

T.C.
EGE ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

**MAKİNE ÖĞRENMESİ VE GÖRÜNTÜ İŞLEME
YÖNTEMLERİNİ KULLANARAK
ET SIĞIRLARINDA CANLI AĞIRLIK TAHMİNİ**

Ahmet Erhan KARAHAN

Danışman: Prof. Dr. Hülya ATIL

Zootečni Anabilim Dalı
Biyometri ve Genetik Doktora Programı

İzmir
2020

Ahmet Erhan KARAHAN tarafından **DOKTORA** tezi olarak sunulan “**Makine Öğrenmesi ve Görüntü İşleme Yöntemlerini Kullanarak Et Sığırlarında Canlı Ağırlık Tahmini**” başlıklı bu çalışma EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliği ile EÜ Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi’nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 23.09.2020 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı : Prof. Dr. Hülya ATIL

Raportör Üye : Prof. Dr. Çiğdem TAKMA

Üye : Prof. Dr. Zeynel CEBECİ

Üye : Prof. Dr. Ercan EFE

Üye : Prof. Dr. Aybars UĞUR

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Doktora Tezi olarak sunduğum “Makine Öğrenmesi ve Görüntü İşleme Yöntemlerini Kullanarak Et Sığırlarında Canlı Ağırlık Tahmini” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

23 / 09 / 2020

İmzası

Ahmet Erhan KARAHAN



ÖZET**MAKİNE ÖĞRENMESİ VE GÖRÜNTÜ İŞLEME YÖNTEMLERİNİ
KULLANARAK ET SİĞİRLARINDA CANLI AĞIRLIK TAHMİNİ**

KARAHAN, Ahmet Erhan

Doktora Tezi, Zootečni Anabilim Dalı

Tez Danışmanı: Prof. Dr. Hülya ATIL

Eylül 2020, 171 sayfa

Çiftlik hayvanlarında canlı ağırlığın belirlenmesi ve düzenli olarak takip edilmesi, hastalıkların erken tesbiti, ilaç miktarlarından tasarruf ve dolayısıyla işletme karlılığı açısından büyük önem arz etmektedir. Ancak işletmelerin büyük çoğunluğunda kantar bulunmadığından, ya da varsa bile tartım işinin zaman alıcı, zahmetli oluşu ve hayvanlarda stres oluşturmamasından dolayı işletmelerin çoğunluğunda tartım yapılmamaktadır. Günümüzde klasik tartım yöntemlerinden kaynaklı olumsuzlukların önüne geçmek amacıyla, hayvanların vücut ölçümleri kullanılarak canlı ağırlıklarının belirlenebileceği çeşitli tahmin yöntemleri geliştirilmiştir. Bu yöntemler tartım işlemine kıyasla daha az zaman almaktadır ancak hayvanlarda stres oluşumunun ve insan yaralanmalarının önüne geçilememektedir. Tez çalışmasında bu geleneksel yöntemlere alternatif olarak canlı ağırlık tahmini gerçekleştirmek amacıyla makine öğrenmesi ve görüntü işleme tekniklerine dayalı bir yöntem önerilmiştir. Bu amaçla, hayvanları rahatsız etmeden, yerden 2.65m yükseklikte konumlandırılmış üç boyutlu sabit bir kamera ile alınan derinlik görüntülerinden sığırların canlı ağırlıkları tahmin edilmiştir. Sığırların tamamı aynı kamera çekim açısında olmadığından canlı ağırlık otomatik olarak tahmin edilememiş ancak her bir sığırdan alınan yaklaşık 10'ar görüntüden uygun olanlar seçilerek klasörlendikten sonra bu klasördeki görüntülerinden sığırların üstten sırt (dorsal) alanları otomatik olarak hesaplanmıştır. Ayrıca derinlik görüntüsünden yararlanılarak sığır yükseklikleri de tahmin edilmiş ve elde edilen bu yükseklik değerleri yöntemlerde girdi (bağımsız değişken) olarak kullanılmıştır. Derinlik görüntüsü kullanılarak sığır yükseklerinin tahmin edildiği görüntü işleme yöntemiyle literatüre yeni bir bakış açısı kazandırılması hedeflenmiştir. Çalışmada makine öğrenmesinde kullanılan yapay sinir ağları yaklaşımı detaylı bir şekilde

incelenmiş, çok sayıda eğitim kombinasyonu denenmiştir. YSA'nın yanı sıra hayvancılık alanında kullanımı yeni olan MARS ve CHAID yöntemleri ile de sığır canlı ağırlıkları tahmin edilmiş ve elde edilen bulgular bir fikir vermesi açısından tez kapsamında sunulmuştur. Yapay sinir ağları (YSA), Çok değişkenli uyarlanabilir regresyon eğrileri (MARS), ve Ki kare otomatik etkileşim belirleme (CHAID) yöntemleri ile yapılan canlı ağırlık tahminlerinde ortalama mutlak hata yüzdesi (MAPE) değerleri sırasıyla %4.15, %4.75 ve %5.36 olarak tespit edilmiştir. Her bir yaklaşımla yapılan tahminlerin R^2 değerleri aynı sırayla 0.9467, 0.9334 ve 0.9211 olarak, hata kareler ortalamasının karekökü (RMSE) değerleri sırasıyla 31.63kg, 32.95kg ve 38.42kg olarak belirlenmiştir. Hayvancılıkta daha sık kullanılan çoklu doğrusal regresyon (ÇDR) yöntemiyle ise MAPE, R^2 ve RMSE değerleri sırasıyla %5.02, 0.9266 ve 37.06kg olarak tespit edilmiştir. Yapay sinir ağlarında deneysel çalışmalar sırasında farklı parametre değerleri kullanılarak 972 yapay sinir ağı eğitimi yapılmıştır. Ayrıca YSA'da kullanılan öğrenme algoritmalarından Levenberg&Marquardt (LM), Bayesian regularization (BR) ve uyarlanabilir öğrenme oranlı gradyan yöntemi (GDX), hatayı azaltma bakımından karşılaştırılmış ve ortalama olarak en düşük MAPE değeri (%4.56), LM algoritması kullanılarak iki katmanda sekiz nöron bulunan (2-8-8-1) ağ mimarisiyle yapılan eğitimden elde edilmiştir. Çalışma sonuçları, sığırlarda canlı ağırlığın makine öğrenmesi ve görüntü işleme yöntemleriyle başarılı bir şekilde tahmin edilebileceğini göstermektedir.

Anahtar sözcükler: Canlı ağırlık, CHAID, çoklu doğrusal regresyon, makine öğrenmesi, MARS, et sığı, tahmin, yapay sinir ağları,

ABSTRACT
LIVE WEIGHT ESTIMATION IN BEEF CATTLE USING IMAGE
PROCESSING AND MACHINE LEARNING METHODS

KARAHAN, Ahmet Erhan

PhD in Department of Animal Science

Supervisor: Prof. Dr. Hülya ATIL

September 2020, 171 pages

Determination and monitoring of body weight in animal husbandry are of great importance in terms of profitability, early diagnosis of diseases, saving the amount of drugs used. However, the majority of farms do not weigh livestock for reasons most farms do not have a scale or, even if they do, the weighing is time consuming, laborious, and can cause stress on animals. In recent years, various estimation methods have been developed to determine the live weights of animals by using body measurements in order to prevent negativity from classical weighing methods. Although these methods take less time compared to weighing, stress formation in animals and injuring human cannot be prevented. In this study, an image processing and machine learning-based method is proposed to estimate body weight as an alternative to these traditional methods. For this purpose, the body weights of cattle were estimated by using their three-dimensional depth view taken with a fixed camera positioned at a height of 2.65m from the ground without disturbing the animals. Since all cattle were not at the same camera shooting angle, their body weight could not be predicted automatically, but approximately 10 images were taken from each cattle and after the appropriate ones were selected and filed, the dorsal areas of the cattle were automatically calculated from the images in this folder. In addition, cattle heights values were also estimated using the depth image were used as inputs (independent variables) in the methods. It is aimed to gain a new perspective to the literature by image processing method in which cattle heights are estimated using depth view. In the study, the artificial neural networks approach used in machine learning was examined in detail, and many training combinations were tried in ANN. In addition to ANN, live weights of cattle were estimated with MARS and CHAID methods, which are new to used in animal

husbandry, and the findings are presented to the reader within the scope of the thesis to give an idea. Mean Absolute Percentage Error (MAPE) values of approach were defined as 4.15, 4.75, and 5.36% in the body weight estimations by using Artificial Neural Network (ANN), Multivariate Adaptive Regression Splines (MARS) and Chi-Square Automatic Interaction Detection (CHAID) methods, respectively. The R^2 values of the estimates by each approach were 0.9467, 0.9334, and 0.9211, and the Root Mean Square Error (RMSE) values were 31.63, 32.95, and 38.42kg respectively. MAPE, R^2 and RMSE values were found to be 5.02%, 0.9266 and 37.06 kg, respectively, with the multiple linear regression (MLR) method, which is more frequently used in animal husbandry. During the experimental studies in artificial neural networks, 972 artificial neural network trainings were carried out using different parameter values. In addition to Levenberg-Marquardt (LM), Bayesian Regularization (BR) and Gradient Descent with Adaptive Learning Rate (GDX) algorithms which are among the learning algorithms used in ANN were compared in terms of error reduction and the lowest MAPE value (4.56%) on average was obtained from LM algorithm with 2-8-8-1 network architecture training. The results of the study show that body weight of cattle can be predicted successfully by machine learning and image processing methods

Keywords: Body weight, CHAID, multiple linear regression, beef cattle, machine learning, MARS, prediction, artificial neural network,

ÖNSÖZ

Çiftlik hayvanlarının canlı ağırlıklarının tahmin edilmesi, işletme karlılığı açısından oldukça önemli olduğundan, uzun yıllar bilim insanları tarafından üzerinde durulan bir konu olmuştur. Bu tez çalışmasının yürütülmesindeki amaç et sığırlarının canlı ağırlıklarını onlara rahatsızlık vermeden ve stres oluşturmadan tahmin etmektir. Önerilen yöntemi kullanarak sığırların canlı ağırlığını tahmin edecek yazılıma sahip, otomatik cihazların geliştirilmesi memnuniyet verici olacaktır. Görüntü işleme ve YSA için ekler kısmında verilen komutların ve metot kısmında anlatılan sığır vücut alanının hesaplandığı yöntemin gelecekte hayvancılıkta ya da farklı alanlarda yapılacak çalışmalara ışık tutmasını ve kolaylık sağlamasını diliyorum. Benzer çalışmaların yapılmasıyla canlı ağırlığın tahminleme başarısının artmasını ve devamında geliştirilecek ağırlık tahmin cihazlarını kullanan işletmelerde daha efektif bir üretim yapılmasını ümit etmekteyim. Umarım yaptığım bu çalışma ile Türk tarımına ve yetiştiricilerimize ufak da olsa bir katkı olmuştur.

Bornova/İZMİR

23/09/2020

Ahmet Erhan KARAHAN



İÇİNDEKİLERSayfa

İÇ KAPAK	ii
KABUL ONAY SAYFASI	iii
ETİK KURALLARA UYGUNLUK BEYANI.....	v
ÖZET	vii
ABSTRACT	ix
ÖNSÖZ.....	xi
İÇİNDEKİLER DİZİNİ.....	xiii
ŞEKİLLER DİZİNİ	xvii
TABLolar DİZİNİ.....	xix
SİMGELER VE KISALTMALAR DİZİNİ.....	xxvii
1.GİRİŞ.....	1
2.GENEL BİLGİLER.....	5
2.1 Klasik Yöntemlerle Sığırların Canlı Ağırlığının Tahmin Edilmesi Amacıyla Yapılan Çalışmalar	5
2.2 Yapay Sinir Ağları Kullanılarak Hayvancılıkta Yapılan Çalışmalar	7

İÇİNDEKİLER (devam)

Sayfa

2.3 Görüntü İşleme Yöntemleri Kullanılarak Hayvancılıkta Yapılan Çalışmalar	11
2.4 Görüntü İşleme Yöntemleri ve Yapay Sinir Ağları Kullanılarak Yapılan Çalışmalar	19
2.5 Görüntü İşleme Yöntemleri ve Yapay Sinir Ağları Kullanılarak Hayvancılıkta Yapılan Çalışmalar.....	20
2.6 Yapay Sinir Ağları ve Diğer İstatistiksel Yöntemlerin Karşılaştırıldığı Çalışmalar	22
2.7 Yapay Sinir Ağlarında Kullanılan Öğrenme Algoritmalarının Karşılaştırıldığı Çalışmalar	24
3.GEREÇ VE YÖNTEM.....	28
3.1 Gereç	28
3.1.1 Hayvan materyali.....	31
3.1.2 Görüntü materyali.....	33
3.2 Yöntem.....	34
3.2.1 Görüntü işleme	34
3.2.2 Yapay sinir ağları.....	43
3.2.3 Çok değişkenli uyarlanabilir regresyon eğrileri (MARS)	64
3.2.4 Çoklu doğrusal regresyon analizi	69

İÇİNDEKİLER (devam)

Sayfa

3.2.5 CHAID analizi.....	72
3.2.6 Modellerin performans değerlendirme ölçütleri	78
4.BULGULAR	79
4.1 Yapay Sinir Ağları Bulguları.....	80
4.1.1 Çapraz doğrulamada ham verilerle deneme	80
4.1.2 Çapraz doğrulamada normalize edilmiş verilerle deneme	93
4.1.3 Çapraz doğrulama için tüm durumlara ilişkin özet	105
4.1.4 Verilerin %70 eğitim, %15 test ve %15 doğrulama setine rastgele ayrılması.	105
4.1.5 Rastgele ayırmada ham verilerle deneme.....	106
4.1.6 Rastgele ayırmada normalize edilmiş verilerle deneme.....	118
4.1.7 Rastgele ayırma için tüm durumlara ait özet.....	129
4.2 Çok Değişkenli Uyarlanabilir Regresyon Eğrileri (MARS) Bulguları.....	132
4.3 Çoklu Doğrusal Regresyon Analizi Bulguları.....	132
4.4 CHAID Analizi Bulguları.....	134
4.5 Makine Öğrenmesi Yaklaşımlarının Performans Değerlendirme Ölçütleri. .	141

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
5.TARTIŞMA	143
6.SONUÇ VE ÖNERİLER.....	153
KAYNAKLAR DİZİNİ	156
TEŞEKKÜR.....	169
ÖZGEÇMİŞ	171
EKLER.....	

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
3.1. Erzurum İli Damızlık Koyun Keçi Yetiştiricileri Birliği büyükbaş hayvan tartım kantarı.....	29
3.2. Structure sensor kamera.....	30
3.3. Üç boyutlu kamera ile elde edilen sığır görüntüleri	31
3.4. Kameranın kantar üzerinde konumlandırılması	31
3.5. Görüntülemenin yapılış şeması.	33
3.6. Çalışmaya dâhil edilmeyen bir sığırın görüntüleri.	34
3.7. İkili (binary) bir görüntünün piksel değerleri	35
3.8. Örnek hesaplamanın yapıldığı iki farklı ağırlıktaki sığır görüntüsü.....	36
3.9. Numarası 28 olan hayvana ait kırılan görüntü.....	37
3.10. Numarası 28 olan hayvana ait görüntünün mavi bileşeni.....	37
3.11. Numarası 28 olan hayvana ait siyah beyaza dönüştürülen görüntü.....	38
3.12. Numarası 115 olan hayvana ait siyah-beyaza dönüştürülmüş görüntü.	39
3.13. Yükseklik tahmini için kalibrasyon görüntüleri ve renk skalası.	40
3.14. Uzak ve yakın mesafede renk değişimi	41
3.15. MATLAB pixel region komutu ile piksel değerlerinin görüntülenmesi	42
3.16. Mesafe ve renk değerleri arasındaki ilişkiyi gösteren grafikler	42

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
3.17. Biyolojik sinir hücreleri.	44
3.18. Bir nöronun yapısının şematik gösterimi	44
3.19. Mantıksal AND operatörü ve XOR operatörünün karşılaştırılması.....	53
3.20. Çok katmanlı ileri beslemeli geri yayımlı yapay sinir ağlarının yapısı.	54
3.21. Hata uzayında gradyan azaltmanın gösterimi.	55
3.22. LM algoritması ile eğitimde performans grafiği.....	60
3.23. BR algoritması ile eğitimde performans grafiği	61
3.24. GDX algoritması ile eğitimde performans grafiği	61
3.25. Temel fonksiyonların gösterimi	66
3.26. MARS modelinin grafiksel gösterimi	67
3.27. CHAID analizi diyagramı örneği	76
4.1. LM algoritmasıyla en düşük MAPE değeri için MATLAB ekran görüntüsü	131
4.2. Alan yüzdesi, yükseklik ve canlı ağırlıklar arasındaki ilişkilere ait yanıt yüzey grafiki.	133
4.3. Et sığırlarında canlı ağırlık tahmini için CHAID analizi diyagramı	136

TABLOLAR DİZİNİ

<u>Tablo</u>	<u>Sayfa</u>
3.1 Structure sensor kamera özellikleri	30
3.2 Hayvan materyalinin ırk ve cinsiyete göre dağılımı.....	32
3.3 Hayvan materyaline ait bazı tanımlayıcı istatistikler.	33
3.4 Bazı toplama fonksiyonları.....	45
3.5 Bazı aktivasyon fonksiyonları	46
3.6 Delta öğrenme kuralıyla örnek bir hesaplama	47
3.7 MATLAB’te kullanılabilir öğrenme algoritmaları.....	58
3.8 Çapraz doğrulama ile verinin ayrılması.....	64
3.9 Modellerin doğruluğunu sınamak için kullanılan performans değerlendirme ölçütleri ve formülleri	78
4.1 Etkisi incelenen dokuz farklı parametre grubu.....	80
4.2 Varsayılan parametrelerle yapılan analiz bulguları.	81
4.3 Ağın eğitiminde geçen süreler ve iterasyon sayıları.....	82
4.4 Birinci grup parametrelerle yapılan analiz bulguları.	83
4.5 Birinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	84
4.6 İkinci grup parametrelerle yapılan analiz bulguları.....	84
4.7 İkinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.	85

TABLOLAR DİZİNİ (devam)

<u>Tablo</u>	<u>Sayfa</u>
4.8 Üçüncü grup parametrelerle yapılan analiz bulguları.	85
4.9 Üçüncü grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	86
4.10 Dördüncü grup parametrelerle yapılan analiz bulguları.	87
4.11 Dördüncü grup parametrelerle eğitimde süreler ve iterasyon sayıları.	87
4.12 Beşinci grup parametrelerle yapılan analiz bulguları.	88
4.13 Beşinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.	89
4.14 Altıncı grup parametrelerle yapılan analiz bulguları.	89
4.15 Altıncı grup parametrelerle eğitimde süreler ve iterasyon sayıları.	90
4.16 Yedinci grup parametrelerle yapılan analiz bulguları.....	91
4.17 Yedinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.	91
4.18 Sekizinci grup parametrelerle yapılan analiz bulguları.	92
4.19 Sekizinci grup parametrelerle eğitimde süre ve iterasyon sayıları.	93
4.20 MATLAB varsayılan parametreleri.....	93
4.21 Varsayılan parametrelerle yapılan analiz bulguları.	94
4.22 Varsayılan parametrelerle eğitimde süreler ve iterasyon sayıları.	94
4.23 Birinci grup parametrelerle yapılan analiz bulguları.	95
4.24 Birinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.	96

TABLOLAR DİZİNİ (devam)

<u>Tablo</u>	<u>Sayfa</u>
4.25 İkinci grup parametrelerle yapılan analiz bulguları.....	96
4.26 İkinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.	97
4.27 Üçüncü grup parametrelerle yapılan analiz bulguları.....	97
4.28 Üçüncü grup parametrelerle eğitimde süreler ve iterasyon sayıları.	98
4.29 Dördüncü grup parametrelerle yapılan analiz bulguları.	99
4.30 Dördüncü grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	99
4.31 Beşinci grup parametrelerle yapılan analiz bulguları.	100
4.32 Beşinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	100
4.33 Altıncı grup parametrelerle yapılan analiz bulguları.	101
4.34 Altıncı grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	102
4.35 Yedinci grup parametrelerle yapılan analiz bulguları.	102
4.36 Yedinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.	103
4.37 Sekizinci grup parametrelerle yapılan analiz bulguları.	104
4.38 Sekizinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	104
4.39 MATLAB varsayılan parametreleri.....	106
4.40 Rastgele ayırmada varsayılan parametrelerle yapılan analiz bulguları.	106

TABLOLAR DİZİNİ (devam)

<u>Tablo</u>	<u>Sayfa</u>
4.41 Rastgele ayırmada varsayılan parametrelerle eğitimde süreler ve iterasyon sayıları.....	107
4.42 Rastgele ayırmada birinci grup parametrelerle yapılan analiz bulguları.	108
4.43 Rastgele ayırmada birinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	108
4.44 Rastgele ayırmada ikinci grup parametrelerle yapılan analiz bulguları.....	109
4.45 Rastgele ayırmada ikinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	110
4.46 Rastgele ayırmada üçüncü grup parametrelerle yapılan analiz bulguları. ...	110
4.47 Rastgele ayırmada üçüncü grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	111
4.48 Rastgele ayırmada dördüncü grup parametrelerle yapılan analiz bulguları.	112
4.49 Rastgele ayırmada dördüncü grup parametrelerle eğitimde süre ve iterasyon sayıları.....	112
4.50 Rastgele ayırmada beşinci grup parametrelerle yapılan analiz bulguları. ...	113
4.51 Rastgele ayırmada beşinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	114
4.52 Rastgele ayırmada altıncı grup parametrelerle yapılan analiz bulguları.	115
4.53 Rastgele ayırmada altıncı grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	115

TABLOLAR DİZİNİ (devam)

<u>Tablo</u>	<u>Sayfa</u>
4.54 Rastgele ayırmada yedinci grup parametrelerle yapılan analiz bulguları....	116
4.55 Rastgele ayırmada yedinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	116
4.56 Rastgele ayırmada sekizinci grup parametrelerle yapılan analiz bulguları.	117
4.57 Rastgele ayırmada sekizinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	117
4.58 MATLAB varsayılan parametreleri.....	118
4.59 Normalize veriler ve varsayılan parametrelerle yapılan analiz bulguları....	119
4.60 Normalize veriler ve varsayılan parametrelerle eğitimde süreler ve iterasyon sayıları.....	119
4.61 Rastgele ayırmada normalize veriler ve birinci grup parametrelerle yapılan analiz bulguları.	120
4.62 Normalize veriler ve 1. grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	121
4.63 Rastgele ayırmada normalize veriler ve ikinci grup parametrelerle yapılan analiz bulguları.	121
4.64 Normalize veriler ve 2. grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	122
4.65 Rastgele ayırmada normalize veriler ve 3. grup parametrelerle yapılan analiz bulguları.....	123

TABLOLAR DİZİNİ (devam)

<u>Tablo</u>	<u>Sayfa</u>
4.66 Normalize veriler ve 3. grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	123
4.67 Rastgele ayırmada normalize veriler ve 4. grup parametrelerle yapılan analiz bulguları.	124
4.68 Normalize veriler ve 4. grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	124
4.69 Rastgele ayırmada nrm. veriler ve beşinci grup parametrelerle yapılan analiz bulguları.	125
4.70 Normalize veriler ve beşinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.....	125
4.71 Rastgele ayırmada nrm. veriler ve altıncı grup parametrelerle yapılan analiz bulguları.	126
4.72 Normalize veriler ve altıncı grup parametrelerle eğitimde süre ve iterasyon sayıları.....	127
4.73 Rastgele ayırmada normalize veriler ve 7. grup parametrelerle yapılan analiz bulguları.	127
4.74 Rastgele ayırmada normalize veriler ve 7. grup parametrelerle yapılan analiz bulguları.	128
4.75 Rastgele ayırma, normalize veriler ve 8. grup parametrelerle yapılan analiz bulguları.	129
4.76 Rastgele ayırma, normalize veriler ve 8. grup parametrelerle yapılan analiz bulguları.	129

TABLÖLAR DİZİNİ (devam)

<u>Tablo</u>	<u>Sayfa</u>
4.77 MARS analizi ile yapılan tahmin için performans ölçütü değerleri	132
4.78 Çoklu doğrusal regresyon analizi ile yapılan tahmin için performans ölçütü değerleri	134
4.79 CHAID analizi ile yapılan tahmin için performans ölçütü değerleri.....	134
4.80 Performans değerlendirme ölçütleri bakımından makine öğrenmesi yaklaşımları.....	141



SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltma</u>	<u>Açıklama</u>
BR	Bayesian regularization algoritması
CHAID	Ki-kare otomatik etkileşim belirleme analizi
ÇDR	Çoklu doğrusal regresyon
ÇKA	Çok katmanlı algılayıcı
GD	Gradyan azaltma (Gradient descent)
GDX	Uyarlanabilir öğrenme oranlı gradyan yöntemi
GIY	Görüntü işleme yöntemleri
KS	Katman sayısı
LM	Levenberg-Marquardt algoritması
LR (λ)	Öğrenme oranı (Learning rate)
MAPE	Ortalama mutlak hata yüzdesi
MARS	Çok değişkenli uyarlanabilir regresyon eğrileri
ML	Makine öğrenmesi (Machine learning)
ÇKA	Çok katmanlı algılayıcı (Multilayer perceptron)
HKO	Hata kareler ortalaması
MU	Momentum güncelleme katsayısı
NS	Nöron sayısı
R^2	Belirleme katsayısı (Coefficient of determination)
EDS	En iyi döngü sayısı

SİMGELER VE KISALTMALAR DİZİNİ (devam)

<u>Kısaltma</u>	<u>Açıklama</u>
BH	Bağıl hata (Relative error)
RMSE	Hata kareler ortalamasının karekökü
SCG	Scaled conjugate gradient algoritması
YSA	Yapay sinir ağırları



1. GİRİŞ

İnsanoğlu için tarihin ilk dönemlerinde “tek, çift ve çok” terimleri, nesneleri saymak için yeterli olmuştur. İnsanlar ticaretle tanıştıklarında, takas için çentikler atmayı öğrenmişlerdir. Tarımla uğraşmaya başladıklarında ise koyunlarının sayısını kil madalyonlara çeşitli semboller çizerek belirlemişlerdir. Ancak her koyun için bir madalyonu işaretlemek ya da çentik atmak zor olduğundan bundan yaklaşık beş bin yıl önce Sümerler, kil tabletlere not edebildikleri 60’lık sayı sistemini bularak ilk rakamların mucidi olmuşlardır (Şeker, 2019). Kısacası tarihin ilk zamanlarından beri insanlar hayatlarını kolaylaştırmak için sayılara, dolayısıyla matematiğe ihtiyaç duymuşlardır.

Medeniyetler geliştikçe insanlar yaşadığı gezegeni tanımlamaya çalışmış, evrende bir düzenin ve çeşitli yasaların varlığını keşfetmişlerdir. Bunları anlamlandırmak için yıllar boyunca birçok bilim insanı daha gelişmiş matematiksel yöntemler bulmak üzere birbirleri ile yarışmıştır. Bu yöntemler karmaşık hesaplamalar içerdiğinden abaküsle yetinmeyerek çeşitli hesaplama araçları geliştirmişlerdir. Geçtiğimiz asırda insanoğlu hesaplamanın çok zor hatta bazı durumlarda imkânsız olduğu bazı problemlere saniyenin onda biri gibi çok kısa zamanlarda çözümler üretecek güçlü bir makine geliştirmiş ve adına da hesaplayıcı anlamında bilgisayar (“Computer”) ismini vermiştir. Ancak bu güçlü hesaplama araçlarına öğrenme becerisinin kazandırılması yıllar süren çalışmalar neticesinde gerçekleşmiştir. Öğrenme insana özgü bir olay olduğundan bu becerinin bilgisayarlara ya da makinalara kazandırılması için insan beyninin çalışma şeklinin bilinmesine ihtiyaç duyulmuştur.

İnsan beyni üzerinde binlerce yıldır araştırmalar yapılmasına rağmen henüz tam anlamıyla çalışma prensibi ve yapısı çözülememiş, dünya üzerindeki en karmaşık yapıya sahip, öyle ki anlaşılabilmesinin tek yolu yine kendisi olan dünyanın en esrarengiz nesnesidir (Winston, 2011). Bu kadar karmaşık bir organın çalışma prensibi taklit edilerek makinelerin insanlara benzer işler yapabilmesi bilim insanlarınca uzun yıllar arzu edilmiş, sinema tarihinde de bilim kurgu filmlerinin vazgeçilmez konuları arasında yer almıştır. İnsan beyninin çalışma prensibi ile ilgili atılan ilk bilimsel adım bir nörofizyolog olan Warren McCulloch ve genç

matematikçi Walter Pitts tarafından 1943 yılında basit bir sinir ağıının modellenmesi olmuştur. Bu tarihten sonra yapay zeka alanındaki çalışmalar hız kazanmıştır.

Karmaşık algoritmalara sahip olan yapay zeka yöntemleri yirminci yüzyılın ortalarında hayatımıza girmiştir ve bilgisayar biliminin son on yılda daha da gelişmesi ve yaygınlaşması ile kullanılabilirliği giderek artmıştır. Bu yöntemler günümüzde savunma, bankacılık, otomotiv, uzay araştırmaları, ekonomi, tıp ve elektronik gibi birçok alanda yaygın bir şekilde kullanılmaktadır (Öztemel, 2003). Her geçen gün gelişmekte olan yapay zeka teknolojileri, en uygun uçuş rotalarının hesaplanmasından, akıllı arabaların trafikte kaza yapmadan ilerleyebilmelerine imkan veren yazılımlarına, betonun basınç dayanımının tahmin edilmesinden, yüz tanıma sistemlerine, klasik müzikte makam tanımadan parmak izinden yüz tanımaya kadar çok geniş bir yelpazede çalışılan popüler bir konudur ve hayvancılık alanında da kullanımı giderek yaygınlaşmaktadır.

Yapay zekanın bir alt dalı olan yapay sinir ağları (YSA), çoklu doğrusal regresyon ve lojistik regresyon gibi klasik istatistiksel yöntemlerden daha yüksek doğrulukla sonuçlar vermesi ve parametrik testlerin varsayımlarını gerektirmemesi nedeniyle birçok alanda olduğu gibi biyometri ve genetik alanında da kullanılan önemli bir araçtır. Çalışmada YSA'nın yanı sıra çoklu doğrusal regresyon, kümeleme analizi ve ayırma analizi gibi klasik yöntemlerden daha iyi sonuçlar üreten çok değişkenli uyarlanabilir regresyon eğrileri (Multivariate Adaptive Regression Splines-MARS) ve Ki-kare otomatik etkileşim belirleme (Chi-Square Automatic Interaction Detection-CHAID) analizleri ile de sığır canlı ağırlıkları tahmin edilmiştir. Bu kapsamda YSA, MARS ve CHAID analizleri görüntü işleme yöntemleriyle birlikte canlı ağırlık tahmininde kullanılarak klasik yöntemlere göre zaman, iş gücü ve tahmin doğruluğu bakımından daha üstün sonuçlara ulaşılması hedeflenmiştir.

Hayvancılık, insanlık tarihinde çok eski ve önemli bir yere sahiptir. Öyle ki yakın tarihe kadar insanlık medeniyetine ait temellerin tarımla atıldığı düşünülmüyordu. İnsanların tarım için yerleşik hayata geçtiği, bundan sonraki süreçlerde hayatlarında dinin etkili olduğu ve tapınaklar inşa ettikleri kabul ediliyordu. Ancak 1995 yılında Şanlıurfa'da gün yüzüne çıkarılan ve Mısır

piramitlerinden 7500 yıl önce, M.Ö. 10000 yılında inşa edilmiş, 2018 yılında UNESCO tarafından Dünya Mirası kalıcı listesine alınan Göbeklitepe tapınağı, tarım, yerleşik hayat, din sıralamasını değiştirmiştir (Özkan, 2016). İnsanların tarım yaptığına dair hiçbir tarihsel bulgunun olmadığı bir dönemde bu tapınağın yapılmış olması ve kazılar esnasında bulunan, tapınak duvarlarına işlenmiş yabani hayvan figürleri bu tapınağı inşa eden insanların avcı-toplayıcı olduğunu göstermektedir. Sıralaması değişmiş olsa bile tarım ve dolayısıyla hayvancılık, insanoğlu için tarihin her döneminde önemli bir yere sahip olmuştur. İnsanoğlunun kırmızı et elde etmek için geliştirdiği ve Afrika'da bulunan taş araç gereçlerin ilk örneklerinin 3-4 milyon yıl öncesine ait olduğu düşünülmektedir. Önceleri avlanarak beslenen insanlar daha sonra evcil hayvanlar yetiştirmeyi öğrenmiş ve bunlardan kırmızı et elde etmişlerdir. İnsanlığın her döneminde et önemli bir rol oynamış, öyle ki hayvanlar için otlak bulunmadığı durumlarda insanlar göç etmiş, yurtlarını terk etmiştir. Kırmızı et aynı zamanda insan sağlığı açısından en önemli besinlerden biridir. İçerdiği yüksek protein, vitamin, antioksidan ve diğer besin elementleri nedeniyle özellikle gelişmekte olan çocuklar, yaşlılar ve hamileler tarafından tüketilmesi elzemdir. Araştırmalar bu gruptaki insanların kırmızı et tüketmediğinde beyin işlevlerinde bazı problemlerin ortaya çıktığını göstermektedir (Mann, 2018). Ayrıca kişi başına kırmızı et tüketimi ülkelerin gelişmişlik seviyelerinin bir göstergesi olarak kabul görmektedir.

Ülkemizde kırmızı et temininde kullanılan kaynakların başında şüphesiz sığır gelmektedir. Sığır yetiştirilen işletmelerde canlı ağırlığın takibi bazı hastalıkların erken tanısı, kullanılacak ilaç miktarının belirlenmesi ve işletme karlılığı açısından önemli konular arasındadır. Ancak ülkemizde sığır işletmelerinin çoğunda canlı ağırlık düzenli olarak takip edilmemektedir. Bunun en önemli nedeni ise işletmelerde kantar bulunmamasıdır. Kantar bulunan işletmelerde dahi tartım süreci zahmetli ve hayvanlarda strese sebebiyet veren bir süreç olduğundan canlı ağırlık takibi yapılmamaktadır.

Bu çalışmanın amacı; bilgisayar bilimindeki gelişmeler ışığında et sığırı işletmeleri için büyük öneme sahip olan canlı ağırlık takibinin kolaylaştırılmasını sağlamak üzere sığırlarda strese neden olmadan, uzaktan alınan üç boyutlu görüntülerinden onların canlı ağırlıklarını tahmin etmektir. Bu amaç doğrultusunda

canlı ağırlık tahmininde kullanılan makine öğrenmesi yaklaşımlarından yapay sinir ağları, çok değişkenli uyarlanabilir regresyon eğrileri, çoklu doğrusal regresyon ve Ki-kare otomatik etkileşim belirleme yöntemleri incelenmiştir. Bu bağlamda, yapay sinir ağlarında kullanılan öğrenme algoritmalarından Levenberg-Marquardt (LM), Bayes düzenlemesi (Bayesian regularization-BR) ve uyarlanabilir öğrenme oranlı gradyan yöntemi (Gradient Descent with Momentum and Adaptive Learning Rate-GDX) algoritmalarının tahmin başarıları irdelenmiştir. Ayrıca yapay sinir ağlarında kullanılan momentum güncelleme katsayısı (Momentum Update-MU), en küçük gradyan (Minimum Gradient-GD), döngü sayısı (Number of iteration) ve öğrenme oranı (Learning Rate-LR, λ) ile katman ve nöron sayılarının hatayı azaltmadaki etkileri belirlenmeye çalışılmıştır.



2. GENEL BİLGİLER

Yapay zekanın bir alt dalı olan makine öğrenmesi, çeşitli matematiksel ve istatistiksel yöntemleri kullanarak bilgisayarların insan zekasına benzer şekilde öğrenebilmesine ve daha önce görmediği bir problemle karşılaştığında çeşitli tahminler ve sınıflandırmalar yapabilmesine imkan veren sistemler olarak tanımlanabilir.

Makine öğrenmesinde kullanılan yapay sinir ağları ve görüntü işleme gibi yöntemlerin tarım alanında özellikle doğrusal olmayan veri yapılarının modellenmesinde iyi birer araç olduğu ve çoklu doğrusal regresyon, lojistik regresyon gibi klasik istatistiksel yöntemlerden daha iyi sonuçlar verdiği birçok araştırmada (Bahreini and Aslaminejad, 2010; Takma vd., 2012; Atıl ve Akıllı, 2015; Sapmaz ve Yercan, 2017) belirtilmiştir.

Bu bölümde öncelikle hayvancılıkta klasik yöntemler kullanılarak canlı ağırlığın tahmin edildiği çalışmalar, daha sonra YSA kullanılarak hayvancılıkta yapılmış çalışmalar, bundan sonra hayvancılıkta canlı ağırlığın YSA ve GİY ile tahmin edildiği çalışmalar ve tarımsal alandaki diğer çalışmalardan bir literatür özeti derlenerek okuyucuya sunulmuştur.

