmış Örneklemeli Eşleme Takip
TA	: Taban Arayışı
TDEB-AÖM	: Tikhonov'un Düzenlenmiş En uygun Budamalı Aşırı Öğrenme Makinası
TGKİB	: Tek-Gizli Katmanlı İleri Beslemeli
YFT	: Yinelemeli Fırsatçı Takip

YFT-SAÖM : Yinelemeli Fırsatçı Takip Algoritmaları tabanlı Seyrek Aşırı
Öğrenme Makinası
YSA : Yapay Sinir Ağları
YSE : Yinelemeli Sert Eşikleme

SİMGELER LİSTESİ

$\hat{w}_i$	: $\hat{w}_i$ 'nin en küçük kareler çözümü
$\bar{y}$	: Hedef verisinin aritmetik ortalaması
$\min \ \cdot\ $	: Minimizasyon operatörü
$(\cdot)^\dagger$	: Moore-Penrose Genelleştirilmiş Ters operatörü
$\ \cdot\ $	: Öklid uzayında norm operatörü
H^*	: Ölçeklenmiş H
Y^*	: Ölçeklenmiş Y
β^*	: Ölçeklenmiş β
$\hat{y}_i$	: Tahmin hedef verisi
$w_i \cdot x_j$	: w_i ve x_j içsel çarpımını ifade
$\hat{x}$	: x 'in en küçük kareler çözümü
$\hat{\beta}$	: β 'nın en küçük kareler çözümü
$\hat{b}_i$	: $\hat{b}_i$ 'nin en küçük kareler çözümü
$\ \cdot\ _0$	: l_0 -normu
$\lambda, \lambda_1, \lambda_2$	: Regülerizasyon parametresi
$(\cdot)^T$	: Transpoz operatörü
(x_i, y_i)	: Veri çifti
A	: Ölçüm matrisi
a_i	: Atom veya A matrisinin i . kolonu
a_t	: Yakınsama vektörü
b_i	: Gizli katman eşik değeri
b_t	: SÖET algoritması güncel katsayı vektörü
c_j	: j . iterasyondaki yaklaşık çözüm/ katsayı vektörü
e	: Gürültü
E	: Maliyet fonksiyonu
G	: A Moore-Penrose Genelleştirilmiş Ters
$g(\cdot)$	: Gizli katman aktivasyon fonksiyonu
H	: Gizli katman çıkış matrisi
I	: Birim matris

J	: ζ değerinden büyük destek vektörü
K	: Seyreklik seviyesi
kk	: Korelasyon katsayısı
m	: İlgili matris satır sayısı
M	: Nöron sayısı
n	: İlgili matris kolon sayısı
N	: Örnek sayısı
O	: Düzeltme sabiti
o_j	: TGKİB ağının çıkışı
P	: Örnek sayısı
r_j	: Artık (rezidü) vektörü
S	: Evrensel Çözüm
s	: x 'i oluşturan katsayı vektörü
S^*	: Evrensel optimal çözüm
t	: Döngü indeksi
T	: YSE algoritması katı eşikleme operatörü
W	: (w_i, β_i) ve b_i 'yi temsil eden vektör
w_i	: Giriş katmanı ağırlıkları
x	: Girdi işareti/verisi
y	: Ölçüm işareti/hedef verisi
Y	: TGKİB ağ çıkışı
z	: SOBV'e göre normalize edilmiş veri
$\acute{a}$	: İleri adım sayısı
α_j	: En yüksek ilişkili atom
β	: Çıkış katmanı ağırlıkları
δ	: KIÖ sabiti
$\acute{e}$	: Hata
ε	: Minimum hata toleransı
η	: Eğim-Temelli öğrenme algoritması öğrenme oranı
$\varkappa$	: Amaç fonksiyonu
λ	: Destek vektörü
$\mathcal{A}$	: İndeks kümesi
$\mathcal{A}_g$	: Geri indeks seti

$\mathcal{A}_i$	: İleri indeks seti
μ	: Geri adım sayısı
σ	: Standart sapma
τ	: Bir optimizasyon çözüm kümesi
Ψ	: Dönüşüm tabanı
Ω	: SÖET algoritması indeksi
ς	: KDET algoritması eşik değeri

1. GİRİŞ

1.1. Motivasyon ve Literatür

Doğrusal olmayan karmaşık eşleşmelere yakınsanabilmesi, klasik parametrik teknikler kullanılarak modellenmesi ve zor olan sistemleri modelleyebilmesi gibi özelliklerinden dolayı İleri Beslemeli Yapay Sinir Ağları (İBYSA) pek çok alanda yaygın bir şekilde kullanılmaktadır. Öte yandan, bu gibi karmaşık modelleme işlemlerinde Yapay Sinir Ağları (YSA) için hızlı öğrenme algoritmaları eksikliği söz konusudur. Geleneksel öğrenme algoritmaları genellikle beklenenden daha yavaş kalmaktadır. Yani, geleneksel metotlar ile bir ağı eğitmek bazen günler hatta haftalar almaktadır [1, 2].

Matematiksel bakış açısıyla İBYSA'nın yakınsama yetenekleri üzerine yapılan araştırmalar kompakt giriş veri seti için evrensel yakınsama ve sınırlı bir eğitim veri seti için yakınsama olmak üzere iki görüş üzerine yoğunlaşmıştır. Pek çok araştırmacı standart çok katmanlı İBYSA'nın evrensel yakınsama yeteneğini keşfetmişlerdir [1, 5].

Hornik eğer aktivasyon fonksiyonu sürekli, sınırlı ve sabit olmayan bir yapıda ise sürekli eşleşmelerin kompakt giriş setleri için bile yakınsanabileceğini kanıtlamıştır [1-3]. Leshno yaptığı çalışmada [4], Hornik'in sonuçlarını geliştirerek polinomal olmayan aktivasyon fonksiyonlu bir ileri beslemeli ağı sürekli aktivasyon fonksiyonu ile yakınsayabileceğini göstermiştir. Pratik uygulamalarda, Yapay Sinir Ağları (YSA) sonlu eğitim veri seti ile eğitilirler [1, 2]. Sınırlı eğitim veri setinde fonksiyon yakınsaması için, Huang ve Bobri N adet gizli düğümlü ve herhangi bir doğrusal olmayan aktivasyon fonksiyonlu TKGİB sinir ağlarının N adet belirgin örnekten tam olarak öğrenebileceğini göstermiştir [6]. Giriş katmanını gizli katmana bağlayan ağırlıkların ve gizli katman eşiklerinin bütün önceki teorik araştırmalarda ve pratik İBYSA öğrenme algoritmalarında olduğu gibi ayarlanması gerekebilir [2]. Aynı zamanda farklı iki katman arasında ağırlık ve eşik gibi parametrelerde bağımlılık söz konusudur [1, 2].

Geçen on yılda, çeşitli İBYSA'larda öğrenme algoritması olarak Eğimli Düşüş tabanlı yöntemler kullanılmıştır. Eğimli Düşüş tabanlı yöntemler uygun olmayan adım aralığında çok yavaş kalabilir veya yerel minimum noktasına kolaylıkla takılabilir [2]. Bu nedenle bazı öğrenme algoritmalarında iyi bir öğrenme başarımı elde etmek için pek çok yinelemeli öğrenme adımları gerekli olabilir [1, 2].

Giriş ağırlıkları ve gizli katman eşikleri rastgele seçilmiş, gizli katmanı M adet nörondan oluşan TGKİB sinir ağına N adet belirgin örnekten tamamen öğrenebildiği [2, 7, 8] nolu çalışmalarda gösterilmiştir. İleri beslemeli ağlarda bütün parametrelerin ayarlanması düşüncesinin aksine giriş ağırlıkları ve birinci gizli katman eşiklerini ayarlamak gerekli olmayabilir. [9] numaralı çalışmada Huang ve diğ. bu yaklaşımı kullanarak bazı benzetim ve pratik uygulamalarda hem hızlı öğrenme yeteneği hem de iyi genelleme başarımı elde etmişlerdir.

TGKİB ağı gizli katmanındaki aktivasyon fonksiyonu sonsuz türevlenebilir ise ağı giriş ağırlıkları ve gizli katman eşik değerlerinin rastgele atanabileceği gösterilmiştir. Giriş ağırlıkları ve gizli katman eşikleri rastgele seçildikten sonra TGKİB ağı doğrusal bir sistem olarak düşünülebilir. Buradan hareketle gizli katmanı çıkış katmanına bağlayan çıkış ağırlıkları gizli katman çıkış matrisinin temel genelleştirilmiş tersi işlemi yardımı ile analitik olarak hesaplanabilmektedir [1, 2, 10]. AÖM olarak isimlendirilen bu yaklaşım TGKİB ağları için yeni bir öğrenme algoritması olarak önerilmiştir [1, 2, 10]. AÖM'nin öğrenme süreci geriye yayılım algoritması gibi geleneksel öğrenme algoritmalarına kıyasla çok daha hızlı öğrenme yeteneğine ve daha iyi bir genelleme başarımına sahiptir [1, 2, 10]. Geleneksel öğrenme algoritmalarından farklı olan bu yeni öğrenme algoritması, hem daha küçük öğrenme hatasına hem de en küçük ağırlık normuna sahiptir [1, 2]. İleri beslemeli ağların genelleme başarımı üzerine öne sürülen Bartlett teoremi en küçük eğitim hatasına sahip en küçük normlu ağırlıklı ağı daha iyi genelleme başarımına sahip olma eğiliminde olduğunu vurgulamaktadır [11]. Böylece, AÖM öğrenme yaklaşımı daha iyi bir genelleme başarımı gösterme eğilimindedir [1]. Buraya kadar anlatılanlar;

- AÖM'nin TGKİB ağları için yeni bir öğrenme algoritması olarak önerildiği,
- Eğimli Düşüş tabanlı öğrenme algoritmasına sahip TGKİB ağların aksine AÖM'de, aktivasyon fonksiyonu sonsuz türevlenebilir olduğunda giriş ağırlıklarının ve gizli katman eşik değerlerinin rastgele seçilebilir ve çıkış ağırlıklarının ise analitik bir yöntemle hesaplanabilir olduğu

şeklinde özetlenebilir [1, 2, 10]. AÖM yapısı gereği hızlı öğrenme yeteneğine, iyi bir genelleme başarımına ve düşük bir işlem karmaşıklığına sahiptir [1, 2, 10]. Belirtilen üstünlüklerinin yanı sıra AÖM'nin çıkış ağırlıklarının, gizli katman matrisinin Moore-Penrose Genelleştirilmiş Tersi işlemi yardımıyla hesaplanması en küçük kareler minimizasyonu işleminden dolayı ağı ezberlemesine yol açabilir [2, 12-14]. Ayrıca eğitim sürecinde kullanılan veri seti eğer tüm veri setinin özelliğini karakterize etmiyorsa AÖM

veri setinde ilgisiz deęişkenlerin var olması problemi ile karşı karşıyadır [12, 15, 16]. Diğer bir dezavantaj ise, AÖM'nin başarımının gizli katman nöron sayısına önemli ölçüde baęlı olmasıdır [12]. AÖM aęının gizli katman nöron sayısı gerekenden az sayıda seçilirse giriş verisi doęru bir şekilde modellenemeyebilir [15]. Tersı durumda yani gizli katman nöron sayısı gerekenden fazla seçildięinde ise AÖM yine ezberleme ile karşı karşıya kalabilir [15]. Ayrıca gizli katman nöron sayısı gerekenden fazla seçildięinde AÖM en küçük kareler çözümünden dolayı tekillik problemi ortaya çıkabilir [15]. Sonuç olarak çözüm kararsız bir hal alabilir. Bahsedilen ilk dezavantajı gidermek için [12] ve [17] numaralı çalışmalarda, eğitim hatasını sınırlandıran L_2 -normu eklenerek, düzenlenmiş bir AÖM yöntemi önerilmiştir. Bu yöntem daha iyi genelleme başarımı sağlarken, AÖM aęının daha karmaşık bir yapıda olmasına yol açar. Huang ve diğerleri, [18] numaralı çalışmalarında yüksek genelleme başarımına sahip bir çekirdek AÖM yöntemi önermişlerdir. Bu yöntem bahsedilen problemin çözümüne fayda sağlamakla birlikte, büyük ölçekli uygulamalarda hesaplama yükü ve matris tersinin hesaplanma zorluğu gibi önemli bir problemi de beraberinde getirmiştir [19, 20]. Miche ve diğerleri tarafından önerilen bir başka yaklaşımda ise, en uygun aę yapısının elde edilmesi için ilk önce gerekenden çok daha fazla sayıda gizli katman nöronu kullanılmış ve daha sonra belirli bir kriteri sağlamayan nöronlar budanarak yüksek genelleme başarımına sahip En uygun Budamalı-AÖM (EB-AÖM) yapısı oluşturulmuştur [12, 21]. EB-AÖM yönteminde fazla nöronların budanması işlemi L_1 -normlu regresyon yöntemi kullanılarak gerçekleştirmektedir [12]. Miche ve diğerleri daha sonra EB-AÖM'ye L_2 -normu ekleyerek L_1 ve L_2 -normlu kademeli yeni bir yöntem ile AÖM'nin başarımını iyileştirmişlerdir [22, 23]. Bu yeni yöntem Tikhonov'un Düzenlenmiş En iyi Budamalı AÖM (TDEB-AÖM) olarak adlandırılmıştır [22, 23]. EB-AÖM ve TDEB-AÖM yöntemleri eğitim hatasının düşürülmesi ve genelleme başarımının iyileştirilmesi gibi iyi özelliklere sahiptir. Fakat bu yöntemler aęın gizli katmanın çok fazla sayıda nörona sahip olmasına ve böylece hesaplama yükünün artmasına sebep olur [19]. En uygun gizli katman nöron sayısı seçimi için önerilmiş bir diğer yöntem ise En küçük Açısıl Regresyon (EAR) tabanlı AÖM yöntemidir [23]. EAR-AÖM olarak isimlendirilen bu yöntemde gizli katman nöron sayısı gerekenden fazla sayıda başlatılır ve L_1 -norm yardımıyla ilgisiz nöronlar elimine edilmektedir. Ayrıca bu yöntem En küçük Mutlak Daralma ve Seçme Operatörü (EMDSO) problemi olarak bilinmektedir [24] ve EAR algoritması L_1 -normunun optimizasyonunda kullanılmaktadır [16, 24]. EAR-AÖM ilgisiz nöronları çok iyi belirlerken, özellikle

seçilmiş düğümler arasında bir korelasyon olduğu durumlarda tekillik problemi söz konusu olabilmektedir [16]. EAR-AÖM yönteminin bu sakıncasının üstesinden gelmek için bir L_2 -norm sınırlaması geliştirilmiştir. Böylece SEN olarak isimlendirilen bu yeni mimari oluşturulmuştur [15, 16]. SEN yönteminde tekillik problemi söz konusu değilken, katsayı büzülmesi ve ayarlanması gereken çok fazla parametre olması gibi sakıncalara sahiptir [16, 25]. Bu sakıncaları gidermek için EAR-EN tabanlı AÖM yöntemi önerilmiştir [16]. EAR-EN-AÖM yöntemi SEN algoritmasının ölçeklenmiş bir versiyonudur [16]. Bu yöntemde iki regülerizasyon parametresinin ayarlanması gerekmektedir. Bu durum EAR-EN-AÖM yönteminin eğitim sürecinin yavaş olmasına yol açmaktadır [15, 16, 22].

Yukarıda bahsedilen düzenlenmiş AÖM yöntemleri, AÖM'nin bazı eksikliklerini gideriyor olsa da en uygun nöron sayısı, tekillik problemi ve hesaplama karmaşıklığı açılarından bakıldığında hala yetersiz kalmaktadır [23, 24, 25].

AÖM öğrenme algoritmasının temelinde, TGKİB ağın doğrusal kabul edilmesi ve bu doğrusal sistemden çıkış ağırlıklarının hesaplanabiliyor olması yatmaktadır. Son zamanlarda doğrusal sistemlerin çözümü için seyreklik yaklaşımı kullanılmaktadır [26]. Seyreklik yaklaşımı, geleneksel yöntemlerin aksine işaretin daha az örnekle elde edilebileceğini önermektedir [27]. Geleneksel Nyquist örnekleme teoremine göre bant sınırlı bir işaretin bilgi kaybı olmayacak şekilde, örneklerinden geri elde edebilmek için işaretin maksimum frekanslı bileşeninin en az iki katı hızında örneklenmesi gerekmektedir [27, 28]. Son çalışmalarda seyrek veya seyrek olarak ifade edilebilen işaretin geri çatmak için işaretin az sayıda doğrusal ölçümü edinilerek yaklaşık kısa temsili alınmaktadır [27-31]. Seyreklik, sürekli zamanlı bir işaret için "bilgi oranının" bant genişliği tarafından önerilenden çok daha küçük olması, ayrık zamanlı bir işaret için ise serbestlik derecesi sayısının işaretin sonlu uzunluğundan çok daha küçük olması olarak ifade edilebilmektedir [27, 28]. Ses, radar ve medikal görüntüleme gibi ölçüm almanın zor ve maliyetli olduğu uygulamalarda işaretlerin yüksek oranlarda sıkıştırılması, gürültüden arındırılması ve gürültüye dayanıklılık işlemlerine tabi tutulması konularında seyreklik önemli bir yer tutmaktadır [27-32]. Sıkıştırılmış Algılama (SA) bir işaretin bilinen bir tabanda seyrek olarak ifade edilebildiği durumlarda Seyrek Geri Çatma (SGÇ) özelliğine sahip, işaret işleme alanına yenilik getirmiş bir teori olmaktadır [27, 28, 32]. SA teorisi herhangi bir alanda seyrek ifade edilebilen bir işaretin geleneksel yöntemlere göre çok daha az sayıda rastgele doğrusal ölçümler kullanılarak geri çatılması işlemine dayanmaktadır [27-32].

SGÇ, eksik-belirlenmiş doğrusal denklemlerin bir seyreklik koşulu altında çözümünü gerektirir [27, 28, 33-36].

Ölçüm matrisinin sıfır uzayı bulunduğundan genellikle tekil bir çözüm elde edilemediği belirtilmiştir [31]. Donoho ve diğerleri ölçüm matrisinin belli şartları sağlaması durumunda esnek bir eşdeğere eşit olduğunu öne sürmüşlerdir [31, 37]. Taban Arayışı (TA) algoritması SGÇ probleminin çözümü için esnek bir optimizasyon yöntemi sunmaktadır [27, 28, 35, 38]. TA algoritması yakınsama oranı çok yüksek olmasına rağmen aşırı işlem karmaşıklığına sahiptir. Yani optimizasyon her adımda tüm vektörler kullanılarak aynı anda yapıldığından işlem karmaşıklığı artmaktadır. Ayrıca bazı araştırmacılar tarafından yapılan çalışmalarda TA algoritmasının yavaş olduğu ileri sürülmüştür [39, 40]. Bu olumsuzlukları gidermek için Yinelemeli Fırsatçı Takip (YFT) algoritmalarının kullanımı önerilmiştir [37-41]. YFT algoritmaları küresel optimum çözümü belirlemek için yerel bir dizi en iyi seçimler yapmaktadır [42]. Konveks gevşeme yöntemleri kombinasyonel seyrek yakınsama problemini ilgili konveks problem ile değiştirir [42]. YFT algoritmaları işareti geri çatismak için tüm vektörleri kullanmak yerine uygun vektörleri kullandığından ve bunların çarpım katsayılarının uygun seçiminden ötürü işlem karmaşıklığı az olan hızlı algoritmalarlardır. Ayrıca yakınsama başarımları da TA algoritmalarının başarımlarına yakındır [29, 31-33, 37-41].

1.2. Tezin Amaç ve Kapsamı

Bu tezde, sonsuz türevlenebilir aktivasyon fonksiyonu için giriş ağırlıkları ve gizli katman eşikleri sabit olduğunda doğrusal bir sistem gibi düşünülen TGKİB ağ için önerilmiş öğrenme algoritması olan AÖM'nin optimizasyonu amaçlanmıştır. AÖM'nin çıkış ağırlıkları seyrek kabul edilerek genel doğrusal bir sistemin çözdürülmesi amaçlanmıştır. Seyrek kabul edilen çıkış ağırlıklarının hesaplatılması işlemi için SGÇ algoritmalarından YFT algoritması tercih edilmiştir. Çalışmada, YFT algoritmalarından ET, DET, YSE, KDET, SÖET, İGT ve DEK yöntemleri incelenmiştir. Bu yöntemlerin herhangi bir seyrek işaret için geri çatma başarımları test edilmiştir. Bahsedilen YFT yöntemleri kullanılarak YFT tabanlı 6 adet AÖM yöntemi önerilmiştir. Önerilen bu algoritmalar, AÖM ve literatürde iyi sonuç verdiği belirlenmiş olan diğer düzenlenmiş AÖM metotları ile karşılaştırılmıştır. Yapılan değerlendirmeler önerilen yaklaşım ile AÖM

gerekenden daha az sayıda nöron ile daha üstün ya da eş değer başarımlar elde edebildiğini göstermiştir.

Bu tezin kapsamı;

- AÖM gizli katman nöron sayısını azaltmak,
- Tekillik probleminden kaçınmak,
- Düşük işlem karmaşıklığına sahip mimari elde etmek,
- Önerilen YFT tabanlı AÖM algoritmalarını temel AÖM ve/veya literatürde iyi başarımlara sahip yöntemler ile karşılaştırmak,

olacaktır.

1.3. Tezin Organizasyonu

Bu doktora tezi sekiz bölüm olarak hazırlanmıştır.

Genel bilgilerin verildiği birinci bölümü izleyen ikinci bölümde kavramların daha iyi anlaşılması için bazı durumlarda doğrusal bir sistemin çözümünü kolaylaştıran Moore-Penrose Genelleştirilmiş Tersli işlemi ve Öklid uzayında $Ax = y, A \in \mathbb{R}^{m \times n}, y \in \mathbb{R}^m$ gibi genel bir doğrusal sistemin en küçük norm en küçük kareler çözümü konuları anlatılacaktır. Ardından TKGİB ağların matematiksel modeli detaylandırılarak yaygın olarak kullanılan Gradyent temelli öğrenme algoritması anlatılacaktır. Daha sonra optimizasyon kavramına değinilecektir. AÖM optimizasyonu için önerilen seyreklik yaklaşımı ve SA hakkında bilgi verilerek, seçilmiş ve uygulamalarda kullanılacak SGÇ algoritmaları anlatılacaktır. Ardından, yapılan çalışmaların değerlendirilmesinde kullanılan başarımlar kriterleri tanıtılacaktır.

Üçüncü bölümde AÖM çıkış ağırlıklarının hesaplanması için YSE algoritması önerilmiştir. YSE tabanlı AÖM yaklaşımı YSE-AÖM olarak isimlendirilmiştir. YSE-AÖM yönteminin başarımlar değerlendirilmesi için tahmin problemlerinde yaygın olarak kullanılan dört veri seti seçilmiştir. Bu bölümde elde edilen sonuçlar ile hazırlanan aşağıdaki çalışmalar yayınlanmak üzere kabul edilmiştir [43, 44].

- Alcin O. F., Sengur A., Ince M. C., "Iterative Hard Thresholding Based Extreme Learning Machine and Its Application to Predictions in Medical Datasets", Brain and Health Informatics, October 29-31, 2013, Maebashi, Gunma, Japan (Abstract Presentation)

- Alçin, Ö. F., Arı, A., Şengür, A., İnce, M. C., “Iterative Hard Thresholding Based Extreme Learning Machine”, 23st Signal Processing and Communications Applications Conference (SIU’15), 16-19 May 2015, (pp. 271-274).

Dördüncü bölümde önerilen Dikgen Eşleme Takip (DET) algoritması tabanlı AÖM açıklanacaktır. Bu yeni yaklaşım DET-AÖM olarak isimlendirilmiştir. DET-AÖM yöntemi ile en uygun gizli katman nöron sayısına, düşük işlem karmaşıklığına sahip ve tekillik problemine bağışık mimari elde edilmiştir. Deneysel çalışmalar yaygın olarak kullanılan dokuz veri seti kullanılarak gerçekleştirilmiştir. DET-AÖM yönteminin başarımı YSA, AÖM, EN, EMDSO yöntemleri ile karşılaştırılmıştır. Elde edilen sonuçlardan hazırlanan aşağıda belirtilen çalışma yayınlanmak üzere kabul edilmiştir [23].

- Alcin, O. F., Sengur, A., Qian, J., & Ince, M. C. OMP-ELM: Orthogonal Matching Pursuit-Based Extreme Learning Machine for Regression. Journal of Intelligent Systems. ISSN (Online) 2191-026X, ISSN (Print) 0334-1860, DOI: 10.1515/jisys-2014-0095, August 2014.

Beşinci bölümde, Sıkıştırılmış Örneklemeli Eşleme Takip (SÖET), Yinelemeli Sert Eşikleme (YSE), DET ve Kademeli Dikgen Eşleme Takip (KDET) algoritmaları tabanlı AÖM yöntemleri detaylandırılmıştır. Bu yöntemler SÖET-AÖM, YSE-AÖM, DET-AÖM ve KDET-AÖM olarak isimlendirilmiştir. Bu önerilen yöntemlerin başarımı yaygın olarak kullanılan 9 veri seti kullanılarak test edilmiştir. SÖET-AÖM, YSE-AÖM, DET-AÖM ve KDET-AÖM yöntemleri önce kendi aralarında karşılaştırılmıştır. Ardından bu dört yöntemden en iyi başarıma sahip DET-AÖM yöntemi ile AÖM yöntemi karşılaştırılıp her iki yöntem için gizli katman sayısına karşı OKHK değişimi grafikler yardımıyla gösterilmiştir. Daha sonra DET-AÖM yöntemi literatürde EMDSO ve EN yöntemleri ile karşılaştırılmıştır. Bu bölümde çalışmalardan elde edilen sonuçlar derlenerek aşağıdaki makale yayınlanmıştır [45].

- Omer F. Alcin, Abdulkadir Sengur, Sedigheh Ghofrani, Melih C. Ince, GA-SELM: Greedy algorithms for sparse extreme learning machine, Measurement, Volume 55, September 2014, Pages 126-132, ISSN 0263-2241, <http://dx.doi.org/10.1016/j.measurement.2014.04.012>.

Altıncı Bölümde seyrek tabanlı AÖM modelinde, çıkış ağırlıklarının seyrek temsilini hesaplamak için YFT algoritması olan İGT tabanlı AÖM yöntemi anlatılacaktır. İGT-AÖM olarak adlandırılan yöntemin başarımı on adet sınıflandırma problemi

kullanılarak test edilmiştir. Ayrıca başarımın etkinliğini vurgulamak için çalışmanın devamında AÖM, EAR, EMDSO ve EN gibi düzenlenmiş AÖM yöntemleri ile önerilen metodun karşılaştırılması yapılmıştır. Bu bölümdeki çalışmalardan elde edilen sonuçlar derlenerek aşağıda belirtilen makale yayınlanmak üzere kabul edilmiştir [46].

- Alçın, Ö. F., Şengür, A. ve İnce, M. C. (2015). “İleri-Geri Takip Algoritması Tabanlı Seyrek Aşırı Öğrenme Makinesi”, *Journal of the Faculty of Engineering & Architecture of Gazi University*, 30(1) 137-143.

Yedinci bölümde önerilen DEK yöntemi tabanlı AÖM yöntemi tanıtılmıştır. Bu yaklaşım DEK-AÖM olarak isimlendirilmiştir. Önerilen DEK-AÖM yöntemin başarım değerlendirmesi üç farklı tip beş zaman serisi tahmin problemi kullanılarak gerçekleştirilmiştir. Her bir zaman serisi için test tahmin eğrisi ve tahmin hatası eğrisi çizdirilmiştir. Ayrıca DEK-AÖM ile AÖM yönteminin karşılaştırılması yapılmıştır. Bu çalışmalardan elde edilen sonuçlar derlenerek makale hazırlanmış olup şu an değerlendirme aşamasındadır.

Sekizinci ve son bölümde, önerilen yöntemlerle yapılan çalışmaların sonuçları tartışılmıştır ve özgün katkıları vurgulanmıştır. Ayrıca ileriye yönelik çalışmalar değerlendirilmiştir.

2. TEORİK ALT YAPI

Giriş ağırlıkları ve gizli katman eşikleri rastgele başlatılmış ise TGKİB sinir ağırları aslında doğrusal bir sistem olarak düşünülebilir. Bundan dolayı kavramların daha iyi anlaşılması için bazı durumlarda doğrusal bir sistemin çözümünü kolaylaştıran Moore-Penrose Genelleştirilmiş Ters ve Öklid uzayında $Ax = y, A \in \mathbb{R}^{m \times n}, y \in \mathbb{R}^m$ gibi genel bir doğrusal sistemin en küçük norm en küçük kareler çözümü konuları bu bölümde anlatılacaktır. Ardından TGKİB ağların matematiksel modeli detaylandırılarak yaygın olarak kullanılan Eğimli Düşüş öğrenme algoritması anlatılacaktır. AÖM optimizasyonu için önerilen Seyreklik yaklaşımı ve sıkıştırılmalı algılama hakkında bilgi verilerek SGÇ algoritmaları özetlenecektir. Daha sonra tez çalışmasında kullanılan başarımlar kriterleri ve normalizasyon işlemi tanıtılacaktır.

2.1. Moore-Penrose Genelleştirilmiş Ters

$Ax = y$ şeklindeki bir doğrusal sistemde A tekil ve hatta kare olmayan bir matris olabilir. Moore-Penrose tersini kullanılarak $Ax = y$ gibi genel bir doğrusal sistemin çözümü çok basit bir hale getirilebilir [1, 2]. $n \times m$ boyutlu bir G matrisi, $m \times n$ boyutlu bir A matrisinin Moore-Penrose Genelleştirilmiş Ters olsun. Böylece (2.1) ifadesi yazılabilir [1, 47].

$$AGA = A, GAG = G, (AG)^T = AG, (GA)^T = GA \quad (2.1)$$

Burada $(.)^T$ operatörü matris transpoz işlemini göstermektedir. Bundan sonra ki ifadelerde sadelik açısından A matrisinin Moore-Penrose Genelleştirilmiş Ters $A^\dagger$ ile gösterilecektir.

2.2. Genel Doğrusal Sistemin En Küçük Norm En Küçük Kareler Çözümü

Genel bir doğrusal sistem olan $Ax = y$ için $\hat{x}$ en küçük kareler çözümü;

$$\|A\hat{x} - y\| = \min_x \|Ax - y\| \quad (2.2)$$

şeklinde ifade edilebilir [1, 2, 47]. Burada $\|\cdot\|$ Öklid uzayında normu göstermektedir. Herhangi bir $y \in \mathbb{R}^m$ için $Ax = y$ şeklindeki doğrusal bir sistemin en küçük norm en küçük kareler çözümü x_0 olduğu söylenebilir [47]. Bu tanım (2.3)'de ifade edilmiştir.

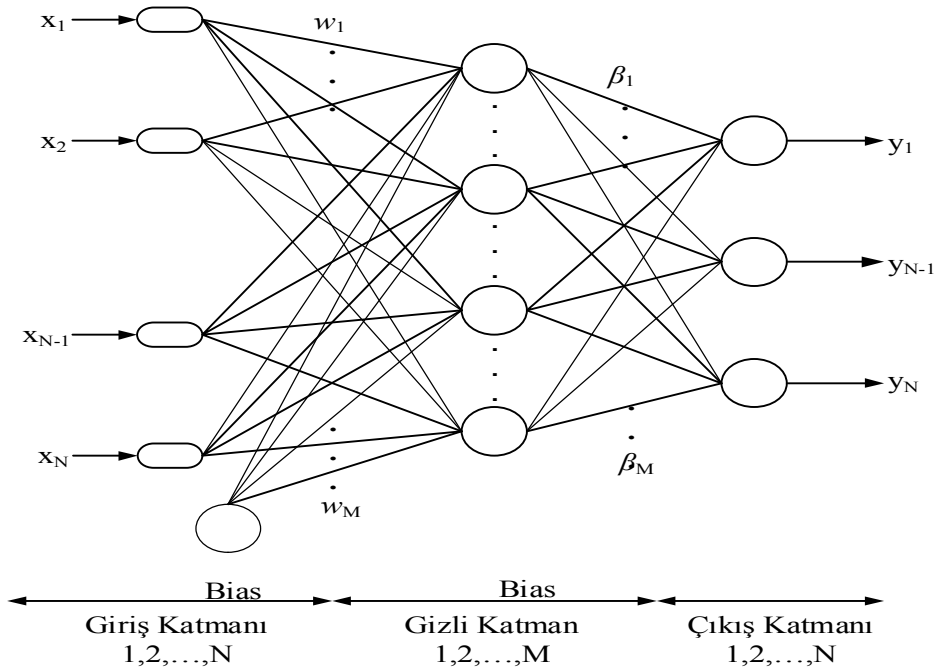
$$\|x_0\| \leq \|x\|, \forall x \in \{x : \|Ax - y\| \leq \|Az - y\|, \forall z \in \mathbb{R}^n\} \quad (2.3)$$

Burada $x_0 \in \mathbb{R}^n$ 'dir.

$Ax = y$ genel doğrusal sistemin minimum normlu en küçük kareler çözümü bir G matrisi (öyle ki Gy) olsun. Burada gerekli ve yeterli şart $G = A^\dagger$ Moore-Penrose tersidir [2, 47].

2.3. Rastgele Tek Gizli Katmanlı İleri Beslemeli Ağlar

TGKİB ağlar farklı sinir ağı mimarileri arasında basit yapıya sahip bir model olmakla birlikte çoğu uygulamada oldukça iyi genelleme performansı gösterebilmektedir [10]. Şekil 2.1'de TGKİB mimarisi gösterilmiştir.



Şekil 2.1. TGKİB ağ mimarisi.

N adet rastgele ve farklı $\{(x_i, y_i) \mid x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n, y_i = [y_{i1}, y_{i2}, \dots, y_{in}]^T \in R^n\}$ veri seti için standart M gizli nöronlu ve $f(x)$ aktivasyon fonksiyonuna sahip TGKİB ağıın matematiksel modeli;

$$\sum_{i=1}^M \beta_i g(w_i \cdot x_j + b_i) = o_j, j = 1, \dots, N \quad (2.4)$$

olarak ifade edilebilir [1, 2, 4, 5, 6]. Eşitlik (2.4)'de, $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ giriş sinir hücrelerini i 'ninci gizli sinir hücrelerine bağlayan ağırlık vektörüdür, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ ise i 'ninci gizli sinir hücrelerine ve çıkış sinir hücrelerine bağlı olan ağırlık vektörüdür. b_i ise i 'ninci gizli sinir hücrelerinin eşik değeridir. $o_j = [o_1, o_2, \dots, o_j]^T$ ise TGKİB ağının çıkışıdır. $w_i \cdot x_j$ ise w_i ve x_j 'nin içsel çarpımını ifade eder. Bu bölümde çıkış nöronları doğrusal seçilmiştir [1, 2]. Standart TGKİB ağ çıkışı bu N tane örnekle çıkışı sıfır ortalama yani $\sum_{j=1}^N o_j - y_j = 0$, ile yakınsayabilir [1, 2]. Mevcut β_i, w_i ve b_i için (2.4) eşitliği;

$$\sum_{i=1}^M \beta_i g(w_i \cdot x_j + b_i) = y_j, j = 1, \dots, N \quad (2.5)$$

şeklinde yazılabilir [1, 2, 10]. Eşitlik (2.5)'te ki N tane eşitlik sade bir şekilde (2.6)'da gösterildiği gibi tanımlanabilir [2, 10].

$$Y = H\beta \quad (2.6)$$

Burada

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_N \cdot x_1 + b_N) \\ \vdots & \dots & \vdots \\ g(w_1 \cdot x_M + b_1) & \dots & g(w_N \cdot x_M + b_N) \end{bmatrix}_{N \times M} \quad (2.7)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_M^T \end{bmatrix}_{M \times 1} \text{ ve } Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times 1} \quad (2.8)$$

olarak ifade edilir. H gizli katman çıkış matrisi olarak isimlendirilir ve H 'ın i . kolonu $x_1, x_2, \dots, x_N$ girişleri ile ilişkili i . nöronun çıkış vektörüdür [1, 10, 23, 45, 46].

