

T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ÇOK KATMANLI AŞIRI ÖĞRENME MAKİNELERİNİN REGRESYON
PROBLEMLERİNDEKİ BAŞARIMLARININ KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

MUHAMMED YILDIRIM

TEZ DANIŞMANI
PROF. DR. CİHAN KARAKUZU

BİLECİK, 2024

10597192

T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ÇOK KATMANLI AŞIRI ÖĞRENME MAKİNELERİNİN REGRESYON
PROBLEMLERİNDEKİ BAŞARIMLARININ KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

MUHAMMED YILDIRIM

TEZ DANIŞMANI
PROF. DR. CİHAN KARAKUZU

BİLECİK, 2024

10597192

BEYAN

“Çok Katmanlı Aşırı Öğrenme Makinelerinin Regresyon Problemlerindeki Başarılarının Karşılaştırılması” adlı yüksek lisans tezinin hazırlık ve yazımı sırasında bilimsel araştırma ve etik kurallarına uyduğumu, başkalarının eserlerinden yararlandığım bölümlerde bilimsel kurallara uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, tezin herhangi bir kısmının Bilecik Şeyh Edebali Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunulmadığımı, aksinin tespit edileceği muhtemel durumlarda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Bu çalışmanın, Bilimsel Araştırma Projeleri (BAP), TÜBİTAK veya benzeri kuruluşlarca desteklenmesi durumunda; projenin ve destekleyen kurumun adı proje numarası ile birlikte, ETİK KURUL onayı alınması durumunda ise ETİK KURUL tarih karar ve sayı bilgilerinin beyan edilmesi gerekmektedir.			
DESTEK ALINMIŞTIR		DESTEK ALINMAMIŞTIR	X
Destek alındı ise;			
Destekleyen kurum;			
Desteğin Türü		Proje Numarası	
1- BAP (Bilimsel Araştırma Projesi)			
2- TÜBİTAK			
Diğer;			
ETİK KURUL onayı var ise;			
ETİK KURUL karar tarih/sayı:	/	

Muhammed Yıldırım

Tarih

.....

İmza

.....

ÖN SÖZ

Tez çalışma dönemim boyunca engin bilgi ve tecrübeleriyle bana yol gösteren kıymetli danışman hocam Prof. Dr. Cihan KARAKUZU'ya saygı ve şükranlarımı sunuyorum.

Ayrıca manevi olarak her zaman desteklerini hissettiğim sevgili eşim Büşra YILDIRIM, kızlarım Nurten YILDIRIM ve Hümeysra YILDIRIM'a tüm kalbimle teşekkür ederim.

Muhammed Yıldırım

2024

ÖZET

ÇOK KATMANLI AŞIRI ÖĞRENME MAKİNELERİNİN REGRESYON PROBLEMLERİNDEKİ BAŞARIMLARININ KARŞILAŞTIRILMASI

Yapay zeka, günümüzdeki gelişmelerle birçok alanda yaygın olarak kullanılmaktadır. Yapay sinir ağları (YSA), büyük veri kümelerinin analizinde etkili olup, zaman ve çaba tasarrufu sağlayarak daha akıllı sistemlerin oluşturulmasına katkıda bulunur. Aşırı öğrenme makinesi (AÖM), YSA'ların eğitim sürecini basitleştirerek daha hızlı öğrenmelerini sağlar. Özellikle büyük veri kümeleri üzerinde çalışırken ortaya çıkan zorlukları aşmak için tasarlanmıştır. 2022'de Kale ve Karakuzu, çok katmanlı aşırı öğrenme makineleri üzerine iki önemli geliştirme yapmıştır. Araştırmacılar, bu ağların dinamik sistemlerdeki modelleme başarımını detaylı bir şekilde inceleyerek, önerilen sistemlerin üstün modelleme yeteneklerine sahip olduğunu belirlemiştir. Bulgular, çok katmanlı aşırı öğrenme makinelerinin sistem modelleme uygulamalarında etkili araçlar olduğunu net bir şekilde göstermektedir. Bu tez çalışmasında, yeni geliştirilen bu iki AÖM yapısının regresyon problemleri üzerindeki başarımları incelenmiştir. Bulgular, yeni geliştirilmiş modellerinin regresyon problemlerini çözmede etkili bir başarımlar sergilediğini ve orijinal ağ yapısına kıyasla daha başarılı sonuçlar verdiğini göstermektedir. Ancak, işlem yapma hızı açısından her durumda üstünlük sağlamadığı gözlemlenmiştir. Elde edilen sonuçlar, modellerin regresyon analizinde geniş bir uygulama potansiyeli olduğunu ve endüstriyel/bilimsel problemlerin çözümünde değerli bir araç olabileceğini işaret etmektedir.

Anahtar kelimeler: Yapay öğrenme, Yapay sinir ağları, Çok katmanlı aşırı öğrenme makinesi, Regresyon

ABSTRACT

COMPARISON OF PERFORMANCE OF MULTILAYER EXTREME LEARNING MACHINES IN REGRESSION PROBLEMS

Artificial intelligence is widely used in many fields with today's developments. Artificial neural networks (ANN) are effective in analyzing large data sets and contribute to the creation of smarter systems by saving time and effort. Extreme learning machine (ELM) allows ANNs to learn faster by simplifying the training process. It is specifically designed to overcome the challenges that arise when working on large data sets. In 2022, Kale and Karakuzu made two important developments on multi-layer extreme learning machines. By examining in detail the modeling performance of these networks in dynamic systems, researchers have determined that the proposed systems have superior modeling capabilities. The findings clearly show that multilayer extreme learning machines are effective in system modeling applications. In this thesis study, the success of these two newly developed ELM structures on regression problems was examined. The findings show that the newly developed models are effective in solving regression problems and provide more successful results than the original network structure. However, it has been observed that it is not advantageous in all cases in terms of processing speed. The results obtained indicate that the models have a wide application potential in regression analysis and can be a valuable tool in solving industrial/scientific problems.

Keywords: Artificial learning, Artificial neural networks, Multilayered extreme learning machine, Regression

İÇİNDEKİLER

	Sayfa
ÖN SÖZ.....	i
ÖZET.....	ii
ABSTRACT.....	iii
İÇİNDEKİLER.....	iv
TABLolar LİSTESİ.....	v
ŞEKİLLER LİSTESİ.....	vi
KISALTMALAR VE SİMGELER LİSTESİ.....	ix
1. GİRİŞ.....	1
2. YAPAY SİNİR AĞLARI.....	6
2.1. Tek Katmanlı Yapay Sinir Ağı.....	8
2.2. Çok Katmanlı Yapay Sinir Ağı.....	9
3. AŞIRI ÖĞRENME MAKİNESİ.....	11
3.1. Çok Katmanlı Aşırı Öğrenme Makinesi.....	12
3.2. Geliştirilmiş Çok Katmanlı Aşırı Öğrenme Makineleri.....	14
4. ÇK-AÖM'LERİN REGRESYON PROBLEMLERİNDE BAŞARIMI.....	17
4.1. Regresyon.....	17
4.2. Veri Kümeleri.....	18
4.3. Yöntem.....	20
4.4. Bulgular.....	22
5. SONUÇLAR VE ÖNERİLER.....	49
KAYNAKÇA.....	50

TABLULAR LİSTESİ

	Sayfa
Tablo 4.1. Veri kümeleri.....	19
Tablo 4.2. Orijinal ÇK-AÖM'nin saniye cinsinden işlem yapma zamanı.....	39
Tablo 4.3. GÇK-AÖM1'in saniye cinsinden işlem yapma zamanı.....	39
Tablo 4.4. GÇK-AÖM2'nin saniye cinsinden işlem yapma zamanı.....	40
Tablo 4.5. Orijinal ÇK-AÖM'nin eğitim fazındaki metrikleri.....	40
Tablo 4.6. GÇK-AÖM1'in eğitim fazındaki metrikleri.....	41
Tablo 4.7. GÇK-AÖM2'nin eğitim fazındaki metrikleri.....	41
Tablo 4.8. Orijinal ÇK-AÖM'nin test fazındaki metrikleri.....	42
Tablo 4.9. GÇK-AÖM1'in test fazındaki metrikleri.....	42
Tablo 4.10. GÇK-AÖM2'nin test fazındaki metrikleri.....	43
Tablo 4.11. GÇK-AÖM1'in orijinale göre eğitim fazındaki metriklerinin yüzdelerik değişimi.	43
Tablo 4.12. GÇK-AÖM1'in orijinale göre test fazındaki metriklerinin yüzdelerik değişimi.....	44
Tablo 4.13. GÇK-AÖM1'in orijinale göre işlem yapma zamanının yüzdelerik değişimi.....	44
Tablo 4.14. GÇK-AÖM2'nin orijinale göre eğitim fazındaki metriklerinin yüzdelerik değişimi	45
Tablo 4.15. GÇK-AÖM2'nin orijinale göre test fazındaki metriklerinin yüzdelerik değişimi...	45
Tablo 4.16. GÇK-AÖM2'in orijinale göre işlem yapma zamanının yüzdelerik değişimi.....	46

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 2.1. Yapay nöron.....	7
Şekil 2.2. Tek katmanlı YSA.....	9
Şekil 2.3. Çok katmanlı YSA.....	10
Şekil 3.1. ÇK-AÖM'nin eğitim sırasında (üst) ve eğitim sonrasında (alt) kullanım mimarisi (d harici giriş ve δ harici çıkışlı m katmanlı).....	14
Şekil 3.2. M gizli katman için GÇK-AÖM1 şeması.....	15
Şekil 3.3. M gizli katman için GÇK-AÖM2 şeması.....	15
Şekil 4.1. ÇK-AÖM'nin Abalone veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	23
Şekil 4.2. ÇK-AÖM'nin Auto-Mpg veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	23
Şekil 4.3. ÇK-AÖM'nin Auto Imports Price veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	24
Şekil 4.4. ÇK-AÖM'nin Boston Housing veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	24
Şekil 4.5. ÇK-AÖM'nin Relative CPU Performance veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	25
Şekil 4.6. ÇK-AÖM'nin Delta Ailerons veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	25
Şekil 4.7. ÇK-AÖM'nin Delta Elevators veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	26
Şekil 4.8. ÇK-AÖM'nin Servo veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	26
Şekil 4.9. ÇK-AÖM'nin İstanbul Stock Exchange veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	27
Şekil 4.10. ÇK-AÖM'nin QSAR fish toxicity veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	27
Şekil 4.11. GÇK-AÖM1'in Abalone veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	28

Şekil 4.12. GÇK-AÖM1'in Auto-Mpg veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	28
Şekil 4.13. GÇK-AÖM1'in Auto Imports Price veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	29
Şekil 4.14. GÇK-AÖM1'in Boston Housing veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	29
Şekil 4.15. GÇK-AÖM1'in Relative CPU Performance veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	30
Şekil 4.16. GÇK-AÖM1'in Delta Ailerons veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	30
Şekil 4.17. GÇK-AÖM1'in Delta Elevators veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	31
Şekil 4.18. GÇK-AÖM1'in Servo veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	32
Şekil 4.19. GÇK-AÖM1'in İstanbul Stock Exchange veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	32
Şekil 4.20. GÇK-AÖM1'in QSAR fish toxicity veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	32
Şekil 4.21. GÇK-AÖM2'nin Abalone veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	33
Şekil 4.22. GÇK-AÖM2'nin Auto-Mpg veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	33
Şekil 4.23. GÇK-AÖM2'nin Auto Imports Price veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	34
Şekil 4.24. GÇK-AÖM2'nin Boston Housing veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	34
Şekil 4.25. GÇK-AÖM2'nin Relative CPU Performance veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	35
Şekil 4.26. GÇK-AÖM2'nin Delta Ailerons veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	35
Şekil 4.27. GÇK-AÖM2'nin Delta Elevators veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	36

Şekil 4.28. GÇK-AÖM2'nin Servo veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	36
Şekil 4.29. GÇK-AÖM2'nin İstanbul Stock Exchange veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	37
Şekil 4.30. GÇK-AÖM2'nin QSAR fish toxicity veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları.....	37
Şekil 4.31. GÇK-AÖM'lerin öğrenme başarımları.....	47

KISALTMALAR VE SİMGELER LİSTESİ

β : Çıkış Ağırlık Vektörü

ϵ : Hata

μ : Aritmetik Ortalama

σ : Standart Sapma

\tilde{N} : Hücre Adedi

AÖM: Aşırı Öğrenme Makinesi

CPU: Central Processing Unit

ÇK-AÖM: Çok Katmanlı Aşırı Öğrenme Makinesi

GÇK-AÖM: Geliştirilmiş Çok Katmanlı Aşırı Öğrenme Makinesi

RMSE: Root Mean Square Error

SLFN: Tek Katmanlı İleri Beslemeli Yapay Sinir Ağı

UCI: University of California, Irvine

YSA: Yapay Sinir Ağı

1. GİRİŞ

Günümüzde yapay zeka giderek daha fazla alanda kullanılan bir teknoloji haline gelmiştir. Bu yükseliş, özellikle makine öğrenimi ve derin öğrenme alanlarındaki algoritmaların büyük bir gelişme kaydetmesiyle mümkün olmuştur. Makine öğrenimi, bilgisayar sistemlerinin öğrenerek karmaşık görevleri başarıyla gerçekleştirebilen modellerin oluşturulmasına olanak tanır. Derin öğrenme ise bu modellerin daha karmaşık ve katmanlı yapılara sahip olmalarını sağlar. Yapay zekanın bu alanlarda sağladığı üstünlükler, özellikle büyük veri kümelerinin analizinde kendini gösterir. Bu algoritmalar, büyük miktardaki veriyi hızlı bir şekilde işleyerek anlamlı desenleri keşfetme yeteneğine sahiptir. Ayrıca, yapay zeka modelleri, karmaşık algoritmalar ve optimizasyon teknikleri kullanarak bu veri kümelerinden anlamlar çıkarabilir. Yapay zeka, birçok sektörde insanlar için zaman ve çaba tasarrufu sağlayarak, daha akıllı ve verimli sistemlerin oluşturulmasına olanak tanımaktadır.

Yapay sinir ağları (YSA), insan beyninin sinir hücrelerinden ilham alarak geliştirilen matematiksel modellerdir. Bu modeller, bilgisayar sistemlerine öğrenme ve problem çözme yetenekleri kazandırmak amacıyla kullanılır. YSA'lar, karmaşık veri kümelerinden desenler çıkarmak ve bu desenlere dayalı öngörülerde bulunmak için kullanılan güçlü araçlardır.

2004 yılında Huang ve ekibi tarafından öne sürülen bir yöntem, YSA'ların eğitim sürecini basitleştirerek çok daha hızlı öğrenmelerini sağlamıştır. Bu yenilikçi yaklaşım, aşırı öğrenme makinesi (AÖM) olarak adlandırılmıştır. AÖM, YSA'ların öğrenme sürecindeki verimliliği artırmak ve daha hızlı sonuçlar elde etmek için tasarlanmış bir metodoloji sunar. AÖM, YSA'ların eğitim sürecindeki karmaşıklığı azaltarak, modelin daha hızlı ve daha etkili bir şekilde öğrenmesine imkan tanır. Bu şekilde, özellikle büyük veri kümeleri üzerinde çalışırken ortaya çıkan zorlukları aşmada önemli bir avantaj sağlar.

YSA'lar, birçok uygulama alanında kullanılmakta olup, görüntü tanıma, ses analizi, dil işleme gibi konularda önemli başarılar elde etmiştir. Aşırı öğrenme makinesi, bu başarıları daha da artırarak yapay sinir ağlarının genel performansını iyileştirme potansiyeli taşımaktadır. Bu gelişmeler, bilgisayar sistemlerinin daha karmaşık görevlere daha etkili bir şekilde adapte olmasını sağlayarak, yapay zeka alanında büyük bir ilerlemenin kapılarını aralamaktadır.

2022 yılında Kale ve Karakuzu tarafından çok katmanlı aşırı öğrenme makineleri konusunda önemli iki geliştirme gerçekleştirilmiştir. Bu yeni ağ mimarileri, nöron

aktivasyonlarını hem eğitim sırasında hem de sonrasında kullanarak dikkat çekmektedir. Araştırmacılar, bu ağların dinamik sistemler üzerindeki modelleme başarımını detaylı bir şekilde inceleyerek, önerilen sistemlerin daha üstün modelleme yeteneklerine sahip olduğunu belirlemişlerdir. Elde edilen bulgular, geliştirilen çok katmanlı aşırı öğrenme makinelerinin sistem modelleme uygulamaları için etkili araçlar olduğunu net bir şekilde göstermektedir.

Bu çalışmada, Kale ve Karakuzu tarafından geliştirilen ve modelleme başarımı araştırılan çok katmanlı aşırı öğrenme makineleri, regresyon problemi içeren veri kümeleri üzerinde denenmiştir. Çalışmanın özgün değeri henüz yeni geliştirilmiş iki farklı çok katmanlı AÖM öğrenme yönteminin regresyon problemlerinde ilk kez kullanılması ve başarımlarının analiz edilmesi olmuştur.

1.1. Literatür Taraması

Geleneksel Yapay Sinir Ağı (YSA) eğitiminde sıkça kullanılan türeve dayalı öğrenme yöntemleri, başarılı sonuçlar elde etmesine rağmen öğrenme süreci oldukça zaman alıcıdır. Huang ve arkadaşları tarafından 2004 yılında geliştirilen AÖM yöntemi, analitik bir hesaplama yaklaşımı benimseyerek kısa sürede hızlı bir eğitim sağlayabilmektedir. AÖM, son derece hızlı çözümler sunması nedeniyle birçok araştırmacı, çeşitli problemler için AÖM'yi tercih etmiştir.

"On-Line Sequential Extreme Learning Machine" başlıklı makalede, Çevrimiçi Sıralı Aşırı Öğrenme Makinesi (OS-ELM) olarak adlandırılan öz yinelenmeli en küçük kareler (RLS) algoritmasına dayanan bir sıralı modifikasyon sunulmuştur (G.-B. Huang vd.,2005).

AÖM'de giriş katmanındaki ağırlık parametreleri ve eşik parametreleri rastgele belirlenirken, "Evolutionary Extreme Learning Machine" başlıklı makalede karma öğrenme algoritması geliştirilmiştir. Bu yöntem, rastgele atanmış bu parametreleri seçmek için farksal gelişim algoritmasını ve çıktı ağırlıklarını analitik olarak belirlemek için Moore-Penrose yöntemini kullanan bir karma öğrenme algoritması önermektedir (Zhu vd., 2005). Bu çalışmanın deneysel sonuçlarına dayanarak, bu yaklaşımın çok daha kompakt ağlarla iyi genelleme başarımı sağlayabildiği rapor edilmiştir.

"Online Sequential Fuzzy Extreme Learning Machine for Function Approximation and Classification Problems" başlıklı çalışmada, fonksiyon yaklaşımı ve sınıflandırma problemleri için çevrimiçi sıralı bulanık aşırı öğrenme makinesi (OS-Fuzzy-ELM) geliştirilmiştir (Rong vd., 2009).

2010 yılında Lan ve arkadaşları, "Constructive Hidden Nodes Selection of ELM" başlıklı çalışmalarında, AÖM algoritmasına dayalı bir yapıcı yöntem uygulayarak AÖM regresörünün mimari tasarımını ele almışlardır. Önerilen CS-ELM'in diğer AÖM ağları (AÖM, EM-ELM ve I-ELM) ile karşılaştırılması, bazı gerçek kıyaslamalı regresyon uygulamalarında gerçekleştirilmiştir (Lan vd., 2010a).

2010 yılında Lan ve arkadaşları tarafından yayınlanan "Two-stage Extreme Learning Machine for Regression" çalışması, AÖM'nin eksik kalan ön ağ yapısının belirlenmesine odaklanmıştır. TS-ELM adı verilen yöntem, iki aşamalı bir süreçten oluşmaktadır. İlk aşamada, ileriye dönük öz yinelemeli algoritma her adımda rastgele üretilen bireylerden gizli düğümü seçmek için uygulanır ve düğümler durdurma kriteri minimum değere gelene kadar ağa eklenir. İkinci aşamada ise, her gizli düğümün önemi değerlendirilir ve önemsiz olanlar ağdan çıkarılır, bu da ağın karmaşıklığını azaltır. TS-ELM, denektaşı regresyon problemleri için AÖM ve EM-ELM ile karşılaştırılmıştır. Sonuçlar TS-ELM'in AÖM'ye kıyasla daha iyi performans gösterdiğini, aynı zamanda EM-ELM ile benzer veya daha iyi sonuçlar elde ettiğini göstermektedir (Lan vd., 2010b).