2.1 Klasik Yöntemlerle Sığırların Canlı Ağırlığının Tahmin Edilmesi Amacıyla Yapılan Çalışmalar

Canlı ağırlığın bilinmesi hayvancılık işletmelerinde çeşitli hastalıkların erken tespiti ve işletme karlılığı açısından büyük önem taşımaktadır. Hayvanların tartılarak ağırlıklarının belirlenmesi işlemi zahmetli olmasının yanı sıra hayvanlarda strese sebep vermesi nedeniyle tercih edilmemektedir. Ayrıca işletmelerin büyük çoğunluğunda tartım için kantar bulunmamaktadır. Bu nedenlerle bilim insanları 1900'lü yılların başından beri göğüs çevresi, sağrı yüksekliği, cidago yüksekliği, vücut uzunluğu, göğüs derinliği, kalça genişliği, karın çevresi ve vücut derinliği gibi çeşitli vücut ölçümlerini kullanarak hayvanların canlı ağırlıklarını tahmin etmeye çalışmışlardır. Canlı ağırlıkla en yüksek ilişkili olan göğüs çevresi geleneksel olarak hayvanların canlı ağırlıklarının tahmin edilmesinde en yaygın kullanılan vücut ölçümüdür. Aşağıda bu alanda Çoklu Doğrusal Regresyon Analizi kullanılarak yapılmış bazı çalışmalara yer verilmiştir.

Branton and Salisbury (1946) 25 baş siyah alaca ve 25 baş esmer olmak üzere toplam 50 sığırın vücut ölçümlerini kullandıkları araştırmada, göğüs çevresi ile canlı ağırlık arasındaki belirleme katsayısını $R^2=0.953$ olarak tespit etmişlerdir. Araştırmacılar sadece siyah alacalarda bu değer $R^2=0.910$ ve sadece esmer sığırlarda $R^2=0.918$ olarak tespit edildiğini bildirmişlerdir.

Tüzemen et al. (1995) 310 baş siyah alaca buzağının canlı ağırlığını tahmin etmek için çeşitli vücut ölçümlerini (göğüs çevresi, vücut uzunluğu, göğüs derinliği, cidago yüksekliği) kullanmışlardır. Çalışmada canlı ağırlıkla en yüksek ilişkili ($R^2=0.812$) vücut ölçümünün göğüs çevresi olduğunu tespit etmişlerdir. Buzağının doğum ağırlığı ile göğüs çevresi arasındaki belirleme katsayısını dişi buzağılarda $R^2=0.638$, erkek buzağılarda $R^2=0.743$ olarak belirlemişlerdir. Araştırmada sütten kesim ağırlığı ile göğüs çevresi arasında dişi buzağılarda $R^2=0.643$, erkek buzağılarda $R^2=0.812$ olarak verilmiştir. Altı aylık ağırlık ile göğüs çevresi arasında ise dişilerde $R^2=0.569$, erkek buzağılarda $R^2=0.731$ olarak tespit etmişlerdir. Çalışma sonunda araştırmacılar sadece göğüs çevresi kullanılarak doğum, sütten kesim ve altı aylık canlı ağırlığın siyah alaca buzağılarda doğru bir şekilde tahmin edilebileceğini bildirmişlerdir.

Bozkurt (2006), Süleyman Demirel Üniversitesi Tarım İşletmesinde bulunan 292 baş esmer sığıra ait çeşitli vücut ölçümlerini (göğüs çevresi, vücut uzunluğu, vücut derinliği, kalça genişliği, sağrı yüksekliği) canlı ağırlığı tahmin etmek için kullanmıştır. Araştırmada, canlı ağırlığı tahmin etmek için tüm vücut özelliklerinin kullanıldığı modelde $R^2=0.941$, göğüs çevresi, kalça genişliği ve vücut derinliği kullanılan modelde $R^2=0.94$, göğüs çevresi ve vücut uzunluğunun kullanıldığı modelde ise $R^2=0.932$ olarak belirlemiştir. Sadece göğüs çevresi kullanıldığı durumda ise $R^2=0.899$ olarak belirlendiğini bildirmiştir.

Özlütürk vd. (2006) Doğu Anadolu Kırmızısı sığırlarda göğüs çevresi ile canlı ağırlığın tahmin edilmesi üzerine yürüttükleri araştırmalarında, doğum ağırlığı 3-24 aylık ve ergin sığırlar olmak üzere farklı yaşlardaki buzağının canlı ağırlıklarını tahmin etmişlerdir. Çalışma sonunda göğüs çevresi ile doğum ağırlığı arasındaki belirleme katsayısını $R^2=0.576$ olarak tespit etmişlerdir. Araştırmada 3, 6, 9, 12, 15, 18 ve 24 ay olmak üzere yedi dönemde erkek esmer sığırlarda canlı

ağırlık ile göğüs çevresi arasındaki belirleme katsayısını sırasıyla 0.446, 0.839, 0.797, 0.891, 0.941, 0.924 ve 0.924 olarak tespit etmişlerdir. Son olarak ergin sığırlarda bu değerin 0.815 olarak belirlendiğini ve göğüs çevresinin canlı ağırlığı tahmin etmede tek başına yeterli bir vücut ölçümü olduğunu bildirmişlerdir.

Koç ve Akman (2007), 1-1.5 yaşlı 18 baş siyah alaca tosunun çeşitli vücut ölçümlerini (göğüs çevresi, sağrı yüksekliği, cidago yüksekliği vs.) kullanarak canlı ağırlıklarını tahmin etmişlerdir. Çalışmada göğüs çevresi, cidago yüksekliği ve sağrı yüksekliği ile canlı ağırlık arasındaki belirleme katsayılarını sırasıyla 0.848, 0.629 ve 0.77 olarak belirlemişlerdir. Araştırmacılar göğüs çevresinin canlı ağırlığı tahmin etmede tek başına yeterli olacağını, cidago yüksekliği ve sağrı yüksekliğinin kullanılması ile canlı ağırlığın tahmin edilmesinde doğruluğun artacağını bildirmişlerdir.

Gruber et al. (2018), Avusturya'daki 167 süt işletmesinden alınan 3750 baş simental, 1056 baş siyah alaca ve 1500 baş esmer sığırın canlı ağırlıklarını tahmin etmek için 11 vücut ölçümünü (göğüs çevresi, karın çevresi, kalça genişliği vs.) kullanmışlardır. Araştırmacılar, göğüs çevresi ve karın çevresi kullanılarak elde edilen model ile yapılan tahminde en yüksek belirleme katsayısını laktasyon döneminde $R^2=0.82$, kuru dönemde $R^2=0.80$ olarak tespit etmişlerdir. Bunun dışında göğüs çevresi ve karın çevresi kullanılarak elde edilen modelde RMSE değerinin 32.5kg, göğüs çevresi, karın çevresi ve kalça genişliğinin kullanıldığı modelde RMSE değerinin 30.4kg olarak tespit edildiğini bildirmişlerdir.

2.2 Yapay Sinir Ağları Kullanılarak Hayvancılıkta Yapılan Çalışmalar

Adamczyk et al. (2005) yaptıkları çalışmada besi sığırlarının büyüme verilerinden onların karkas performans özelliklerini yapay sinir ağları ile tahmin etmeye çalışmışlardır. Çalışmada 120 günlük beside 500 kg ağırlığa kadar ulaşan 104 baş et sığırının ırk, yaş, kesim ve besi dönemindeki canlı ağırlığı yapay sinir ağlarında girdi değişkenleri, sıcak karkas ağırlığı, soğuk karkas ağırlığı gibi 23 karkas performans özelliğini çıktı değişkenleri olarak tanımlamışlardır. Araştırmacılar yapay sinir ağlarında ileri beslemeli geri yayılım algoritması, bir gizli katmanda 30 nöron ve lojistik aktivasyon fonksiyonunu kullanmışlar ve döngü

sayısını 10000 ile sınırlandırmışlardır. Çalışma sonunda araştırmacılar et sığırlarının sıcak karkas ağırlığının yapay sinir ağları ile başarılı bir şekilde ($R^2=0.941$) tahmin edilebildiğini belirtmişlerdir. Ayrıca araştırmacılar soğuk karkas ağırlığı, etin kuru madde ve protein içeriği, etin su tutma kapasitesi gibi diğer karkas performans özelliklerinin de ($R^2=0.656-0.792$) tahmin edilebileceğini bildirmişlerdir.

Fernandez et al. (2007) süt keçilerinde süt verimini tahmin etmek amacıyla 300 baş süt keçisi sürüsüne sahip bir işletmeden homojen yapıya sahip bir grup oluşturmak amacıyla 35 baş süt keçisine ait verileri kullanmışlardır. Araştırmacılar beslenme, süt üretimi, laktasyon süresi, ilk kontrol ile doğum sonrasına kadar geçen sürelerden oluşan verileri kullandıkları çalışmada yapay sinir ağlarının tahminleme başarısını incelemişlerdir. Çalışma sonunda performans kriteri olarak aldıkları hata kareler ortalaması (MSE) değerinin 0.36 olarak belirlendiğini, yapay sinir ağlarının süt verimini tahminlemede uygun bir yöntem olduğunu bununla birlikte bu fikrin güçlendirilmesi için bu konuda daha fazla çalışma yapılmasının gerektiğini bildirmişlerdir.

Hassan et al. (2009) yapay sinir ağlarının süt parametrelerindeki değişikliklere dayanarak majör ve minör mastitis patojenlerini saptama potansiyelini araştırmak amacıyla yürüttükleri çalışmada, 4852 süt örneğinden oluşan bir veri setini kullanarak iki farklı öğrenme modeli (eğitilmiş ve eğitimsiz) ile yapay sinir ağlarının başarısını test etmişlerdir. Çalışma sonunda iki modelin de enfeksiyon etkeninin bulaştığı örneklerle bulaşmayanlar arasındaki ayrımı etkili bir şekilde yapabildiği, eğitimsiz öğrenme modeline sahip yapay sinir ağlarının geleneksel mikrobiyolojik yöntemlerden elde edilen sonuçlarla daha iyi bir uyum sağladığı sonucuna vardıklarını bildirmişlerdir.

“Yapay sinir ağları ve tarımda bir uygulama” isimli doktora çalışmasında Küçükönder (2011), Kahramanmaraş ili Dere köy bölgesinde bulunan 30 başlık kapalı duraklı süt sığırı ahırında ölçülen barınak içi sıcaklık nem indeksi değerleri, kuru termometre sıcaklığı ve çiğlenme noktası sıcaklığı değişkenlerinden oluşan veri setini değerlendirmiştir. Sıcaklık ve nem indeksi değerlerini literatürde Frank Wierama tarafından önerilen formüller aracılığıyla hesapladıktan sonra yapay sinir

ağlarını da kullanarak yaptığı tahminin sonuçlarını karşılaştırmalı olarak incelemiştir. Araştırmacı YSA'da farklı modeller kullanıldığında tahminin gücü değişse de klasik yöntemle elde edilen değerlerle yüksek oranda tutarlı olduğunu bildirmiştir. Ek olarak çalışmada ağ tasarımı, ara katman sayısı, katmanlarda bulunacak nöron sayısı, aktivasyon fonksiyonu seçimi, mimari yapının oluşturulması ve ağın eğitiminde kullanılacak olan öğrenme algoritmalarının seçimi gibi işlem adımlarını detaylı olarak incelemiştir. Çalışma sonunda 13 farklı eğitim algoritmasını inceleyen araştırmacı en az hata (MAPE değeri % 0.0061) ile yapılan tahmini Bayesian Regularization algoritmasını kullandıkları durumda elde ettiklerini bildirmiştir.

Akıllı ve Atıl (2014) süt sığırcılığında yapay zekânın kullanımını tanıttıkları çalışmada, tarımsal alanda bulanık mantık ve yapay sinir ağları ile yapılmış çalışmalara yer vermişlerdir. Yapay zekâ yöntemlerinin süt sığırcılığı alanında yetiştirme ve besleme ile ilgili araştırmalarda kullanılan bazı istatistiksel analiz yöntemlerine alternatif yöntemler olabileceğini, bulanık mantığın karar destek sistemlerinde, yapay sinir ağlarının ise tahminleme amacıyla kullanılabileceğini bildirmişlerdir.

Ali et al. (2015) çeşitli vücut özelliklerinden yararlanarak Harnai koyunlarında canlı ağırlığı tahmin etmek amacıyla yapay sinir ağları ve karar ağacı yöntemlerinden Sınıflandırma ve regresyon ağacı (Classification And Regression Tree-CART), CHAID ve Exhaustive CHAID analizlerinin etkinliğini, karşılaştırmalı olarak incelemiştir. Çalışmada 161 baş Harnai koyunundan alınan vücut uzunluğu, göğüs çevresi gibi dokuz farklı parametreyi girdi değişkenleri olarak, canlı ağırlığı ise çıktı değişkeni olarak kullanmışlardır. Veri setini %70 eğitim, %20 doğrulama ve %10 test seti olacak şekilde ayıran araştırmacılar yöntemlerin tahmin performansını değerlendirmede RMSE, mutlak bağıl hata (Relative Absolute Error-RAE), R^2 ve düzeltilmiş R^2 değerlerini kullanmışlardır. Çalışma sonunda bu değerleri CHAID için sırasıyla 1.509, 0.0564, 0.8377 ve 0.8335 olarak, Exhaustive CHAID için sırasıyla 1.488, 0.0556, 0.8421 ve 0.8381 olarak, CART için 1.560, 0.0583, 0.8264 ve 0.8220 olarak ve yapay sinir ağları için bu değerleri sırasıyla 1.589, 0.0594, 0.8199 ve 0.8154 olarak hesaplamışlardır. Araştırmacılar tahmin başarısına göre yöntemlerin Exhaustive

CHAID, CHAID, CART ve YSA şeklinde sıralanabileceğini, karar ağacı yöntemlerinin canlı ağırlığı tahmin etmedeki başarısının YSA'dan daha yüksek olduğunu bildirmişlerdir.

Eyduran et al. (2017) Pakistanda Yerli Beetal ırkı keçilerin canlı ağırlıklarını tahmin etmek için Radyal Temelli Fonksiyonlu (Radial Bases Function-RBF) yapay sinir ağları, çoklu doğrusal regresyon, CHAID, CART, çok katmanlı algılayıcı modeli 1 (Multilayer Perceptron-MLP1) ve çok katmanlı algılayıcı modeli 2 (Multilayer Perceptron-MLP2) olmak üzere altı farklı yöntem ve modelin etkinliğini incelemişlerdir. Çalışmada 205 baş keçiye ait canlı ağırlık verisi kullanan araştırmacılar göğüs çevresi, boyun uzunluğu ve vücut uzunluğu gibi altı farklı parametre kullanarak canlı ağırlığı tahmin etmeyi hedeflemişlerdir. Çalışma sonunda araştırmacılar en yüksek R^2 değerinin radyal temelli fonksiyon kullanıldığında 0.7470 olarak hesaplandığını, çoklu doğrusal regresyonda bu değer 0.7430 olduğunu, CHAID, MLP2, CART ve MLP1 modellerinde bu değer sırasıyla 0.7183, 0.6954, 0.6743 ve 0.6722 olarak tespit edildiğini bildirmişlerdir. Çalışmada elde edilen en düşük MAPE değerinin (%6.8779) çoklu doğrusal regresyon yöntemi kullanıldığında, en yüksek MAPE değerinin ise (%8.1208) CART yöntemi kullanıldığında tespit edildiğini bildirmişlerdir.

Pakistanda yetiştirilen 107 baş Mangali koçuna ait altı vücut özelliğini kullanarak (vücut uzunluğu, göğüs çevresi, testis ölçümleri vb.) koçların canlı ağırlıklarını tahmin etmek amacıyla Çelik vd. (2017) altı farklı yöntemi (CART, CHAID, Exhaustive CHAID, MARS, MLP) karşılaştırmışlardır. Çalışma sonunda tahmin hatasını azaltma bakımından yöntemleri en iyiden kötüye doğru CART, CHAID ve Exhaustive CHAID, MARS_2, MARS_1, RBF, MLP şeklinde tespit ettiklerini bildirmişlerdir. Araştırmacılar bu yöntemlerle yapılan tahminlerden MAPE değerlerini sırasıyla %5.797, %6.471, %6.486, %7.145, %7.142, %7.572 ve %7.856 olarak belirlemişlerdir. Ayrıca araştırmacılar bu yöntemler arasından yalnızca MARS yönteminde bir tahmin modeli bulunduğunu bildirmişlerdir.

Karadaş et al. (2017) yerli Akkaraman koyunlarının laktasyon süt verimlerinin tahminlenmesi amacıyla YSA ve veri madenciliği algoritmalarını karşılaştırmalı olarak incelemişlerdir. Bu amaçla 250 Akkaraman işletmesinden

elde edilen veterinerlik giderleri, sağım sayısı, laktasyon süresi gibi 10 farklı girdi parametresini kullanarak elde ettikleri verileri %80 eğitim %20 test seti olacak şekilde ayırmışlardır. Çalışma sonunda laktasyon süt verimini tahmin etmedeki başarılarına göre algoritmaların Exhaustive CHAID, CHAID, YSA, GLM ve CART şeklinde sıralanabileceğini, elde edilen RMSE değerlerinin aynı sırayla 18.41kg, 18.41kg, 18.78kg, 19.21kg, 23.08kg olarak tespit edildiğini ve bu sebeple karar ağacı algoritmaları kullanıldığında yapay sinir ağlarına göre daha düşük hatayla tahmin yapılabildiğini bildirmişlerdir.

Takma ve Gevrekçi (2018) yumurta verimini tahmin etmek amacıyla 1955 yumurtacı tavuktan alınan verilerin %75'ini eğitim %25'ini test setine ayırmışlardır. Kuluçka dönemi, canlı ağırlık, eşeyssel olgunluğa ulaşma, sıra ve eşeyssel olgunluk ağırlığının girdi olarak kullanıldığı çalışmada çıktı olarak yumurta üretimi tahmin edilmiştir. Araştırmacılar YSA'nın performansını değerlendirmede R^2 , RMSE ve MAD değerlerini kullanmış, test setinde bu değerleri sırasıyla 0.82, 0.447 ve 3.353 olarak tespit etmişlerdir. Çalışma sonunda YSA'nın yumurta verimini tahmin etmede güçlü araçlardan biri olduğunu bildirmişlerdir.

2.3 Görüntü İşleme Yöntemleri Kullanılarak Hayvancılıkta Yapılan Çalışmalar

Kmet et al. (2000) boğaların karkas performansını tahmin etmek için video görüntü analizinden yararlandıkları çalışmada, görüntülerin alınmasında Cannon SVC ION RC-260 kamera kullanmışlardır. Çalışmada 63 baş Simental boğaya ait görüntülerden boğaların canlı ağırlıklarını doğrusal regresyon modeli ile yüksek bir doğrulukla ($R^2=0.94$) tahmin etmişlerdir. Ayrıca kullandıkları yöntemin boğaların performans testinde bir seleksiyon kriteri olarak kullanılabileceğini ve bu yöntemin düşük maliyetli olmasının yanı sıra pratik olduğunu bildirmişlerdir.

Lines et al. (2001), yüzmekte olan somon balıklarının ağırlığını tahmin etmek için görüntü işleme yöntemlerini kullanmışlardır. Bu amaçla ağırlıkları 0.7 ile 5.7 kg aralığında değişen 17 somon balığından 60 görüntüyü çeşitli uç çıkarma ve bölütleme gibi görüntü işleme yöntemlerine tabi tutarak görüntülerin arka planını ve görüntülerdeki diğer balıkları elemine etmişlerdir. Araştırmacılar balıkların

bulunduğu kafes ortamının değişkenliğinin ve karmaşık doğasının görüntüleme esnasında zorluk çıkardığını bildirmişlerdir. Çalışma sonunda görüntü işleme yöntemleriyle otomatik olarak hesaplanabilen doğrusal boyutlar kullanıldığında somon balığının ağırlığının %0.5'ten daha düşük bir hata ile tahmin edilebileceğini belirtmişlerdir.

Goyache et al. (2001) makine öğrenmesinde kullanılan algoritmalarından üç tanesini (Cubist, SAFE ve M5) kullanarak bir çalışma yürütmüşlerdir. Çalışmada Asturiana de los Valles et sığırlarını önden, yan ve arkadan olmak üzere üç açıdan görüntülemiş ve bu dijital görüntülerden 63 ana nokta belirleyerek çeşitli vücut özelliklerinin (0.325-0.810) aralığında değişen belirleme katsayıları ile tahmin edilebildiğini bildirmişlerdir. Araştırmacılar çalışma sonunda makine öğrenmesi ve görüntü işleme yöntemlerinin sığırlarda konformasyon değerlendirmesinde kullanışlı bir araç olduğunu belirtmişlerdir.

Özkaya (2006) tez çalışmasında besi sığırlarında çeşitli vücut özelliklerini kullanarak canlı ağırlık ve karkas parametrelerini görüntü işleme yöntemleriyle tahmin etmeye çalışmıştır. Çalışmada öncelikle vücut ölçümleri alınan hayvanlardan daha sonra 155 görüntü elde eden araştırmacı, görüntü kalitesi bakımından uygun olan 50 görüntü ile çalışmayı yürütmüştür. Araştırmacı her biri 1.56cm² olan ve 270 adet kareden oluşan asetat planimetreyi kullanarak karkas alanını hesapladıktan sonra ayrıca bu alanı sayısal kamera ile de görüntülemiş ve elde ettiği verileri çalışmada kullanmıştır. Araştırmacı klasik yöntemlerle yaptığı tahminde sadece göğüs çevresi ile elde edilen R² değerinin 0.81 olduğunu, bütün ölçümler (vücut alanı, sağrı yüksekliği, vücut uzunluğu, cidago yüksekliği, sağrı genişliği ve beden derinliği) kullanıldığında ise bu değerin 0.86 olarak tespit edildiğini bildirmiştir. Besi sığırlarının canlı ağırlıklarının tespitinde klasik yöntemde sadece göğüs çevresinin yeterli bir parametre olabileceğini ifade eden araştırmacı, görüntü işleme yönteminin canlı ağırlığı ve vücut ölçülerini tahmin gücünün düşük olduğunu bildirmiştir.

Bewley et al. (2008) görüntü işleme yöntemini kullanarak süt sığırlarının deri altı yağının dolaylı olarak ölçümünde hem saha araştırmalarında hem de yetiştiriciler tarafından çiftliklerde yaygın olarak kullanılan bir yöntem olan vücut

kondisyon puanının tahmin potansiyelini araştırmışlardır. Çalışmada 0.5 puan aralıklarla belirlenen puanlama sisteminde tahmin başarısının %100, 0.25 puan aralıklı yapıldığı yöntemde ise bu başarının %92.79 olduğunu ve bu değerin daha fazla sığır kullanılarak çalışma yapılması durumunda yükselebileceğini bildirmişlerdir.

Stajenko et al. (2008) et sığırlarının yan tarafından AGEMA 570 termal kamera kullanarak termografik olarak elde ettikleri görüntülerden sığırların vücut ölçülerini ve canlı ağırlıklarını tahmin etmeyi amaçlamışlardır. Çeşitli vücut özelliklerinin elle ölçülmesi ve termal görüntüler yardımıyla yapılan tahminler arasında yüksek düzeyde ve anlamlı korelasyon ($r=0.74$) bulunduğunu, özellikle döl kontrolü boyunca boğaların ağırlıklarının takip edildiği dönemde, bu yöntemle tahmin yapmanın hayvanlarda stresi de azaltabileceğini bildirmişlerdir.

Krukowski (2009), tez çalışmasında ineklerin vücut kondisyon puanlarını onların görüntülerinden otomatik olarak tespit edebilecek bir yöntem önermiştir. Araştırmacı SR-3000 marka kamera ile elde ettiği üç boyutlu görüntülere filtreleme, bölümlendirme, normalleştirme ve figür çıkarma gibi çeşitli görüntü ön işlemleri yapmıştır. Bu işlemlerden sonra vücut kondisyon puanlamasının 0.5 puan aralıklarla yapıldığı durumda, tahminin başarısının %100, 0.25 aralıklarla yapıldığı durumda ise %79 olduğunu ve üç boyutlu görüntülerinden sığırların vücut kondisyon puanlamasının başarılı bir şekilde tahmin edilebileceğini bildirmiştir. İkinci durumda başarının düşük çıkmasının nedenleri arasında tahminde kullanılan yöntemlerin yetersiz olabileceğini ve yapay sinir ağları gibi yöntemler kullanıldığında %79 olan bu başarının daha da yükselebileceğini düşündüklerini bildirmiş, gelecek çalışmalar için yapay sinir ağlarını önermişlerdir.

Mollah et al. (2010) etlik piliçlerde bilgisayar destekli dijital görüntü analizi ile canlı ağırlığı tahmin etme olanaklarını araştırmıştır. Araştırmacılar, 7-42 günlük büyüme sürecindeki rastgele 20 civcive ait 1200 görüntüyü kullanarak piliçlerin vücut yüzey alanlarını hesaplayıp onların canlı ağırlıklarını, çoklu doğrusal regresyon yardımıyla 0.999'luk bir uyumla tahmin edebildiklerini bildirmişlerdir. Yöntemin en büyük avantajını ise uzaktan ağırlık belirleme yöntemi olması nedeniyle strese sebebiyet vermemesi olarak tanımlamışlardır.

Taşdemir (2010), sığırlarda çeşitli vücut özelliklerini ve canlı ağırlığı tahmin etmek amacıyla yaptığı doktora araştırmasında, ağırlıkları bilinen 220 baş siyah alaca ineği, sağımhane çıkışına kurulan ve hayvanın geçişinin bir sensör yardımıyla otomatik olarak tespit edildiği özel olarak tasarlanmış bir fotoğraf stüdyosunda, yandan ve üstten ikişer açıdan görüntülemiştir. Araştırmacı görüntülerin otomatik olarak alınmasında Delphi programlama dilinde geliştirilen görüntü analizi yazılımını, vücut özelliklerinin görüntüden hesaplanmasında MATLAB paket programını kullanmıştır. Buradan elde edilen veriler yardımıyla bulanık mantıktan yararlanarak geliştirdiği bulanık kural tabanlı sistem ve regresyon analizini kullanarak siyah alaca ineklerin canlı ağırlığını tahmin etmiştir. Çalışma sonunda regresyonla tahmin edilen sonuçlar ile canlı ağırlık arasında $R^2=0.96$ ve bulanık kural tabanlı sistem ile canlı ağırlık arasında $R^2=0.98$ değerini elde ettiğini ve görüntü işleme yöntemleriyle canlı ağırlığın güvenilir bir şekilde tahmin edilebileceğini bildirmiştir.

Görüntü işleme yöntemleri ile sığır ve mandalarda çeşitli vücut parametrelerini (sağrı yüksekliği, vücut uzunluğu vs.) tahmin etmek için Önal (2011) 82 baş boz ırkı sığırı ve 98 baş Anadolu mandasına ait görüntüleri kullandığı bir doktora araştırması yapmıştır. Çalışma için bir platform hazırlamış ve bu platform üzerinde çeşitli referans noktaları belirleyerek hayvanlarda görüntüleme yapmıştır. Bu referans noktalarını Image-Pro Plus 4.5 yazılımına tanıtmış ve programda “measure, measurement” sekmesini kullanarak vücut özelliklerine ait ölçümler yapmıştır. Araştırmacı vücut alanı kullanılarak canlı ağırlığın tahmin edilebileceğini (en yüksek $R^2=0.92$), vücut uzunluğu kullanıldığında ise tahmin gücünün düşük olduğunu bildirmiştir. Yine her bir türe ait cidago yüksekliği, sağrı yüksekliği, sırt yüksekliği ve oturak yumru yüksekliğinin tahmininde görüntü işleme yöntemlerinin ölçü bastonu ve ölçü şeridi gibi araçlarla yapılan klasik ölçümlere alternatif olarak kullanılabileceğini, vücut uzunluğu ve gövde uzunluğunu belirlemede ise görüntü işleme yöntemlerinin tahmin gücünün düşük olduğunu bildirmiştir.

Kongsro (2014) domuzlarda görüntüden canlı ağırlığı tahmin etmek için üç boyutlu Microsoft Kinect kamera kullanmıştır. Çalışmada 50 adet domuzun baş, kulak ve kuyruk kısmını çıkardıktan sonra domuzlara ait görüntülerin haritalarını

elde etmiştir. Araştırmacı domuzun gerçek ağırlığı ile tahmin edilen ağırlığı arasındaki R^2 değerinin 0.99 olduğunu ve %4-%5 ortalama mutlak hata yüzdesi (MAPE) ile domuzların canlı ağırlıklarının tahmin edebildiğini bildirmiştir.

Kashiha et al. (2014), her birinde 10 domuz bulunan dört kafese, yerden 2.20 m yükseklikte bir kamera yerleştirerek domuzların canlı ağırlığını tahmin etmeyi amaçlamışlardır. Araştırmacılar elde ettikleri görüntülerdeki piksellerden hesapladıkları domuz alanını kullanarak çeşitli matematiksel modeller yardımıyla domuzlarda canlı ağırlığı %97.5'luk duyarlılıkla tahmin edebildiklerini bildirmişlerdir.

Alikhanov et al. (2015), yumurta ağırlığını dolaylı olarak ölçmek için görüntü işleme yöntemlerinden yararlanmışlardır. Bu amaçla 26 yumurtadan elde edilen görüntüleri öncelikle gri resme sonra siyah beyaz resme dönüştürmüşler ve buradan çevre, alan, majör ve yan eksen, şekil katsayıları ve yumurta hacmi gibi geometrik değişkenleri hesaplayarak regresyon analizi yardımıyla yumurta ağırlığını tahmin etmişlerdir. Bu değişkenlerden en önemli olanların yumurta alanı ve hacmi olduğunu, tahmin edilen model için $R^2=0.9439$ olarak tespit edildiğini bildirmişlerdir.

Görüntü işleme yöntemlerini kullanarak et sığırlarının canlı ağırlığının belirlenmesi isimli çalışmalarında Pradana et al. (2016), et sığırlarının yan tarafından ve önünden aldıkları görüntülere çeşitli görüntü segmentasyon ve uç çıkarma yöntemlerini uygulamışlardır. Araştırmacılar görüntüledikleri hayvanların kafa ve ayak kısımları dışında kalan bölgeyi siyah beyaz resme dönüştürmüşler ve doğrusal regresyon eşitlikleri yardımıyla canlı ağırlığı tahmin etmeye çalışmışlardır. Çalışma sonunda %73.21'lik duyarlılıkla sığırların canlı ağırlığını tahmin edebildiklerini bildirmişlerdir.

Bektaş (2016), yüksek lisans tez çalışmasında sığırlarda sağrı, sırt ve cidago yüksekliği ile vücut uzunluğu gibi çeşitli morfolojik özellikleri görüntü işleme yöntemleri yardımıyla tahmin etmeye çalışmıştır. Çalışmada elle yapılan ölçüm ve görüntü işleme yöntemleri ile yapılan tahminler arasındaki farkın istatistiksel olarak önemliliğini test etmek amacıyla bağımlı iki grup için t testini kullanmıştır. Çalışma

sonunda arařtırmacı oturak yumru geniřlięi ($p=0.085$), cidago ykseklięi ($p=0.887$), sırt ykseklięi ($p=0.214$), vcut uzunluęu ($p=0.307$) ve gvde uzunluęunun ($p=0.084$) elle yapılan lmleri ve grnt iřleme yntemleri yardımıyla tahmin edilen deęerleri arasında istatistiksel olarak bir farklılık olmadığını belirtmiřtir. Sz konusu vcut zelliklerinin grnt iřleme yntemleriyle bařarılı bir řekilde tahmin edileceęini belirten arařtırmacı saęrı ykseklięi ($p=0.010$), oturak yumru ykseklięi ($p=0.049$) ve gęs derinlięinin ($p=0.007$) tahmin edilmesinde yntemin bařarısının dřk olduęunu bildirmiřtir.

Shi et al. (2016) binokler grntleme sistemleri ile LabVIEW platformunu birlikte kullanarak domuzların otomatik olarak grntlenmesi ve canlı aęırlıklarının tahmin edilmesi amacıyla bir alıřma yrtmřlerdir. alıřmada 80 gnde 10 domuzdan elde edilen 1460 derinlik grntsn ve basit doęrusal regresyon analizini kullanarak domuzların vcut uzunluęu ($R^2=0.94$), cidago ykseklięi ($R^2=0.99$) ve canlı aęırlıęını ($R^2=0.9931$) yksek belirleme katsayıları ile tahmin etmiřlerdir. alıřma sonunda arařtırıcılar alıřmada kullandıkları yntemle domuzlarda vcut uzunluęunu 1.88cm, cidago ykseklięini 0.81cm ve canlı aęırlıęı 1.759kg hata ile tahmin edebildiklerini bildirmiřlerdir.

Amraei et al. (2018) etlik pililerin canlı aęırlıklarını grnt iřleme yntemleriyle tahmin etmek amacıyla yrttkleri arařtırmada yerden 2m ykseklikte konumlandırılmıř SAMSUNG SM-N9005 dijital kamerayı kullanarak 30 etlik piliten 2440 grnt elde etmiřlerdir. Alınan grntlerde pilicin bulunduęu alanın arka plandan ayıklanması iin grnt iřleme yntemlerinden grnt bltleme ve adaptif eřikleme kullanıldıęı arařtırmada pilicin bulunduęu alan beyaz, kalan kısım siyah olacak řekilde grnt ikili yapıya dnřtrlmřtir. Pilicin bař ve kuyruk kısmının kaldırılmasında Chan-Vase yntemini kullanan arařtırıcılar daha sonra bu grntden alan, evre ve eksen uzunlukları gibi eřitli parametreleri hesaplamıřlardır. Elde edilen lm deęerleri, dinamik veri tabanlı bir modelleme yaklařımı olan Transfer Fonksiyon modeli kullanarak etlik pililerde canlı aęırlıęı tahmin etmiřlerdir. alıřma sonunda arařtırıcılar etlik pililerin canlı aęırlıkları ile yapılan tahmin arasındaki belirleme katsayısının $R^2=0.98$ olduęunu bildirmiřlerdir.

Huang et al. (2018) Çin'in yerli sığır ırklarından olan Qinchuan ırkı sığırlarda çeşitli vücut özelliklerini (cudago yüksekliği, sağrı yüksekliği, sırt yüksekliği, göğüs derinliği ve vücut uzunluğu) tahmin etmek amacıyla bir çalışma yürütmüşlerdir. Araştırmacılar üç boyutlu bir görüntüleme aracı olan LIDAR sensör aygıtını kullanarak, ilki 157cm, ikincisi 179cm ve sonuncusu 155cm uzaklıktan üç adet sığırı yan kısmından görüntülemişlerdir. Bundan sonra çeşitli görüntü işleme yöntemlerini (görüntü bölütleme vs.) kullanarak sığırların olduğu kısmı arka plandan ayıklamışlardır. Bu aşamadan sonra çeşitli filtreleme yöntemleri, K-means kümeleme analizi gibi bir dizi işlemde geçirdikleri görüntüden vücut ölçümlerini tespit etmişlerdir. Çalışma sonunda araştırmacılar belirlenen vücut özellikleri için elle yapılan ölçüm ile görüntü işleme yöntemleri yardımıyla hesaplanan değerleri kullanarak en düşük hata ile (%0.2) sırt yüksekliğini ve en yüksek hata (%2) ile sağrı yüksekliğini yüksek bir doğrulukla tahmin ettiklerini belirtmişlerdir. Ayrıca elle yapılan ölçümün 30-70 dakika arasında sürdüğünü oysa belirtilen görüntü işleme yöntemleri yardımıyla bu ölçümün yaklaşık beş dakikalık bir sürede tamamlanabildiğini belirtmişlerdir.

Cozler et al. (2019) Fransa'nın Le Rheu eyaletinde bulunan Mejusseume deneysel süt istasyonunda ineklerin üç boyutlu vücut şekillerini kaydedip analiz edebilen bir aracı test etmek amacıyla çalışma yürütmüşlerdir. Çalışmada göğüs çevresi, göğüs derinliği ve sağrı yüksekliği gibi vücut özelliklerini tahmin etmek amacıyla 30 baş Siyah Alaca inek, deney istasyonuna yerleştirilen her biri bir lazer projektör ile desteklenmiş beş kamera bulunan hareketli bir düzenekle taranmıştır. Araştırmacılar kameralardan dördü portalın her iki tarafında yerden 0.40m ve 1.77m yükseklikte biri ise tepe noktasında yerden 3m yükseklikte olacak şekilde düzenek üzerinde kameraları konumlandırmışlardır. Çalışma sonunda elle ölçülen çeşitli vücut özelliklerine ait ölçümler ile otomatik olarak tahmin edilen ölçümler arasında en düşüğü $R^2=0.38$ ve en yükseği $R^2=0.79$ olan belirleme katsayıları hesaplamışlardır. Araştırmacılar elle ölçümün 2.5 dakikada, otomatik ölçümün 15 dakikada yapıldığını ancak burada görüntünün alınmasının hızlı olmasına rağmen görüntü işlemenin uzun zaman almasından dolayı sürenin uzadığını, görüntü işleme süreleri kısaltılabilirse daha hızlı bir tahmin yapılabileceğini belirtmişlerdir.