2.4. Eğitim-Temelli Öğrenme Algoritması

Daha önce yapılan çalışmalarda [6-8] analiz edildiği gibi eğer gizli katman nöron sayısı eğitim örneklerine eşit ise ($M = N$) H matrisi kare ve tersi alınabilir ve TGKİB ağırlar bu eğitim örneklerini sıfır hata ile yakınsayabilir [1, 2]. Çoğu durumda gizli nöron düğüm sayısı farklı eğitim örnek sayısından çok küçük ($M \ll N$) olur ve H matrisi kare bir matris değildir ve $Y = H\beta$ ifadesini sağlayan w_i, b_i ve β_i ($i = 1, \dots, M$) olmayabilir. Bu durumda (2.9) eşitliğini sağlayan $\hat{w}_i, \hat{b}_i, \hat{\beta}$ bulunmalıdır [1, 2].

$$\|H(\hat{w}_1, \dots, \hat{w}_M, \hat{b}_1, \dots, \hat{b}_M)\hat{\beta} - Y\| = \min_{w_i, b_i, \beta_i} \|H(w_1, \dots, w_M, b_1, \dots, b_M)\beta - Y\| \quad (2.9)$$

Burada maliyet fonksiyonu E , minimize edilmelidir [1].

$$E = \sum_{j=1}^N \left(\sum_{i=1}^M \beta_i g(w_i x_j + b_i) - y_j \right)^2 \quad (2.10)$$

H bilinmediğinde, eğitim-temelli öğrenme algoritmaları genellikle $\|H\beta - Y\|$ ifadesinin minimumunu bulduran arama algoritmaları kullanır. Eğitim-temelli algoritmaları kullanılan minimizasyon işleminde (w_i, β_i) ve b_i 'den oluşan W vektörü yinelemeli olarak (2.11)'de belirtildiği gibi ayarlanır [1, 2].

$$W_k = W_{k-1} - \eta \frac{\partial E(W)}{\partial W} \quad (2.11)$$

Burada η öğrenme oranını göstermektedir.

İleri beslemeli sinir ağlarında kullanılan popüler öğrenme algoritmalarından biri de geriye yayılım algoritmasıdır [2]. Geriye-yayılım algoritmasında eğimler yayılım işlemi ile çıkıştan girişe doğru verimli bir şekilde hesaplanabilir [1]. Geriye-yayılım öğrenme algoritmasında bazı olumsuzlukları aşağıda belirtildiği gibi sıralanabilir [1-2].

- Öğrenme oranı η çok küçük olduğunda öğrenme algoritması çok yavaş yakınsar. Öğrenme oranı η çok büyük olduğunda ise algoritma kararsız olmaya başlar.
- Geriye-yayılım algoritmasında yerel minimum noktalarının var olması öğrenme sürecinde algoritmanın bu noktalara takılma riskine neden olacaktır [48].
- Sinir ağlarında ezberleme durumu söz konusu olabilir ve kötü bir genelleme performansı elde edilir. Ayrıca maliyet fonksiyonunu minimizasyon işlemlerinde geçerleme ve uygun durdurma yöntemleri gerekli olmaktadır.

- Eğitim-tabanlı öğrenme pek çok uygulamada aşırı zaman almaktadır.

AÖM bu zorlukları çözmek için önerilmiş bir öğrenme algoritmasıdır [1, 2, 10, 45,46].

2.5. Tek Gizli Katmanlı İleri Beslemeli Ağlar En küçük kareler çözümü

TGKİB ağları yaygın anlayışın dışında bir yapıya sahiptir [1]. TGKİB ağlarda giriş ağırlıklarını, gizli katman eşiklerini ve gizli katman çıkış matrisini güncellemeye gerek yoktur ve öğrenme sürecinde bir kere atandıktan sonra bu parametreler sabit kalabilir [1, 2, 10, 45,46].

Bu konu ile ilgili literatür çalışmalarında sonsuz türevlenebilen aktivasyon fonksiyonlu giriş ağırlıkları rastgele atanmış TGKİB ağın farklı örneklerden çok küçük hata değeri ile öğrenebileceği gösterilmiştir [8]. Benzetim çalışmaları ve pratik uygulamalarda iyi sonuçlar üretilirken yapılan çalışmalarda giriş ağırlıklarına ek olarak gizli katman eşiklerinin rastgele seçilebileceği ispatlanmıştır [3, 9, 49].

Benzetim ve teorik çalışma sonuçları giriş ağırlıkları ve gizli katman eşik değerlerinin rastgele başlatıldıktan sonra ayarlanmaya gerek duyulmadığını göstermektedir [1, 2]. Giriş ağırlığı w_i ve gizli katman eşiği b_i sabit ise eşitlik (2.9)'dan görüleceği gibi TGKİB ağın eğitilmesi çıkış ağırlıkları $\hat{\beta}$ 'nın doğrusal sistem olan $Y = H\beta$ ifadesinden en küçük kareler çözümü ile bulunmasıdır [1, 2].

$$\|H(\hat{w}_i, \dots, \hat{w}_M, \hat{b}_i, \dots, \hat{b}_M)\hat{\beta} - Y\| = \min_{\beta_i} \|H(w_i, \dots, w_M, b_i, \dots, b_M)\beta - Y\| \quad (2.12)$$

Teorem 2.1'e göre yukarıda verilen doğrusal sistemin en küçük kareler çözümü şöyle ifade edilebilir;

$$\hat{\beta} = H^\dagger Y \quad (2.13)$$

Daha önceden de vurgulandığı gibi aşağıda belirtilen önemli konular söz konusudur. Bu konulardan ilki olarak minimum eğitim hatasına ulaşılacak istenmesi söylenebilir. Eşitlik (2.13)'ün özel çözümü doğrusal bir sistem olan $Y = H\beta$ eşitliğinin en küçük kareler çözümü bir tanedir. Yani en küçük eğitim hatasına (2.14) eşitliğinin özel çözümü ile ulaşılabilir.

$$\|H\hat{\beta} - Y\| = \|HH^\dagger Y - Y\| = \min_{\beta} \|H\beta - Y\| \quad (2.14)$$

Bütün öğrenme algoritmaları minimum öğrenme hatasına ulaşmak isterken pek çoğu yerel minimum veya sınırlı eğitim iterasyonu yüzünden bunu gerçekleştiremeyebilir [2, 10]. Söz konusu ikinci önemli durum, en küçük ağırlıklar normu ve genelleme performansdır. Ayrıca $\hat{\beta} = H^\dagger Y$ ifadesinin (2.15)'te ifade edilen özel çözümü $Y = H\beta$ doğrusal sisteminin en küçük kareler çözümleri arasında en küçük norma sahiptir [2].

$$\|\hat{\beta}\| = \|H^\dagger Y\| = \|\beta\|, \forall \beta \in \{\beta : \|H\beta - Y\| \leq \|Hz - Y\|, \forall z \in \mathbb{R}^{M \times N}\} \quad (2.15)$$

[11, 50] çalışmalarında Bartlett'in belirttiği gibi; eğitim örneklerinde küçük karesel hata ve çok küçük ağırlık genliğine sahip ileri beslemeli ağlar için Vapnik-Chervonenkis boyutu genelleme performansında etkilidir. İleri beslemeli ağlarda ağırlıkların genliği çok önemli olmaktadır [1]. Küçük ağırlıklı ağlar daha iyi genelleme performansı gösterebilme eğilimine sahiptir. Bu durum Eğitim-temelli öğrenme algoritmaları için zorlayıcı bir unsur olabilir [1, 2].

Son olarak $Y = H\beta$ doğrusal sisteminin en küçük normlu en küçük kareler çözümünün ($\hat{\beta} = H^\dagger Y$) tek olduğu söylenebilir [1, 2].

2.6. Tek Gizli Katmanlı İleri Beslemeli Ağlar için Öğrenme Algoritması

TGKİB ağlar için basit bir öğrenme algoritması önerilen yöntem Aşırı Öğrenme Makinası (AÖM) olarak isimlendirilmiştir [1, 2, 10, 23].

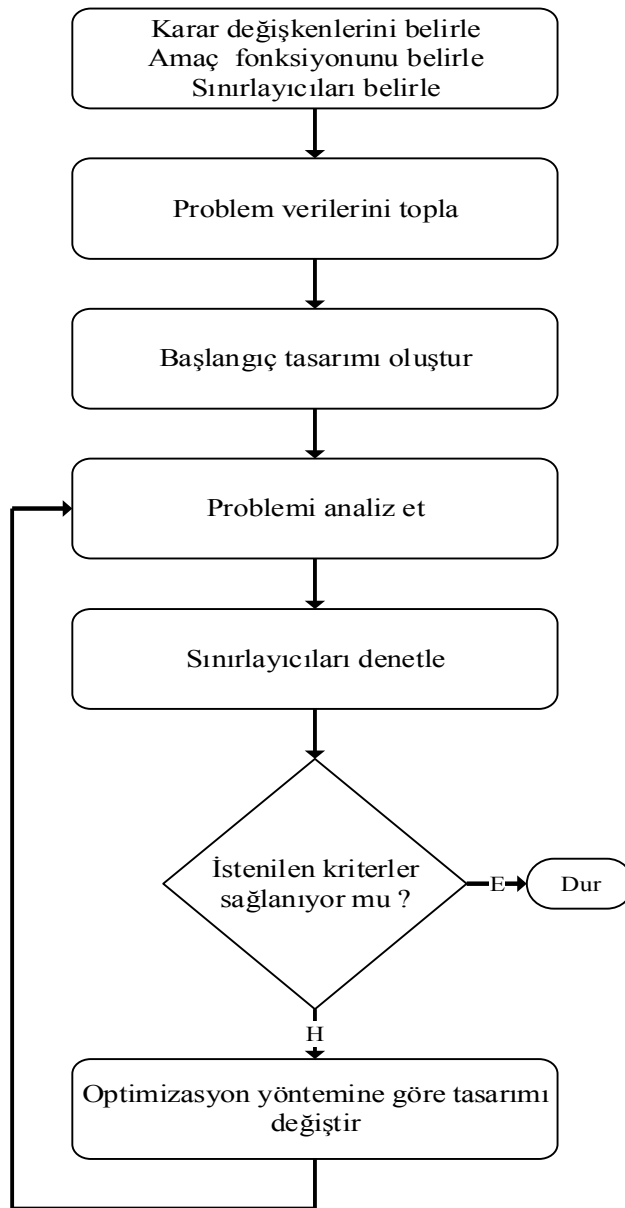
Herhangi bir $\{x_i, y_i\}$ veri çifti, M gizli katman nöron sayısı ve $g(x)$ aktivasyon fonksiyonu için AÖM algoritması işlem akışı şu şekilde özetlenebilir;

- Giriş ağırlıkları w_i ve gizli katman eşiklerini b_i ($i=1, \dots, M$) değerlerini rastgele ata;
 - Gizli katman çıkış matrisini hesapla.
 - Çıkış ağırlıkları β hesapla
- şeklinde özetlenebilir [1, 2, 45, 46].

2.7. Optimizasyon

Optimizasyon bir problemin belirli şartlar altında muhtemel tüm çözümleri arasından en uygun olanın bulunması demektir [51]. Optimizasyon işleminde en uygun

değeri bulmak için denetimimiz altındaki değişkenler *karar değişkenleri* olarak adlandırılır. Karar değişkenlerinin istenen amaç üzerine etkilerinin analitik olarak ifade edilmesiyle *amaç fonksiyonu* elde edilir. Karar değişkenlerinin bazı durumlarda belli aralıklarda değerler alması istenir, karar değişkeni üzerindeki bu kısıtlamalara *sınırlayıcılar* denir. Bu tanımlamalar ile optimizasyon, karar değişkenlerinin olası bütün varyasyonları içinden tüm sınırlamaları sağlayan minimizasyon ya da maksimizasyon işlemi ile en uygun amaç fonksiyonunun bulunması işlemidir [51-55]. Şekil 2.2’de bir problem için herhangi bir optimizasyon yöntemine ait akış şeması verilmiştir [53].



Şekil 2.2. Optimizasyonu akış şeması [53].

Bir tasarım aşamasında iki temel adım, probleme ait modelin oluşturulması ve çözüm için bu modelin kullanılmasıdır. Bazı problemler için sistem modellerine gerek duyulmadan da çözümlere ulaşmak mümkündür. Problemin çözümü sistemden elde edilen geri beslemelerden faydalanılarak farklı çözümlerin doğrudan fiziksel sistem üzerinde denenmesi ile elde edilebilir. Bu durumda problemin çözümü için model oluşturulmasına gerek duyulmaz. Fakat bu çözüm türü çoğu kez uygulanamamaktadır [54]. Problemi modellemenin avantajları arasında bilgisayar benzetimlerinin pratik uygulamalara göre çok daha hızlı gerçekleştirilebilmesi yer almaktadır. Böylelikle ilk olarak problemin modeli oluşturulur ardından bu modele göre çözümler üretilir. Eğer model hatalı oluşturulmuş ise gerçek problem için çözüm üretilemez ya da model için iyi bir çözüm üretilmiyorsa model doğru olsa bile optimizasyon yapılamayabilir. Bir problemin çözümü ya da bir tasarımın oluşturulması aşamalarında harcanan iş gücünün minimize edilmesi veya arzu edilen kazancın maksimize edilmesi için bir takım kararlar alınması gerekmektedir. Bu süreçte optimizasyon emek ve zaman konusunda tasarruf sağlamaktadır [51-55].

Matematiksel olarak optimizasyon; problem sınırlayıcılarını sağlayan çözümlerin oluşturduğu bir τ kümesi ile bir $\mathcal{X}$ amaç fonksiyonu kümesinin oluşturduğu ($\mathcal{X} : \tau \rightarrow \mathbb{R}$ şartını sağlayan) $(\tau, \mathcal{X})$ ikilisi olarak tanımlanabilir. Optimizasyon probleminin temelinde çözüm kümesi içinde $\forall S \in \mathcal{X} \tau(S^*) \leq \tau(S)$ (veya $\forall S \in \mathcal{X} \tau(S^*) \geq \tau(S)$) şartını sağlayan S^* çözümünü bulmaktır. Burada S evrensel çözüm ve S^* ise evrensel optimal çözüm veya yalnızca optimal çözüm olarak isimlendirilir [54].

Bir optimizasyon problemi daha önceden de belirtildiği gibi minimizasyon veya maksimizasyon problemi şeklinde tanımlanabilir. Optimizasyon problemi, optimal çözümün görüntüsünün amaç fonksiyonu üzerine yansması diğer olası çözümlerden küçük ya da eşit ise minimizasyon; büyük ya da eşit ise maksimizasyon problemidir [51]. Bu tanımlamadan hareketle optimizasyon belirli bir çözüm uzayında amaç fonksiyonu değerini minimum ya da maksimum yapan eleman veya elemanların belirlenmesi işlemidir [51-55].

2.7.1. Optimizasyon Yöntemlerinin Sınıflandırılması

Optimizasyon yöntemleri klasik yöntemler ve modern sezgisel yöntemler olmak üzere iki ana gruba ayrılabilir [54]. Bu sınıflama Şekil 2.3'te gösterilmiştir. Klasik

yöntemlere kesin çözüm veren yöntemler ve modern sezgisel yöntemlere ise yaklaşık çözüm üreten yöntemler denebilir [54].



Şekil 2.3. Optimizasyon yöntemlerinin sınıflandırılması [54].

Yaklaşık çözüm üreten yöntemler değişkenlerin seçim aşamasında her bir değişken için hesaplanan puanı dikkate alır. Yani iki elemandan birinin tercih edilmesi gereken durumda puanı daha iyi olan tercih edilir. Yinelemeli fırsatçı algoritmalar yaklaşık çözüm üreten yöntemler arasına girer. Yinelemeli fırsatçı algoritmalarda, bir yineleme adımı

genel duruma dikkat edilmeksizin o adım için optimum seçim yapılır. Her bir yineleme adımımda elde edilen yerel optimumun, genel optimum noktasına ulaştırması beklenir [56].

2.8. Seyrek Geri Çatma ve Sıkıştırılmalı Algılama

SGÇ yöntemi pek çok işaret işleme uygulamasında kullanılmaktadır. İletilmesi ve depolanması zor bir işaretin modeli pek çok pratik uygulamalarda kullanılmaktadır. SGÇ ile gerek duyulan depolama alanı çok aza indirgenir. Gerçek değerli (R^N), $N \times 1$ boyutlu, ayrık-zamanlı bir x hedef işareti olsun. Bu x işareti herhangi bir dik dönüşümle seyrek veya bir dönüşüm tabanında seyrek olarak ifade edilebiliyorsa bu işareti Nyquist oranının belirttiğinden çok daha az sayıda örnek ile örneklenebilir [27]. Seyrek ifadenin elde edilebilmesi için dalgacık, Fourier veya kosinüs gibi bazı dönüşüm tabanları kullanılabilir. Ψ dönüşüm tabanını temsil etmek üzere dönüşüm işlemi (2.16)'daki gibi ifade edilebilir [27-28].

$$x = \sum_{i=1}^N s_i \Psi_i \text{ veya } x = \Psi s \quad (2.16)$$

Burada s , $N \times 1$ boyutunda katsayı vektörüdür. Açıkça görüleceği üzere x işareti zaman, s ise Ψ tabanında temsil edilen eşdeğer gösterimleridir. Seyreklik, bilinmeyen N boyutlu bir x işaretini oluşturan değerlerin K tanesinin sıfırdan farklı diğer $N-K$ tane katsayının sıfır olması durumu olarak ifade edilebilir. Bu durum matematiksel olarak (2.17)'de ki gibi ifade edilebilir.

$$\|x\|_0 = K \quad (2.17)$$

(2.17)'den görüleceği gibi seyrek bir x işaretinin l_0 normu bir işaretin sıfırdan farklı değerlerinin sayısını tanımlar [27]. Bu durumda x işaretine K -seyrek denir [27-33]. Eğer $K \ll N$ ise x sıkıştırılabilir bir işarettir. Şekil 2.4'te 1024×1 boyutlu $K=50$ olan seyrek bir x işareti gösterilmiştir. SA teorisi seyrek olarak ifade edilebilen işaretin anlamlı bilgi içeren kısmının ölçülebilmesini konu alır. Bu durum (2.18)'de ifade edilmiştir [27-34].

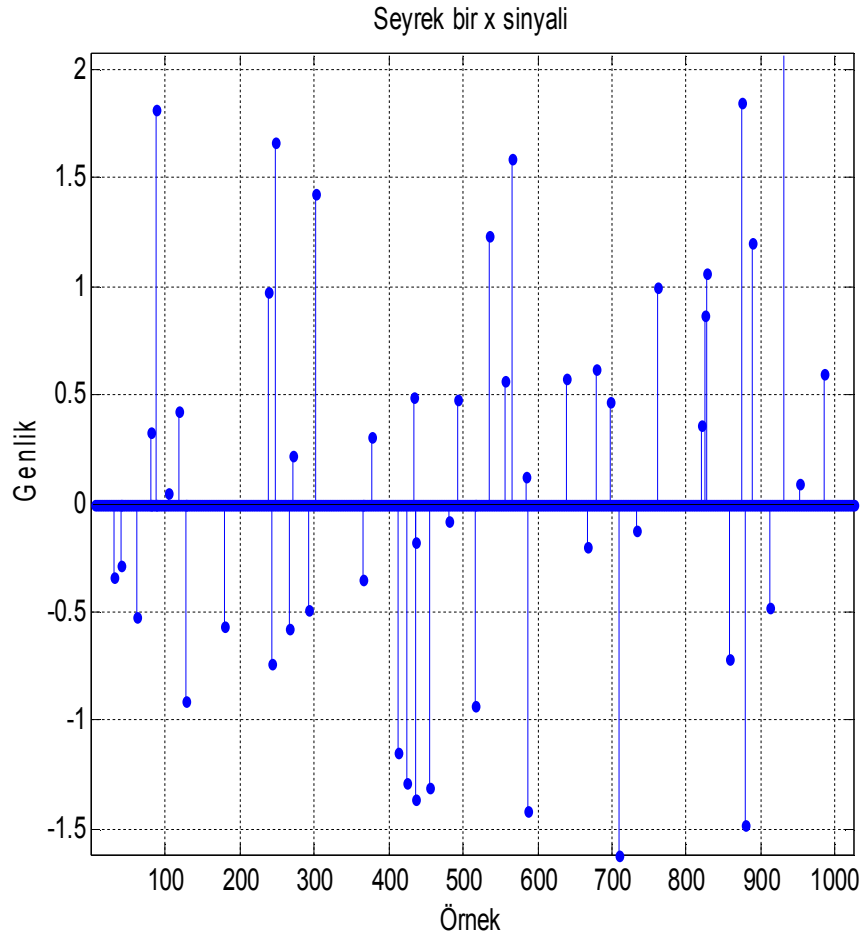
$$y = Ax \quad (2.18)$$

Burada x seyrek kabul edilen bilinmeyen işaret, y bilinen $M \times 1$ boyutlu ölçüm vektörü, A ise $M \times N$ boyutlu ($M \ll N$) ölçüm matrisidir. A matrisinin kolonları *atom* olarak

isimlendirilir. Ayrıca ölçümler belirli bir seviye gürültü ile bozulmuşsa SA işareti belirli bir hata oranıyla geri çatmaktadır [27-34]. Gürültülü ölçüm modeli (2.19)'da verilmiştir.

$$y = Ax + e \quad (2.19)$$

Denklemden e , $M \times 1$ boyutlu gürültüyü temsil etmektedir.



Şekil 2.4. Seyrek bir x işareti

K -seyrek x işaretini geri çatmak için SA gerekli M (minimum) ölçüm sayısı (2.20)'de belirtilmiştir [31].

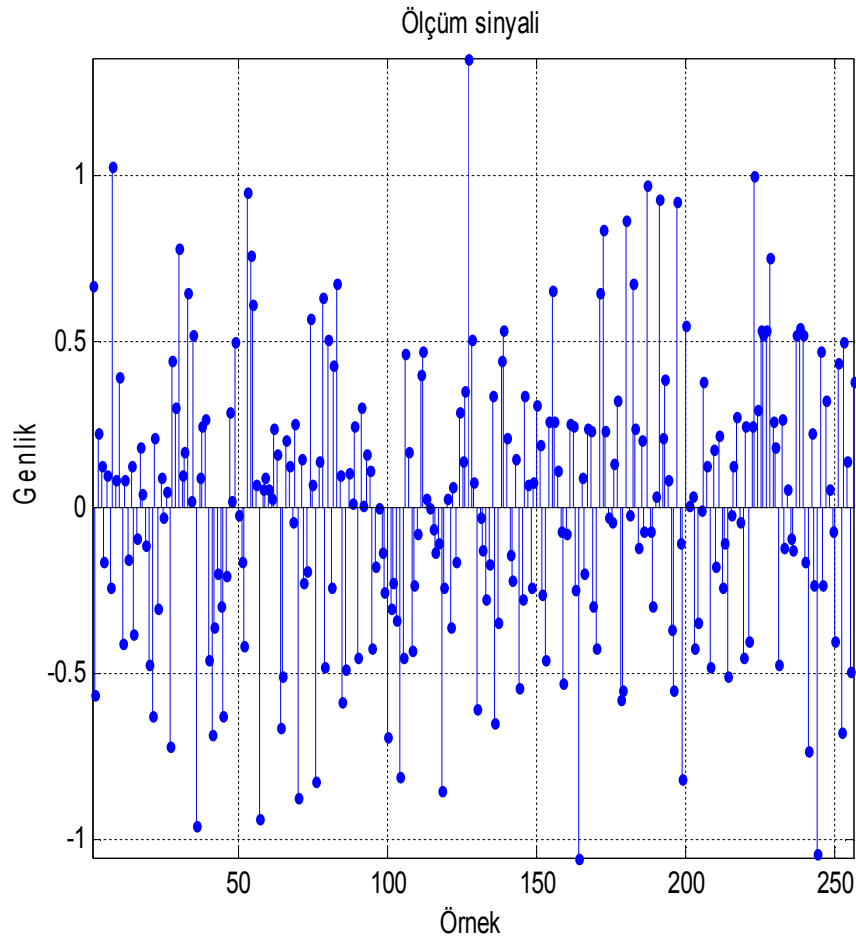
$$M = O \times K \times \log\left(\frac{N}{K}\right) \quad (2.20)$$

Denklemden O , eşitlik için kullanılacak sabittir. Bu bağıntı gerekli ölçüm sayısının M seyreklik K ile doğru orantılı artacağını, işaretin boyutu ile artışın logaritmik olduğunu göstermektedir [31]. Şekil 2.4'deki x işaretini geri çatmak için olan M değeri $O=1.7$ alınarak (2.20)'den yaklaşık 256 bulunur. Rastgele 256×1024 boyutlu ölçüm matrisi

kullanılarak ve varyansı 0.01 olan rastgele gürültü eklenmiş ölçüm işareti Şekil 2.5’de verilmiştir. (2.18)’de verilen problemin çözümü (2.21)’de verilmiştir.

$$\underset{x}{\operatorname{argmin}} \|x\|_0 \text{ öyleki } Ax = y \quad (2.21)$$

(2.18) ve (2.19)’da gösterilen modelde A matrisinin sıfır uzayı bulunduğundan, sistemi sağlayan sonsuz sayıda çözüm bulunmaktadır [27-34].



Şekil 2.5. Ölçüm işareti

Bundan dolayı bu denklem sistemini çözmek için sıfır normu (2.21)’de gösterilen doğrusal programlama ile çözülebilen dış bükey en iyileme (l_1) minizasyonu olarak bilinen bir norm ile değiştirilerek daha kısa sürede kararlı bir çözüm yapılabileceği ispatlanmıştır [28, 32].

$$\underset{x}{\operatorname{argmin}} \|x\|_1 \text{ öyleki } Ax = y \quad (2.22)$$

(2.19)'da verilen, gürültüye maruz kalmış ölçüm modelinin çözümü için veri kısıtı gevşetilerek (2.23)'de verilen çözüm yapılabilir [27-29].

$$\underset{x}{\operatorname{argmin}} \|x\|_1 \text{ öyle ki } \|Ax - y\|_2 \leq \varepsilon \quad (2.23)$$

Burada ε ölçümde ki gürültünün miktarını belirten sınırdır. (2.18) ve (2.19)'da verilen problemlerin çözümünde genellikle dış bükey en iyileme l_1 olarak da bilinen TA ve YFT algoritmaları kullanılır [27, 35].

2.9. Seyrek Geri Çatma Algoritmaları

SGÇ probleminin çözümü için TA algoritması yaygın olarak kullanılan ve üzerinde çalışılmaya devam edilen doğrusal programlama temelli bir yöntemdir. TA algoritması işlem karmaşıklığı fazla ve dolayısıyla yavaş olduğundan bu yönetime alternatif olarak yinelemeli algoritmalar önerilmiştir. Yinelemeli algoritmalar hızlıdır fakat yakınsama performansları TA algoritması kadar iyi değildir [36].

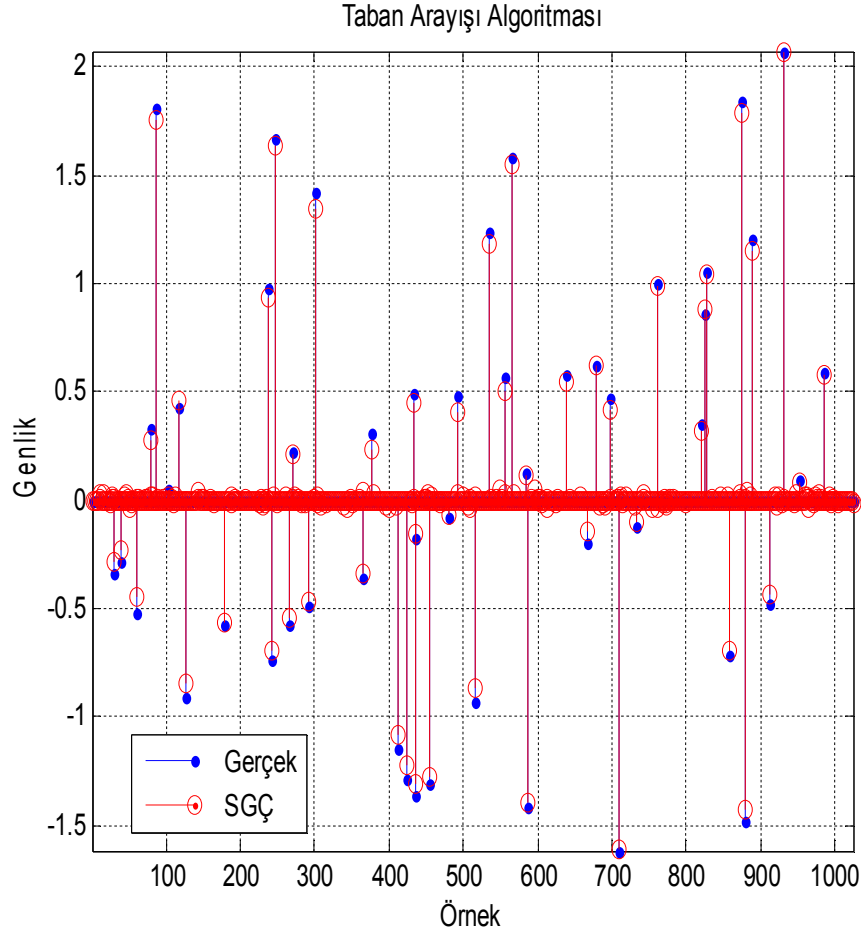
2.9.1. Taban Arayışı Algoritması

Eşitlik (2.21)'de verilen SGÇ probleminin l_0 normu ile çözümü zordur. TA algoritması l_0 normunu l_1 normu ile değiştirerek problemin çözümünü kolaylaştırır. TA doğrusal programlama temelli bir algoritmadır. TA algoritması kullanılarak seyrek işaretin geri elde edilmesi için A matrisinin KİÖ sahip olması gerekmektedir [27-31,33]. Bütün seyrek ifade edilebilen x vektörleri için, $M \times N$ boyutlu A matrisi KİÖ sabitine, ($\delta \in (0,1)$) sahiptir. Denklem sistemini M tane ölçüm kullanarak kesin çözmek için A matrisinin belli özelliklere sahip olması gerekmektedir [27, 33-35]. Bu özellik (2.24)'te verilmiştir.

$$(1 - \delta) \|x\|_2^2 \leq A \|x\|_2^2 \leq (1 + \delta) \|x\|_2^2 \quad (2.24)$$

A matrisi; elemanları bağımsız ve özdeş seçilmiş rastgele Gaussian, Bernoulli matrisleri seçilirse KİÖ kriteri büyük olasılıkla sağlanmaktadır [27-42]. (2.19)'da verilen gürültülü ölçüm işareti için veri kısıtı gevşetilerek (2.23)'de EMDSO olarak da bilinen optimizasyon yöntemi ile çözüm elde edilir. Şekil 2.2'de görülen 1024×1 boyutlu x işaretini tekrar elde edilmek istenen işaret olarak ele alalım. Şekil 2.4'de verilen seyrek bir x işaretinin TA algoritması ile geri çatılması sonucu elde edilen işaret Şekil 2.6'da verilmiştir. Şekil 2.6 incelendiğinde TA'nın işaretin sıfırlarını tam olarak geri çatıldığını ve

sıfırdan farklı işaret değerlerini de belirli bir hata oranında geri çattığı görülmektedir. Geri çatılan işaret ile gerçek işaretin ne kadar örtüştüğü, diğer bir ifade ile geri çatım başarımı için OKHK hesaplanmıştır. Hesaplanan OKHK değeri 0.0121'dir.



Şekil 2.6. Elde edilmiş işaret ile gerçek işaretin karşılaştırılması

2.9.2. Yinelemeli Algoritmalar

Yinelemeli yöntemler dış bükey en iyileme (l_1) yöntemlerinin aksine büyük ve artıklı bir vektör kümesinin içinden tercihen en küçük vektör alt kümesiyle işaret ayrışımı yaparak çözüme adım adım ulaşmayı sağlayan algoritmalarlardır. Bu algoritmaların avantajı hesaplama karmaşıklığının azaltılması ve genellikle çözüme daha hızlı ulaşılabilmesidir. Döngüyü durdurma kriteri olarak, işaretin seyreklik seviyesi K , ölçüm matrisinin satır sayısı veya belirli bir hata seviyesi seçilebilir [29, 31-33, 37-42, 58]. Bu bölümde ET, DET, KDET, YSE, SÖET, İGT ve DEK algoritmaları incelenmiştir.

2.9.3. Eşleme Takip Algoritması

ET algoritması dikgen bir kısıtlama olmaksızın yapılan uyarlanabilir yakınsamalara dayanmaktadır. ET algoritması işaret vektörünü x 'e eşleştirmek amacıyla A matrisinin uygun kolonları arasında ve birleştirme katsayılarını aramasından dolayı eşleştirme arayışı ismini almıştır [39, 42 58, 59]. ET algoritması iki adımdan oluşur; birincisi atom seçim adımı, ikincisi artık vektör güncelleme adımıdır. Atom seçilmesi (2.25)'te belirtildiği gibi güncel artık vektör ile en yüksek ilişkili atomun belirlenmesidir [59].

$$\alpha_j = \arg \max_{a_i} \left| \langle r_{j-1}, a_i \rangle \right| \quad (2.25)$$

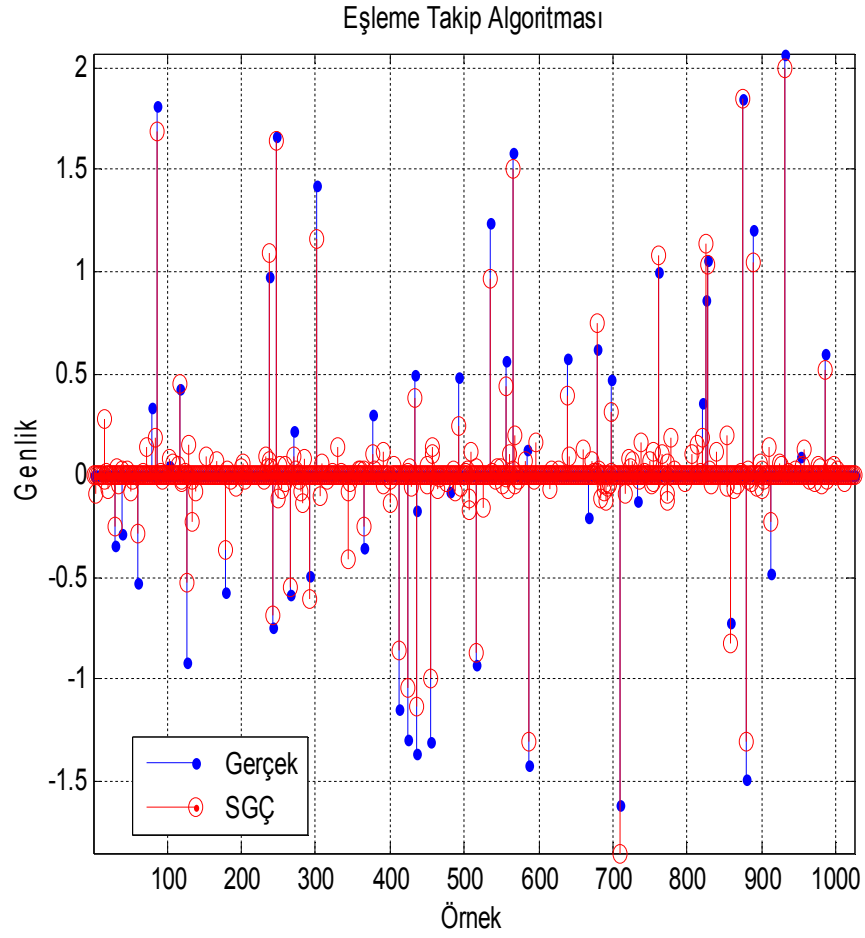
Denklemden a_i A matrisinin i . ($1 \leq i \leq N$) atomu, r_{j-1} bir $j-1$. iterasyondaki artık vektörü göstermektedir. Güncelleme adımı bir önceki artık vektörden ilişkili kısım çıkartılarak gerçekleştirilir [59].

$$r_j = r_{j-1} - c_j \alpha_j \quad (2.26)$$

Yukarıdaki denklemde $c_j = \langle r_{j-1}, a_i \rangle$ 'dir. j . iterasyondaki yaklaşık çözüm (2.27)'de verilmiştir [59].

$$y_j = \sum_{k=1}^j c_k \alpha_k \quad (2.27)$$

ET algoritması, belirtilen hata sınırına ulaşıldığında veya döngü sayısı t 'ye ulaştığında sonlandırılır. Bu algoritma tüm sözlük elemanlarıyla yapılmak yerine tek tek yapılması sebebiyle en iyi sonuca ulaşamamaktadır [58]. Şekil 2.2'de verilen x işaretini ET algoritması kullanılarak geri çatılmış hali Şekil 2.7'de verilmiştir. ET algoritmasının OKHK'sı 0.0525 olarak hesaplanmıştır. ET algoritmasının başarımı TA algoritmasının başarımından daha kötüdür. Bu durum Şekil 2.6 ve Şekil 2.7 karşılaştırıldığında da açıkça görülmektedir.



Şekil 2.7. x işaretinin ET algoritması ile elde edilmesi

2.9.4. Dikgen Eşleme Takip Algoritması

DET algoritması ölçüm işaretini kullanarak her adımda oluşturulan artık vektör ile en fazla ilişkili atomları seçip, her adımda artık vektörü belirlenen sonlandırma kriterine kadar güncelleme işlemine dayanır. DET algoritmasında ET algoritmasından farklı olarak iz düşüm yönleri dikleştirilerek yakınsama performansı artırılmıştır. Bunun sonucu olarak da işlem karmaşıklığı artmaktadır. DET algoritması Tablo 2.1’de özetlenmiştir [30, 36, 37].