"Fuzzy Extreme Learning Machine for Classification" başlıklı çalışmada, FELM yapısı önerilmiştir. Bu öneri, geleneksel AÖM yapısındaki dengesizlik problemleri ve ağırlıklı sınıflandırma problemleri gibi farklı girdi noktaları sınıflardan birine tam olarak atanamama eksikliğini gidermektedir. Bu yapı, bulanık mantık kavramının geleneksel AÖM yöntemine dahil edildiği bir model olarak değerlendirilebilir. Ağırlıklı sınıflandırma problemleri için FELM'in, deneysel sonuçlarla gösterildiği üzere, AÖM'den daha mantıklı sonuçlar verdiği belirtilmiştir (W. B. Zhang ve Ji, 2013).

"Sparse Bayesian Extreme Learning Machine for Multi-classification" başlıklı çalışmada, AÖM'nin sınıflandırma problemlerindeki çıkış ağırlıklarının öğrenilmesi için seyrek bayesci yaklaşım sunulmuştur. Seyrek Bayesci AÖM (SBAÖM) olarak adlandırılan bu metotta, ağın çıktılarının marjinal olasılığı hesaplanmakta ve öğrenme sürecinde gereksiz birçok gizli nöronun otomatik olarak kaldırılmasıyla ağın performansı artırılmaktadır. Ek olarak, eğitim hatasını minimize etmek yerine marjinal olasılığı artırmak suretiyle, SBAÖM modeli eğitim verisine aşırı uyum sağlanmasını önlemektedir (Luo vd., 2014).

"Aşırı öğrenme makinelerinin seyrek geri çatma algoritmaları ile optimizasyonu" başlıklı doktora tezinde Yinelemeli Sert Eşikleme (YSE) algoritmasının en uygun gizli

katman nöron sayısını belirleme ve eğitim süreci karmaşıklığı gibi konular, AÖM ve düzenlenmiş AÖM yöntemleriyle karşılaştırılarak incelenmiştir. AÖM'de olduğu gibi, YSE-AÖM yönteminde de giriş katmanından gizli katmana bağlanan giriş ağırlıkları ve eşikler rastgele seçilir. Daha sonra, çıkış katmanı ağırlıkları YSE algoritması kullanılarak hesaplanır ve bu şekilde seyreklik sayesinde ilgisiz nöronlar otomatik olarak elenerek en uygun nöron sayısına sahip ağ elde edilir (Alçin, 2015).

AÖM'ye yönelik geliştirmelerden biri, iki gizli katmanlı AÖM olan TELM'dir. Bu modelde aynı sayıda nörona sahip 2 gizli katmanı vardır. AÖM'de olduğu gibi, giriş katmanı ile ilk gizli katman arasındaki bağlantılar rastgele ağırlıklandırılır, eşikler de rastgele ağırlıklandırılır. Bu başlatma ortogonal bir başlatmadır, çünkü bu ortogonalleştirmenin sınıflandırma problemlerinde daha iyi performans ürettiği gözlemlenmiştir. TELM, 1. ve 2. gizli katman arasındaki bağlantı ağırlıkları ve eşik olan 2. gizli katmanın parametrelerini elde etmek için yeni bir yöntem sunmuştur. Rastgele başlatılan ağırlıkların ve eşiklerin ortogonallığı, regresyon ve bazı sınıflandırma problemlerinde yapılan deneylerin gösterdiği gibi, başarımlar ve sonuçların iyileştirilmesinde bir rol oynamış olabilir (Qu vd., 2016).

2018 yılında, AÖM'nin girdi ağırlıklarının ve gizli nöronlarının değerlerini optimize etmek için Competitive Swarm Optimizer'ı (CSO) kullanan yeni bir model önerilmiştir. Modelin amacı genelleme başarımlarını arttırmak, sınıflandırıcıyı kararlı kılmak ve gizli katmandaki nöron sayısını azaltarak daha kompakt ağlar üretmektir. Önerilen model 15 tıbbi sınıflandırma problemine dayalı olarak denenmiştir. Deneysel sonuçlar, önerilen modelin daha az sayıda gizli nöron ve daha yüksek kararlılık ile daha iyi genelleme başarımlarını elde edebileceğini göstermektedir (Eshtay vd., 2018).

“A Novel Human Activity Recognition Scheme for Smart Health Using Multilayer Extreme Learning Machine” başlıklı çalışmada, geliştirilen şemada, tanımlama doğruluğu, doğrusal olmayan ve yerel bir benzerlik ölçüsünün, yani çekirdek riskine duyarlı kaybın, yığılmış aşırı öğrenme makinesi olarak adlandırılan yeni bir çok katmanlı sinir ağı öğrenme algoritmasına dahil edilmesiyle artırılmaktadır (Chen vd., 2018).

Zhang ve arkadaşları, yığılı AÖM otomatik kodlayıcı (ELM-AE), artık AÖM ve yerel alıcı alan tabanlı AÖM (ELM-LRF) dahil olmak üzere ÇK-AÖM'lerin geliştirilmesine ilişkin kapsamlı bir inceleme gerçekleştirdi ve bunların uygulamalarını ele aldılar. Ayrıca rastgele

sinir ađları ile geleneksel derin öğrenme arasındaki bađlantıyı da arařtırdılar (Zhang vd., 2020).

Yapılan literatür arařtırması analizi sonucunda, AÖM'nin makine öğrenimi alanında diđer algoritmalara göre öne çıkan bazı avantajlara sahip olduđu gözlemlenmiřtir. AÖM, verilerin daha hızlı bir eğitim sürecinden geçmesi ve yüksek genelleme başarımı sunmasıyla öne çıkmaktadır. Sunulduđu yıldan itibaren sürekli olarak geliştirilen AÖM, küçük ölçekli, orta ölçekli ve büyük ölçekli veri kümeleriyle yapılan çalışmalarda başarıyla kullanılmıřtır.

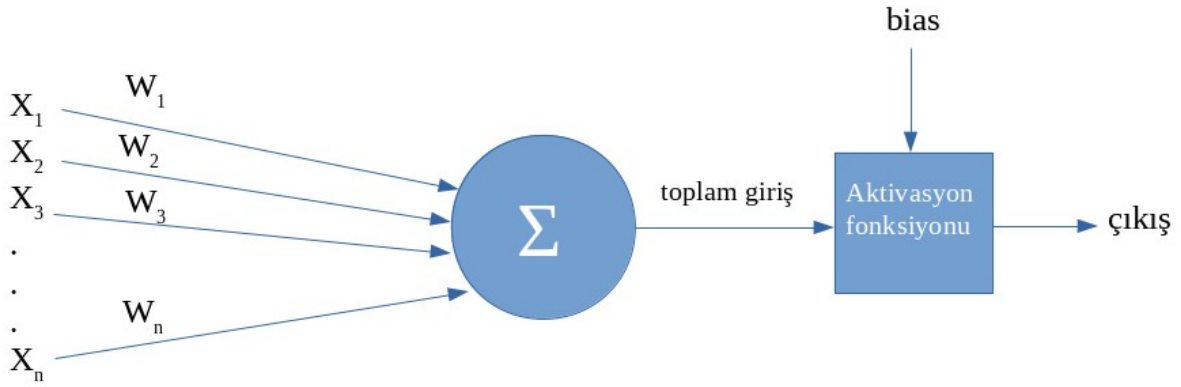
2. YAPAY SİNİR AĞLARI

İnsan beyni, karmaşıklığı ve etkileyici işlevselliğiyle bilim dünyasının büyük bir hayranlık kaynağıdır. Beyin, milyarlarca nöron adını verdiğimiz özelleşmiş hücrelerin entegre bir ağı olarak tasvir edilebilir. Nöronlar, birbirleriyle karmaşık bir şekilde iletişim kurarak çeşitli zihinsel işlevleri gerçekleştirirler. Nöronlar arasındaki iletişim, elektriksel sinyallerin gönderilip alınmasıyla gerçekleşir. Bu elektriksel sinyaller, nöronlardan birinin diğerine bilgi iletmek için kullandığı bir tür dil gibidir. Beyin, bu karmaşık iletişim ağı sayesinde düşünme, öğrenme, hafıza oluşturma ve duygusal tepkiler gibi bir dizi karmaşık süreci gerçekleştirir.

YSA, biyolojik bir beyinde bulunan nöronları modelleyen bir yapay zeka yaklaşımıdır. İnsan beyni, YSA mimarisinin temel ilham kaynağıdır. İnsan beyninde bulunan nöronlar, son derece karmaşık bir yapı oluşturarak birbirleriyle etkileşime girerler ve bilgi işleme süreçlerini idame ettirirler. Yapay sinir ağları da benzer bir yaklaşımı takip eder (Haykin S, 1999). Bu ağlar, bir problemi çözmek veya belirli bir görevi gerçekleştirmek için bir araya gelen yapay nöronlardan oluşur. Her bir yapay nöron, bir yazılım modülü gibi düşünülebilecek düğümleri temsil eder ve bilgi işleme süreçlerini yönetir. YSA'lar, birbirine bağlı birimlerden veya düğümlerden oluşan yapay nöronlar üzerine kuruludur. Bu yapay nöronlar, biyolojik nöronların temel işlevlerini taklit etmek üzere tasarlanmıştır. Her bir bağlantı, biyolojik beyindeki sinapsları anımsatan şekilde, diğer nöronlara sinyal iletebilir. Bu yapay nöronlar arasındaki etkileşim, önceden belirlenmiş algoritmalar veya matematiksel hesaplamalar aracılığıyla gerçekleşir. YSA temellerini atan ilk çalışma, "A Logical Calculus of Ideas Immanent in Nervous Activity" adlı makaledir (McCulloch ve Pitts, 1943). Bu makalede, nöronların basit matematiksel modelleri tanımlanarak ve bu modeller bağlantılar aracılığıyla bir araya getirilerek, karmaşık bilişsel işlevler simüle edilmiştir. 1986'da "Learning Representations by Back-Propagating Errors" (Hataları Geri Yayararak Temsilleri Öğrenme) adlı çalışma (Rumelhart vd., 1986) ile yeni bir öğrenme yöntemi geliştirildi. Geri yayılım, ağın çıkışındaki hata ile başlayarak, ağın içindeki katmanlara doğru hataların geriye doğru yayılmasına dayanır. Bu sayede, her bir katmandaki ağırlıkların ve bağlantıların güncellenmesi sağlanır, böylece ağın öğrenmesi ve temsil gücü artırılır.

YSA'lar, öğrenme süreçleri sırasında ağırlıkları ayarlayarak ve veri kümelerini analiz ederek görevlerini yerine getirirler. Bu, insan beyninin öğrenme ve bilgi işleme yeteneklerini taklit etmeye çalışan bir yaklaşımdır. Bu nedenle, yapay sinir ağları, biyolojik sistemlerden

ilham alarak çeşitli karmaşık görevleri başarıyla gerçekleştirebilirler. Yapay bir nöron, gelen sinyalleri alır, bunları işler ve ardından kendisine bağlı nöronlara sinyal gönderir. Bu sinyaller genellikle gerçel sayılar olarak ifade edilirler ve nöronun çıkışı, girişlerinin toplamının doğrusal olmayan bir fonksiyonuyla hesaplanır. Nöronlar öğrenme süreci ilerledikçe ayarlanan ağırlıklara sahiptir. Ağırlıklar, bir bağlantıdaki sinyalin gücünü artırmak veya azaltmak amacıyla kullanılır. Her bir yapay nöronun, sinyal göndermek için belirli bir eşiği vardır. Yani, nöronun çıkışı, toplam sinyalin bu eşik değerini aşması durumunda gerçekleşir. Bu mekanizma, biyolojik nöronların eşik değerine benzer bir şekilde çalışır.



Şekil 2.1. Yapay nöron

Şekil 2.1.'de yapay sinir ağını oluşturan bir nöronun yapısı gösterilmiştir. Burada $X_1, X_2, X_3, \dots, X_n$ giriş düğümleridir. Her bir giriş kendisinin bağlantı ağırlığı ($W_1, W_2, W_3, \dots, W_n$) ile çarpılarak toplanır. Yani ağırlıklandırılmış girişlerin toplamı oluşturulur. Bu şekilde elde edilen toplam girişe bias yani eşik eklenir. Girişler toplamının eşik eklenmiş haline net giriş adı verilir. Net giriş aktivasyon fonksiyonundan geçirilerek çıkış hesaplanır (Öztemel, 2003).

YSA'lar, çeşitli sektörlerde başarıyla uygulanan çok yönlü araçlar haline gelmiştir. Tıbbi görüntü sınıflandırması ve tıbbi teşhis alanında kullanıldığında, karmaşık görüntü verilerini analiz ederek hastalıkların tanısında ve tedavisinde kritik bir rol oynar. Bu, hastaların daha hızlı ve kesin bir şekilde teşhis edilmesine olanak tanır ve tedavi planlarını daha kişiselleştirmeyi mümkün kılar.

Pazarlama alanında büyük veri kümelerini değerlendirerek kullanıcı davranışlarının anlaşılmasına yönelik kullanıldığında, kullanıcılara daha özelleştirilmiş içerik sunma ve reklam stratejilerini optimize etme potansiyeli sunar. Yapay sinir ağları, sosyal medya

platformlarından elde edilen geniş veri kümelerini analiz ederek kullanıcı eğilimlerini belirleme ve bu bilgileri pazarlama stratejilerini şekillendirmek için kullanma yeteneği açısından da önemli bir rol oynar.

Finansal araçların geçmiş verilerinin işlenmesiyle finansal tahminler için kullanıldığında YSA'ların ekonomik değişkenleri analiz ederek gelecekteki finansal eğilimleri öngörmelerine olanak tanır. Bu, yatırımcılara güvenilir ve bilinçli yatırım kararları alma konusunda yardımcı olacaktır. Yapay sinir ağları, finansal piyasalardaki karmaşık ilişkileri ve değişkenleri anlama kapasitesi ile finans sektöründe etkili bir araç olarak kullanılabilir.

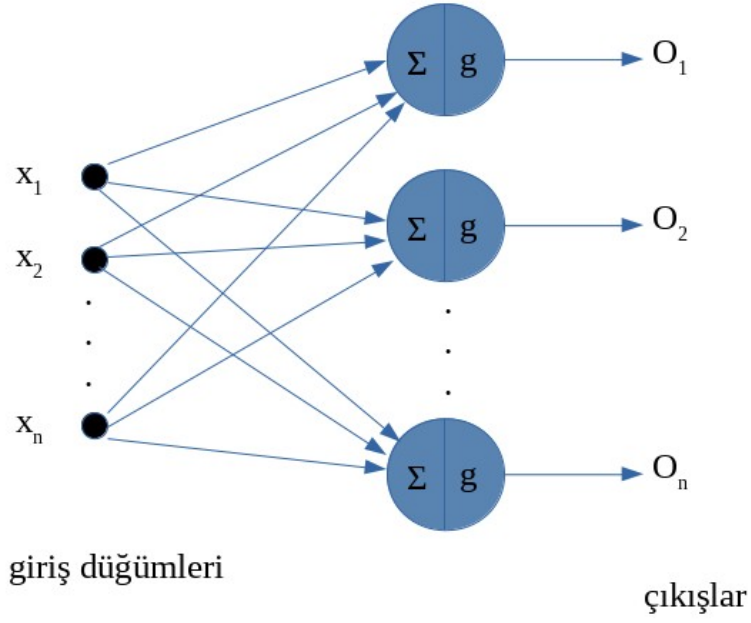
YSA'lar enerji tüketim verilerini analiz ederek gelecekteki elektrik yükü ve enerji talebi tahmini için kullanılabilir. Bu analiz, enerji şirketlerine kaynaklarını daha etkili bir şekilde yönetme ve enerji taleplerine daha hızlı ve doğru bir şekilde yanıt verme konusunda stratejik bir avantaj sağlar. Ayrıca, enerji verimliliği ve sürdürülebilirlik hedeflerine ulaşmada önemli bir rol oynar.

YSA'ların üretim süreçlerini izlemesi hataları tespit etmelerine ve kaliteyi artırmaya olanak tanır. Bu, üretim endüstrisinde operasyonel verimliliği en üst düzeye taşıma ve kalite standartlarını koruma çabalarında önemli bir katkı sağlar. Yapay sinir ağları, üretim hatlarındaki karmaşık ilişkileri analiz ederek süreçlerin optimize edilmesine ve hataların en aza indirilmesine yönelik stratejiler geliştirebilir.

Bu örnekler, YSA'nın geniş bir uygulama yelpazesi olduğunu göstermektedir ve bu teknolojinin birçok sektörde veri analizi, tahmin ve karar destek süreçlerinde önemli bir rol oynadığını göstermektedir.

2.1. Tek Katmanlı Yapay Sinir Ağı

Tek katmanlı yapay sinir ağı, yalnızca bir giriş katmanı, bir çıkış katmanı ve bu iki katman arasında bağlantıları içeren basit bir yapay sinir ağı türüdür. Şekil 2.2'de örnek bir tek katmanlı YSA gösterilmiştir.

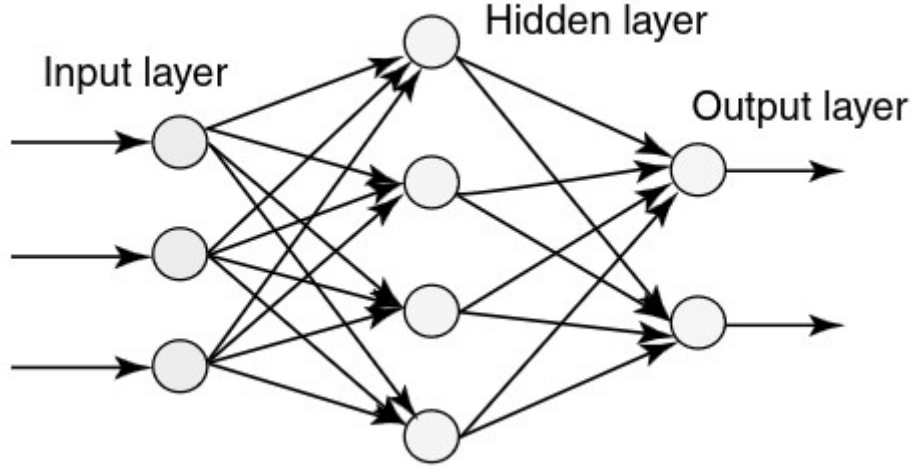


Şekil 2.2. Tek katmanlı YSA

Tek katmanlı sinir ağları, özellikle basit sınıflandırma görevleri gibi temel problemleri çözmek için kullanılabilirler. Daha karmaşık problemleri çözmek için genellikle çok katmanlı yapay sinir ağlarına ihtiyaç duyulur. Çok katmanlı yapay sinir ağları, daha fazla esneklik ve öğrenme kapasitesi sunar.

2.2. Çok Katmanlı Yapay Sinir Ağı

YSA'lar genellikle katmanlar halinde düzenlenir. Farklı katmanlar, girdiler üzerinde farklı dönüşümler gerçekleştirebilir. Sinyaller, katmanları birkaç kez geçtikten sonra, ilk katmandan (giriş katmanı) başlayarak son katmana (çıkış katmanı) doğru hareket eder. Bu katmanlı yapı, daha karmaşık örüntüleri ve ilişkileri öğrenme yeteneğini artırır. Şekil 2.3'de örnek bir çok katmanlı YSA ağı gösterilmiştir.



Şekil 2.3. Çok katmanlı YSA
Kaynak: (Abraham, 2005:902)

Çok katmanlı yapay sinir ağları üç ana kısma ayrılır: giriş katmanı, gizli katmanlar ve çıkış katmanı.

Giriş Katmanı: Giriş verilerini alan ilk katmandır. Her bir giriş özelliği bir düğümle temsil edilir, bu düğümler giriş verilerini ağa aktarır.

Gizli Katmanlar: Çok katmanlı YSA'lar çok sayıda gizli katmana ve her katmanda farklı sayıda nörona sahip olabilir. Ancak çoğu durumda bir veya iki katmana sahip olmak yeterlidir. Giriş katmanından çıkış katmanına kadar olan hesaplamalar bu katmanlarda gerçekleşir. Bu katmanlarda bulunan her bir düğüm, önceki katmandaki tüm düğümlerle bağlantılıdır. Her bağlantıda bir ağırlık bulunur ve her düğümde bir aktivasyon fonksiyonu uygulanır.

Çıkış Katmanı: Ağın son katmanıdır, problemin türüne bağlı olarak farklı türlerde aktivasyon fonksiyonları içerebilir.

Gizli katmanların eklenmesi çok katmanlı YSA'ların daha karmaşık işlevleri öğrenebilme yeteneğini artırır. Bu yapı, doğrusal olarak ayrılamayan ilişkileri modelleyebilir ve daha geniş bir veri yelpazesini kapsayabilir. Ayrıca, bu ağlar, geri yayılım algoritması gibi öğrenme algoritmalarını kullanarak ağırlıkları ve eşikleri ayarlayarak üzerinde çalışılan sorunun çözümünü öğrenebilirler.