Shi et al. (2019) domuzların vücut uzunluğu, vücut genişliği, vücut yüksekliği, kalça genişliği ve kalça yüksekliği gibi vücut özelliklerini tahmin etmek amacıyla Çin'in Tianjin şehrinde bulunan Hui Kang domuz çiftliğinde mobil bir görüntüleme sistemi kullanarak bir araştırma yürütmüşlerdir. Bu amaçla 70, 100 ve 130 günlük dönemde 200 domuzu kullandıkları çalışmada, su içme alanında radyo frekanslı kimlik tanılama (RFID) cihazı ile domuzları tespit ederek otomatik olarak görüntüleme yapmışlardır. Daha sonra uç ve figür çıkarma gibi çeşitli görüntü işleme yöntemlerini kullanarak domuz görüntüsünün arka plandan ayıklanmasını sağlamışlardır. Araştırmacılar çalışma sonunda elle yapılan ölçümlerle mobil görüntüleme sistemi kullanılarak yapılan tahminden elde ettikleri vücut ölçümleri arasında 70 günlük dönemde (0.82-0.93) aralığında, 100 günlük dönemde (0.92-0.94) aralığında ve 130 günlük dönemde tüm vücut özellikleri için 0.93'lük belirleme katsayıları hesaplandığını bildirmişlerdir. Araştırmacılar kullandıkları mobil görüntüleme sisteminin donanım ve yazılımı birlikte kullanabilen zeki ve otomatik bir ölçüm sistemi olduğunu ve geleneksel ölçüm yöntemlerinin yerine başarılı bir şekilde kullanılabileceğini bildirmişlerdir.

Yan et al. (2019), Yak (Tibet) sığırlarının iki boyutlu görüntülerinden Photoshop (version CS5, PS) programı aracılığıyla elde ettikleri çeşitli vücut ölçüm parametrelerini kullanarak canlı ağırlıklarını tahmin etmeye çalışmışlardır. Bu amaçla yerden 75cm yükseklikte konumlandıkları Canon 400D kamera ile sığırlara yaklaşık 5m mesafeden görüntüleme yapmışlardır. Araştırmacılar 55 Yak sığırına ait ilkbahar, yaz ve kış mevsiminde görüntüler almışlardır. Çalışmada en yüksek belirleme katsayısının yaz mevsiminde elde edildiğini, bu mevsime ait verilerle ve çoklu doğrusal regresyonda vücut diyagonal uzunluğu, cidago yüksekliği ve vücut alanının bulunduğu modelde R^2 değerini 0.972 olarak tespit ettiklerini bildirmişlerdir. En düşük MAPE değerini bahar mevsiminde %3.19 olarak ve en düşük RMSE değerini yaz mevsiminde 7.52kg olarak tespit etmişlerdir.

2.4 Görüntü İşleme Yöntemleri ve Yapay Sinir Ağları Kullanılarak Yapılan Çalışmalar

Babalık ve Botsalı (2010) araştırmalarında Çeşit-1252 türü durum buğdayını, camsı ve camsı olmayan daneler bakımından sınıflandırmaya çalışmışlardır. Bu amaçla bir masaüstü tarayıcı yardımı ile elde ettikleri buğday daneleri görüntülerinden, görüntü işleme yöntemleriyle elde ettikleri histogram bilgilerini kullanarak, yapay sinir ağı modellerinin sınıflandırma başarılarını incelemişlerdir. İlk yöntem olarak eğitimsiz öğrenme modeli olan “Öz Düzenlemeli Harita” modelinin, daha sonra ise eğitilmiş öğrenme modeli olan “Çok Katmanlı Algılayıcı (ÇKA)” modelinin sınıflandırma başarısını test etmişlerdir. Çalışma sonunda öz düzenlemeli harita modelinin ortalama test başarısını %97.5 olarak tespit etmişlerdir. ÇKA ile yapılan sınıflandırmada ise bu başarının, gizli katmanda 10 nöronun kullanıldığı yapıda %100 olduğunu belirten araştırmacılar durum buğdayının sınıflandırılmasında görüntü işleme ve yapay sinir ağlarının başarılı bir şekilde kullanılabileceğini bildirmişlerdir.

Taşova (2011) yüz tanıma için görüntü işleme ve yapay sinir ağlarını kullanmıştır. Bu amaçla kişilerin yüz fotoğraflarından belirli sayıda öznitelik vektörleri üretmiş, bunları kullanarak eğittiği yapay sinir ağlarının yüz tanımda başarılı olduğunu, hatalı işlemlerin ise görüntü alınırken yerle yüz arasındaki açının tam 90 derece olmaması, çekim hatası, poz değişkenliği ve dış etmenler gibi sebeplerden meydana geldiğini bildirmiştir.

Görüntü işlemenin tarımsal alanda uygulanabilirliğini test etmek amacıyla Sabancı vd. (2012) araştırmalarında, patatesleri boyut olarak sınıflandırmaya çalışmışlardır. Bu amaçla öncelikle değişik ebatlardaki patatesler 1.3 mega-piksel CCD sensörlü bir web-cam kullanılarak fotoğraflamışlardır. Daha sonra bunlar gri seviye resimlere dönüştürülmüş ve program yardımıyla değişik ebattaki patateslere ait histogram bilgilerini elde etmişlerdir. Histogram, görüntü üzerindeki piksellerin değerlerinin grafiksel ifadesidir. Elde edilen histogram yardımıyla ve görüntü işleme yöntemlerinden “otsu metodu” kullanarak eşikleme yapan araştırmacılar böylece görüntü üzerindeki nesnelerin arka planı ile nesne hatlarının çıkartılmasını sağlamışlardır. Son olarak elde ettikleri ikili resim bilgilerinin boyutlarını eşitlemiş

ve resimleri sütun matrise dönüştürmüşlerdir. Patateslerin ikili resim bilgileri, logaritmik sigmoid aktivasyon fonksiyonu kullanılan çok katmanlı ileri beslemeli geri yayılım sinir ağı ile eğitilmiş ve çalışma sonunda patateslerin başarılı bir şekilde büyük, orta ve küçük boy patatesler şeklinde üç grupta sınıflandırıldığını bildirilmişlerdir.

Betonun basınç dayanımını belirlemek amacıyla, Çankaya vd. (2013) görüntü üzerinde analitik bir model oluşturmuştur. Araştırmacılar modellemenin başarısını test etmek için fotoğraflama işleminden sonra aldıkları numuneleri pres makinasında kırmış ve dayanım değerlerini not etmişlerdir. Çalışma sonunda görüntü işleme ve yapay sinir ağlarını beraber kullanarak önerdikleri metodu, numunelerin pres makinasında kırıldığı tahribatlı deney metodu ile uyumunun yaklaşık %98.88 olduğunu bildirmişlerdir.

Sabancı (2013) doktora çalışmasında, üzerinde ilaçlama ünitesi, kontrol ünitesi, bilgisayar ve kameraların olduğu bir ilaçlama robotu geliştirmiştir. Araştırmacı robot yardımıyla sıra arasındaki yabancı otları görüntü işleme tekniklerini kullanarak, renk bilgisine göre, sıra üzerindeki otları ise görüntü işleme teknikleri ve yapay sinir ağlarını kullanarak, şekil ve renk bilgisine göre tespit etmiş ve ilaçlama sıvısı (mürekkepli su) uygulamıştır. Araştırmacı çalışma sonunda geliştirilen robotun geleneksel ilaç uygulamasına göre yaklaşık %54 daha az ilaç kullandığını, bununla insan, hayvan ve çevre sağlığının korunmasının yanı sıra ilaçlama maliyetlerini de düşürdüğünü bildirmiştir.

2.5 Görüntü İşleme Yöntemleri ve Yapay Sinir Ağları Kullanılarak

Hayvancılıkta Yapılan Çalışmalar

Yeni doğan kuzuları post desenine göre sınıflandırmayı amaçlayan Khojastehkey et al. (2015) 300 kuzuyu dijital kamera ile 40cm mesafeden 4320x3240 piksel boyutunda fotoğraflamışlardır. Daha sonra çekilen fotoğraflar iki farklı senaryo için görüntü işleme teknikleriyle incelenmiştir. İlk senaryoda fotoğraflardan alan, çevre, çap, yatay eksen uzunluğu, dikey eksen uzunluğu gibi çok sayıda morfolojik ve yapısal figürler çıkarmışlardır. İkinci senaryoda ise yeni geliştirilen bir yöntemi kullanmışlardır. Araştırmacılar bu yöntemde orijinal resmi 4

eşit alt resme ayırarak daha önce klasik senaryoda çıkardıkları figürlere, orijinal resmin parçaları arasındaki varyans ve korelasyonları da ekleyerek, ilk senaryoda 21 olan figür sayısını 44' e çıkarmışlardır. Çalışma sonunda araştırmacılar ilk senaryo ile yapılan sınıflandırmada yapay sinir ağlarının sınıflama başarısının %92, ikinci senaryoda ise bu başarının %100 olduğunu bildirmişlerdir.

Wongsriworaphon et al. (2015) araştırmalarında çiftlik şartlarında domuzların canlı ağırlıklarını, onları rahatsız etmeksizin tahmin etmeye yarayan görüntü işlemeye dayalı bir yaklaşım önermişlerdir. Çalışmada bilgisayar destekli görüntü yakalama ve Vektör Quantize Zamansal Çağrışimli Bellek (Vector-Quantized Temporal Associative Memory-VQTAM) olarak bilinen denetimli öğrenme algoritmasına sahip bir yapay sinir ağı modeli kullanmışlardır. Araştırmacılar çiftlik tabanının 2.80m üstüne bekleme alanına dik olacak şekilde monte ettikleri 640x480 piksel çözünürlükteki bir dijital kamera ile 90x160cm'lik bir alanda bekleyen domuzların üzerinden görüntü almışlardır. Elde edilen görüntülerden ilk olarak domuzun ağırlık merkezinden sınır noktalarına olan uzaklıkları, sonra domuzun çevre uzunluğu için figür çıkarımları yapılmış ve buradan hesaplanan değerler 25, 49, 100 ve 169 nöronun kullanıldığı yapay sinir ağlarıyla analiz edilmiştir. Çalışma sonunda araştırmacılar 169 nöronun kullanıldığı durumda ortalama olarak %2.94'lük bir hata ile domuz canlı ağırlıklarının tahmin edilebildiğini bildirmişlerdir.

Amraei et al. (2017) etlik piliçlerin canlı ağırlıklarının Makine Görmesi (Machine Vision) ve yapay sinir ağları kullanılarak tahmin etmek için yürüttükleri araştırmada 1 günlük yaşta 30 etlik pilici 42 günlük büyüme dönemi boyunca sabah ve akşam olmak üzere günde iki kez görüntülemişlerdir. Araştırmacılar çeşitli görüntü işleme yöntemlerini kullanarak pilicin görüntüde bulunan kısmı dışındaki gürültü ve arka planı kaldırmış, görüntüden altı farklı vücut parametresi hesaplamışlardır. Çalışmada yapay sinir ağlarında kullanılan Bayesian Regularization, Levenberg-Marquardt, Scaled Conjugate Gradient ve Gradient Descent algoritmaları tahmin başarısı açısından karşılaştırılmıştır. Veri setinin test ve eğitim seti olarak ayrıldığı çalışmada araştırmacılar etlik piliçlerin canlı ağırlıkları ile tahmin edilen ağırlıkları arasındaki R^2 değerlerini sırasıyla 0.984, 0.98, 0.967 ve 0.946 olarak belirlemişlerdir. Araştırmacılar makine görmesi ve yapay sinir ağlarını

kullanarak yürüttükleri araştırmada tespit edilen en yüksek hatanın 50 gramdan daha az olduğunu belirtmişlerdir.

Bhatt et al. (2018) koyunların görüntülerinden canlı ağırlıklarını tahmin etmek amacıyla bir çalışma yürütmüşlerdir. Çalışmada 52 koyun görüntüsünü öncelikle program yardımıyla manuel olarak kırpan araştırmacılar hayvan görüntüsü dışındaki kısımları elemine edecek şekilde hayvanın olduğu kısmı turuncu renkle işaretlemişlerdir. Daha sonra araştırmacılar işaretlenen alanı piksel alan hesaplama tekniği ile hesaplamışlar ve elde ettikleri verilerin %80'ini eğitim %20'si test seti olacak şekilde çapraz doğrulama ile setlere ayırmışlardır. Yapay sinir ağlarında iki gizli katman, 5 ve 10 nöron kullanılan çalışma sonunda araştırmacılar, tahmin edilen canlı ağırlık değeri ile gerçek ağırlık değeri arasındaki R^2 değerini en yüksek 0.81 olarak belirlemişlerdir.

2.6 Yapay Sinir Ağları ve Diğer İstatistiksel Yöntemlerin Karşılaştırıldığı Çalışmalar

Grzesiak et al. (2003) 902 baş siyah alaca ineğe ait kısmi laktasyon kayıtlarını kullanarak 305 günlük laktasyon süt verimini tahminlemek amacıyla YSA ve ÇDR yöntemlerini karşılaştırmalı olarak incelemişlerdir. Veri setininin %50 eğitim, %25 test ve %25 doğrulama setine ayrıldığı çalışmada, YSA ile tahminde RMSE değerinin 485.67kg, ÇDR ile tahminde RMSE değerini 517.31kg olarak belirlemişlerdir. Sonuç olarak araştırmacılar YSA'nın diğer biyometrik yöntemlere alternatif olarak kullanılabileceğini bildirmişlerdir.

Peraı et al. (2010) kanatlı beslenmesinde kullanılan et ve kemik ununun gerçek metabolizma edilebilir enerjisini tahmin etmek için bir araştırma yürütmüşlerdir. Çalışmada YSA, parçalı regresyon ve ÇDR analizlerini karşılaştırmalı olarak incelemişlerdir. Çalışma sonunda gerçek metabolizma edilebilir enerjinin YSA ($R^2=0.94$) kullanıldığında, ÇDR ($R^2=0.38$) ve parçalı regresyondan ($R^2=0.36$) daha yüksek doğrulukla tahmin edilebileceğini bildirmişlerdir.

Takma vd. (2012) siyah alaca ineklerin süt verimi üzerine laktasyon süresi, servis periyodu ve buzağılama yılının etkisini araştırdıkları çalışmada ÇDR ve YSA yöntemlerinin tahmin başarısını karşılaştırmışlardır. Çalışmada beş laktasyon dönemindeki toplam 305 baş sığra ait veriler kullanılmış, YSA ile elde edilen tahminin başarısının (en yüksek $R^2=0.85$ ve en düşük $MAPE=\%6.1$), ÇDR'den (en yüksek $R^2=0.75$ ve en düşük $MAPE=\%24.7$) daha yüksek olduğunu bildirmişlerdir. Araştırmacılar ileri beslemeli geri yayımlı YSA'nın süt verimini tahmin etmede tercih edilebileceğini bildirmişlerdir.

Dongre et al. (2012) Sahiwal sığırlarında ilk laktasyon süt veriminin tahmininde ÇDR ve YSA'nın tahmin etkinliğini karşılaştırdıkları çalışmada 49 yıl boyunca 51 boğadan yavrulatan 643 Sahiwal ineğine ait 12854 test günü verisini kullanmışlardır. Araştırmacılar veri setinin alt setlere ayrılmasında eğitim ve test seti için $\%66.67$ - $\%33.33$, $\%75$ - $\%25$, $\%80$ - $\%20$ ve $\%90$ - $\%10$ olacak şekilde dört farklı veri ayırma durumunu incelemişlerdir. Çalışmada ayrıca LM, BR ve SCG algoritmalarının etkinliği de incelemişlerdir. Veri setinin $\%90$ eğitim ve $\%10$ test setine ayrıldığı durumda en iyi tahmin sonucunun elde edildiğini belirten araştırmacılar, iki gizli katmanda üç ve beş nöron olması durumunda bütün algoritmalarda en iyi tahmin sonucuna ulaşmışlardır. Çalışma sonunda SCG ($R^2=0.8608$) algoritmasının, BR ($R^2=0.8587$) ve LM ($R^2=0.8575$) algoritmalarından daha yüksek belirleme katsayısı ile tahmin yaptığını bildirmişlerdir. Ayrıca tüm YSA algoritmaları ile elde edilen sonuçların ÇDR ($R^2=0.8516$) analizinden daha yüksek belirleme katsayısı ile tahmin ürettiğini bu nedenle YSA'nın 305 günlük ilk laktasyon süt verimini tahmin etmede kullanışlı bir alternatif araç olduğunu belirtmişlerdir.

Atıl ve Akıllı (2015), 1000 baş siyah alaca sığra ait süt, yağ ve protein verimi ile ilkine buzağılama yaşı, sağım günü ve mevsimi, buzağılama aralığı kayıtlarını kullanarak YSA ve Kümeleme analizinin sınıflandırma başarılarını karşılaştırmışlardır. Çalışmada YSA'nın ($R^2=0.999$, $MAPE=\%1.77$ ve $RMSE=1.7599$) Kümeleme analizinden ($R^2=0.949$, $MAPE=\%10.48$ ve $RMSE=1.7758$) daha yüksek doğrulukla sınıflandırma yapabildiğini bildirmişlerdir.

Atıl ve Akıllı (2016), süt sığırlarının sınıflandırılmasında YSA ve K-means Kümeleme analizini karşılaştırmalı olarak incelemişlerdir. Araştırmacılar çalışmada hayvancılık alanında verim özellikleri ve sınıflandırma, örüntü tanıma, optimizasyon ve ileri dönük tahminlerin yapıldığı çalışmalara da değinmiş ve çalışma sonunda süt sığırlarının sınıflandırılmasında Yapay sinir ağlarının Kümeleme analizinden daha iyi sonuç verdiğini bildirmişlerdir.

Zhag and Goh (2016), inşaat mühendisliğinde sıklıkla kullanılan kazık temeller için üretilen kazıkların çakılabilirliğini tahmin etmek için 4072 kazık kullanarak yürüttükleri araştırmada YSA ve MARS analizlerinin tahmin performanslarını karşılaştırmışlardır. Araştırmacılar, maksimum basınç gerilmeleri için geri yayımlı YSA kullanıldığında $R^2=0.970$ ve MARS kullanıldığında $R^2=0.957$ olarak hesaplamışlardır. Çalışma sonunda iki yöntemin yakın sonuçlar ürettiğini, MARS yönteminin esnek modeller oluşturabildiği için YSA'dan daha verimli olarak düşünülebileceğini bildirmişlerdir.

2.7 Yapay Sinir Ağlarında Kullanılan Öğrenme Algoritmalarının Karşılaştırıldığı Çalışmalar

Moosavizadeh et al. (2011) toprak su tutma eğrisini tahminlemek için yapay sinir ağlarını kullanmışlardır. Çalışma verisini 59 toprak tekstürü örneğinde 11 farklı basınçta elde edilmiş toplam 659 hacimsel su içeriği verisi oluşturmaktadır. Veri setinin %70 eğitim, %30 test seti olarak ayrıldığı çalışmada beş girdi değişkeni ile iki gizli katman ve bir çıktı katmanı olması durumunda LM ve BR algoritmalarının etkinliği incelenmiştir. Çalışma sonunda araştırmacılar toprak su tutma eğrisinin YSA kullanılarak başarılı bir şekilde tahmin edilebileceğini, LM algoritması kullanıldığında elde edilen belirleme katsayısının ($R^2=0.9645$), BR algoritmasından ($R^2=0.9361$) daha yüksek bulunduğunu bildirmişlerdir.

Bui et al. (2012) heyelan duyarlılık haritalarında YSA'nın uygulanma potansiyelini araştırmışlardır. Çalışmada eğim, görünüş, arazi kullanımı, toprak tipi, yağış, yollara ve nehirlerle olan mesafe gibi 10 farklı parametre girdi olarak, heyelanın var olup olmaması çıktı katmanı olarak kullanılmıştır. Araştırmacılar YSA ile elde edilen sonuçları daha sonra bir coğrafi bilgi sistemine aktarmış ve heyelan

duyarlılık haritası oluşturmuşlardır. Algoritmaların başarılarını değerlendirmede uzaktan algılamada kullanılan başarı oranını dikkate aldıklarını belirten araştırmacılar, bu oranın LM algoritmasında (%86.1) iken BR algoritmasında (%90.3) olduğunu dolayısıyla BR algoritmasının LM algoritmasından daha iyi sonuçlar ürettiğini bildirmişlerdir. Ayrıca araştırmacılar literatürde LM algoritmasının en hızlı algoritmalarından biri olarak bilindiğini ancak bu algoritma ile eğitimde aşırı uyum problemi gözlemlediklerini belirtmişlerdir.

Akkol vd. (2017) yapay sinir ağları ve çoklu doğrusal regresyon yöntemlerini karşılaştırmak amacıyla 475 baş kıl keçisine ait morfolojik özellikleri kullanmışlardır. Araştırmacılar ağırlık tahmini bakımından yapay sinir ağlarında MAPE değerinin (%4.7957) çoklu doğrusal regresyon analizine göre (%4.8706) daha düşük olduğunu ve bu nedenle son yıllarda YSA'nın ÇDR'ye alternatif olarak sıklıkla kullanıldığını bildirmişlerdir. Yapay sinir ağlarında üç farklı geri yayılım algoritmasının başarısını da test eden araştırmacılar, ortalama mutlak hata yüzdesi (MAPE) bakımından Bayesian Regularization algoritmasının tahmin başarısının (%4.7957) araştırmada test edilen Scale Conjugate (%4.8248) ve Levenberg Marquardt (%4.8975) algoritmalarından daha düşük olduğunu bildirmişlerdir.

Eren vd. (2016) su kaynaklarına bulaştığında kanserojen etki yaratabilen Cr (VI) (Krom-6) ağır metalinin giderilmesinde kullanılan polimer içerikli membranların (PIMs) etkinliğini araştırmışlardır. Bu amaçla geliştirilecek YSA'da üç farklı algoritmanın (BR, LM ve SCG) tahmin hatasını azaltmadaki etkisini karşılaştırmalı olarak incelemişlerdir. Çalışmada 460 veri noktasından sağlanan veriler kullanan araştırmacılar zaman, özüt miktarı, özüt tipi, film kalınlığı, plastikleştiricinin tipi ve miktarı gibi faktörleri girdi değişkenleri olarak, elimine etme miktarını ise çıktı değişkeni olarak ele almışlardır. Araştırmacılar veri setini alt setlere ayırırken BR algoritması için verinin %30'u test %70'ini eğitim setine, LM ve SCG algoritmaları için %70 eğitim, %15 test ve %15 doğrulama setine ayırmıştır. Performans değerlendirme ölçütü olarak belirleme katsayısının (R^2) kullanıldığı çalışmada araştırmacılar, LM algoritmasının ($R^2=0.97$), BR ($R^2=0.95$) ve SCG ($R^2=0.72$) algoritmalarından daha iyi sonuç verdiğini bildirmişlerdir.

Kayri (2016), LM ve BR algoritmalarının tahmin başarısını incelediği çalışmada 2247 öğrenciden alınan 25 değişkenden oluşan veri setini %70 eğitim, %30 test setine ayırmıştır. Araştırmacı aşırı uyum probleminin üstesinden gelmek amacıyla geliştirilen BR ve LM algoritmalarının diğer düzenleme tekniklerine göre hatayı azaltmada daha etkin olduğunu bildirmiştir. Araştırmacı bu iki algoritmayı kullanarak yaptığı denemelerde BR algoritmasını ($R^2=0.3044$) kullandığında, LM algoritmasından ($R^2=0.2789$) daha yüksek belirleme katsayısı elde ettiğini bildirmiştir.

Rahimi et al. (2018) yağış tahmini amacıyla zaman serileri için Doğrusal Olmayan Dışsal Girdili Otoregresif Ağ (The Nonlinear Autoregressive Network With Exogenous Inputs-NARX) modelinde kullanılan LM ve BR algoritmalarının etkinliğini doğrusal regresyon analiziyle karşılaştırmalı olarak incelemişlerdir. Çalışmada Malezya’da bulunan Küresel Navigasyon Uydu Sistemleri (Global Navigation Satellite Systems-GNSS) istasyonundan alınan sıcaklık, basınç ve nem verilerini kullanmışlardır. Araştırmacılar çalışma verisini rastgele %70 eğitim, %15 test ve %15 doğrulama setine ayırmışlardır. Araştırmacılar yapay sinir ağlarında LM ile eğitimde, 25 döngüde $RMSE=0.0266$, $R^2=0.5972$, BR ile eğitimde 347 döngüde $RMSE=0.0308$, $R^2=0.5699$ olarak tespit etmiş, iki algoritma karşılaştırıldığında LM algoritmasının daha iyi sonuçlar ürettiğini bildirmişlerdir. Doğrusal regresyon ve yapay sinir ağları karşılaştırıldığında ise doğrusal regresyonla yapılan tahminin ($RMSE=7.611$ ve $R^2=0.01326$) performans değerlendirme ölçütleri bakımından YSA’dan çok geride kaldığını, YSA’nın yağış tahmininde klasik yöntemle göre daha başarılı olduğunu bildirmişlerdir.

Mukherjee and Rajanikanth (2019) hava kirliliği için önemli bir bileşen olan nitrik oksidin üzerinde bulunan plazma etkisini simüle etmek amacıyla bir çalışma yürütmüşlerdir. Bu amaçla 5 KW gücünde dizel bir jeneratörü laboratuvar ortamında çalıştırarak gaz emisyon kaynağı olarak kullanmış, voltaj, frekans, akış hızı ve motor yükü verilerini YSA’da girdi değişkenleri olarak tanımlamışlardır. Araştırmacılar veri ayırmada altı katmanlı çapraz doğrulama yapmışlar, verinin %83’ünü eğitim ve %17’sini test setine ayırmışlardır. Çalışma sonunda BR algoritmasının hem eğitim ($R^2=0.9980$) hem de test setinde ($R^2=0.9506$) LM

algoritmasından (eđitim setinde $R^2=0.8742$, test setinde $R^2=0.7903$) daha iyi sonular rettiđini bildirmiřlerdir.



3. GEREÇ VE YÖNTEM

Bu bölümde öncelikle hayvan materyalinin nasıl temin edildiği, daha sonra makine öğrenmesinde kullanılmak üzere üç boyutlu kamera ile elde edilen hayvan görüntülerinden görüntü işleme yöntemleri yardımıyla verilerin nasıl elde edildiği hakkında bilgi verilmiş, ayrıca MATLAB komutları kullanılarak görüntülerin sayısallaştırılması bir örnekle anlatılmaya çalışılmıştır. MATrix ve LABoratory kelimelerinin kısaltması olan MATLAB, teknik bilimler ve mühendislik alanındaki hesaplamaların yapılması için geliştirilmiş bir yazılım ve programlama dilidir. Bu çalışmada GİY ve YSA analizleri MATLAB R2013a paket programı, ÇDR ve CHAID analizleri SPSS v25 paket programı ve MARS analizi Salford Predictive Modeler 8 programı deneme sürümü ile yapılmıştır.

Çalışmanın yöntem kısmında öncelikle makine öğrenmesinin genel bir tanıtımı yapılmış ve makine öğrenmesinde kullanılan yaklaşımlardan YSA detaylı bir şekilde anlatılmış, daha sonra MARS, ÇDR ve CHAID analizleri hakkında bilgi verilmiştir. Yapay sinir ağlarında, ağ topolojisinin oluşturulmasında kullanılan parametrelerden bu çalışmada etkisi incelenen Momentum Güncelleme Katsayısı (Momentum Update- MU), Öğrenme Oranı (Learning Rate- LR, λ), En Çok Başarısız Döngü Sayısı (max fail) ve Döngü Sayısının (Iteration Number) tanıtımı yapılmıştır. Bundan sonra verilerin setlere ayrılmasında kullanılan Rastgele Ayırma ve Çapraz Doğrulama (Cross Validation-CV) yöntemleri ele alınmıştır. Ayrıca ağın performansını etkileyen bir başka faktör olan öğrenme algoritmalarından bu çalışmada etkisi araştırılan Levenberg ve Marquardt (Levenberg-Marquardt-LM), Bayes düzenlemesi (Bayesian Regularization-BR) ve Adaptif Öğrenme Oranlı Gradyan (Gradient Descent with Momentum and Adaptive Learning Rate- GDX) algoritmaları hakkında bilgi verilmiştir.

3.1 Gereç

Çalışmada sığırların üç boyutlu görüntülerinden yola çıkılarak canlı ağırlıklarının tahmin edilmesi amaçlanmış ve hayvan materyali arayışına girilmiştir. Öncelikle sağımhane, duraklar ve yürüyüş yolları gibi görüntülemenin daha hassas yapılabileceği alanlara sahip süt sığır işletmeleri araştırılmıştır. Ancak

İzmir ilinde görüŖülen st sĖırı iŖletmelerinde sĖırların canlı aĖırlıklarının kayıt altına alınmadıĖı bilgisine ulaŖılmıŖtır. Besi iŖletmelerinde ise durumun farklı olmadıĖı ve sĖırların aĖırlıklarının sadece kesime giderken kamyon zerinde topluca tartıldıĖı bilgisi alınmıŖtır.

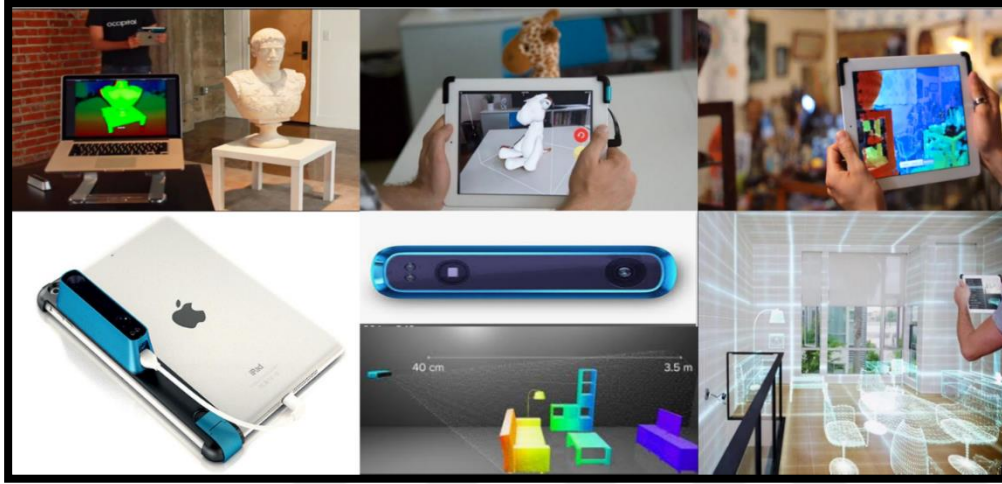
SĖırlarının aĖırlıklarını kayıt altına alan iŖletme bulmak amacıyla modern hayvancılık iŖletmelerini destekleyen Erzurum Tarım ve Kırsal Kalkınmayı Destekleme Kurumu KoordinatrlĖ ile grŖlmŖtr. Kurum personelinden alınan bilgide destekledikleri altı çiftliĖe tartım iin kantar verildiĖi ancak bu iŖletmelerden drdnn faaliyete gemediĖi, bir iŖletmede tartımın altı ayda bir yapıldıĖı ve bu iŖletmede ise son tartımın bir ay nce gerekleŖtiĖi, kalan son iŖletmede henz hi tartım yapılmadıĖı bilgisine ulaŖılmıŖtır.



Ŗekil 3.1. Erzurum İli Damızlık Koyun Kei YetiŖtiricileri BirliĖi bykbaŖ hayvan tartım kantarı

İstenilen zellikte iŖletme bulunamadıĖından canlı hayvan tartımının yapıldıĖı Erzurum İli Damızlık Koyun Kei YetiŖtiricileri BirliĖi hayvan pazarında bulunan Ŗekil 3.1’de grlen kantara, kurbanlık satıŖı iin getirilen hayvanlardan grnt alınması kararlaŖtırılmıŖ, birlik baŖkanından izin alınmasını mteakip sĖırların tartımı sırasında grntleme yapılmıŖtır. Kantarın tonajının fazla olması nedeniyle sĖırları kantarda kamera altında sabitlemek, btn sĖırları aynı

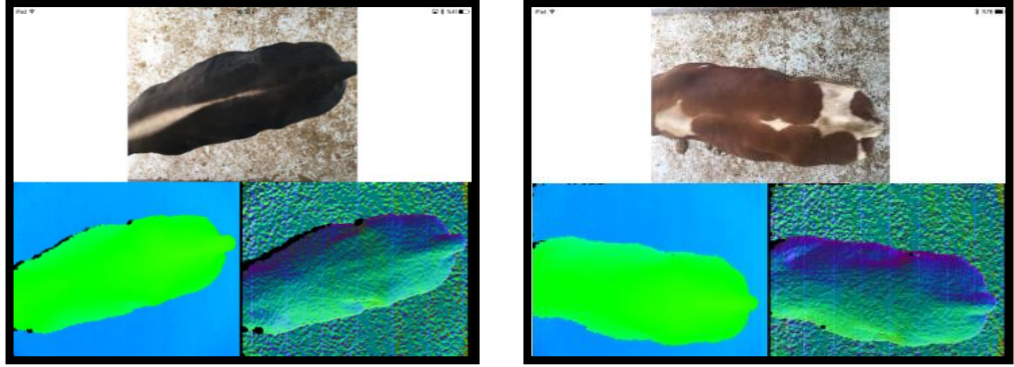
pozisyonda tutmak, tartım sonucunu merak eden yetiştiricilerin kantarın etrafında toplanması ve buna bağlı olarak hayvanların stresli oluşu gibi birçok olumsuzlukla karşılaşmıştır. Bu olumsuzluklara rağmen Apple Ipad Pro tablet bilgisayara bağlı Şekil 3.2’de görülen, Tablo 3.1’de teknik özellikleri verilen Structure Sensor üç boyutlu kamera ile 573 baş sığıra ait Şekil 3.3’teki gibi 6246 kare görüntü elde edilmiştir.



Şekil 3.2. Structure sensor kamera

Tablo 3.1. Structure sensor kamera özellikleri

TEKNİK ÖZELLİKLER	
Boyutlar	119.2mm-29mm-28mm
Ağırlık	95gr
Tavsiye edilen menzil	40cm-350cm
Hassasiyet	40cm’de 0.5mm(%0.15)-3m’de 30mm(%1)
Çözünürlük	VGA(640X480)-QVGA(320X240)
Kare hızı	Saniye başına 30/60 kare
Batarya ömrü	3-4 Saat aktif kullanım, 1000 saat bekleme
Görüntüleme alanı	Yatay 58°-Dikey 45°



Şekil 3.3. Üç boyutlu kamera ile elde edilen sığır görüntüleri

Kamera Şekil 3.4'te görüleceği üzere 265cm yükseklikte ve 5m² çekim alanıyla konumlandırılmıştır.



Şekil 3.4. Kameranın kantar üzerinde konumlandırılması

3.1.1 Hayvan materyali

Çalışmanın hayvan materyalini Erzurum İli Damızlık Koyun Keçi Yetiştiricileri Birliği canlı hayvan pazarına kurbanlık satışı için getirilen, Tablo 3.2'de ırk ve cinsiyete göre dağılımı verilen 573 baş sığırdan sadece erkek

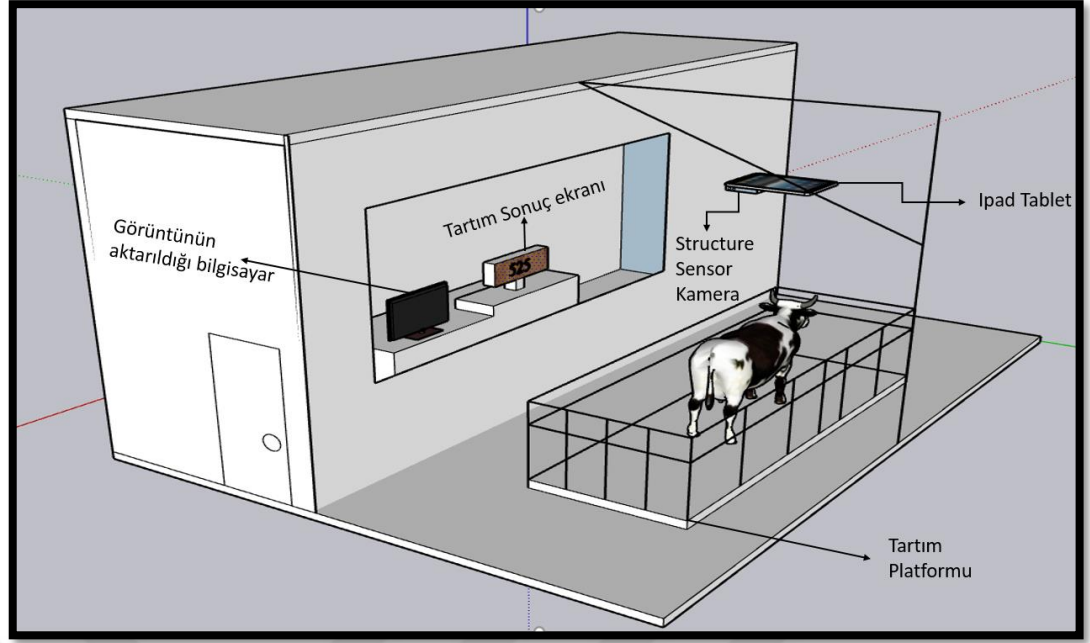
hayvanlardan oluşan ve uygun konum ve kamera açısında görüntü elde edilmiş 244 baş sığır oluşturmaktadır.