Tablo 2.1. DET algoritması

Girdi: A , y ; Çıktı: x

İndeks kümesi: $\Lambda_t \leftarrow \emptyset$, Artık vektör: $r_0 \leftarrow y$, döngü indeksi: $t \leftarrow 1$ başlatılır.

Döngü

$\lambda_t \leftarrow \mathbf{argmax} |\langle r_{t-1}, A_{\omega} \rangle|$ A_{ω} : A matrisinin kolonlarıdır.

$\Lambda_t \leftarrow \Lambda_{t-1} \cup \lambda_t$, İndeks kümesi güncellenir

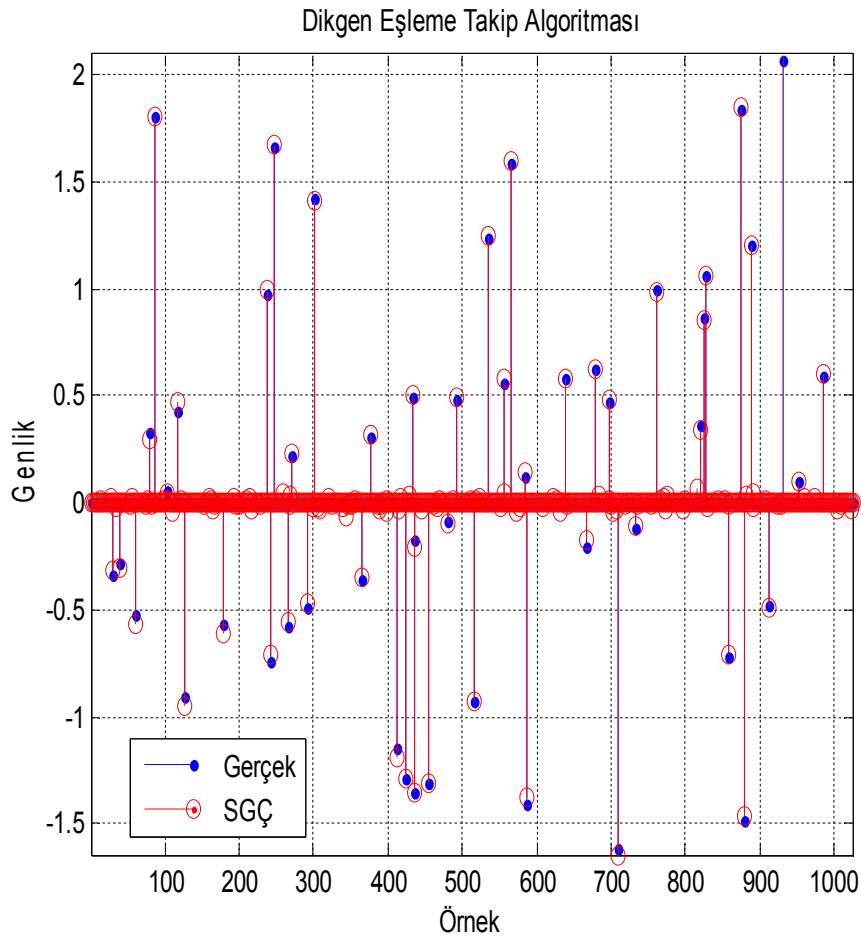
$x(\lambda_t) \leftarrow \min \|y - \sum_{j=1}^t x(\lambda_j) A_{\lambda_j}\|_2$

$r_t \leftarrow y - \sum_{j=1}^t x(\lambda_j) A_{\lambda_j}$, Artık vektör güncellenir.

$t \leftarrow t + 1$, Döngü indeksi artırılır.

Koşul sağlandığında döngüden çık

Şekil 2.8’de seyrek x işaretinin DET algoritması ile geri çatılmış işareti verilmiştir.



Şekil 2.8. x işaretinin DET algoritması ile elde edilmesi

Şekil 2.8 incelendiğinde DET'in işaretin sıfırlarını tam olarak geri çattığını ve sıfırdan farklı işaret değerlerini de belirli bir hata oranında geri çattığı görülmüştür. Ayrıca Şekil 2.6 ile Şekil 2.7 karşılaştırıldığında DET algoritması işareti ET algoritmasından daha iyi yakınsadığı görülmektedir.

2.9.5. Kademeli Dikgen Eşleme Takip Algoritması

KDET algoritması, Donoho [40] tarafından geliştirilmiştir. DET algoritmasından farklı olarak belirli bir eşik seviyesinden büyük (λ_t) indeksler seçilir. Bu algoritma daha az iterasyon sayısı ile daha iyi yakınsama sağlamaktadır. Bununla birlikte uygun bir eşik değerinin ayarlanması çok önemlidir. Çünkü farklı eşik değeri için farklı sonuçlar ortaya çıkacaktır [36, 40]. KDET algoritması Tablo 2.2'de özetlenmiştir.

Tablo 2.2. KDET algoritması

Girdi: A , y ; Çıktı: x

İndeks kümesi: $\Lambda_t \leftarrow \emptyset$, Artık vektör: $r_0 \leftarrow y$, döngü indeksi: $t \leftarrow 1$ başlatılır.
yinele

$\lambda_t \leftarrow \text{argmax} |\langle r_{t-1}, A_\omega \rangle|$, A_ω : A matrisinin kolonlarıdır.

$J = \{j: |\lambda_t| > t_t \sigma_t\}$ Belirtilen eşikten büyük λ_t indeksi seçilir.

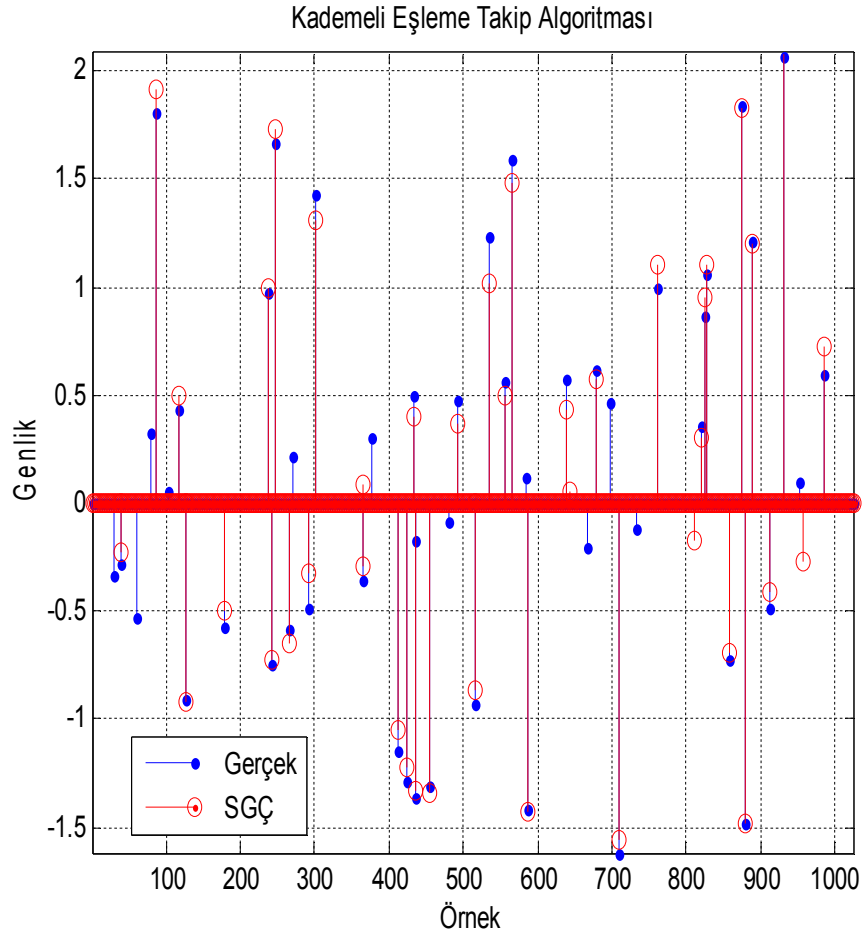
$\Lambda_t \leftarrow \Lambda_{t-1} \cup J$ İndeks kümesi güncellenir.

$x(\lambda_j) \leftarrow \min \left\| y - \sum_{j=1}^t x(\lambda_j) A_{\lambda_j} \right\|_2$

$r_t \leftarrow y - \sum_{j=1}^t x(\lambda_j) A_{\lambda_j}$, Artık vektör güncellenir.

$t \leftarrow t + 1$, Döngü indeksi artırılır.

Koşul sağlandığında döngüden çık



Şekil 2.9. x işaretinin KDET algoritması ile elde edilmesi

Şekil 2.9’da seyrek x işaretinin KDET algoritması ile geri çatılmış işaret verilmiştir. KDET algoritması için OKHK 0.0364 hesaplanmıştır.

2.9.6. Yinelemeli Sert Eşikleme

YSE algoritması ölçüm işareti ile ölçüm matrisinin yansımasına katı eşikleme uygulamaktadır. Bu katı eşikleme operatörü T , (2.28)’de tanımlanmıştır [29, 60]. YSE algoritması Tablo 2.3’te özetlenmiştir. Şekil 2.10’da seyrek x işaretinin YSE algoritması ile geri çatılmış işaret gösterilmiştir.

$$T(x) = \begin{cases} x, & |x| \geq \frac{\theta}{2} \\ 0, & x < \frac{\theta}{2} \end{cases} \quad (2.28)$$

Tablo 2.3. YSE algoritması

Girdi: A , y ; Çıktı: x

Katsayı vektörü: $\mathbf{c}_0 \leftarrow \mathbf{0}$, Yakınsama vektörü: $\mathbf{a}_0 \leftarrow A\mathbf{c}_0 \leftarrow \mathbf{0}$, Artık vektör: $\mathbf{r}_0 \leftarrow \mathbf{y}$,
döngü indeksi: $t \leftarrow 1$

Döngü

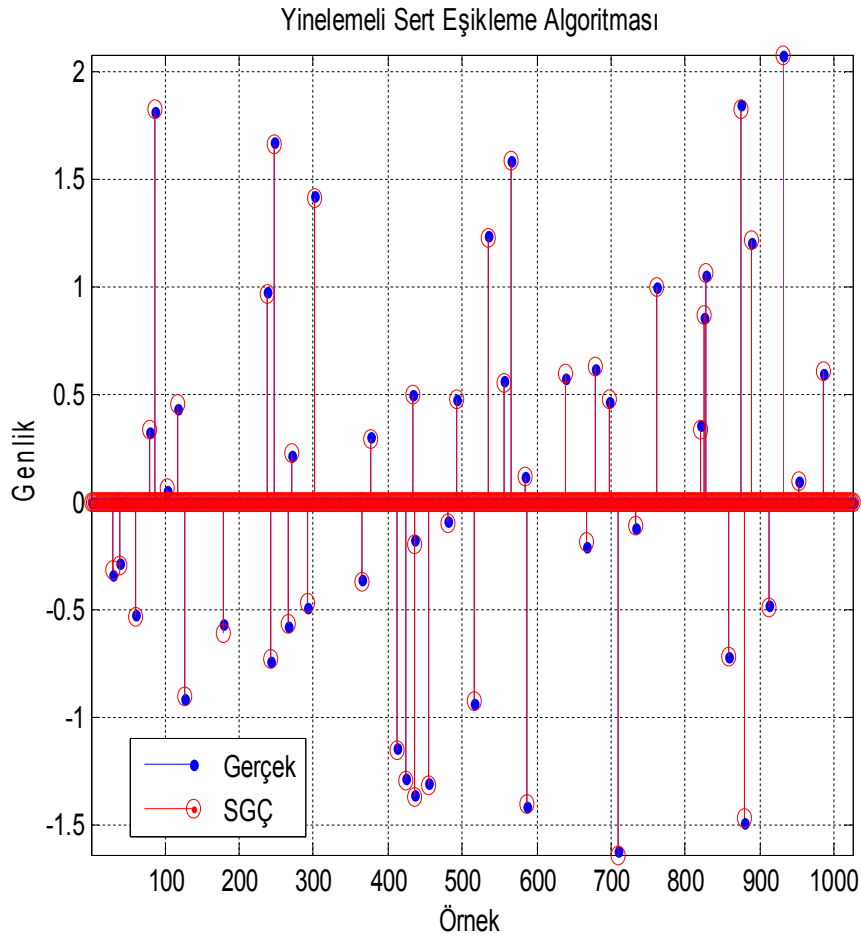
$\mathbf{c}_t \leftarrow T_\theta(\mathbf{c}_{t-1} + A^T(\mathbf{y} - A\mathbf{c}_{t-1}))$ Katsayı vektörü güncellenir.

$\mathbf{a}_t \leftarrow A\mathbf{c}_t$, Yakınsama vektörü güncellenir.

$\mathbf{r}_t \leftarrow \mathbf{y} - \mathbf{a}_t$, Artık vektör güncellenir.

$t \leftarrow t + 1$, Döngü indeksi artırılır.

Koşul sağlandığında döngüden çık, (Çıktı: $\mathbf{x} = \mathbf{c}$).



Şekil 2.10. x işaretinin YSE algoritması ile elde edilmesi

YSE algoritması için OKHK 0.0029 hesaplanmış ve bu sonuçtan hareketle x işaretini geri çatmada YSE algoritması KDET algoritmasından daha iyi başarıma sahiptir.

2.9.7. Sıkıştırılmış Örneklemeli Eşlemeli Takibi Algoritması

Sıkıştırılmış Örneklemeli Eşlemeli Takip (SÖET) algoritması her adımda artık vektör kullanarak hesapladığı hatanın seyreklik değerinin 2 katı kadar en güçlü destek kümesini seçerek, en küçük kareler çözümü ile durma kriteri sağlanana kadar yinelemeye devam eder. SÖET algoritması Tablo 2.4'te detaylandırılmıştır [29, 61].

Tablo 2.4. SÖET algoritması

Girdi: A, y ; Çıktı x

Katsayı vektörü: $c_0 \leftarrow 0$, Artık vektör: $r_0 \leftarrow y$, döngü indeksi: $t \leftarrow 1$

Döngü

$e \leftarrow A^T r$, Hatayı güncelle.

$\Omega \leftarrow e_{2K}$, İndeks set

$T \leftarrow \Omega \cup \text{supp}(c_{t-1})$,

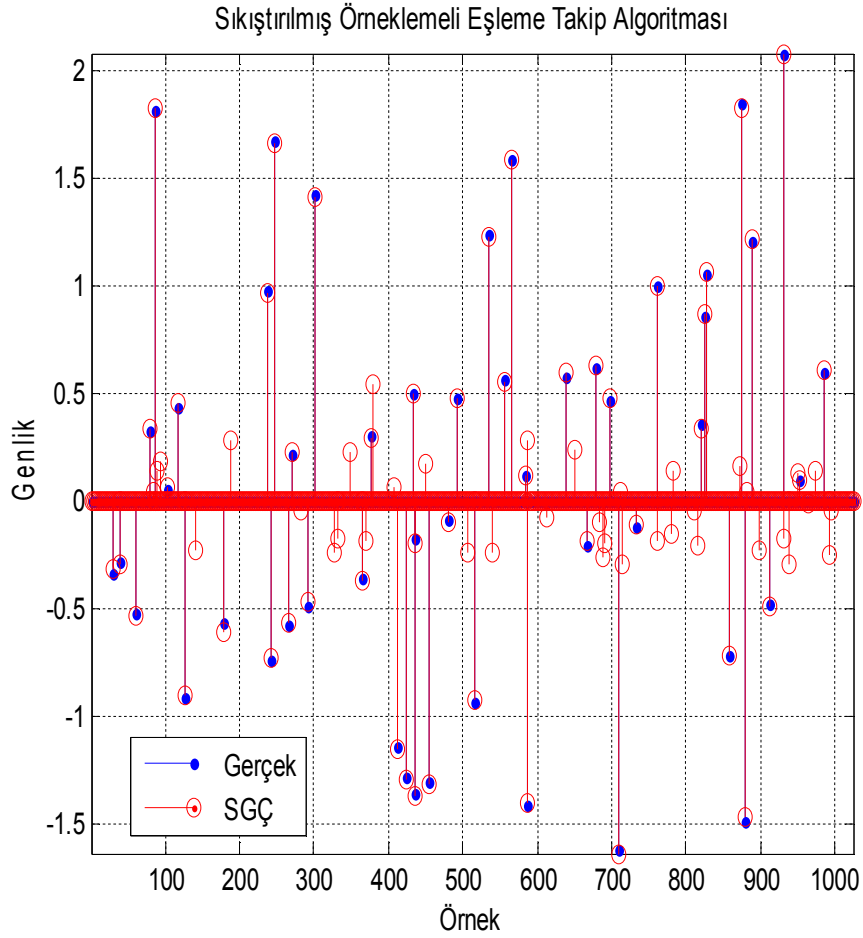
$b_t \leftarrow A_T \backslash y, b_{|T^c} \leftarrow 0$

$c_{0t} \leftarrow b_t, r \leftarrow y - Ac_t$

$t \leftarrow t + 1$, Döngü indeksi artırılır.

Koşul sağlandığında döngüden çık, (Çıktı: $x = c$).

Şekil 2.11'de gerçek ve SÖET algoritması kullanılarak geri elde edilmiş x seyrek işareti gösterilmiştir. SÖET algoritması için OKHK değeri 0.0392 olarak hesaplanmıştır. SÖET algoritması işaretin sıfırdan farklı değerlerini belirli bir hata oranı ile yakalamıştır. Ancak, SÖET algoritması işaretle olmayan bazı değerler de üretmiştir.



Şekil 2.11. x işaretinin SÖET algoritması ile elde edilmesi

2.9.8. İleri-Geri Takip Algoritması

Yinelemeli fırsatçı takip algoritmaları, SGÇ problemleri için alternatif bir yöntemdir [36]. Yinelemeli algoritmalar, işareti geri çatmak için tüm vektörleri kullanmak yerine uygun vektörleri kullandıkları ve bunların çarpım katsayılarının uygun seçiminden ötürü, işlem karmaşıklıkları az ve dolayısıyla hızlıdırlar [29, 41, 42]. İGT algoritması iki aşamalı eşiklemeli yinelemeli fırsatçı takip algoritmasıdır. İlk aşama olan ileri adımda, geri çatılmak istenen seyrek vektörün sıfır olmayan dizinlerini içeren destek kümesini ileri adım sayısı α kadar yeni dizin ekleyerek genişletir. İkinci aşama μ adım sayısından oluşan geri adımdır. Geri adımda destek kümesinden $\mu < \alpha$ adet dizin çıkarılır. Böylelikle destek kümesi her döngüde genişletilmiş olur. İGT algoritması Tablo 2.5'te özetlenmiştir [62, 63].

Tablo 2.5. İGT algoritması

Giris H : Ölçüm Matrisi, Y : Ölçüm işareti **Cıkıs** β : Çıkış ağırlıkları

(ε : minimum hata toleransı, K_{maks} : seyreklik seviyesi, T : indeks kümesi, r : rezidü, k : iterasyon sayısı)

Baslangıc $T^0 \leftarrow \emptyset$, $r^0 \leftarrow Y$, $k \leftarrow 1$

Döngü

İleri adım

$T_i = \operatorname{argmax}_{J:|J|=\alpha} \|H_J^* r^{k-1}\|_1$ İleri indeks setini oluştur.

$\tilde{T}^k = T^{k-1} \cup T_i$ İndeks setini genişlet.

$w = \operatorname{argmin}_w \|Y - H_{\tilde{T}^k} w\|_2$

Geri adım

$T_g = \operatorname{argmin}_{J:|J|=\mu} \|w_J\|_1$ Geri indeks seti oluştur.

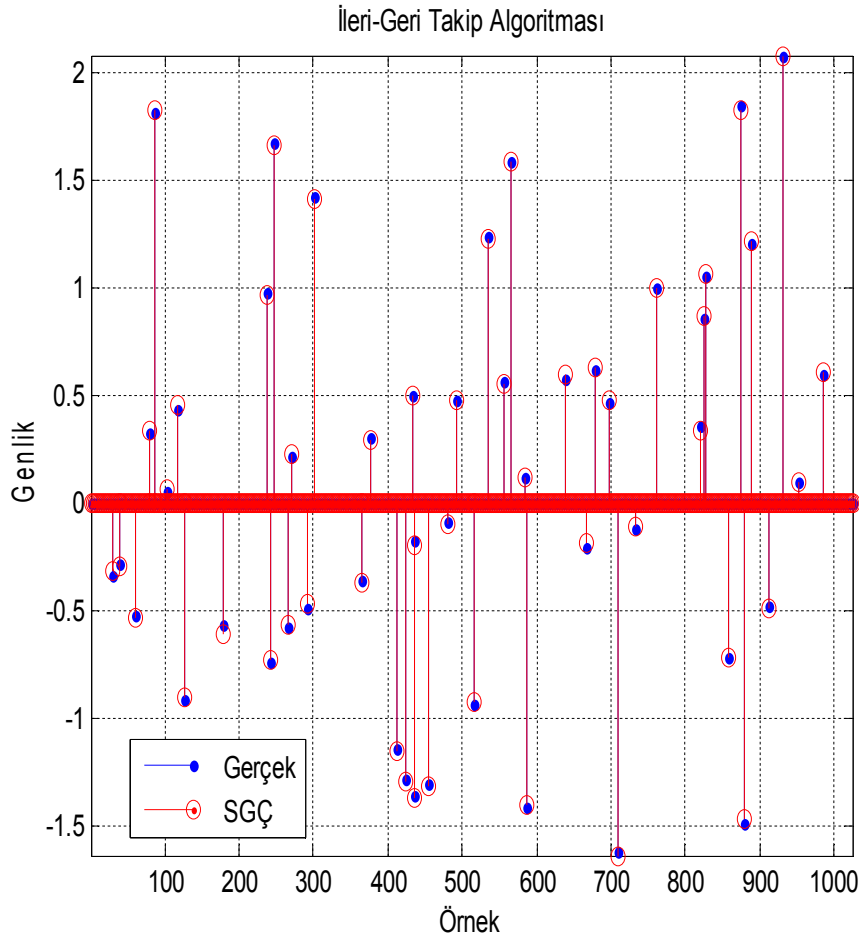
$T^k = \tilde{T}^k - T_g$ Geri indeks setlerini kaldır.

$w = \operatorname{argmin}_w \|Y - H_{T^k} w\|_2$

$r^k = Y - H_{T^k} w$ rezidü güncelle.

Sonlandırma $\|r^k\|_2 \leq \varepsilon \|Y\|_2$ veya $|T^k| \geq K_{maks}$

Cıkıs $\beta = w$



Şekil 2.12. x işaretinin İGT algoritması ile geri çatılması

İGT algoritması ile ilgili detaylı bilgi [38] numaralı çalışmada verilmiştir. Şekil 2.12’de seyrek bir x işaretinin İGT algoritması ile geri elde edilmiş hali verilmektedir. İGT algoritması yaklaşık 0.0029 OKHK değerine sahiptir. İGT algoritması OKHK kriteri göz önüne alındığında YSE ile aynı başarıma sahiptir.

2.9.9. Dikgen En Küçük Kareler

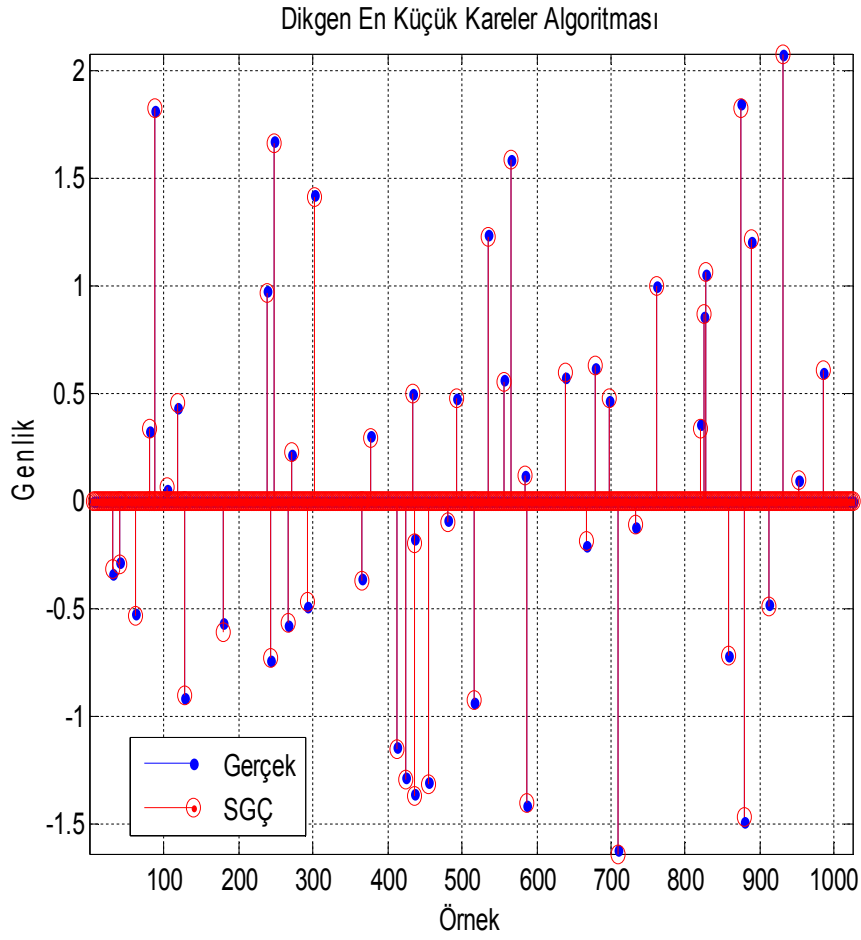
Dikgen ayrıştırma, EK problemlerinin çözümü için nümerik yöntemlerde sıklıkla tercih edilmektedir [64]. Ayrıca, radyal tabanlı ağların ağırlık parametresini hesaplamak için kullanılmaktadır. Dikgen En küçük Kareler (DEK) yöntemi Chen ve diğ. [65] tarafından radyal tabanlı ağlar için en uygun merkezi, geniş bir aday kümesi içinden seçmek için geliştirilmiştir. DEK yöntemi Eşleme Takip ve Dikgen Eşleme Takip algoritmaları gibi yinelemeli fırsatçı takip algoritmasıdır. Literatürdeki önemli

çalıřmalarda, DEK ve DET yöntemlerinin yapısal olarak birbirine benzetildiđi görölmektedir [64]. Bu benzetmeye rađmen, DEK yöntemi seçim adımında DET yönteminden ayrılmaktadır. Çünkü DEK yöntemi, minimum rezidü hatasına sahip ölçüm matrisinin kolon vektörünü dikgenleřtirme iřleminden sonra seçer [64, 68]. DEK yöntemi Tablo 2.6’da özetlenmiřtir [64].

Tablo 2.6. DEK algoritması.

Giriř: y , A Çıkıř: x
Bařla: $r^0 = t$, $c^0 = 0$, $\Lambda^0 = \emptyset$. $t = 1$
Döngü
$\alpha_t = \underset{\Lambda^t = \Lambda^{t-1} \cup \alpha}{argmin} \ y - A_{\Lambda_{\alpha}^t} A_{\Lambda_{\alpha}^t}^+ y\ _2$
$\Lambda^t = \Lambda^{t-1} \cup \alpha_t$
$c_{\Lambda^t}^t = A_{\Lambda_{\alpha}^t}^+ y$
$r^n = y - Ac^n$
$t = t + 1$
Döngü řartı sađlandığında dur,
Çıkıř: $x = c$

řekil 2.13’te seyrek x iřaretinin DEK algoritması ile geri elde edilmiř hali verilmiřtir. DEK algoritması için OKHK deđeri 0.0029 olarak hesaplanmıřtır.



Şekil 2.13. x işaretinin DEK algoritması ile elde edilmesi

2.9. Başarım Kriterleri ve Normalizasyon Yöntemleri

2.9.1. Ortalama Karesel Hatanın Karekökü

OKHK ifadesi eşitlik (2.29)'da verilmiştir [69-70].

$$OKHK = \sqrt{\frac{\sum_{i=1}^P (y_i - \hat{y}_i)^2}{P}} \quad (2.29)$$

Burada y_i veri setindeki gerçek değeri, $\hat{y}_i$ tahmin edilen değeri ve P ise y_i 'nin boyutunu göstermektedir.

2.9.2. Standart Sapma

Standart sapma veri değerlerinin dağılımı hakkında bilgi veren bir kriterdir. Standart Sapma σ denklemi (2.30)'da verilmiştir [71, 72].

$$\sigma = \sqrt{P^{-1} \sum_{i=1}^P (y_i - \bar{y})^2} \quad (2.30)$$

2.9.3. Korelasyon Katsayısı

Korelasyon katsayısı, bağımsız iki değişkenin arasındaki ilişkinin yönü ve büyüklüğü hakkında bilgi verir. Korelasyon Katsayısı (2.31)'de verilmiştir [73, 74].

$$kk = \frac{\sum_{i=1}^P (y_i - \bar{y}) \sum_{i=1}^P (\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^P (y_i - \bar{y})^2 \sum_{i=1}^P (\hat{y}_i - \bar{\hat{y}})^2}} \quad (2.31)$$

2.9.4. Ortalama Mutlak Hata

Ortalama Mutlak Hata (OMH) tahmin verisinin gerçek veriye ne kadar yakın olduğunu ölçmek için kullanılan bir büyüklüktür. OMH kriterini hesaplayan yöntem (2.32)'de verilmiştir [75, 76, 77].

$$OMH = P^{-1} \sum_{i=1}^P (y_i - \hat{y}_i) \quad (2.32)$$

2.9.5. Ortalama Mutlak Yüzde Hata

Ortalama Mutlak Yüzde Hata (OMYH) istatistikte doğruluğu ölçmek için kullanılan bir yöntemdir. Genellikle doğruluğu yüzde olarak göstermek için kullanılır ve (2.33)'te gibi formüle edilebilir.

$$OMYH = P^{-1} \sum_{i=1}^P \left(\frac{y_i - \hat{y}_i}{y_i} \right) \quad (2.33)$$

2.9.6. Sıfır Ortalama ve Birim Varyansa Göre Normalizasyon

Normalizasyon işleminde sıkça kullanılan yöntemlerden biri de Sıfır Ortalama ve Birim Varyansa (SOBV) yöntemidir. SOBV yöntemine göre verilerin yeni değerlere dönüştürülürken aritmetik ortalaması ve standart sapmanın da dikkate alınması prensibine dayanır. SOBV ifadesi (2.34)'te verilmiştir [80].

$$z = \frac{y - \bar{y}}{\sigma_y} \quad (2.34)$$

Burada z normalize edilmiş veriyi, y normalize edilecek veriyi, $\bar{y}$ normalize edilecek verinin aritmetik ortalamasını, σ_y ise normalize edilecek verinin standart sapmasını göstermektedir.

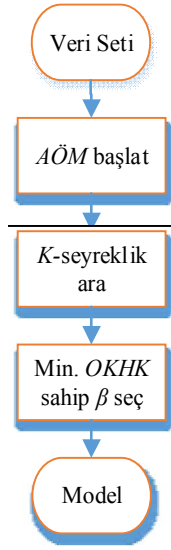
3. YİNELEMELİ SERT EŞİKLEME TABANLI AŞIRI ÖĞRENME MAKİNASI

3.1. Giriş

Bu bölümde, AÖM'nin çıkış ağırlıklarının hesaplanması için bir YFT algoritması önerilmiştir. Özellikle YSE algoritmasının en uygun gizli katman nöron sayısının elde edilmesi ve eğitim süreci karmaşıklığı gibi konularda AÖM ve düzenlenmiş AÖM yöntemlerine karşı üstünlüğü incelenmiştir. YSE algoritması ilgilenilen maliyet fonksiyonunu optimize etmektedir. Böylece YSE-AÖM yöntemi düşük işlem karmaşıklığına sahip olmaktadır [12, 14]. Literatürdeki düzenlenmiş AÖM çalışmalarına benzer şekilde YSE-AÖM tahmin problemlerine uygulanmıştır. Önerilen yöntemde YSE algoritması ilgili nöronları budamak için kullanılmıştır. Böylece en uygun ağ boyutu otomatik olarak ayarlanmaktadır. AÖM'de olduğu gibi, YSE-AÖM yönteminde de giriş katmanından gizli katmana bağlayan giriş ağırlıkları ve eşikler rastgele seçilmektedir. Daha sonra çıkış katmanı ağırlıkları YSE algoritması kullanılarak hesaplatılır. Böylece seyreklik sayesinde ilgisiz nöronlar otomatik olarak budanarak en uygun nöron sayısına sahip ağ elde edilmiş olur. Deneysel çalışmalardan elde edilen sonuçlar YSE tabanlı AÖM'nin, AÖM'ye üstünlüğünü göstermiştir.

3.2. Önerilen YSE-AÖM yöntemi

YSE algoritması m seyreklik seviyesi parametresinin belirlenmesine gerek duyar. m seyreklik parametresi modelde kullanılan gizli düğüm sayısı hakkında bilgi vermektedir. [15]'te belirtildiği gibi m_{min} dan m_{maks} 'a artan bir K dizisi tanımlanmıştır. Sonuç olarak K tane model elde edilmiştir. En iyi modeli seçmek için OKHK ölçütü dikkate alınmıştır. Önerilen yöntemin akış şeması Şekil 3.1'de gösterilmiştir.



Şekil 3.1. Önerilen YSE-AÖM yöntemi akış şeması

3.2. Deneysel Çalışmalar

Önerilen YSE-AÖM yönteminin başarımını test etmek için kullanılan veri setleri UCI [81] ve Statlib [82]’den elde edilmiştir. Kullanılan veri setleri hakkında bilgi Tablo 3.1’de özetlenmiştir.

Tablo 3.1. Kullanılan veri setleri

Veri Seti	Özellik Sayısı	Örnek Sayısı	
		Eğitim	Test
Breast Cancer	33	130	64
Pima Indians Diabetes	9	512	256
Bupa	7	230	115
Body Fat	15	168	84

Şekil 3.1’de gösterilen akış şemasının veri seti hazırlama adımında veri setleri SOBV göre normalize edilmiştir [83] ve rastgele üç parçaya bölünüp, üçte ikisi eğitim, kalanı ise test için kullanılmıştır.

Syreklilik seviyesi K parametresinin belirlenmesi için iki adımlı bir arama algoritması kullanılmıştır. İlk adımda K , Breast Cancer ve Body Fat veri setleri için 5’den 50’ye 5’lik artımla, Pima ve Bupa veri setleri içinse 10’dan 100’e 10’luk artımla arama algoritması gerçekleştirilmiştir. Her K değeri için OKHK hesaplanmış ve düşük OKHK

değerine sahip K diğer adım için seçilmiştir. İkinci adımda ise elde edilen uygun K değeri için K-5 den K+5'e 1'er artımla arama algoritması düşünülmüştür. En küçük OKHK değerine sahip en uygun K değeri seçilmiştir. Deneyler 100 defa tekrarlanmış ve elde edilen sonuçların OKHK'sı, standart sapması ve işlem sürenin ortalaması hesaplanmıştır. YSE-AÖM yöntemi için elde edilen sonuçlar Tablo 3.2'de verilmiştir. Tablo 3.2'de ortalama OKHK, ortalama standart sapma σ , gizli katman nöron sayısı Nöron ve ortalama süre Süre (saniye biriminde) ile gösterilmiştir.

Tablo 3.2. YSE-AÖM sonuçları

Veri seti	OKHK	σ	Nöron	Süre (s)
Breast Cancer	0.2665	0.0094	200(14)	0.0850
Pima I. D.	0.4095	0.0067	200(16)	0.2114
Bupa	0.2356	0.0074	400(14)	0.2081
Body Fat	0.0273	0.0039	250(52)	0.0816

Tablo 3.2'den görüldüğü gibi en küçük ortalama OKHK değeri ve ortalama standart sapma değeri Body Fat veri seti için elde edilmiştir. Deneysel çalışmada Breast Cancer ve Pima Indians Diabetes veri setleri için 200 nöron, Bupa veri seti için 400 nöron ve Body Fat veri seti için 250 nöron düşünülmüştür, en düşük sayıda nöron sayısına sahip model (14 nöron) Breast Cancer ve Bupa veri setleri için elde edilmiştir. Deneysel çalışmada en düşük nöron sayısına sahip model Breast Cancer ve Bupa veri setlerinde elde edilirken ilgili ortalama OKHK değerleri en fazla nöron kullanan Body Fat veri seti kadar iyi değildir. Ortalama OKHK açısından en kötü sonuç Pima Indians Diabetes veri setinde elde edilmiştir.