3. AŞIRI ÖĞRENME MAKİNESİ

Aşırı Öğrenme Makinesi (AÖM), tek katmanlı ileri beslemeli YSA'lar için geleneksel eğitim yöntemlerine alternatif olarak önerilen bir öğrenme yöntemidir. Bu yaklaşım, 2004 yılında "Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks" başlıklı bir makalede (G.-B. Huang vd., 2004) tanıtılmıştır.

Geleneksel YSA'larda, giriş ve gizli katmanları birbirine bağlayan ağırlıklar küçük rastgele değerlerle başlatılır ve eğitim sırasında ayarlanır. AÖM'de bu ağırlıklar rastgele olarak belirlenir ve eğitim sürecinde güncellenmez. Gizli katman nöronlarının aktivasyon fonksiyonları sabittir ve genellikle sigmoid veya radyal tabanlı gibi basit fonksiyonlar seçilir. Bu fonksiyonlar eğitim sırasında değiştirilmez, bu da öğrenme sürecini basitleştirir. Gizli katman ile çıkış katmanını birbirine bağlayan çıkış ağırlıkları, gizli katman çıkış matrisinin Moore-Penrose tersi kullanılarak analitik olarak hesaplanır. Bu hesaplama ile ağırlıklar tek bir adımda belirlenir. AÖM tek gizli katmana odaklanır ve eğitim yalnızca çıkış katmanı ağırlıklarını içerir. Bu, eğitim sürecini basitleştirir ve geleneksel geriye yayılım yöntemlerine kıyasla hesaplama açısından verimli kılar.

N adet rastgele farklı örnek (x_i, t_i) için, burada $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in \mathbb{R}^n$ ve $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in \mathbb{R}^m$, gizli katmanında \tilde{N} adet hücreli ve $g(x)$ aktivasyon fonksiyonlu AÖM matematiksel olarak eşitlik (3.1)'de verilmiştir.

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = o_j, j=1, \dots, N \quad (3.1)$$

Burada $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$, i'inci gizli nöronu ve giriş düğümlerini birbirine bağlayan giriş ağırlık vektörüdür, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$, i'inci gizli nöronu ve çıkış düğümlerini birbirine bağlayan çıkış ağırlık vektörüdür ve b_i , i'inci gizli nöronun eşiğidir. $w_i \cdot x_j$, iç çarpımı belirtir. Çıkış nöronları doğrusal olarak seçilmiştir. $g(x)$ aktivasyon fonksiyonlu \tilde{N} gizli katman hücresine sahip geleneksel SLFN'lerde bu N örneğe sıfır hatayla yakalaşabilmesi eşitlik (3.2) formülüyle hesaplanmaktadır.

$$\sum_{j=1}^{\tilde{N}} \tilde{N} [o_j - t_j] = 0 \quad (3.2)$$

3.1'deki N denklemleri kısaca şu şekilde yazılabilir:

$$H\beta = T \quad (3.3)$$

$$H(w_1, \dots, w_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, x_1, \dots, x_n) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \dots & \vdots \\ g(w_1 \cdot x_n + b_1) & \dots & g(w_{\tilde{N}} \cdot x_n + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \quad (3.4)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_{\tilde{N}}^T \end{bmatrix}_{N \times m} \quad (3.5)$$

Denklem 3.4'deki H gizli katman çıkış matrisidir. H matrisinin i'inci sütunu i numaralı gizli düğüm çıktısını x_1, x_2, \dots, x_N girişlerine göre ifade eder. Eşitlik (3.3)'den eşitlik (3.6)'da verilen çıkış bağlantı ağırlık kestirimi elde edilebilir. AÖM'de öğrenmeyi ifade eden eşitlik budur. H^\dagger , H matrisinin Moore-Penrose tersini ifade eder.

$$\hat{\beta} = H^\dagger T \quad (3.6)$$

O AÖM'nin çıkış matrisi olmak üzere belirlenen β parametreleri ile AÖM eşitlik (3.8)'de verilen hesaplama ile çıkış üretir.

$$O = H\beta \quad (3.7)$$

Bu şekilde öğrenme tamamlanmış olur. AÖM'nin üstünlükleri arasında hızlı öğrenme, basitlik ve iyi genelleme başarımı bulunmaktadır. Ancak, klasik AÖM'nin belirli karmaşık görevlerde veya veri kümelerinde geleneksel yöntemler kadar iyi performans gösteremeyebileceğini belirtmek önemlidir. Ayrıca, gizli katmandaki tekrarlı eğitim eksikliği, değişen veri dağılımlarına uyum sağlamada sınırlılık getirebilir.

AÖM'nin tanıtılmasından bu yana, yaklaşıma yönelik daha fazla gelişme ve değişiklik olmuş, yeteneklerini artırmayı ve belirli zorluklara çözüm getirmeyi amaçlayan varyasyonlar ortaya çıkmıştır.

3.1. Çok Katmanlı Aşırı Öğrenme Makinesi

Çok katmanlı aşırı öğrenme makinesi (ÇK-AÖM), AÖM yöntemini çok katmanlı sinir ağı yapısına genişleten bir öğrenme modelidir. ÇK-AÖM'nin temel amacı, çok katmanlı bir yapı kullanarak daha karmaşık özellikleri öğrenmek ve genelleme yeteneğini artırmaktır. Ancak, bu hedefe ulaşmak için derin öğrenme yöntemlerinde olduğu gibi ağırlıkların sürekli olarak güncellenmesi yerine, ağırlıkların başlangıçta rastgele seçilip eğitim sürecinde güncellenmemesi prensibini korur. Bu yaklaşım ise daha az hesaplama maliyetini yanında

getirir. ÇK-AÖM büyük veri kümelerinin hızlı bir şekilde işlenmesi ve öğrenme modelinin hızlı bir şekilde uyarlanması gereken durumlar için avantajlıdır.

H'nin Moore-Penrose tersi, $H^T H$ tekil olmadığında $H^\dagger = (H^T H)^{-1} H^T$ 'dir. Moore-Penrose tersi, eğer HH^T tekil değilse $H^\dagger = H^T (HH^T)^{-1}$ olarak da tanımlanır. Ridge regresyon teorisini kullanarak, $(1/\lambda)$ değerleri HH^T 'nin diyagonal elemanlarına eklenir. Tek katman için öğrenme denklemi şu şekildedir:

$$\beta = H^T \left(\frac{1}{\lambda} + HH^T \right)^{-1} T \quad (3.8)$$

$k = 2, \dots, M$ olan her bir gizli katman k için giriş gösterimleri vardır. k 'inci katman için n adet harici giriş kullanıldığında $X^{(k)} = [x_1^{(k)}, \dots, x_n^{(k)}]$ olarak temsili girişler gösterilebilir. $\beta^{(k)} = [\beta_1^{(k)}, \dots, \beta_n^{(k)}]$ ise transformasyon matrisidir. Sonuçta temsili girişin hesaplaması şöyle olur:

$$X^{(k)} = H^{(k)} \beta^{(k)} \quad (3.9)$$

Denklem 3.10 kullanılarak $\beta^{(k)}$ eşitlik (3.10) aracılığıyla öğrenilir.

$$\beta^{(k)} = (H^{(k)})^T \left(\frac{1}{\lambda} + H^{(k)} (H^{(k)})^T \right)^{-1} X^{(k)} \quad (3.10)$$

M katman için son katman çıktı matrisi $H^{(M)}$ olarak tanımlanır. Bu katman çıktı matrisi kullanılarak son katmanın çıktı ağırlık matrisi β şu şekilde öğrenilir:

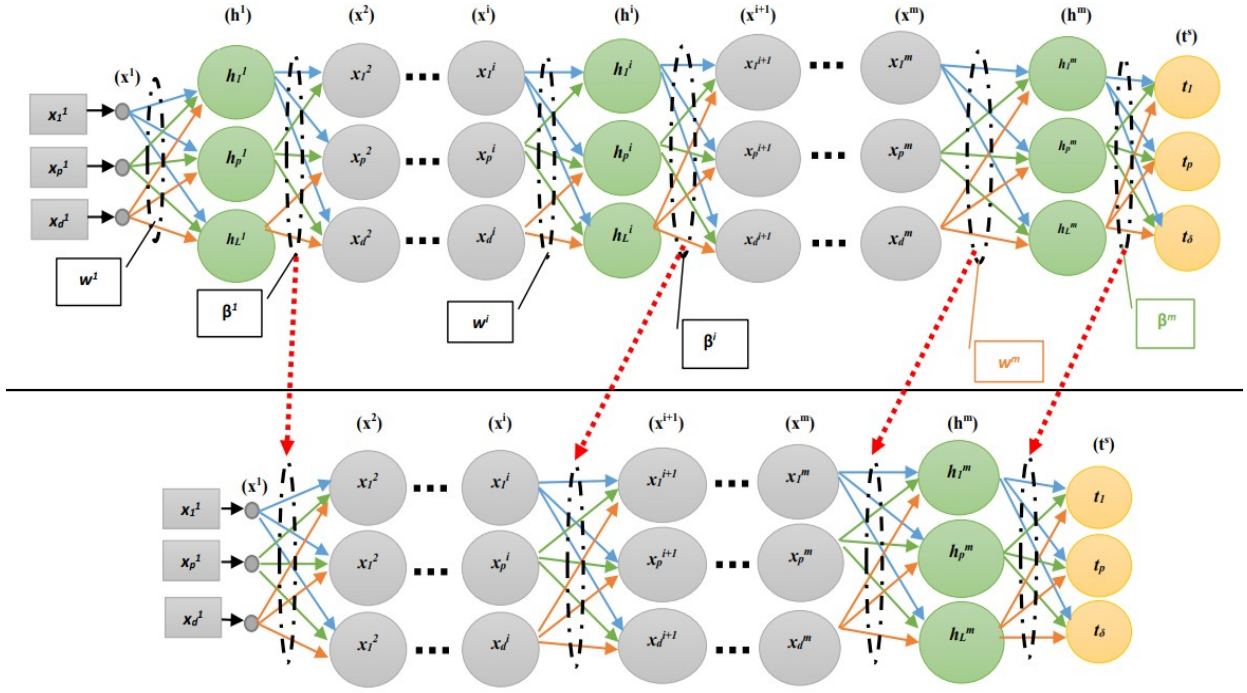
$$\beta = (H^{(M)})^T \left(\frac{1}{\lambda} + H^{(M)} (H^{(M)})^T \right)^{-1} T \quad (3.11)$$

(Tang vd., 2015)

Eğitim Fazı: Giriş verileri bir giriş katmanına beslenir, bu katmanın çıkışında bu girişler elde edilecek şekilde hesap yapılarak katmanın çıkış bağlantı ağırlıkları ($\beta^{(k)}$) belirlenir. Son katmana kadar her katman için bu işlem tekrar edilir. Son katman çıkış bağlantı ağırlıkları klasik AÖM'nin analitik hesabı ile belirlenir.

Test Fazı: Harici girişler eğitim fazında belirlenen $\beta^{(k)}$ 'lar ile çarpılarak son katman girişine kadar temsili bir giriş oluşturulur. Son katmanda klasik AÖM hesabı yapılarak çıkış elde edilir.

Şekil 3.1 eğitim ve test fazında kullanılan ağ mimarisini göstermektedir.

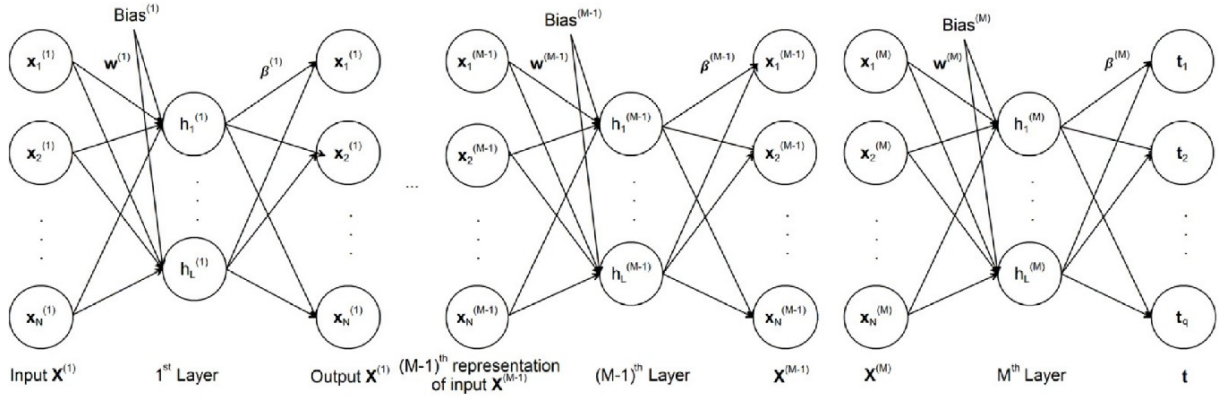


Şekil 3.1. ÇK-AÖM'nin eğitim sırasında (üst) ve eğitim sonrasında (alt) kullanım mimarisini (d harici giriş ve δ harici çıkışlı m katmanlı)

Kaynak: (Parmaksız, 2023: 92; Parmaksız ve Karakuzu, 2024)

3.2. Geliştirilmiş Çok Katmanlı Aşırı Öğrenme Makineleri

"Multilayer extreme learning machines and their modeling performance on dynamical systems" başlıklı çalışmada iki yeni çok katmanlı aşırı öğrenme makinesi geliştirmiştir (Kale ve Karakuzu, 2022). ÇK-AÖM ile eğitim ve test aşamalarında aynı yapıyı kullanan ancak modelleme başarımını artıracak ve hesaplama süresini azaltacak şekilde yazarlar tarafından iki yeni ÇK-AÖM yapısı tasarlanmıştır. Geliştirilmiş çok katmanlı aşırı öğrenme makinesi 1 (GÇK-AÖM1)'de (Şekil 3.2) her katman ortonormal olarak rastgele atanmış bağlantı ağırlıklarına sahiptir. Geliştirilmiş çok katmanlı aşırı öğrenme makinesi 2 (GÇK-AÖM2) (Şekil 3.3) ise sadece birinci katmanda ortonormal olarak rasgele atanan bağlantı ağırlıklarına sahiptir. Sonraki katmanların bağlantı ağırlıkları, bir önceki katmanın çıktı ağırlık matrisinden alınır.



Şekil 3.2. M gizli katman için GÇK-AÖM1 şeması

Kaynak: (Kale ve Karakuzu, 2022)

GÇK-AÖM1 algoritmasının eğitim aşamasının algoritması şu şekildedir:

Start;

İlk katman için ağırlık parametreleri $w^{(1)}$ ve Bias⁽¹⁾ rastgele atanır.

Denklem 3.4 kullanılarak $H^{(1)}$ hesaplanır.

Ardından, çıkış ağırlık parametresi $\beta^{(1)}$ Denklem 3.10 kullanılarak öğrenilir.

For $k=2:M$ **do**

Mevcut katmanın temsili girişi önceki katmanın çıkış matrisi $H^{(k-1)}$ ile önceki katmanın çıkış ağırlık parametreleri $\beta^{(k-1)}$ çarpılarak hesaplanır.

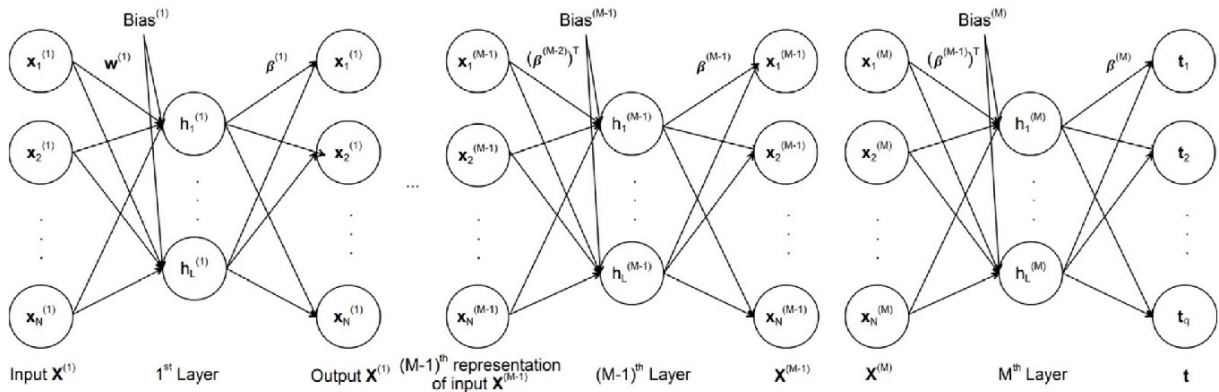
Mevcut katman için ağırlık parametreleri $w^{(k)}$ ve Bias^(k) rastgele atanır.

Denklem 3.4 kullanılarak $H^{(k)}$ hesaplanır.

Ardından, çıkış ağırlık parametresi $\beta^{(k)}$ Denklem 3.10 kullanılarak öğrenilir (Eğer $k = M$ ise Denklem 3.11 kullanılır).

end

Test aşaması için de aynı yapı kullanılır. Test aşamasının tek farkı, sadece eğitim aşamasında ileri hesaplama için giriş ağırlık parametreleri w , biaslar ve çıkış ağırlık parametreleri β kullanılır.



Şekil 3.3. M gizli katman için GÇK-AÖM2 şeması

Kaynak: (Kale ve Karakuzu, 2022)

GÇK-AÖM2 algoritmasının eğitim aşamasının algoritması şu şekildedir:

Start;

İlk katman için ağırlık parametreleri $w^{(1)}$ ve Bias⁽¹⁾ rastgele atanır.

Denklem 3.4 kullanılarak $H^{(1)}$ hesaplanır.

Ardından, çıkış ağırlık parametresi $\beta^{(1)}$ Denklem 3.10 kullanılarak öğrenilir.

For k=2:M **do**

Mevcut katmanın temsili girişi önceki katmanın çıkış matrisi $H^{(k-1)}$ ile önceki katmanın çıkış ağırlık parametreleri $\beta^{(k-1)}$ çarpılarak hesaplanır. Giriş bağlantı ağırlık parametreleri ($w^{(k)}$), önceki katmanın çıkış ağırlık parametrelerinin transpozu olan $(\beta^{(k-1)})^T$ olarak seçilir. Bias(k) rastgele atanır.

Denklem 3.4 kullanılarak $H^{(k)}$ hesaplanır.

Ardından, çıkış ağırlık parametresi $\beta^{(k)}$ Denklem 3.10 kullanılarak öğrenilir (Eğer k = M ise Denklem 3.11 kullanılır).

end

Test aşaması GÇK-AÖM1 ile aynı yapıdadır.

Kısaca bahsedilen bu geliştirilmiş ÇK-AÖM'ler hesaplama için hem eğitimde hem de eğitim sonrasında nöronları kullanırlar. Bu özellikleriyle, orijinal ÇK-AÖM'den ayrılarak eğitim sırasında elde edilen temsil matrisini kullanmamaları sayesinde daha adaptif bir hesaplama yeteneği kazanmışlardır (Kale ve Karakuzu, 2022).

4. ÇK-AÖM'LERİN REGRESYON PROBLEMLERİNDE BAŞARIMI

Bu bölümde, 2022 yılında Kale ve Karakuzu tarafından geliştirilen iki yeni ÇK-AÖMnin (GÇK-AÖM1 ve GÇK-AÖM2), regresyon problemleri özelinde orijinal ÇK-AÖM'ye kıyasla sistemli bir analize tabi tutularak elde edilen sonuçlar verilmiştir. Bu üç model 10 adet veri kümesi üzerinde çalıştırılmış ve elde edilen bulgular istatistiksel metrikler aracılığıyla analiz edilmiştir.

4.1. Regresyon

Regresyon, bir değişkenin diğer bir veya birden fazla değişkenle ilişkisini inceleyen bir istatistiksel analiz yöntemidir. Regresyon analizi, bağımlı değişkenin (sonuç değişkeni) bir veya birden fazla bağımsız değişkenle (neden değişkenleri) ilişkisini modellemek için kullanılır. Regresyon analizi, özellikle sosyal bilimler, ekonomi, finans ve mühendislik gibi alanlarda sıklıkla kullanılan bir araçtır. Örneğin, bir şirketin satışlarının hava durumu, rekabet, reklam harcamaları ve fiyat gibi faktörlere bağlı olduğunu düşünelim. Bu durumda, regresyon analizi, bu faktörlerin satışlar üzerindeki etkisini ölçmek için kullanılabilir.

Bağımlı değişken, genellikle anlamak veya tahmin etmek istediğimiz değişkendir, bu arada bağımsız değişken(ler) ise bağımlı değişkenin değerini etkileyen faktör(ler)dir. Regresyon analizi, bu faktörlerin bağımlı değişken üzerindeki etkisini nicel olarak ölçmeye çalışır. Temelde iki tür regresyon bulunur: basit regresyon ve çoklu regresyon. Basit regresyon, bir bağımlı değişken ve bir bağımsız değişken arasındaki ilişkiyi incelerken, çoklu regresyon, bir bağımlı değişken ile birden fazla bağımsız değişken arasındaki ilişkiyi inceler.

$$y = \beta_0 + \beta_1 x + \varepsilon \quad (4.1)$$

Denklem 4.1'de doğrusal regresyon formülü verilmiştir. Burada; y bağımlı değişken, x bağımsız değişken, β_0 , doğrunun y -ksenini kestiği noktadaki sabit terim, β_1 , eğim (x 'in katsayısı), ε , hata (gerçek değerler ile model tarafından tahmin edilen değerler arasındaki fark) olarak ifade edilir.