Kantarın tartım platformunun büyük olması, bütün hayvanlardan aynı pozisyonda görüntü elde edilememesine neden olmuştur. Özellikle büyük hayvanların vücudunun tamamının kameranın çekim alanına sığmaması ve yetiştiriciler tarafından hayvanların ürkmesini önlemek amacıyla başlarına çuval geçirilmesi gibi nedenlerle görüntüler kaydedilirken, hayvanın boyun kısmından itibaren kalça kısmını da kapsayacak şekilde uygun görüntü açısı yakalandığında ekran görüntüsü alınmıştır. Dişi sığırlara ait görüntüler sayıca az olması ve erkek sığırlardan morfolojik olarak farklılık gösterebileceği düşünülerek elimine edilmiş ve analize dâhil edilmemiştir.

Tablo 3.2. Hayvan materyalinin ırk ve cinsiyete göre dağılımı

Sığırın ırkı	Dişi	Erkek	Toplam
Boz	1	2	3
Esmer		22	22
Esmer Melezi	1	97	98
Doğu Anadolu Kırmızısı		7	7
Siyah Alaca		3	3
Siyah Alaca Melezi		16	16
Simental		18	18
Simental Melezi	2	261	263
Şerole Melezi		33	33
Yerli Kara	5	2	7
Tespit edilemeyen	20	83	103
Toplam	29	544	573

Görüntüler İpad'in *Air Play* özelliği ile kablosuz olarak bir dizüstü bilgisayara aktarılmıştır. Bu bilgisayarda üstten gözlemlenen hayvan uygun pozisyona geldiğinde İpad'a bluetooth aracılığı ile bağlanan ve kendi klavyesi olan *Magic Keyboard* adlı klavye yardımıyla, ekran görüntüsü kaydedilmiştir. Görüntüleri alınan hayvanlara ait ırk, küpe, cinsiyet, ağırlık ve görüntünün alındığı saatler not edilmiştir. Görüntülemenin yapılış şeması Şekil 3.5'te verilmiş, tartım sonuçları her hayvan için fotoğraflanarak kayıt altına alınmıştır.



Şekil 3.5. Görüntülemenin yapılış şeması.

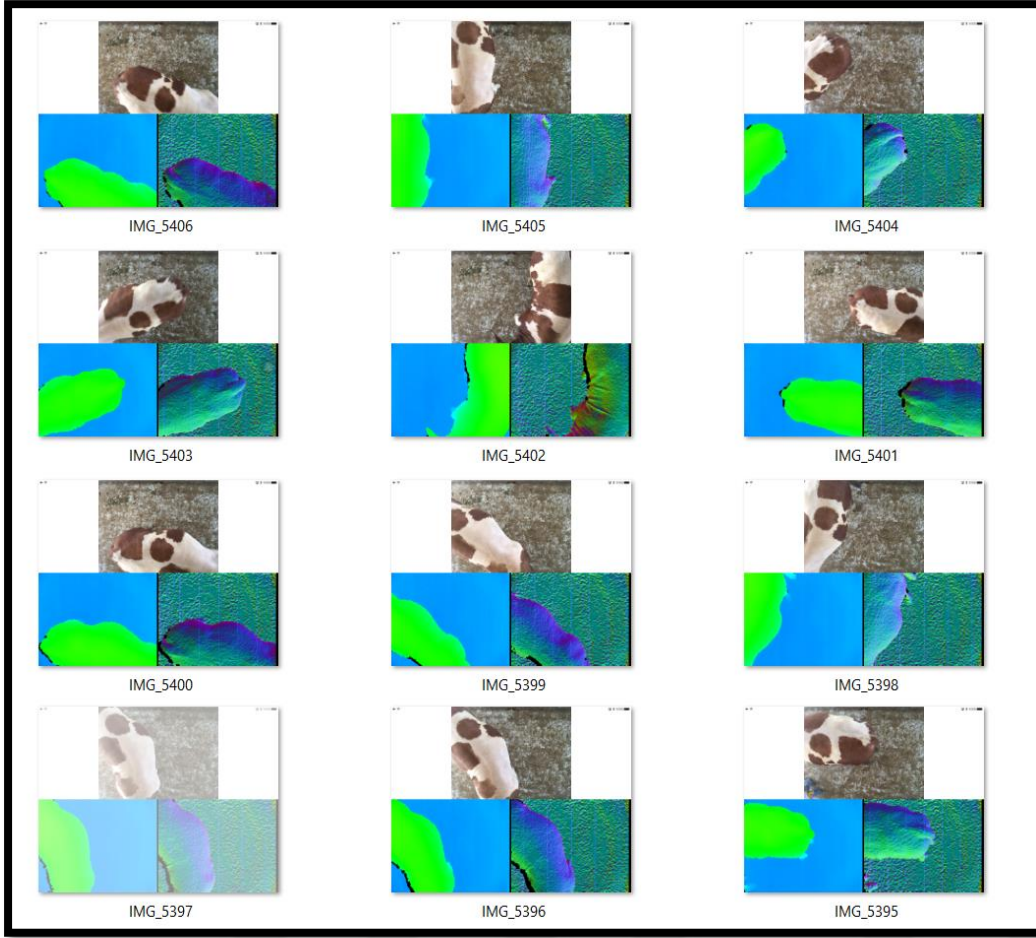
Çalışmada ağırlıkları 255 kg ile 1000 kg arasında değişen Tablo 3.3'te bazı tanımlayıcı istatistikleri verilen 244 erkek et sığına ait görüntü kullanılmıştır.

Tablo 3.3. Hayvan materyaline ait bazı tanımlayıcı istatistikler.

Sığırınırkı	Gözlem sayısı	Ortalama ağırlık (kg)	Standart hata (kg)
Şarole Melezi	2	635	70
Doğu Anadolu Kırmızıısı	6	477	41.37
Esmer Melezi	70	579	16.83
Siyah Alaca Melezi	12	495	24.9
Simental Melezi	134	652	9.47
Yerli Kara	20	421	16.13
Genel	244	600	8.68

3.1.2 Görüntü materyali

Çalışma kapsamında görüntülenen 573 baş sığırın her birinden yaklaşık 11 kare görüntü alınmış, ancak Şekil 3.6'daki gibi analiz için uygun konum veya kamera açısında olmayan hayvanlar analize dâhil edilmemiştir.



Şekil 3.6. Çalışmaya dâhil edilmeyen bir sığırın görüntüleri.

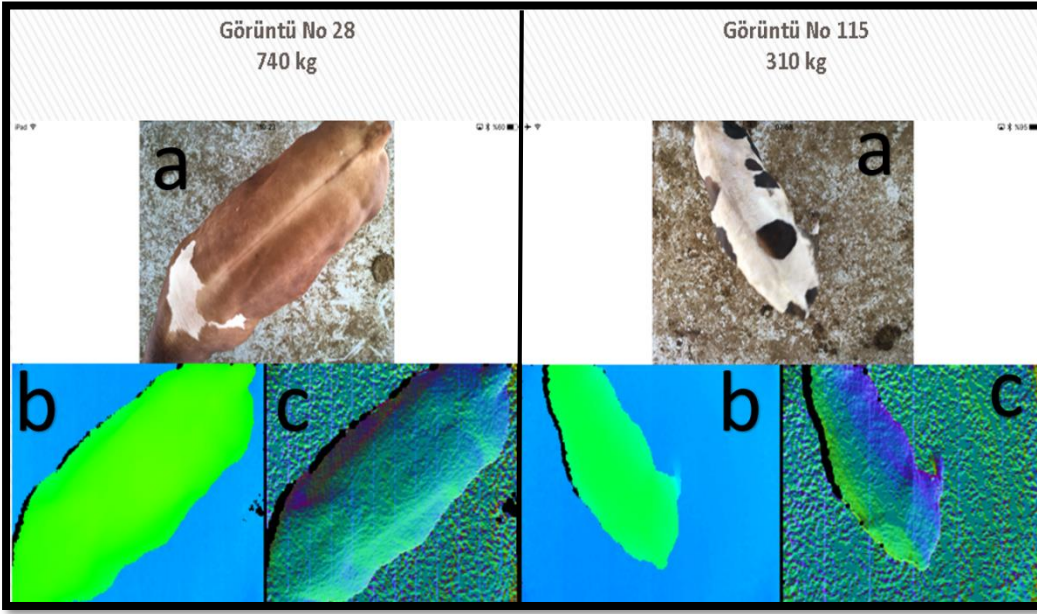
3.2 Yöntem

Bu kısımda öncelikle tez çalışmasında kullanılan görüntü işleme yöntemleri hakkında bilgi verilmiş, MATLAB komutlarıyla alan ve yükseklik hesaplamalarının nasıl yapıldığı anlatılmıştır. Daha sonra makine öğrenmesi ve makine öğrenmesinde kullanılan yaklaşımlardan yapay sinir ağları, çok değişkenli uyarlanabilir regresyon eğrileri, çoklu doğrusal regresyon ve Ki-kare otomatik etkileşim belirleme analizi hakkında bilgi verilmiştir.

3.2.1 Görüntü işleme

Dijital bir görüntü, en küçük parçası piksel adı verilen parçalardan oluşur ve her pikselin sayısal bir değeri vardır. Örneğin ikili (siyah-beyaz) bir resmin piksel değerleri 0 ve 1 rakamlarından oluşan bir matristir ve görüntünün dijital olarak gösterimi Şekil 3.7'deki gibidir.

Canlı ağırlık tahmini için elde edilen görüntüler, aşağıdaki komutlar kullanılarak işlenmiş ve makine öğrenmesi yaklaşımlarında kullanılmak üzere rakamsal veriler elde edilmiştir. Komutların anlatılabilmesi için, görüntü numarası 28 ve 115, ağırlıkları sırasıyla 740 ve 310 kg olan Şekil 3.8’deki iki hayvana ait görüntüler örnek olarak kullanılmıştır. Görüntüler incelendiğinde hayvanın gerçek görüntüsü (a), derinlik bilgisinin olduğu görüntü (b) ve yüzey (c) görüntülerinden oluşan üç kare mevcuttur.



Şekil 3.8. Örnek hesaplamaların yapıldığı farklı ağırlıklardaki iki sığırın görüntüsü: Sığırın gerçek görüntüsü (a), derinlik bilgisinin olduğu görüntü (b) ve yüzey görüntüsü (c).

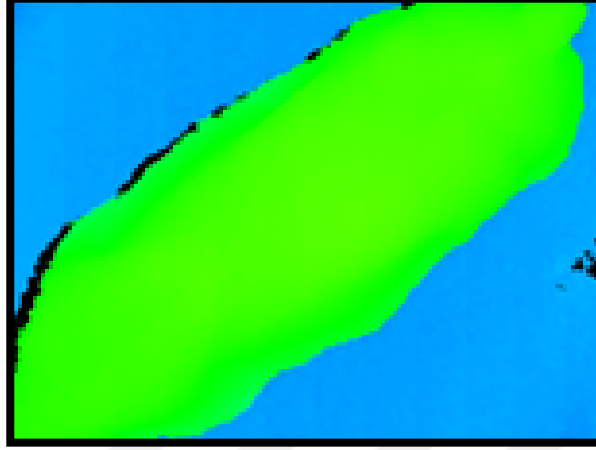
Öncelikle görüntünün uzantısı MATLAB’a tanıtılarak görüntü bir *a* değişkeninin içerisine aşağıdaki komut yardımıyla atanmıştır.

```
a=imread('C:\Users\user\Desktop\IHG\28.PNG');
```

Bu görüntüden hayvanın kapladığı alanın yüzdesinin ve hayvanın yüksekliğinin hesaplanması için kullanılacak olan Şekil 3.8’de “b” harfiyle gösterilen sol alt kısımdaki derinlik bilgisinin olduğu görüntü;

```
a=imcrop(a,[0 768 1010 768]);
```

komutuyla kırpılmış ve *imshow(a)* komutuyla Şekil 3.9’da görüldüğü gibi ekranda gösterilmiştir.

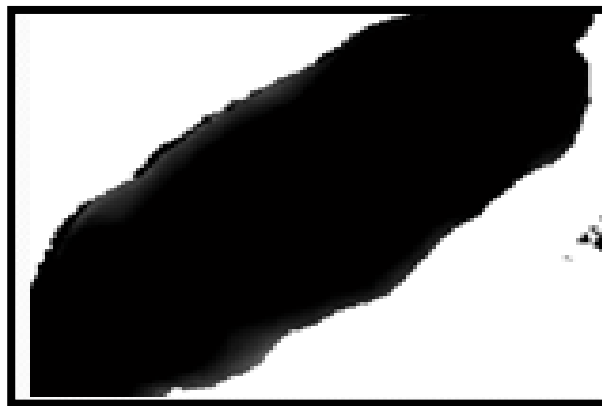


Şekil 3.9. Numarası 28 olan hayvana ait kırpılan görüntü.

Daha sonra görüntüden zemini ayıklamak için görüntünün Red, Green ve Blue (RGB) bileşenlerinden Blue (mavi) bileşeni,

```
a=a(:,:,3);
```

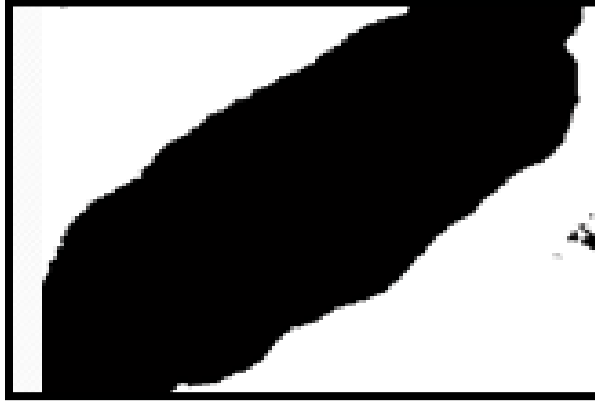
komutu kullanılarak, görüntü yeniden *imshow(a)* komutu ile Şekil 3.10’daki gibi görüntülenmiştir.



Şekil 3.10. Numarası 28 olan hayvana ait görüntünün mavi bileşeni

Bundan sonra aşağıdaki komut yardımıyla resim siyah beyaza dönüştürülerek Şekil 3.11’deki gibi görüntülenmiştir;

BW=im2bw(a); Imshow(a)



Şekil 3.11. Numarası 28 olan hayvana ait siyah beyaza dönüştürülen görüntü.

Hayvanın alanının toplam görüntü alanı içindeki payını hesaplamak için önce,

» *Alan=bwarea(BW);*

komutu kullanılmış ve Alan = 345340 sonucu elde edilmiştir. Bu sonuç aşağıdaki komutla piksel değerlerine bölünerek 1'den çıkarılmış ve siyah bölgenin yani hayvanın görüntüde kapladığı alanın yüzdesi 0.5548 (%55.48) olarak belirlenmiştir.

» *Alan=1-[Alan/(768*1010)];*

115 görüntü nolu hayvan için aşağıdaki komutların tamamı tek seferde çalıştırılarak,

a=imread('C:\Users\user\Desktop\IHG\115.PNG');

a=imcrop(a,[0 768 1010 768]);

a=a(:, :, 3);

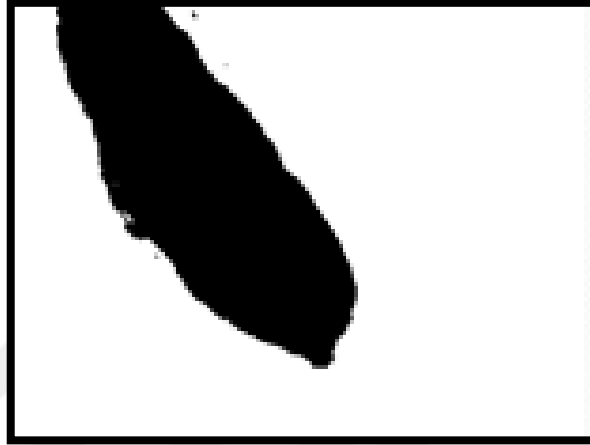
BW=im2bw(a);

imshow(BW)

$t = \text{bwarea}(BW);$

$\text{Alan} = 1 - [t / (768 * 1010)];$

115 görüntü numaralı hayvana ait alan 0.2228 yani % 22.28 ve görüntüsü Şekil 3.12'deki gibi elde edilmiştir.



Şekil 3.12. Numarası 115 olan hayvana ait siyah-beyaza dönüştürülmüş görüntü.

Daha sonra bu hesaplamaların IHG isimli klasörde bulunan bütün hayvanlar için otomatik olarak yapılmasını sağlayan aşağıdaki komutlar yardımıyla bütün hayvanların alan değerleri hesaplanmış ve bir A değişkenine kaydedilmiştir.

$\text{dosyayeri} = 'C:\Users\user\Desktop\IHG\';$

$\text{dosyaturu} = '.png';$

$\text{icerik} = \text{dir} ([\text{dosyayeri}, '*', \text{dosyaturu}]);$

$\text{Rsayisi} = \text{size}(\text{icerik}, 1);$

$\text{A} = \text{zeros}(1, \text{Rsayisi});$

$\text{for } k = 1 : \text{Rsayisi}$

$\text{string} = [\text{dosyayeri}, \text{icerik}(k, 1).name];$

```
Resim=imread(string);
```

```
Resim=imcrop(Resim,[0 768 1010 768]);
```

```
Resim=Resim(:,:,3);
```

```
BW=im2bw(Resim);
```

```
d=imadjust(Resim,[0 1],[1 0]);
```

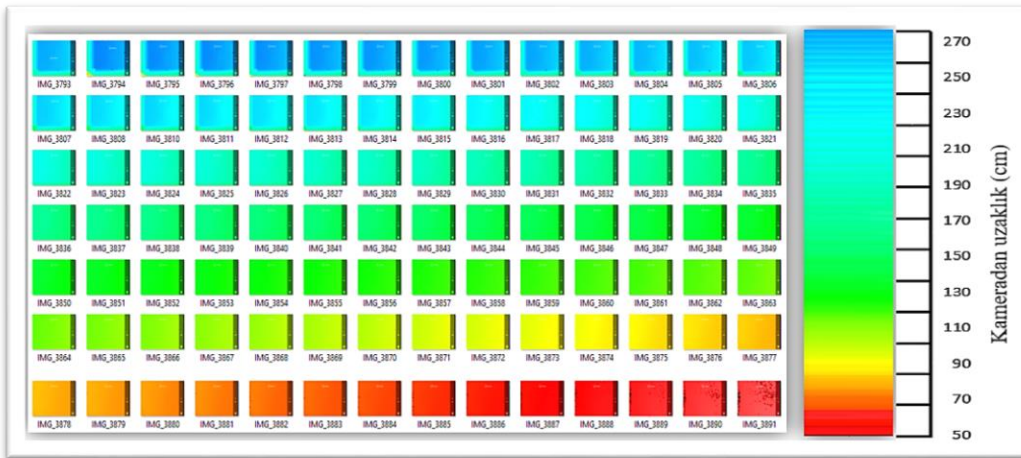
```
Alan=bwarea(d);
```

```
Alan=Alan/(768*1010);
```

```
A(k,:)=Alan;
```

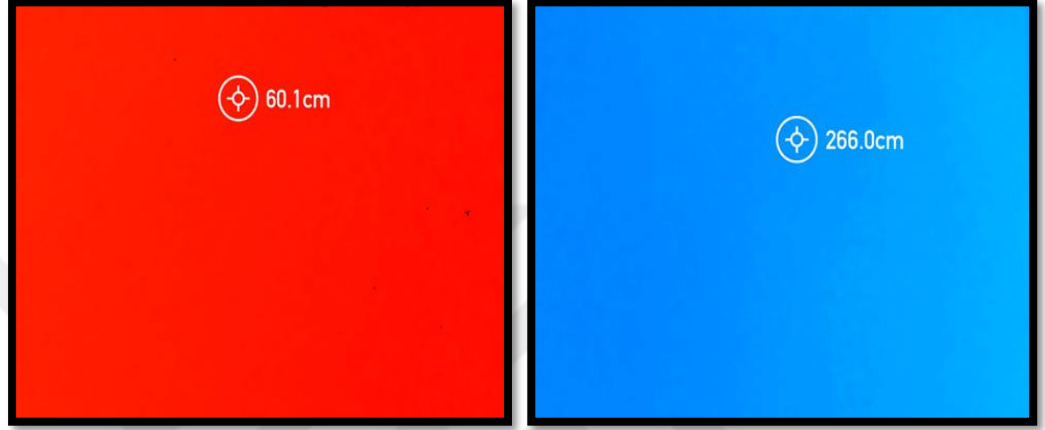
```
End
```

Sığırların alanları hesaplandıktan sonra elde edilen üç boyutlu görüntüler, sığırlarının yüksekliklerini tahmin etmek için de kullanılmıştır. Üç boyutlu kamera ile görüntü elde edilirken görüntünün iki boyutu için Ipad'in kendi kamerası kullanılırken, üçüncü boyut için gönderilen bir lazer ışığının geri dönüş hızı hesaplanarak mesafeyi tanımlayan bir renk değeri atanmaktadır.



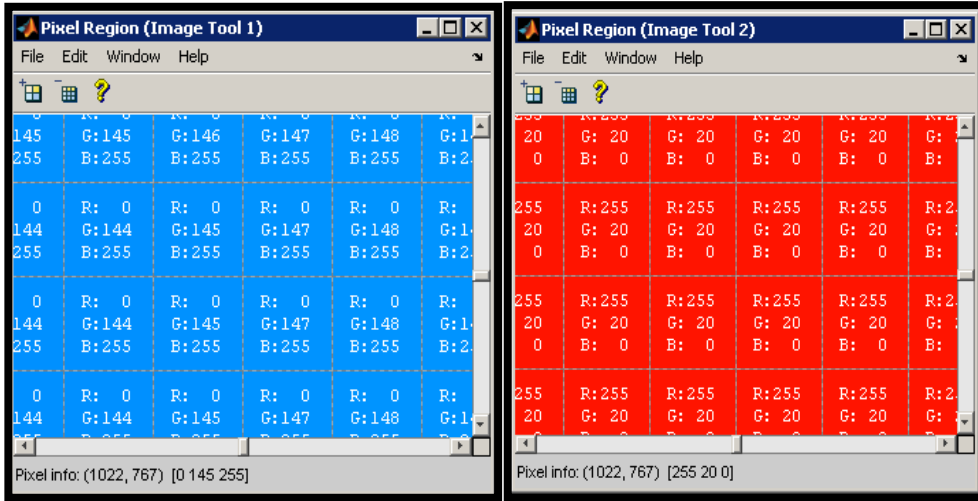
Şekil 3.13. Yükseklik tahmini için kalibrasyon görüntüleri ve renk skalası.

Tez çalışması kapsamında, structure sensor kameradaki bu renk ve mesafe ilişkisini belirlemek maksadıyla 266cm mesafeden başlanarak 50cm'ye kadar 2'şer cm aralıklarla görüntüleme yapılmış ve Şekil 3.13'te görülen renk skalası elde edilmiştir. Renk skalasındaki uzak mesafe (mavi renkli) ile yakın mesafelerin (kırmızı renkli) renk bilgilerini göstermek amacıyla iki resim Şekil 3.14'te verilmiştir.



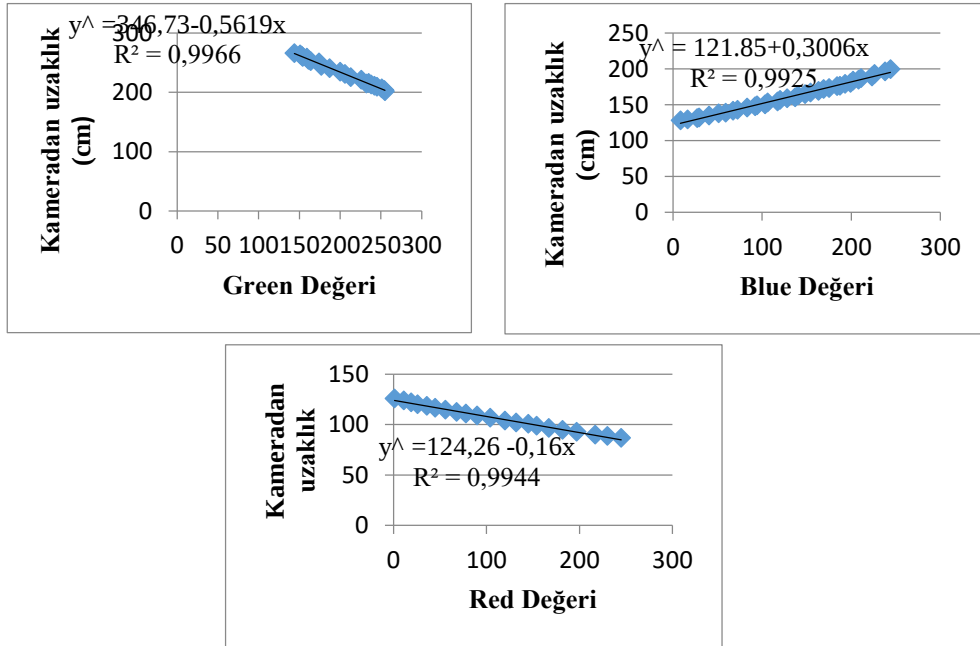
Şekil 3.14. Uzak ve yakın mesafede renk değişimi

Her bir mesafe ile görüntünün RGB değerleri arasındaki ilişkiye ait belirleme katsayısı $R^2=0.9966$ olarak hesaplanmıştır. Belirleme katsayısı iki değişkenin arasındaki pearson korelasyon katsayısının karesi olup bağımlı değişkendeki değişimin yüzde kaçının bağımsız değişken(ler) tarafından açıklandığını gösterir. Belirleme katsayısının yüksek bir değer olması nedeniyle, et sığırlarının görüntülerinden yüksekliklerini düşük bir hata ile tahmin etmek mümkün olmaktadır. Dolayısıyla, gerçek değerlerine çok yakın olan bu yükseklik verileri de yapay sinir ağlarında girdi değişkeni olarak kullanılmıştır. Şekil 3.14'te mavi renkte görülen ve 266 cm'den alınan klasördeki ilk görüntü MATLAB'de önce bir A değişkenine atanmış ve imtool(a) komutu ile resim araçlarında açılmıştır. Daha sonra "pixel region" komutu ile her bir pixel'e ait RGB değerlerine ulaşılmıştır. Örneğin 266 cm mesafeden alınan görüntünün RGB değerleri Şekil 3.15'te görüldüğü gibi R:0, G:145, B:255 olarak belirlenmiştir. Benzer şekilde 60.1cm'den alınan görüntü için RGB değerleri sırasıyla 255, 20, 0 olarak tespit edilmiştir.



Şekil 3.15. MATLAB pixel region komutu ile piksel değerlerinin görüntülenmesi

Mesafesi belli olan 244 görüntünün tümü için bu RGB değerleri alınarak kaydedilmiş daha sonra Red, Green ve Blue değerleri için Şekil 3.16'daki grafikler elde edilmiştir. Mesafe ve renk değerleri arasındaki belirleme katsayısının %100'e yakın bir değer olduğu görülmüştür. Hayvanların görüntüsünde sağrı yüksekliğine denk gelecek şekilde kuyruk kısmındaki RGB değerleri yardımıyla, daha önce anlatıldığı şekilde MATLAB'de imtool komutu kullanılarak, bu değerlere karşılık gelen hayvan yükseklikleri tahmin edilmiştir.



Şekil 3.16. Mesafe ve renk değerleri arasındaki ilişkiyi gösteren grafikler

Hayvanların görüntüleri kaydedilirken duruş pozisyonları aynı olmadığı için bu işlem otomatik olarak yapılamamış, bu nedenle bütün görüntüler tek tek incelenerek sığırların sağrı yükseklikleri hesaplanmıştır.

3.2.2 Yapay sinir ağları

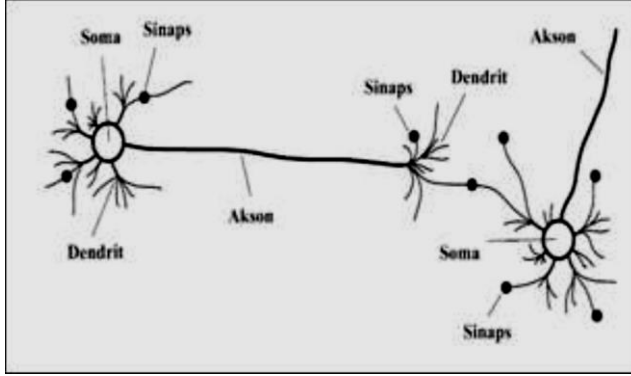
Yapay zekanın bir alt dalı olan makine öğrenmesi, istatistik ve bilgisayar biliminin hesaplama gücünden yararlanarak makinelerin insanlara benzer şekilde bir olayı öğrenebilmesini, bir konu hakkında tahminde bulunmasını ve karar verebilmesini sağlayan tekniklerdir. Bu amaçla makine öğrenmesinde birçok istatistiksel yöntem ve matematiksel modelden yararlanılmaktadır. Bu tez kapsamında YSA, MARS, ÇDR ve CHAID analizleri kullanılarak makine öğrenmesi gerçekleştirilmiştir.

Günümüz dünyasında bilgisayarlar yaşantımızın vazgeçilmez bir parçası olmuştur. Çalışma prensipleri her geçen gün geliştirilen bu makineler, bazı durumlarda konunun uzmanlarından daha iyi karar verebilecek aygıtlar haline gelmiştir. Öyle ki cebimizdeki telefonlar dahi akıllı olarak nitelendirilmektedir. Bilgisayar ve telefonların öğrenebilmelerini sağlayan sistemler, genel olarak yapay zeka olarak adlandırılmaktadır. Bu sistemlerin çok hızlı gelişmesi bazı bilim adamlarını tedirgin etmiştir. Bu konuda dünyanın önde gelen bilim adamlarından Stephen Hawking “Yapay zeka, kendisini geliştirmeyi sürdürebilir ve hatta kendisini yeniden biçimlendirebilir. Son derece yavaş bir biyolojik evrimle sınırlı olan insanlar, bu tür bir güçle yarışamaz” şeklindeki endişesini dile getirmiştir. Gelecek için endişe etmeli miyiz? Tartışılabilir ancak bugün, bu sistemler insanoğlu için hayatı azami derecede kolaylaştırmaktadır.

Yapay sinir ağları, insan beyninin belirli bir öğrenme görevini yerine getirme prensibini esas alan bir makine öğrenmesi yaklaşımıdır. YSA ile tahmin, sınıflandırma, kümeleme kontrol ve karar verme gibi problemlere hızlı çözümler üretilebilir (Uğur, 2010).

İnsan beyninde yaklaşık 100 milyar sinir hücresi bulunur. İnsan düşünüp karar verirken ya da bir uyarana tepki verirken bu sinir hücrelerinden bazıları elektriksel olarak bilgi alışverişinde bulunur. Biyolojik bir sinir hücresi Şekil

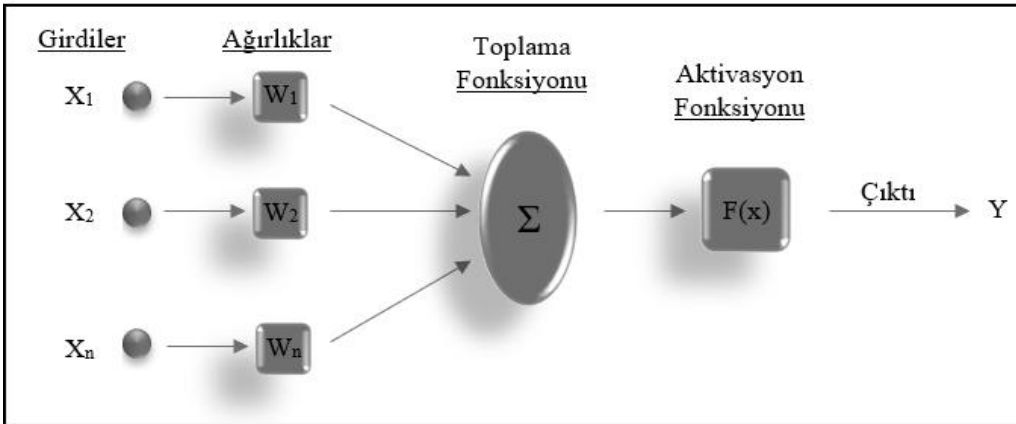
3.17’de görüleceği üzere hücre gövdesi, dentrit ve axon adında üç önemli kısımdan oluşmaktadır.



Şekil 3.17. Biyolojik sinir hücreleri (Akıllı ve Atıl, 2014’ten).

Biyolojik bir sinir hücresinde axonlar aracılığıyla hücre gövdesine gelen bilgi, işlendikten sonra gerek görülmesi durumunda (sinir hücresinin görevine bağlı olarak belirli bir eşik değerin üzerinde ise) dentritler aracılığıyla axon uçlarında bulunan ve fiziksel olarak var olmayan sinaps adı verilen boşluklara iletilir ve elektriksel olarak diğer sinir hücrelerine sinyal gönderilir. Bu şekilde iki sinir hücresi bilgi alışverişinde bulunur.

Nöron (yapay sinir hücresi) ise biyolojik bir sinir hücresinin fonksiyonlarının mantıksal-matematiksel olarak taklit etmeye çalışan yapay sinir ağlarının en temel elemanıdır. Şekil 3.18’de bir nöronun yapısı gösterilmiştir.



Şekil 3.18. Bir nöronun yapısının şematik gösterimi

Bir nöron; girdi, ağırlıklar, toplama ve aktivasyon fonksiyonları ile çıktı elemanlarından oluşur. Nöronlar bir araya gelerek sinir ağlarını oluştururlar. Nöronun elemanları aşağıda açıklanmıştır;

❖ Girdiler

Şekil 3.18’de X harfi ile gösterilen elemanlar bir nörona gelen bilgilerdir ve girdi olarak adlandırılmaktadır. Ağın öğrenmesi istenilen örneklerden gelen bilgiler girdi olabileceği gibi hatanın geri yayılması esnasında hücrenin kendisinden gelen bilgiler de girdi olarak adlandırılmaktadır.

❖ Ağırlıklar

Nörona gelen girdiler rastgele olarak atanan başlangıç ağırlık değerleri ile çarpılarak toplama fonksiyonuna gönderilir. Daha sonra bu ağırlıkların değerleri güncellenmek suretiyle ağın tahmin hatası azaltılmaya çalışılmaktadır. Öğrenme gerçekleştiğinde, ağın bilgileri ağırlıklarda saklanmakta, bir tahmin yapılmak istendiğinde bu ağırlık değerleri kullanılarak tahmin yapılmaktadır.

Tablo 3.4. Bazı toplama fonksiyonları (Çayıroğlu, 2019’dan)

Toplam $Net = \sum_{i=1}^N X_i * W_i$	Ağırlık değerleri girdiler ile çarpılır ve bulunan değerler birbirleriyle toplanarak Net girdi hesaplanır.
Çarpım $Net = \prod_{i=1}^N X_i * W_i$	Ağırlık değerleri girdiler ile çarpılır ve daha sonra bulunan değerler birbirleriyle çarpılarak Net Girdi hesaplanır.
Maksimum $Net = \max(X_i * W_i)$	n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en büyüğü Net girdi olarak kabul edilir.
Minimum $Net = \min(X_i * W_i)$	n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en küçüğü Net girdi olarak kabul edilir.
Çoğunluk $Net = \sum_{i=1}^N \text{sgn}(X_i * W_i)$	n adet girdi içinden girdilerle ağırlıklar çarpıldıktan sonra pozitif ile negatif olanların sayısı bulunur. Büyük olan sayı hücrenin Net girdisi olarak kabul edilir.
Kümülatif toplam $Net = Net(eski) + \sum_{i=1}^N X_i * W_i$	Hücreye gelen bilgiler ağırlıklı olarak toplanır. Daha önce hücreye gelen bilgilere yeni hesaplanan girdi değerleri eklenerek hücrenin Net girdisi hesaplanır.

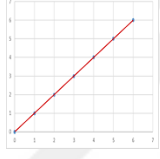
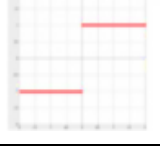
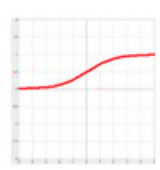
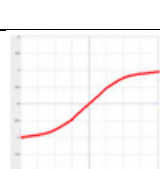
❖ Toplama fonksiyonu

Girdi bilgileri, Tablo 3.4'te görülen ve YSA'nın öğrenmesi istenilen olaya göre seçilmesi gereken toplama fonksiyonlarından birinden geçirilerek Net girdi hesaplanır.