Deneysel çalışma normal AÖM'ye YSE-AÖM'de için kullanılan veri setleri uygulanarak sürdürülmüştür. AÖM ile aynı şartlarda bire bir karşılaştırma yapabilmek için bütün veri setleri aynı yolla işlenmiştir. AÖM yönteminde en uygun nöron sayısı YSE-AÖM'ye benzer yolla elde edilerek, her bir veri seti için 100 defa döngüye sokulup elde edilen sonuçların ortalaması hesaplanmıştır. AÖM için elde edilen sonuçlar Tablo 3.3'de YSE-AÖM yöntemi ile karşılaştırmalı olarak verilmiştir. Tablo 3.3'ten görüldüğü gibi önerilen yöntem Breast Cancer, Pima I.D., ve Bupa veri setleri için daha iyi ortalama OKHK değeri ortaya koymuştur. YSE-AÖM yöntemi Body Fat veri seti için ortalama OKHK açısından AÖM yönteminden daha kötü başarıma sahiptir. Ortalama standart

sapma açısından Body Fat veri seti hariç diğer veri setlerinde YSE-AÖM yöntemi üstün başarımlar göstermiştir. Gizli katman nöron sayısı bakımından YSE-AÖM yöntemi Breast Cancer için yaklaşık 64 kat, Pima için yaklaşık 3 kat, Bupa için yaklaşık 6 kat ve Body Fat için 15 kat daha az nöron sayısına sahiptir.

Tablo 3.3. AÖM ile YSE-AÖM yönteminin karşılaştırılması

Veri Seti	AÖM				YSE-AÖM			
	OKHK	σ	Nöron	Süre (s)	OKHK	σ	Nöron	Süre (s)
Breast C.	0.3236	0.0164	904	0.1072	0.2665	0.0094	14	0.0850
Pima I. D.	0.4269	0.0125	54	0.6533	0.4095	0.0067	16	0.2114
Bupa	0.5244	0.1114	95	0.2062	0.2356	0.0074	14	0.2081
Body F.	0.0075	0.0004	780	0.1266	0.0273	0.0039	52	0.0816

Bupa veri setinde AÖM nöron sayısı 95 olduğundan geçen süre yinelemeli yapıya sahip olan YSE-AÖM yönteminden daha kısadır. Deney işlem süresi bakımından YSE-AÖM yöntemi Bupa veri seti dışındaki bütün setlerinde daha üstün başarımlara sahiptir. Tablo 3.3'ten görüldüğü gibi önerilen YSE-AÖM yöntemi AÖM ile karşılaştırıldığında daha iyi başarımlara sahiptir.

4. REGRESYON PROBLEMLERİ İÇİN DİKGEN EŞLEME TAKİP ALGORİTMASI TABANLI AŞIRI ÖĞRENME MAKİNASI

4.1. Giriş

Bu bölümde AÖM'nin çıkış ağırlıkların seyrek olarak elde edilmesi için düşük işlem karmaşıklığına ve tekillik problemine dayanıklı bir algoritma önerilmiştir. Önerilen bu yöntem yaygın kullanılan regresyon problemine uygulanmıştır. DET-AÖM yönteminin amacı en uygun nöron sayısına sahip AÖM ağını oluşturmaktır. Önerilen DET-AÖM yönteminin YSA, AÖM, EMDSO ve EN yöntemleri ile karşılaştırılması ayrıntılı olarak incelenmiştir.

4.2. Geliştirilmiş AÖM Yöntemleri

Bu bölümde literatürde önerilmiş başlıca AÖM yöntemleri incelenmiştir.

4.2.1. Düzenlenmiş AÖM Yöntemleri

Bölüm 2'de anlatıldığı gibi tekillik AÖM için önemli bir problemdir. Özellikle gizli katman nöron sayısı giriş veri sayısından fazla olduğu durumlarda veya gizli katman çıkış matrisi atomları birbiri ile ilişkili olduğu durumlarda AÖM'nin tekillik problemi söz konusudur [15]. AÖM çıkış ağırlıklarını hesaplatan EK çözümü (4.1)'deki gibi yazılabilir.

$$L(X, Y; \beta) = \|Y - H\beta\|^2 \quad (4.1)$$

Araştırmacılar tekillik problemini çözmek için (4.1) ifadesine L_2 -normu sınırlandırıcısı eklemeyi önermişlerdir [15, 21, 84]. Bu yaklaşım (4.2)'de formüle edilmiştir.

$$L(X, Y; \beta, \lambda) = \|Y - H\beta\|^2 + \lambda_2 \|\beta\|^2 \quad (4.2)$$

Burada λ_2 regülerizasyon parametresidir. Çıkış ağırlıkları (4.3)'ten hesaplanır [21].

$$\beta = (H^T H + \lambda_2 I)^{-1} H^T \quad (4.3)$$

Burada I birim matrisi göstermektedir.

4.2.2. En Küçük Açısai Regresyon AÖM Yöntemi

Bölüm 1 ve 2’de vurgulandığı gibi düşük sayıda nöronlu AÖM ağı veriyi modellemede başarısız olabilirken, çok fazla sayıda nöron seçildiğinde ise ezberlemeye yol açabilir [16-84]. İlk önce AÖM ağı gerekenden fazla nöron sayısı ile başlatılır ve ardından AÖM ağındaki gereksiz nöronları elimine etmek için bir L_1 -normu sınırlandırıcısı kullanılmıştır. Sonuç olarak, AÖM yönteminin geliştirilmesini sağlayan optimum nöronların belirlenmesi işlemi otomatik olarak yapılmıştır [16]. En küçük Açısai Regresyon tabanlı AÖM (EAR-AÖM) yöntemi (4.4)’te formüle edilmiştir.

$$L(X, Y; \beta, \lambda) = \|Y - H\beta\|^2 + \lambda_1 \|\beta\|^1 \quad (4.4)$$

Eşitlik (4.4) En küçük Mutlak Daralma ve Seçme Operatörü (EMDSO) problemi olarak da bilinmektedir. Burada λ_1 sınırlama faktörünü ve $\|\cdot\|^1$ ise L_1 -normu sınırlandırıcısını temsil etmektedir. EAR algoritması kullanılarak (4.4) ifadesi etkin bir şekilde çözdürülmektedir [16, 24, 84].

4.2.3. En Küçük Açısai Regresyon Elastik Net AÖM Yöntemi

EAR-AÖM yöntemi Gram matrisinden dolayı tekillik problemi ile karşı karşıya kalabilmektedir [16, 24]. Bu durum Gram matrisi tam ranka sahip olmadığında EAR çözümünün kararsız olması olarak da bilinmektedir. EAR-AÖM’nin bu eksikliğini üstesinden gelebilmek için bir L_2 -normu sınırlandırıcısı eklenerek EAR-EN-AÖM olarak isimlendirilen yeni bir yöntem önerilmiştir. EAR-EN-AÖM yöntemi (4.5)’te ifade edilmiştir.

$$L(X, Y; \beta, \lambda_1, \lambda_2) = \|Y - H\beta\|^2 + \lambda_1 \|\beta\|^1 + \lambda_2 \|\beta\|^2 \quad (4.5)$$

Bu yaklaşım bir SEN yöntemidir ve EMDSO problemi olarak (4.6)’daki gibi yazılabilmektedir [16].

$$L(X, Y^*; \beta^*, \gamma) = \|Y^* - H^* \beta^*\|^2 + \lambda \|\beta^*\|^1 \quad (4.6)$$

Burada;

$$Y^* = \begin{pmatrix} Y \\ O_{N \times 1} \end{pmatrix}, H^* = (1 + \lambda_2)^{-1/2} \begin{pmatrix} H \\ \sqrt{\lambda_2} I_{N \times N} \end{pmatrix}$$

$$\beta^* = \sqrt{1+\lambda_2} \beta, \lambda = \lambda_1 \sqrt{1+\lambda_2}$$

olarak ifade edilebilmektedir.

Bu tanımlamalardan sonra (4.6) EAR yöntemiyle çözdürülebilir;

$$\beta^* = \arg \min_{\beta} L(X, Y^*; \beta^*, \gamma) \quad (4.7)$$

Böylece (4.7) ifadesinin çözümü;

$$\beta = \frac{1}{\sqrt{1+\lambda_2}} \beta^* \quad (4.8)$$

olarak elde edilebilmektedir. Eşitlik (4.5)'in tekillik problemi için çözüm getirdiği görülmekte iken bu algoritmada ayarlanması gereken iki parametre vardır (λ_1, λ_2). Söz konusu bu parametre ayarlama durumu yavaş eğitim sürecine yol açabileceğinden EAR-EN-AÖM yöntemi için bir kısıtlama oluşturmaktadır [16].

4.2.4. Tikhonov'un Düzenlenmiş En Uygun Budamalı AÖM Yöntemi

EAR-AÖM'nin tekillik problemi Miche ve diğ. tarafından önerilmiştir [22]. TDEB-AÖM yönteminde ilk önce EAR algoritması kullanılarak AÖM ağındaki gereksiz nöronlar kaldırılmaktadır. Daha sonra AÖM ağına düzenlenmiş EK problemini çözmek için bir L_2 -normu sınırlandırıcısı eklenmiştir. Ayrıca araştırmacılar [22] numaralı çalışmada λ_1 ve λ_2 parametrelerini ayrı ayrı ayarlayarak parametre ayarlama sürecini hızlandırmayı amaçlamışlardır. Araştırmacıların bu yaklaşımına göre $\lambda_2=0$ kabul edilerek λ_1 parametresi, ardından ayarlanmış λ_1 sabit tutularak λ_2 parametresi ayarlanmaktadır. Bu işlem ile en uygun λ_1 ve λ_2 değerleri elde edilmeye çalışılmıştır [22]. Yazarlar, λ_1 ve λ_2 parametrelerinin aynı uzayda olmadığı sürece en uygun parametrenin elde edilemeyeceğini vurgulamaktadır.

4.3. Önerilen Dikgen Eşleme Takip AÖM Yöntemi

Bölüm 2.9.4'te anlatıldığı gibi DET algoritması seyreklik seviyesinin belirlenmesine gereksinim duymaktadır. Bu parametre AÖM ağının gizli katmanında kaç adet nöronun aktif olarak kullanılacağı hakkında bilgi vermektedir. [85] numaralı çalışmada belirtildiği gibi, m_{maks} 'dan m_{min} 'a azalan logaritmik bir K dizisi oluşturulmuştur.

Böylece K adet model oluşturulmuştur. K adet model içinden en iyi modeli seçmek için OKHK kriteri kullanılmıştır. Önerilen DET-AÖM yöntemi işlem adımları;

- AÖM ağının (2.4)'te verilen w ve b parametrelerini başlat
- Gizli katman nöron sayısı N 'yi ve DET algoritması seyreklik seviyesi K 'yi seç
- Gizli katman çıkış matrisi H' 'ı hesapla
- Çıkış ağırlıkları β 'yi hesaplamak için DET algoritmasını kullan
- Her bir seyreklik seviyesi K 'ya ait model için OKHK'yı hesapla
- En düşük OKHK değerine sahip β katsayılarını seç

olarak özetlenebilir.

4.4. DeneySEL Çalışma

Bu çalışmada UCI makina öğrenmesi veri merkezinden [81] alınan veri setleri kullanılmıştır. Deneyde kullanılan veri setleri karşılaştırma yapılacak yöntemler ile uygunluk sağlanması aynı seçilmiştir. İlgili veri setleri hakkında özet bilgiler Tablo 4.1'de verilmiştir. Veri setlerine SOBv normalizasyon işlemi uygulanmıştır [83].

Tablo 4.1. Kullanılan veri setleri hakkında özet bilgiler

Veri seti	Özellik Sayısı	Örnek Sayısı	
		Eğitim	Test
Abalone	8	2784	1393
Ailerons	40	9166	4584
Auto Price	15	106	53
Auto MPG	7	265	133
Boston Housing	13	337	169
California Housing	8	13760	6880
Delta Ailerons	5	4752	2377
Delta Elevators	6	6344	3173
Triazines	60	106	53

Diğer yöntemler ile geçerli bir karşılaştırma için eğitim ve test veri setleri aynı yöntemlerle işlenmiştir. Bütün veri setlerinin üçte ikisi eğitim için kalan üçte biri ise test için kullanılmıştır. Optimum seyreklik seviyesi belirlemek için iki seviyeli arama

algoritması kullanılmıştır. İlk adım kaba arama mekanizmasıdır. Her veri seti için kaba arama aralık ve artım miktarları Tablo 4.2’de gösterilmiştir.

Arama algoritmasının ikinci adımında en iyi OKHK değerine sahip K değeri birer artım ile $K \pm 10$ olacak şekilde yeniden aratılmıştır. Öncekine benzer şekilde bu aralıkta en düşük OKHK değerine sahip K değeri optimum değer olarak belirlenmiştir. Bütün deneyler her bir veri seti 100 kere döndürülmüştür ve başarımların ortalaması hesaplatılmıştır.

Tablo 4.2. Veri setleri için kullanılan artım ve aralık miktarları

Veri Seti	Artım	Aralık	
		Min.	Maks.
Auto Price, Triazines	10	10	100
Abalone, Auto MPG, Boston Housing, Delta E.	20	20	200
Ailerons, California Housing, Delta A.	50	100	500

DET-AÖM yöntemi sonuçları Tablo 4.3’te verilmiştir. En iyi değerler Tablo 4.3’te koyu olarak vurgulanmıştır. Tablo 4.3’ten görüleceği gibi Auto Price veri seti için en düşük ortalama OKHK değeri elde edilmiştir. Auto Price veri seti için deneyler 100 nöron ile başlatılmasına rağmen optimum nöron sayısı 16 olarak elde edilmiştir. Ayrıca en düşük nöron sayısına sahip Triazines veri seti için ortalama OKHK açısından Auto Price veri setinde olduğu gibi bir başarımlar elde edilememiştir. Ortalama OKHK açısından Ailerons, Delta A. ve Triazines veri setlerinde en kötü sonuçlara ulaşılmıştır. Tablo 4.3’ten önemli bir gözlem ise standart sapma düşük kabul edilebilmektedir. Bu durum önerilen DET-AÖM yönteminin kararlılığını göstermektedir. Deneylerde son nöron sayısı Tablo 4.3’ün ilk sütununda parantez içinde verilmiştir. Bu durum AÖM ile karşılaştırıldığında nöron sayısı azaltılarak işlem karmaşıklığının giderildiğini göstermektedir. Önerilen yöntemin ileri değerlendirmesi için Korelasyon Katsayısı ve OMH ve bunların standart sapmaları hesaplatılmıştır [86]. İlgili sonuçlar Tablo 4.3’te verilmiştir.

Tablo 4.3. DET-AÖM yöntemi sonuçları

Veri Seti	Nöron	OKHK	OKHK σ	Korelasyon. Katsayısı	Kor. Kat. σ	OMH	OMH σ
Abalone	200(38)	0.0743	5.27e-4	0.7425	4.9 e-3	0.0533	3e-4
Auto Price	100(16)	0.0720	8.2e-3	0.8753	2.75 e-2	0.0554	5.9e-3
Ailerons	1000(72)	0.1833	4.099e-4	0.5782	3e-3	0.1419	4e-4
Boston H.	200(40)	0.0898	8.1e-3	0.8858	1.69 e-2	0.0632	3.6e-3
California H.	1000(150)	0.1160	6.884e-4	0.8457	3.3 e-3	0.0883	9e-4
Delta E.	200(33)	0.1084	2.71e-4	0.8011	1.1 e-3	0.0821	2e-4
Delta A.	200(100)	0.1842	1.4e-3	0.8503	2.9 e-3	0.1413	1.4e-3
Auto MPG	100(20)	0.1056	8.7e-3	0.8913	1.62 e-2	0.0729	7e-3
Triazines	100(16)	0.1774	1.41e-2	0.3526	6.46e-2	0.1252	6.7e-3

Deneyler AÖM, EN, EMDSO karşılaştırmaları yapılacak şekilde sürdürülmüştür. Bu karşılaştırmada Abalone, Auto Price, Ailerons, Boston Housing, Delta E. Ve California Housing veri setleri kullanılmıştır. Belirtilen bu veri setleri için AÖM, EN ve EMDSO yöntemleri sonuçları [15]'den karşılaştırma yapmak amacıyla edinilmiştir. Sonuçlar Tablo 4.4'te gösterilmiştir. Önerilen DET-AÖM yöntemi diğer yöntemler ile karşılaştırıldığında daha üstün başarıma sahip olduğu Tablo 4.4'ten görülmektedir. Özellikle DET-AÖM yöntemi Abalone, Auto Price ve Boston Housing Veri setlerinde OKHK kriteri açısından daha başarımlı göstermiştir. Ayrıca DET-AÖM yöntemi standart sapma kriterinde de diğer yöntemler ile karşılaştırıldığında daha iyi sonuç üretmektedir. Sonuçlar değerlendirildiğinde genel olarak DET-AÖM yöntemi tartışmasız olarak en uygun gizli katman nöron sayısına sahip olduğu söylenebilmektedir.

Tablo 4.4. DET-AÖM ile YSA, AÖM, EN ve EMDSO ile karşılaştırması

		Veri Seti					
		Abalone	Auto P.	Aileron	Boston H.	Delta E.	California H.
YSA	OKHK	0.1382	0.4257	0.1869	0.1178	0.1109	0.1057
	σ	0.0506	0.1113	0.0038	0.0340	0.0007	0.0018
	Nöron	40	20	30	10	20	20
AÖM	OKHK	0.6558	0.4051	0.4527	0.4867	0.6036	0.5071
	σ	0.007	0.041	0.004	0.052	0.002	0.004
	Nöron	40	20	600	80	100	400
EN	OKHK	0.6518	1.036	0.4476	0.4587	0.6045	0.5114
	σ	0.004	0.019	0.005	0.033	0.002	0.004
	Nöron	200(26.1)	50(1.1)	1000(146.4)	200(39.8)	200(25.4)	1000(153.3)
EMDSO	OKHK	0.6499	1.0281	0.4471	0.4465	0.6057	0.5078
	σ	0.003	0.023	0.004	0.031	0.002	0.004
	Nöron	200(21.6)	50(2.1)	1000(142)	200(41.4)	200(16.8)	1000(149.1)
DET-AÖM	OKHK	0.0743	0.0720	0.1833	0.0898	0.1084	0.1160
	σ	5.27e-4	0.0082	4.099e-4	0.0081	2.71e-4	6.884e-4
	Nöron	200(38)	200(16)	1000(72)	200(40)	200(33)	1000(150)

Ayrıca bu çalışmada aynı veri setleri kullanılarak YSA ile karşılaştırma yapmak amacıyla tekrar yapılmış ve sonuçlar Tablo 4.4'te gösterilmiştir. YSA mimarisinde üç-katmanlı ve gizli katmanda veri setine bağlı olarak (Tablo 4.4'te belirtildiği gibi) nöron sayısı seçilmiştir. YSA gizli katmanında tanjant-sigmoid ve çıkış katmanında doğrusal aktivasyon fonksiyonu kullanılmıştır. YSA Levenberg-Marquardt geri yayılım öğrenme algoritması kullanılarak eğitilmiştir. DET-AÖM yöntemi elde edilen ortalama OKHK değerlerine göre Abalone, Auto Price, Boston Housing ve Delta E. veri setlerinde YSA'dan daha üstün başarıma sahiptir. YSA California Housing veri setinde önerilen yöntemden daha iyi başarımlar göstermektedir. Ayrıca σ kriteri dikkate alındığında DET-AÖM'nin daha kararlı bir yöntem olduğu görülmektedir.

5. YFT-SAÖM: YİNELEMELİ FIRSATÇI TAKİP ALGORİTMALARI TABANLI SEYREK AŞIRI ÖĞRENME MAKİNASI

5.1. Giriş

Bu bölümde AÖM'yi teklik, parametre ayarlaması, en uygun gizli katman nöron sayısı ve eğitim karmaşıklığı gibi problemlere karşı güçlendirmek için YFT algoritmaları tabanlı SAÖM yöntemi önerilmiştir. YFT-SAÖM yönteminde de diğer önerilen yöntemlerde olduğu gibi AÖM'nin çıkış ağırlıkları seyrek kabul edilerek YFT algoritmaları hesaplatılması düşünülmüştür. Bu çalışmanın temel amacı SÖET, YSE, DET ve KDET algoritmalarının uygunluğunu araştırmaktır. YFT-SAÖM yöntemlerinin başarımlı regresyon problemleri kullanılarak test edilmiştir. İlk başta büyük ağ mimarisi seçilmiştir. Ardından önerilen YFT algoritmaları kullanılarak en uygun nöron sayısı elde edilmeye çalışılmıştır. Deneysel çalışmada ilk önce YFT algoritmalarının birbirleri arasında karşılaştırmaları yapılmıştır. Daha sonra kabul edilebilir başarıma sahip yöntem temel AÖM ve diğer düzenlenmiş AÖM yöntemleri ile karşılaştırılmıştır.

5.2. Önerilen YFT-SAÖM Yöntemi

Bu bölümde önerilen yöntem hakkında tanıtıcı bilgiler ve deneysel çalışma hakkında detaylar verilmiştir. Deneyde kullanılan bütün YFT yöntemleri için döngü durdurma kriteri olarak seyreklik seçilmiştir. YFT-AÖM yönteminde gizli katman nöron sayısı ilk başta gerekenden fazla seçildiğinden seyreklik parametresi gizli katmanda aktif olarak kaç nöron kullanılacağı hakkında bilgi içermektedir. Ayrıca YSE ve KDET algoritmaları için bir eşik değerleri gereklidir. Her iki yöntem için de eşik değerleri 0.5 olarak seçilmiştir. Seyreklik seviyesi K 'nın belirlenmesi için m azalma ile m_{maks} 'dan m_{min} 'e azalan K tane dizi oluşturulmuştur. En uygun K değerinin belirlenmesi için OKHK kriteri kullanılmıştır. Önerilen YFT-AÖM yöntemi işlem adımları Tablo 5.1'de gösterilmiştir.

Tablo 5.1. Önerilen YFT-SAÖM yöntemi

Adım1:	AÖM ağının (2.4)'de verilen w ve b parametrelerini başlat.
Adım2:	Gizli katman nöron sayısı N 'yi ve DET algoritması seyreklik seviyesi K 'yı seç.
Adım3:	Gizli katman çıkış matrisi H 'ı hesapla.
Adım4:	Çıkış ağırlıkları β 'yı hesaplamak için bir YFT algoritmasını kullan.
Adım5:	Her bir seyreklik seviyesi K 'ya ait model için OKHK'yı hesapla.
Adım6:	En düşük OKHK değerine sahip β katsayılarını seç.

5.3. Deneysel Çalışmalar

Bu bölümde UCI [81] makina öğrenmesi veri merkezi ve Statlib [82] veri arşivinden alınan veri setleri kullanılarak önerilen YFT-SAÖM yönteminin başarımı incelenmiştir. Özellikleri Tablo 5.2'de verilen dokuz veri seti ikili sınıflı veya çok sınıflı problemlerden oluşmaktadır. Bu veri setleri [15, 22] çalışmalarında vurgulandığı gibi örnek ve değişken sayıları bakımından heterojen olmaları nedeniyle seçilmiştir.

Tablo 5.2. Deneylerde kullanılan veri setleri hakkında bilgi.

Veri Seti	Özellik	Örnek Sayısı	
	Sayısı	Eğitim	Test
Abalone	8	2784	1393
Auto price	15	106	53
Body Fat	15	168	84
Boston Housing	13	337	169
Breast Cancer	33	130	164
Bupa	7	230	115
Pima Indians Diabetes	9	512	256
Servo	5	112	55
Triazines	60	106	53

Deneysel çalışmalara başlamadan önce en uygun nöron sayısını belirleme işlemini açıklamak yerinde olacaktır. Daha önceki çalışmalarda da olduğu gibi gizli katman nöron sayısı gerekenden fazla üretilir ve önerilen bir YFT algoritması ile seyreklik seviyesi K 'ya bağlı seyrek çıkış ağırlıkları katsayı vektörü elde edilmektedir. Böylece çıkış ağırlıklarının

sıfırdan farklı katsayıların sayısı gerekli gizli katman nöron sayısını göstermektedir. Deneysel çalışmalar aşağıdaki adımlar ışığında yapılmıştır.

1. YFT-SAÖM yöntemlerinin kapsamlı bir şekilde analiz edilerek en iyi yöntem, AÖM ile karşılaştırılmıştır.

2. Önerilen YFT-SAÖM yönteminin model boyutu ve tahmin doğruluğu bakımından başarımının geçerliliğini göstermek için EN ve EMDSO yöntemleri ile karşılaştırılması yapılmıştır.

Bütün veri setleri aynı yöntemle işlenmiştir. Veri setleri üçe bölünüp veri setinin üçte ikisi eğitim kalan üçte birlik kısım ise test için kullanılmıştır. Deneyler MATLAB ortamında 2.2 GHz işlemci 8GB ram sahip bilgisayarda gerçekleştirilmiştir. Bütün veri setleri SOBV'ye göre normalize edilmiştir. YFT-SAÖM ve AÖM yöntemlerinin gizli katmanında sigmoid ve çıkış katmanında ise doğrusal aktivasyon fonksiyonu kullanılmıştır. Daha önceki çalışmalarda olduğu gibi YFT algoritmaları döngü durdurma kriteri olarak seyreklik parametresi K kullanılmıştır. Seyreklik parametresi K 'nın belirlenmesi için iki aşamalı arama algoritması kullanılmıştır. İlk aşama kaba arama algoritmasıdır ve kaba arama algoritmasında her veri seti için artım ve aralıklar Tablo 5.3'te gösterilmiştir.

Tablo 5.3. Veri setleri için kullanılan artım ve aralık miktarları

Veri Seti	Artım	Aralık	
		Min.	Maks.
Auto P., Body Fat, Boston H., Breast Cancer, Pima I. D., Servo, Triazines	5	5	50
Abalone	10	10	100
Ailerons, Delta E.	20	20	200
California Housing	50	50	500

En uygun K bulunması işleminde her model için başarımlık kriteri olarak OKHK değeri dikkate alınmıştır. İkinci arama aşamasında ilk aşamada en düşük OKHK değerli K için aralık $K \pm 10$ birer artımlı olacak şekilde belirlenerek tekrar OKHK değeri hesaplatılmıştır. Böylece OKHK kriterine bağlı olarak en uygun K değeri elde edilmiştir.

Deneyler her veri seti için 100 kere tekrarlanmıştır ve OKHK ve OKHK değerinin standart sapmasının ortalaması hesaplatılmıştır.

5.3.1. YFT-AÖM Yöntemlerinin Karşılaştırılması

Farklı nöron sayılarına YFT-AÖM yöntemleri karşılaştırılması yapılmıştır. Önerilen dört YFT algoritması ortalama OKHK, OKHK standart sapması, gizli katman nöron sayısı ve deney (eğitim+test) süresi Tablo 5.4’de verilmiştir.

Tablo 5.4. YFT-AÖM yöntemlerinin karşılaştırılması

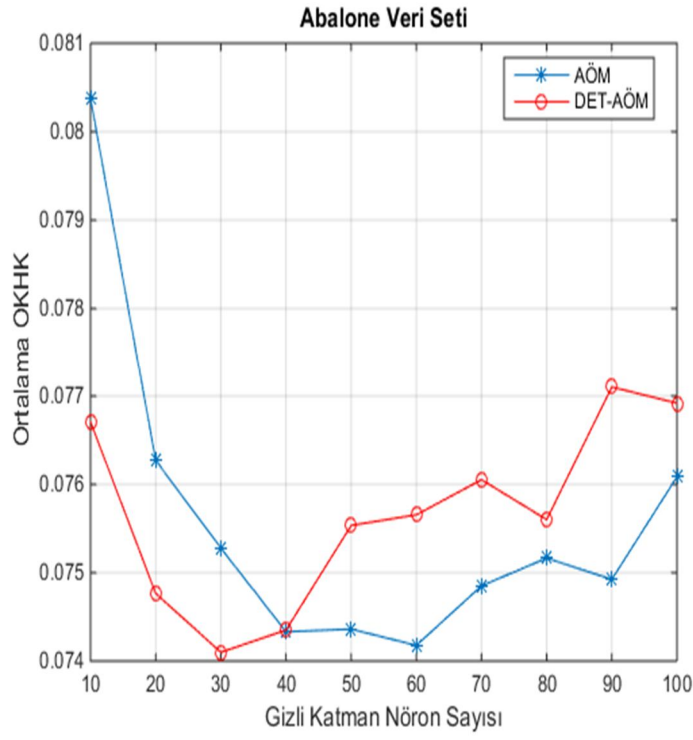
		Veri Seti								
		Abalone	Auto P.	Body F.	Boston H.	Breast C.	Bupa	Pima I.D.	Servo	Triazines
SÖET	OKHK	0.0764	0.0988	0.0232	0.0947	0.3064	0.2858	0.4264	0.1114	0.2238
	σ	0.0015	0.0215	0.0039	0.007	0.0261	0.0257	0.0110	0.0109	0.0278
	Nöron	(100)100	(50)50	(50)50	(50)50	(50)50	(50)50	(50)50	(50)50	(50)50
	Süre (s)	622.7	23.57	29.99	33.66	25.95	28.31	51.85	23.34	22.67
YSE	OKHK	0.0747	0.0775	0.0234	0.0973	0.2740	0.2394	0.4094	0.1068	0.2131
	σ	7.3e-4	0.0071	0.0041	0.0048	0.0131	0.0099	0.007	0.0102	0.0264
	Nöron	(29)100	(21)50	(25)50	(35)50	(16)50	(11)50	(17)50	(30)50	(35)50
	Süre (s)	4.52	0.135	8.307	0.223	6.173	5.605	15.34	0.19	12.80
DET	OKHK	0.0743	0.0731	0.0230	0.0949	0.2663	0.2407	0.4039	0.1057	0.1845
	σ	6.2e-4	0.0081	0.0037	0.0061	0.0101	0.0102	0.0047	0.0086	0.0145
	Nöron	(37)100	(13)50	(37)50	(25)50	(6)50	(14)50	(7)50	(20)50	(10)50
	Süre (s)	17.37	0.367	1.938	1.093	0.206	0.432	0.324	0.568	0.281
KDET	OKHK	0.0766	0.0942	0.0231	0.0929	0.3069	0.2825	0.4243	0.1112	0.2180
	σ	0.0014	0.0157	0.0035	0.0064	0.0283	0.0227	0.0120	0.0104	0.0230
	Nöron	(26)100	(11)50	(15)50	(9)50	(15)50	(35)50	(15)50	(16)50	(14)50
	Süre (s)	129.3	2.070	3.102	3.746	2.974	9.339	5.644	2.822	2.723

Tablo 5.4 incelendiğinde DET algoritmasının veri setlerinin çoğunda diğer YFT yöntemlerinden daha hızlı olduğu görülmektedir. Ayrıca DET algoritması ortalama OKHK ve standart sapması açısından diğer YFT yöntemlerinden daha üstün başarıma sahiptir. YSE yöntemi OKHK ve standart sapma kriterleri açısından önerilen YFT yöntemleri arasında ikinci OKHK ve standart sapma değerlerine sahiptir. Deneylerde en kötü sonuçlara sahip yöntem SÖET algoritması olmuştur. Ek olarak önerilen YFT algoritmaları

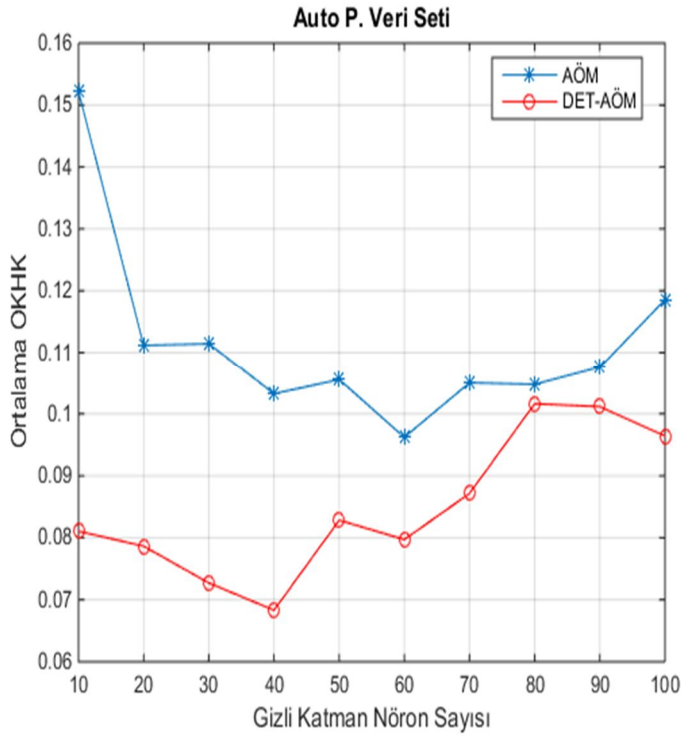
içinden KDET algoritmasının en düşük nöron sayısına sahip olduğu Tablo 5.4'ten açıkça görülmektedir. Temel AÖM ile en iyi YFT yönteminin karşılaştırılması çok önemlidir. AÖM ile en iyi YFT yöntemi karşılaştırma sonuçları Tablo 5.5'te verilmiştir. Abalone, Auto P., Breast Cancer, Bupa, Pima Indian Diabetes., Servo ve Triazines veri setleri için DET algoritması tabanlı AÖM yöntemi ortalama OKHK ve standart sapma açısından AÖM yönteminden daha üstün sonuçlar üretmiştir. AÖM yönteminin sadece Boston Housing veri seti için en iyi ortalama OKHK ve standart sapma değerlerine sahip olduğu Tablo 5.5'te görülmektedir. Ayrıca Body Fat veri seti için her iki yöntem ortalama OKHK ve standart sapma açısından aynı başarıyı elde edilmiştir. Bütün veri setleri için DET algoritması tabanlı AÖM yöntemi daha az nörona gereksinim duymaktadır. Şekil 5.1'de DET ve AÖM yöntemleri için gizli katman nöron sayısına bağlı olarak ortalama OKHK değişim eğrisi gösterilmiştir.