Çoklu regresyon durumunda, birden fazla bağımsız değişken olacaktır, denklem şöyle değişir:

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \dots + \beta_k \cdot x_k + \varepsilon \quad (4.2)$$

Formülde x_1, x_2, \dots, x_k bağımsız değişkenlerdir, β_0 sabit terim, $\beta_1, \beta_2, \dots, \beta_k$ ise ilgili katsayılarıdır.

4.2. Veri Kümeleri

Kaliforniya Üniversitesi, Irvine tarafından yönetilen UCI Machine Learning Repository, çeşitli disiplinlerden gelen gerçek dünya problemlerine odaklanan geniş bir veri havuzu ve araştırma kaynağıdır. Bu platform, farklı disiplinlerden ve uygulama alanlarından gelerek, geniş bir konu yelpazesini kapsayan çeşitli veri kümelerine ev sahipliği yapmaktadır. Bu özel veri havuzu, her bir veri kümesinin detaylı açıklamalarını içeren bir dokümantasyon sağlar. Kullanıcılar, veri kümelerini daha iyi anlamak ve analiz etmek için bu bilgileri kullanabilirler. Ayrıca, veri kümeleri genellikle önceden işlenmiş ve temizlenmiş olduğundan, araştırmacılar zamanlarını analize odaklanarak geçirebilirler. Her bir veri kümesi, yapılandırılmış verilerden oluşmakta ve genellikle özellikler, hedef değişkenler ve açıklamalar içermektedir. Bu sayede araştırmacılar, çeşitli analizler yaparak ve çeşitli modeller geliştirerek makine öğrenimi alanındaki çeşitli konuları keşfetme fırsatına sahiptirler. UCI Machine Learning Repository'nin sunduğu avantajlardan biri de topluluk tabanlı bir yaklaşım benimsemesidir. Kullanıcılar, platform üzerinden deneyimlerini paylaşabilir, sorular sorabilir ve diğer araştırmacılarla etkileşimde bulunabilirler. Bu, makine öğrenimi alanındaki bilgi paylaşımını teşvik eder ve farklı bakış açılarından yararlanmayı sağlar.

UCI Machine Learning Repository, sınıflandırma, regresyon, kümeleme gibi farklı makine öğrenimi problemleri için kullanılabilen çeşitli veri kümeleri sunmaktadır. Bu veri kümeleri, makine öğrenimi algoritmalarının eğitilmesi ve değerlendirilmesi için değerli bir kaynaktır. Araştırmacılar, kendi projelerine uygun veri kümelerini seçerek, çeşitli algoritmaları test edebilir ve bu veri kümeleri üzerinden yeni metodolojiler geliştirebilirler. Bu kaynak, çeşitli veri kümeleri aracılığıyla makine öğrenimi alanındaki metodolojilere dair derinlemesine bir anlayış geliştirmek isteyenlere hitap eder. Veri kümelerinde bulunan yapılandırılmış veriler, analizcilerin ve araştırmacıların farklı veri madenciliği tekniklerini uygulamalarına ve bu teknikleri geliştirmelerine olanak tanır.

Bu çalışmada kullanılmak üzere UCI Machine Learning Repository içerisinde 10 adet regresyon ilişkili görevler için paylaşılmış veri kümesi belirlenmiştir. Belirlenen veri kümeleri Tablo 4.1.'de verilmiştir. Tabloyu takiben de veri kümeleri hakkında kısa bilgilendirme yapılmıştır.

Tablo 4.1. Veri kümeleri

Veri Kümesi Adı	Örnek Sayısı	Özellik Sayısı	Normalizasyon
Abalone	4177	9	Z-Score
Auto-Mpg	392	9	Z-Score
Auto Imports Price	205	26	Z-Score
Boston Housing	506	14	Z-Score
Relative CPU Performance	209	9	Z-Score
Delta Ailerons	7129	6	Z-Score
Delta Elevators	9517	7	Z-Score
Servo	167	5	Z-Score
İstanbul Stock Exchange	536	8	Z-Score
QSAR fish toxicity	908	7	Uygulanmadı

Abalone: Abalone canlılarının yaşını fiziksel ölçümlerden tahmin etme amaçlı bir veri kümesidir (Nash vd., 1995).

Auto-Mpg: Araçların galon başına mil cinsinden şehir içi yakıt tüketimiyle ilgilidir (Quinlan, 1993).

Auto Imports Price: Bir otomobilin çeşitli karakteristikler açısından özellikleri, atanan sigorta risk derecesi, diğer otomobillere kıyasla kullanımdaki normalleştirilmiş kayıpları olmak üzere üç tip parametrelerden oluşan bir veri kümesidir (Schlimmer, 1987).

Boston Housing: Boston mahallelerindeki konut değerleri ile ilgilidir (Harrison ve Rubinfeld, 1978).

Relative CPU Performance: İşlemcilerin bağıl performans değerleri ile ilgilidir (Feldmesser, 1987).

Delta Ailerons: F16 savaş uçağının kanatçıklarını kontrol etme görevinden elde edilmiş verilerdir (Camacho, 2000).

Delta Elevators: F16 savaş uçağının asansörlerini kontrol etme görevinden elde edilmiş verilerdir (Camacho, 2000).

Servo: Bir servomekanizmanın bir adım değişikliğine yanıt vermesi için gereken süreyi tahmin etme ile ilgilidir (Ulrich, 1993).

İstanbul Stock Exchange: 5 Haziran 2009'dan 22 Şubat 2011'e kadar İstanbul Menkul Kıymetler Borsası'nın diğer yedi uluslararası endeks ile getirilerini içermektedir; SP, DAX, FTSE, NIKKEI, BOVESPA, MSCE_EU, MSCI_EM (Akbiçic, 2013).

QSAR fish toxicity: 908 kimyasaldan oluşan bir sette Pimephales promelas balığına yönelik akut su toksisitesini tahmin etmek için kantitatif regresyon QSAR modellerini geliştirmek için kullanılmış bir veri kümesidir (Ballabio vd., 2019).

4.3. Yöntem

Çalışma Matlab ortamında Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz işlemci ve 16 GB RAM içeren donanım kullanılarak yapılmıştır. Veri kümelerini işleyebilmek adına Matlab ortamında kullanılacak kodlar yazılarak, veriler çalışma ortamına başarıyla yüklenmiştir. Veri kümelerinin %70'i öğrenme sürecine dahil edilmiş ve geriye kalan %30'u test aşamasında değerlendirilmek üzere ayrılmıştır. Kategorik değişkenler, ağırların öğrenme sürecine etkili bir şekilde katkı sağlayabilmesi için sayısal forma dönüştürülmüştür. Normalizasyon gereken durumlarda, veriler Z-Score normalizasyon yöntemi kullanılarak uygun bir formata getirilmiştir.

Z-Score, istatistiksel bir yöntem olup bir veri kümesindeki değerleri standart normal dağılıma dönüştürmek amacıyla kullanılan bir normalizasyon tekniğidir. Bu yöntem, veri kümesindeki değerlerin ortalama değerden çıkartılıp standart sapmaya bölünerek uygulanır. Elde edilen sonuçlar, standart normal dağılımın z-puanları olarak adlandırılan değerlerdir.

Z-Score normalizasyonu şu şekildedir:

$$Z = \frac{X - \mu}{\sigma} \quad (4.3)$$

Burada; **Z** normalize edilmiş değeri, **X** orijinal değeri, **μ** veri kümesinin aritmetik ortalamasını, **σ** veri kümesinin standart sapmasını temsil eder.

Bu normalizasyon yöntemi, veri kümesindeki değerlerin birbirleriyle karşılaştırılabilir hale gelmesini sağlar. Normalize edilmiş değerlerin aritmetik ortalaması 0, standart sapması ise 1 olacaktır. Bu, veri kümesindeki değerlerin dağılımının merkezi ve yayılma özelliklerini standartlaştırarak, istatistiksel analiz ve karşılaştırmaların daha tutarlı bir şekilde yapılmasına

olanak tanır. Z-Score normalizasyonu ayrıca, aykırı değerlerin etkilerini azaltarak model başarımını artırmak ve veri madenciliği uygulamalarında daha güvenilir sonuçlar elde etmek amacıyla sıklıkla kullanılır.

Çok katmanlı aşırı öğrenme makineleri başarım analizinde, orijinal ÇK-AÖM, GÇK-AÖM1 ve GÇK-AÖM2 olmak üzere toplamda üç farklı çok katmanlı aşırı öğrenme makinesi kullanılmıştır. Her bir veri kümesi üzerinde her bir ağ ve onun eğitim algoritması 100 kez çalıştırılarak istatistiksel güvenilirliği artırılmıştır. Çalışmanın sonuçları, her aşırı öğrenme makinesi ve veri kümesi kombinasyonu için elde edilen metrikler, başarım grafikleri ve çalışma alanı değişkenleri titizlikle kaydedilmiştir.

Başarım metriği olarak Root Mean Square Error (RMSE) kullanılarak, her modelin öğrenme başarımını kapsamlı bir şekilde değerlendirilmiştir. Bu metrik, modellerin tahminlerinin gerçek değerlere olan uzaklığını ölçerek, modellerin genel başarımını ölçmede etkili bir yöntem sağlamaktadır.

RMSE, hata fonksiyonu olarak kullanılan bir metriktir, özellikle regresyon analizlerinde ve tahmin modellerinin değerlendirilmesinde sıkça kullanılan bir başarım ölçütüdür.

RMSE, modelin gerçek değerlerle tahmin ettiği değerler arasındaki farkların karelerinin ortalamasının kareköküdür. Formülü şu şekildedir:

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (4.4)$$

Denklem 4.4 içerisindeki; n veri sayısını, Y_i gerçek değerleri, \hat{Y}_i , modelin tahmin ettiği değerleri temsil eder.

RMSE'nin düşük olması, modelin gerçek değerlere daha yakın tahminler yaptığını gösterir ve bu nedenle daha iyi bir başarımı ifade eder. RMSE, hataların büyüklüğünü ölçer ve bu ölçüm, modelin ne kadar etkili olduğunu değerlendirme konusunda kullanışlı bir araç sağlar.

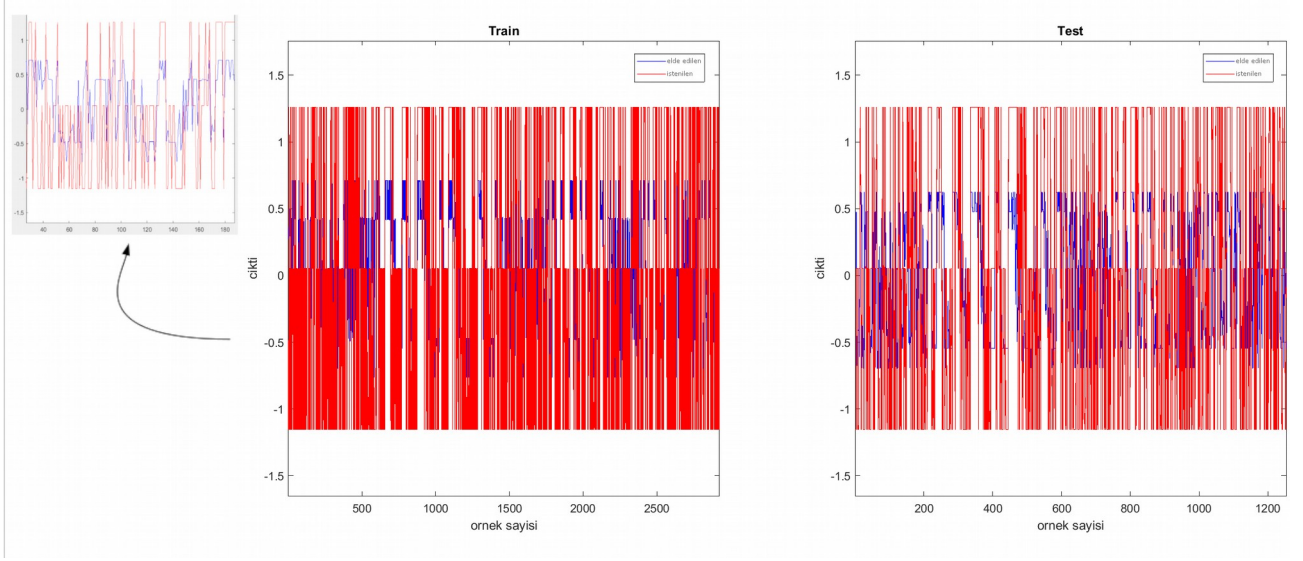
RMSE değerleri, modellerin öğrenme performansını nicel olarak değerlendirmek adına kullanılan temel bir ölçüdür. Elde edilen başarım grafikleri, modellerin öğrenme sürecindeki ilerlemesini görsel olarak temsil ederken, çalışma alanı değişkenleri, modellerin öğrenme sürecindeki etkileşimlerini daha derinlemesine incelemek için kullanılmıştır.

Çalışmanın sonuçlarına göre, her bir aşırı öğrenme makinesi için belirlenen metrikler, başarımlar, başarımlar ve çalışma alanı değişkenleri incelenmiştir. Bu analiz, her algoritmanın öğrenme sürecindeki güçlü ve zayıf yönlerini anlamak adına önemli bir içgörü sunmaktadır.

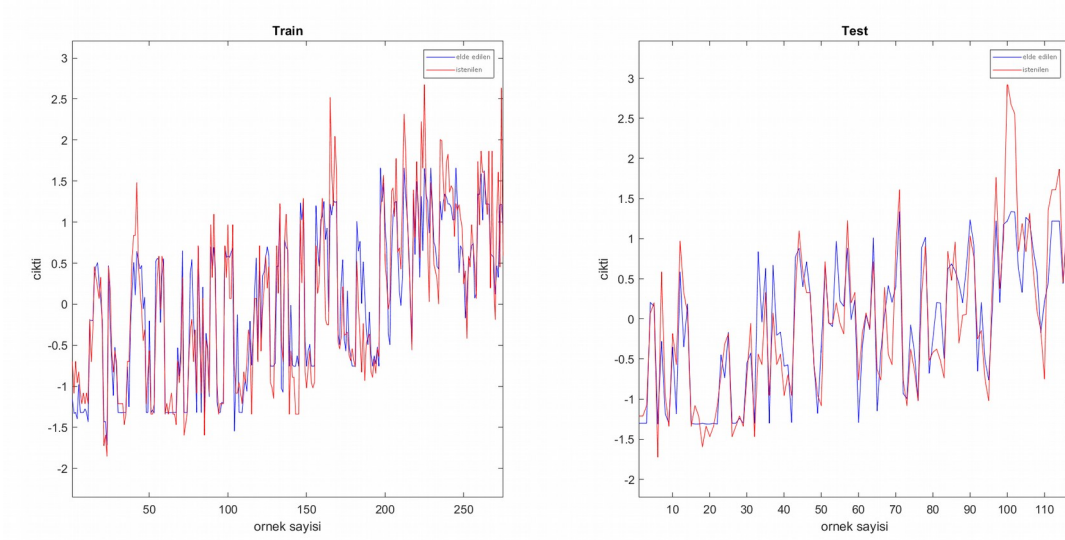
4.4. Bulgular

Bu alt bölümde, çeşitli örnek veri kümeleri ile elde edilen çözümlerin ayrıntılı analizi sunulacak ve sonuçlar bölümünde üstünlükleri ile olası sınırlamaları tartışılacaktır. Gerçekleştirilen deneysel çalışmada ÇK-AÖM, GÇK-AÖM1 ve GÇK-AÖM2 yapılarının regresyon problemleri için çalışma sonuçları kaydedilmiştir. Bulgular ile matematiksel düşünce ve problem çözme stratejilerinde alternatif yaklaşımların keşfedilmesine katkıda bulunulması amaçlanmaktadır.

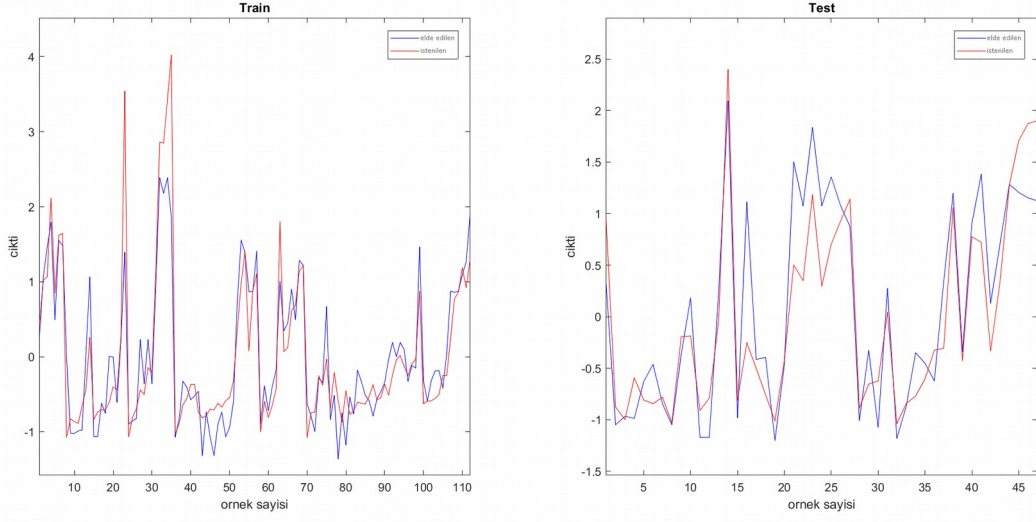
Şekil 4.1 - Şekil 4.30 arasında her bir veri kümesi için ÇK-AÖM'lerin öğrenme ve test fazındaki başarımlarını gösteren grafikler paylaşılmıştır. Bunlar 100 koşma içerisindeki en iyi başarımların grafikleridir. Grafiklerdeki kırmızı çizimler beklenen, mavi çizimler elde edilen değerleri ifade etmektedir. Kırmızı ve mavi çizimlerin daha iyi örtüşmesi modellerin daha iyi öğrenmesi anlamına gelir.