❖ Aktivasyon fonksiyonu

YSA'da toplama fonksiyonundan gelen bilgiler Tablo 3.5'te görülen aktivasyon fonksiyonları arasından tercih edilen birinden geçirilerek girdi değerine karşılık üretilecek çıktı değeri hesaplanır. ÇKA modelinde bu fonksiyonun türevi alınabilir bir fonksiyon olması gerekir. Tez çalışması kapsamında bu fonksiyonlardan sigmoid fonksiyonu kullanılmıştır.

Tablo 3.5. Bazı aktivasyon fonksiyonları (Çayıroğlu, 2019)

Doğrusal (Lineer) Aktivasyon Fonksiyonu		$F(Net) = A * Net$ (A sabit bir sayı)	Doğrusal problemleri çözmek için kullanılır. Toplama fonksiyonundan çıkan sonuç, belli bir katsayı ile çarpılarak hücrenin çıktısı olarak hesaplanır.
Adım (Step) Aktivasyon Fonksiyonu		$F(Net) = \begin{cases} 1 & \text{Eğer } Net > \text{Eşik değeri} \\ 0 & \text{Eğer } Net \leq \text{Eşik değeri} \end{cases}$	Gelen Net girdinin belirlenen bir eşik değerin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır.
Sigmoid Aktivasyon Fonksiyonu		$F(Net) = \frac{1}{1 + e^{-Net}}$	Sigmoid aktivasyon fonksiyonu sürekli ve türevi alınabilir bir fonksiyondur. Doğrusal olmayışı dolayısıyla yapay sinir ağı uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir
Tanjant Hiperbolik Aktivasyon Fonksiyonu		$F(Net) = \frac{e^{Net} - e^{-Net}}{e^{Net} + e^{-Net}}$	Sigmoid fonksiyonuna benzer. Sigmoid fonksiyonunda çıktı değerleri 0 ile 1 arasında değişirken bu fonksiyonun çıktı değerleri -1 ile 1 arasında değişmektedir.
Eşik Değer Fonksiyonu		$F(Net) = \begin{cases} 0 & \text{Eğer } Net \leq 0 \\ Net & \text{Eğer } 0 < Net < 1 \\ 1 & \text{Eğer } Net \geq 1 \end{cases}$	Gelen bilgiler 0'dan küçük ya da eşit olduğunda 0 çıktısı, 1' den büyük ya da eşit olduğunda 1 çıktısı, 0 ile 1 arasında olduğunda ise yine kendisini veren çıktılar üretir.
Sinüs Aktivasyon Fonksiyonu		$F(Net) = \sin(Net)$	Öğrenilmesi düşünülen olayların sinüs fonksiyonuna uygun dağılım gösterdiği durumlarda kullanılır

❖ Çıktı

Şekil 3.18’de Y harfi ile gösterilen eleman, bir nörona gelen girdi bilgilerinin çeşitli hesaplamalar ve fonsiyonlardan geçirildikten sonra elde edilen çıktı bilgileridir. Bu bir tek çıktı olabileceği gibi birden fazla çıktı da olabilmektedir. Geri yayımlı ağda, ağın ürettiği çıktı bilgileri, ağın öğrenmesi beklenen çıktı bilgileri ile karşılaştırılarak elde edilen hatanın minimize edilmesi amaçlanır.

3.2.2.1 Yapay sinir ağlarında kullanılan öğrenme kuralları

Yapay sinir ağlarında çoğu öğrenme kuralı Hebb öğrenme kuralına dayanmaktadır. Bilinen en eski öğrenme kuralı olan Hebb kuralı bir hücrenin kendisi aktif ise bağlı bulunduğu hücreyi aktif yapmaya, pasif ise pasif yapmaya çalışır. YSA’da en sık kullanılan öğrenme kuralları Kohonen, Hopfield ve Delta öğrenme kurallarıdır. Delta öğrenme kuralı, Hebb kuralının biraz daha geliştirilmiş halidir. Bu kurala göre beklenen çıktı ile gerçekleşen çıktı arasındaki farkın azaltılması için ağırlıkların sürekli değiştirilmesi gerekir ve ağın ürettiği çıktı ile beklenen çıktı arasındaki farkın yani hatanın kareler ortalamasının en az olması hedeflenmektedir (Öztemel, 2003). Delta kuralı için örnek bir hesaplama Tablo 3.6’da verilmiştir.

Tablo 3.6. Delta öğrenme kuralıyla örnek bir hesaplama

X	Y	Ç		Ç	Δ		Ç
0	0	0	$0 \times 1 + 0 \times 1 = 0 < 5$	0	0	$0 \times 3 + 0 \times 3 = 0 < 5$	0
0	1	0	$0 \times 1 + 1 \times 1 = 1 < 5$	0	0	$0 \times 3 + 1 \times 3 = 3 < 5$	0
1	0	0	$1 \times 1 + 0 \times 1 = 1 < 5$	0	0	$1 \times 3 + 0 \times 3 = 3 < 5$	0
1	1	1	$1 \times 1 + 1 \times 1 = 2 < 5$	0	1	$1 \times 3 + 1 \times 3 = 6 > 5$	1

Tablo 3.6’da görülen X ve Y değişkenleri girdi değerleridir. Çıktı değeri Ç, eşik değeri b, gözlenen değerler ile beklenen değerler arasındaki fark yani delta Δ simgesiyle ve ağırlıklar W harfleriyle gösterilir. Delta öğrenme kuralı anlatılırken mühendislik çalışmalarında sıklıkla kullanılan mantıksal AND operatörü değerleri kullanılmıştır. Mantıksal AND operatörüne göre $X=0$ ve $Y=0$ iken $\Ç=0$; $X=0$ ve $Y=1$ iken aynı zamanda $X=1$, $Y=0$ iken $\Ç=0$; $X=1$ ve $Y=1$ iken ise $\Ç=1$ değerini almaktadır. Ağırlık değerleri (W_1 , W_2), öğrenme oranı (λ) ve eşik değeri (b) başlangıçta rastgele olarak seçilir ve küçük değerler alınması durumunda ağın

eğitiminin uzaması, büyük değerler alınması durumunda ise ağırlık lokal minimuma takılması söz konusu olabilir. En iyi değerlerin seçimi için geliştirilmiş herhangi bir yöntem yoktur (Öztemel, 2003).

Örnek uygulamada hesaplamaların kısa sürmesi ve konunun anlaşılmasını kolaylaştırmak amacıyla ağırlıklar 1, öğrenme oranı 2 ve eşik değeri 5 olarak alınmıştır. Toplama fonksiyonu olarak Tablo 3.4'te formülü gösterilen fonksiyonlardan toplam fonksiyonu, aktivasyon fonksiyonu olarak ise Tablo 3.5'te formülü gösterilen adım fonksiyonu kullanılmıştır. Hesaplamalar Eşitlik 3.1-3.7'de verilen denklemler yardımıyla aşağıdaki gibi olacaktır;

Toplama fonksiyonu;

$$\Sigma = X_1 * W_1 + Y_1 * W_2 \quad (3.1)$$

$$\Sigma = 0 * 1 + 0 * 1 = 0 \text{ olarak hesaplanmıştır.}$$

Aktivasyon fonksiyonu eşik değeri 5 olarak alındığından;

$$\text{Eğer } \Sigma > 5 \Rightarrow \zeta = 1, \Sigma < 5 \Rightarrow \zeta = 0 \text{ ise}$$

$\Sigma < 5$ olduğundan örneğimiz için $\zeta = 0$ olacaktır.

Beklenen çıktı değeri de 0 olduğundan beklenen değeri ile gözlenen değeri arasındaki fark yani $\Delta = 0$ olacaktır. Bu durumda ağırlıklarda bir güncelleme yapılmaksızın bir sonraki örnek veri girdisi olarak ağırlığa sunulur. Hesaplamalar aşağıdaki gibidir;

Toplama fonksiyonu Eşitlik 3.2 yardımıyla;

$$\Sigma = X_2 * W_1 + Y_2 * W_2 \quad (3.2)$$

$$\Sigma = 0 * 1 + 1 * 1 = 1 \text{ olarak hesaplanacaktır.}$$

Aktivasyon fonksiyonu ise eşik değeri 5 olarak alındığından;

Eğer $\Sigma > 5 \Rightarrow \zeta = 1$, $\Sigma < 5 \Rightarrow \zeta = 0$ ise;

$\Sigma < 5$ olduğundan örneğimiz için yine $\zeta = 0$ olacaktır.

Yine ağırlıklarda bir güncelleme yapılmayarak bir sonraki girdi değerleri ağı gösterilir. Bir sonraki girdi değerleri aynı olduğundan ağı çıktısı olarak 0 değerini üreteceği açıktır. Bu nedenle işlem tekrar edilmeyecek bir sonraki veri ile işlemler yapılacaktır. Son veride hem X değişkeni hem de Y değişkeni 1 değerlerini almıştır. O halde hesaplamalar aşağıdaki gibi olacaktır.

Toplama fonksiyonu Eşitlik 3.3 yardımıyla;

$$\Sigma = X_4 * W_1 + Y_4 * W_2 \quad (3.3)$$

$\Sigma = 1 * 1 + 1 * 1 = 2$ olarak hesaplanmıştır.

Aktivasyon fonksiyonu ise eşik değeri 5 olarak alındığından

Eğer $\Sigma > 5 \Rightarrow \zeta = 1$, $\Sigma < 5 \Rightarrow \zeta = 0$ ise;

$\Sigma < 5$ olduğundan örneğimiz için yine $\zeta = 0$ olacaktır.

Ağı yine çıktısı olarak sıfır değerini üretmektedir. Ancak beklenen değeri bu aşamada 1 olduğundan gözlenen değeri 1 birimlik fark oluşacaktır. Yani $\Delta = 1$ olmaktadır. Bu aşamada 1 birimlik farkın ağı elemanlarına dağıtılması yani ağırlıkların güncellenmesi gerekmektedir. Bu noktada bir diğer etken ise öğrenme oranıdır. Ağırlıklar güncellenirken öğrenme oranı dikkate alınır. Yeni ağırlıklar Eşitlik 3.4 ve 3.5 yardımıyla;

$$W_3 = W_1 + \lambda * X_4 \quad (3.4)$$

$$W_4 = W_2 + \lambda * Y_4 \quad (3.5)$$

$W_3 = 1 + 2 * 1 = 3$ olarak ve

$W_4 = 1 + 2 * 1 = 3$ olarak hesaplanmıştır.

Güncellenen yeni ağırlıklar ile hesaplamalar yeniden yapılacak olursa;

Toplama fonksiyonu Eşitlik 3.6 yardımıyla;

$$\Sigma = X_1 * W_3 + Y_1 * W_4 \quad (3.6)$$

$\Sigma = 0 * 3 + 0 * 3 = 0$ olarak hesaplanmıştır.

Aktivasyon fonksiyonu ise eşik değer 5 olarak alındığından;

Eğer $\Sigma > 5 \Rightarrow \zeta = 1$, $\Sigma < 5 \Rightarrow \zeta = 0$ ise

$\Sigma < 5$ olduğundan örneğimiz için $\zeta = 0$ olacaktır.

Diğer iki girdi değerleri için de bu hesaplamalar yapılacak olursa çıktı değerinin 0 olduğu görülecektir. Yine beklenen değerle bir fark oluşmadığı için ağırlıklarda herhangi bir güncelleme yapılmaz. Son girdi değerleri için işlem aşağıdaki gibi olacaktır.

Toplama fonksiyonu Eşitlik 3.7 yardımıyla;

$$\Sigma = X_4 * W_3 + Y_4 * W_4 \quad (3.7)$$

$\Sigma = 1 * 3 + 1 * 3 = 6$ olarak hesaplanmıştır.

Aktivasyon fonksiyonu ise eşik değer 5 olarak alındığından;

Eğer $\Sigma > 5 \Rightarrow \zeta = 1$, $\Sigma < 5 \Rightarrow \zeta = 0$ ise

$\Sigma > 5$ olduğundan $\zeta = 1$ olacaktır.

Son aşamada beklenen değer 1 olduğundan ağırlık ürettiği çıktı değeri ile bir fark oluşmamıştır. Diğer üç girdi değişkeni içinde beklenen ve gözlenen arasında

fark olmadığından güncellenen yeni ağırlıklarla ağın eğitimi tamamlanmış olur. Yani mantıksal AND operatörü öğrenilmiştir denilir.

Ağın öğrenmesi, böyle basit bir örnekte tek bir nöron ve tek katmanlı algılayıcı ile mümkün olabilir. Ancak sinir ağları genellikle daha karmaşık problemlerin çözümlerinde ve doğrusal olmayan ilişkiler incelendiğinde tercih edilmektedir. Doğrusal olmayan veri yapılarından en bilineni XOR problemi ve bu problem yapay sinir ağları çalışmalarının durma noktasına gelmesine neden olmuştur. Öyle ki Minsky and Papert (1969) tarafından yazılan “Perceptron” isimli kitapta algılayıcıların XOR problemini çözemediği ispatlanınca, YSA için yaklaşık 20 sene süren bir duraklama devrine girilmiştir. YSA çalışmalarının yeniden hız kazanması çok katmanlı algılayıcı modelinin bulunması ile olmuştur.

3.2.2.2 Yapay sinir ağlarında öğrenme stratejileri

Genel olarak YSA’da üç farklı öğrenme stratejisi kullanılmaktadır. Bunlar eğitilmiş öğrenme, destekleyici öğrenme ve eğitimsiz öğrenme stratejileridir.

❖ Eğitilmiş öğrenme

Ağın öğrenmesi istenilen çıktılar ve bu çıktıların öğrenilmesi için gereken girdilerin ağa gösterildiği strateji, eğitmenli öğrenme stratejisi olarak adlandırılmaktadır (Haykin, 2009). Çok katmanlı algılayıcı modeli bu stratejiyle geliştirilmiştir. Çalışmamız kapsamında ağa canlı ağırlık çıktısı ve bu çıktıyı öğrenebilmesi için alan ve yükseklik girdileri verilmiştir. Bu nedenle çalışmamızda kullandığımız strateji, eğitilmiş öğrenme stratejisidir.

❖ Destekleyici öğrenme

Ağa girdi değişkenlerinin sunulduğu ancak çıktı değişkenlerinin verilmediği bunun yerine üretilen çıktıların doğru veya yanlış olduğu ile ilgili bir sinyal verildiği ağ stratejisidir (Öztemel, 2003).

❖ Eğitimsiz öğrenme

Ağa bir çıktı ya da sinyal verilmeyip ağın çıktıları kendisinin ürettiği ağ stratejisidir. Çıktıların ne anlama geldiği kullanıcı tarafından yorumlanmaktadır. Genellikle kümeleme problemlerinin çözümünde kullanılmaktadır (Öztemel, 2003).

3.2.2.3 Yapay sinir ağı modelleri

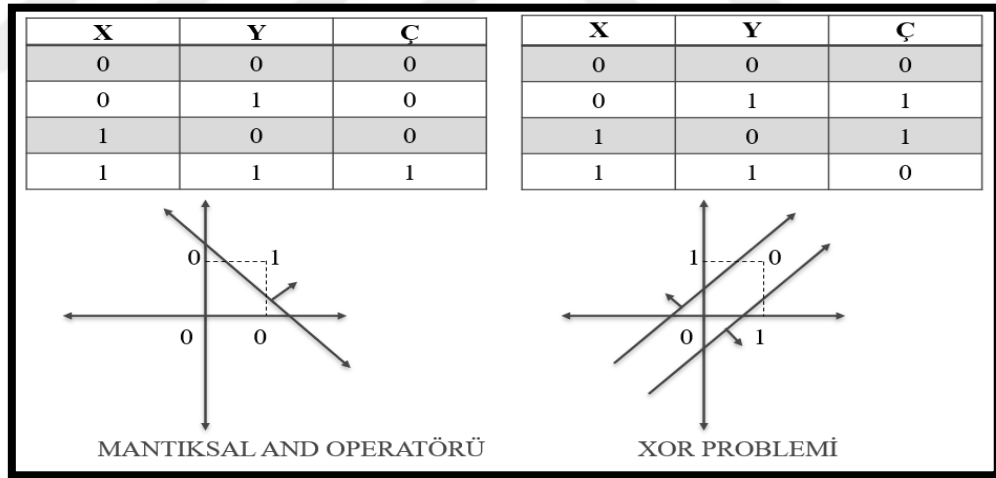
YSA fikri ortaya atıldığı günden günümüze kadar farklı problemlere çözüm bulmak amacıyla yeni modeller geliştirilmeye devam edilmiştir. Bu modellerden bazıları (Öztemel, 2003; Ghosh-Dastidar and Adeli 2009; Haykin, 2009) aşağıda verilmiştir;

- Tek katmanlı algılayıcı
- Basit algılayıcı modeli (perceptron)
- Adaline/Madaline
- Çok katmanlı algılayıcı (eğitilmiş öğrenme)
- LVQ modeli (destekleyici öğrenme)
- ART ağları (eğitimsiz öğrenme)
- SOM ağları
- Evrimsel yapay sinir ağları ve
- İğnecikli yapay sinir ağlarıdır.

Çok katmanlı algılayıcı modeli, YSA çalışmalarının durma noktasına geldiği bir dönemde geliştirilmiş ve YSA tarihinde yeni bir dönemin başlamasına neden olmuştur. Bu model mühendislik problemlerinin hemen hemen hepsine çözümler üretebilecek güçtedir (Öztemel, 2003). Özellikle sınıflama, tanıma ve tahmin problemleri için çok önemli bir çözüm aracı olması nedeniyle çalışmamızda bu model kullanılmıştır. Bu nedenlerle bu bölümde YSA modellerinden sadece ÇKA modelinin tanıtımı yapılacaktır.

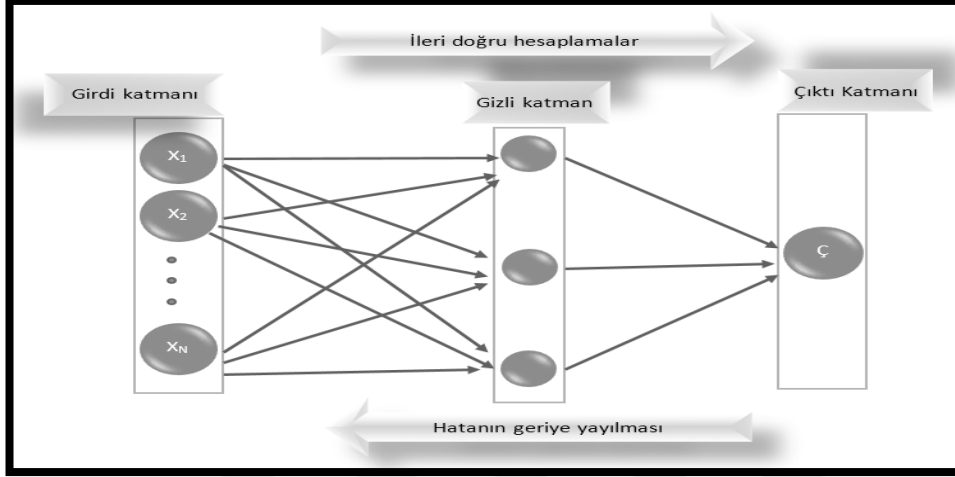
❖ Çok Katmanlı Algılayıcı (ÇKA)

YSA’da girdi katmanı ile çıktı katmanı arasına gizli katman ya da katmanlar eklendiğinde doğrusal olmayan veri yapılarının öğrenilmesinde başarı sağlanmaktadır. Şekil 3.19’da görülen Mantıksal AND operatörü ve XOR operatörünün karşılaştırılması incelenecek olursa, mantıksal AND operatöründe değişkenler arasında doğrusal bir ilişki olduğundan tek bir doğru çizerek sınıflara ayrılabilir. Ancak XOR operatöründe ise tek bir eşik değerinin üzerinde kalan kısım ele alındığında yani $X=0$ ve $Y=0$ iken çıktı değerinin 0 olması sağlanabilirken $X=1$ ve $Y=1$ iken çıktı değerinin 0 olması sağlanamamaktadır. Daha önce mantıksal AND operatöründe anlatılan hesaplamalar dikkate alınırsa 5 eşik değerinin üstünde kalan kısım için 1 çıktı değeri üretilmesi sağlanmıştı. Burada XOR operatörünün çözümünde bir gizli katman dolayısıyla ikinci bir eşik değeri -5 olarak alınırsa $X=1$ ve $Y=1$ iken 0 çıktısının üretilmesi sağlanabilir. Yani +5 eşik değerinin üstü ve -5 eşik değerinin altında 1 çıktısı üretilirken arada kalan değerler için 0 çıktısı üretilmekte ve ağın eğitimi bu şekilde tamamlanmaktadır.



Şekil 3.19. Mantıksal AND operatörü ve XOR operatörünün karşılaştırılması

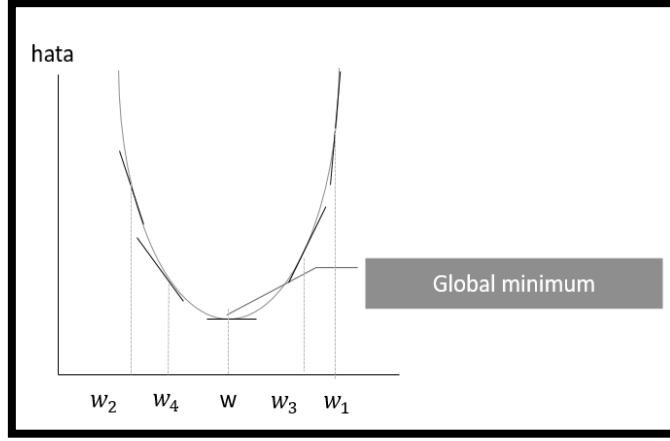
Şekil 3.20’de yapısı gösterilen ileri beslemeli, geri yayımlı çok katmanlı algılayıcı (Feed forward, backpropagation multilayer perceptron) modelinde amaç beklenen çıktı değerleri ile gözlenen çıktı değerleri arasındaki hatanın kareler ortalamasının azaltılmasıdır.



Şekil 3.20. Çok katmanlı ileri beslemeli geri yayımlı yapay sinir ağlarının yapısı.

❖ İleri beslemeli, geri yayımlı ağlar

İleri beslemeli geri yayımlı yapay sinir ağlarında iki temel adım vardır. Öncelikle ileri doğru, girdi bilgileri ile ağırlıklar bir toplama fonksiyonundan geçirilerek bir çıktı değeri üretilir, daha sonra bu çıktı değerleri ağın öğrenmesi istenen çıktı değerleri ile karşılaştırılarak hata hesaplanır, hatanın azaltılması için geriye doğru hesaplama yapılarak ağırlıklar güncellenir. Hatanın azaltılması bir en iyileme problemidir ve YSA’da hatayı en aza indirmek amacıyla çeşitli optimizasyon tekniklerinden yararlanılmaktadır. En temel yöntem Gradyan azaltma (GD) yöntemi olup bu yöntemin amacı türevi alınabilir bir fonksiyon yardımıyla Şekil 3.21’de görülen hata uzayında global minimuma ulaşmak için eğimin azaltılmasıdır. Literatürde dik iniş, eğim azaltma ve gradyan iniş olarak da geçen bu yöntemde hata eğrisi için türevi alınabilir bir nokta seçilerek eğim hesaplanır, daha sonra eğimin tersi yönünde giderek eğim adım adım azaltılmak suretiyle minimuma ulaşılmaya çalışılır. Bu azaltma işlemi için gerçekleşen her bir adım iterasyon olarak adlandırılmaktadır.



Şekil 3.21. Hata uzayında gradyan azaltmanın gösterimi.

Şekil 3.21’de görülen eğimin sıfır olduğu w noktası hatanın en az olduğu noktadır. Bu değerin sıfır olması arzu edilir ancak her zaman mümkün olmamaktadır. Bu nedenle belirli bir hata toleransı kabul edilerek minimum gradyan değeri (GD) belirlenir.

3.2.2.4 Yapay sinir ağlarında ağ performansını etkileyen parametreler

YSA’da ağ performansını etkileyen pek çok parametre vardır. Bunlardan tez çalışması kapsamında etkinliği araştırılanlar aşağıda özetlenmiştir.

❖ Momentum güncelleme katsayısı (MU)

Öğrenme algoritmasının lokal minimuma takılmaması için geliştirilmiş bir katsayıdır. Bu çalışmada birisi MATLAB tarafından önerilen (0.001) olmak üzere üç farklı momentum güncelleme (momentum update) katsayısı (0.001, 0.005 ve 0.01) çeşitli ön denemeler sonucunda etkili olabileceği düşünülerek eğitim denemelerinde kullanılmıştır.

❖ Öğrenme oranı (LR)

Öğrenme oranı (learning rate) ağırlıkların ne kadar hızlı güncelleneceğini belirleyen bir katsayıdır. Bu çalışmada yapılan ön çalışma sonucunda üç farklı

öğrenme oranı (0.01, 0.05 ve 0.1) dikkate alınarak öğrenme oranının etkinliği incelenmiştir.

❖ **Döngü sayısı**

Döngü sayısı bir durdurma kriteri olarak kullanılmaktadır. Bu kriterin doğru seçilmediği durumda ağıın fazla eğitilmesine (overtraining), bu da eğitim setinde çok az hata olmasına karşın test setinde yüksek hatalar oluşmasına neden olmaktadır. Yani ağ öğrenmesi gerekirken ezberlemektedir. Aşırı eğitimin önüne geçmek için döngü sayısı (epoch) sınırlandırılır. Bu çalışmada, üç farklı döngü sayısı (100, 500 ve 1000) çeşitli ön denemelerden sonra belirlenmiş ve bu değerler kullanılarak eğitilen ağların performansı değerlendirilmiştir.

❖ **En iyi döngü sayısı (EDS)**

Yapay sinir ağlarında durdurma kriterleri döngü sayısı ile sınırlı değildir. Örneğin gradyan istenilen düzeye ulaştığında da yine ağın eğitimi durmaktadır. Bu gibi döngü sayısı dışındaki parametrelere bağlı olarak eğitimin durması durumunda ağın kaç döngü yaptığı yine tez kapsamında kayıt altına alınmış ve bu açıdan da eğitim algoritmalarının etkinliği irdelenmiştir.

❖ **Gradyan (GD)**

Yapay sinir ağlarında hatalar Şekil 3.25'deki gibi bir hata uzayında dağılım göstermektedir. Bu hataların en azını belirlemek için bir gradyan değeri ile eğimin tersi yönde türev alınarak eğim azaltılır ve minimum hata değeri bulunmaya çalışılır. Bu çalışmada, yapılan ön denemeler sonucunda etkili olabileceği düşünülen üç farklı minimum gradyan değeri (10^{-7} , 10^{-6} ve 10^{-5}) kullanılarak ağların performansına etkisi incelenmiştir.

❖ **Katman sayısı (KS)**

Çok katmanlı algılayıcı modelinde girdi ve çıktı katmanının yanı sıra gizli katman(lar) eklenerek doğrusal olmayan veri yapılarının öğrenilmesi hedeflenmektedir. XOR problemi için verilen örnekte anlatıldığı gibi veriyi

sınıflandırabilmek için tek bir doğru yeterli olmadığından bir ara katman eklenerek çözüm üretilmişti. Katman sayısına bu husus dikkate alınarak verinin yapısına göre karar verilebilir. Tez çalışmasında katman sayısının artırılmasının ağ performansına etkisi de incelenmek istediğinden üç farklı katman sayısı (1-2-3) denenmiş ve elde edilen sonuçlar araştırma bulgularında verilmiştir.

❖ Nöron sayısı (NS)

YSA'da her bir katmanda nöronlar (sinir hücreleri) bulunmaktadır. Girdi ve çıktı katmanlarındaki nöronların sayısı bu katmanlardaki değişken sayısına eşittir (Ataseven, 2013). Gizli katmanda kaç nöron bulunması gerektiği ile ilgili herhangi bir kural bulunmamaktadır. Bu da bir dizi deneme yapmayı gerektirmektedir. Çalışmada yükseklik ve alan girdi değişkeni, ağırlık ise çıktı değişkeni olarak kullanıldığından girdi katmanında iki, çıktı katmanında bir nöron bulunmaktadır. Gizli katman nöron sayısının kaç olacağı bilinmediğinden çeşitli denemeler yapılması gerekebilir. Tez çalışmasında yapılan bir ön deneme sonucu ağ performansı için uygun olacağı düşünülen gizli katmanda üç, beş ve sekiz nöron bulunması durumları denenmiş, nöron sayısının hatayı azaltmadaki etkinliği incelenerek bulgular kısmında verilmiştir.

❖ Müsaade edilen en çok başarısız döngü sayısı (Max fail)

YSA'da eğitimin gereksiz yere devam etmesinin önlenmesi amacıyla uygulanan durdurma kriterlerinden birisi de doğrulama setinde hatanın artık azalmadığı ardışık döngü sayısının sınırlandırılmasıdır. MATLAB'te bu değer varsayılan olarak altı alınmıştır. Yani ağı eğitimi esnasında altı döngü boyunca doğrulama setinde hata art arda azalmazsa eğitim durdurulmaktadır. BR algoritmasında doğrulama seti olmadığından durdurma kriteri yoktur. BR algoritmasıyla yapılan denemelerde görüldüğü kadarıyla genellikle MU katsayısı belirlenen değere ulaşıncaya eğitim durmaktadır. Müsaade edilen en çok başarısız döngü sayısı, GDX ve LM algoritmalarında doğrulama seti olduğu için eğitimin durdurulmasında etkili bir kriter olmuştur. Tez çalışmasında bu değer altı olarak alınması GDX algoritmasında yetersiz eğitimle sonuçlanmış ve hatalar yüksek çıkmıştır. Birkaç denemeden sonra LM ve GDX algoritmalarının

karşılaştırılabileceği müsaade edilen en çok başarısız döngü sayısı 250 olarak belirlenmiştir. Bu değerin 250 alınmasıyla GDX algoritması en çok döngü sayısı olan 1000 döngüye kadar eğitime devam etmiştir. LM algoritmasında Şekil 3.22’de görüleceği üzere 13 döngüde en düşük hatayı yakalamasına karşın $250+13=263$ döngüye kadar eğitime devam etmiştir. Çalışma bulgularımıza göre LM algoritmasında bu değerin 10 olarak alınması önerilmektedir. Ancak GDX algoritmasında bu değeri yüksek tutmakta (döngü sayısının %25’i yeterli olabilir) fayda görülmektedir.

3.2.2.5 YSA’da kullanılan öğrenme algortimaları

YSA’da kullanılan birçok öğrenme algoritması bulunmaktadır. Bunlardan MATLAB’de kullanabilir olanlar Tablo 3.7’de verilmiştir. Daliakopoulos et al. (2005) bu algoritmaların LM, BR ve GDX algoritmalarının diğerlerinden performans bakımından daha etkili olduğunu bildirmiştir. Bu nedenle çalışmada, LM, BR ve GDX algoritmaları, hatayı azaltmadaki başarıları ve eğitimde geçen süreleri bakımından karşılaştırılmıştır.

Tablo 3.7. MATLAB’te kullanılabilir öğrenme algoritmaları.

ÖĞRENME ALGORİTMASI
Levenberg-Marquardt (LM)
Bayesian Regularization (BR)
Gradient Descent with Momentum and Adaptive Learning Rate (GDX)
BFGS Quasi-Newton (BFG)
Resilient Backpropagation (RP)
Scaled Conjugate Gradient (SCG)
Conjugate Gradient with Powell/Beale Restarts (CGB)
Fletcher-Powell Conjugate Gradient (CGF)
Polak-Ribière Conjugate Gradient (CGP)
One Step Secant (OSS)
Gradient descent with momentum (GDM)
Gradient descent (GD)

❖ Levenberg-Marquardt algoritması

LM algoritması Levenberg ve Marquardt adlı bilim insanları tarafından Newton’un optimizasyon metodu ile gradyan azaltma metodunun güçlü yönlerini bir araya getirmek için geliştirilmiştir. Newton metodunda hızlı bir şekilde lokal

veya global minimuma yakınsama sağlanırken, gradyan azaltma metodunda yavaş bir yakınsama olmasına rağmen daha az lokal minimum problemi yaşanmaktadır. LM iki yöntemi beraber ele alarak hem daha hızlı hem de lokal minimuma takılmadan bir eğitim gerçekleşmesini amaçlamaktadır (Haykin, 2009).

LM öğrenme algoritması hata kareler ortalamalarının azaltılmasında Hessian matrisini kullanır. Hessian matrisi Eşitlik 3.8’de yer almaktadır;

$$H=J^T J \quad (3.8)$$

Burada J Jacobian matrisidir ve ağ hatalarının ağırlıklara göre birinci türevi olup Eşitlik 3.9 ile hesaplanır;

$$J(n)=\frac{\delta e(n)}{\delta w(n-1)} \quad (3.9)$$

Burada;

n : iterasyon sayısı,

δ : türev sembolü,

e : ağ hataları vektörü,

w : bağlantı ağırlıklarıdır.

Bu durumda gradyan denklemi;

$$g = J^T e \quad (3.10)$$

ve bağlantı ağırlıkları;

$$w_{k+1}=w_k[J^T J+\mu I]^{-1} J^T e \quad (3.11)$$

şeklinde hesaplanır. Burada;

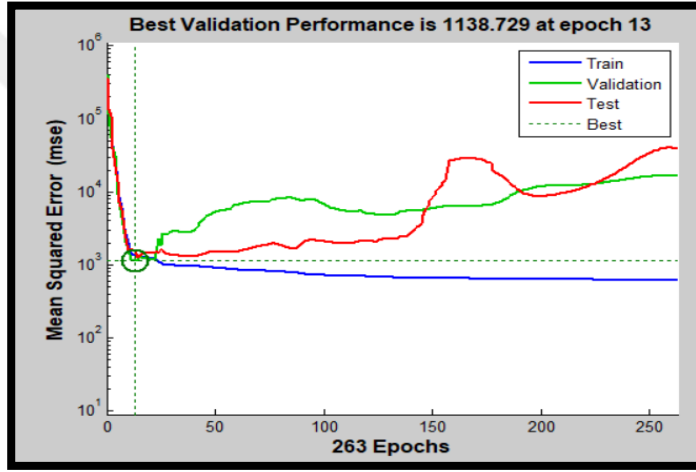
w_k : k. iterasyondaki ağırlık

I : birim matris

μ : Marquardt parametresidir.

Marquardt parametresi sıfır olduğunda Levenberg Marquardt algoritması Newton algoritması adını alır (Yetkin, 2015).

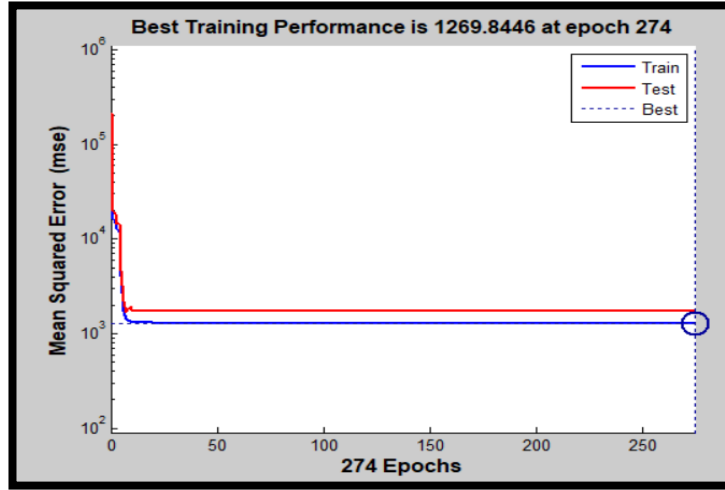
Şekil 3.22’de tez çalışması kapsamında yapılan eğitimlerden birine ait LM algoritması performans grafiği verilmiştir. LM algoritması çalışmada kullanılan diğer algoritmalarla göre daha az döngü sayısı ile eğitimi bitirmiştir. Ancak k-katlı çapraz doğrulama ile verinin ayrıldığı durumda, doğrulama (validation) setinin olmayışı nedeniyle eğitim durdurulmamış ve müsaade edilen son döngüye kadar devam etmiştir. Şekil 3.22’den anlaşılaçağı gibi bu durum LM algoritmasında yüksek hatalara neden olmaktadır.



Şekil 3.22. LM algoritması ile eğitimde performans grafiği

❖ Bayesian Düzenleme Algoritması:

Bu algoritma Levenberg-Marquardt algoritması esas alınarak geliştirilmiş bir algoritma olup (Küçükönder, 2011) amaç fonksiyonunun parametreleri için otomatik olarak optimum değerleri ayarlayan bir algoritmadır (Daliakopoulos et al., 2004).