Tablo 5.5. YFT-SAÖM (DET) yöntemi ile AÖM'nin karşılaştırma sonuçları

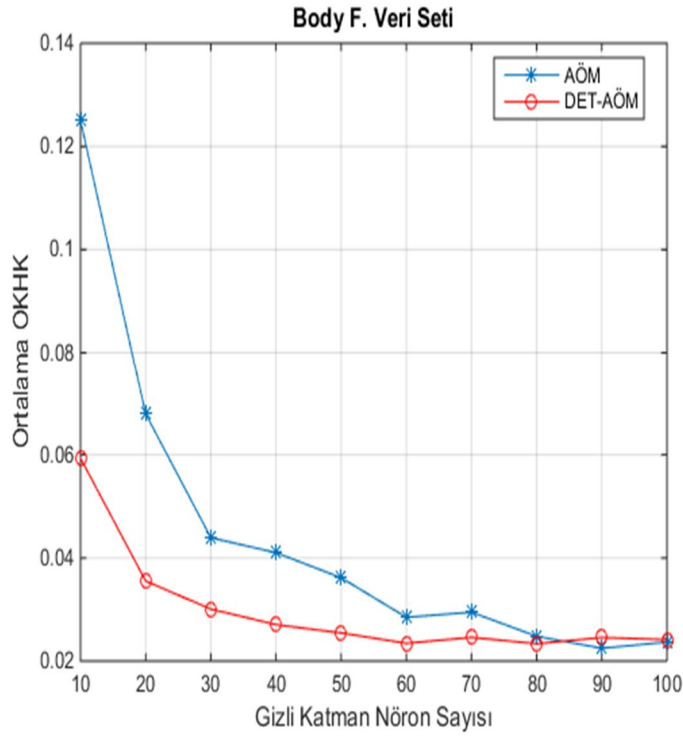
		Veri Seti								
		Abalone	Auto P	Body F	Boston H	Breast C	Bupa	Pima ID	Servo	Triazines
DET	OKHK	0.0743	0.0731	0.0230	0.0949	0.2663	0.2407	0.4039	0.1057	0.1845
	σ	6.2e-4	0.0081	0.0037	0.0061	0.0101	0.0102	0.0047	0.0086	0.0145
	Nöron	(37)100	(13)50	(37)50	(25)50	(6)50	(14)50	(7)50	(20)50	(10)50
	Süre (s)	17.37	0.367	1.938	1.093	0.206	0.432	0.324	0.568	0.281
AÖM	OKHK	0.0770	0.1055	0.0230	0.0903	0.3098	0.2884	0.4258	0.1486	0.2320
	σ	0.0028	0.0199	0.0037	0.0058	0.0264	0.0245	0.0116	0.0301	0.271
	Nöron	100	50	50	50	50	50	50	50	50
	Süre (s)	7.4809	0.2971	0.3172	0.4019	0.3091	0.3814	0.4942	0.2880	0.2999



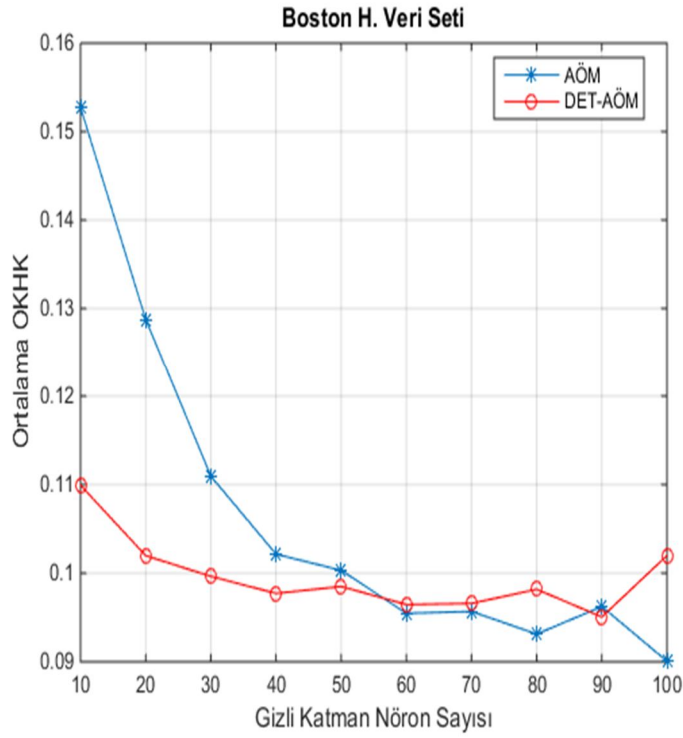
(a) Abalone veri seti



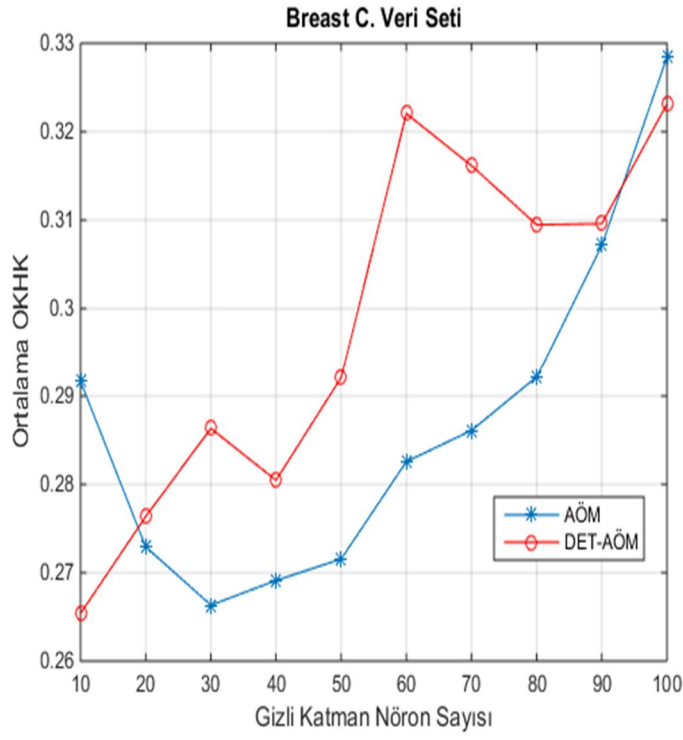
(b) Auto P. veri seti



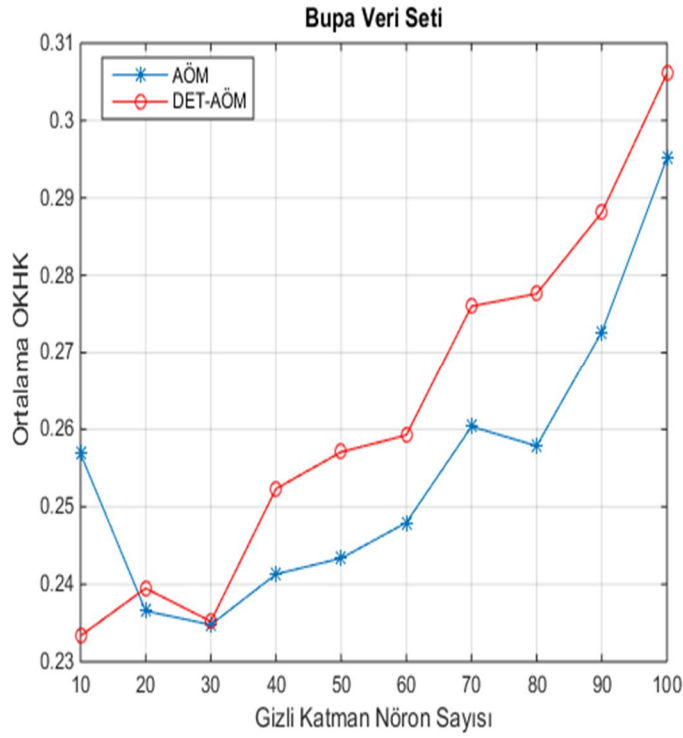
(c) Body Fat veri seti



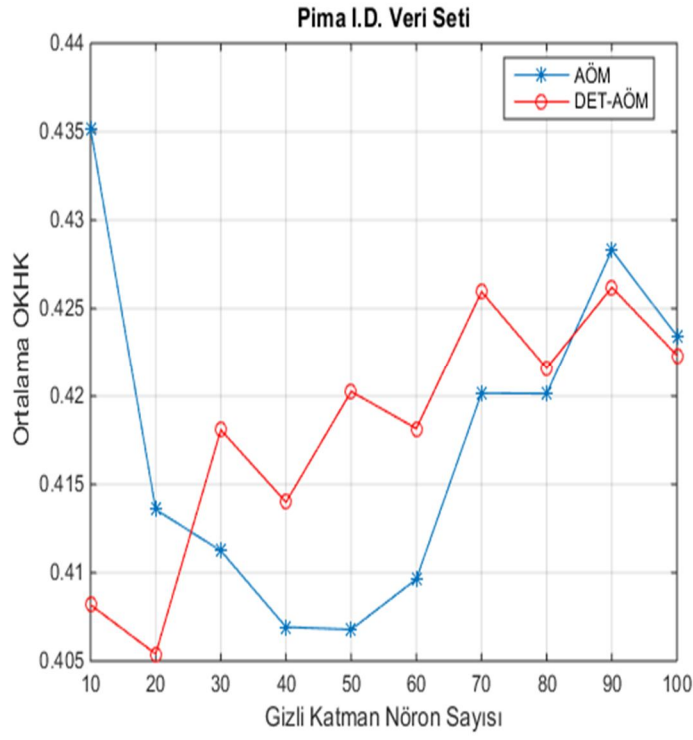
(d) Boston Housing veri seti



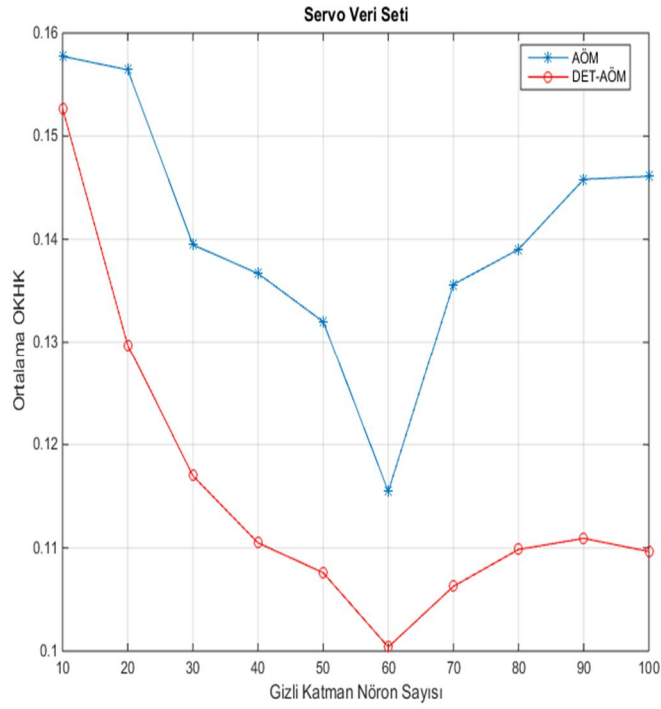
(e) Breast Cancer veri seti



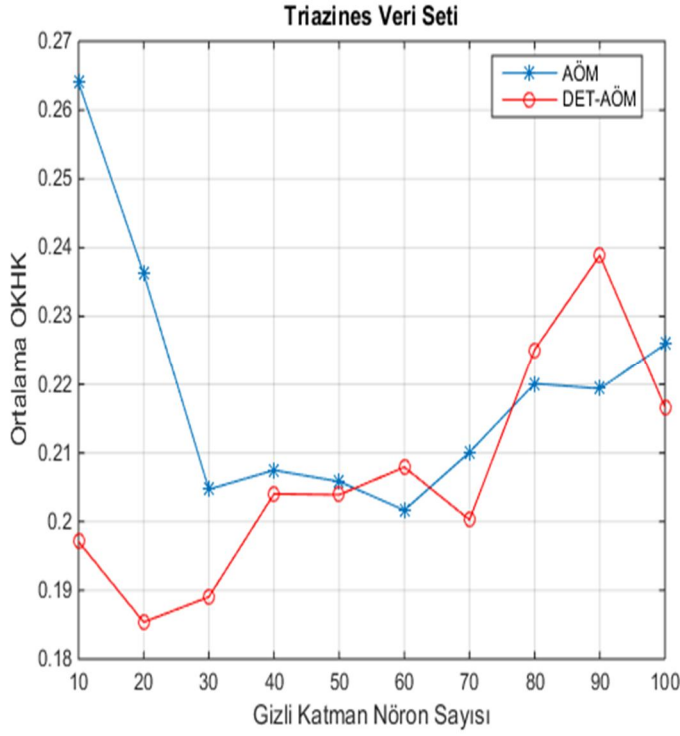
(f) Bupa veri seti



(g) Pima Indians Diabetes veri seti



(h) Servo veri seti



(i) Triazines veri seti

Şekil 5.1. Gizli katman nöron sayısı artışına bağlı DET ve AÖM'nin ortalama OKHK değişimi

Şekil 5.1 DET tabanlı AÖM yönteminin daha az nöronla en düşük ortalama OKHK değerine sahip olduğu görülmektedir. Özellikle Auto Price, Breast Cancer, Bupa, Pima Indians Diabetes, Servo ve Triazines veri setlerinde en düşük ortalama OKHK değerine yaklaşık 20 nöron altında ulaşılmıştır. Ayrıca Abalone veri seti için 37 nöron sayısı en düşük ortalama OKHK için yeterli olmaktadır. Öte yandan Şekil 5.1'den AÖM yönteminin Boston Housing veri seti için en düşük ortalama OKHK değerine sahip olduğu görülmektedir.

5.3.2. DET Algoritmasının EMDSO ve EN ile Karşılaştırılması

Bu kısımda yapılan deneylerde en iyi başarıma sahip olan DET algoritması EMDSO ve EN yöntemleri ile karşılaştırılması yapılmıştır. EN ve EMDSO yöntemi için deneysel çalışmalar [15] numaralı çalışmadan karşılaştırma yapmak amacıyla alınmıştır. Abalone, Auto Price, Boston Housing veri setlerine ek olarak Tablo 5.6'da verilen Ailerons, California Housing Delta E. veri setleri eklenerek altı veri seti ile deneyler ve karşılaştırma yapılmıştır. Bu ek veri setleri örnek sayısı fazla olduğundan seçilmiştir.

Tablo 5.6. Seçilmiş veri setleri hakkında bilgi

Veri Seti	Özellik Sayısı	Örnek Sayısı	
		Eğitim	Test
Aileron	40	9166	4584
California Housing	8	13760	6880
Delta elevators	6	6344	3173

DET ile EMDSO ve EN yöntemleri ile karşılaştırma sonuçları Tablo 5.7’de gösterilmiştir. Gizli katman nöron sayısı bakımından Aileron, Boston Housing veri setlerinde DET algoritması, Abalone, Delta E., California veri setlerinde EMDSO yöntemi ve son olarak Auto Price, veri setinde EN yönteminin en düşük değerlere sahip olduğu Tablo 5.7’de görülmektedir. Ortalama OKHK kriterleri açısından en iyi başarımlar DET algoritması tabanlı AÖM yöntemi üretmiştir. Özellikle Abalone, Auto Price ve Boston Housing veri setlerinde ortalama OKHK kriteri bakımından büyük iyileşme sağlanmıştır. DET algoritması ortalama OKHK kriteri açısından bakıldığında Aileron, Delta E. ve California Housing veri setlerinde ise EMDSO ve EN yöntemlerine göre daha kabul edilebilir başarımlara sahip olmuştur. Bu sonuçlara ek olarak standart sapma açısından DET algoritması karşılaştırma yapılan diğer yöntemlerden daha düşük değerlere sahip olduğu Tablo 5.7’den açıkça görülmektedir. Bir başka anlatımla önerilen DET algoritması tabanlı AÖM yönteminin diğer yöntemler ile karşılaştırıldığında oldukça kararlı bir yapıya sahip olduğu söylenebilir. Son olarak gizli katman nöron sayısı, başarımlar ve kararlı yapı kriterleri bir bütün olarak değerlendirildiğinde önerilen yöntemin belirgin bir üstünlüğe sahiptir.

Tablo 5.7. YFT-AÖM (DET) yöntemi ile EN ve EMDSO yöntemlerinin karşılaştırma sonuçları

Veri Seti		Abalone	Auto P	Aileron	Boston H	Delta E	California H
EN	OKHK	0.6518	1.039	0.4476	0.4587	0.6045	0.5114
	σ	0.004	0.019	0.005	0.033	0.002	0.004
	Nöron	(26.1)200	(1.1)50	(146.4)1000	(39.8)200	(25.4)200	(153.3)1000
EMDSO	OKHK	0.6499	1.0281	0.4471	0.4465	0.6057	0.5078
	σ	0.003	0.023	0.004	0.031	0.002	0.004
	Nöron	(21.6)200	(2.1)50	(142)1000	(41.4)200	(16.8)200	(149.1)1000
DET	OKHK	0.0743	0.0731	0.1833	0.0949	0.1084	0.1160
	σ	6.2e-4	0.0081	4.09e-4	0.0061	2.71e-4	6.88e-4
	Nöron	(37)100	(13)50	(72)200	(25)50	(33)200	(150)500

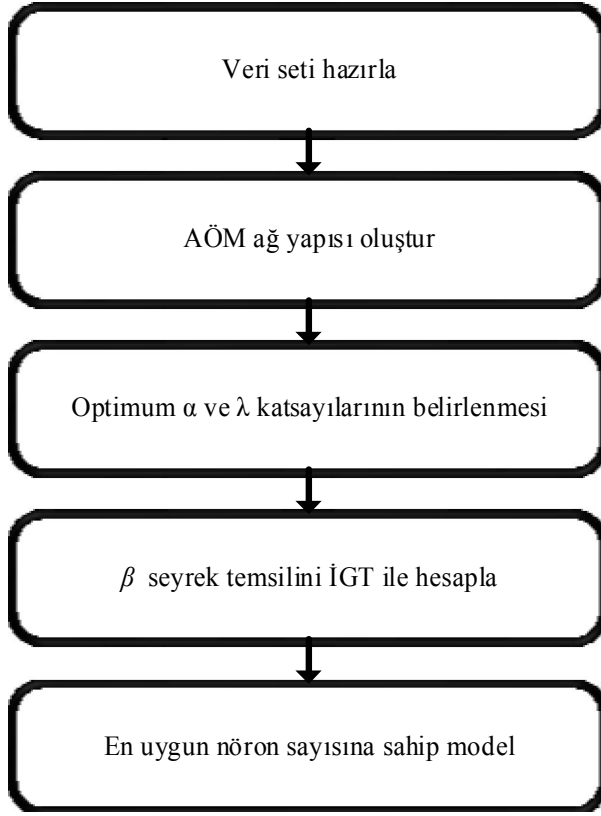
6. İLERİ-GERİ TAKİP ALGORİTMASI TABANLI SEYREK AŞIRI ÖĞRENME MAKİNESİ

6.1. Giriş

Bu bölümde AÖM'nin seyrek kabul edilen çıkış ağırlıklarının hesaplanması için İGT yöntemi önerilmiştir. Önerilen seyrek tabanlı AÖM yöntemi İGT-AÖM olarak isimlendirilmiştir. Bu yaklaşımda, İGT-AÖM yöntemi biyomedikal veri setlerinin sınıflandırılması problemlerine uygulanmıştır. Yukarıda bahsedildiği gibi İGT-AÖM'nin genel amacı çıkış katmanı ağırlıklarının seyrek yaklaşım ile hesaplanarak düşük işlem karmaşıklığına ve en uygun gizli katman nöron yapısına sahip ağ yapısının elde edilmesidir.

6.2. Önerilen İGT-AÖM Yöntemi

Bu bölümde, önerilen İGT-AÖM yönteminin temellerine değinilmiş kısaca anlatılmış ve deneysel çalışmalarla ilgili ayrıntılı bilgiler verilmiştir. İGT-AÖM yöntemi akış şeması Şekil 6.1'de gösterilmiştir. İGT-AÖM yönteminde, İGT algoritması durdurma kriteri olarak hata (ϵ) seçilmiştir. İGT algoritması için ileri adım sayısı α ve geri adım sayısı μ belirlenmesi önemlidir. Deneysel çalışmada en uygun α ve μ belirlenmesi için basit bir arama algoritması kullanılmıştır. Arama algoritmasında adım boyutlarının belirlenmesi [16] numaralı çalışmada gösterildiği gibi α , M 'ye bağlı ve μ katsayısı ise α katsayısına bağımlı olarak seçilmiştir. Kullanılan arama algoritmasında α adım boyutu $0.1*M$ ile $0.3*M$ aralığında birer artımla taranmış ve μ adım boyutu ise $\alpha-1$ yapılmıştır. Arama algoritmasının her adımında İGT algoritması 100 kere çalıştırılarak en yüksek doğruluğa sahip α ve μ katsayıları seçilmiştir. Seçilen uygun adım parametrelili İGT algoritması kullanılarak çıkış katmanı ağırlıklarının seyrek temsili hesaplanmıştır. Ardından oluşturulan İGT-AÖM yöntemi ile optimum model elde edilmiştir.



Şekil 6.1. Önerilen İGT-AÖM yöntemi akış şeması

6.3. Deneysel Çalışmalar

Bu bölümde, İGT-AÖM performansı UCI [81], Statlib [82] ve Kent Ridge Bio-medical Data Set Repository [87]'den alınan biyomedikal veri setleri kullanılarak test edilmiştir. Bu veri setlerinden yedi tanesi iki ve üç tanesi ise çok sınıflıdır. Tablo 6.1'de ilgili veri setlerinin özellikleri verilmiştir.

Deneysel çalışmada; doğruluk yüzdesi, standart sapma, kullanılan nöron sayısı ve deneysel çalışma süresi başarı ölçütleri olarak ele alınmıştır. Doğruluk yüzdesi, doğru tahmin edilen veri sayısının tüm veri sayısına oranının 100 katı ve Nöron Sayısı en yüksek doğruluk yüzdesine sahip nöron sayısı olarak tanımlanmıştır. Deneysel çalışmanın geçerliliğini artırmak için 10-kat çapraz geçерleme tekniği kullanılmıştır ve elde edilen sonuçların ortalaması hesaplanmıştır. Modellerde kullanılan en yüksek doğruluk yüzdesine sahip en uygun nöron sayısını belirlemek için arama algoritması kullanılmıştır. Arama algoritmasında nöron sayısı beşten başlatılarak eğitim veri setinde kullanılan örnek

sayısının onda bir oranında artış miktarı ile 1000 nörona kadar artırılarak en uygun nöron sayısı elde edilmiştir.

Tablo 6.1. Kullanılan veri setlerinin özellikleri

Veri Seti	Özellik	Sınıf	Örnekler	
	Sayısı	Sayısı	Eğitim	Test
Breast Cancer	10	2	615	68
Iris	4	3	135	15
Kidney	6	3	68	8
Liver	7	2	306	34
Lung Cancer	56	3	24	3
Lupus	3	2	78	9
Ovarian Cancer	15154	2	228	25
Parkinsons	23	2	175	20
Pima Indian Diabetes	8	2	691	77
Prostate Tumor	12600	2	91	10

Deneysel çalışmanın geçerliliğini sağlamak amacıyla her bir veri seti için benzetim 100 defa tekrarlanmış ve performans analizi için ortalama doğruluk yüzdesi, standart sapma, kullanılan nöron sayısı ve benzetim süresi hesaplanmıştır. Veri setlerinin hepsi SOBV'ye sahip olacak şekilde normalize edilmiştir [27]. Bütün AÖM'ler için gizli katman aktivasyon fonksiyonu sigmoid fonksiyonu, çıkış katmanı aktivasyon fonksiyonu ise doğrusal fonksiyon olarak seçilmiştir. Deneyler 2.2 GHz işlemci ve 8 GB belleğe sahip bir bilgisayarda gerçekleştirilmiştir. İGT-AÖM sınıflandırıcı sonuçları Tablo 6.2'de verilmiştir.

Tablo 6.2. İGT-AÖM yöntemi sonuçları

Veri seti	İGT-AÖM			
	Doğr. (%)	σ	Nöron Sayısı	Süre (s)
Breast C.	92,35	3,38	10	3,36
Iris	98	3,22	5	1,62
Kidney	44,28	22,78	49	1,02
Liver	70,58	9,3	10	1,81
Lung C.	95	14,81	17	1,91
Lupus	53,75	13,24	5	0,84
Ovarian C.	97,2	3,78	100	11,25
Parkinsons	90,00	6,77	13	2,45
Pima I. D.	74,736	7,22	10	3,96
Prostate T.	89,00	5,67	89	14,18

Tablo 6.2’de ortalama doğruluk yüzdesi Doğr. ile ve doğruluğun standart sapma ise σ ile gösterilmiştir. Tablo 6.2’den görülebileceği üzere İGT-AÖM en iyi başarımlar değerini Iris veri seti için üretmiştir. Bunun yanı sıra Breast Cancer, Ovarian Cancer, Lung C. ve Parkinsons veri setleri için kabul edilebilir sonuçlar elde edilmiştir. Ayrıca en düşük standart sapmaya Iris veri setinde ulaşılmıştır. Kidney, Lung C. ve Lupus veri setleri için standart sapma değerleri en yüksek sonuçlar üretmiştir. Doğruluk ve standart sapma açısından en kötü sonuçlar ise Kidney veri seti için elde edilmiştir. Deneysel çalışmalarda, İGT-AÖM başarımlar sonuçları analiz edildikten sonra yöntemin geçerliliğini göstermek için AÖM ve EAR-AÖM, EMDSO-AÖM, EN-AÖM ile İGT-AÖM karşılaştırması yapılmıştır.

6.3.1. İGT-AÖM ile AÖM Karşılaştırılması

AÖM yönteminin başarımlarını da yine Tablo 6.1’deki veri setleri kullanılarak değerlendirilmiştir. Normal AÖM yöntemi ile İGT-AÖM yöntemi karşılaştırma sonuçları Tablo 6.3’te verilmiştir. Tablo 6.3’ten görüldüğü gibi genel olarak İGT-AÖM yöntemi AÖM yönteminden daha iyi sonuçlar üretmiştir. Özellikle nöron sayısı bakımından İGT-AÖM yöntemi bütün veri setlerinde AÖM yöntemine üstünlük sağlamıştır. Breast Cancer, Iris ve Pima I.D. veri setlerinde İGT-AÖM yöntemi tüm kriterlerde en iyi sonuçlara sahiptir. İGT-AÖM yöntemi; Liver, Lupus ve Pima I.D. veri setleri için normal AÖM yönteminden yaklaşık iki kat daha yüksek doğruluk değerine sahiptir.

Tablo 6.3. İGT-AÖM ile AÖM karşılaştırması

Veri seti	AÖM				İGT-AÖM			
	Doğr. (%)	σ	Nöron	Süre (s)	Doğr. (%)	σ	Nöron	Süre (s)
Breast C.	74,11	5,46	917	8,01	92,35	3,38	10	3,36
Iris	72,66	9,13	785	9,36	98	3,22	5	1,62
Kidney	45,71	14,75	801	10,42	44,28	22,78	49	1,02
Liver	20,29	7,77	966	10,48	70,58	9,3	10	1,81
Lung C.	95	15,81	288	0,44	95	14,81	17	1,91
Lupus	17,5	16,87	72	1,75	53,75	13,24	5	0,84
Ovarian C.	98,4	2,79	900	8,54	97,2	3,78	100	11,25
Parkinsons	86,31	5,08	513	1,02	90,00	6,77	13	2,45
Pima I. D.	31,18	8,77	956	8,10	74,736	7,22	10	3,96
Prostate T.	93	8,23	990	13,64	89,00	5,67	89	14,18

Ovarian Cancer ve Kidney veri setlerinde AÖM yöntemi İGT-AÖM yönteminden çok az bir fark ile daha iyi doğruluk yüzdesine ve standart sapmaya sahiptir. Lung C. Veri setinde her iki yöntem aynı doğruluk yüzdesi üretirken İGT-AÖM yöntemi yaklaşık olarak 17 kat daha az nöron sayısı ile bu sonucu elde etmiştir. Süre açısından sonuçlar incelendiğinde Lung C., Ovarian Cancer, Parkinsons veri setleri dışında bütün veri setlerinde İGT-AÖM yöntemi iyi sonuçlar üretebilmiştir.

6.3.2. İGT-AÖM Yönteminin EAR, EMDSO ve EN ile Karşılaştırılması

İGT-AÖM'nin başarımı ayrıca literatürde önerilen diğer EMDSO, EAR, EN gibi düzenlenmiş AÖM yöntemleri ile karşılaştırılmıştır. Elde edilen sonuçlar Tablo 6.4'te gösterilmiştir. Tablo 6.4'ten görüldüğü gibi İGT- AÖM yöntemi Lupus ve Pima Indians Diabetes veri setleri dışındaki tüm veri setlerinde doğruluk açısından en iyi başarıma sahiptir. Ayrıca, Iris veri setinde İGT-AÖM yöntemi süre haricindeki bütün ölçütlerde en iyi sonuçları üretmiştir. İGT-AÖM yöntemi Kidney veri seti için diğer yöntemlerden yaklaşık bir buçuk kat daha iyi doğruluk değerine sahiptir. Yine Tablo 6.4'ten görüldüğü gibi EN, Lupus ve Prostate T. veri setlerinde doğruluk ölçütünde kabul edilebilir sonuçlara sahiptir. Genel itibariyle EMDSO deneysel çalışma süresi bakımından İGT-AÖM

yönteminden daha iyi sonuç vermiştir. Bütün veri setleri için İGT-AÖM yöntemi en uygun nöron sayısına sahip yöntemdir. Tablo 6.4'ten çıkarılacak genel bir yorumla İGT-AÖM yöntemi; doğruluk başarımı, standart sapma ve nöron sayısı bakımından EAR, EMDSO ve EN yöntemlerinden daha iyi başarıma sahiptir.

Tablo 6.4. İGT-AÖM ile EAR, EMDSO ve EN karşılaştırma sonuçları

Veri Seti	İGT-AÖM				EMDSO				EAR				EN			
	Doğr. (%)	σ	Nöron	Süre (s)	Doğr. (%)	σ	Nöron	Süre (s)	Doğr. (%)	σ	Nöron	Süre (s)	Doğr. (%)	σ	Nöron	Süre (s)
Breast C.	92,35	3,38	10	3,36	82,94	6,83	673	1,57	88,82	3,33	795	1,39	90,29	3,11	917	59,36
Iris	98	3,22	5	1,62	96,66	4,71	405	0,44	92	6,88	405	1,28	96,66	3,51	270	3,97
Kidney	44,28	22,78	49	1,02	30	15,72	819	0,66	27,14	10,54	315	1,70	28,57	16,49	378	5,01
Liver	70,58	9,3	10	1,81	63,52	8,56	306	0,38	39,11	7,84	306	3,39	67,64	6,79	306	8,39
Lung C.	95	14,81	17	1,91	60	21,08	108	1,9	75	26,35	288	1,99	80	25,81	198	9,13
Lupus	53,75	13,24	5	0,84	78,75	14,49	864	0,58	56,25	13,51	576	1,14	63,75	17,12	216	4,52
Ovarian C.	97,2	3,78	100	11,25	90,4	5,39	675	7,89	86,8	9,62	675	10,41	93,2	3,79	900	14,44
Parkinsons	90	6,77	13	2,45	87,36	7,92	171	0,41	55,26	12,95	684	2,11	83,15	7,76	513	4,69
Pima I. D.	74,736	7,22	10	3,96	76.31	6.73	684	1.19	70,78	6,87	888	15,51	65	6,74	684	48,18
Prostate T.	89	5,67	89	14,18	76	8.43	540	13.57	72	19,88	810	15,06	78	11,07	450	17,07

7. ZAMAN SERİSİ TAHMİNİ İÇİN YENİ BİR YAKLAŞIM: DEK-AÖM

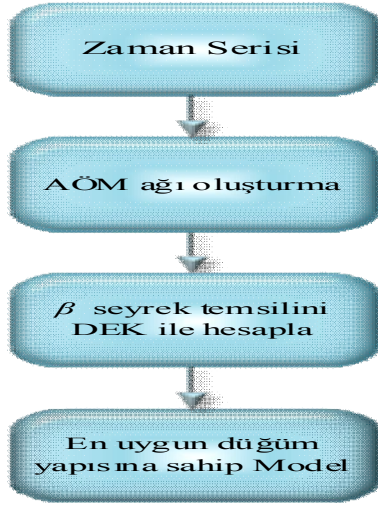
7.1. Giriş

Zaman serisi tahmin problemi dinamik sistemlerin modellenmesi tekniklerinin gelişiminde önemli bir yere sahiptir [88-90]. Ayrıca doğrusal olmayan zaman serisi tahmin problemi doğrusal olmayan sinyal işleme gibi pratik alanlarda ilgi duyulan bir alan olmuştur [88-90]. Zaman serilerini tahmin etmek için DVM, YSA ve makina öğrenmesi yöntemleri kullanılmaktadır [88, 91, 92]. Literatürde yapılan çalışmalarda durağan zaman serileri, durağan olmayan zaman serilerine oranla daha fazla çalışılan bir konu olmuştur [88]. Böylece durağan olmayan zaman serileri tahmin uygulamalarına ilgi artmaktadır [88, 93, 94]. Merkezi limit gömülü teoremine göre [88, 95], bir zaman serisi faz uzayına ve zaman korelasyondan uzamsal korelasyona ayrışabilir. Evrensel yakınsama başarımı ile YSA uzamsal korelasyonu verimli bir şekilde yakınsayabilir [88]. Geleneksel yapay sinir ağlarında yaygın bir şekilde kullanılmasına rağmen Eğitim tabanlı öğrenme algoritmaları yerel minimum noktasına kolay bir şekilde takılma ve yavaş yakınsama özelliğine sahiptir [88, 96, 97]. YSA'nın eğitim sürecini hızlandırmak için pek çok öğrenme yöntemi geliştirilmiştir veya iyileştirilmiştir [88]. Bu yöntemler arasında AÖM'ler çok verimli bir yaklaşımdır.

Tezin bu bölümünde; zaman serilerinin tahmini problemi için AÖM'nin en uygun nöron sayısı ile yüksek başarıma sahip seyreklik tabanlı bir AÖM yöntemi önerilmiştir. Bu yaklaşımda AÖM'nin çıkış ağırlıklarının seyrek olduğu kabul edilerek, çıkış ağırlıklarının seyrek temsili Dikgen Eşleme Takip algoritması hesaplanmıştır. DEK algoritması tabanlı önerilen yöntem DEK-AÖM olarak isimlendirilmiştir. Önerilen DEK-AÖM yöntemi ile AÖM yöntemi karşılaştırılmıştır.

7.2. Önerilen DEK-AÖM Yöntemi

Bu kısımda, önerilen DEK-AÖM yöntemi hakkında ve deneysel çalışmalarla ilgili bilgi verilmiştir. Önerilen DEK-AÖM yöntemi Şekil 7.1'de özetlenmiştir.



Şekil 7.1. Önerilen DEK-AÖM yöntemi

DEK-AÖM yönteminde, DEK algoritması için durdurma kriteri olarak seyreklik seçilmiştir. Başlangıçta seyreklik seviyesi 100 olarak düşünülmüş ve deneyler süresince yöntem iyi sonuç vermemiş ise artırılmıştır. AÖM'nin çıkış ağırlıkları DEK algoritması ile elde edilmiştir. Böylelikle daha az nöron sayısına sahip model elde edilmiştir.

7.3. Deneysel Sonuçlar

Bu bölümde önerilen yöntemin verimliliğini göstermek için üç farklı tip zaman serisi kullanılmıştır. İlk grupta DJIA modern finansal zaman serisi ve Ocean zaman serisi, ikinci grupta Mackey-Glass kaotik zaman serisi son olarak ise 2-B DDO zaman serisi kullanılmıştır. Deneysel çalışmalarda geçerliliğini artırmak için 10-kat çapraz geçişleme uygulanmıştır. Önerilen yöntemin verimliliğini göstermek amacıyla AÖM ile karşılaştırılması yapılmıştır. Tahmin başarımı ortalama OKHK, OMYH, OMH ve OKHK değerinin Standart Sapma (σ) kullanılarak ölçülmüştür. AÖM ağında ve DEK-AÖM ağında *sigmoid* (*sig*), *sinusoidal* (*sin*), *hardlim*, *triangular basis* (*tribas*) ve *radyal basis* (*radbas*) gibi geleneksel aktivasyon fonksiyonlarına ek olarak *elliott* aktivasyon fonksiyonu kullanılmıştır. *Elliot* aktivasyon fonksiyon bağıntısı (7.1)'de verilmiştir [98].

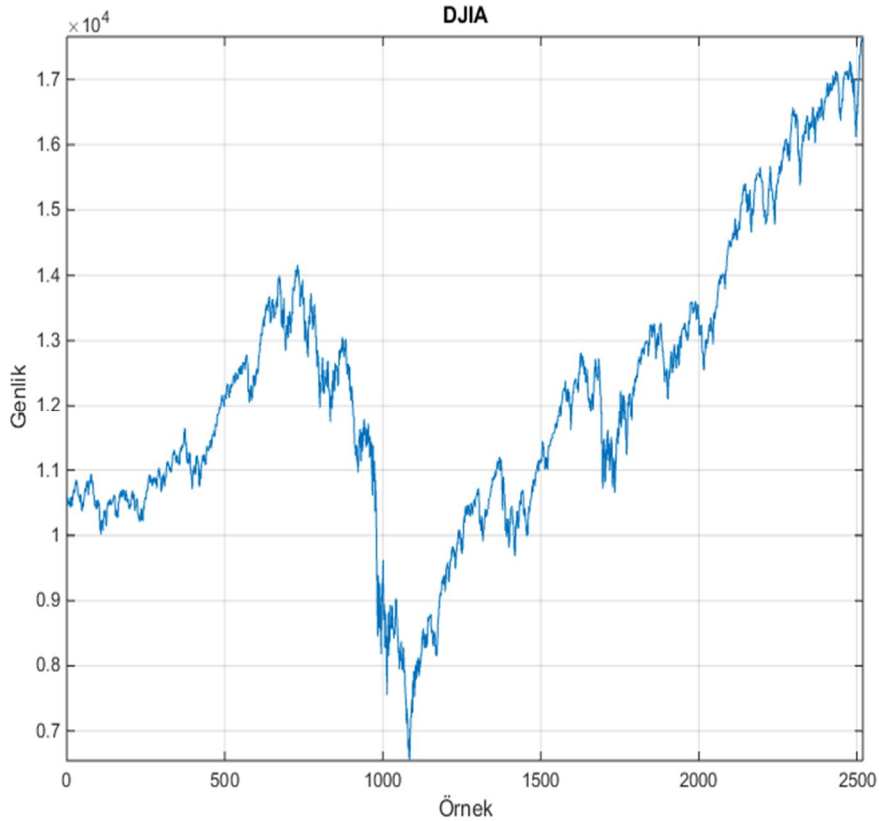
$$y = \frac{x * s}{1 + |x * s|} \quad (7.1)$$

Burada s diklik parametresi, x giriş ve y ise aktivasyon fonksiyonu çıkışını göstermektedir [98]. Deneylerde AÖM gizli katmanda 1000 nöron kullanılarak

gerçekleştirilmiştir. DEK-AÖM yönteminde ise 100 nöron kullanılmıştır. Daha sonra AÖM ile DEK-AÖM karşılaştırması adımıyla önerilen yöntemin güvenilirliği artırmak için deney 100 nöronlu AÖM için de yapılmıştır. Deneysel çalışma sonuçları her bir veri seti için ele alınarak irdelenmiştir.

7.3.1. DJIA Zaman Serisi

Önerilen yöntem verimliliğini göstermek için Modern finansal zaman serisine uygulanmıştır. Deneyde kullanılan modern finansal zaman serisi 15 Kasım 2004 den 14 Kasım 2014 kadar DJIA'nın borsa endeksinden oluşmaktadır [100]. Deneyde kullanılan DJIA finansal zaman serisi Şekil 7.2'de gösterilmiştir.