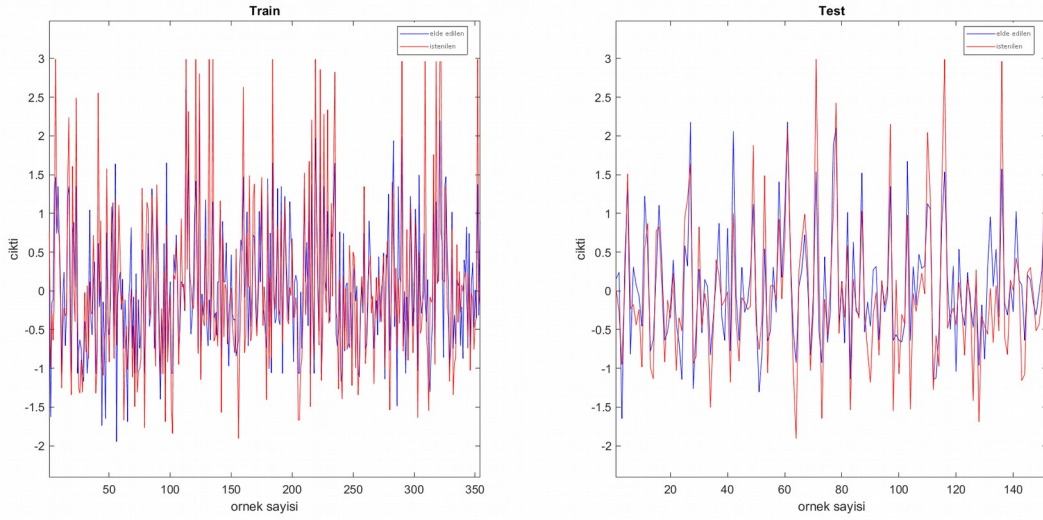
Şekil 4.1. ÇK-AÖM'nin Abalone veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



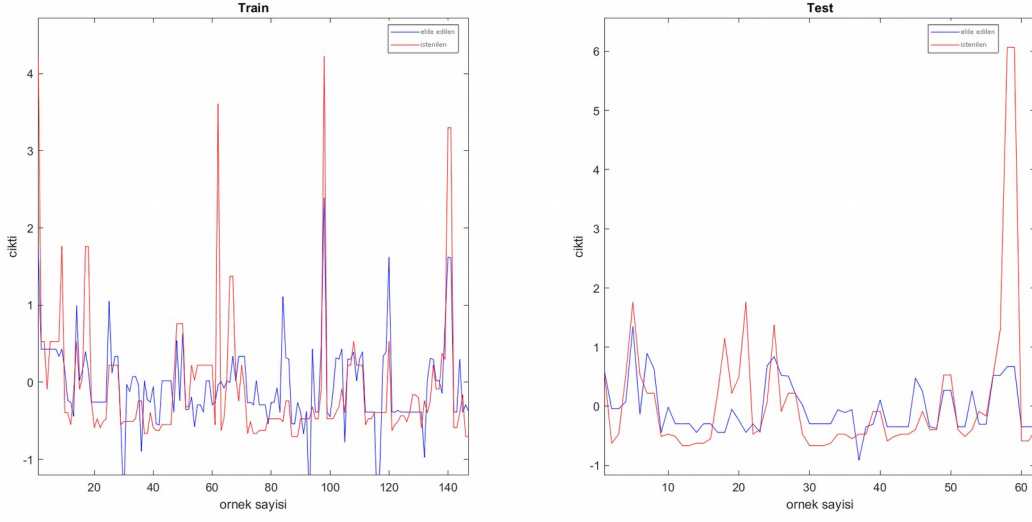
Şekil 4.2. ÇK-AÖM'nin Auto-Mpg veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



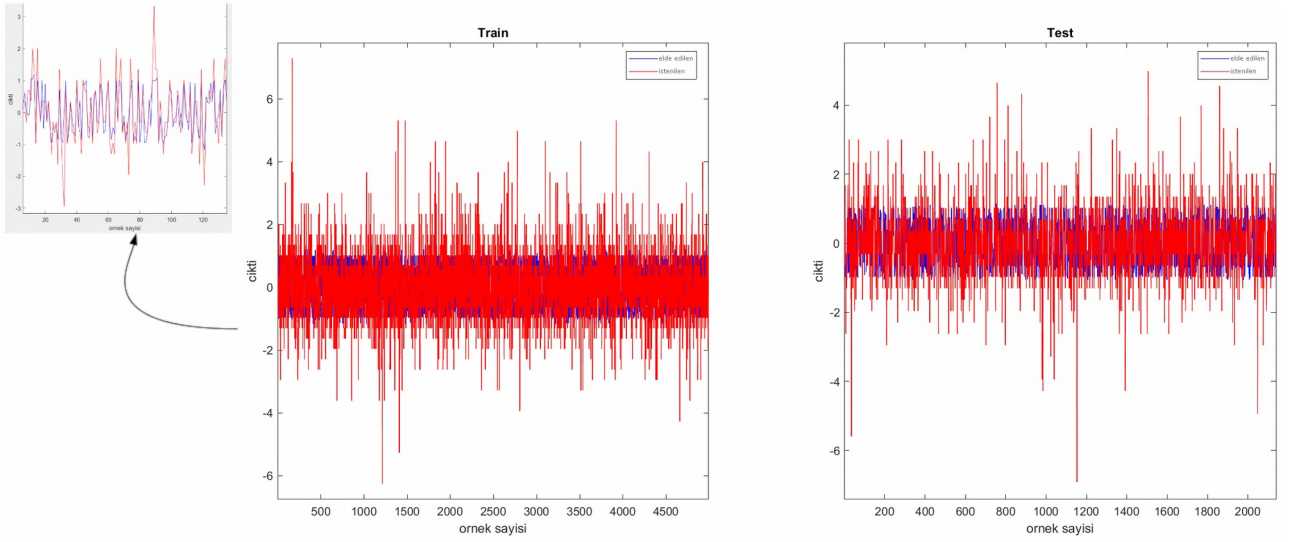
Şekil 4.3. ÇK-AÖM'nin Auto Imports Price veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



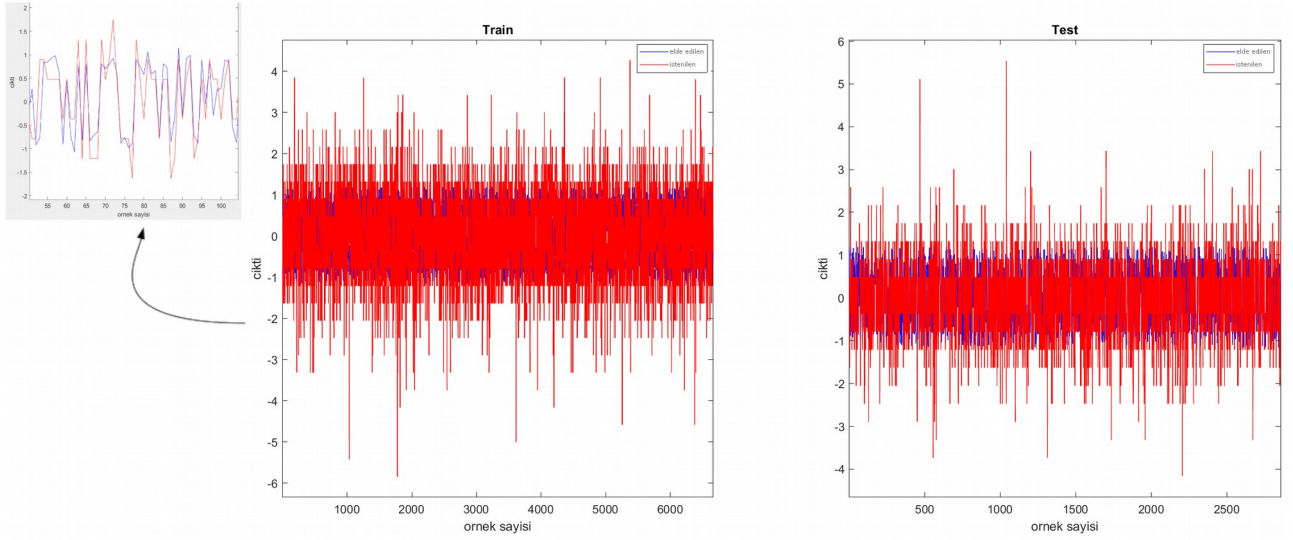
Şekil 4.4. ÇK-AÖM'nin Boston Housing veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



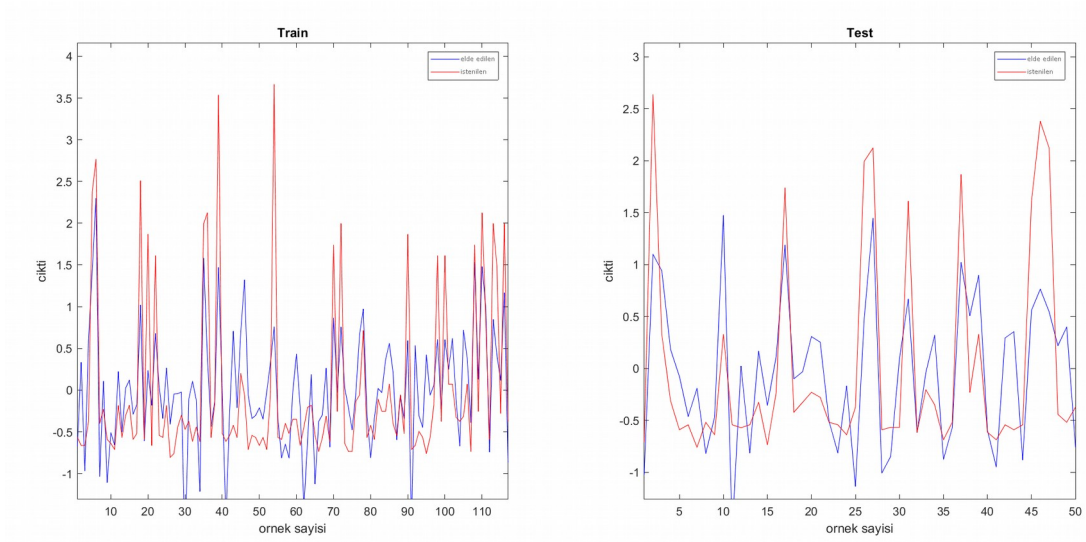
Şekil 4.5. ÇK-AÖM'nin Relative CPU Performance veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



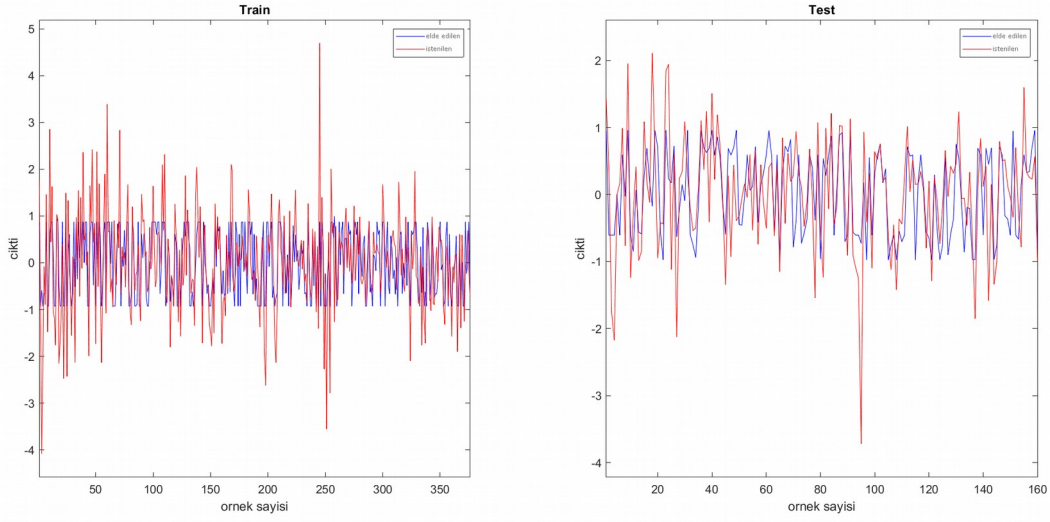
Şekil 4.6. ÇK-AÖM'nin Delta Ailerons veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



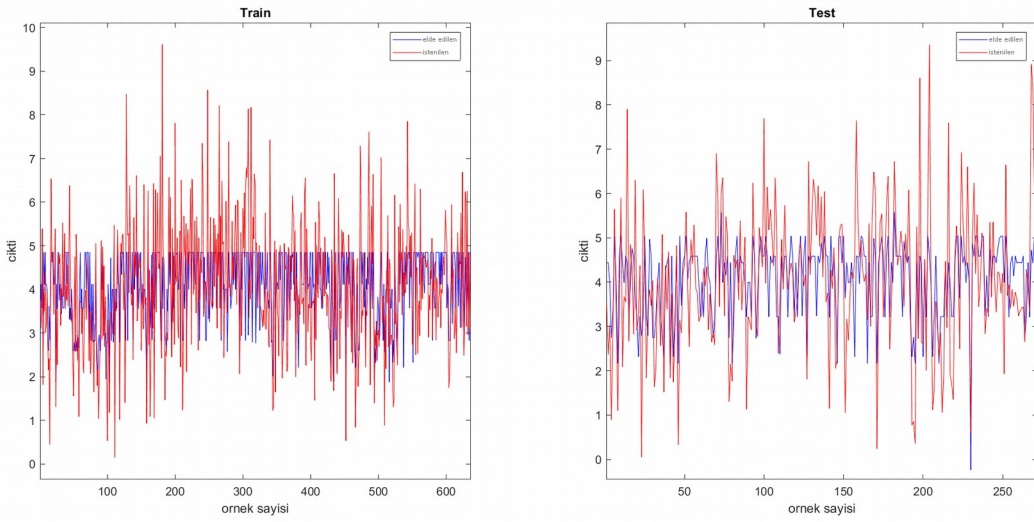
Şekil 4.7. ÇK-AÖM'nin Delta Elevators veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



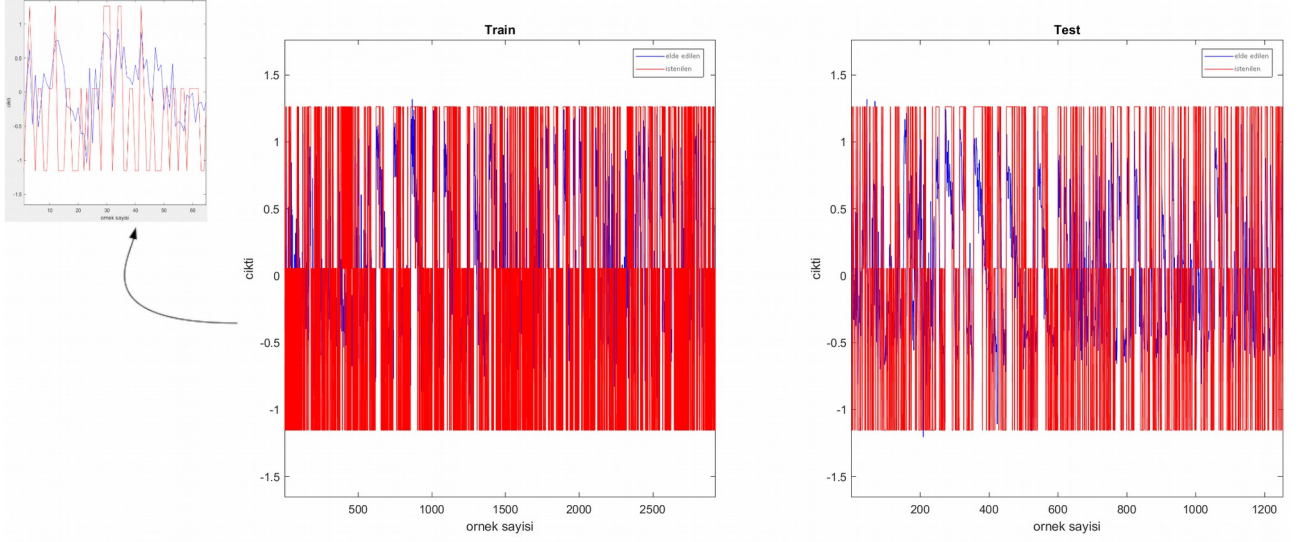
Şekil 4.8. ÇK-AÖM'nin Servo veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



Şekil 4.9. ÇK-AÖM'nin İstanbul Stock Exchange veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

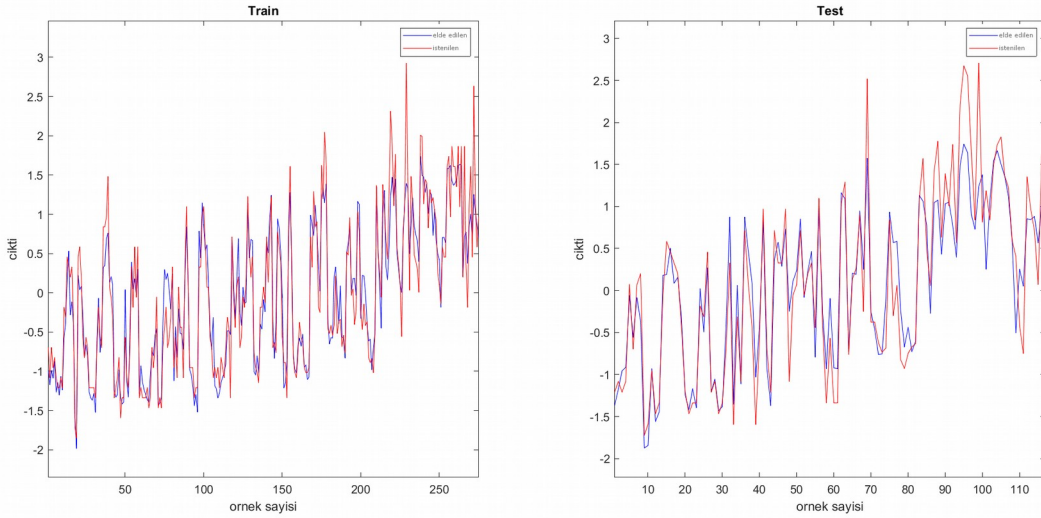


Şekil 4.10. ÇK-AÖM'nin QSAR fish toxicity veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

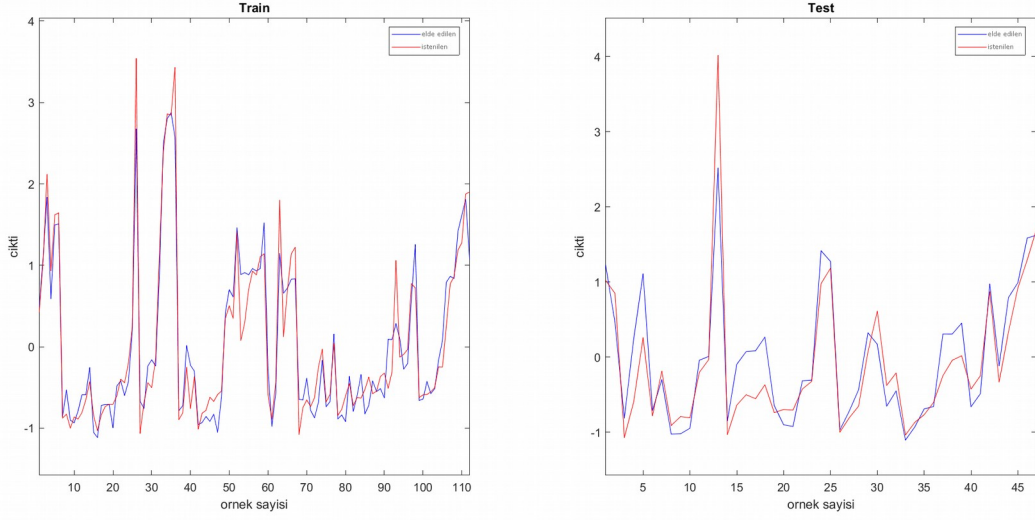


Şekil 4.11. GÇK-AÖM1'in Abalone veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

Abalone veri kümesi 4177 örnek ve 9 özellik içermektedir. Veriler canlılığın fiziksel ölçümlerini içeren sayılardan oluşmaktadır. Şekil 4.1. ve Şekil 4.11. karşılaştırıldığında ÇK-AÖM'nin öğrenme grafiği beklenene göre daha küçük değerlerde seyretmektedir. GÇK-AÖM1'in grafiğinde görüldüğü üzere ÇK-AÖM'den daha iyi bir öğrenme grafiği çıkarmaktadır.

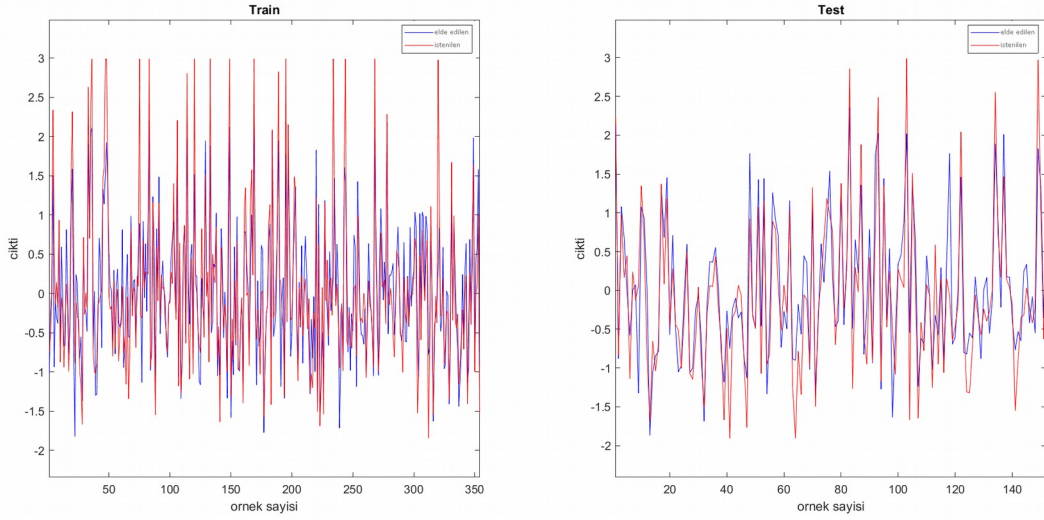


Şekil 4.12. GÇK-AÖM1'in Auto-Mpg veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



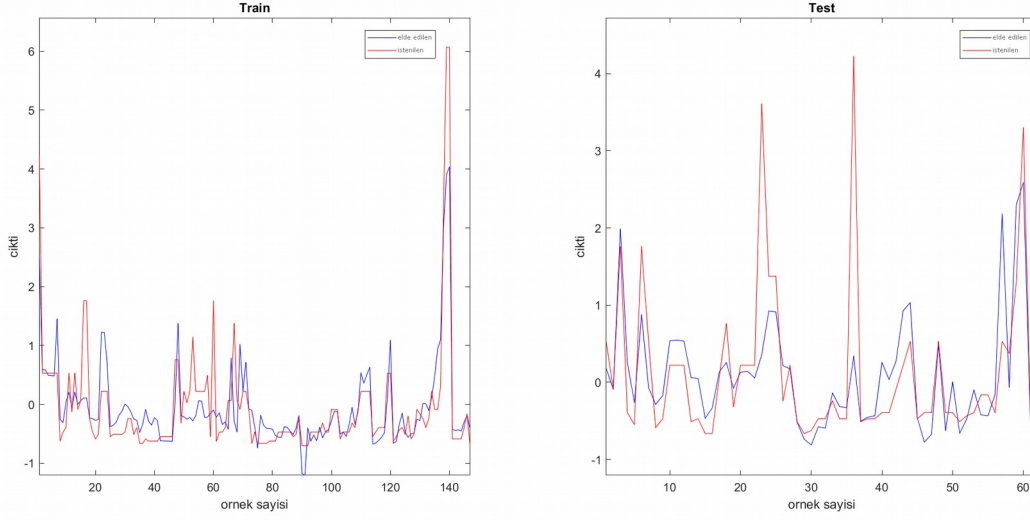
Şekil 4.13. GÇK-AÖM1'in Auto Imports Price veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

Şekil 4.13 ve Şekil 4.23 beraber incelendiğinde Auto Imports Price veri kümesi için GÇK-AÖM1'in daha iyi bir öğrenme gerçekleştirdiği görülmektedir. Test fazında ise GÇK-AÖM2'nin daha başarılı olduğu görülmektedir.