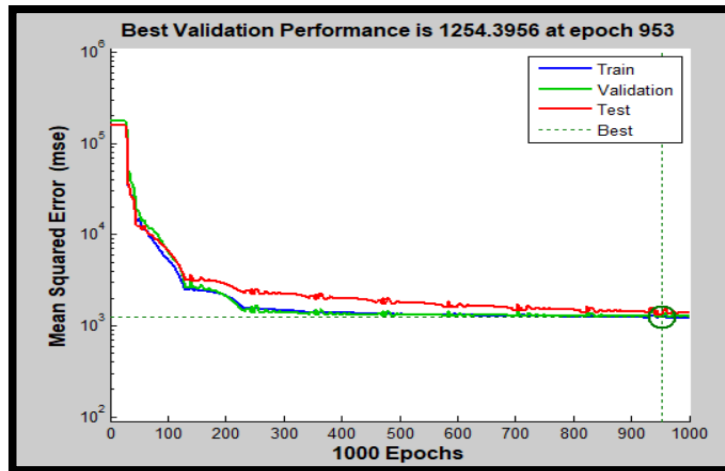


Şekil 3.23. BR algoritması ile eğitimde performans grafiği

Şekil 3.23'te tez çalışması kapsamında BR algoritmasıyla yapılan bir eğitimin performans grafiği verilmiştir. BR algoritmasıyla eğitimde önce hatalarda büyük bir azalma olmuş daha sonra 274 döngüye kadar çok az azalarak devam etmiş ve 274 döngüde eğitim sonlanmıştır.

❖ Değişken öğrenme oranlı gradyan azaltma algoritması (GDX)

Bu yöntem ağırlık ve eşik değerlerinin hesaplanmasında geri yayılım algoritmasını kullanır. Her değişken momentum ile gradyan azaltarak düzeltilir.



Şekil 3.24. GDX algoritması ile eğitimde performans grafiği

GDX algoritması, optimizasyonun her bir adımında performans düşerse öğrenme oranının artırılması ilkesine dayanır (Daliakopoulos et al., 2004). Şekil

3.24'te tez kapsamında yapılan eğitimlerden birine ait performans grafiği verilmiştir. GDX algoritmasıyla eğitimde, ağı eğitimi 1000 döngüye kadar devam etmekte ve hatalar giderek azalmaktadır. Grafikten de anlaşılacağı üzere döngü sayısının düşük tutulması GDX algoritmasında yüksek hatalarla sonuçlanabilir. Döngü sayısının artırılması durumunda hataların daha da azalıp azalmayacağı sorusu akla gelebilir. Bunu tespit etmek üzere çalışmamızda 100 bin döngüye kadar deneme yapılmasına rağmen hatalarda kayda değer bir azalma olmadığı (MAPE=%4.85) gözlemlenmiştir.

3.2.2.6 Yapay sinir ağlarında verilerin normalizasyonu

Yapay sinir ağlarında ağırlıklar başlangıçta genellikle küçük değerler olarak alındığından veri seti büyük rakamlardan oluşuyorsa döngü sayısı ve eğitim süresi uzamaktadır. Hatta bazı durumlarda ağın lokal bir minimuma takılıp hatalı sonuçlar üretmesine sebep olmaktadır. Bu problemin etkisini gidermek amacıyla verilere çeşitli normalizasyon yöntemleri uygulanmaktadır. Verilerin normalize edilmesinde genellikle aşağıda formülleri verilen lineer, min-max ve z-skör normalizasyon yöntemlerinden biri kullanılmaktadır. Bu çalışmada z-skör normalizasyonu kullanılmıştır.

❖ Lineer normalizasyon

Bir veri setindeki tüm verilerin en büyük değere (max) bölünmesi ile verilerin normalize edilmesi işlemidir. Veriler 0 ile 1 aralığında değerler alır (Arı ve Berberler, 2017). Bu normalizasyon tekniğine ilişkin matematiksel gösterim Eşitlik 3.12'de verilmiştir;

$$X_{nor_i} = \frac{x_i}{x_{max}} \quad (3.12)$$

❖ Min-Max normalizasyonu

Veri setindeki bütün verileri 0 ile 1 aralığında olacak şekilde standartlaştıran bu yöntemde minimum (min) ve maksimum (max) değerler dikkate alınarak diğer

bütün değerler bu değerlere göre normalleştirilir (Yavuz ve Deveci, 2012). Bu normalizasyon tekniğine ilişkin matematiksel gösterim Eşitlik 3.13'te verilmiştir;

$$X_{nor_i} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (3.13)$$

❖ Z-Skor normalizasyonu

Aritmetik ortalama ve standart sapma dikkate alınarak yapılan bu normalizasyon tekniğinde veriler normalleştirilerek, veriler arasındaki uzaklık ortadan kaldırılır ve verilerdeki uç noktalar azaltılır (Yavuz ve Deveci, 2012). Bu normalizasyon tekniğine ilişkin matematiksel gösterim Eşitlik 3.14'te verilmiştir;

$$X_{nor_i} = \frac{x_i - \bar{x}}{\sigma} \quad (3.14)$$

3.2.2.7 K katlı çapraz doğrulama

YSA'da veriler test, eğitim ve geçerlilik setlerine ayrılırken genellikle verilerin rastgele olarak belirli oranlarda alt setlere ayrılması tercih edilir. Bu oran MATLAB varsayılan ayarlarında %70 eğitim, %15 test ve %15 doğrulama seti şeklindedir. Veri seti rastgele olarak ayrıldığında bazı durumlarda setlere düşen gözlemler, şansa bağlı olarak uygun geldiğinden düşük hatalar ile eğitim gerçekleştirilebilmektedir. Ancak bu hatalar gelecekte yapılacak tahminler için tutarlılık göstermeyebilir. Özellikle çalışma kapsamında farklı algoritmaların hatayı azaltmadaki etkinliği de incelendiğinden bu durumun önüne geçilebilmesi adına veri seti yedi defa rastgele alt setlere ayrılarak eğitilen yedi ağın ortalama hataları dikkate alınmıştır. Bunun dışında veri setindeki her bir gözlemin hem test hem de eğitim setinde kullanıldığı çapraz doğrulama yöntemi ile de çalışma verisi alt setlere ayrılmış ve rastgele ayırma ile elde edilen hataların doğruluğu da test edilmek istenmiştir. Bu amaçla verinin yaklaşık olarak %15'i test seti %85'i eğitim seti olacak şekilde yedi parçaya ayrılmış, bir parçası test ve altı parçası eğitim verisi olacak şekilde eğitim gerçekleştirilmiş, bu yedi ağla yapılan tahminlerdeki MAPE, RMSE ve R² değerlerinin ortalamaları alınmıştır.

Gerçekte biyolojik veriler için tahmin yapılmak istendiğinde rastgele ayırma ile eğitilen yedi ağdan en düşük hata ile sonuçlanan ağ kullanarak tahmin yapmak mümkündür. Ancak çapraz doğrulamada en düşük hata ile sonuç veren ağın kullanılması doğru olmaz, çapraz doğrulama kullanılmak üzere bir ağ eğitmek içim değil, genelde makine öğrenmesi modelinin başarısını tahminlemek için kullanılmaktadır (A. Uğur, 2019, sözlü görüşme). Çapraz doğrulamanın şematik gösterimi Tablo 3.8’de verilmiştir. Tablo 3.8’de görüleceği üzere veri seti öncelikle yedi kısma ayrılır daha sonra setin ilk bölümü (K1) test seti kalan kısım (K2’den K7’ye kadar) eğitim seti olacak şekilde veri seti ayarlanır. Akabinde verinin ikinci 1/7’lik kısmı test seti kalan kısımların tamamı eğitim seti olacak şekilde ve daha sonra bütün veri hem test setinde hem de eğitim setinde kullanılacak şekilde gruplandırılarak işlem sonlandırılır.

Tablo 3.8. Çapraz doğrulama ile verinin ayrılması.

Kısım	1.Set	2.Set	3.Set	4.Set	5.Set	6.Set	7.Set
K1	T	E	E	E	E	E	E
K2	E	T	E	E	E	E	E
K3	E	E	T	E	E	E	E
K4	E	E	E	T	E	E	E
K5	E	E	E	E	T	E	E
K6	E	E	E	E	E	T	E
K7	E	E	E	E	E	E	T

E: Eğitim seti T: Test seti

Veri sıralı olarak 7 parçaya ayrılacağı gibi rastgele olarak seçilen 7 gruba da ayrılmaktadır. MATLAB’te “*crossval*” komutu ile veri seti rastgele olarak ayrılabilir. Bu çalışmada rastgele olarak 7 gruba ayırma tercih edilmiştir. Bu tercihin sebebi 244 veri olmasından dolayı test setine verinin yaklaşık olarak %15’inin dâhil edilmesidir.

3.2.3 Çok değişkenli uyarlanabilir regresyon eğrileri (MARS)

Aralarında sebep sonuç ilişkisi bulunan iki değişken arasındaki doğrusal ilişkiyi belirlemek ve bu ilişkiden yola çıkarak bağımlı değişkenin bilinmeyen bir değerini tahmin etmeye yarayan tekniğe basit doğrusal regresyon analizi

denmektedir. Bağımlı değişkenin tahmin doğruluğunun artırılması amacıyla birden çok bağımsız değişken bir arada kullanılmakta ve bu teknik çoklu doğrusal regresyon olarak adlandırılmaktadır. Bilgisayar biliminde geçtiğimiz on yıldaki gelişmeler ışığında, bu klasik regresyon yöntemlerinden ziyade karmaşık algoritmalar kullanan yeni regresyon yöntemleriyle bağımlı değişkenin değeri tahmin edilmeye çalışılmaktadır. Bu regresyon yöntemlerinden biri de yüksek boyutlu veri setleri için esnek bir modelleme sunan MARS yöntemidir (Friedman, 1995). Yöntem Chicago doğumlu Amerikan fizikçi Jerome Friedman tarafından tanıtılmıştır.

MARS, bağımlı değişkenin tahmin edilmesinde bir dizi doğrusal regresyon kullanan parametrik olmayan bir yöntemdir. Düğüm olarak adlandırılan sınırlar arasında regresyon hattının eğiminin değişmesine izin verilerek doğrusal regresyon parçaları belirlenir. Sınırlar belirlenirken doğrusal temel fonksiyonlardan yararlanılır (Bozağaç, 2014). Parçalı temel fonksiyonlar ve kombinasyonları kullanılarak regresyon yöntemlerinde kullanılan hem ileri doğru hem de geriye doğru ilerleme algoritmaları yardımıyla MARS modeli elde edilebilir (Temel vd., 2010).

MARS işlemi ileriye gidiş (forward pass), geriye dönüş (backward pass) ve yumuşatma (smoothing) olmak üzere üç aşamada gerçekleştirilir. İleriye gidiş aşamasında, bağımsız değişkenlere ait eklem fonksiyonları ve isteniyorsa değişkenler arasındaki interaksiyonları kapsayan büyük bir uzayda arama yapar. Geriye dönüşte önceki aşamada belirlenen fonksiyonlardan modelin tahmin gücüne en az düzeyde katkı yapanlar temizlenir. Son aşamada bölüm sınırlarının sürekliliğini sağlamak için bir yumuşatma işlemi uygulanır. Bir MARS modeli tahmin eşitliği genel olarak matris gösterimiyle aşağıdaki gibi yazılabilir (Cebeci, 2020).

$$y = f(X) + \varepsilon = \beta_0 + mMI = | \beta_m h_m(X) + \varepsilon \quad 3.15$$

Burada:

M: Modeldeki terimlerin sayısı

y: Bağımlı değişken vektörü (veya çok bağımlı değişken olduğunda matris olur)

X: Bağımsız değişkenler matrisi

β_0 : Kesme yüksekliği

β_m : m. terime ait regresyon katsayısı

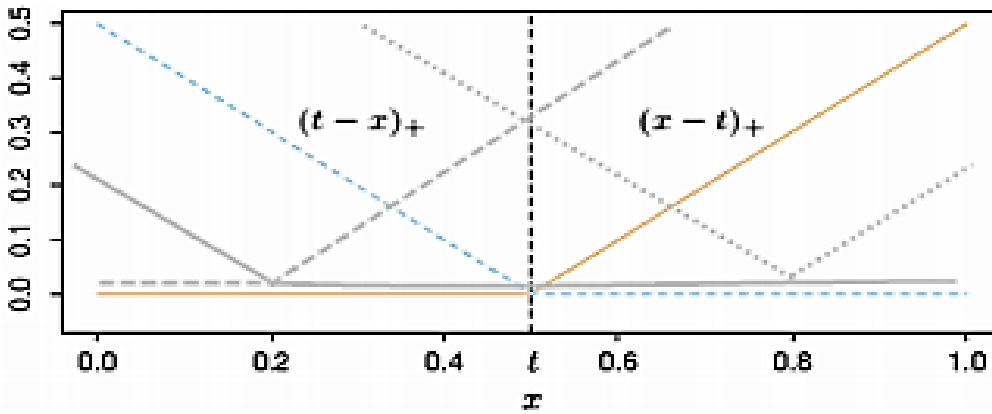
$h_m(X)$: Aday fonksiyonlar kümesi C’de m. terime ait dayanca fonksiyon veya bu fonksiyonlardan iki ve daha fazlasının çarpımı şeklinde bir interaksiyon terimidir.

MARS yönteminde bağımsız değişkenlere ait verilerin tamamı birkaç alt kümeye ayrılır ve her alt bölge için farklı matematiksel denklemler belirlenir. Bu matematiksel denklemlerle MARS, bağımsız değişkenler ve bağımlı değişkenin alt bölgeleri arasında bağlantılar oluşturur. Bu bağlantıların belirlenmesinde Eşitlik 3.16 ve 3.17’de verilen fonksiyonlardan yararlanılır (Sevimli, 2009; Köksal, 2017).

$$(x - t)_+ = \begin{cases} x - t & , \\ 0 & , \end{cases} \quad \begin{array}{l} \text{eğer } x > t \\ \text{diğer durum} \end{array} \quad (3.16)$$

$$(x - t)_- = (t - x)_+ = \begin{cases} t - x & , \\ 0 & , \end{cases} \quad \begin{array}{l} \text{eğer } t > x \\ \text{diğer durum} \end{array} \quad (3.17)$$

Bu temel fonksiyonlar bağımsız değişkenlerin gözlenen değerlerini en uygun t düğüm noktalarıyla aralıklara bölen parçalı doğrusal regresyonlardır ve bunlar Şekil 3.25’te görüldüğü gibi birbirinin yansıması olan çiftlerden oluşur. MARS modeli oluşturulurken amaç her bağımsız değişken için x_{ij} gözlem değerlerindeki düğüm noktalarında bu çifti bulmaktır (Toprak, 2015).

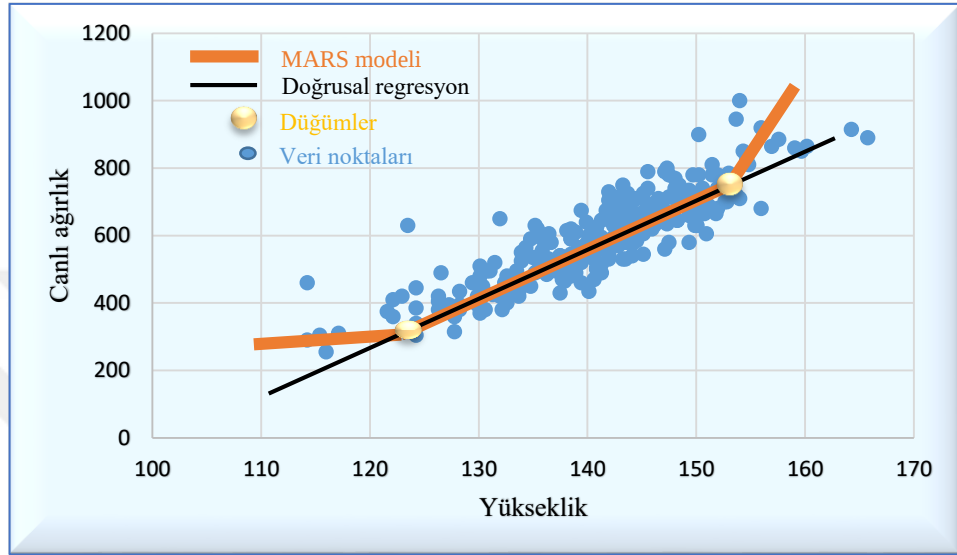


Şekil 3.25. Temel fonksiyonların gösterimi (Koc ve Bozdoğan, 2015'ten).

Yöntemin önemli avantajlarından biri parametrik testlerin gerektirdiği varsayımları gerektirmemesidir. Diğer önemli bir avantajı ise bağımsız değişkenin

bağımlı değişkenle ve bağımsız değişkenlerin birbirleri ile olan etkileşiminin modelle tanımlanması ve grafiksel bir gösterimle sunulmasıdır.

Şekil 3.26’da tez çalışmasına ait sığırlarda canlı ağırlıklar ile sığırların yükseklikleri arasındaki ilişkinin serpilme diyagramı üzerinde MARS modelinin grafiksel bir gösterimi verilmiştir.



Şekil 3.26. MARS modelinin grafiksel gösterimi

Şekil 3.26’dan sığırların yükseklikleri ile canlı ağırlıkları arasında her ne kadar doğrusal bir ilişki söz konusu olsa da doğrusal regresyona ait tahmin denklem grafiği ile 125cm’den daha düşük ve 155cm’den daha yüksek boya sahip sığırlar için yapılacak tahminde hataların yüksek olacağı görülmektedir. Çoklu doğrusal regresyon yönteminin aksine MARS yönteminde temel fonksiyonlar yardımıyla 125cm’den düşük ve 155cm’den daha yüksek sığırlar için de daha yüksek doğrulukla tahmin yapılabilmektedir. Bu da yöntemin başarısını artırmaktadır.

Yöntemde temel fonksiyonlar kullanıldığından parçalı regresyon yöntemine benzemektedir. Ancak iki yöntem arasındaki en önemli fark temel fonksiyonların MARS yönteminde otomatik olarak hesaplanmasıdır. Parçalı regresyon yönteminde MARS yöntemine benzer şekilde noktasal dağılım düğümlerle aralıklara ayrılmakta ancak bu düğümler araştırmacı tarafından ya deneme başında ya da sonunda dağılım incelenerek manuel olarak yapılmaktadır (Şahin, 2009). MARS yönteminde temel fonksiyonlar elde edilirken regresyon doğrusunun eğiminin

değiştirdiği yani bir noktadan diğerine geçildiği yer olan düğüm noktaları kullanılır. Her bir düğümün belirlenmesi farklı değişken kombinasyonlarının denenmesi ile elde edilir (Orhan vd., 2018).

Bu yöntemde hesaplamalar iki temel adımda yapılmaktadır. İlk adımda ileri doğru bütün temel fonksiyonlar (olası bütün etkileşimler) elde edilecek şekilde modelin karmaşıklığı en yüksek seviyeye ulaşmaya kadar temel fonksiyonlar eklenerek model büyür. Bu fonksiyonlar, olası bütün bağımsız değişkenlerle ve bu değişkenlerin birbirleriyle olan kombinasyonlarını kapsamaktadır. İkinci adımda bağımlı değişkenle en az ilişkili olan temel fonksiyonlar Eşitlik 3.18’de verilen genelleştirilmiş çapraz doğrulama (Generalized Cross Validation-GCV) kriteri kullanılarak modelden çıkarılır (Chou et al., 2004). Bu adıma budama adımı da denilmektedir (Temel vd., 2010).

$$LOF(\widehat{f}_M) = GCV(M) = \frac{1}{N} \sum_{i=1}^N |y_i - \widehat{f}_M(x_i)|^2 / \left[1 - \frac{c(M)}{N}\right]^2 \quad (3.18)$$

MARS yönteminin modeli Eşitlik 3.18’de verilmiştir (Özfalcı, 2008);

$$Y = \beta_0 + \sum_{k=1}^K \alpha_k \beta_k(X_t) + \epsilon_i \quad (3.19)$$

Burada;

k : Düğüm sayısı,

K : Temel fonksiyon sayısı,

X : Bağımsız değişken,

α_k : k. temel fonksiyonun katsayısı,

β_0 : Modeldeki sabit terim,

t : Düğüm değeri,

$\beta_k(X_t)$: k. Temel fonksiyondur.

Temel fonksiyona ait matematiksel gösterim Eşitlik 3.19’da verilmiştir:

$$\prod_{R=1}^{L_m} [S_{l,m}(X_{v(l,m)} - t_{l,m})] \quad (3.20)$$

Burada;

- L_m : Etkileşim derecesini,
- $S_{L,m}$: -1 veya +1 işaret değerini,
- $X_{v(l,m)}$: Bağımsız değişken değerini ve
- $t_{1,m}$: Düğüm değerini göstermektedir.

3.2.4 Çoklu doğrusal regresyon analizi

Doğrusal regresyon analizi, iki ya da daha fazla değişkenin aralarındaki doğrusal ilişkiyi belirlemek ve bağımlı değişkenin bilinmeyen bir noktadaki değerini tahmin edebilmek amacıyla geliştirilmiş bir yöntemdir. Bağımlı değişkeni etkileyen bağımsız değişkenin bir tane olması durumunda yöntem basit doğrusal regresyon, birden fazla bağımsız değişken olması durumunda çoklu doğrusal regresyon adını almaktadır.

Biyolojik olayların çoğunda bağımlı değişkeni etkileyen birden fazla faktör olması nedeniyle basit doğrusal regresyon analizi bilimsel çalışmalar için yetersiz kalmaktadır. Bu nedenle araştırmalarda genellikle çoklu doğrusal regresyon analizi tercih edilmektedir. Çoklu doğrusal regresyon analizi, aralarında ilişki bulunan bir bağımlı ve birden fazla bağımsız değişkeni kullanarak gelecekle ilgili tahminde bulunmaya yarayan bir istatistiksel yöntemdir (Takma vd., 2012). Yöntemin amacı bağımsız değişkenleri kullanarak bağımlı değişkendeki toplam varyasyonu açıklamaktır (Kayaalp, 2015).

Çoklu doğrusal regresyon analizi; bağımlı ve bağımsız değişkenler arasındaki ilişkinin matematiksel modellerle açıklanması ve değişkenler arasındaki bağıntıların bulunması, bağımlı değişkeni hangi bağımsız değişkenin daha çok etkilediğinin belirlenmesi, değişkenler arasındaki karmaşık yapının açıklanarak verinin özetlenmesi ve çoklu korelasyon katsayısı ile çoklu belirleme katsayısının hesaplanması amacıyla da kullanılır (Alpar, 2013). Yöntemin modeli Eşitlik 3.21’de verilmiştir:

$$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_j X_j + \dots + \beta_p X_p + \mathcal{E} \quad (3.21)$$

Burada Y_i bağımlı değişkenin i . gözlem değeri, $\beta_0, \beta_1, \beta_2, \dots, \beta_j, \dots, \beta_p$ ise regresyon katsayılarıdır. Bu regresyon katsayılarından herhangi bir β_j katsayısının tahmini, diğer değişkenler sabit tutulduğunda X_j değişkeninde meydana gelen bir birimlik değişime karşılık Y_i değişkeninde meydana gelecek değişim miktarını verir.

Bağımsız değişken sayısı iki olarak düşünülürse model ve parametrelerinin tahmin edilmesinde yapılan işlem basamakları aşağıdaki gibi olacaktır (Efe vd., 2000).

Öncelikle modeldeki hatanın kareler ortalamasını minimum yapmak için hata terimi yalnız bırakılır ve her iki tarafın kareleri alınacak olursa Eşitlik 3.22 elde edilir.

$$\sum (Y_i - \beta_0 - \beta_1 X_1 - \beta_2 X_2)^2 = \sum \epsilon_i^2 \quad (3.22)$$

Eşitlik 3.21'de $\beta_0, \beta_1, \beta_2$ için kısmi türev alınarak sıfıra eşitlenirse hata kareler ortalamasını minimum yapacak $\beta_0, \beta_1, \beta_2$ parametrelerinin tahminleri elde edilmiş olur.

β_0 için kısmi türev alınıp sıfıra eşitlenerek Eşitlik 3.23,

$$\sum Y = (n\hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2)^2 \quad (3.23)$$

β_1 için kısmi türev alınıp sıfıra eşitlenerek Eşitlik 3.24,

$$\sum X_1 Y = \hat{\beta}_0 \sum X_1 + \hat{\beta}_1 \sum X_1^2 + \hat{\beta}_2 \sum X_1 X_2 \quad (3.24)$$

β_2 için kısmi türev alınıp sıfıra eşitlenerek Eşitlik 3.25,

$$\sum X_2 Y = \hat{\beta}_0 \sum X_2 + \hat{\beta}_1 \sum X_1 X_2 + \hat{\beta}_2 \sum X_2^2 \quad (3.25)$$

şeklinde elde edilir.

Çoklu doğrusal regresyonda hesaplamalar matris notasyonlarında yapılmaktadır. β değerinin tahminlenmesinde genellikle en küçük kareler yöntemi tercih edilmektedir. Yöntemin modeli matris notasyonu ile Eşitlik 3.26'da verilmiştir (Çerçi 2010):

$$Y=X\beta+\mathcal{E} \quad (3.26)$$

Burada;

Y : Bağımlı değişken vektörü,

X : Bağımsız değişken matrisi,

β : Katsayılar vektörü,

$\mathcal{E}$: Hata vektörüdür.

Regresyon modeli ise Eşitlik 3.27'de verilmektedir:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & X_{12} & \dots & X_{1p} \\ 1 & X_{21} & X_{22} & \dots & X_{2p} \\ 1 & X_{31} & X_{32} & \dots & X_{3p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & \dots & X_{np} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \mathcal{E}_1 \\ \mathcal{E}_2 \\ \mathcal{E}_3 \\ \vdots \\ \mathcal{E}_p \end{bmatrix} \quad (3.27)$$

Üç bilinmeyenli bu denklemler $\beta_0, \beta_1, \beta_2$ için çözüldüğünde çoklu doğrusal regresyonda modelin parametreleri Eşitlik 3.27-3.29'daki gibi olacaktır.

$$\beta_0 = \bar{Y} - \beta_1 \bar{X}_1 - \beta_2 \bar{X}_2 \quad (3.28)$$

$$\beta_1 = \frac{s_{x_1 y} s_{x_2 x_2} - s_{x_2 y} s_{x_1 x_2}}{s_{x_1 x_1} s_{x_2 x_2} - (s_{x_1 x_2})^2} \quad (3.29)$$

$$\beta_2 = \frac{s_{x_2 y} s_{x_1 x_1} - s_{x_1 y} s_{x_1 x_2}}{s_{x_1 x_1} s_{x_2 x_2} - (s_{x_1 x_2})^2} \quad (3.30)$$

Çoklu doğrusal regresyon analizi doğrusal ve parametrik bir test olması nedeniyle bazı varsayımları gerektirir. Bunlar (Alpar, 2013; Osborne, 2003; Williams 2013);

- 1- X_j değişkeni için değerler sabittir, rastlantı değişkeni değildir.
- 2- Hata ortalamaları sıfırdır.
- 3- Değişkenler normal dağılım göstermektedir.
- 4- Değişkenler hata olmadan (güvenilir) ölçülmüş ve varyansları homojendir.
- 5- Bağımlı ve bağımsız değişkenler arasında doğrusal bir ilişki vardır.
- 6- Bağımsız değişkenler arasında yüksek derecede bir ilişki (otokorelasyon) yoktur, şeklindedir.

Bağımsız değişkenler arasında yüksek derecede bir ilişki olup olmadığının saptanması için kullanılan en yaygın yöntem varyans artırıcı faktör (Variance Inflation Factor-VIF) kullanılmasıdır. VIF değeri, her defasında bir bağımsız değişkenin bağımlı değişken, diğerlerini bağımsız değişken olarak alınmasıyla hesaplanan belirleme katsayıları yardımıyla Eşitlik 3.31 kullanılarak hesaplanır.

$$VIF(X_i) = 1/(1 - R_i^2) \quad (3.31)$$

Bağımlı kabul edilen bağımsız değişken ile bağımsız değişkenler arasında ilişki yoksa belirleme katsayısı sıfır olacağından VIF değeri 1 olacaktır. İlişki var ve örneğin $R_1^2 = 0.90$ düzeyinde ise VIF değeri Eşitlik 3.31 yardımıyla,

$$VIF(X_1) = \frac{1}{1-R_1^2} = \frac{1}{1-0.90} = 10 \quad (3.32)$$

olarak hesaplanacaktır. Eğer $VIF \geq 10$ ise anlamlı çoklu bağlantı durumu söz konusudur (Albayrak, 2012).

3.2.5 CHAID analizi

Bilimsel çalışmalarda tahmin ve sınıflandırma problemlerinin çözümünde sıklıkla tercih edilen istatistiksel yöntemlerden biri de karar ağaçlarıdır. Karar ağaçları, bağımlı değişkenin bağımsız değişkenlerle ve bağımsız değişkenlerin kendi aralarında olan ilişkilerini dikkate alan, bir dizi karar kuralları uygulamak suretiyle veri setini kendi içinde mümkün olduğunca homojen, gruplar arasında heterojen olacak şekilde küçük gruplara bölen ağaç şeklinde diyagrama sahip bir makine öğrenmesi yaklaşımıdır. Karar ağaçları yönteminde ağaç diyagramının

oluşturulması amacıyla pek çok algoritma geliştirilmiştir. Bu algoritmalara ağaç büyütme algoritmaları da denilmektedir. Bilimsel çalışmalarda kullanılan karar ağacı algoritmalarından bazıları aşağıda verilmiştir (Yılmaz, 2012; Albayrak ve Yılmaz, 2009; Hssina et al, 2014).

- ID3
- C4.5
- C5.0
- AID (Automatic Interaction Detector)
- CART (Classification and Regression Trees)
- SLIQ (Supervised Learning in Quest)
- SPRINT (Scalable Parallelizable Induction of Decision Trees)
- QUEST (Quick, Unbiased, Efficient Statistical Tree)
- MARS (Multivariate Adaptive Regression Splines)
- CHAID (Chi-Squared Automatic Interaction Detector)
- Exhaustive CHAID (Exhaustive Chi-Squared Automatic Interaction Detector)

Karar ağacı algoritmalarının çoğunda ağaç diyagramı ikili olarak dallanmaktadır. Yani veri seti tekrarlamalı olarak bağımsız değişkenle en yüksek varyansa sahip olacak şekilde iki gruba ayrılarak analize devam edilir. Tez çalışmasında kullanılan CHAID analizi yöntemi, Kass (1980) tarafından bu algoritmaların ikiden fazla dallanmama ve eksik gözlemlerden etkilenme dezavantajlarını ortadan kaldırmak amacıyla geliştirilen, bilimsel araştırmalarda en yaygın kullanılan karar ağacı algoritmalarından biridir. Bu analizde kullanılan değişkenler kesikli, sürekli veya kategorik olabilmekte ve aynı anda modele dahil edilebilmektedir. Yöntemin önemli avantajları ağaç şeklindeki diyagram çıktısı sayesinde sonuçların kolayca yorumlanabilmesi ve parametrik testlerin gerektirdiği normallik ve doğrusallık gibi varsayımların bu yöntemde olmayışdır.

Ağaç diyagramı oluşturulurken bağımlı değişkenin sürekli bir değişken olması durumunda Eşitlik 3.33'te verilen F test istatistiği kullanılır (Ölmez, 2017):

$$F = \frac{\sum_{i=1}^I \sum_{n \in D} \frac{w_n f_n I(x_n=i)(\bar{y}_i - \bar{y})^2}{I-1}}{\sum_{i=1}^I \sum_{n \in D} \frac{w_n f_n I(x_n=i)(y_n - \bar{y}_i)^2}{N_f - I}} \quad (3.33)$$

$$p = Pr(F(I-1, N_f - I) > F) \quad (3.34)$$

Burada;

D : Bir düğüm içindeki gözlem sayısını,

x_n : Bağımsız değişkenin n 'inci grubunun i . gözlem değerini,

y_n : Bağımlı değişkenin n 'inci grubunun i . gözlem değerini,

w_n : n 'inci grubun grup ağırlığı,

p : olasılık değeri,

f_n : n 'inci grubun frekansıdır. $\bar{y}_i$, $\bar{y}$ ve N_f Eşitlik 3.35-3.37'deki formüller yardımıyla hesaplanır:

$$\bar{y}_i = \frac{\sum_{n \in D} w_n f_n y_n I(x_n=i)}{\sum_{n \in D} w_n f_n I(x_n=i)} \quad (3.35)$$

$$\bar{y} = \frac{\sum_{n \in D} w_n f_n y_n}{\sum_{n \in D} w_n f_n} \quad (3.36)$$

$$N_f = \sum_{n \in D} f_n \quad (3.37)$$

Bağımlı değişkenin kesikli olması durumunda ise Eşitlik 3.38'de verilen Ki-kare istatistiği kullanılarak veri seti kendi içinde homojen, birbirileri arasında heterojen gruplara ayrılır:

$$\chi^2 = \sum \frac{(G_i - B_i)^2}{B_i} \quad (3.38)$$

Formülde;

χ^2 : Ki-kare değerini,

G_i : Gözlenen değeri,

B_i : Beklenen değeri ifade etmektedir.

Ağaç büyütme sürecinde adım adım benzer kategorilerin birleştirilmesi işlemi yapılır. Bu işlem değişkenler arasında benzerlik olmadığına istatistiksel

olarak karar verilmeye kadar devam eder. Bu karar için Bonferroni düzeltilmiş p -değerinden yararlanılır. Bonferroni düzeltmesi, birkaç bağımlı veya bağımsız istatistiksel testin eşzamanlı olarak gerçekleştirilmesi sırasında kullanılan çoklu karşılaştırma düzeltmesidir (Seong, 2009). CHAID analizi algoritmasında ağaç büyüme sürecinde aşağıda verilen sekiz adım uygulanır (Billington et al,):

1.Adım: Eğer X sadece bir kategori içeriyorsa analiz durdurulur ve düzeltilmiş p değeri 1 olacak şekilde ayarlanır.

2.Adım: Eğer X iki kategoriye sahipse 8. Adıma gidilir.

3.Adım: Değilse eğer izin verilen X kategorisi çifti bulunur, izin verilen bir kategori çifti bağımlı değişkenle en yüksek p -değerine (bağımlı değişken sürekli ise Eşitlik 3.32’de verilen denklem yardımıyla p olasılık değeri hesaplanır, bağımlı değişken kesikli ise ki kare testi için p olasılığı değeri hesaplanır) sahip böylece en az farklı yani en benzer çifttir.

4.Adım: En benzer çift için hesaplanan alfa değerinin kullanıcı tarafından belirlenen alfa değerinden (zirai çalışmalarda genellikle $\alpha=0.05$ olarak alınır) büyük olup olmadığı kontrol edilir. Büyükse bu çift tek bir bileşik kategoride birleştirilir. Ardından yeni bir X kategorileri grubu oluşturulur. Olmazsa 7. Adıma gidilir.

5.Adım: (İsteğe bağlı) Yeni oluşturulan bileşik kategori üç veya daha fazla orijinal kategoriden oluşuyorsa, p kategorisinin en küçük olduğu bileşik kategori içinde en iyi ikili bölünme bulunur. Eğer p -değeri belirli bir alfa seviyesinden daha büyük değilse ikili bölünme gerçekleştirilir.

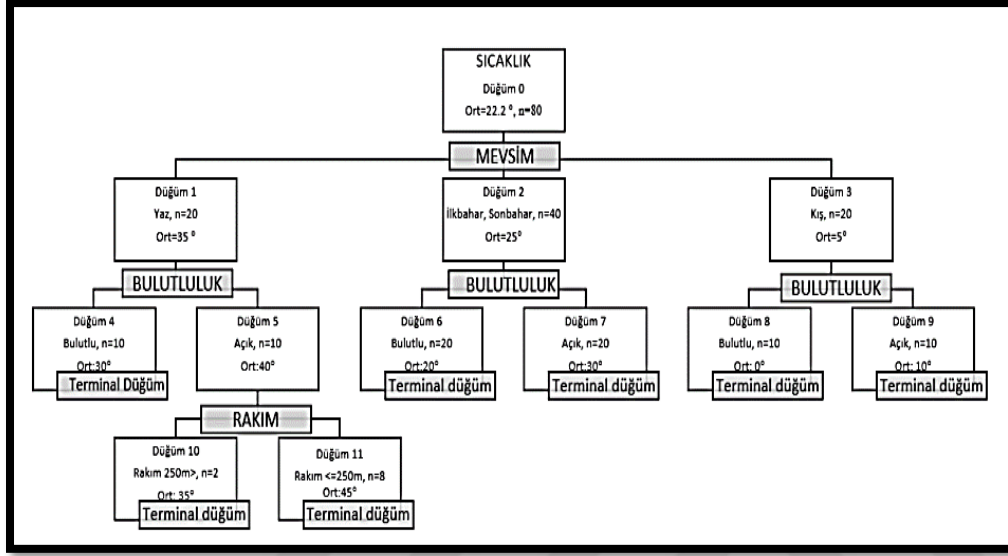
6.Adım: 2.Adıma gidilir.

7.Adım: (İsteğe bağlı) Çok az gözlem olan herhangi bir kategori (kullanıcı tarafından belirlenen minimum gözlem boyutuyla karşılaştırıldığında), p -değerlerinin en büyüğü tarafından ölçülen en benzer diğer kategoriyle birleştirilir.

8.Adım: Birleştirilen kategoriler için düzeltilmiş p -değeri Bonferroni düzeltmeleri uygulanarak hesaplanır.