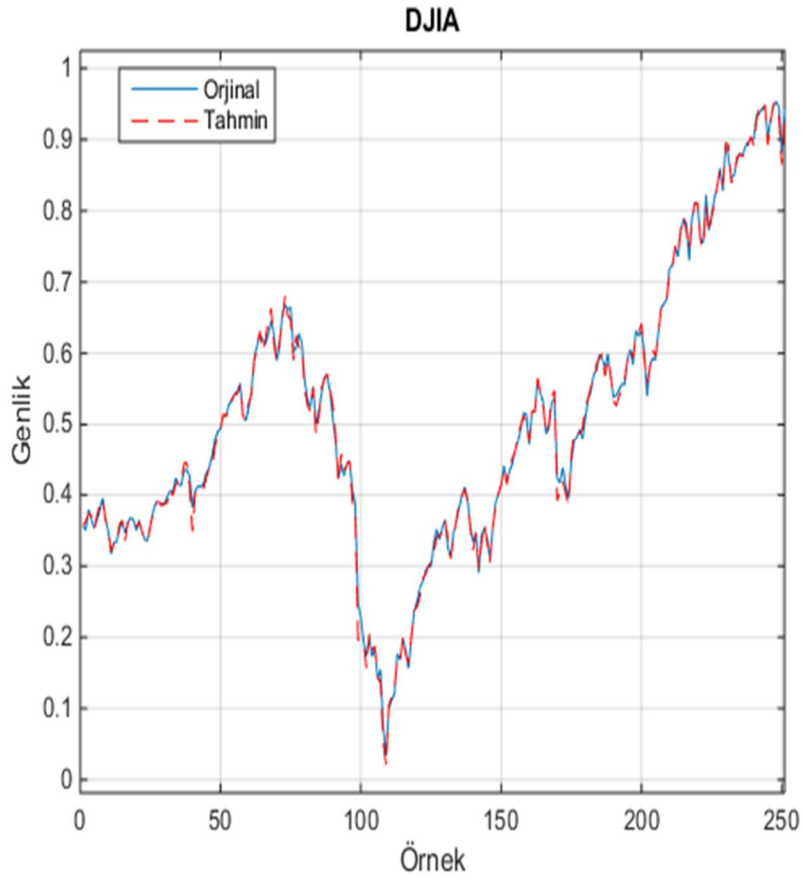
Şekil 7.2. DJIA finansal zaman serisi.

DJIA zaman serisi AÖM yöntemine uygulanmış ve elde edilen sonuçlar Tablo 7.1'de gösterilmiştir. DJIA veri seti için en iyi OKHK değeri *tribas* aktivasyon fonksiyonunda elde edilmiştir. Bu değere paralel olarak en düşük OMYH ve OMH değerleri de *tribas* aktivasyon fonksiyonu ile elde edilmiştir. Süre açısından bakıldığında *radbas* aktivasyon fonksiyonlu AÖM öne çıkmaktadır. DJIA zaman serisi *tribas*

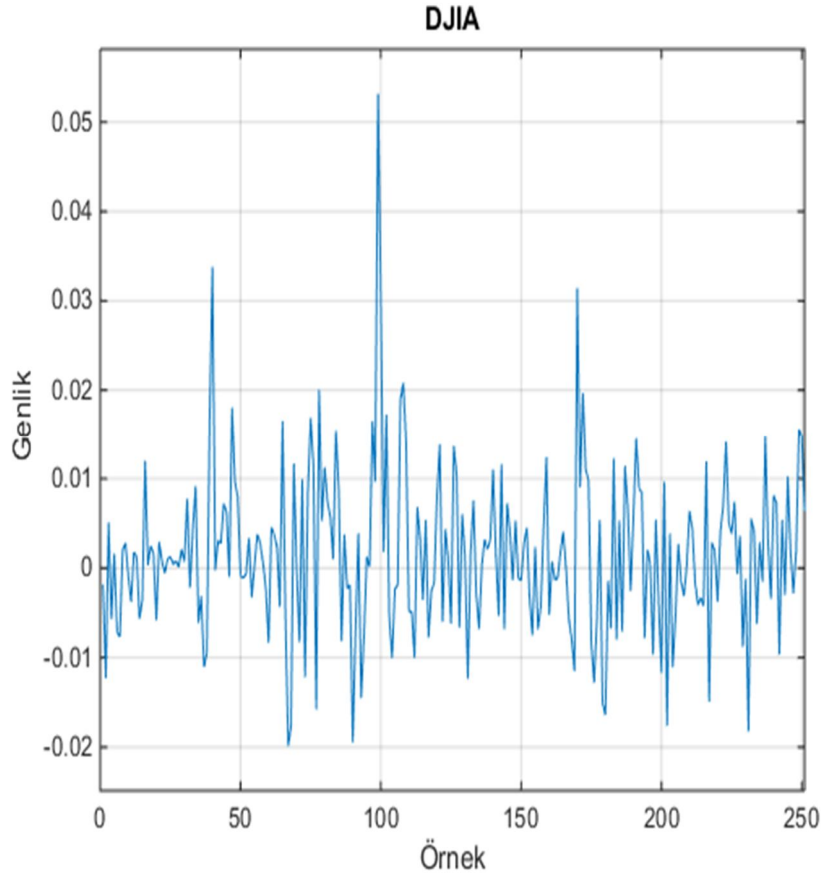
aktivasyon fonksiyonlu AÖM test tahmin eğrisi Şekil 7.3'te, AÖM tahmin hatası Şekil 7.4'de gösterilmiştir.

Tablo 7.1. AÖM DJIA zaman serisi sonuçları

f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
sig	0.04259	12.9436	0.03259	9.33e-4	0.434	1000
sin	0.04862	15.9815	0.03710	12e-4	0.414	1000
hardlim	0.01280	2.9680	0.00930	6.6e-4	0.668	1000
tribas	0.01184	2.4203	0.00797	20.9e-4	0.528	1000
radbas	0.04028	11.5454	0.03104	9.2e-4	0.404	1000
Elliot	0.01431	2.3557	0.00828	41.8e-4	0.504	1000



Şekil 7.3. DJIA zaman serisi AÖM tahmin eğrisi



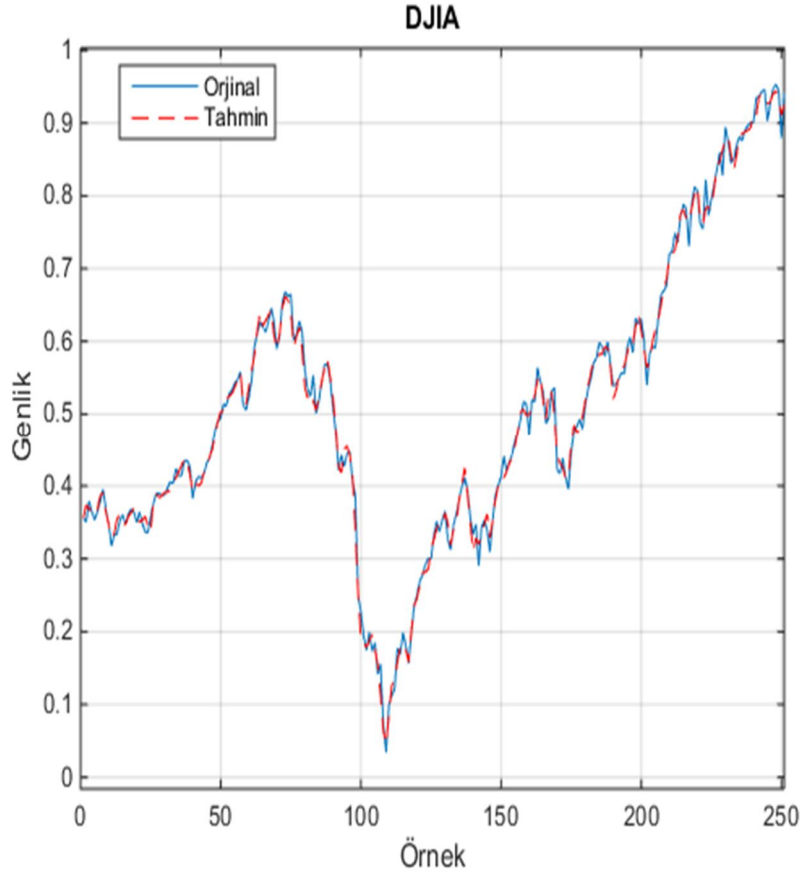
Şekil 7.4. DJIA zaman serisi AÖM tahmin hatası eğrisi

Şekil 7.3 ve Şekil 7.4'ten görüleceği gibi, AÖM tahmin eğrisi gerçek eğriye gayet yakındır. DJIA zaman serisi DEK-AÖM sonuçları Tablo 7.2'de gösterilmiştir. Tablo 7.2'den görüleceği gibi, AÖM'de olduğu gibi DEK-AÖM'de de *tribas* aktivasyon fonksiyonu ile OKHK, OMYH ve OMH değerleri elde edilmiştir.

Tablo 7.2. DEK-AÖM DJIA zaman serisi sonuçları

f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
sig	0.04253	12.8940	0.03253	9.21e-4	0.513	(100)1000
sin	0.04861	15.9790	0.03709	12.04e-4	0.521	(100)1000
hardlim	0.01615	3.8237	0.01221	7.04e-4	0.607	(100)1000
tribas	0.01494	3.4813	0.01107	6.95e-4	0.606	(100)1000
radbas	0.04019	11.5190	0.03100	9.18e-4	0.510	(100)1000
elliott	0.01510	3.5311	0.01126	7.51e-4	0.602	(100)1000

Şekil 7.5’de DJIA zaman serisi için DEK-AÖM yöntemi test tahmin eğrisi ve Şekil 7.6’da tahmin hatası eğrisi verilmiştir. Şekil 7.5 ve Şekil 7.6’dan görüleceği gibi, DEK-AÖM tahmin eğrisi gerçek eğriye yakın olduğu görülmektedir.

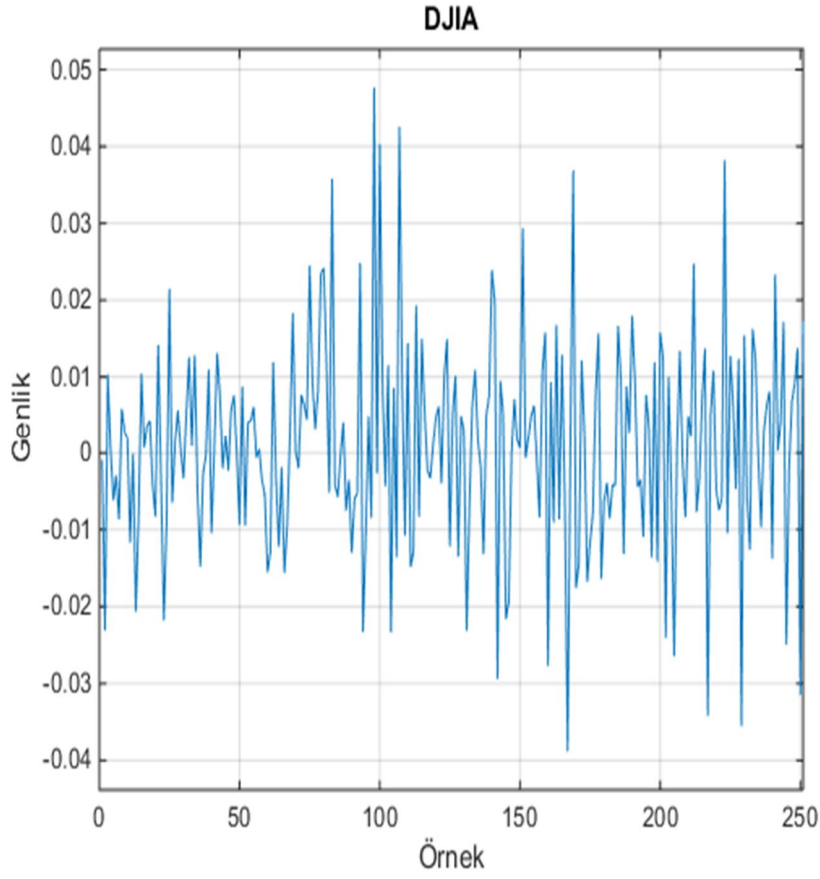


Şekil 7.5. DJIA zaman serisi DEK-AÖM tahmin eğrisi

Tablo 7.3’te DJIA zaman serisi için AÖM ve DEK-AÖM yöntemleri karşılaştırmalı sonuçları verilmiştir.

Tablo 7.3. DJIA zaman serisi için AÖM ve DEK-AÖM karşılaştırması

	f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
AÖM	tribas	0.01184	2.4203	0.00797	20.9e-4	0.528	1000
AÖM	tribas	0.01968	4.8521	0.01472	9.37e-4	0.012	100
DEK-AÖM	tribas	0.01494	3.4813	0.01107	6.95e-4	0.606	100



Şekil 7.6. DJIA zaman serisi DEK-AÖM tahmin hatası eğrisi

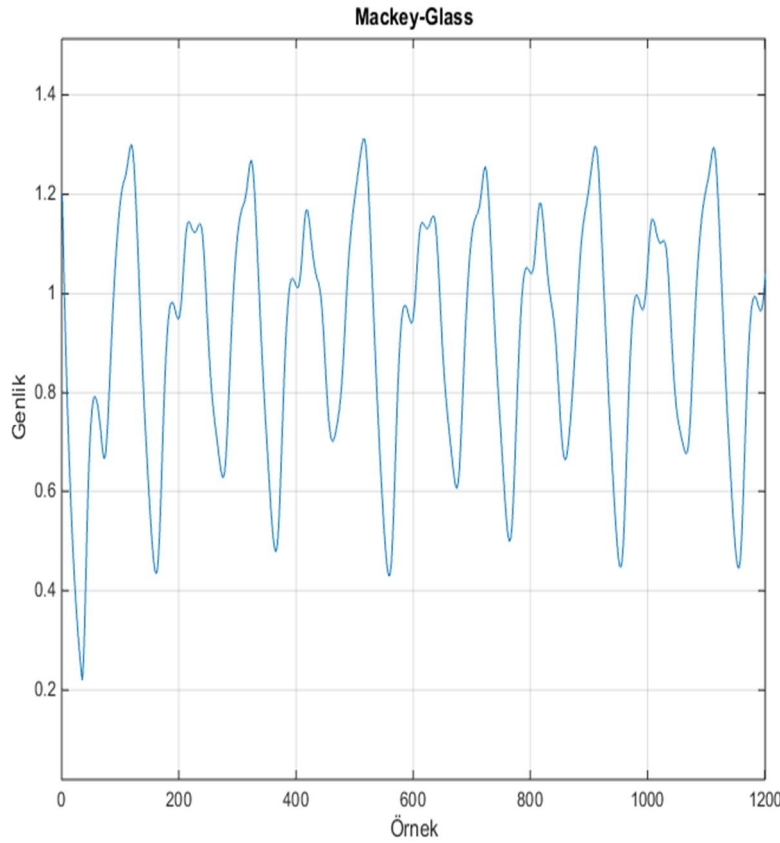
Tablo 7.3 incelendiğinde, 1000 nöronlu AÖM yöntemi OKHK, OMYH, OMH ve süre açısından DEK-AÖM yöntemi sonuçlarına göre daha iyi başarıma sahiptir. Diğer yandan önerilen DEK-AÖM yöntemi 100 nöron ile OKHK açısından AÖM yakın başarıma sahip ve daha kararlı bir sonuç üretmiştir. Süre açısından 1000 nöronlu AÖM yöntemi yaklaşık 0.0771 saniye farkla daha kısa sürede sonuç üretmiştir. Ayrıca 100 nöronlu AÖM dikkate alındığında süre bazında 1000 nöronlu AÖM ve DEK-AÖM yöntemlerine karşı üstün başarı sağlarken diğer kriterlerde 1000 nöronlu AÖM ve DEK-AÖM yöntemlerinden geri kalmaktadır. Tablo 7.3'ten görüleceği üzere, DJIA zaman serisi için önerilen DEK-AÖM yöntemi ile en uygun model elde edilebilmiştir.

7.3.2. Mackey-Glass Zaman Serisi

Önerilen yöntemin performansını test etmek için kullanılan ikinci zaman serisi ise durağan olmayan Mackey-Glass kaotik zaman serisi (7.2)'den elde edilmiştir [101].

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x(t-\tau)^{10}} - 0.1x(t) \quad (7.2)$$

Burada $x(t)$ t anında ki zaman serisi değeri olup $\tau = 17$, $x(0) = 1.2$ ve $t \leq 0$ için $x(t) = 0$ şartları seçildiğinde periyodik ve yakınsak olmayan kaotik zaman serisi elde edilir. Elde edilen Mackey Glass zaman serisi Şekil 7.7'de gösterilmiştir.



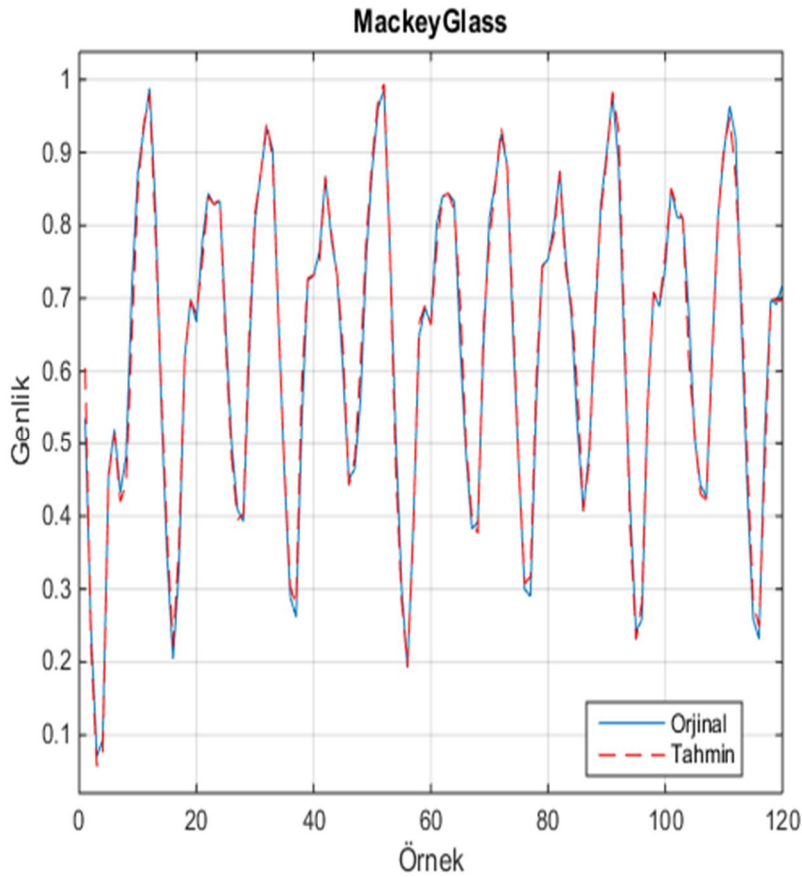
Şekil 7.7. Mackey-Glass zaman serisi

Mackey-Glass zaman serisi için elde edilen AÖM sonuçları Tablo 7.4'te verilmiştir. Mackey-Glass zaman serisinde *hardlim* aktivasyon fonksiyonlu AÖM yöntemi en iyi OKHK ve OMH değerlerine sahipken, OMYH kriterinde ikinci en iyi değere sahiptir. *Radbas* aktivasyon fonksiyonlu AÖM σ ve süre açısından iyi sonuç üretmiştir. Şekil 7.8'de Mackey-Glass zaman serisi için *hardlim* aktivasyon fonksiyonlu AÖM yöntemi test tahmin

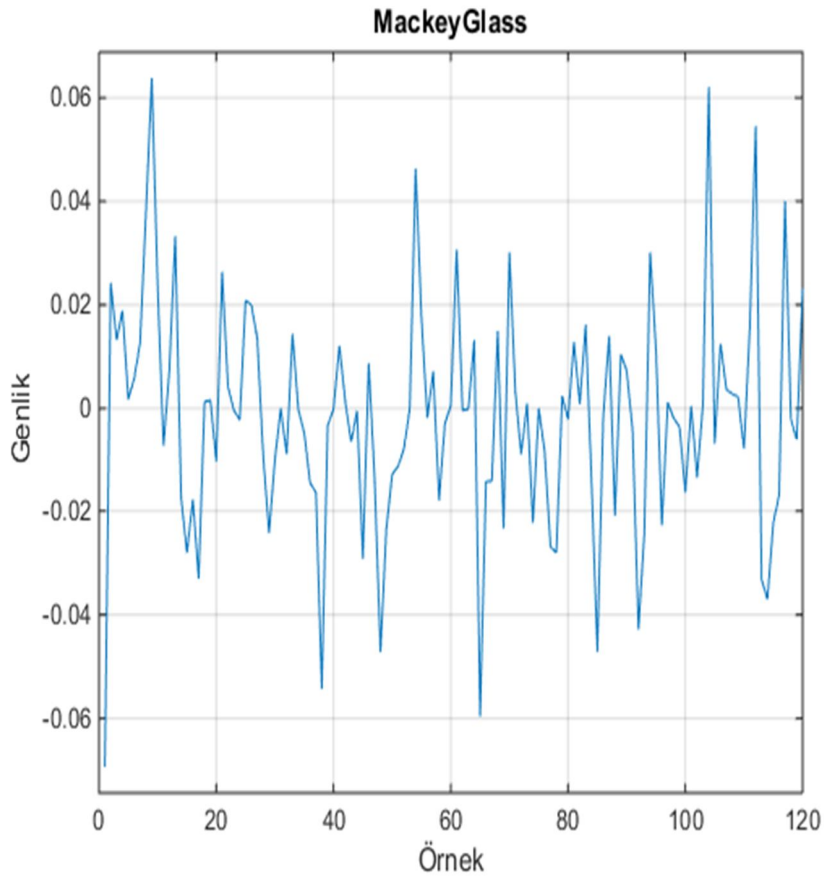
eğrisi ve Şekil 7.9’da AÖM yöntemi test tahmin hatası verilmiştir. Şekil 7.8 ve Şekil 7.9’dan görüleceği üzere AÖM yöntemi tahmin eğrisi gerçek eğriye oldukça yakındır.

Tablo 7.4. Mackey-Glass zaman serisi AÖM sonuçları

f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
sig	0.19477	37.8788	0.16829	2.5e-4	0.288	1000
sin	0.20103	41.9520	0.17088	4.7e-4	0.264	1000
hardlim	0.02714	4.8007	0.01714	35.4e-4	0.310	1000
tribas	0.12606	4.0625	0.02392	0.1781	0.366	1000
radbas	0.19199	36.8344	0.16345	1.5e-4	0.262	1000
elliott	0.15670	5.0468	0.02933	0.1703	0.504	1000



Şekil 7.8. Mackey-Glass zaman serisi AÖM tahmin eğrisi



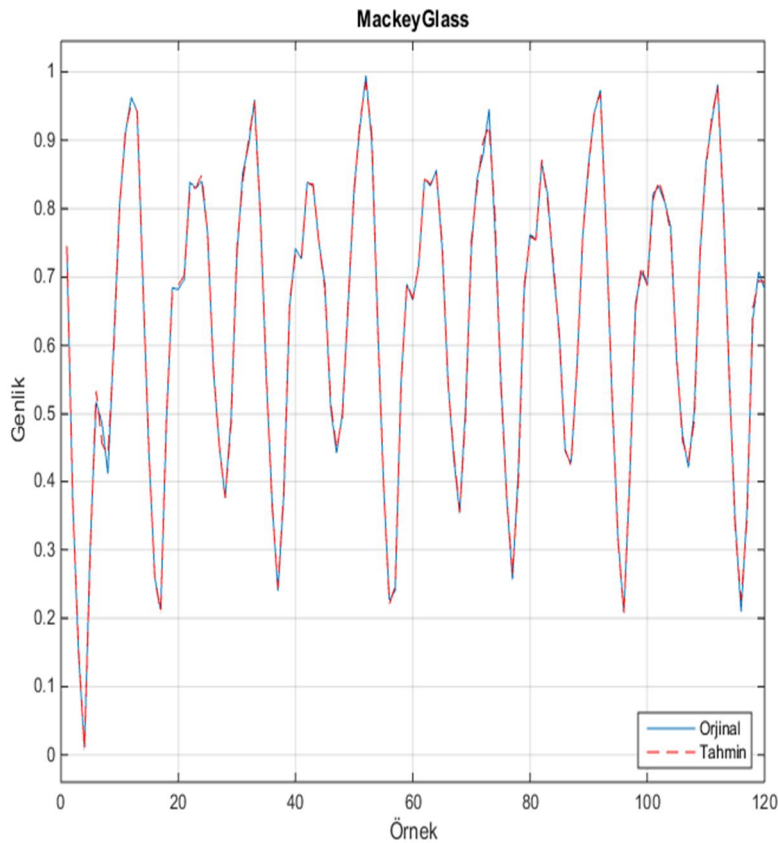
Şekil 7.9. Mackey-Glass zaman serisi AÖM tahmin hatası

Tablo 7.5’te Mackey-Glass zaman serisi DEK-AÖM yöntemi sonuçları verilmiştir. Sonuçlar incelendiğinde *elliott* aktivasyon fonksiyonlu DEK-AÖM yöntemi 0.00942 OKHK ile diğer aktivasyon fonksiyonları içinde en iyi sonucu üretmiştir. Ayrıca *elliott* aktivasyon fonksiyonlu DEK-AÖM OMYH ve OMH kriterlerinde en iyi sonuçlara sahiptir. Standart sapma açısından *sigmoid* aktivasyon fonksiyonlu DEK-AÖM yöntemi iyi sonuç vermiştir.

Tablo 7.5. Mackey-Glass zaman serisi DEK-AÖM sonuçları

f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
sig	0.01946	37.8903	0.16837	2.92e-4	0.253	(100)1000
sin	0.20704	45.7014	0.17355	21.12e-4	0.248	(100)1000
hardlim	0.04701	7.9159	0.03507	51.52e-4	0.296	(100)1000
tribas	0.01253	3.0112	0.00974	16.24e-4	0.321	(100)1000
radbas	0.18713	33.9030	0.15936	5.81e-4	0.249	(100)1000
elliott	0.00942	1.7939	0.00713	13.09e-4	0.294	(100)1000

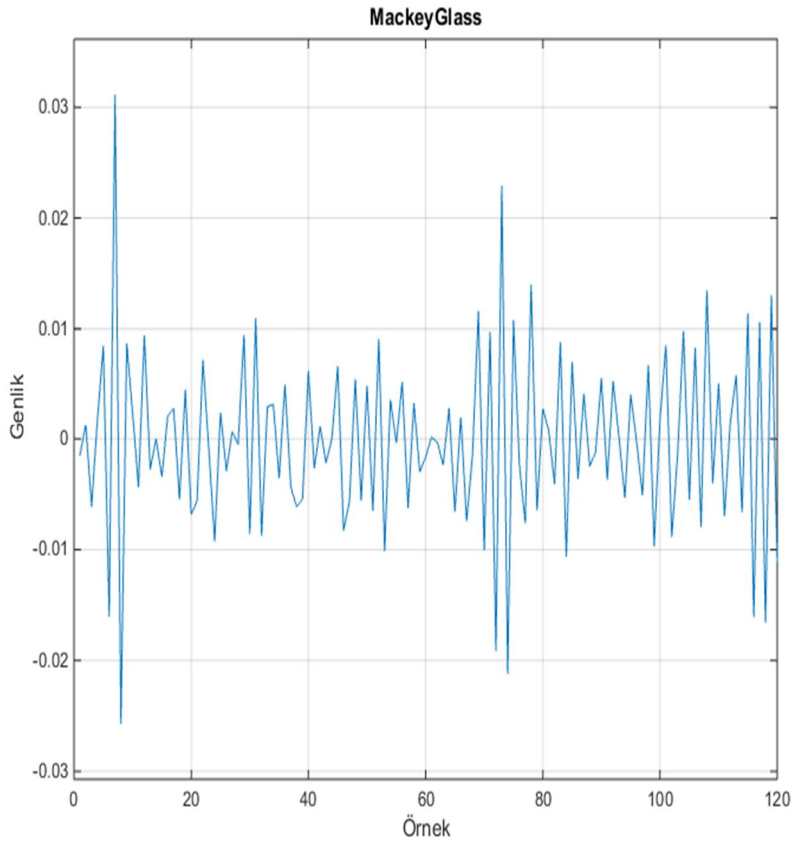
Mackey-Glass zaman serisi *elliott* aktivasyon fonksiyonlu DEK-AÖM test tahmin eğrisi Şekil 7.10'da ve DEK-AÖM test tahmin hatası Şekil 7.11'de gösterilmiştir. Şekil 7.10 ve Şekil 7.11'den de görüleceği gibi, DEK-AÖM yöntemi tahmin sonuçları gerçek değerlere yakındır.

**Şekil 7.10.** Mackey-Glass zaman serisi DEK-AÖM tahmin eğrisi

Tablo 7.6. Mackey-Glass zaman serisi için AÖM ve DEK-AÖM karşılaştırması

	f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
AÖM	hardlim	0.02714	4.8007	0.01714	35.40e-4	0.310	1000
AÖM	tribas	0.04557	8.9957	0.03367	61.04e-4	0.005	100
DEK-AÖM	elliott	0.00942	1.7939	0.00713	13.09e-4	0.294	100

Tablo 7.6'dan görüldüğü gibi DEK-AÖM yöntemi OKHK, OMYH, OMH ve σ kriterlerinde 1000 nöronlu AÖM yönteminden yaklaşık 2.5 kat ve süre açısından 1.05 kat daha bir iyi performansa sahiptir. Önerilen yöntem DEK-AÖM yöntemi AÖM yönteminden 10 kat daha az nöron sayısına sahiptir.

**Şekil 7.11.** Mackey-Glass zaman serisi DEK-AÖM tahmin hatası eğrisi

Tablo 7.6'dan görüleceği gibi DEK-AÖM yöntemi OKHK, OMYH, OMH ve σ kriterlerinde kendi ile aynı nöron sayısına sahip AÖM yönteminden yaklaşık 5 kat daha iyi

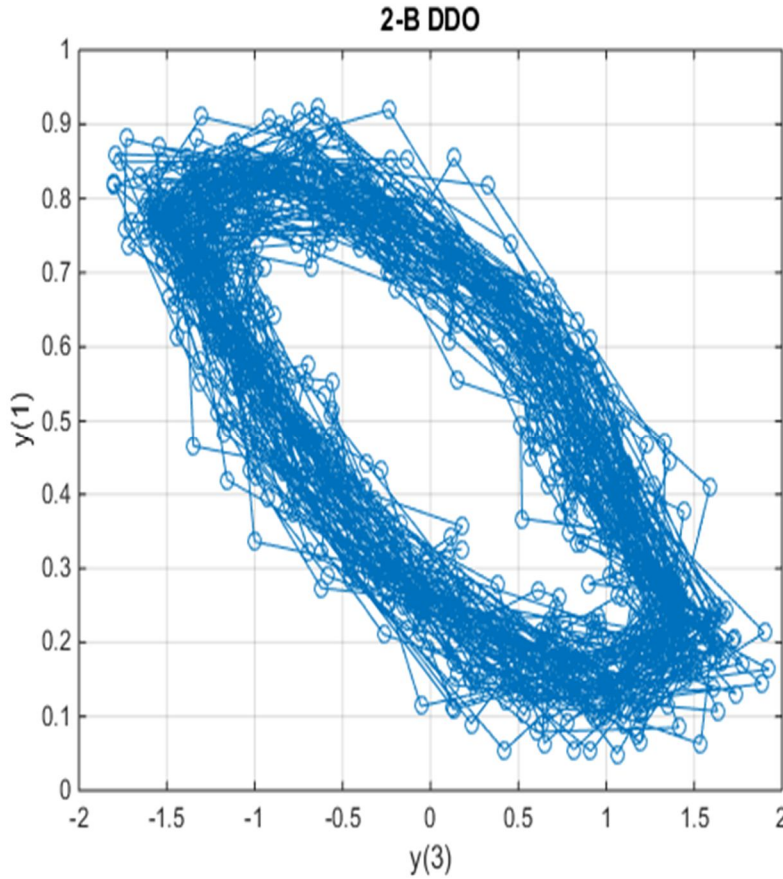
bir başarıma sahiptir. Süre açısından 100 nöronlu AÖM yöntemi DEK-AÖM ile arasında yaklaşık 289 ms fark vardır. Mackey-Glass zaman serisi için DEK-AÖM ile en uygun model elde edilmiştir.

7.3.3. 2-B Doğrusal ve Durağan Olmayan Zaman Serisi

2-B Doğrusal ve Durağan Olmayan (DDO) zaman serisi (7.3)'de belirtilmiştir [102].

$$y(t) = [0.8 - 0.5 * \exp(-y^2(t-1))] * y(t-1) - [0.3 - 0.9 * \exp(-y^2(t-1))] * y(t-2) + 0.1 * \sin + e(t) \quad (7.3)$$

Burada $e(t)$ seriye eklenen gürültüyü temsil etmektedir. Başlangıç şartları $y(1)=0, y(2)=1$ alınmıştır. Şekil 7.12'de 2-B DDO zaman serisi gösterilmiştir.



Şekil 7.12. 2-B DDO zaman serisi

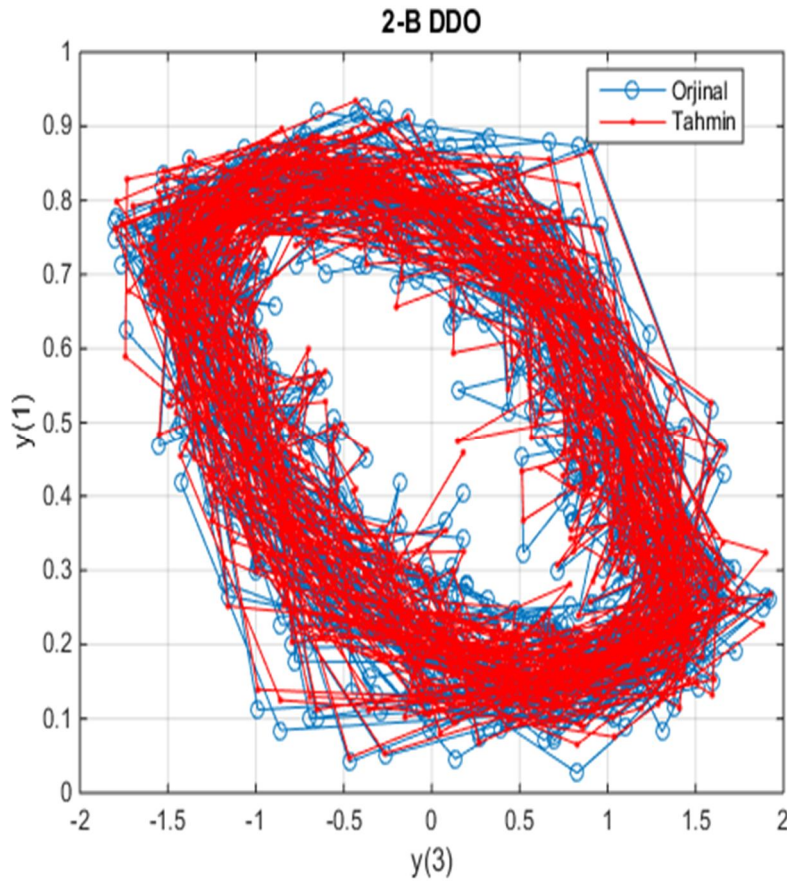
2-B DDO zaman serisi AÖM yöntemine uygulanmış ve elde edilen sonuçlar Tablo 7.7’de verilmiştir.

Tablo 7.7. 2-B DDO zaman serisi AÖM sonuçları

f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
sig	0.03332	8.2190	0.02610	13.5e-4	1.173	1000
sin	0.03268	8.2331	0.02590	8.4e-4	1.135	1000
hardlim	0.03458	8.6613	0.02734	8.8e-4	1.660	1000
tribas	0.03552	8.6829	0.02770	12e-4	1.403	1000
radbas	0.03911	8.5281	0.02688	77.1e-4	1.172	1000
elliott	6.84958	52.9872	0.24383	21.2440	1.384	1000

Elliot aktivasyon fonksiyonlu AÖM dışında diğer AÖM’lerde yaklaşık aynı sonuçlar elde edilmiştir. *Sin* aktivasyon fonksiyonlu AÖM ile en düşük OKHK, OMH ve σ elde edilmiştir. Süre açısından bakıldığında *hardlim* aktivasyon fonksiyonlu AÖM kabul edilebilir sonuçlar üretmiştir. *Elliot* aktivasyon fonksiyonu sonuçlardan görüleceği gibi bu zaman serisi için yanlış bir seçim olmuştur.