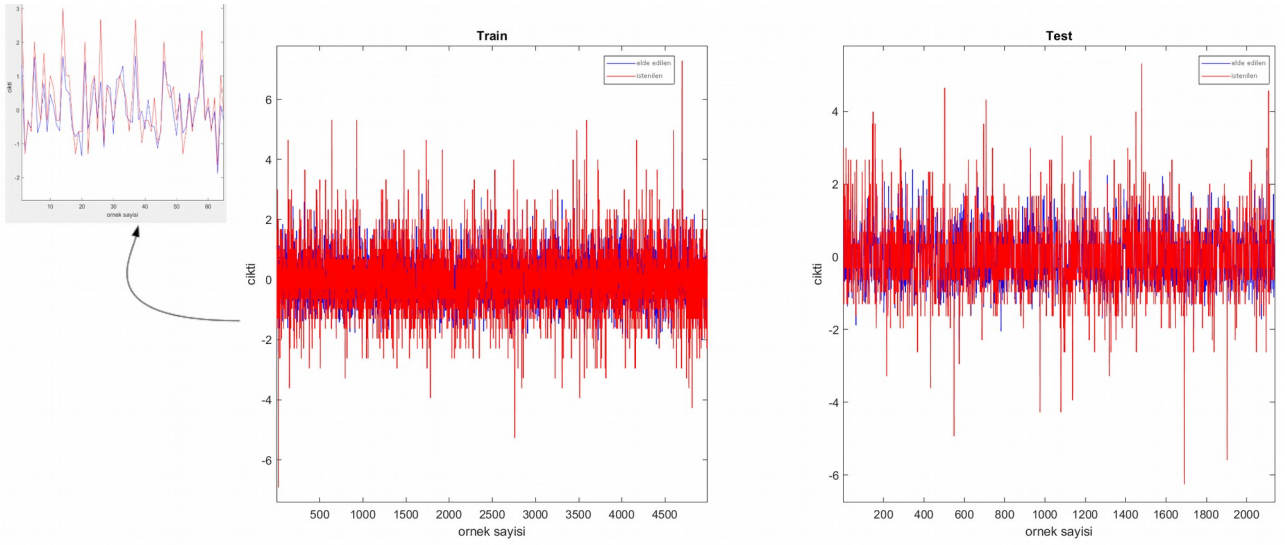


Şekil 4.14. GÇK-AÖM1'in Boston Housing veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

Şekil 4.14 ve Şekil 4.24 karşılaştırıldığında GÇK-AÖM1'in Boston Housing veri kümesi için daha başarılı çalıştığı yorumu yapılabilir.

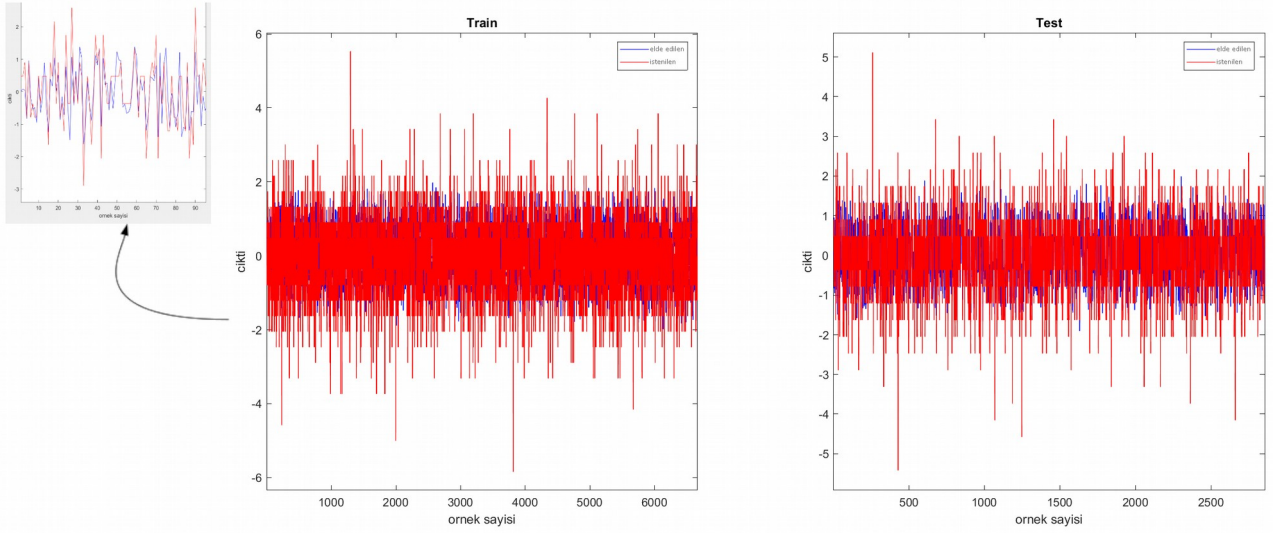


Şekil 4.15. GÇK-AÖM1'in Relative CPU Performance veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

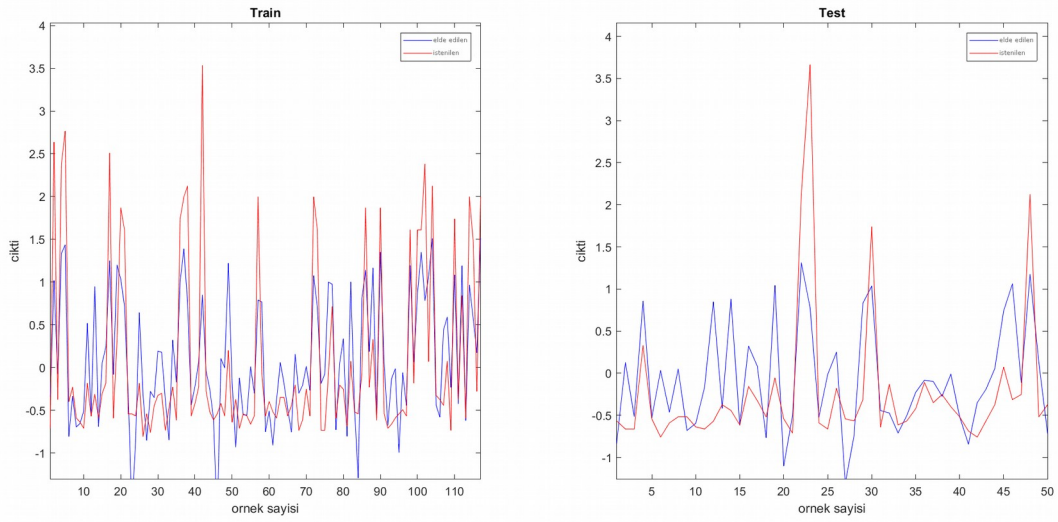


Şekil 4.16. GÇK-AÖM1'in Delta Ailerons veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

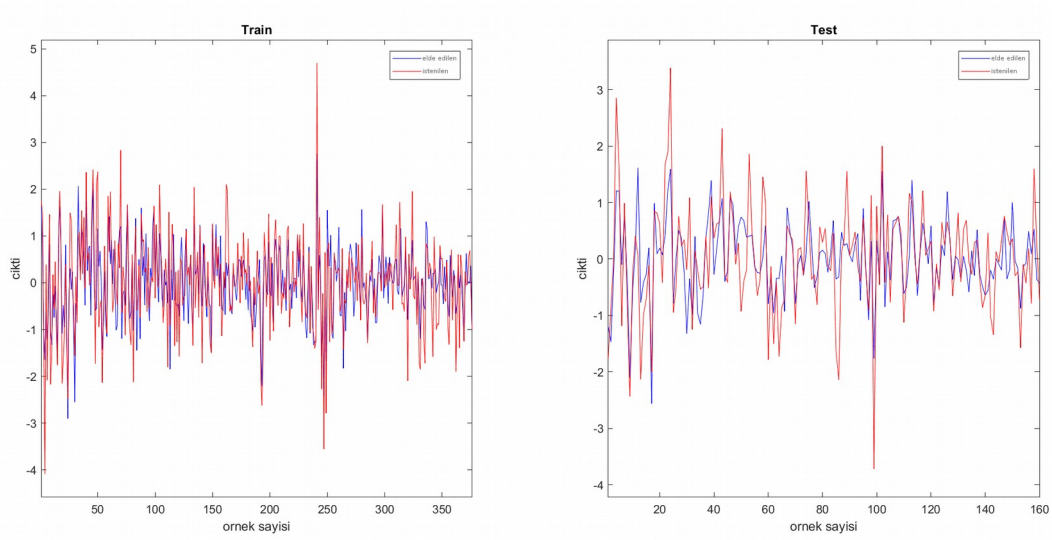
Şekil 4.16 ve Şekil 4.26 incelendiğinde GÇK-AÖM1 ve GÇK-AÖM2'nin başarımlarının yakın olduğu gözlemlenmektedir.



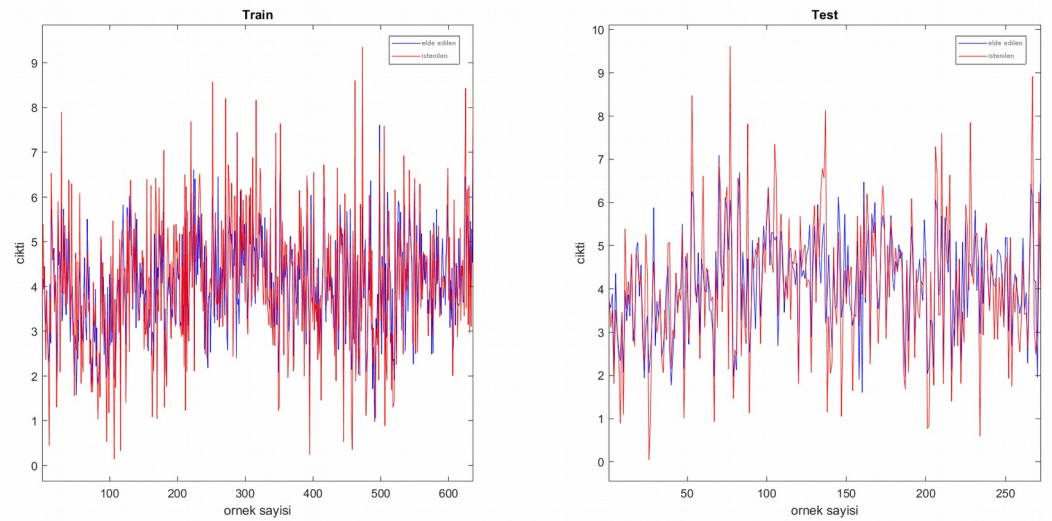
Şekil 4.17. GÇK-AÖM1'in Delta Elevators veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



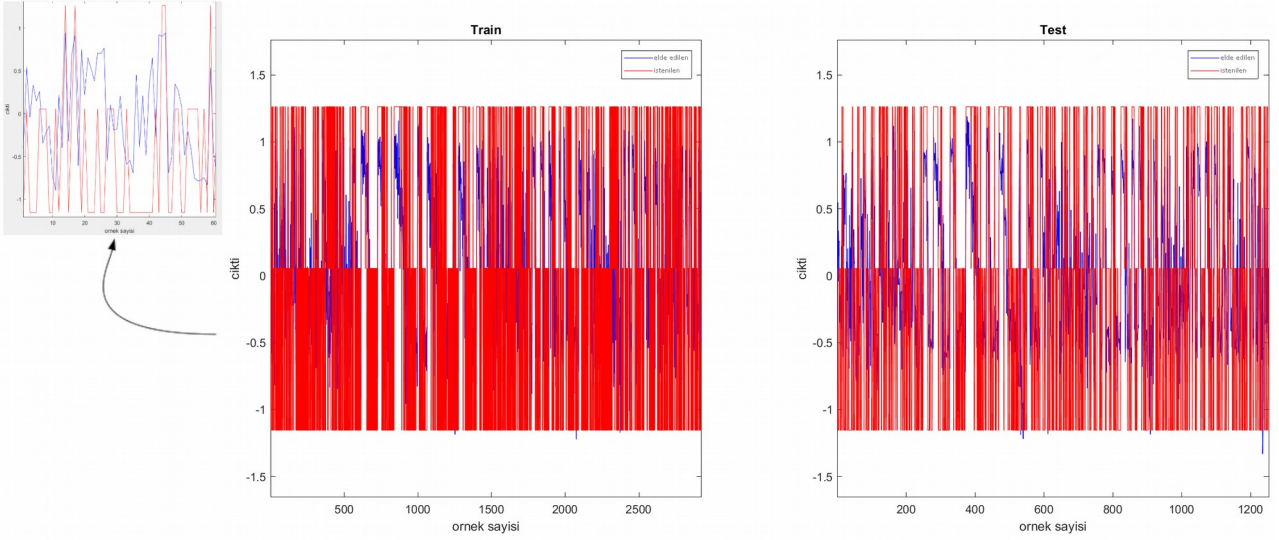
Şekil 4.18. GÇK-AÖM1'in Servo veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



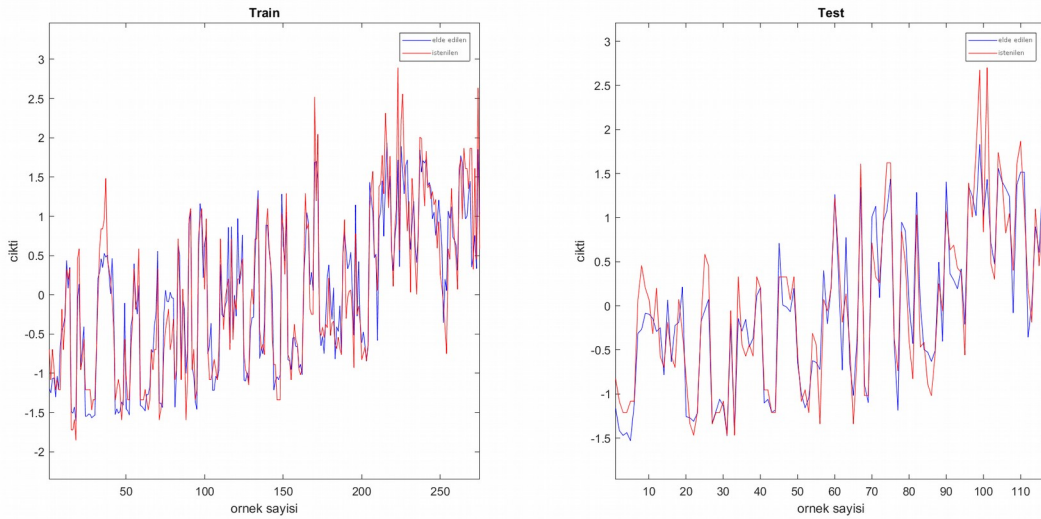
Şekil 4.19. GÇK-AÖM1'in İstanbul Stock Exchange veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



Şekil 4.20. GÇK-AÖM1'in QSAR fish toxicity veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

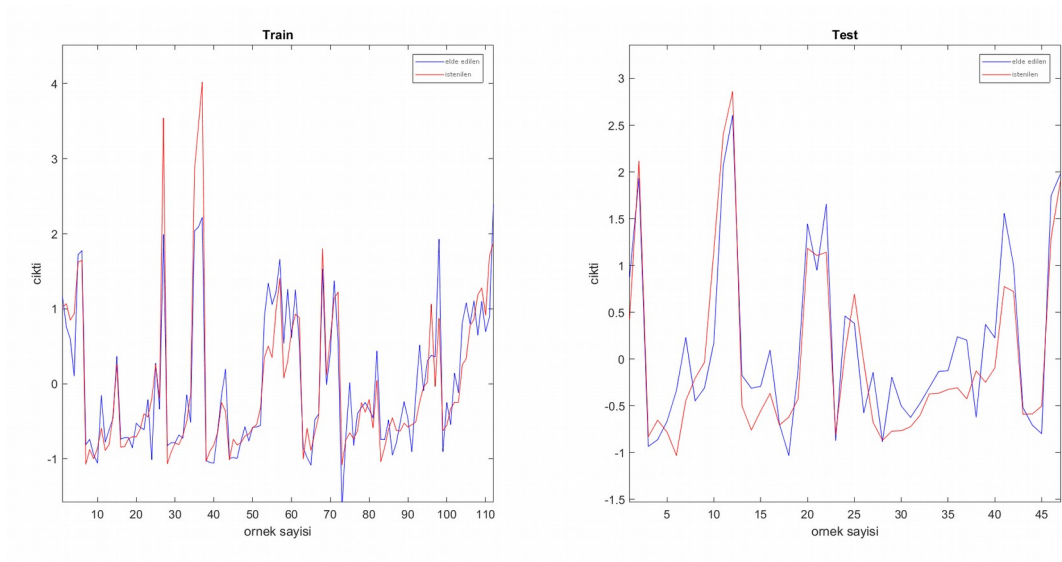


Şekil 4.21. GÇK-AÖM2'nin Abalone veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

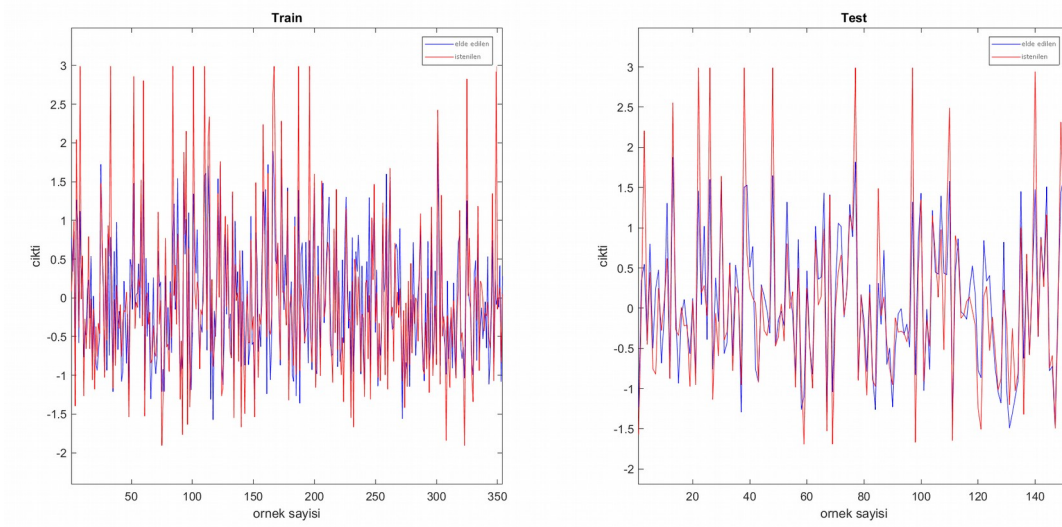


Şekil 4.22. GÇK-AÖM2'nin Auto-Mpg veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

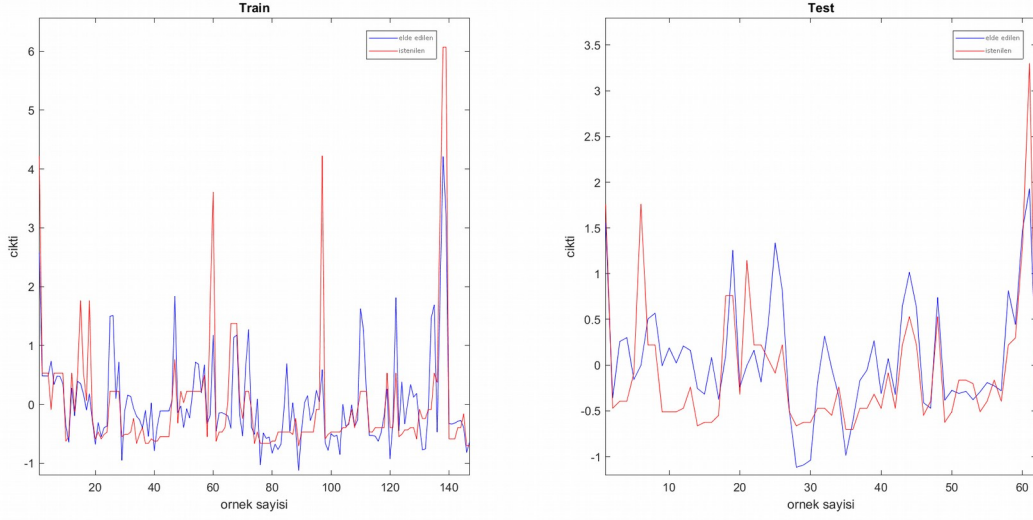
Şekil 4.12 ve Şekil 4.22 beraber incelendiğinde GÇK-AÖM1'in öğrenme, GÇK-AÖM2'nin test aşamasında daha başarılı olduğu görülmektedir. Şekil 4.2. göz önüne alındığında ise her ikisinin de ÇK-AÖM'den başarılı olduğu yorumu yapılabilir.