Ağaç diyagramında bölünme, bağımlı değişkenle en yüksek ilişkiye sahip bağımsız değişkenin kendi içinde benzer birbiri arasında farklı gruplara ayrılması ile kök düğümden başlar. Bu aşamadan sonra bağımlı değişkenle en yüksek ilişkiye sahip ikinci bağımsız değişken tespit edilerek bölünme ve dallanma devam eder. Mümkün olan bütün bölünmeler sonlandırılıncaya ya da belirli durdurma kriterleri

tamamlanıncaya kadar (örneğin belirli bir ağaç derinliğine kadar) analiz tekrarlamalı olarak devam eder. Bölünmenin bittiği durumdaki düğümler terminal düğüm olarak adlandırılmaktadır. Verilerin analizi sonucunda kökü yukarıda olan ters bir ağaca benzeyen, sonuçların kolayca yorumlanabildiği grafiksel bir gösterimle sonuçlar özetlenir.



Şekil 3.27. CHAID analizi diyagramı örneği

Şekil 3.27’de yöntemi tanıtmaya yönelik örnek olarak, gerçek olmayan verilerden oluşturulmuş bir CHAID diyagramı çıktısı verilmiştir. Burada sıcaklık bağımlı değişkenini mevsim, bulutluluk ve rakım bağımsız değişkenlerinin etkilediği varsayılmıştır. İlkbahar, yaz, sonbahar ve kış mevsimlerinde 20’şer günde, havanın açık ve bulutlu olduğu günlerde ve rakımın 0 ile 500m arasında olduğu 10 noktadan veri alındığı kabul edilmiştir.

Bağımlı değişken sıcaklık, mevsimden sonra en çok havanın bulutluluk durumundan etkilenmiş, havanın açık olduğu durumda yaz mevsiminde 10 günün sıcaklık ortalaması 40°C’ye kadar (Düğüm 5) yükselirken bu noktada bir başka bağımsız değişken olan rakım etkili olmuştur. Rakım bağımsız değişkeni diğer iki bağımsız değişkenin aksine sürekli bir değişkendir ve 0 ile 500 arasında bütün değerleri alabilmektedir. Verilen örnekte sıcaklık bakımından rakım 250m’nin altı ve üstü olacak şekilde iki gruba ayrılmıştır. Rakımı 250m’nin üstünde olan iki bölgeye ait verilerden bu iki günün ortalaması 35°C (Düğüm 10) olarak

belirlenmiştir. Rakımı 250m'nin altında olan 8 bölgede ise sıcaklık ortalaması 45° (Düğüm 11) olarak tespit edilmiştir. Düğüm 11 aynı zamanda “rakımı 250m'nin altındaki yerleşim birimlerinde, havanın açık olduğu yaz mevsiminde yılın en yüksek sıcaklık ortalaması 45°C olarak tespit edilmiştir” şeklinde de yorumlanabilir.

Bütün mevsimlerde havanın bulutlu ve açık olması etkili bir faktör iken rakım faktörü sadece yaz mevsiminde ve havanın açık olduğu durumda etkili bulunmuş ve bu nedenle diğer gruplarda yeniden dallanma olmamıştır. Örneğimiz için ağaç derinliği üçtür. Bağımsız değişken sayısının çok ve bağımsız değişkenlerin tamamının bağımlı değişken üzerinde etkisinin önemli olduğu durumlarda ağaç derinliği de artmaktadır. Bu nedenle uygulanan CHAID analizinde kullanıcı tarafından bir durdurma kriteri olarak ağaç derinliği de belirlenebilmektedir. Dikkat edilmesi gereken nokta önemli bazı ilişkilerin gözden kaçırılmaması için ağaç derinliğinin çok düşük seçilmemesidir. Verilen örnek için ağaç derinliği iki olarak kabul edilseydi rakımın etkisi diyagramda görülemeyecekti. Bir diğer durdurma kriteri ise gruplara düşecek gözlem sayılarının sınırlandırılmasıdır. Gözlem sayılarının doğru seçilmesi de aslında önemli olan bazı ilişkilerin gözden kaçırılmaması için dikkatle seçilmelidir. Verilen örnek için gözlem sayısı ebeveyn düğümlerde (Düğüm 1, 2, 3 ve 5) beş gözlem ve çocuk düğümlerde (Düğüm 4-11) iki gözlem olacak şekilde bölünmeye müsaade edilmiştir. Çocuk düğümlerde en az üç gözlem bulunması istenseydi Düğüm 10'daki gözlem sayısı iki olduğundan bölünme gerçekleşmeyecek, rakımın etkisi yine göz ardı edilmiş olacaktı.

Ağaç büyütme işlemi tamamlandıktan sonra karar ağaçlarında budama süreci devreye girmektedir. Birçok karar ağacı algoritmasında CHAID analizinde olduğu gibi çocuk ve ebeveyn düğümlerdeki gözlem sayıları ile ağaç derinliği değerinin belirli bir standardı bulunmamaktadır. SPSS'te bu değerler varsayılan olarak ebeveyn düğüm için 100 ve çocuk düğüm için 50 olarak seçilmiştir. Varsayılan ağaç derinliği ise CHAID için üç, sınıflandırma ve regresyon ağacı için beştir. Araştırmacı bu noktada konusundaki uzmanlığını da dikkate alarak çeşitli değerler vermek suretiyle ve deneme yanılma yoluyla uygun ağaç yapısına ulaşabilir. Bu da CHAID analizinin dezavantajlarından biridir.

3.2.6 Modellerin performans değerlendirme ölçütleri

İstatistiksel yöntemlerin karşılaştırılmasında modellerin doğruluğunu test etmek için kullanılan en yaygın performans değerlendirme ölçütlerinden bazıları Ortalama Mutlak Hata Yüzdesi (MAPE), Ortalama Mutlak Hata (MAE), Belirleme Katsayısı (R^2) ve Hata Kareler Ortalamasının Karekökü (RMSE) kriterleridir. Çalışmada bu performans değerlendirme ölçütlerinden MAPE, RMSE ve R^2 değerleri kullanılmış, Tablo 3.9'da bu performans değerlendirme ölçütleri ve formülleri verilmiştir.

Tablo 3.9. Modellerin doğruluğunu sınamak için kullanılan performans değerlendirme ölçütleri ve formülleri

PERFORMANS DEĞERLENDİRME ÖLÇÜTLERİ	EŞİTLİK
MAPE	$MAPE = \frac{1}{n} \sum_{i=1}^n \left \frac{Y_i - \hat{Y}_i}{Y_i} \right \times 100$
RMSE	$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}}$
R^2	$R^2 = \left[\frac{\sum Y_i \hat{Y}_i - \frac{(\sum Y_i)(\sum \hat{Y}_i)}{n}}{\sqrt{(\sum Y_i^2 - \frac{(\sum Y_i)^2}{n})(\sum \hat{Y}_i^2 - \frac{(\sum \hat{Y}_i)^2}{n})}} \right]^2$

* Y_i : Değişkenin i. gözlem değeri, $\hat{Y}_i$: Değişkenin i. tahmin değerini göstermektedir.

4. BULGULAR

Bu çalışmada, et sığırlarında canlı ağırlığı tahmin etmek amacıyla üç boyutlu kamera ile görüntülenen sığırların görüntü matrisleri çeşitli görüntü işleme yöntemlerine tabi tutularak, sığırların alan ve yükseklik değerleri elde edilmiştir. Daha sonra bu veriler kullanılarak makine öğrenmesi yaklaşımlarından yapay sinir ağları (YSA), çok değişkenli uyarlanabilir regresyon eğrileri (MARS), çoklu doğrusal regresyon (ÇDR) ve Ki-kare otomatik interaksyon belirleme (CHAID) analizleri yardımıyla sığırların canlı ağırlıkları tahmin edilmeye çalışılmıştır.

Çalışmanın bu kısmında, yapay sinir ağlarında tahmin hatasını en aza indirmek için uygun ağ topolojisinin oluşturulması amacıyla öncelikle üç farklı katman sayısı (KS) (1, 2, 3) ve her katman için üç farklı nöron sayısının (NS) (3, 5, 8) etkisi incelenmiştir. Ayrıca yapay sinir ağlarında kullanılan öğrenme algoritmalarından Levenberg-Marquardt (Levenberg-Martquardt-LM), Uyarlanabilir Öğrenme Oranlı Gradyan (Gradient Descent with Momentum and Adaptive Learning Rate-GDX) ve Bayes Düzenlemesi (Bayesian Regularization - BR) algoritmaları kullanılarak bu algoritmaların tahmin hatasını azaltmadaki etkinliği incelenmiştir. Verilerin eğitim ve test setlerine ayrılmasında veriler k-katlı çapraz doğrulama ile yaklaşık olarak %85 eğitim ve %15 test setine ve rastgele olarak %70 eğitim %15 test %15 doğrulama setine ayrılarak eğitim denemeleri yapılmıştır. Böylece üç farklı algoritma, üç katman sayısı ve üç nöron sayısı ile dokuz parametre kombinasyonundan oluşan, iki ayrı veri ayırma durumu için ve verilerin normalize edilip edilmeme durumları olmak üzere ($3 \times 3 \times 3 \times 9 \times 2 \times 2 = 972$) 972 farklı ağ denemesi yapılmış ve elde edilen bulgular CHAID analizi ile özetlenerek EK-6'da verilmiştir.

Alan, yükseklik ve ağırlıklara ait verilerin normallik varsayımı Kolmogorov-Simironov test istatistiği ($p > 0.01$) ile kontrol edilmiş ve tüm değişkenlerin normal dağıldığı tespit edilmiştir. Yapay sinir ağlarında verilerdeki uç değerlerin etkisini azaltmak ve eğitimi daha verimli hale getirmek amacıyla çeşitli normalizasyon yöntemlerinden yararlanılmaktadır. Bu çalışmada verilere z-skor normalizasyonu uygulanmış, ham veriler ve normalizasyon yapılmış veriler kullanılarak yapılan eğitimler, tahmin hatasını azaltma bakımından karşılaştırılmıştır. Bunun dışında yapılan ön çalışma sonucunda momentum güncelleme katsayısı (Momentum

Update-MU), öğrenme oranı (Learning Rate-LR), minimum gradyan (GD) ve döngü sayıları (Iteration Number) parametreleri için Tablo 4.1’de verilen ve hatayı azaltmada etkisi olduğu düşünülen bazı parametre değerleri de kullanılarak eğitim denemeleri yapılmıştır. Bu amaçla biri standart olmak üzere (VP) sekiz farklı parametre grubu oluşturulmuş ve ağı eğitiminde kullanılmıştır. Tüm bu durumlar için ayrı ayrı elde edilen tablolar çalışmanın bulgular kısmında ve bütün analiz sonuçlarına ait özetleri gösteren iki ayrı tablo EK 2 ve EK 3’te verilmiştir.

4.1 Yapay Sinir Ağları Bulguları

YSA’da canlı ağırlığın tahmin edilmesinde veriler test ve eğitim setlerine ayrılırken iki farklı strateji izlenmiştir. İlkinde veriler çapraz doğrulama (Cross Validation-CV) ile yedi gruba ayrılmış ve her defasında bu gruptan biri test diğer altısı eğitim seti olacak şekilde yedi farklı ağ eğitilmiş ve MAPE, RMSE ve R^2 değerlerinin ortalaması alınmıştır. İkinci stratejide ise veriler %70 eğitim, %15 doğrulama ve %15 test seti olacak şekilde rastgele ayrılmış ve bu şekilde eğitilen 7 ağı yine ortalama MAPE, RMSE ve R^2 değerleri kaydedilmiştir.

4.1.1 Çapraz doğrulamada ham verilerle deneme

Bu bölümde sığırların görüntülerinden görüntü işleme sonucu elde edilen ham veriler çapraz doğrulama ile test ve eğitim setine ayrılarak EK 1a’daki MATLAB komutları yardımıyla yapay sinir ağlarında eğitimler yapılmıştır.

Tablo 4.1. Etkisi incelenen dokuz farklı parametre grubu

	Parametreler			
	MU	λ	GD	Döngü
VP	0.001	0.01	10^{-7}	1000
1	0.005	0.01	10^{-7}	1000
2	0.010	0.01	10^{-7}	1000
3	0.001	0.05	10^{-7}	1000
4	0.001	0.10	10^{-7}	1000
5	0.001	0.01	10^{-6}	1000
6	0.001	0.01	10^{-5}	1000
7	0.001	0.01	10^{-7}	500
8	0.001	0.01	10^{-7}	100

Yapılan ön eğitim denemelerinde eğitim parametrelerinden bazılarının değerlerinde yapılan beş, on ve yüz katlık artışların hatalarda değişime neden olduğu gözlemlenmiş, bu nedenle veri ayırma yönteminin yanı sıra yapay sinir ağlarında kullanılan eğitim parametrelerinden dördünün (MU, λ , GD ve Döngü sayısı) etkisini incelemek amacıyla MATLAB'ın varsayılan parametre değerlerine ek olarak çeşitli kombinasyonlarda oluşturulmuş, Tablo 4.1'de verilen sekiz farklı parametre grubu ile de denemeler yapılmış ve elde edilen bulgular (Tablo 4.2-4.38) okuyucuya sunulmuştur.

4.1.1.1 MATLAB varsayılan parametre değerleriyle canlı ağırlık tahmini

Tablo 4.1'de kalın rakamlarla verilen parametre grubu, MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre (VP) değerleridir.

MATLAB varsayılan parametreleriyle yapılan analiz sonucunda Tablo 4.2'deki bulgulara ulaşılmıştır. Tablo 4.2 incelendiğinde en düşük MAPE değerinin (%4.95) LM algoritması kullanıldığında elde edildiği, BR (%4.99) ve GDX (%5.1) algoritmalarında bu değer daha yüksek olduğu görülmektedir. Ancak tüm katman ve nöron sayılarında yapılan hataların ortalaması dikkate alındığında LM algoritması kullanıldığında daha yüksek MAPE değeri (%13.6) ile tahmin yapıldığı görülmektedir. Bunun nedeni veriler çapraz doğrulama ile test ve eğitim setine ayrıldığında doğrulama setinin bulunmamasıdır.

Tablo 4.2. Varsayılan parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	7.83	108	0.7347	5.07	37.48	0.9252	5.24	37.73	0.9194
	5	4.95	36.74	0.9277	5.09	37.75	0.9246	5.27	39.12	0.9145
	8	12.86	199.94	0.7772	5.27	38.67	0.9241	5.35	39.30	0.9184
2	3	5.16	38.04	0.9241	5.37	42.15	0.9143	5.10	37.47	0.9285
	5	6.14	54.57	0.8365	4.99	37.74	0.9241	5.38	38.77	0.9167
	8	13.77	206.16	0.4549	5.15	38.27	0.9203	6.05	44.63	0.8998
3	3	6.80	77.95	0.7936	5.11	37.57	0.9130	5.06	37.01	0.9272
	5	11.2	250.52	0.6313	7.58	53.41	0.7903	6.61	47.76	0.8707
	8	53.68	852.29	0.1340	7.02	49.83	0.7986	5.61	39.28	0.9197
Ortalama		13.60	202.69	0.6904	5.63	41.43	0.8927	5.52	40.12	0.9128

* Kalın rakamlarla gösterilen değerler her bir algoritmada elde edilen en düşük MAPE değerleridir.

Tablo 4.3'te verilen En iyi Döngü Sayıları (EDS) ve süreler incelendiğinde, normalde ortalama 25 döngü ile eğitimi sonlandıran LM algoritmasının doğrulama setinin yokluğunda 1000 döngüye kadar iterasyon yaptığı görülmektedir. Yani LM algoritmasında doğrulama seti olmadığında aşırı eğitim söz konusu olmakta ve ağ verilen örnekleri ezberlemektedir. Buna bağlı olarak eğitim setinde düşük hatalar olurken test setlerinde hata ortalaması çok yüksek (en yüksek %53.68) olmaktadır.

Ayrıca Tablo 4.3 incelendiğinde bir katmanda beş nöron bulunduğu durumda LM algoritmasında en iyi döngü sayısının 180 olduğu, bu durumda LM algoritması ile en düşük MAPE değerinin (%4.95) elde edildiği (Tablo 4.2) görülmektedir. Bu durum LM algoritmasıyla diğer katman ve nöron sayıları ile yapılan denemelerden elde edilen yüksek MAPE değerlerinin aşırı eğitimden kaynaklandığı tezini doğrulamaktadır. Diğer iki algoritma doğrulama setinin olmayışından bir miktar etkilense de LM algoritması kadar etkilenmemiştir.

Öte yandan BR algoritmasında katman ve nöron sayısını artırmak yüksek hatalara neden olmuştur. Üç gizli katman ve beş nöron (%7.58) ile sekiz nöron (%7.02) kullanıldığı durumda MAPE değeri yükselmiştir. GDX algoritması da yine nöron sayısının yüksek olmasından kısmen etkilenmiştir.

Tablo 4.3. Ağın eğitiminde geçen süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	7.7750	1000	2.4610	377	0.7700	1000
	5	1.4430	180	1.8490	237	0.7400	1000
	8	7.3900	1000	1.1810	126	0.9030	1000
2	3	7.7900	1000	1.6400	142	0.7860	1000
	5	8.9960	1000	3.8270	371	0.8330	1000
	8	9.5640	1000	10.3590	1000	0.8180	1000
3	3	9.2170	1000	3.5090	313	0.8650	1000
	5	10.7530	1000	2.2520	189	0.8810	1000
	8	12.5770	1000	16.5290	1000	0.9180	998
Ort.		8.3894	909	4.8452	417	0.8349	1000

Eğitim esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları (EDS) Tablo 4.3'te verilmiştir. Tablo 4.3 incelendiğinde GDX algoritmasının ortalama 1000 döngü ile eğitimi bitirmesine karşın en kısa (0.77sn) zamanda eğitimi tamamlayan algoritma olduğu görülmektedir. Bu parametre grubunda BR algoritması iki ve üç katmanda sekiz nöronun olduğu ağ yapısında 10

saniyenin üzerinde bir zamanda eğitimi 1000 döngü yaparak bitirmiştir. LM algoritmasında ise ortalama bir eğitimin 2.5sn'de sonlandırıldığı dikkate alındığında bu parametre grubu ve veri ayırma yöntemi için eğitimde geçen sürelerin ortalaması da (8.3894sn) yükselmiştir.

4.1.1.2 Birinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece MU değerinin 0.005 olarak alınmasıyla elde edilmiş ve 7 farklı ağ bu parametrelerle eğitilmiştir. Elde edilen bulgular Tablo 4.4'te verilmiştir. Tablo 4.4 incelendiğinde en düşük MAPE değeri (%5) ile tahminin yapıldığı BR algoritmasında bir önceki parametre grubunda olduğu gibi katman ve nöron sayısının artması (3 katman 8 nöron) daha yüksek MAPE değerine (%7.76) neden olmuştur. LM algoritması ile yine doğrulama setinin olmayışı nedeniyle özellikle iki ile üç gizli katman olması durumunda ve 8 nöron kullanıldığında yüksek MAPE değeri ile (%21.27 ve %29.16) tahmin yapılmıştır.

Tablo 4.4. Birinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.12	37.99	0.9255	5.00	37.23	0.9208	5.35	39.10	0.9183
	5	5.47	44.32	0.8996	5.00	36.81	0.9230	5.28	38.62	0.9198
	8	6.12	68.33	0.7701	5.12	38.19	0.9226	5.01	36.78	0.9259
2	3	5.37	39.62	0.9206	5.12	38.03	0.9247	5.31	39.43	0.9121
	5	5.90	52.03	0.8560	5.14	38.20	0.9238	5.11	37.41	0.9270
	8	21.27	354.45	0.5095	5.16	38.55	0.9192	5.28	38.25	0.9161
3	3	6.37	89.11	0.6548	5.10	37.91	0.9214	5.18	38.32	0.9211
	5	11.51	175.98	0.5697	5.16	38.83	0.9194	5.36	39.17	0.9197
	8	29.16	368.02	0.2111	7.76	54.71	0.7959	5.33	38.31	0.9209
Ortalama		10.70	136.65	0.7019	5.40	39.83	0.9079	5.25	38.38	0.9201

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.005, 0.01, 10^{-7} ve 1000 olarak alınmıştır.

GDX algoritması bu parametre grubu için en istikrarlı sonuçları üretmiş, doğrulama setinin olmayışından, katman ve nöron sayılarından çok etkilenmemiştir. Bu gruptaki parametrelerle yapılan eğitim esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.5'te verilmiştir. Tablo 4.5 incelendiğinde üç katman, her katmanda beş ve sekiz nöron olduğu durumda,

yapılan işlem sayısı arttığı için tüm algoritmalarla daha uzun zamanda eğitimin tamamlandığı görülmektedir. GDX algoritması yine ortalama olarak 1 sn'nin altında bir zamanda eğitimi tamamlamıştır.

Tablo 4.5. Birinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.4290	63	0.8130	100	0.8080	996
	5	7.2900	1000	1.3880	192	0.9410	999
	8	9.5560	1000	1.0720	141	0.8650	995
2	3	8.8120	1000	1.0010	127	0.7750	1000
	5	9.5320	1000	2.1980	251	0.8310	997
	8	10.4600	1000	1.0270	72	0.8930	994
3	3	11.0420	1000	2.3720	181	0.9060	998
	5	12.5760	1000	4.4520	265	1.0120	1000
	8	14.4350	1000	4.6160	280	1.0600	1000
Ort.		9.3480	896	2.1043	179	0.8990	998

4.1.1.3 İkinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece MU değerinin 0.01 olarak alınmasıyla elde edilmiş ve 7 farklı ağ bu parametrelerle eğitilmiştir. Eğitim sonunda elde edilen performans ölçütü değerleri kaydedilerek Tablo 4.6'da verilmiştir. Tablo 4.6 incelendiğinde bu parametre grubu için en düşük MAPE değeri (%4.96) ile tahmin üreten algoritmanın GDX algoritması olduğu görülmektedir.

Tablo 4.6. İkinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.16	37.78	0.9223	5.17	38.58	0.9275	4.96	37.50	0.9234
	5	5.23	39.42	0.9187	5.18	38.41	0.9232	5.32	39.12	0.9168
	8	5.76	47.22	0.8811	5.12	37.72	0.9237	5.57	40.99	0.9124
2	3	5.12	38.05	0.9229	5.13	37.87	0.9227	5.06	37.30	0.9253
	5	11.19	233.49	0.6144	5.06	37.74	0.9253	5.51	39.32	0.9192
	8	25.79	614.80	0.4516	5.11	38.43	0.9209	5.51	40.57	0.9130
3	3	5.30	39.37	0.9138	5.29	39.19	0.9222	5.49	39.63	0.9166
	5	10.86	208.88	0.6548	5.09	37.18	0.9221	5.86	41.74	0.9052
	8	49.41	806.99	0.1487	5.10	37.49	0.9194	5.43	38.42	0.9255
Ortalama		13.76	229.56	0.7143	5.14	38.07	0.9230	5.41	39.40	0.9175

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.01, 0.01, 10^{-7} ve 1000 olarak alınmıştır.

Eğitim esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.7’de verilmiştir.

Tablo 4.7. İkinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.5700	79	0.7390	108	0.7860	999
	5	6.8710	1000	1.4290	141	0.8500	1000
	8	7.7920	1000	2.2910	254	0.7870	1000
2	3	8.0540	1000	6.1840	723	0.7710	1000
	5	8.9550	1000	2.6850	278	0.9520	1000
	8	9.1770	1000	1.1240	107	0.8650	995
3	3	8.8390	1000	5.2610	547	0.8240	1000
	5	9.9810	1000	10.9880	1000	0.9020	1000
	8	12.2170	1000	6.8020	484	0.9330	996
Ort.		8.0507	898	4.1670	405	0.8522	999

Tablo 4.7 incelendiğinde süreler bakımından önceki iki parametre grubundan çok farklı sonuçlar oluşmadığı görülmektedir. GDX algoritmasıyla diğer denemelerde olduğu gibi ortalama olarak 1sn’nin altında eğitimler tamamlanmakta, LM algoritmasında 1000 döngüye kadar iterasyon yapılmakta ve eğitimde geçen süreler uzamaktadır. BR algoritmasında ise yine katman sayısının fazla olması 1000 döngüye kadar eğitimin devam etmesine neden olmaktadır.

4.1.1.4 Üçüncü grup parametrelerle eğitim

Tablo 4.8. Üçüncü grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.21	38.53	0.9234	5.03	37.50	0.9272	5.29	38.37	0.9220
	5	5.22	40.97	0.9206	5.14	38.09	0.9291	5.22	38.24	0.9268
	8	12.72	370.77	0.6565	5.06	37.39	0.9261	5.2	38.05	0.9184
2	3	5.46	39.98	0.9191	5.26	38.17	0.9240	5.15	38.14	0.9245
	5	6.85	64.58	0.8112	5.14	38.05	0.9279	5.53	40.48	0.9090
	8	52.18	707.58	0.4053	5.05	37.70	0.9271	5.24	37.89	0.9235
3	3	5.77	57.54	0.8711	5.11	38.36	0.9188	5.42	40.40	0.9159
	5	17.09	252.29	0.5841	5.07	37.19	0.9266	7.59	53.64	0.8247
	8	32.33	528.24	0.1624	7.49	50.99	0.7918	8.21	55.10	0.8330
Ortalama		15.87	233.39	0.6949	5.37	39.27	0.9110	5.87	42.26	0.8998

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.05, 10^{-7} ve 1000 olarak alınmıştır.

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece λ değerinin 0.05 olarak alınmasıyla elde edilmiş ve yedi farklı ağ bu parametrelerle eğitilerek elde edilen performans ölçütü değerleri Tablo 4.8’de verilmiştir.

Tablo 4.8 incelendiğinde bu parametre grubunda en düşük MAPE değeri BR algoritmasında, bir gizli katman ve 3 nöronun kullanıldığı durumda %5.03 olarak tespit edilmiştir. LM algoritması yine doğrulama setinin olmayışı nedeniyle yüksek hatalarla (%52.18) tahminler üretmiştir. GDX algoritması katman ve nöron sayısından etkilenmiş, 3 katmanda 5 ve 8 nöron kullanıldığı durumda yüksek hatalar (%8.21) oluşmuştur.

Eğitim esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.9’da verilmiştir. Tablo 4.9 incelendiğinde bu parametre grubunda da yine en kısa sürede (0.7820sn) GDX algoritması ile eğitim yapıldığı görülmektedir.

Tablo 4.9. Üçüncü grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	6.1500	1000	1.2000	158	0.7820	993
	5	7.1410	1000	0.9810	109	0.8050	994
	8	8.7110	1000	1.8320	199	0.8440	998
2	3	9.5870	1000	10.6390	910	0.8840	998
	5	9.8940	1000	5.4090	492	0.8990	1000
	8	10.8280	1000	6.4210	540	0.8260	1000
3	3	10.3400	1000	3.4410	283	0.9670	995
	5	11.8840	1000	2.1580	174	1.0450	998
	8	13.6640	1000	14.6560	914	0.9160	1000
Ort.		9.7999	1000	5.1930	420	0.8853	997

4.1.1.5 Dördüncü grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece λ değerinin 0.1 olarak alınmasıyla elde edilmiş ve yedi farklı ağ bu parametrelerle eğitilmiştir. Eğitim sonunda performans ölçütü değerleri kaydedilerek Tablo 4.10’da verilmiştir. Tablo 4.10 incelendiğinde bu parametre

grubunda en düşük MAPE değeri, BR algoritmasında iki gizli katman ve beş nöronun kullanıldığı durumda %5.08 olarak tespit edildiği görülmektedir.

GDX algoritması katman ve nöron sayısından etkilenmiş, üç katmanda sekiz nöron kullanıldığı durumda yüksek MAPE değeri (%6.09) elde edilmiştir. Ayrıca GDX algoritmasında tahmin değerleri ile gerçek gözlem değerleri arasındaki belirleme katsayısı ($R^2=0.9272$) bu parametre grubunda diğer algoritmalarından kısmen daha yüksek bulunmuştur.

Tablo 4.10. Dördüncü grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.20	37.79	0.9255	5.09	37.42	0.9238	5.14	37.90	0.9234
	5	5.53	45.27	0.8778	5.10	37.86	0.9265	5.38	39.77	0.9218
	8	5.61	43.79	0.8975	5.13	37.81	0.9219	5.68	40.44	0.9131
2	3	7.78	52.94	0.7863	5.24	38.81	0.9223	5.30	38.20	0.9218
	5	6.85	68.96	0.7550	5.08	37.78	0.9252	5.50	39.82	0.9137
	8	16.79	300.30	0.4407	5.14	38.01	0.9233	5.14	37.87	0.9272
3	3	12.31	239.06	0.5920	5.26	39.03	0.9210	5.97	43.54	0.8939
	5	8.22	99.27	0.6896	5.12	38.27	0.9195	5.37	40.09	0.9059
	8	38.60	570.73	0.1269	6.82	48.15	0.7925	6.09	44.19	0.8915
Ortalama		11.88	162.01	0.6768	5.33	39.24	0.9084	5.51	40.20	0.9125

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.1, 10^{-7} ve 1000 olarak alınmıştır.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.11’de verilmiştir.

Tablo 4.11. Dördüncü grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.7490	119	0.9340	118	0.7870	1000
	5	7.1660	1000	3.2880	436	2.0920	998
	8	7.2910	1000	1.5720	215	0.9020	985
2	3	0.0630	6	1.5200	164	0.8340	1000
	5	8.6930	1000	5.2340	551	0.9340	1000
	8	9.0710	1000	1.7900	180	0.9500	998
3	3	9.0600	1000	1.8750	196	0.8180	1000
	5	9.9560	1000	2.0590	204	0.9200	998
	8	12.5210	1000	5.3140	361	0.8870	1000
Ort.		7.1744	792	2.6207	269	1.0138	998

Tablo 4.11 incelendiğinde bu parametre grubunda BR algoritmasında hiçbir katman ve nöron sayısında 1000 döngüye kadar iterasyon yapılmadığı görülmektedir. Öğrenme oranının büyütülmesi BR algoritmasında döngü sayılarını sınırlamış olabilir.

4.1.1.6 Beşinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece GD değerinin 10^{-6} olarak alınmasıyla elde edilmiş ve yedi farklı ağ bu parametrelerle eğitilmiştir. Eğitim sonunda performans ölçütü değerleri kaydedilerek Tablo 4.12’de verilmiştir.

Tablo 4.12. Beşinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.18	38.57	0.9290	5.18	38.40	0.9256	5.29	38.66	0.9250
	5	5.38	38.81	0.9241	5.10	37.95	0.9234	5.37	38.58	0.9254
	8	5.46	40.59	0.9230	5.16	38.34	0.9246	5.28	39.58	0.9141
2	3	6.45	94.47	0.8289	5.12	37.59	0.9284	5.42	38.99	0.9140
	5	12.28	264.59	0.7401	5.22	38.36	0.9183	5.32	39.12	0.9218
	8	17.23	319.67	0.4989	5.14	38.23	0.9178	5.43	39.13	0.9292
3	3	6.01	75.56	0.8301	5.13	38.43	0.9191	5.33	39.83	0.9165
	5	10.14	92.48	0.6996	5.21	38.26	0.9257	5.69	42.22	0.9040
	8	44.32	571.13	0.1686	7.39	53.49	0.7936	5.50	39.62	0.9196
Ortalama		12.49	170.65	0.7269	5.41	39.89	0.9085	5.40	39.53	0.9188

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-6} ve 1000 olarak alınmıştır.

Tablo 4.12 incelendiğinde bu parametre grubunda en düşük MAPE değerinin BR algoritması, bir gizli katman ve beş nöronun kullanıldığı durumda %5.10 olarak tespit edildiği görülmektedir. LM algoritması yine doğrulama setinin olmayışı nedeniyle yüksek MAPE değerleri (%44.32) ile tahminler üretmiştir. BR algoritması üç katman ve sekiz nöronun olduğu yapıda yüksek MAPE değeri ile (%7.39) tahmin üretmiştir.

Ağ eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.13’de verilmiştir. Tablo 4.13 incelendiğinde eğitimde geçen süreler

bakımından önceki parametre gruplarından önemli düzeyde farklı sonuçlar elde edilmediği görülmüştür

Tablo 4.13. Beşinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	5.9490	1000	0.9550	136	0.7650	998
	5	6.5030	1000	0.5010	65	0.7930	1000
	8	7.2750	1000	0.9860	99	0.8210	1000
2	3	7.7260	1000	2.2430	270	0.8170	1000
	5	8.8670	1000	4.9500	528	0.9100	1000
	8	9.0240	1000	9.1080	912	0.8670	1000
3	3	9.2570	1000	2.0740	189	0.9030	1000
	5	10.0680	1000	10.0440	754	1.0910	1000
	8	12.2780	1000	3.6340	210	0.9310	1000
Ort.		8.5497	1000	3.8328	351	0.8776	1000

4.1.1.7 Altıncı grup parametrelerle eğitim

Bu gruptaki parametreler sadece GD değerinin 10^{-5} olarak alınmasıyla elde edilmiş ve yedi farklı ağ bu parametrelerle eğitilmiştir. Eğitim sonunda ortalama performans ölçütü değerleri kaydedilerek Tablo 4.14’de verilmiştir.

Tablo 4.14. Altıncı grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.21	38.53	0.9234	5.10	37.71	0.9224	5.29	38.38	0.9204
	5	5.22	40.97	0.9206	5.14	38.23	0.9263	5.31	37.93	0.9222
	8	12.72	370.77	0.6565	5.08	37.76	0.9260	5.05	36.99	0.9263
2	3	5.27	38.56	0.9191	5.07	37.84	0.9290	5.18	38.09	0.9191
	5	6.89	75.97	0.7726	5.20	38.63	0.9205	5.16	39.06	0.9180
	8	14.16	159.52	0.5069	5.15	38.20	0.9254	5.14	37.52	0.9260
3	3	5.55	44.53	0.9049	5.30	39.95	0.9086	5.38	39.09	0.9151
	5	14.24	351.05	0.5103	5.15	38.19	0.9209	5.42	40.58	0.9157
	8	33.19	511.31	0.1261	6.80	50.20	0.7845	5.11	37.22	0.9266
Ortalama		11.38	181.25	0.6934	5.33	39.63	0.9071	5.23	38.32	0.9210

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-5} ve 1000 olarak alınmıştır.

Tablo 4.14 incelendiğinde bu parametre grubunda en düşük hata (MAPE) GDX algoritması, bir gizli katman ve sekiz nöronun kullanıldığı durumda %5.05

olarak tespit edildiği görülmüştür. Bu parametre grubunda LM algoritması yine doğrulama setinin olmayışı nedeniyle yüksek MAPE değerleriyle (%33.19) tahminler üretmiştir. BR algoritması 3 katman ve 8 nöronun olduğu yapıda yüksek MAPE değeri ile (%6.8) tahmin üretmiştir.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.15'te verilmiştir. Tablo 4.15 incelendiğinde LM algoritmasının yine gereksiz yere 1000 döngüye kadar eğitime devam ettiği, GDX algoritmasının 1sn'nin altında eğitimi tamamladığı görülmektedir. BR algoritmasının bu parametre grubunda üç katman ve sekiz nöron kullanılan yapıda lokal minimuma takıldığı söylenebilir. Zira ortalama 250 döngü ile eğitimi tamamlayan algoritma 13 döngü ile eğitimi tamamladığı, Tablo 4.14 incelendiğinde bu grupta MAPE değerinin de ortalamanın bir miktar üzerinde olduğu görülebilmektedir.

Tablo 4.15. Altıncı grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	7.6500	1000	3.5560	480	0.7730	996
	5	11.2860	1000	3.0530	237	0.9460	998
	8	10.1990	1000	2.9000	327	0.8340	1000
2	3	10.7330	1000	2.6530	230	0.9290	1000
	5	9.1390	1000	1.6510	168	0.8070	998
	8	13.2500	1000	2.4490	203	0.9030	996
3	3	10.6160	1000	2.9880	270	0.9410	1000
	5	13.8880	1000	1.7410	145	0.8860	1000
	8	12.8310	1000	0.3820	13	1.1030	995
Ort.		11.0658	1000	2.3748	230	0.9024	998

4.1.1.8 Yedinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece döngü sayısının 500 olarak alınmasıyla elde edilmiş ve yedi farklı ağ bu parametrelerle eğitilmiştir. Eğitim sonunda elde edilen performans ölçütü değerleri Tablo 4.16'da verilmiştir. Tablo 4.16 incelendiğinde bu parametre grubunda en düşük MAPE değeri BR algoritması, iki gizli katman ve sekiz nöronun kullanıldığı durumda %5.04 olarak tespit edildiği görülmüştür.