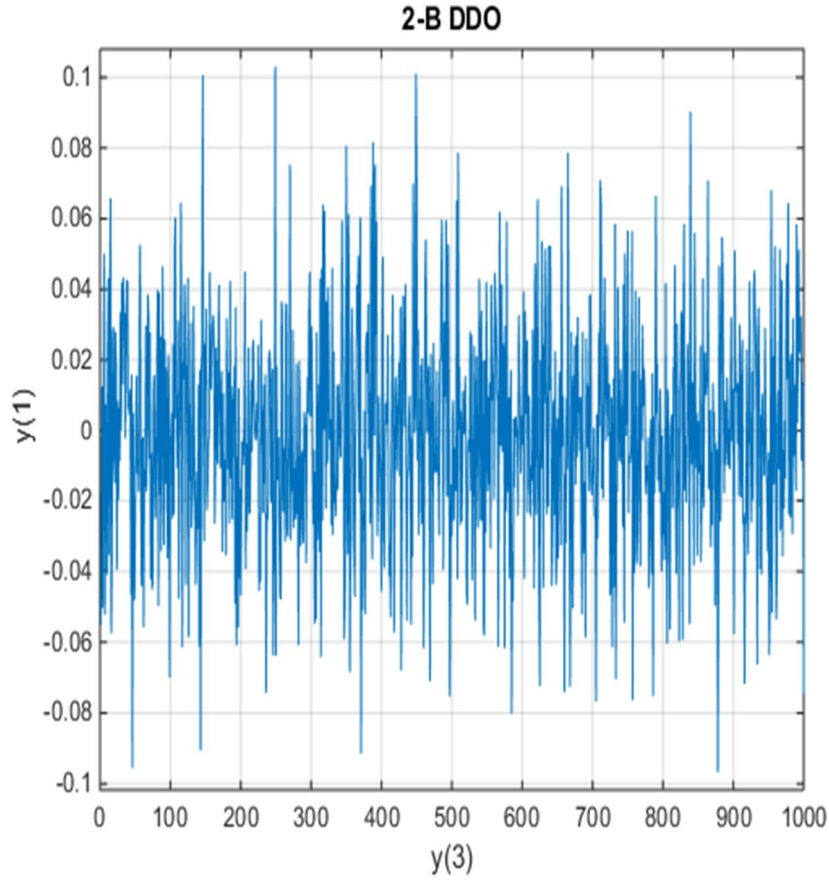
2-B DDO zaman serisi için AÖM yöntemi test tahmin eğrilesi Şekil 7.13’te ve AÖM yöntemi test tahmin hatası eğrileri Şekil 7.14’te gösterilmiştir. Şekil 7.13 ve Şekil 7.14’ten görüleceği gibi, AÖM tahmin eğrisi gerçek eğriye oldukça yakındır.



Şekil 7.13. 2-B DDO zaman serisi AÖM tahmin eğrisi

Tablo 7.8. 2-B DDO zaman serisi DEK-AÖM sonuçları

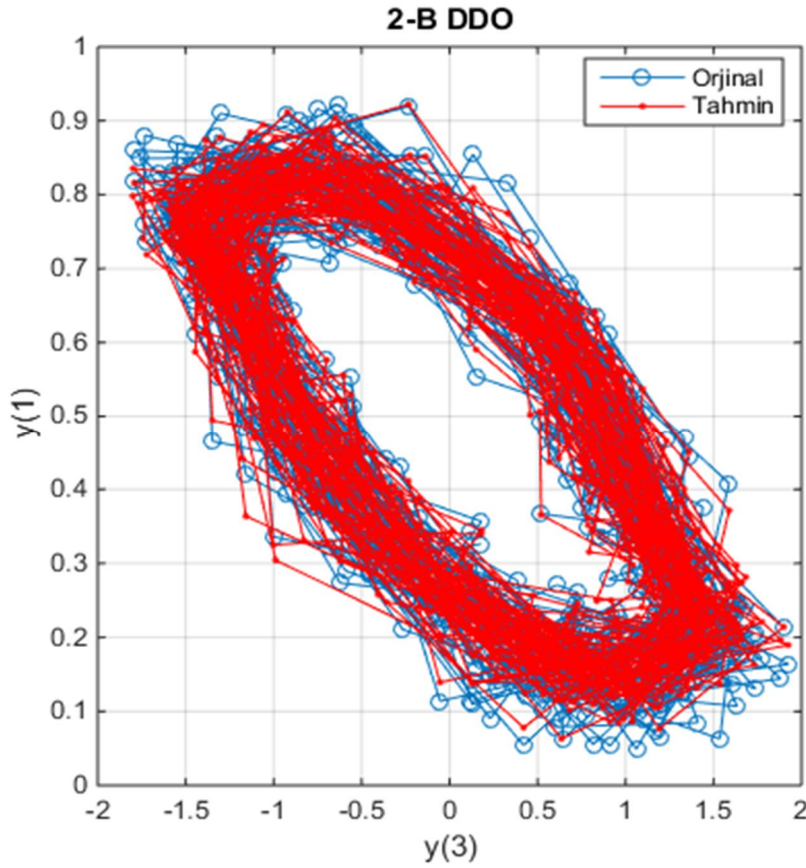
f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
sig	0.03260	8.2451	0.02589	8.28e-4	2.446	(100)1000
sin	0.03263	8.1976	0.02586	9.34e-4	2.157	(100)1000
hardlim	0.03465	8.6411	0.02740	9.09e-4	2.072	(100)1000
tribas	0.03271	8.2423	0.02598	7.82e-4	2.066	(100)1000
radbas	0.03263	8.2507	0.02590	8.52e-4	2.120	(100)1000
elliott	0.03264	8.1977	0.02591	8.27e-4	2.281	(100)1000



Şekil 7.14. 2-B DDO zaman serisi AÖM tahmin eğrisi

2-B DDO zaman serisi için DEK-AÖM sonuçları Tablo 7.8’de verilmiştir. Tablo 7.8’de verilen sonuçlar incelendiğinde en iyi OKHK değerine *sig* aktivasyon fonksiyonlu DEK-AÖM ile ulaşılmıştır. OMYH ve OMH kriterleri açısından *sin* aktivasyon fonksiyonlu DEK-AÖM σ ve süre açısından *tribas* aktivasyon fonksiyonlu DEK-AÖM iyi sonuçlara sahiptir.

2-B DDO zaman serisi için *hardlim* aktivasyon fonksiyonlu DEK-AÖM yöntemi test tahmin eğrisi Şekil 7.15 ve DEK-AÖM yöntemi test tahmin hatası eğrisi Şekil 7.16’da verilmiştir. Şekil 7.15’ten görüldüğü gibi DEK-AÖM yöntemi ile gerçek eğriye yakın bir tahmin eğrisi elde edilmiştir.

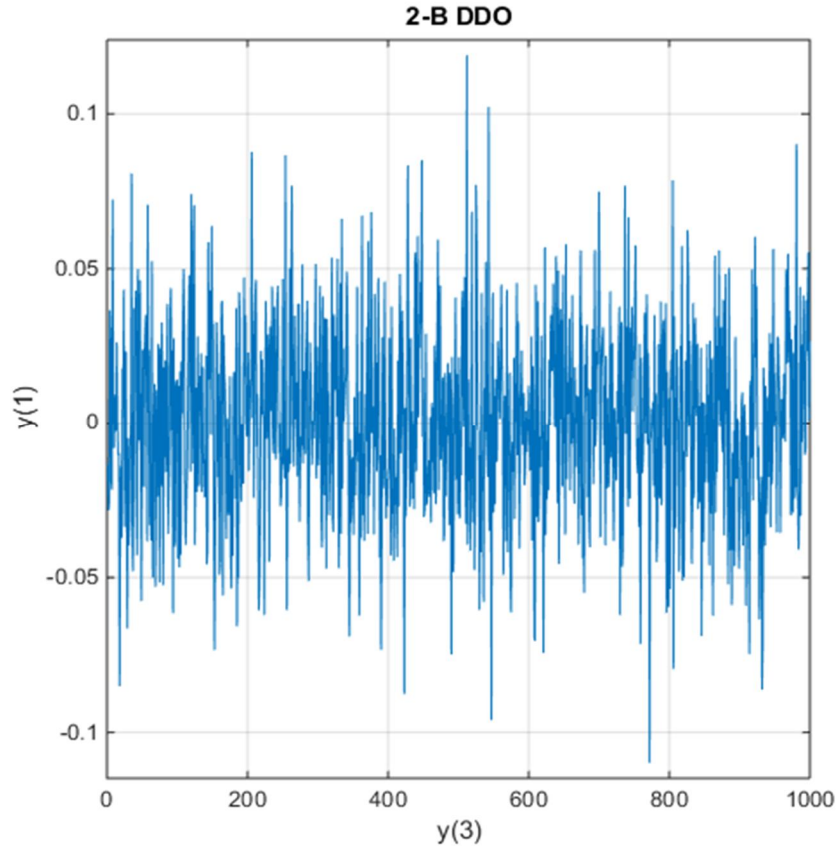


Şekil 7.15. 2-B DDO zaman serisi DEK-AÖM tahmin eğrisi

Tablo 7.9. 2-B DDO zaman serisi için AÖM ve DEK-AÖM karşılaştırması

	f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
AÖM	sin	0.03268	8.2331	0.02590	8.40e-4	1.135	1000
AÖM	sin	0.03273	8.2256	0.02589	8.42e-4	9e-5	100
DEK-AÖM	sig	0.03260	8.2451	0.02589	8.28e-4	2.446	100

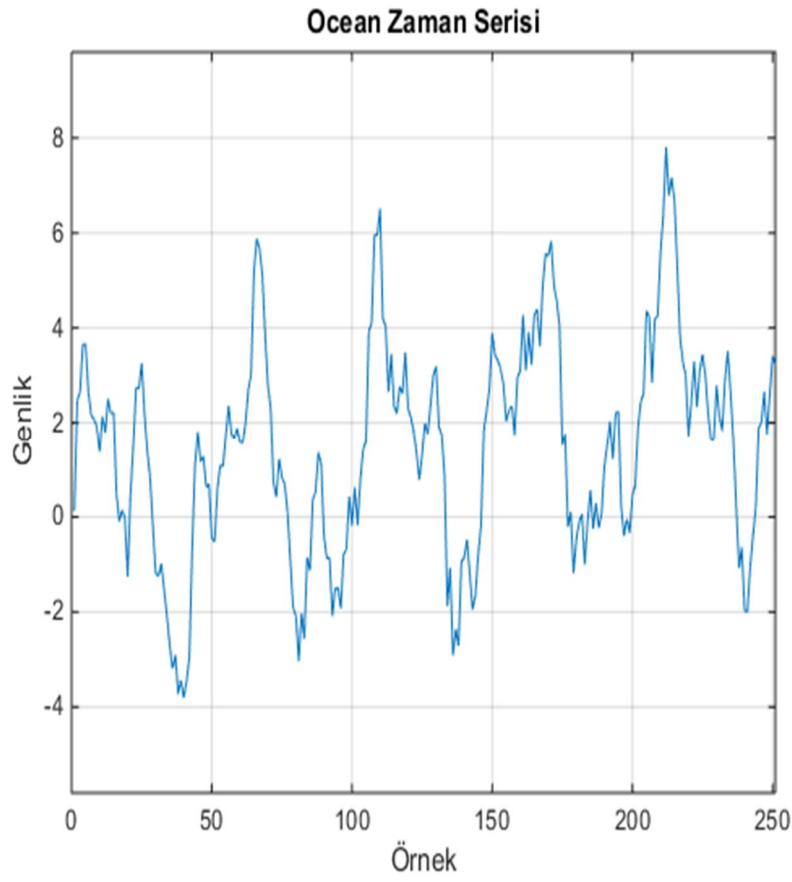
Tablo 7.9’da verilen sonuçlar incelendiğinde, OKHK ile σ kriterlerinde DEK-AÖM, OMYH kriterinde 100 nöronlu AÖM, süre açısından ise 100 nöronlu AÖM yöntemi ile kabul edilebilir sonuçlar elde edilmiştir. OMH kriterinde ise AÖM’ler ve DEK-AÖM yakın bir performansa sahip olmuştur. Yapılan deneysel sonuçlar dikkate alındığında, DEK-AÖM ile 100 nöronlu AÖM’ye göre daha iyi başarıma sahip en uygun model elde edilmiştir.



Şekil 7.16. 2-B DDO zaman serisi DEK-AÖM tahmin eğrisi

7.3.4. Ocean Zaman Serisi

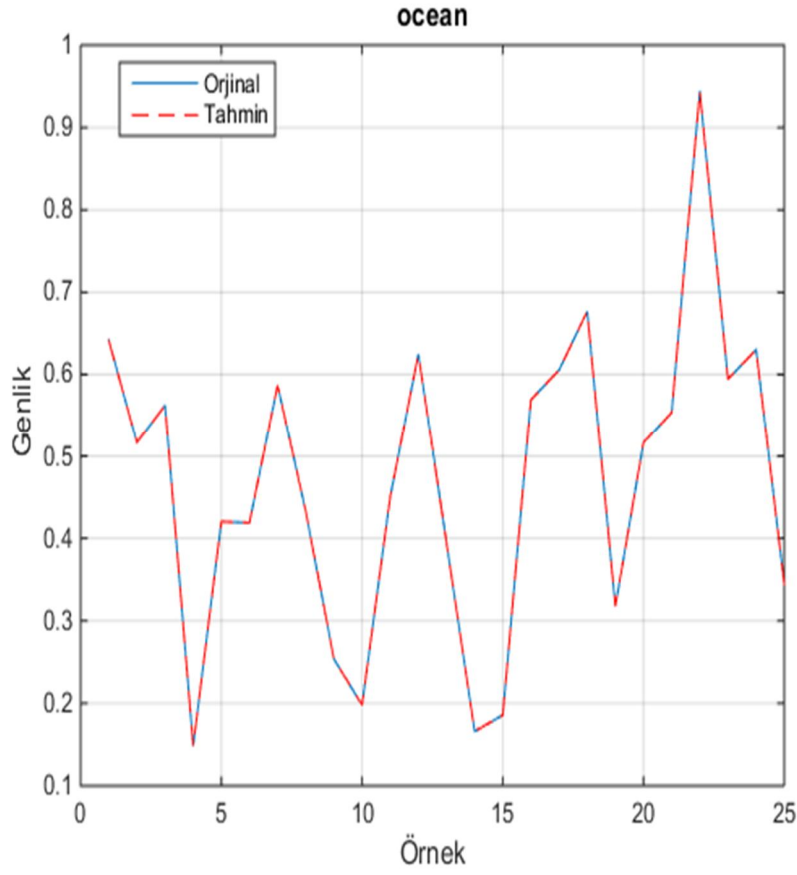
Bu zaman serisi [103]'den elde edilen yapay zaman serisidir. Deneyde kullanılan Ocean zaman serisi Şekil 7.17'de gösterilmiştir. Tablo 7.10'da Ocean zaman serisi AÖM sonuçlarına yer verilmiştir. Bu zaman serisi için *sin* aktivasyon fonksiyonlu AÖM süre dışındaki bütün kriterlerde en iyi başarıma sahiptir. *Sin* aktivasyonundan sonra süre kriterinde en iyi başarıma sahip *sig* aktivasyonu fonksiyonlu AÖM iyi başarıma sahip olduğu görülmektedir. *sin* aktivasyon fonksiyonlu AÖM test eğrisi Şekil 7.18'da ve AÖM test tahmin hata eğrisi Şekil 7.19'da verilmiştir. Şekil 7.18'da *sin* aktivasyon fonksiyonlu AÖM sonuçlarının mükemmel derecede gerçek eğriye yakınsadığı görülmektedir.



Şekil 7.17. Ocean zaman serisi

Tablo 7.10. Ocean zaman serisi AÖM sonuçları

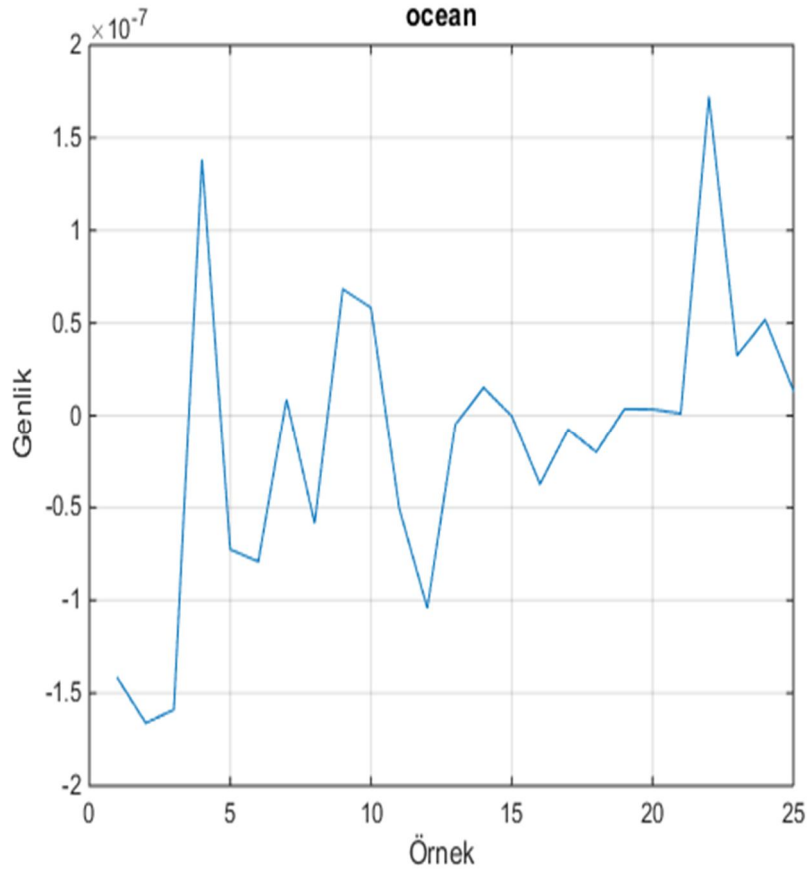
f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
sig	4.95e-6	5.816e-4	1.52e-6	4.740e-6	0.0411	1000
sin	0.31e-6	3.89e-5	1.4e-7	4.40e-7	0.0417	1000
hardlim	0.0126	3.215	9.22e-3	4.133e-3	0.0422	1000
tribas	7.98e-3	1.293	4.09e-3	5.422e-3	0.0416	1000
radbas	9.699e-4	0.109	3.06e-4	9.099e-4	0.0419	1000
elliott	3.155e-4	0.046	1.52e-4	1.786e-4	0.0414	1000



Şekil 7.18. Ocean zaman serisi zaman serisi AÖM tahmin eğrisi

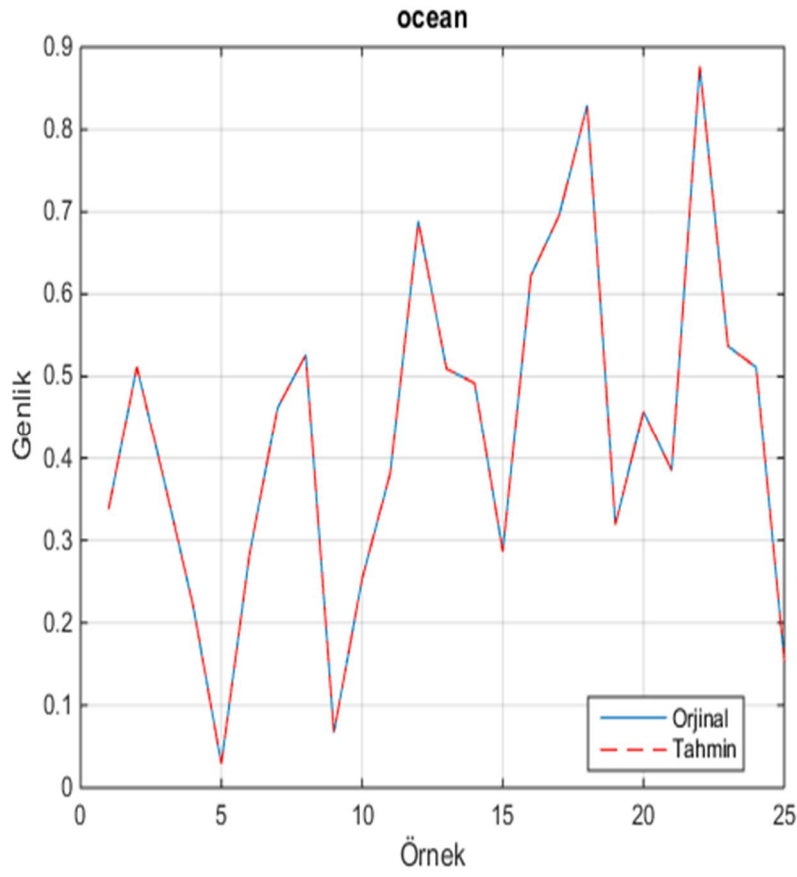
Tablo 7.11. Ocean zaman serisi DEK-AÖM sonuçları

f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
sig	6.44e-6	1.29e-3	27.1e-7	2.8e-6	0.588	(200)1000
sin	0.635e-6	1.43e-4	3.92e-7	7.09e-7	0.534	(200)1000
hardlim	0.0455	11.037	0.0335	0.0134	0.808	(200)1000
tribas	0.0132	2.979	0.0079	0.0103	0.785	(200)1000
radbas	0.0014	0.192	5.67e-4	1.63e-3	0.763	(200)1000
elliott	1.06e-5	1.78e-3	6.11e-6	7.70e-6	0.761	(200)1000



Şekil 7.19. Ocean zaman serisi zaman serisi AÖM tahmin hata eğrisi

Tablo 7.11 incelendiğinde Ocean zaman serisi için *sin* aktivasyon fonksiyonlu DEK-AÖM yöntemi bütün kriterlerde en iyi sonuçlara sahiptir. *Sin* aktivasyon fonksiyonundan sonra *sig* aktivasyon fonksiyonlu DEK-AÖM ikinci en iyi başarıma sahip olmuştur. DEK-AÖM tahmin eğrisi Şekil 7.20’de DEK-AÖM Tahmin hatası eğrisi Şekil 7.21’de verilmiştir. Şekil 7.20’den görüldüğü gibi DEK-AÖM yöntemi sonuçları gerçek sonuçlara mükemmel derecede yakınlık göstermektedir.

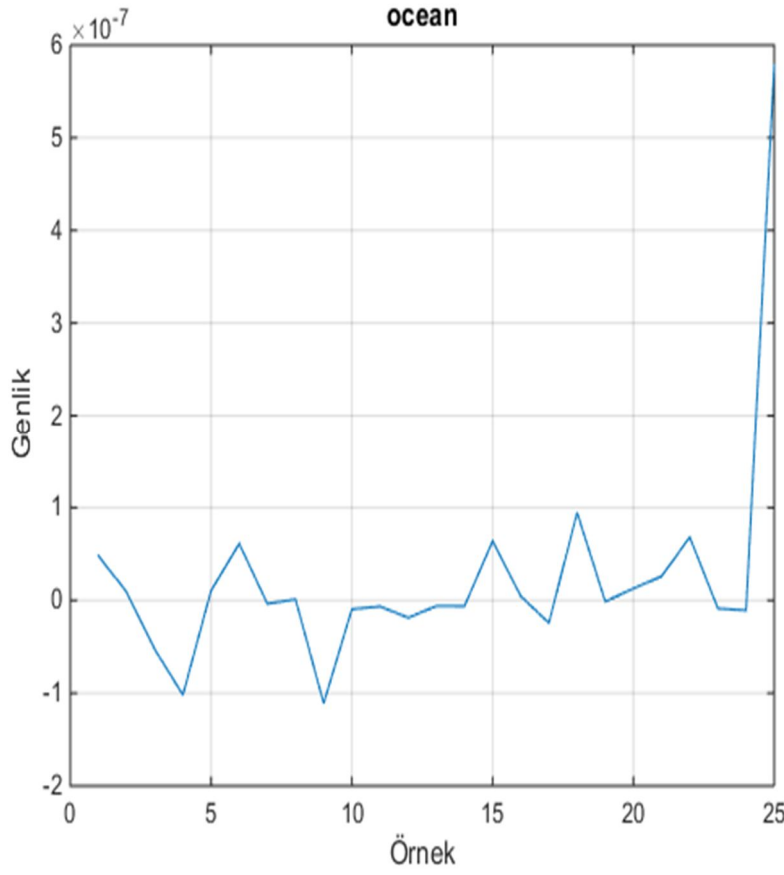


Şekil 7.20. Ocean zaman serisi DEK-AÖM tahmin eğrisi

Tablo 7.12’de Ocean zaman serisi için karşılaştırmalı sonuçlar verilmiştir. Tablo 7.12’den görüleceği gibi 1000 nöronlu AÖM yöntemi ile DEK-AÖM yöntemi kıyaslandığında OKHK, OMYH, OMH ve σ kriterlerinde 1000 nöronlu AÖM yöntemi öne çıkarken önerilen DEK-AÖM yöntemi 5 kat daha az nöron sayısına sahiptir.

Tablo 7.12. Ocean zaman serisi için AÖM ve DEK-AÖM karşılaştırması

	OKHK	OMYH	OMH	σ	Süre (s)	f(.)	Nöron
AÖM	0.31e-6	3.89e-5	1.4e-7	4.40e-7	0.0417	sin	1000
AÖM	0.708e-6	2.10e-4	5.40e-7	4.81e-7	0.012	sin	200
DEK-AÖM	0.635e-6	1.43e-4	3.92e-7	7.09e-7	0.534	sin	200

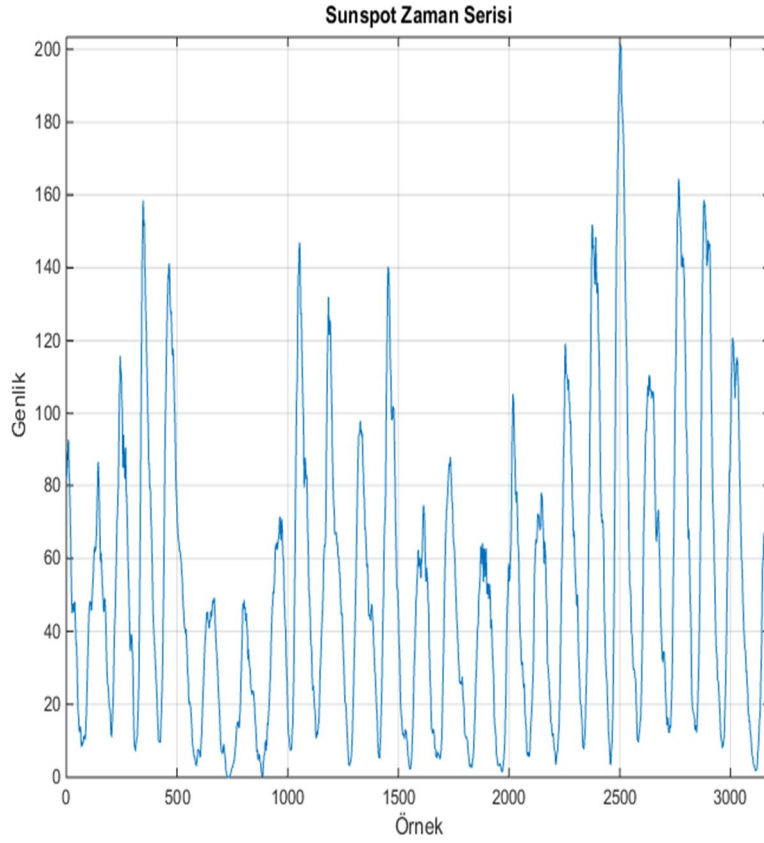


Şekil 7.21. Ocean zaman serisi DEK-AÖM tahmin hatası eğrisi

Yine Tablo 7.12’den görüleceği üzere DEK-AÖM ile aynı nöron sayısına sahip AÖM yöntemi karşılaştırıldığında OKHK, OMYH, OMH kriterlerinde DEK-AÖM yöntemi daha iyi başarıma sahipken σ kriterinde 100 nöronlu AÖM yöntemi daha iyi bir sonuç üretmiştir. Süre açısından karşılaştırıldığında; DEK-AÖM yöntemi 1000 nöronlu AÖM yönteminden yaklaşık 12.8 kat, 100 nöronlu AÖM yönteminden ise yaklaşık 44.5 kat bir oranla geride kalmaktadır.

7.3.5. Sunspot Zaman Serisi

Sunspot zaman serisi 1749-Mart 2013 yılına kadarki geçen sürede elde edilen verilerden oluşmaktadır [104]. Deneyde kullanılan Sunspot zaman serisi Şekil 7.22’de gösterilmiştir.



Şekil 7.22. Sunspot zaman serisi

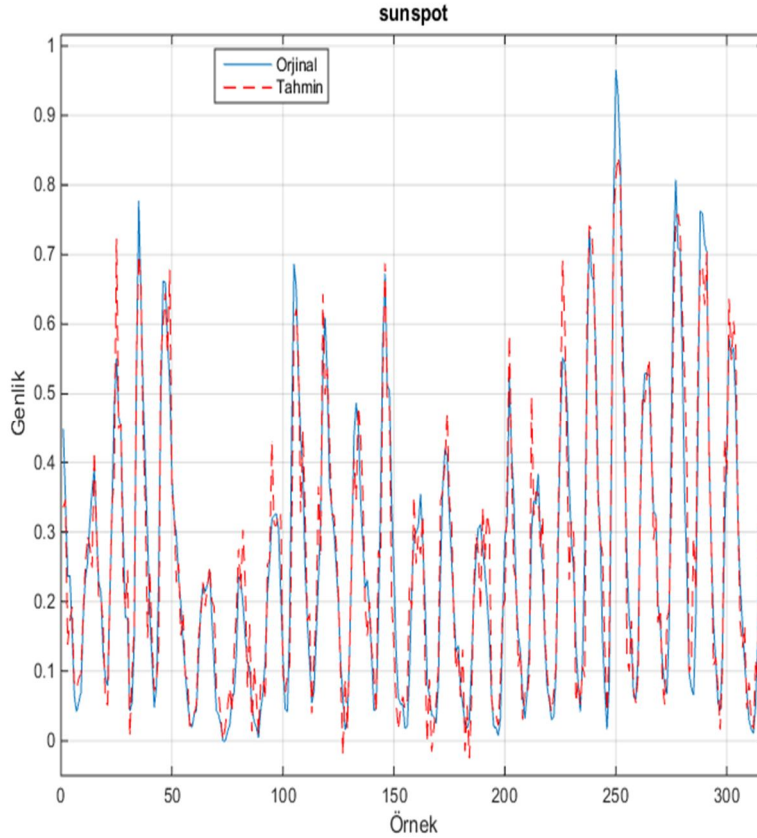
AÖM yöntemi Sunspot zaman serisine uygulanmış ve elde edilen sonuçlar Tablo 7.13'te verilmiştir.

Tablo 7.13. Sunspot zaman serisi AÖM sonuçları

f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
sig	12.606	133.341	0.786	31.206	0.599	1000
sin	9.607	102.933	0.5845	29.599	0.543	1000
hardlim	0.0701	41.704	0.0512	4.76e-3	0.725	1000
tribas	0.0817	37.997	0.0507	0.0171	0.679	1000
radbas	71.009	697.374	4.703	145.81	0.616	1000
elliott	0.570	45.869	0.0826	0.889	0.658	1000

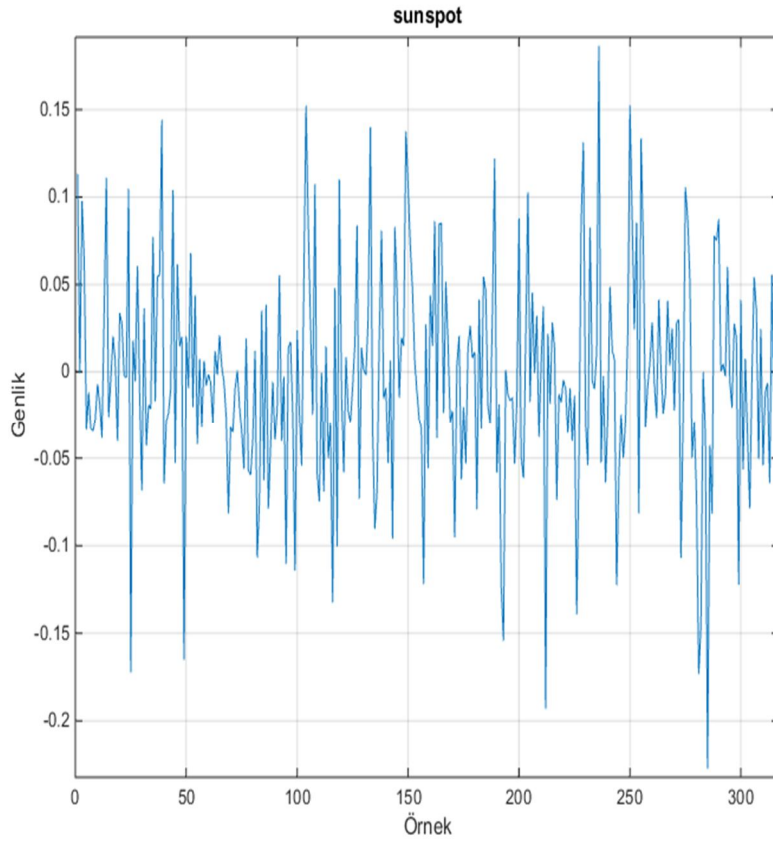
Tablo 7.13'ten görüleceği gibi *Hardlim* aktivasyon fonksiyonlu AÖM ile en iyi OKHK ve σ , buna ek olarak ikinci en iyi OMYH ve OMH değeri elde edilmiştir. *Tribas*

aktivasyon fonksiyonlu AÖM yöntemi *Hardlim*'li AÖM'ye yakın OKHK performansı göstermiş olup OMYH ve OMH açısında en iyi sonuçları üretmiştir. *Elliot*, *sinüs*, *radbas* ve *sigmoid* aktivasyon fonksiyonlu AÖM'ler Sunspot zaman serisi için kötü sonuçlara sahiptir. AÖM yöntemi tahmin eğrisi Şekil 7.23'te ve tahmin hatası ise Şekil 7.24'te verilmiştir.



Şekil 7.23. Sunspot zaman serisi AÖM tahmin eğrisi

Tablo 7.14'te Sunspot zaman serisi önerilen DEK-AÖM yöntemi sonuçları verilmiştir. Tablo 7.14 incelendiğinde *elliot* aktivasyon fonksiyonlu DEK-AÖM OKHK, OMYH, OMH ve σ kriterlerinde en iyi sonuçlara sahiptir. *Tribas* aktivasyon fonksiyonlu DEK-AÖM yöntemi *elliot*'a yakın bir performans göstermiştir.

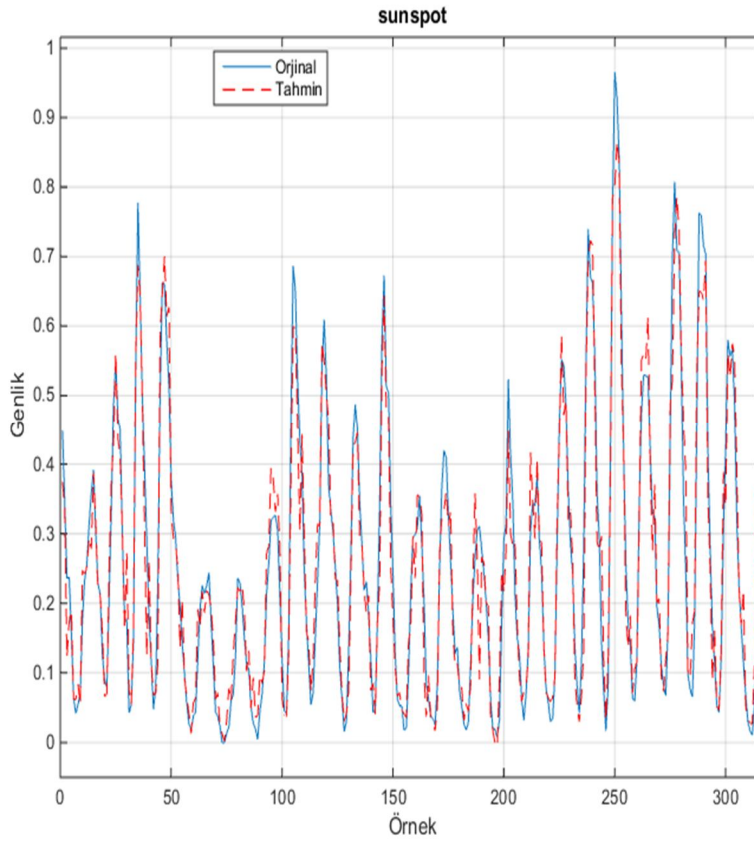


Şekil 7.24. Sunspot zaman serisi AÖM tahmin hatası eğrisi

Tablo 7.14. Sunspot zaman serisi DEK-AÖM sonuçları

f(.)	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
sig	0.0667	35.043	0.0436	0.0213	0.754	(100)1000
sin	0.1019	35.202	0.0457	0.1256	0.750	(100)1000
hardlim	0.0610	35.781	0.0445	0.0382	0.759	(100)1000
tribas	0.0586	34.289	0.0421	0.0041	0.829	(100)1000
radbas	0.0698	35.051	0.0431	0.0315	0.753	(100)1000
elliott	0.0580	33.592	0.0416	0.0037	0.759	(100)1000

Şekil 7.25'te DEK-AÖM yöntemi tahmin eğrisi ve Şekil 7.27'de DEK-AÖM yöntemi tahmin hatası verilmiştir.