Şekil 4.23. GÇK-AÖM2'nin Auto Imports Price veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

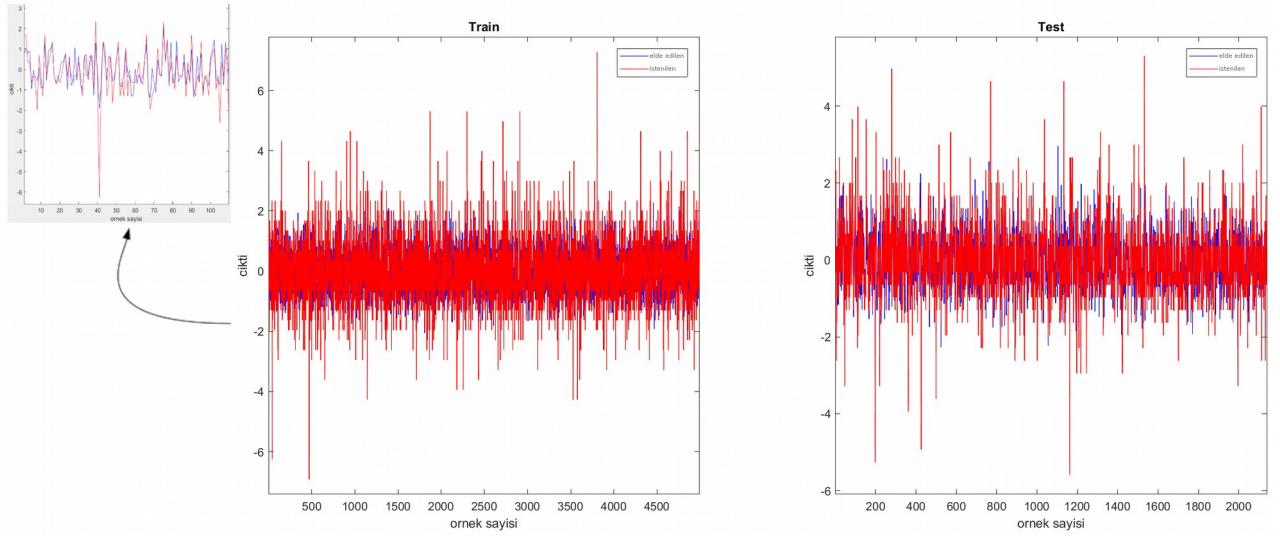


Şekil 4.24. GÇK-AÖM2'nin Boston Housing veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

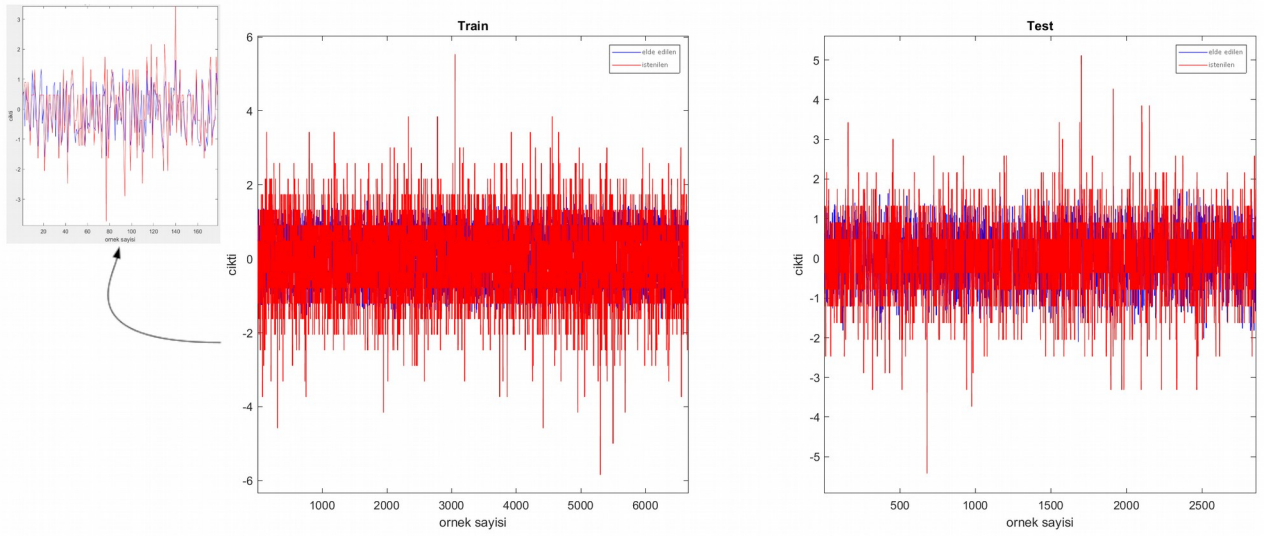


Şekil 4.25. GÇK-AÖM2'nin Relative CPU Performance veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

Şekil 4.5.'e göre Şekil 4.15. ve Şekil 4.25. karşılaştırıldığında GÇK-AÖM2 test fazında çok daha başarılı iken öğrenme fazında GÇK-AÖM1'in gerisinde kalmaktadır.

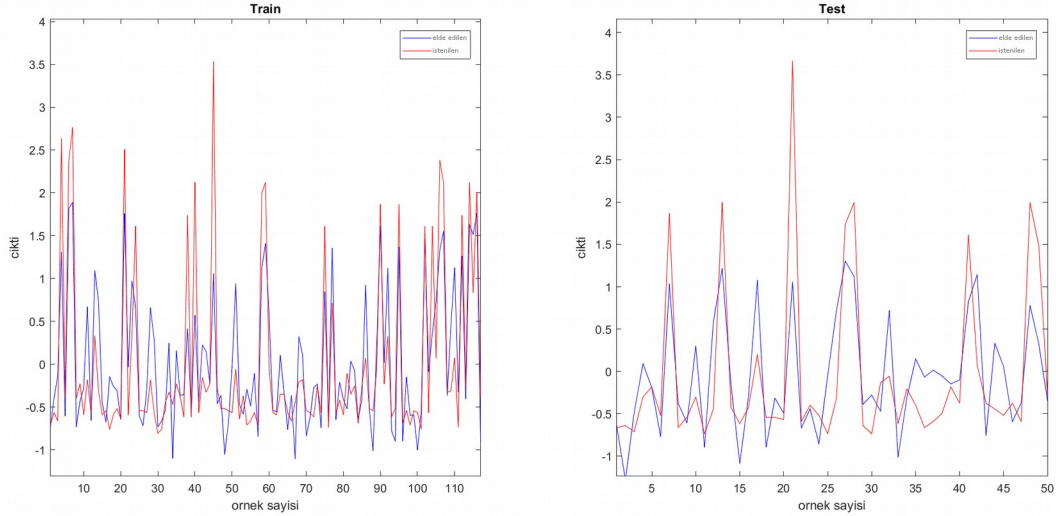


Şekil 4.26. GÇK-AÖM2'nin Delta Ailerons veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



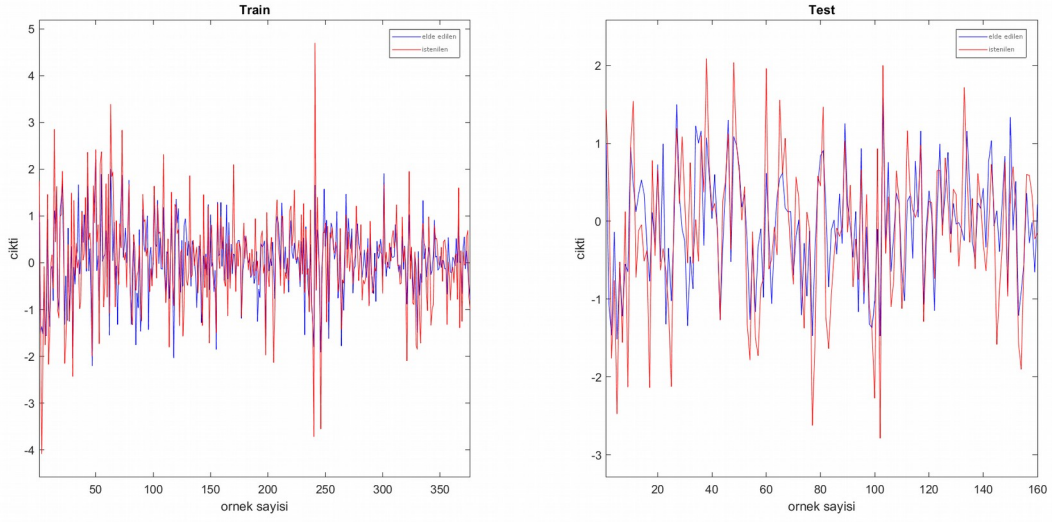
Şekil 4.27. GÇK-AÖM2'nin Delta Elevators veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

Şekil 4.17 ve Şekil 4.27 incelendiğinde GÇK-AÖM1 ve GÇK-AÖM2'nin başarımlarının birbirine yakın olduğu gözlemlenmektedir.

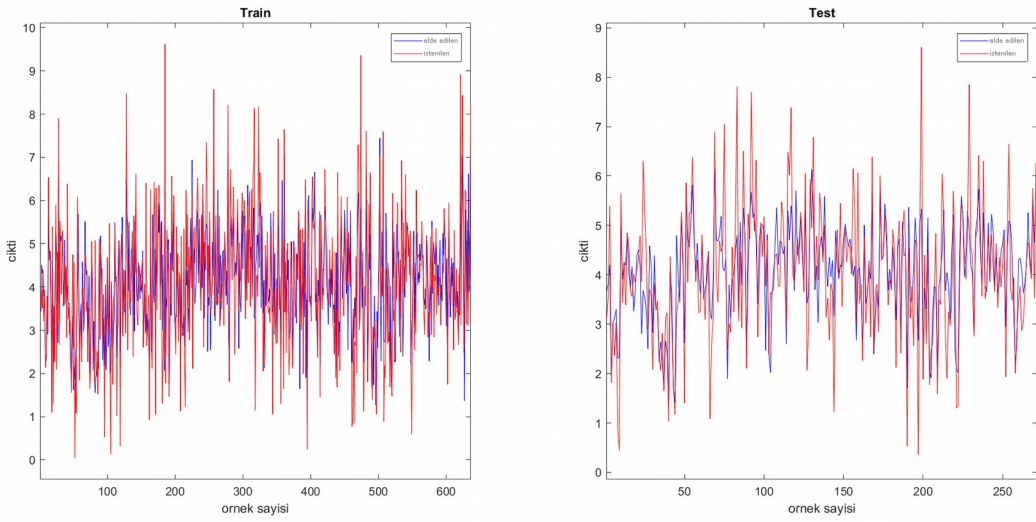


Şekil 4.28. GÇK-AÖM2'nin Servo veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

Şekil 4.8, Şekil 4.18 ve Şekil 4.28 incelendiğinde GÇK-AÖM'lerin daha iyi başarımlar sergilediği yorumu yapılabilir.



Şekil 4.29. GÇK-AÖM2'nin İstanbul Stock Exchange veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları



Şekil 4.30. GÇK-AÖM2'nin QSAR fish toxicity veri kümesi için öğrenme ve test aşamalarındaki en iyi başarımları

Şekil 4.10, Şekil 4.20 ve Şekil 4.30 beraber incelendiğinde GÇK-AÖM'lerin ÇK-AÖM'ye kıyasla daha üstün bir başarı sergilediği gözlemlenmektedir.

Tablo 4.2 – Tablo 4.10. arasında işlem yapma zamanı, eğitim ve test fazları için metrikler, Tablo 4.11. – Tablo 4.16. içerisinde ise değişim oranları verilmiştir. Değişim oranlarının hesaplanmasında aşağıdaki formül kullanılarak hesaplanan değerler yukarı yuvarlanmıştır.

$$100 - \left(\frac{100 \times \text{Orijinal}}{\text{Geliştirilmiş}} \right) \quad (4.5)$$

Örneğin Orijinal AÖM için 10 olan değer geliştirilmiş bir AÖM için 4 olduğunda;

$$100 - \left(\frac{100 \times 10}{4} \right) = -150 \quad (4.6)$$

yani %150 bir azalma olduğu hesaplanır.

Sunulan tablolar, eğitim ve test aşamalarında elde edilen başarımların değişim metriklerini ayrıntılı bir şekilde analiz etmeyi mümkün kılmaktadır. Tabloların içeriğine baktığımızda, negatif yüzdeler ilgili ölçümlerdeki azalmayı, pozitif yüzdeler ise artışı temsil etmektedir. Bu değerler, başarı metriklerinin eğitim ve test aşamaları boyunca nasıl değiştiğini gösterirken, bu değişimlerin yönünü belirlemede yardımcı olmaktadır.

Tablo 4.2. Orijinal ÇK-AÖM'nin saniye cinsinden işlem yapma zamanı

Veri Kümesi	Zaman			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	2,34465	2,285585	2,778999	0,065477
Auto-Mpg	0,015258	0,011437	0,035518	0,003133
Auto Imports Price	0,005141	0,00333	0,036198	0,003637
Boston Housing	0,021114	0,016514	0,026896	0,002711
Relative CPU Performance	0,005139	0,00385	0,007593	0,000769
Delta Ailerons	9,64104	9,281384	10,988251	0,230254
Delta Elevators	22,401905	20,245399	23,805531	0,824557
Servo	0,003803	0,002858	0,005319	0,000561
İstanbul Stock Exchange	0,02667	0,021021	0,037515	0,003234
QSAR fish toxicity	0,06974	0,062732	0,086861	0,005633

Tablo 4.3. GÇK-AÖM1'in saniye cinsinden işlem yapma zamanı

Veri Kümesi	Zaman			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	2,351695	2,278851	2,666947	0,061271
Auto-Mpg	0,013724	0,010358	0,028768	0,002361
Auto Imports Price	0,005008	0,003889	0,007042	0,000662
Boston Housing	0,02014	0,016578	0,02723	0,002295
Relative CPU Performance	0,005245	0,00414	0,007111	0,000576
Delta Ailerons	9,62942	9,381657	10,82835	0,193427
Delta Elevators	22,482612	19,969602	23,806884	0,869154
Servo	0,004229	0,003129	0,00697	0,000689
İstanbul Stock Exchange	0,024883	0,020646	0,032602	0,002629
QSAR fish toxicity	0,068445	0,061546	0,092772	0,005787

Tablo 4.4. GÇK-AÖM2'nin saniye cinsinden işlem yapma zamanı

Veri Kümesi	Zaman			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	2,319448	2,264361	2,42006	0,032761
Auto-Mpg	0,012894	0,009583	0,030891	0,002613
Auto Imports Price	0,004484	0,003472	0,006879	0,000726
Boston Housing	0,020209	0,016149	0,027957	0,002465
Relative CPU Performance	0,005086	0,004047	0,007723	0,000749
Delta Ailerons	9,644272	9,298899	10,81437	0,309682
Delta Elevators	22,443984	19,976427	23,678841	0,861648
Servo	0,004173	0,003182	0,006126	0,000629
İstanbul Stock Exchange	0,024293	0,020513	0,031162	0,002479
QSAR fish toxicity	0,06979	0,061506	0,093641	0,006651

Tablo 4.5. Orijinal ÇK-AÖM'nin eğitim fazındaki başarımların metrikleri

Veri Kümesi	Train RMSE			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	0,882468	0,872732	0,895971	0,004282
Auto-Mpg	0,509041	0,421662	0,596035	0,034212
Auto Imports Price	0,535542	0,461429	0,607845	0,031415
Boston Housing	0,731446	0,640535	0,858934	0,041398
Relative CPU Performance	0,751419	0,664418	0,797899	0,025993
Delta Ailerons	0,731183	0,653095	0,855126	0,043335
Delta Elevators	0,711736	0,651719	0,841738	0,037475
Servo	0,794293	0,713422	0,879362	0,037467
İstanbul Stock Exchange	0,81908	0,771681	0,865165	0,017047
QSAR fish toxicity	1,274204	1,131604	3,405549	0,221348

Tablo 4.6. GÇK-AÖM1'in eğitim fazındaki başarımleri

Veri Kümesi	Train RMSE			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	0,846435	0,834331	0,858007	0,005328
Auto-Mpg	0,355122	0,32966	0,380798	0,011306
Auto Imports Price	0,370184	0,293892	0,433927	0,022685
Boston Housing	0,549781	0,501972	0,579328	0,01542
Relative CPU Performance	0,621732	0,532381	0,701745	0,038401
Delta Ailerons	0,566499	0,556264	0,578155	0,004452
Delta Elevators	0,604081	0,600042	0,608637	0,001652
Servo	0,72135	0,672148	0,749774	0,017033
İstanbul Stock Exchange	0,646064	0,633615	0,659716	0,005192
QSAR fish toxicity	0,895423	0,880958	0,914905	0,007708

Tablo 4.7. GÇK-AÖM2'nin eğitim fazındaki başarımleri

Veri Kümesi	Train RMSE			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	0,864354	0,84288	0,880541	0,007141
Auto-Mpg	0,406559	0,361384	0,463973	0,022315
Auto Imports Price	0,505856	0,441246	0,564079	0,025179
Boston Housing	0,63067	0,56772	0,729596	0,024227
Relative CPU Performance	0,853153	0,716902	0,93987	0,039922
Delta Ailerons	0,620114	0,581574	0,684697	0,019924
Delta Elevators	0,639717	0,620938	0,68624	0,010924
Servo	0,670422	0,569557	0,746407	0,035261
İstanbul Stock Exchange	0,701792	0,666156	0,74382	0,018944
QSAR fish toxicity	1,07307	0,964736	1,253585	0,050997

Tablo 4.8. Orijinal ÇK-AÖM'nin test fazındaki başarımları metrikleri

Veri Kümesi	Test RMSE			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	0,891272	0,876062	0,904873	0,005521
Auto-Mpg	0,569123	0,456118	0,6812	0,045854
Auto Imports Price	0,558124	0,4558	0,782666	0,058965
Boston Housing	0,611095	0,51632	0,747182	0,04515
Relative CPU Performance	1,16812	1,095807	1,714209	0,070596
Delta Ailerons	0,749724	0,666191	0,875042	0,045347
Delta Elevators	0,688992	0,619329	0,820552	0,037815
Servo	0,804273	0,704874	0,938345	0,044155
İstanbul Stock Exchange	0,784211	0,724293	0,957357	0,03427
QSAR fish toxicity	1,47835	1,273779	3,584879	0,223769

Tablo 4.9. GÇK-AÖM1'in test fazındaki başarımları metrikleri

Veri Kümesi	Test RMSE			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	0,861305	0,844805	0,872465	0,005761
Auto-Mpg	0,438086	0,402422	0,478145	0,017661
Auto Imports Price	0,51253	0,396903	0,642057	0,048644
Boston Housing	0,512981	0,469431	0,587998	0,022508
Relative CPU Performance	0,859449	0,767422	0,973966	0,042622
Delta Ailerons	0,579118	0,564526	0,593032	0,005741
Delta Elevators	0,611155	0,606866	0,616979	0,002037
Servo	0,747004	0,703476	0,777231	0,017293
İstanbul Stock Exchange	0,672432	0,656355	0,689402	0,007081
QSAR fish toxicity	1,036048	1,008098	1,075339	0,012797

Tablo 4.10. GÇK-AÖM2'nin test fazındaki başarımları metrikleri

Veri Kümesi	Test RMSE			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	0,836162	0,814778	0,860364	0,008917
Auto-Mpg	0,396931	0,338577	0,458476	0,025422
Auto Imports Price	0,506336	0,39981	0,638015	0,050302
Boston Housing	0,595761	0,515467	0,688515	0,032978
Relative CPU Performance	0,670268	0,524681	0,801699	0,053288
Delta Ailerons	0,607682	0,56507	0,659971	0,019188
Delta Elevators	0,630727	0,607642	0,681196	0,010958
Servo	0,746193	0,66896	0,800437	0,0287
İstanbul Stock Exchange	0,712203	0,667242	0,79667	0,025408
QSAR fish toxicity	0,972824	0,864944	1,290098	0,073147