Şekil 7.25. Sunspot zaman serisi DEK-AÖM tahmin eğrisi

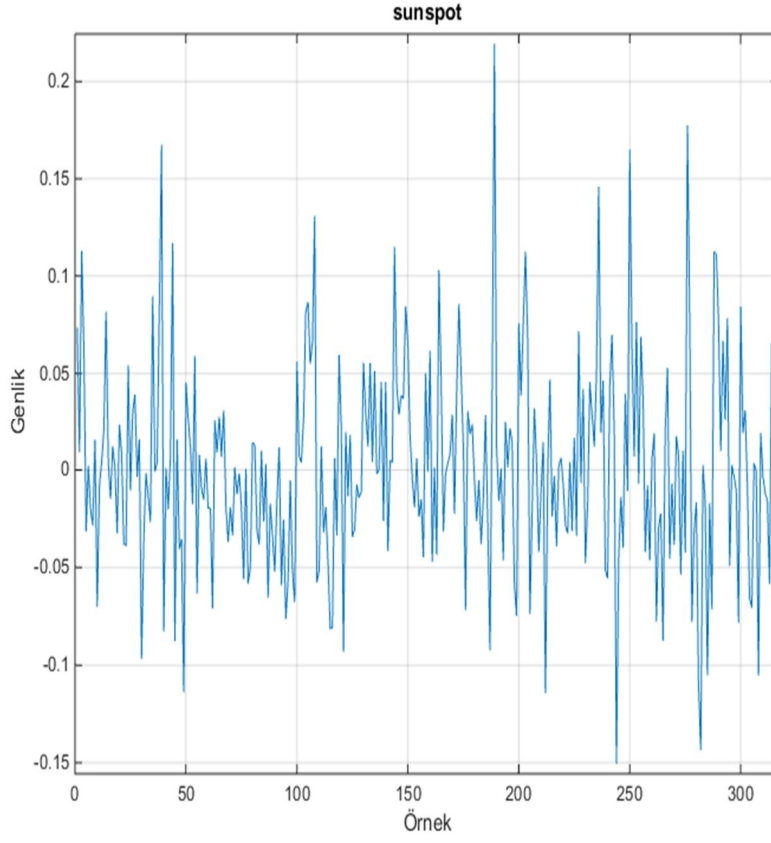
Sunspot zaman serisi AÖM ile DEK-AÖM yöntemi karşılaştırma sonuçları Tablo 7.15'te verilmiştir.

Tablo 7.15. Sunspot zaman serisi için AÖM ve DEK-AÖM karşılaştırması

	$f(.)$	OKHK	OMYH	OMH	σ	Süre (s)	Nöron
AÖM	hardlim	0.0701	41.704	0.0512	4.766e-3	0.725	1000
AÖM	hardlim	0.0650	41.012	0.0476	3.887e-3	0.0144	100
DEK-AÖM	elliot	0.0580	33.592	0.0416	3.7e-3	0.759	100

DEK-AÖM yöntemi, 1000 nöronlu AÖM yönteminden süre dışında bütün kriterlerde iyi başarıma sahip olduğu görülmektedir. Süre açısından bu iki yöntem arasında 34ms bir fark vardır. DEK-AÖM ile aynı nöron sayısına sahip AÖM yöntemi

karşılaştırıldığında süre açısından yaklaşık 52 kat DEK-AÖM yöntemi geri kalırken diğer bütün kriterlerde DEK-AÖM yöntemi iyi başarıma sahiptir.



Şekil 7.26. Sunspot zaman serisi DEK-AÖM tahmin hatası eğrisi

8. SONUÇLAR VE DEĞERLENDİRME

Bu tez çalışmasında AÖM'nin optimizasyonu için YFT tabanlı altı farklı yöntem kullanılmıştır. AÖM ağının gizli katman nöron sayısı gerekenden fazla düşünülerek YFT yöntemleri ile çıkış katmanı ağırlıkları hesaplanarak ilk başta fazla seçilen nöronlar budanmıştır. Böylece en uygun gizli katman nöron sayısı ve düşük işlem karmaşıklığına sahip AÖM yapısı elde edilmiştir. Bu tez çalışmasında kullanılan YFT algoritmalarından YSE, DET, SÖET, KDET algoritmalarının döngü durdurma kriteri için seyreklik seviyesi kullanılmıştır. Bu dört algoritmanın seyreklik seviyesinin seçilmesi için en düşük OKHK değerine bağlı iki aşamalı arama algoritması kullanılmıştır. İGT algoritması için döngü durdurma kriteri seyreklik bilgisi yerine hata değeri kullanılmıştır. DEK algoritması için döngü durdurma kriteri olarak seyreklik bilgisi kullanılmış olup Ocean veri seti için 200 diğer veri setleri için 100 değeri seçilmiştir. Önerilen yöntemlerin etkinliği regresyon, sınıflandırma ve zaman serisi tahmin problemlerine uygulanarak gösterilmiştir. Deneysel çalışmalardan elde edilen sonuçların değerlendirilmesi bölüm bazında yapılmıştır.

8.1. Sonuçların Değerlendirilmesi

Bölüm 3: Bu bölümde AÖM ağının optimizasyonu için YSE-AÖM yöntemi önerilmiştir. YSE-AÖM yönteminin etkinliğini göstermek amacıyla temel AÖM ile karşılaştırılmıştır. Bu bölümde deneylerde Breast Cancer, Pima Indians Diabetes, Bupa ve Body Fat Veri setleri kullanılmıştır. Breast Cancer veri setinde yaklaşık 64 kat, Pima Indians Diabetes veri setinde yaklaşık 3 kat, Bupa veri setinde yaklaşık 6 kat ve Body Fat veri setinde 15 kat daha az nöron elde edilmiştir. Body Fat veri seti dışındaki bütün veri setlerinde YSE-AÖM yöntemi daha iyi başarıma ve daha kararlı sonuçlara sahiptir. Bupa veri seti dışındaki bütün veri setlerinde YSE-AÖM yöntemi deneyleri daha kısa süre içinde gerçekleştirmiştir. Genel itibarıyla YSE-AÖM yöntemi ile en uygun nöron sayısına ve genel itibarıyla düşük işlem karmaşıklığına sahip ağ yapısı elde edilmiştir.

Bölüm 4: Bu bölümde DET algoritması tabanlı düzenlenmiş AÖM yöntemi önerilmiştir. Önerilen yöntemin performansını göstermek amacıyla yaygın kullanılan dokuz adet regresyon problemi ile deneyler yapılmıştır. Ayrıca DET-AÖM yöntemi AÖM, EMDSO ve EN gibi yöntemler ile karşılaştırılmıştır. Önerilen yöntem diğer yöntemler ile

karşılaştırıldığında hem kompakt yapıda ağ mimarisi sağlamakta hem de daha üstün başarımlar göstermektedir. Özetle DET-AÖM yöntemi en uygun nöron sayısı ile temel AÖM, EMDSO ve EN gibi yöntemlerden daha iyi bir başarımla değerine sahiptir. Önerilen yöntemin geleneksel YSA ile karşılaştırılması yapılmıştır. YSA'nın daha az nöron sayısına sahipken başarımları DET-AÖM başarımlarından çok daha iyi olduğu görülmektedir.

Bölüm 5: Bu çalışmada AÖM ağının gizli katman nöron sayısının otomatik budanması işlemini yapan dört adet YFT algoritması önerilmiştir. Önerilen yöntemin üstünlüğünü göstermek amacı ile farklı deneyler yapılmıştır. İlk olarak önerilen YFT algoritmaları kendi aralarında karşılaştırılarak en iyi başarımla sahip algoritma temel AÖM ile karşılaştırılmıştır. Ayrıca DET algoritması ile AÖM karşılaştırmasının pekiştirilmesi ve görselleştirilmesi için gizli katman nöron sayısının artışına karşın ortalama OKHK değişim eğrisi çizdirilmiştir. Ardından en iyi başarımla sahip DET algoritması ile literatürde sıkça kullanılan EMDSO ve EN yöntemleri ile karşılaştırma yapılmıştır. EMDSO ve EN yöntemleri kompakt yapıda mimariye sahip olmaktadır. Fakat AÖM'ye benzer genelleme başarımları üretmiştir. Öte yandan önerilen yöntem AÖM, EMDSO ve EN yöntemleri ile karşılaştırıldığında kompakt ağ mimarisine ve üstün başarımla sahip olmuştur. Özetle önerilen YFT-SAÖM yöntemi hem hızlı eğitim sürecini koruyabilmektedir hem de optimum gizli katman nöron sayısını otomatik olarak seçebilmektedir. Ayrıca önerilen yöntemin doğruluğu AÖM ve EMDSO ile EN yöntemlerinden daha iyi olmaktadır.

Bölüm 6: Bu çalışmada, AÖM ağının çıkış ağırlıklarının hesaplanması için SGÇ yöntemlerinden İGT algoritması kullanılmıştır. İGT algoritmasının önemli parametrelerinden olan α ve μ belirlenmesi işlemleri için en yüksek doğruluğu bulmayı amaçlayan bir arama algoritması kullanılmıştır. Deneysel çalışmalarda biyomedikal veri setlerinin sınıflandırılması yapılmıştır. İGT-AÖM yönteminin üstünlüğünü göstermek için normal AÖM ve EAR, EMDSO ve EN yöntemleri ile karşılaştırması yapılmıştır. Yapılan deneysel çalışmaların ışığında İGT-AÖM yöntemi ile amaçlandığı gibi yüksek doğruluk yüzdesine sahip en uygun nöronlu model elde edilmiştir. İGT-AÖM yöntemi sayesinde normal AÖM, EAR, EMDSO ve EN yöntemlerinden dokuz kat daha az nöron sayısına sahip model oluşturulmuştur. İGT-AÖM yöntemi Standart sapma açısından %60 oranında normal AÖM, EAR, EMDSO ve EN yöntemlerine üstünlük sağlamıştır. İGT-AÖM yönteminde kullanılan α ve μ katsayıları için kullanılan arama algoritmasından dolayı deneysel çalışma süresi açısından EMDSO, EAR yöntemlerinin gerisinde kalırken, genel itibarıyla normal AÖM ve Elastic Netten daha iyi sonuçlara sahiptir. Sonuç olarak,

önerilen İGT-AÖM yöntemi genel itibariyle sınıflandırma doğruluğu en uygun nöron sayısı ve standart sapma kriterleri bakımından üstün bir başarıma sahipken deneysel çalışma süresi bakımından EAR ve EMDSO yönteminin daha az başarıma sahiptir.

Bölüm 7: Çeşitli değişkenli zaman serilerinin tahmin problemini çözmek için DEK-AÖM önerilmiştir. Önerilen yöntem 3 tip zaman serisi kullanılarak test edilmiş, 1000 ve 100 nöronlu AÖM ile karşılaştırılmıştır. DJIA zaman serisi için önerilen DEK-AÖM yöntemi 1000 nöronlu AÖM yönteminin çok az bir fark ile gerisinde kalırken bu başarıyı 10 kat daha az nöron ile elde etmiştir. AÖM yönteminde nöron sayısı 100 olduğunda elde edilen sonuçlar süre dışında bütün kriterlerde DEK-AÖM'nin gerisinde kalmaktadır. Öte yandan DJIA zaman serisi için DEK-AÖM ile elde edilen modelin daha kararlı olduğu sonuçlardan görülmektedir. Mackey-Glass zaman serisi için önerilen DEK-AÖM yöntemi 1000 nöronlu AÖM yönteminden bütün kriterlerde daha iyi sonuç vermiştir. Yine aynı zaman serisi için DEK-AÖM yöntemi 100 nöronlu AÖM yöntemine göre süre hariç bütün kriterlerde daha iyi bir başarıyı sergilemiştir. 2-boyutlu DDO zaman serisi için önerilen DEK-AÖM 1000 nöronlu AÖM yönteminden süre dışındaki bütün kriterlerde daha iyi performans sonuçlarına sahiptir. OKHK ile σ kriterlerinde DEK-AÖM, OMYH kriterinde 100 nöronlu AÖM, süre açısından ise 100 nöronlu AÖM yöntemi ile iyi sonuçlar elde edilmiştir. Yapılan deneysel sonuçlar dikkate alındığında, bu zaman serisi için DEK-AÖM ile hem 1000 nöronlu AÖM hem de 100 nöronlu AÖM'ye göre daha iyi başarıya sahip en uygun model elde edilmiştir. Önerilen 100 nörona sahip DEK-AÖM yöntemi Ocean zaman serisini modelleyememiştir. Bundan dolayı Ocean zaman serisinde 200 nöronlu DEK-AÖM yöntemi kullanılmıştır. Ayrıca karşılaştırmanın sağlıklı yapılabilmesi için AÖM nöron sayısı da 200 olarak belirlenmiştir. 1000 nörona sahip AÖM yöntemi DEK-AÖM yöntemine göre daha iyi sonuçları üretmiştir. DEK-AÖM ile aynı nöron sayısına sahip AÖM yöntemi karşılaştırıldığında OKHK, OMYH, OMH kriterlerinde DEK-AÖM yöntemi daha iyi başarıya sahipken σ ve süre kriterinde 100 nöronlu AÖM yöntemi daha iyi bir sonuç üretmiştir. Sun spot zaman serisi için DEK-AÖM yöntemi, 1000 nöronlu AÖM yönteminden süre dışında bütün kriterlerde iyi başarıya sahip olduğu görülmektedir. Süre açısından ise kabul edilebilir bir fark söz konusudur. DEK-AÖM ile aynı nöron sayısına sahip AÖM yöntemi karşılaştırıldığında yine süre açısından DEK-AÖM yöntemi geri kalırken diğer bütün kriterlerde DEK-AÖM yöntemi iyi başarıya sahiptir. Yapılan deneylerden elde edilen veriler DEK-AÖM yöntemi ile daha az nöron sayısı ile genellikle daha iyi başarıya sahip modelin elde edildiğini göstermiştir. Önerilen yöntemdeki DEK

algoritması yinelemeli bir yapıya sahip olduğu için süre açısından AÖM yöntemin gerisinde kalmaktadır.

8.2. Öneriler

Tez çalışması süresince yapılan çalışmalara dayanarak ileri çalışmalar olarak önerilebilecek bazı konular şunlardır;

- Çekirdek tabanlı AÖM yöntemi benzetim çalışmalarında ve pratik uygulamalarda iyi bir test sonucu göstermektedir. Çekirdek tabanlı AÖM yöntemi büyük ölçekli verilerde ve sıralı gelen verilerle çalışırken uzun eğitim süresi ve hafıza taşması gibi problemlerle karşı karşıya kalmaktadır. Bundan dolayı çekirdek tabanlı AÖM yöntemine seyreklik yaklaşımı uygulanarak yeni bir metot önerilebilir.
- Bu tez çalışmasında sunulduğu gibi önerilen AÖM yöntemleri regresyon, sınıflandırma ve zaman serisi problemleri için iyi genelleme performansı göstermiştir. Veriler, önerilen AÖM özellik eşleme uzayında iyi bir şekilde yakınsanabilir veya sınıflandırılabilir. Böylece önerilen AÖM yöntemlerinin kümeleme problemlerinde nasıl başarımlar göstereceği incelenebilir.
- İmge işleme teknolojilerinin hızlı gelişmesi ve pratik imge işleme uygulamalarına artan talep imge sınıflandırma ve tanılama teknolojileri son zamanlarda hızla gelişmektedir. Önerilen AÖM yöntemlerinin imge işleme uygulamalarında artan hız ve doğruluğu talebini ne ölçüde karşılayabileceği araştırılabilir.

KAYNAKLAR

- [1] **Huang, G. B., Zhu, Q. Y., and Siew, C. K.**, 2004, Extreme learning machine: a new learning scheme of feedforward neural networks, *IEEE International Joint Conference on Neural Networks*, Budapest, 25-29 July, **2**, 985-990.
- [2] **Huang, G. B., Zhu, Q. Y. and Siew, C. K.**, 2006, Extreme learning machine: theory and applications. *Neurocomputing*, **70(1)**, 489-501.
- [3] **Hornik, K.**, 1991, Approximation capabilities of multilayer feedforward networks. *Neural networks*, **4(2)**, 251-257.
- [4] **Leshno, M., Lin, V. Y., Pinkus, A. and Schocken, S.**, 1993, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, *Neural networks*, **6(6)**, 861-867.
- [5] **Ito, Y.**, 1992, Approximation of continuous functions on R^d by linear combinations of shifted rotations of a sigmoid function with and without scaling, *Neural Networks*, **5(1)**, 105-115.
- [6] **Huang, G. B. and Babri, H.**, 1998, Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions, *IEEE Transactions on Neural Networks*, **9(1)**, 224-229.
- [7] **Tamura, S. I. and Tateishi, M.**, 1997, Capabilities of a four-layered feedforward neural network: four layers versus three, *IEEE Transactions on Neural Networks*, **8(2)**, 251-255.
- [8] **Huang, G. B.**, 2003, Learning capability and storage capacity of two-hidden-layer feedforward networks, *IEEE Transactions on Neural Networks*, **14(2)**, 274-281.
- [9] **Huang, G. B. and Siew, C. K.**, 2006, Real-time learning capability of neural networks. *IEEE Transactions on Neural Networks*, **17(4)**, 863-878.
- [10] **Hongming, Z.**, 2013, Extreme Learning Machine for Classification and Regression, *PhD Thesis*, School of Electrical & Electronic Engineering, the Nanyang Technological University, Singapore.
- [11] **Bartlett, P. L.**, 1998, The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network, *IEEE Transactions on Information Theory*, **44(2)**, 525-536.

- [12] **Luo, J., Vong, C. M. and Wong, P. K.**, 2014, Sparse Bayesian Extreme Learning Machine for Multi-classification, *Neural Networks and Learning Systems*, **25(4)**, 836-843, 2014.
- [13] **Banerjee, K. S.**, 1973, Generalized inverse of matrices and its applications, *Technometrics*, **15(1)**, 197-197.
- [14] **Bishop, C. M.**, 2006 Pattern recognition and machine learning, Springer New York Inc., New York.
- [15] **Martínez-Martínez, J. M., Escandell-Montero, P., Soria-Olivas, E., MartíN-Guerrero, J. D., Magdalena-Benedito, R. and Gómez-Sanchis, J.**, 2011. Regularized extreme learning machine for regression problems, *Neurocomputing*, **74(17)**, 3716-3721.
- [16] **Shi, L. C. and Lu, B. L.**, 2013, EEG-based vigilance estimation using extreme learning machines, *Neurocomputing*, **102**, 135-143.
- [17] **Huang, G. B., Li, M. B., Chen, L. and Siew, C. K.**, 2008, Incremental extreme learning machine with fully complex hidden nodes, *Neurocomputing*, **71(4)**, 576-583.
- [18] **Huang, G. -B., Zhou, H., Ding, X. and Zhang, R.**, 2012, Extreme learning machine for regression and multiclass classification, *Systems, Man, and Cybernetics, Part B: Cybernetics*, **42(2)**, 513-529.
- [19] **Li, G., Wen, C., Li, Z. G., Zhang, A., Yang, F. and Mao, K.**, 2013, Model-Based Online Learning with Kernels, *Neural Networks and Learning Systems*, **24(3)**, 356-369.
- [20] **Kivinen, J., Smola, A. J. ve Williamson, R. C.**, 2004, Online learning with kernels, *Signal Processing*, **52(8)**, 2165-2176.
- [21] **Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C. and Lendasse, A.**, 2010. OP-ELM: optimally pruned extreme learning machine, *IEEE Transactions on Neural Networks*, **21(1)**, 158-162.
- [22] **Miche, Y., Van Heeswijk, M., Bas, P., Simula, O. and Lendasse, A.**, 2011, TROP-ELM: a double-regularized ELM using LARS and Tikhonov regularization, *Neurocomputing*, **74(16)**, 2413-2421.

- [23] **Alcin, O. F., Sengur, A., Qian, J. and Ince, M. C.,** 2015, OMP-ELM: Orthogonal Matching Pursuit-Based Extreme Learning Machine for Regression, *Journal of Intelligent Systems*, **24(1)**, 135-143.
- [24] **Efron, B., Hastie, T., Johnstone, I. and, Tibshirani, R.,** 2004, Least angle regression, *The Annals of statistics*, **32(2)**, 407-499.
- [25] **Zou, H. and Hastie, T.,** 2005, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67(2)**, 301-320.
- [26] **Natarajan, B. K.,** 1995, Sparse approximate solutions to linear systems, *SIAM journal on computing*, **24(2)**, 227-234.
- [27] **Cand , E. J. and Wakin, M. B.,** 2008, An introduction to compressive sampling, *IEEE Signal Processing Magazine*, **25(2)**, 21-30.
- [28] **Baraniuk, R.G.,** 2007, Compressive Sensing, *Signal Processing Magazine*, **24(4)**, 118-121.
- [29] **Karakuş, C. ve G rb z, A.C.,** 2011, Yinelemeli seyrek geri oluřturma algoritmalarının karřılařtırılması, *19. Sinyal İřleme ve İletiřim Uygulamaları*, Hacettepe  niversitesi, Antalya, 20-22 Nisan, s.857-860.
- [30] **Gilbert, A. and Indyk, P.,** 2010, Sparse Recovery Using Sparse Matrices, *Proceedings of the IEEE*, **98(6)**, 937-947.
- [31] **Ayas, L. ve G rb z, A.C.,** 2010, Sıkıřtırılmıř algılamada gerekli  l  m sayısının analizi, *18. Sinyal İřleme ve İletiřim Uygulamaları*, Dicle  niversitesi, Diyarbakır, 22-24 Nisan, s.914-917.
- [32] **Teke, O., Gurbuz, A.C. ve Arikan, O.,** 2012, Seyrek geri  atma i in yeni bir OMP y ntemi, *20. Sinyal İřleme ve İletiřim Uygulamaları*,  zye in  niversitesi, Mu la, 18-20 Nisan, s1-4.
- [33] ** nce, T., Nacaro lu, A. ve Watsuji, N.,** 2010, Seyrek Olarak Bozulmuř  l  mlerden Blok Seyrek Sinyallerin Geri Kazanımı, *Elektrik-Elektronik-Bilgisayar M hendisli i Sempozyumu*, Bursa, 2-5 Aralık, s. 622-625.
- [34] **Argaez, M., Ramirez, C. and Sanchez, R.,** 2011, An l_1 -algorithm for underdetermined systems and applicatons, *Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, Texas, 18-20 March, s.1-6.

- [35] **Candes, E.J. ve Tao, T.**, 2005, Decoding by linear programming, *Information Theory*, **51(12)**, 4203-4215.
- [36] **Needell, D.**, 2009, Topics in Compressed Sensing, PhD Dissertation, Mathematics, Univ. of California, Davis.
- [37] **Donoho, D. L. and P. B. Stark.**, 1989, Uncertainty principles and signal recovery, *SIAM J. Appl. Math.*, **49(3)**, 906–931.
- [38] **Chen, S. and Donoho, D. L.**, 1994, Basis Pursuit, *1994 Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove 31 October-02 November, 41-44.
- [39] **Mallat, S.G. and Zhang, Z.**, 1993, Matching pursuits with time-frequency dictionaries, *IEEE Transactions on Signal Processing*, **41(12)**, 3397-3415.
- [40] **Donoho, D. L., Tsaig, Y., Drori, I. and Starck, J.-L.**, 2007, Sparse solution of underdetermined linear equations by stagewise Orthogonal Matching Pursuit (StOMP), *IEEE Transactions on Information Theory*, **58(2)**, 1094-1121.
- [41] **Temlyakov, V.N.**, 1999, Greedy Algorithms and M-Term Approximation with Regard to Redundant Dictionaries, *Journal of Approximation Theory*, **98(1)**, 117–145.
- [42] **Tropp, J.A.**, 2004, Greed is good: algorithmic results for sparse approximation, *IEEE Transactions on Information Theory*, **50(10)**, 2231-2242.
- [43] **Alcin O. F., Sengur A. and Ince M. C.**, 2013, Iterative Hard Thresholding Based Extreme Learning Machine and Its Application to Predictions in Medical Datasets, *Brain and Health Informatics*, Maebashi, October 29-31, 2013,.
- [44] **Alçin, Ö. F., Arı, A., Şengür, A. and İnce, M. C.**, Yinelemeli Sert Eşikleme Tabanlı Aşırı Öğrenme Makinası, 23. *Sinyal İşleme ve İletişim Uygulamaları*, İnönü Üniversitesi, Malatya, 16-19 May 2015, 271-274.
- [45] **Alcin, O. F., Sengur, A., Ghofrani, S., and Ince, M. C.**, 2014, GA-SELM: Greedy algorithms for sparse extreme learning machine, *Measurement*, **55**, 126-132.
- [46] **Alçin, Ö. F., Şengür, A. ve İnce, M. C.**, 2015, İleri-geri takip algoritması tabanlı seyrek aşırı öğrenme makinesi. *Journal of the Faculty of Engineering & Architecture of Gazi University*, **30(1)**, 126-132.

- [47] **Serre, D.**, 2002, *Matrices: Theory and Applications*, Springer New York Inc., New York.
- [48] **Haykin, S.**, 1999, *Neural Networks: A comprehensive foundation*, Prentice Hall, New Jersey.
- [49] **Huang, G. B., Chen, L. and Siew, C. K.**, 2006, Universal approximation using incremental feedforward networks with arbitrary input weights, *IEEE Transactions on Neural Networks*, **17(4)**, 879-892.
- [50] **Bartlett, P. L.**, 1997, For valid generalization, the size of the weights is more important than the size, *Advances in neural information processing systems*, **9**, 134.
- [51] **Cebeci, S.**, Alt küme bulma tabanlı ayırık optimizasyon problemleri için ayırık parçacık sürü optimizasyonu modelleri, *Yüksek Lisans Tezi*, Gebze Yüksek Teknoloji Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü, Gebze.
- [52] **Alataş, B.**, Kaotik haritalı parçacık sürü optimizasyonu algoritmaları geliştirme, *Doktora Tezi*, F. Ü. Fen Bilimleri Enstitüsü, Elazığ.
- [53] **Kaymaz, İ.**, 2015, Optimizasyon teknikleri, Atatürk Üniversitesi, http://194.27.49.11/makine/ikaymaz/optimizasyon/dosyalar/DERS_1_OPTIMIZASYONA_GIRIS.pdf, 02 haziran 2015.
- [54] **Akay, B.**, Nümerik optimizasyon problemlerinde yapay arı kolonisi algoritmasının performans analizi, *Doktora Tezi*, E.Ü. Fen Bilimleri Enstitüsü, Kayseri.
- [55] **Küçükkülahlı, E.**, Doğrusal programlama problemlerinin meta sezgisel yöntemlerle çözümü, *Yüksek Lisans Tezi*, D. Ü. Fen Bilimleri Enstitüsü, Düzce.
- [56] **Kaya, O.**, Kombinatoriyal optimizasyon problemlerinde bir sınıfının genetik algoritmaları ile çözümü üzerine, *Yüksek Lisans Tezi*, E. Ü. Fen Bilimleri Enstitüsü, İzmir.
- [57] **Rubinstein, R., Zibulevsky, M. and Elad, M.**, 2008, Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit, *CS Technion*, **40(8)**, 1-15.

- [58] **Karabulut, G. Z. Kurt, T. ve Yongacoglu, A.**, 2006, Sıralı taban seçim algoritmalarının iletişim problemlerine uygulamaları, *14. Sinyal İşleme ve İletişim Uygulamaları*, Antalya, Nisan, 1-4, 17-19.
- [59] **Rath, G. and Sahoo A.**, 2009, A comparative study of some greedy pursuit algorithms for sparse approximation, *17th European Signal Processing Conference (EUSIPCO 2009)*, Glasgow in Scotland, 24-28 August, 398-402.
- [60] **Blumensath, T. and Davies, M. E.**, 2009, How to use the iterative hard thresholding algorithm, *Signal Processing with Adaptive Sparse Structured Representations, (SPARS'09)*, Saint Malo in France, 1-20 March.
- [61] **Cevher, V. and Waters, A.**, 2014, The CoSamp Algorithm, Lecture Notes, <http://www.ece.rice.edu/~vc3/elec633/CoSaMP.pdf>, 01 Mayıs 2014.
- [62] **Karahanoglu, N. B. ve Erdogan, H.**, 2013, Compressed sensing signal recovery via forward-backward pursuit”, *Digital Signal Processing*, **23(5)**, 1539-1548.
- [63] **Karahanoglu, N. B. ve Erdogan, H.**, Optimal forward-backward pursuit for the sparse signal recovery problem, *21. Sinyal İşleme ve İletişim Uygulamaları*, Girne-KKTC, 24-26 Nisan, 1-4.
- [64] **Blumensath, T. and Davies, M. E.**, 2007, On the difference between orthogonal matching pursuit and orthogonal least squares, <http://www.personal.soton.ac.uk/tb1m08/papers/BDOMPvsOLS07.pdf>, 01 Temmuz 2015.
- [65] **Chen, S., Cowan, C. F. and Grant, P. M.**, 1991, Orthogonal least squares learning algorithm for radial basis function networks, *IEEE Transactions on Neural Networks*, **2(2)**, 302-309.
- [66] **Chen, S., Chng, E. S. and Alkadhim, K.**, 1996, Regularized orthogonal least squares algorithm for constructing radial basis function networks, *International Journal of Control*, **64(5)**, 829-837.
- [67] **Rebollo-Neira, L. and Lowe, D.**, 2002, Optimized orthogonal matching pursuit approach, *IEEE Signal Processing Letters*, **9(4)**, 137-140.
- [68] **Soussen, C., Gribonval, R., Idier, J. and Herzet, C.**, 2013, Joint k-step analysis of orthogonal matching pursuit and orthogonal least squares, *IEEE Transactions on Information Theory*, **59(5)**, 3158-3174.

- [69] **Holmes, S.**, RMSE Error, 2015, <http://statweb.stanford.edu/~susan/courses/s60/split/node60.html>, 01 Temmuz 2015.
- [70] https://en.wikipedia.org/wiki/Root-mean-square_deviation, Root-mean-square deviation, 01 Temmuz 2015.
- [71] https://tr.wikipedia.org/wiki/Standart_sapma, Standart sapma, 01 Temmuz 2015.
- [72] **Lane, D.**, Standard deviation and variance, 2015, <http://davidmlane.com/hyperstat/A16252.html>, 01 Temmuz 2015.
- [73] **İlhan, Ş. ve Carus, S.**, 2015, Korelasyon Analizi, <http://ormanweb.sdu.edu.tr/dersler/sarus/korelasyon.pdf>, 01 Temmuz 2015
- [74] <https://tr.wikipedia.org/wiki/Korelasyon>, Korelasyon, 01 Temmuz 2015.
- [75] **Willmott, C. J. and Matsuura, K.**, 2005, Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, *Climate research*, **30(1)**, 79.
- [76] **Chai, T. and Draxler, R. R.**, 2014, Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature, *Geoscientific Model Development*, **7(3)**, 1247-1250.
- [77] https://en.wikipedia.org/wiki/Mean_absolute_error, Mean absolute error, 01 Temmuz 2015.
- [78] MAPE <http://www.vanguardsw.com/business-forecasting-101/mean-absolute-percent-error-mape/>, Mean absolute percent error, 01 Temmuz 2015.
- [79] https://en.wikipedia.org/wiki/Mean_absolute_percentage_error, Mean absolute percent error, 01 Temmuz 2015.
- [80] **Özkan, Y.**, 2013, Veri madenciliği yöntemleri, Papatya yayıncılık, İstanbul.
- [81] **Frank, A., Asuncion, A.**, UCI Machine Learning Repository, 2011, <http://archive.ics.uci.edu/ml>, 11 Ocak 2011.
- [82] **Vlachos, P.**, StatLib Dataset Archive Carnegie Mellon University, 2005, <http://lib.stat.cmu.edu/datasets/>, 23 Şubat 2005.
- [83] MATLAB version 7.10.0. Natick, Massachusetts: The MathWorks Inc.
- [84] **Miche, Y., Bas, P., Jutten, C., Simula, O. and Lendasse, A.**, 2008, A Methodology for Building Regression Models using Extreme Learning Machine: OP-ELM, Proceedings of the 16th European Symposium on Artificial Neural Networks, Belgium, April, 247-252.

- [85] **Friedman, J., Hastie, T. and Tibshirani, R.**, 2010, Regularization paths for generalized linear models via coordinate descent, *Journal of statistical software*, **33(1)**, 1-22.
- [86] **Esen, H., Ozgen, F., Esen, M. and Sengur, A.**, 2009, Artificial neural network and wavelet neural network approaches for modelling of a solar air heater, *Expert systems with applications*, **36(8)**, 11240-11248.
- [87] **Li, J. and Liu, H.**, 2002, Kent ridge bio-medical data set repository, Inst. for Infocomm Research, <http://sdmc.lit.org.sg/GEDatasets/Datasets.html>, 01 Ocak 2002.
- [88] **Wang, X. and Han, M.**, 2014, Online sequential extreme learning machine with kernels for nonstationary time series prediction, *Neurocomputing*, **145**, 90-97.
- [89] **Zhao, P., Xing, L. and Yu, J.**, 2009, Chaotic time series prediction: From one to another, *Physics Letters A*, **373(25)**, 2174-2177.
- [90] **Li, D., Han, M. and Wang, J.**, 2012, Chaotic time series prediction based on a novel robust echo state network, *IEEE Transactions on Neural Networks and Learning Systems*, **23(5)**, 787-799.
- [91] **Kim, K. J.**, 2003, Financial time series forecasting using support vector machines *Neurocomputing*, **55(1)**, 307-319.
- [92] **Qin, P., Nishii, R. and Yang, Z. J.**, 2012, Selection of NARX models estimated using weighted least squares method via GIC-based method and l_1 -norm regularization methods, *Nonlinear Dynamics*, **70(3)**, 1831-1846.
- [93] **Yee, P. and Haykin, S.**, 1999, A dynamic regularized radial basis function network for nonlinear, nonstationary time series prediction, *IEEE Transactions on Signal Processing*, **47(9)**, 2503-2521.
- [94] **Ye, Y., Squartini, S. And Piazza, F.**, 2013, Online sequential extreme learning machine in nonstationary environments, *Neurocomputing*, **116**, 94-101.
- [95] **Takens, F.**, Detecting strange attractors in turbulence, Springer Berlin Heidelberg, Warwick, 1981.
- [96] **Heskes, T. M., Slijpen, E. T. and Kappen, B.**, 1992, Learning in neural networks with local minima, *Physical Review A*, **46(8)**, 5221.

- [97] **Wilamowski, B. M. and Yu, H.**, 2010, Neural network learning without backpropagation, *IEEE Transactions on Neural Networks*, **21(11)**, 1793-1803.
- [98] **Sibi, P., Jones, S. A. and Siddarth, P.**, 2013, Analysis of different activation functions using back propagation neural networks, *Journal of Theoretical and Applied Information Technology*, **47(3)**, 1264-1268.
- [99] **Lim, J. S., Lee, S. and Pang, H. S.**, 2013, Low complexity adaptive forgetting factor for online sequential extreme learning machine (OS-ELM) for application to nonstationary system estimations, *Neural Computing and Applications*, **22(3-4)**, 569-576.
- [100] <https://research.stlouisfed.org/fred2/series/DJIA/downloaddata>, DJIA time series, 01 Ocak 2014.
- [101] <http://www.mathworks.com/help/fuzzy/examples/chaotic-time-series-prediction.html> MacKey Glass time series, 01 Ocak 2014.
- [102] **Wang, N., Er, M. J. and Han, M.**, 2014, Parsimonious extreme learning machine using recursive orthogonal least squares, *IEEE Transactions on Neural Networks and Learning Systems*, **25(10)**, 1828-1841.
- [103] <http://www2.ocean.washington.edu/oc540/lec01-12/timeseries.m>, Ocean time series, 01 Ocak 2014.
- [104] http://www.esrl.noaa.gov/psd/gcos_wgsp/Timeseries/SUNSPOT/, Sun Spot time series, 01 Ocak 2014.

ÖZGEÇMİŞ

Doğum Tarihi	: 23/02/1983
Doğum Yeri	: Malatya
Lise	: Doğanşehir Çok Programlı Lise (1997–2000)
Lisans	: Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Öğretmenliği (2003–2007)
Yüksek Lisans	: Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim Dalı (2008-2011)
Doktora	: Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı
İş Tecrübesi	: Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümünde Araştırma Görevlisi (2009-Devam)
İlgi Alanları	: Makina Öğrenmesi, Seyreklik, Seyrek geri çatma