Tablo 4.11. GÇK-AÖM1'in orijinale göre eğitim fazındaki metriklerinin yüzdelik değişimi

Veri Kümesi	Train RMSE			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	-%4,26	-%4,61	-%4,43	+%19,64
Auto-Mpg	-%43,35	-%27,91	-%56,53	-%202,61
Auto Imports Price	-%44,67	-%57,01	-%40,09	-%38,49
Boston Housing	-%33,05	-%27,61	-%48,27	-%168,47
Relative CPU Performance	-%20,86	-%24,81	-%13,71	+%32,32
Delta Ailerons	-%29,08	-%17,41	-%47,91	-%873,39
Delta Elevators	-%17,83	-%8,62	-%38,3	-%2168,47
Servo	-%10,12	-%6,15	-%17,29	-%119,97
İstanbul Stock Exchange	-%26,79	-%21,8	-%31,15	-%228,34
QSAR fish toxicity	-%42,31	-%28,46	-%272,23	-%2771,67

Tablo 4.12. GÇK-AÖM1'in orijinale göre test fazındaki metriklerinin yüzdelerik deęişimi

Veri Kümesi	Test RMSE			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	-%3,48	-%3,7	-%3,72	+%4,17
Auto-Mpg	-%29,92	-%13,35	-%42,47	-%159,64
Auto Imports Price	-%8,9	-%14,84	-%21,9	-%21,22
Boston Housing	-%19,13	-%9,99	-%27,08	-%100,6
Relative CPU Performance	-%35,92	-%42,8	-%76,01	-%65,64
Delta Ailerons	-%29,46	-%18,01	-%47,56	-%689,88
Delta Elevators	-%12,74	-%2,06	-%33	-%1756,41
Servo	-%7,67	-%0,2	-%20,73	-%155,34
İstanbul Stock Exchange	-%16,63	-%10,36	-%38,87	-%383,98
QSAR fish toxicity	-%42,7	-%26,36	-%233,38	-%1648,61

Tablo 4.13. GÇK-AÖM1'in orijinale göre işlem yapma zamanının yüzdelerik deęişimi

Veri Kümesi	Zaman			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	+%0,3	-%0,3	-%4,21	-%6,87
Auto-Mpg	-%11,18	-%10,42	-%23,47	-%32,7
Auto Imports Price	-%2,66	+%14,38	-%414,04	-%449,4
Boston Housing	-%4,84	+%0,39	+%1,23	-%18,13
Relative CPU Performance	+%2,03	+%7,01	-%6,78	-%33,51
Delta Ailerons	-%0,13	+%1,07	-%1,48	-%19,04
Delta Elevators	+%0,36	-%1,39	+%0,01	+%5,14
Servo	+%10,08	+%8,67	+%23,69	+%18,58
İstanbul Stock Exchange	-%7,19	-%1,82	-%15,07	-%23,02
QSAR fish toxicity	-%1,9	-%1,93	+%6,38	+%2,67

Tablo 4.14. GÇK-AÖM2'nin orijinale göre eğitim fazındaki metriklerinin yüzdelik değişimi

Veri Kümesi	Train RMSE			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	-%2,1	-%3,55	-%1,76	+%40,04
Auto-Mpg	-%25,21	-%16,68	-%28,47	-%53,32
Auto Imports Price	-%5,87	-%4,58	-%7,76	-%24,77
Boston Housing	-%15,98	-%12,83	-%17,73	-%70,88
Relative CPU Performance	+%11,93	+%7,33	+%15,11	+%34,9
Delta Ailerons	-%17,92	-%12,3	-%24,9	-%117,51
Delta Elevators	-%11,26	-%4,96	-%22,66	-%243,06
Servo	-%18,48	-%25,26	-%17,82	-%6,26
İstanbul Stock Exchange	-%16,72	-%15,85	-%16,32	+%10,02
QSAR fish toxicity	-%18,75	-%17,3	-%171,67	-%334,05

Tablo 4.15. GÇK-AÖM2'nin orijinale göre test fazındaki metriklerinin yüzdelik değişimi

Veri Kümesi	Test RMSE			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	-%6,6	-%7,53	-%5,18	+%38,09
Auto-Mpg	-%43,39	-%34,72	-%48,58	-%80,38
Auto Imports Price	-%10,23	-%14,01	-%22,68	-%17,23
Boston Housing	-%2,58	-%0,17	-%8,53	-%36,91
Relative CPU Performance	-%74,28	-%108,86	-%113,83	-%32,49
Delta Ailerons	-%23,38	-%17,9	-%32,59	-%136,33
Delta Elevators	-%9,24	-%1,93	-%20,46	-%245,1
Servo	-%7,79	-%5,37	-%17,23	-%53,86
İstanbul Stock Exchange	-%10,12	-%8,56	-%20,17	-%34,88
QSAR fish toxicity	-%51,97	-%47,27	-%177,88	-%205,92

Tablo 4.16. GÇK-AÖM2'in orijinale göre işlem yapma zamanının yüzdelik değişimi

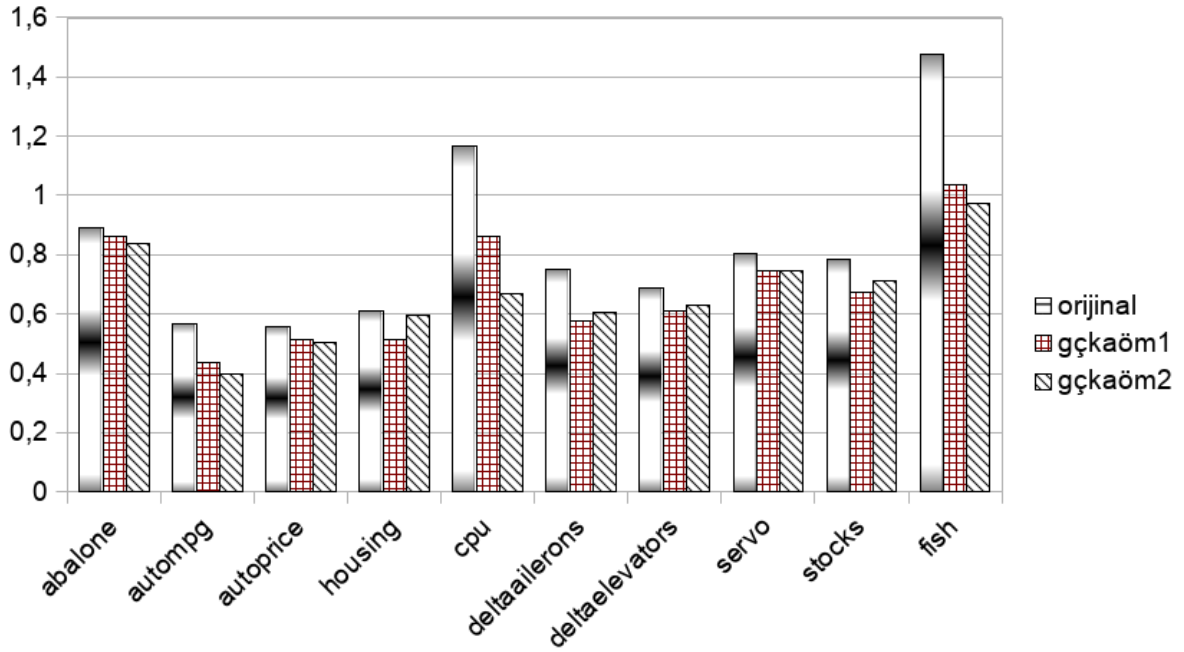
Veri Kümesi	Zaman			
	Ortalama	En iyi	En Kötü	Standart Sapma
Abalone	-%1,09	-%0,94	-%14,84	-%99,87
Auto-Mpg	-%18,34	-%19,35	-%14,98	-%19,91
Auto Imports Price	-%14,66	+%4,09	-%426,22	-%400,97
Boston Housing	-%4,48	-%2,27	+%3,8	-%9,98
Relative CPU Performance	-%1,05	+%4,87	+%1,69	-%2,68
Delta Ailerons	+%0,04	+%0,19	-%1,61	+%25,65
Delta Elevators	+%0,19	-%1,35	-%0,54	+%4,31
Servo	+%8,87	+%10,19	+%13,18	+%10,82
İstanbul Stock Exchange	-%9,79	-%2,48	-%20,39	-%30,46
QSAR fish toxicity	+%0,08	-%2	+%7,25	+%15,31

Tabloların dikkatlice incelenmesi sonucunda, özellikle test aşamalarında RMSE değerlerinin düştüğü gözlemlenmektedir. Bu durum, geliştirilen modellerin eğitim sonrasında gerçek değerlere daha yakın ve tutarlı tahminler yapma yeteneğini işaret etmektedir. Düşük RMSE değerleri, modelin tahminlerinin gerçek değerlere daha yakın olması anlamına gelir ve genel başarımının arttığını gösteren önemli bir ölçüdür.

Ayrıca, tüm aşamalarda standart sapmaların düşmesi, modelin öğrenme sürecinde daha istikrarlı ve tutarlı bir başarımla sergilediğini göstermektedir. Düşük standart sapmalar, modelin farklı veri noktalarına karşı daha tutarlı ve güvenilir tahminlerde bulunduğunu gösterir. Bu durum, modelin genel öğrenme sürecinin daha sağlam olduğunu ve çeşitli veri kümelerine daha iyi adapte olduğunu gösteren bir gösterge olarak değerlendirilebilir.

Tablolardan elde edilen veriler, geliştirilen modellerin öğrenme sürecinde başarılı bir şekilde ilerlediğini, test aşamalarında daha iyi başarımla gösterdiğini ve tahminlerin tutarlılığının arttığını ortaya koymaktadır. Bu bulgular, modelin güvenilirliğini ve uygulanabilirliğini artırarak, regresyon analizindeki etkinliğini vurgulamaktadır.

GÇK-AÖM1 ve GÇK-AÖM2'nin hesaplama için harcadıkları süreler Tablo 4.4. ve Tablo 4.7. üzerinden incelendiğinde eğitim ve test süreçlerindeki kadar başarılı değer düşüşleri görülememektedir. Analiz sonuçları, GÇK-AÖM1 modelinin Auto-Mpg, Auto Imports Price, Boston Housing, Delta Ailerons, İstanbul Srock Exchange ve QSAR fish toxicity veri kümelerindeki ortalama işlem süreleri açısından ÇK-AÖM'ye göre daha hızlı bir performans sergilediğini ortaya koymaktadır. GÇK-AÖM2 modelinin ise Abalone, Auto-Mpg, Auto Imports Price, Boston Housing, Relative CPU Performance, İstanbul Srock Exchange veri kümelerinde ÇK-AÖM'ye göre daha hızlı çalıştığı görülmüştür. Elde edilen gözlemler, GÇK-AÖM'lerin diğer veri kümeleri için ÇK-AÖM'ye yakın hızlarda çalıştığını, Servo veri kümesi için %10'a varan oranda daha yavaş çalıştığını ortaya koymaktadır. GÇK-AÖM'lerde yaşanan yavaş çalışma sorununun temel sebebi temsili girişlerin hesaplanmasında gizli katman çıkış matrisinin kullanılması olarak söylenebilir.



Şekil 4.31. GÇK-AÖM'lerin öğrenme başarımları

Şekil 4.31'deki grafik, yeni geliştirilen ÇK-AÖM'lerin test aşamasındaki başarılarını, orijinaline göre RMSE metriği cinsinden bar grafikte göstermektedir. Grafik üzerindeki değerler, 100 koşmanın ortalaması alınarak elde edilmiştir.

Grafik incelendiğinde, GÇK-AÖM1'in ele alınan problemlerde orijinale göre yüzdeler olarak ortalama [3,48 42,7] aralığında değişen daha üstün bir başarı sağladığı tespit edilmiştir. GÇK-AÖM2'ye göre ise Boston Housing, Delta Ailerons, Delta Elevators ve İstanbul Stock Exchange veri kümelerinde daha başarılı olmuştur.

GÇK-AÖM2'nin ele alınan problemlerde orijinale göre yüzdeler olarak ortalama [2,58 74,28] aralığında değişen daha üstün bir başarı sağladığı tespit edilmiştir. GÇK-AÖM1'e göre ise Abalone, Auto-Mpg, Auto Imports Price, Relative CPU Performance, Servo, QSAR fish toxicity veri kümelerinde daha başarılı olmuştur.

5. SONUÇLAR VE ÖNERİLER

Sonuç olarak, bu çalışma ile 3 farklı çok katmanlı aşırı öğrenme makinesinin farklı veri kümeleri üzerindeki başarımı kapsamlı bir şekilde değerlendirilmiştir. Elde edilen bulgular, bu algoritmaların hangi koşullarda daha etkili olduğunu anlamak adına değerli bir katkı sağlamaktadır. Elde edilen tüm verilerin detaylı bir analizi sonucunda, yeni geliştirilmiş ÇK-AÖM modellerinin regresyon problemlerini çözmede etkili bir performans sergilediği ve bu modellerin orijinal versiyona kıyasla daha başarılı sonuçlar ortaya koyduğu gözlemlenmiştir. Bu bulgular, ÇK-AÖM modifikasyonlarının regresyon analizindeki etkinliğini vurgulayarak, bu yeni yaklaşımların potansiyel üstünlüklerine işaret etmektedir.

Bu çalışmada elde edilen veriler ışığında, GÇK-AÖM1 ve GÇK-AÖM2'nin regresyon problemleri içeren veri kümelerini öğrenme başarısı gözlemlenmiş olmakla birlikte, işlem zamanı açısından bu denli başarılı ve getirili olmadığı gözlemlenmiştir. GÇK-AÖM'lerin her birisi 10 veri kümesinden 6'sında ÇK-AÖM'ye göre daha hızlı hesaplama yapmıştır. Varılan bu sonuç, ileride yapılacak çalışmalarda öğrenme başarısı ve işlem yapma hızı arasında seçim yapmayı gerektirebilir.

Tüm sonuçların titizlikle incelenmesi, yeni geliştirilmiş ÇK-AÖM modellerinin regresyon problemlerini çözme konusundaki üstün yeteneklerini ortaya koymaktadır. Kale ve Karakuzu tarafından yapılan bu iyileştirmeler, orijinal ÇK-AÖM'ye kıyasla daha başarılı sonuçlar elde etmiş ve regresyon analizinde daha etkin bir başarımla sergilemiştir. Elde edilen bu olumlu sonuçlar, modellerin daha karmaşık veri kümelerine uyum sağlama ve genel tahmin gücünü artırma konusundaki etkinliklerini vurgulamaktadır. Bazı veri kümeleri için GÇK-AÖM1 daha başarılı veya daha hızlı çalışırken bazı veri kümeleri için GÇK-AÖM2 daha başarılı veya daha hızlı çalışmıştır. Bu farkların orijinal ÇK-AÖM'de yapılan iyileştirmelerle bir bağlantısı tespit edilememiştir.

Elde edilen bu bulgular, regresyon analizinde kullanılan ÇK-AÖM modellerinin daha geniş bir uygulama alanına sahip olabileceği ve çeşitli endüstriyel veya bilimsel alanlardaki problemlerin çözümünde değerli bir araç olabileceği konusunda önemli bir potansiyeli işaret etmektedir. Bu modellerin görece üstün başarımı, gelecekteki araştırmalara ilham kaynağı olacaktır.

KAYNAKÇA

Abraham, A. (2005). Artificial neural networks. *Handbook of measuring system design*.

Akbilgic, O. (2013). İstanbul Stock Exchange. *UCI Machine Learning Repository*. DOI: 10.24432/C54P4J.

Alçın, Ö. F. (2015). Aşırı öğrenme makinelerinin seyrek geri çatma algoritmaları ile optimizasyonu. (Yayımlanmamış Doktora Tezi). Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.

Ballabio, D. vd. (2019). QSAR fish toxicity. *UCI Machine Learning Repository*. DOI: 10.24432/C5JG7B.

Camacho, R. (2000). *Delta Ailerons*. [Erişim: 18.06.2023, https://www.dcc.fc.up.pt/~ltorgo/Regression/delta_ailerons.html]

Camacho, R. (2000). *Delta Elevators*. [Erişim: 18.06.2023, https://www.dcc.fc.up.pt/~ltorgo/Regression/delta_elevators.html]

Chen, M., Li, Y., Luo, X., Wang, W., Wang, L., & Zhao, W. (2018). A novel human activity recognition scheme for smart health using multilayer extreme learning machine. *IEEE Internet of Things Journal*, 6(2), 1410-1418. DOI: 10.1109/JIOT.2018.2856241.

Dua, D. & Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science

Eshtay, M., Faris, H., & Obeid, N. (2018). Improving extreme learning machine by competitive swarm optimization and its application for medical diagnosis problems. *Expert Systems with Applications*, 104, 134-152. DOI: 10.1016/j.eswa.2018.03.024.

Feldmesser, J. (1987). Computer Hardware. *UCI Machine Learning Repository*. DOI: 10.24432/C5830D.

Harrison Jr, D., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1), 81-102.

Haykin S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall International Inc.

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. *IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)* (Vol. 2, pp. 985-990). Ieee.

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3), 489-501.

Huang, G.-B., Liang, N.-Y., Rong, H.-J., Saratchandran, P. & Sundararajan, N. (2005). On-Line Sequential Extreme Learning Machine. *Proceedings of the IASTED International Conference on Computational Intelligence*. s. 232-237.

Kale, G. A., & Karakuzu, C. (2022). Multilayer extreme learning machines and their modeling performance on dynamical systems. *Applied Soft Computing*, 122, 108861.

Lan, Y., Soh, Y. C. & Huang, G. bin. (2010a). Constructive hidden nodes selection of extreme learning machine for regression. *Neurocomputing*, 73(16–18), 3191–3199. DOI:10.1016/j.neucom.2010.05.022

Lan, Y., Soh, Y. C. & Huang, G. bin. (2010b). Two-stage extreme learning machine for regression. *Neurocomputing*, 73(16–18), 3028–3038. DOI:10.1016/j.neucom.2010.07.012

Luo, J., Vong, C. & Wong, P. (2014). Sparse Bayesian Extreme Learning Machine for Multi-classification. *Neural Networks and Learning Systems, IEEE Transactions on*. 25. 836-843. DOI: 10.1109/TNNLS.2013.2281839.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115-133.

Nash, W. vd. (1995). Abalone. *UCI Machine Learning Repository*. DOI: 10.24432/C55C7W.

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

Parmaksız, H. (2023). RF parmak izi kullanılarak IoT cihaz tanımada meta aşırı öğrenme makinası tabanlı başarımların analizi. (Yayımlanmamış Doktora Tezi). *Bilecik Şeyh Edebali Üniversitesi, Lisansüstü Eğitim Enstitüsü, Bilecik*.

Parmaksız, H., & Karakuzu, C. (2024). Identification of IoT Devices via RF Fingerprint Using Meta Extreme Learning Machines. *Future Generation Computer Systems*, Submitted Jan 03, 2024 (under review).

Qu, B. Y., Lang, B. F., Liang, J. J., Qin, A. K. & Crisalle, O. D. (2016). Two-hidden-layer extreme learning machine for regression and classification. *Neurocomputing*, 175(PartA), 826–834. DOI:10.1016/j.neucom.2015.11.009

Quinlan, R. (1993). Auto MPG. *UCI Machine Learning Repository*. DOI: 10.24432/C5859H.

Rong, H. J., Huang, G. bin, Sundararajan, N., Saratchandran, P. & Rong, H. J. (2009). Online Sequential Fuzzy Extreme Learning Machine for Function Approximation and Classification Problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(4), 1067–1072. DOI:10.1109/TSMCB.2008.2010506

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.

Schlimmer, J. (1987). Automobile. *UCI Machine Learning Repository*. DOI: 10.24432/C5B01C.

Tang, J., Deng, C., & Huang, G. B. (2015). Extreme learning machine for multilayer perceptron. *IEEE transactions on neural networks and learning systems*, 27(4), 809-821.

Ulrich, Karl. (1993). Servo. *UCI Machine Learning Repository*. DOI: 10.24432/C5Q30F.

Zhang, J., Li, Y., Xiao, W. & Zhang, Z. (2020). Non-iterative and Fast Deep Learning: Multilayer Extreme Learning Machines. *Journal of the Franklin Institute*, 357(13), 8925–8955. DOI:10.1016/j.jfranklin.2020.04.033

Zhang, W. B. ve Ji, H. B. (2013). Fuzzy extreme learning machine for classification. *Electronics Letters*, 49(7), 447–448. doi:10.1049/el.2012.3642

Zhu, Q. Y., Qin, A. K., Suganthan, P. N. ve Huang, G. bin. (2005). Evolutionary extreme learning machine. *Pattern Recognition*, 38(10), 1759–1763.
DOI:10.1016/j.patcog.2005.03.028