Tablo 4.16. Yedinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.12	38.57	0.9193	5.12	38.04	0.9215	5.23	38.09	0.9222
	5	5.27	38.78	0.9143	5.12	37.95	0.9197	5.52	40.36	0.9066
	8	5.47	39.16	0.9191	5.21	38.88	0.9288	5.20	38.38	0.9177
2	3	5.35	38.93	0.9140	5.06	37.49	0.9268	5.23	37.79	0.9218
	5	6.63	60.40	0.8314	5.15	37.85	0.9289	5.58	44.41	0.9006
	8	10.48	116.34	0.5784	5.04	37.57	0.9220	5.49	39.91	0.9150
3	3	9.48	69.21	0.7102	5.05	37.02	0.9231	5.89	44.65	0.8933
	5	6.00	44.30	0.8954	5.09	38.02	0.9226	5.83	43.04	0.9012
	8	25.76	300.09	0.2922	5.14	38.04	0.9247	5.46	41.43	0.9165
Ortalama		8.84	82.86	0.7749	5.11	37.87	0.9242	5.49	40.90	0.9105

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-7} ve 500 olarak alınmıştır.

LM algoritması yine doğrulama setinin olmayışı nedeniyle yüksek MAPE değeri (%25.76) ile tahminler üretmiş ancak döngü sayısının azalması MAPE değerini kısmen düşürmüştür. BR ve GDX algoritması döngü sayısı 500'e düşürüldüğünde katman ve nöron sayısından etkilenmemiştir.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.17'de verilmiştir. Döngü sayısı 500 ile sınırlandırıldığında LM ve GDX algoritması maksimum döngü sayısına kadar eğitime devam ettiği, eğitimde geçen sürelerin iki algoritmada da yarı yarıya düşüş gösterdiği Tablo 4.17'den anlaşılmaktadır. BR algoritmasında sürelerde kayda değer bir azalma olmamıştır.

Tablo 4.17. Yedinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	3.0240	500	0.5960	83	0.4860	494
	5	3.2410	500	1.8860	240	0.4020	497
	8	3.9070	500	0.7360	97	0.4390	500
2	3	3.9630	500	1.4490	175	0.4010	495
	5	4.2580	500	2.1260	203	0.5020	497
	8	4.6970	500	5.0670	500	0.4390	494
3	3	0.0630	3	1.9890	207	0.4490	500
	5	4.9100	500	5.4940	500	0.5080	500
	8	5.8310	500	4.2110	212	0.4790	494
Ort.		3.7660	445	2.6171	246	0.4561	497

4.1.1.9 Sekizinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece döngü sayısının 100 olarak alınmasıyla elde edilmiş ve yedi farklı ağ bu parametrelerle eğitilmiştir. Eğitim sonunda performans ölçütü değerleri kaydedilerek Tablo 4.18’de verilmiştir.

Tablo 4.18. Sekizinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.22	37.38	0.9276	5.18	38.39	0.9255	6.42	47.85	0.8823
	5	4.85	35.19	0.9291	5.14	37.47	0.9287	6.51	47.44	0.8789
	8	6.47	60.77	0.8010	5.13	37.66	0.9236	7.21	49.69	0.8574
2	3	5.13	37.56	0.9273	5.07	37.95	0.9266	7.59	58.02	0.8161
	5	5.75	52.00	0.8835	5.04	37.10	0.9259	6.51	46.62	0.8877
	8	8.11	106.16	0.7422	5.14	38.26	0.9229	7.08	50.04	0.8633
3	3	5.05	36.56	0.9306	5.03	37.02	0.9219	12.97	88.95	0.7098
	5	5.67	44.50	0.8966	5.14	37.67	0.9266	8.13	58.61	0.8183
	8	7.04	57.93	0.8406	7.18	52.55	0.7902	6.81	49.39	0.8706
Ortalama		5.92	52.01	0.8754	5.34	39.34	0.9102	7.69	55.18	0.8427

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-7} ve 100 olarak alınmıştır.

Tablo 4.18 incelendiğinde, LM algoritmasında, varsayılan dahil dokuz parametre grubundan en düşük MAPE değerinin (%4.85) bu gruptaki parametrelerle elde edildiği görülmektedir. Ancak GDX algoritmasında ise tam tersi bir durum söz konusu olmakta, döngü sayısının azaltılması MAPE ve RMSE değerlerini artırmaktadır.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.19’de verilmiştir. Tablo 4.19 incelendiğinde döngü sayısının 100 ile sınırlandırılmasıyla bütün algoritmaların 1sn’nin altında eğitimi tamamladığı görülmektedir.

Tablo 4.19. Sekizinci grup parametrelerle eğitimde süre ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.6480	100	0.7330	100	0.0970	100
	5	0.7020	100	0.7410	100	0.2890	100
	8	0.7180	100	0.7650	100	0.1010	100
2	3	0.7630	100	0.8550	100	0.1010	100
	5	0.0575	100	0.9180	100	0.1010	100
	8	0.8160	100	1.0340	100	0.1320	100
3	3	0.8860	100	0.8930	100	0.1220	100
	5	0.9710	100	1.0330	100	0.1160	100
	8	1.1870	100	1.3190	100	0.1540	100
Ort.		0.7498	100	0.9212	100	0.1348	100

4.1.2 Çapraz doğrulamada normalize edilmiş verilerle deneme

Çalışmanın bu kısmında sığırların görüntülerinden çeşitli görüntü işleme yöntemleriyle elde edilen ham veriler z-skorlarına dönüştürülerek verilere standartlaştırma uygulanmış, çapraz doğrulama ile test ve eğitim setine ayrılan veriler ve EK 1b'deki komutlar yardımıyla yapay sinir ağlarında eğitim yapılmıştır. Aşağıda dokuz farklı parametre grubu için yapılan analiz sonuçları verilmiştir.

4.1.2.1 MATLAB varsayılan parametre değerleriyle canlı ağırlık tahmini

Bu gruptaki parametreler Tablo 4.20'de görülen ve MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerleridir.

Tablo 4.20. MATLAB varsayılan parametreleri

MATLAB varsayılan parametreleri			
MU	λ	GD	Döngü
0.001	0.01	10^{-7}	1000

Yapılan analiz sonucunda Tablo 4.21'deki bulgulara ulaşılmıştır. Bulgular incelendiğinde BR algoritması (MAPE=%5.07) dışında standartlaştırmanın yapılmadığı durumdaki eğitimden hataların büyük farklılık (standartlaştırmanın olmadığı durumda %4.95) göstermediği anlaşılmaktadır.

Tablo 4.21. Varsayılan parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.25	39.90	0.9153	5.07	37.62	0.9254	5.10	37.41	0.9262
	5	13.19	443	0.7725	5.08	37.90	0.9168	5.24	38.21	0.9229
	8	7.71	5794	0.8058	5.07	37.65	0.9268	5.54	41.11	0.9113
2	3	7.48	57.31	0.8366	5.10	37.89	0.9274	5.28	39.05	0.9194
	5	12.95	237	0.7063	5.12	38.10	0.9224	5.22	37.97	0.9200
	8	15.66	252	0.4451	5.11	38.02	0.9256	5.27	39.36	0.9279
3	3	5.92	58.00	0.8055	5.10	37.88	0.9251	5.36	38.99	0.9193
	5	17.50	467	0.5531	5.16	38.33	0.9241	5.62	39.80	0.9189
	8	37.97	505	0.2322	5.16	67.59	0.6602	5.19	36.77	0.9324
Ortalama		77.07	872.58	0.6747	5.11	41.22	0.8949	5.31	38.74	0.9220

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.22’de verilmiştir. Tablo 4.22 incelendiğinde LM algoritması bir katmanda üç nöronun bulunduğu durum hariç eğitimin yine 1000 döngüye kadar devam ettiği görülmektedir.

Tablo 4.22. Varsayılan parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
KS	NS	LM		BR		GDX	
		Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.5360	75	0.9480	121	0.7390	1000
	5	7.5830	1000	1.5040	192	0.7650	1000
	8	7.4040	1000	1.6040	165	0.9120	1000
2	3	8.6210	1000	0.7550	90	0.8660	995
	5	9.8370	1000	2.0140	198	0.8460	1000
	8	10.9790	1000	1.3720	136	0.9850	998
3	3	10.2410	1000	9.2940	1000	0.8980	998
	5	10.0950	1000	2.8780	253	1.5230	1000
	8	14.7920	1000	12.3590	360	0.9270	1000
Ort.		8.8987	897	3.6364	279	0.9401	999

BR algoritmasında üç katman ve üç nöronun olduğu yapıda 1000 döngüye kadar eğitim devam etmiş ancak diğer durumlarda aşırı eğitim gözlenmemiştir. GDX algoritmasında daha önceki parametre gruplarından farklı sonuçlar elde edilmemiştir.

4.1.2.2 Birinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece MU değerinin 0.005 olarak alınmasıyla elde edilmiştir.

Tablo 4.23. Birinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.33	45.98	0.8982	5.06	37.70	0.9271	5.14	38.32	0.9130
	5	5.19	36.71	0.9274	5.15	38.23	0.9226	5.25	40.71	0.9227
	8	5.57	42.37	0.9043	5.09	37.39	0.9239	5.52	40.99	0.9075
2	3	5.31	39.15	0.9188	5.16	38.18	0.9219	5.29	38.32	0.9257
	5	12.39	266	0.5261	5.19	37.81	0.9244	5.49	39.56	0.9123
	8	20.28	299	0.3151	5.05	37.33	0.9257	5.37	38.73	0.9198
3	3	5.33	38.05	0.9222	5.10	37.43	0.9245	5.52	38.79	0.9251
	5	16.28	297	0.5036	5.07	37.92	0.9279	5.08	37.28	0.9294
	8	39.37	577	0.2042	7.53	51.02	0.7900	5.38	39.33	0.9178
Ortalama		12.78	182.36	0.6800	5.38	39.22	0.9098	5.34	39.11	0.9193

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.005, 0.01, 10^{-7} ve 1000 olarak alınmıştır.

Yedi farklı ağ bu parametrelerle eğitilmiştir. Eğitim sonunda performans ölçütü değerleri kaydedilerek Tablo 4.23'te verilmiştir. Tablo 4.23 incelendiğinde bu parametre grubunda en düşük MAPE değeri BR algoritması, iki gizli katman ve sekiz nöronun kullanıldığı durumda %5.04 olarak tespit edildiği görülmektedir. LM algoritması yine doğrulama setinin olmayışı nedeniyle yüksek MAPE değerleri (%39.37) ile tahminler üretmiş BR algoritması üç nöron ve sekiz katmanın olduğu durumda yüksek MAPE değeri (%7.53) ile tahmin üretmiştir.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.24'te verilmiştir. Tablo 4.24 incelendiğinde BR algoritmasının üç defa 1000 döngüye kadar eğitime devam ettiği görülmektedir. GDX ve LM algoritmalarında ise yine önceki parametre gruplarıyla çok farklı sonuçlar elde edilmemiştir.

Tablo 4.24. Birinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	6.0900	1000	5.6140	676	0.7500	1000
	5	7.5090	1000	1.4540	129	0.7330	1000
	8	7.5970	1000	3.2450	309	0.7420	1000
2	3	8.4590	1000	8.6160	1000	0.9140	1000
	5	8.7950	1000	1.7680	187	0.7910	994
	8	9.7740	1000	4.0480	370	0.8540	1000
3	3	9.5340	1000	9.4410	1000	0.8120	1000
	5	11.2520	1000	5.9630	549	0.8600	1000
	8	13.8610	1000	14.2480	1000	0.8930	1000
Ort.		9.2079	1000	6.0441	580	0.8166	999

4.1.2.3 İkinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece MU değerinin 0.01 olarak alınmasıyla elde edilmiş ve yedi farklı ağ bu parametrelerle eğitilmiştir. Bu parametre grubunda en düşük MAPE değerinin BR algoritması, bir gizli katman ve üç nöronun kullanıldığı durumda %5.03 olarak tespit edildiği Tablo 4.25'ten görülmektedir.

Tablo 4.25. İkinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.14	38.37	0.9241	5.03	37.50	0.9272	5.29	38.36	0.9221
	5	5.76	59.63	0.8786	5.14	38.09	0.9291	5.22	38.25	0.9266
	8	5.12	36.52	0.9252	5.06	37.37	0.9262	5.19	37.99	0.9186
2	3	5.94	53.13	0.8684	5.26	38.17	0.9240	5.15	38.23	0.9240
	5	49.50	1509	0.6460	5.12	38.16	0.9272	5.56	40.41	0.9092
	8	29.90	454	0.5641	5.05	37.70	0.9271	5.09	37.59	0.9236
3	3	6.07	63.39	0.8035	5.24	39.60	0.9134	5.34	39.68	0.9198
	5	16.53	308	0.4872	5.1	37.21	0.9263	5.26	38.80	0.9218
	8	26.33	313	0.2226	5.42	42.58	0.9197	5.52	39.39	0.9109
Ortalama		16.69	315	0.7021	5.16	38.49	0.9245	5.29	38.74	0.9196

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.01, 0.01, 10^{-7} ve 1000 olarak alınmıştır.

Ağ eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.26'de verilmiştir. Tablo 4.26 incelendiğinde bir katmanda üç nöron kullanıldığında LM algoritmasında aşırı eğitim oluşmadığı, BR algoritmasında üç

katman ve sekiz nöron bulunan yapıda aşırı eğitim olduğu ve GDX algoritmasında diğer parametre gruplarına göre bir farklılık olmadığı görülmektedir.

Tablo 4.26. İkinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.3010	40	1.2230	161	0.7510	996
	5	6.9870	1000	0.9030	115	0.7490	1000
	8	7.1970	1000	1.3590	169	0.7800	1000
2	3	7.6860	1000	7.9760	827	0.7790	1000
	5	9.5590	1000	6.4800	491	0.8070	1000
	8	10.1540	1000	5.9200	557	0.8440	1000
3	3	10.2800	1000	6.5610	676	0.8560	1000
	5	9.9330	1000	2.2780	181	0.8560	997
	8	13.8450	1000	15.6540	1000	0.8860	1000
Ort.		8.4380	893	5.3727	464	0.8120	999

4.1.2.4 Üçüncü grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece MU değerinin 0.05 olarak alınmasıyla elde edilmiştir. Eğitim sonunda performans ölçütü değerleri kaydedilerek Tablo 4.27’de verilmiştir. Bu parametre grubunda en düşük MAPE değeri BR algoritması, üç gizli katman ve üç nöronun kullanıldığı durumda %4.98 olarak tespit edilmiştir.

Tablo 4.27. Üçüncü grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.47	46.02	0.8987	5.13	37.37	0.9210	5.11	37.14	0.9309
	5	5.91	73.09	0.8598	5.07	37.40	0.9247	5.17	37.88	0.9276
	8	5.71	44.87	0.8678	5.10	37.56	0.9270	5.45	39.51	0.9147
2	3	6.92	109.75	0.7636	5.19	39.28	0.9234	5.07	36.89	0.9253
	5	21.52	770.92	0.6738	5.05	37.63	0.9246	5.44	38.23	0.9175
	8	30.81	511.95	0.2872	5.12	37.92	0.9264	5.34	38.39	0.9182
3	3	6.41	71.45	0.7958	4.98	37.21	0.9276	5.21	38.19	0.9251
	5	12.73	227.79	0.5780	5.10	37.62	0.9236	5.26	38.80	0.9218
	8	39.70	548.48	0.1493	5.11	37.86	0.9257	5.36	38.61	0.9197
Ortalama		15.02	267.15	0.6527	5.09	37.76	0.9249	5.27	38.18	0.9223

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.05, 0.01, 10^{-7} ve 1000 olarak alınmıştır.

LM algoritmasında yüksek MAPE değerleri (%39.7) ile tahminler üretilmiştir. GDX ve BR algoritmalarının LM algoritmasının aksine katman ve nöron sayılarından etkilenmediği Tablo 4.27’den görülmektedir.

Ağın eğitimi esnasında geçen süreler ve en iyi döngü sayıları Tablo 4.28’de verilmiştir. Tablo 4.28 incelendiğinde bir katmanda üç ve sekiz nöron bulunduğu durumlar hariç LM algoritmasında 1000 döngüye kadar eğitimin devam ettiği görülmektedir. BR algoritmasında bir katmanda beş nöron bulunan yapıda 1000 döngüye kadar eğitime devam edilmiştir. GDX algoritması için diğer parametre gruplarıyla bir farklılık oluşmamıştır.

Tablo 4.28. Üçüncü grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.6720	99	2.7440	424	0.7650	989
	5	1.7100	204	7.4110	1000	0.7400	1000
	8	10.5510	1000	0.9180	113	0.7560	994
2	3	7.8050	1000	2.3120	236	0.7820	997
	5	11.84	1000	3.1740	371	0.8180	1000
	8	8.9080	1000	3.4250	355	0.8870	996
3	3	14.1970	1000	3.5570	349	0.8710	991
	5	9.9020	1000	1.5190	116	0.8560	997
	8	11.5460	1000	1.4500	95	0.8870	1000
Ort.		8.5701	811	2.9456	340	0.8180	996

4.1.2.5 Dördüncü grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece λ değerinin 0.1 olarak alınmasıyla elde edilmiştir. Eğitim sonunda performans ölçütü değerleri kaydedilerek Tablo 4.29’de verilmiştir. Tablo 4.29 incelendiğinde bu parametre grubunda en düşük MAPE değerinin GDX algoritması, bir gizli katman ve üç nöronun kullanıldığı durumda %4.96 olarak tespit edildiği görülmektedir. Doğrulama setinin olmayışı nedeniyle LM algoritmasıyla yüksek MAPE değerleri (%40.87) ile tahminler yapıldığı görülmektedir. GDX ve BR algoritmaları katman ve nöron sayılarından etkilenmemiştir.

Tablo 4.29. Dördüncü grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.47	46.02	0.8987	5.03	37.50	0.9272	5.27	38.11	0.9251
	5	5.91	73.09	0.8598	5.14	38.09	0.9291	5.71	40.68	0.9092
	8	5.71	44.87	0.8678	5.06	37.39	0.9261	4.96	36.79	0.9334
2	3	6.92	109.75	0.7636	5.26	38.17	0.9240	5.09	37.78	0.9235
	5	21.52	770.92	0.6738	5.14	38.05	0.9279	5.75	41.68	0.9136
	8	30.81	511.95	0.2871	5.05	37.70	0.9271	5.55	39.78	0.9192
3	3	5.29	38.97	0.9210	5.11	38.36	0.9188	5.17	37.22	0.9260
	5	14.56	274.50	0.3521	5.12	38.19	0.9239	5.50	39.91	0.9067
	8	40.87	510.57	0.1605	5.03	37.29	0.9238	5.23	37.47	0.9252
Ortalama		15.23	264.52	0.6427	5.10	37.86	0.9253	5.36	38.82	0.9202

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.1, 10^{-7} ve 1000 olarak alınmıştır.

Ağın eğitimi esnasında geçen süreler ve en iyi döngü sayıları Tablo 4.30'da verilmiştir. Tablo 4.30 incelendiğinde bir önceki grupta olduğu gibi LM algoritmasında bir katmanda üç ve beş nöronun olduğu yapı hariç 1000 döngüye kadar eğitimin sürdüğü görülmektedir. BR algoritmasında üç katmanda beş nöron bulunan yapıda ise 1000 döngü ile eğitim tamamlanmıştır.

Tablo 4.30. Dördüncü grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.6680	99	1.0830	161	0.7230	999
	5	1.4020	204	1.2410	109	0.7520	1000
	8	8.3540	1000	1.7180	201	0.7460	1000
2	3	8.0710	1000	7.9500	936	0.7680	1000
	5	8.8640	1000	4.9130	520	0.8110	1000
	8	9.1300	1000	7.8120	564	0.8190	1000
3	3	10.0580	1000	2.9920	283	0.8230	995
	5	10.1890	1000	12.3960	1000	0.8750	1000
	8	11.9460	1000	2.7320	192	1.1190	1000
Ort.		7.6313	811	4.7597	441	0.8262	999

4.1.2.6 Beşinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece gradyan değerinin 10^{-6} olarak alınmasıyla elde edilmiş ve yedi farklı ağ bu parametrelerle eğitilmiştir. Eğitim sonunda performans ölçütü değerleri kaydedilerek Tablo 4.31'de verilmiştir. Tablo 4.31 incelendiğinde bu parametre

grubunda en düşük MAPE değerinin BR algoritması, bir gizli katman ve üç nöronun kullanıldığı durumda %5.03 olarak tespit edildiği görülmektedir. LM algoritması yine doğrulama setinin olmayışı nedeniyle yüksek MAPE değeri (%40.87) ile tahminler üretmiştir.

Tablo 4.31. Beşinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.47	46.03	0.8987	5.03	37.50	0.9272	5.29	38.36	0.9221
	5	5.91	73.09	0.8598	5.14	38.09	0.9291	5.22	38.25	0.9266
	8	5.71	44.87	0.8678	5.06	37.39	0.9261	5.19	37.99	0.9186
2	3	6.92	109.75	0.7636	5.26	38.17	0.9240	5.15	38.23	0.9240
	5	21.52	770.92	0.6738	5.14	38.05	0.9279	5.56	40.41	0.9092
	8	30.81	238.50	0.2871	5.05	37.70	0.9271	5.09	37.60	0.9236
3	3	5.29	38.97	0.9210	5.11	38.36	0.9188	5.34	39.68	0.9198
	5	14.56	274.50	0.3521	5.07	37.19	0.9266	5.26	38.80	0.9218
	8	40.87	510.57	0.1605	7.50	51.05	0.7916	5.52	39.39	0.9109
Ortalama		15.23	234.13	0.6427	5.37	39.28	0.9109	5.29	38.75	0.9196

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-6} ve 1000 olarak alınmıştır.

GDX algoritması katman ve nöron sayılarından etkilenmezken BR algoritması üç katmanda sekiz nöron bulunan yapıda yüksek MAPE değeri ile (%7.5) tahmin üretmiştir.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.32’de verilmiştir.

Tablo 4.32. Beşinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
KS	NS	LM		BR		GDX	
		Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.7150	92	1.2410	161	0.7360	996
	5	1.4900	169	0.8060	109	0.7410	1000
	8	7.3650	1000	1.9780	201	0.7610	1000
2	3	8.2700	1000	7.9420	936	0.8030	1000
	5	8.8760	1000	4.9000	520	0.8180	1000
	8	9.3060	1000	5.9790	564	0.8410	1000
3	3	10.5110	1000	2.6260	283	0.9510	1000
	5	10.6230	1000	1.7700	169	0.9240	997
	8	13.9270	1000	12.7670	892	0.9120	1000
Ort.		7.8981	807	4.4454	426	0.8319	999

Tablo 4.32 incelendiğinde sonuçların diğer parametre gruplarıyla elde edilen süre ve en iyi döngü sayılarından büyük farklılık göstermediği söylenebilir. Bu parametre grubundaki tek farklılık BR algoritmasında 1000 döngüye kadar eğitimin devam ettiği her hangi bir katman ve nöron sayısı yapısının bulunmamasıdır.

4.1.2.7 Altıncı grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece gradyan değerinin 10^{-5} olarak alınmasıyla elde edilmiş ve yedi farklı ağ bu parametrelerle eğitilmiştir. Eğitim sonunda performans ölçütü değerleri kaydedilerek Tablo 4.33’de verilmiştir. Tablo 4.33 incelendiğinde bu parametre grubunda en düşük MAPE değerinin BR algoritması, bir gizli katman ve 3 nöronun kullanıldığı durumda %5.03 olarak tespit edildiği görülmektedir. LM algoritması yine doğrulama setinin olmayışı nedeniyle yüksek MAPE değeri (%40.87) ile tahminler üretmiştir.

Tablo 4.33. Altıncı grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.56	49.96	0.8874	5.03	37.50	0.9272	5.29	38.36	0.9221
	5	5.91	73.08	0.8599	5.14	38.09	0.9291	5.22	38.25	0.9266
	8	5.68	44.41	0.8709	5.06	37.39	0.9261	5.19	37.99	0.9186
2	3	6.92	109.75	0.7636	5.26	38.17	0.9240	5.15	38.23	0.9240
	5	21.52	770.89	0.6737	5.14	38.05	0.9279	5.56	40.41	0.9092
	8	30.81	511.95	0.2871	5.05	37.70	0.9271	5.09	37.59	0.9236
3	3	5.29	38.97	0.9210	5.11	38.36	0.9188	5.34	39.68	0.9198
	5	14.56	274.50	0.3521	5.07	37.19	0.9266	5.26	38.80	0.9218
	8	40.87	510.57	0.1605	7.50	51.05	0.7916	5.52	39.39	0.9109
Ortalama		15.24	264.90	0.6418	5.37	39.28	0.9109	5.29	38.74	0.9196

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-5} ve 1000 olarak alınmıştır.

Bu grupta beşinci grup parametrelerle yapılan eğitimde elde edilen MAPE değerlerine çok yakın MAPE değerleri elde edilmiştir. Minimum MAPE değerleri birebir aynıdır. Gradyan değerini artırmak, MAPE değerlerinde bir değişime neden olmamaktadır. Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.34’te verilmiştir. Tablo 4.34 incelendiğinde LM

algoritmasının bir katmanda üç farklı nöron sayısı içinde 1000 döngüye kadar eğitime devam etmediği görülmektedir. BR ve GDX algoritmalarında bir önceki parametre grubu ile aynı süre ve döngü sayıları elde edilmiştir. Gradyan değerinin artırılması aynı MAPE değerlerinin elde edilmesinin yanı sıra aynı döngü sayısı ile eğitimin tamamlanmıştır. Önceki gruptan farklı olarak sadece sürelerde çok küçük farklılıklar oluşmuştur.

Tablo 4.34. Altıncı grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.6160	86	1.1030	161	0.7650	996
	5	0.9530	145	0.9550	109	0.7710	1000
	8	2.9980	437	1.4660	201	0.7550	1000
2	3	7.7650	1000	8.2890	936	0.8170	1000
	5	8.4070	1000	4.9030	520	0.8060	1000
	8	8.9790	1000	5.8160	564	0.8300	1000
3	3	10.8750	1000	3.0270	283	0.8170	1000
	5	10.0990	1000	2.1220	169	0.8720	997
	8	12.2020	1000	12.9590	892	0.9020	1000
Ort.		6.9882	741	4.5156	426	0.8150	999

4.1.2.8 Yedinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece döngü sayısının 500 olarak alınmasıyla elde edilmiş ve yedi farklı ağ bu parametrelerle eğitilmiştir.

Tablo 4.35. Yedinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.47	46.02	0.8987	5.03	37.50	0.9272	5.43	39.57	0.9171
	5	5.66	60.18	0.8781	5.14	38.09	0.9291	5.23	38.64	0.9251
	8	5.69	44.70	0.8691	5.06	37.39	0.9261	5.37	38.92	0.9141
2	3	6.09	72.57	0.7817	5.26	38.17	0.9240	5.30	39.53	0.9183
	5	8.12	142.02	0.7290	5.14	38.05	0.9279	5.91	42.46	0.9000
	8	22.06	361.58	0.2941	5.05	37.70	0.9271	5.25	38.22	0.9207
3	3	5.16	37.86	0.9269	5.11	38.36	0.9188	5.86	43.44	0.9013
	5	12.62	238.56	0.4901	5.07	37.19	0.9266	5.29	39.53	0.9184
	8	35.37	614.35	0.0963	7.50	51.11	0.7914	5.81	41.34	0.9025
Ortalama		11.80	179.76	0.6627	5.37	39.28	0.9109	5.49	40.18	0.9131

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-7} ve 500 olarak alınmıştır.

Eğitim sonunda performans ölçütü değerleri kaydedilerek Tablo 4.35'te verilmiştir. Tablo 4.35 incelendiğinde bu parametre grubunda en düşük MAPE değerinin yine BR algoritması, bir gizli katman ve üç nöronun kullanıldığı durumda %5.03 olarak tespit edildiği görülmektedir. GDX algoritması katman ve nöron sayılarından etkilenmezken BR algoritması bir önceki grupta olduğu gibi yüksek MAPE değeri ile (%7.5) tahmin üretmiştir.

Tablo 4.36. Yedinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.6560	99	1.1720	161	0.4550	499
	5	1.5970	204	0.8370	109	0.4020	495
	8	3.9040	500	1.5060	201	0.4930	497
2	3	4.1270	500	4.4810	500	0.4230	499
	5	4.6170	500	4.5120	500	0.6120	500
	8	4.1430	500	4.8810	500	0.4740	500
3	3	4.4630	500	2.5610	283	0.4590	494
	5	4.9280	500	1.9910	169	0.4740	500
	8	5.9630	500	7.0330	500	0.4890	500
Ort.		3.8220	423	3.2193	325	0.4757	498

Ağ eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.36'da verilmiştir. Maksimum döngü sayısının azaltılmasının yine eğitim sürelerinde bir düşüşe neden olduğu Tablo 4.36'dan görülmektedir.

4.1.2.9 Sekizinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB tarafından varsayılan parametrelerden farklı olarak sadece döngü sayısının 100 olarak alınmasıyla elde edilmiş ve yedi farklı ağ bu parametrelerle eğitilmiştir. Eğitim sonunda performans ölçütü değerleri kaydedilerek Tablo 4.37'de verilmiştir.

Tablo 4.37 incelendiğinde bu parametre grubunda en düşük MAPE değerinin BR algoritması, iki gizli katman ve üç nöronun kullanıldığı durumda %4.99 olarak tespit edildiği görülmektedir. LM algoritması önceki tüm parametre gruplarının aksine döngü sayısının 100 ile sınırlandırılmasından olumlu etkilenmiştir. Bunun nedeni aşırı eğitim olmamasıdır. Bu nedenle LM algoritması daha düşük MAPE değerleri ve düşük MAPE ortalaması ile verilerin rastgele ayrıldığı durumda elde edilen MAPE değerlerine yakın tahminler üretmiştir. Ancak bu grup parametrelerle

eğitimde LM algoritması katman ve nöron sayısından etkilenmiş, üç katman ve sekiz nöronun olduğu yapıda MAPE değeri %8.37 ile tahmin üretmiştir.

Tablo 4.37. Sekizinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.14	37.79	0.9251	5.09	37.52	0.9204	6.87	51.33	0.8607
	5	5.80	50.63	0.8573	5.09	37.15	0.9290	6.82	49.62	0.8761
	8	5.89	61.54	0.8017	5.13	38.18	0.9219	5.97	43.86	0.8947
2	3	5.22	37.47	0.9240	4.99	37.09	0.9270	7.39	54.32	0.8573
	5	5.52	41.32	0.9059	5.09	38.13	0.9174	6.08	44.02	0.8995
	8	6.45	51.47	0.8478	5.08	38.15	0.9235	6.06	46.00	0.8891
3	3	5.04	36.94	0.9281	5.04	37.38	0.9306	7.85	55.75	0.8367
	5	6.54	64.00	0.8086	5.15	38.21	0.9268	6.09	44.05	0.8954
	8	8.37	87.89	0.7475	5.15	38.00	0.9243	5.79	44.13	0.9037
Ortalama		6.00	52.12	0.8607	5.09	37.76	0.9245	6.55	48.12	0.8792

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-7} ve 100 olarak alınmıştır.

GDX ve BR algoritmaları katman ve nöron sayılarından etkilenmemiştir. Döngü sayısının 100 olarak sınırlandırılmasıyla bütün algoritmalar bir saniyenin altında eğitimi tamamlamıştır.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.38’de verilmiştir.

Tablo 4.38. Sekizinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
KS	NS	LM		BR		GDX	
		Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.5420	82	0.7020	94	0.1160	100
	5	0.7180	100	0.7820	100	0.1000	100
	8	0.7390	100	0.7490	100	0.1010	99
2	3	0.9350	100	0.8210	100	0.1160	98
	5	0.8020	100	0.9870	100	0.1000	100
	8	1.0380	100	1.0100	100	0.1000	89
3	3	0.8650	100	0.9330	100	0.1240	100
	5	0.9710	100	1.0810	100	0.1540	100
	8	1.1020	100	1.3500	100	0.1310	98
Ort.		0.8569	98	0.9350	99	0.1158	98

Tablo 4.38 incelendiğinde döngü sayısının 100 ile sınırlandırılması nedeniyle süre ve en iyi döngü sayılarında doğal bir düşüş olduğu görülmektedir. Tablo

4.38'den LM algoritmasında aynı döngü sayıları söz konusu olduğunda BR ve GDX algoritmaları gibi 1sn'nin altında eğitimin tamamlandığı görülmektedir. GDX algoritması yinede sekiz kat daha hızlı eğitimi tamamlamıştır. Ancak BR ve LM algoritmaları söz konusu olduğunda iki algoritma arasında eğitim süreleri bakımından büyük bir fark olmadığı görülmektedir.

4.1.3 Çapraz doğrulama için tüm durumlara ilişkin özet

Verilerin çapraz doğrulama ile test ve eğitim setine ayrıldığı durum için MATLAB tarafından otomatik olarak verilen parametre grupları dahil olmak üzere dokuz farklı parametre grubu, normalize edilmiş ve ham veriler olarak iki farklı veri grubu için toplam 18 adet tablo elde edilmiştir. Bu tablolardan her bir algoritma için en düşük MAPE değeri ile sonuç veren durumlar ve bunlara ait en iyi döngü sayıları, eğitimde geçen süreler ile ortalama hataları gösteren özet tablo EK 2a, EK 2b ve EK 2c'de verilmiştir.

EK 2a incelendiğinde çapraz doğrulama için tüm durumlarda en düşük MAPE değeri (%4.85), LM algoritması ve standartlaştırma olmayan durumda, 8. parametre grubunda yani döngü sayısının 100 olarak sınırlandırıldığı durumda elde edilmiştir. BR ve GDX için bu değerler sırasıyla %4.98 ve %4.96 olarak elde edilmiştir. Çapraz doğrulama ile veriler ayrıldığında parametre gruplarının büyük çoğunluğunda BR algoritması ile minimum MAPE değerleri elde edilmiştir.

4.1.4 Verilerin %70 eğitim, %15 test ve %15 doğrulama setine rastgele ayrılması.

YSA'da veriler genellikle %70 eğitim, %15 test ve %15 geçerlilik setine rastgele olarak ayrılır. MATLAB'te bu veri ayırma biçimi varsayılan olarak seçilidir. Tez çalışması kapsamında çapraz doğrulama ile karşılaştırmak amacıyla bu veri ayırma şekli ile de denemeler yapılmıştır. Bu ayırma yönteminde daha önce çapraz doğrulama ile verilerin ayrıldığı durumda olduğu gibi dokuz farklı parametre grubunun etkisi incelenmiştir. Ayrıca verilerde normalizasyonun etkisinin de incelendiği tablolar ve ağırlık eğitiminde geçen süreler ile eğitimin sonlandırıldığı en iyi döngü sayılarının verildiği toplam 36 tablo elde edilerek (Tablo 4.40-Tablo 4.76) her bir parametre grubu için verilen başlıkların altında okuyucuya sunulmuştur.

4.1.5 Rastgele ayırmada ham verilerle deneme

Çalışmanın bu kısmında et sığırlarından alınan üç boyutlu görüntülerin çeşitli görüntü işleme yöntemlerine tabi tutulmasıyla elde edilen sığır alanı ve yüksekliğine ait ham veriler herhangi bir standartlaştırmaya tabi tutulmadan rastgele olarak %70 eğitim, %15 test ve %15 doğrulama setine ayrılarak EK 1c'deki komutlar yardımıyla yapay sinir ağlarında eğitim gerçekleştirilmiştir. Dokuz farklı parametre grubu için yapılan analiz sonuçları aşağıda verilmiştir.

4.1.5.1 Rastgele ayırmada MATLAB varsayılan parametreleriyle eğitim

Bu gruptaki parametreler, Tablo 4.39'daki MATLAB paket programı tarafından otomatik olarak atanan varsayılan parametre değerleridir.

Tablo 4.39. MATLAB varsayılan parametreleri

MATLAB varsayılan parametreleri			
MU	λ	GD	Döngü
0.001	0.01	10^{-7}	1000

Yapılan eğitimler sonucunda Tablo 4.40'daki bulgulara ulaşılmıştır.

Tablo 4.40. Rastgele ayırmada varsayılan parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.8	36.09	0.9321	4.96	36.87	0.9275	5.16	38.53	0.9219
	5	4.87	35.97	0.9313	4.95	36.79	0.9278	5.09	38.24	0.9231
	8	4.66	34.65	0.9372	4.99	36.92	0.9273	5.11	38.03	0.9233
2	3	4.84	38.06	0.9211	4.94	36.79	0.9280	5.13	37.53	0.9260
	5	4.80	35.93	0.9318	4.89	36.55	0.9288	5.16	37.53	0.9256
	8	4.68	35.26	0.9355	4.95	36.78	0.9279	4.88	36.06	0.9309
3	3	4.98	36.61	0.9306	4.87	36.20	0.9300	5.53	40.56	0.9123
	5	5.04	37.06	0.9269	4.77	35.76	0.9318	7.62	53.44	0.8088
	8	4.71	37.03	0.9266	7.16	50.98	0.7959	4.98	36.43	0.9299
Ortalama		4.82	36.30	0.9303	5.16	38.18	0.9139	5.41	39.59	0.9113

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

Tablo 4.40 incelendiğinde bu parametre grubunda en düşük MAPE değerinin LM algoritması, bir gizli katman ve 8 nöronun kullanıldığı durumda %4.66 olarak tespit edildiği görülmektedir. LM algoritması verilerin rastgele ayrıldığı durumda

verilerin bir kısmının doğrulama setine ayrılması nedeniyle çapraz doğrulamaya göre daha kararlı ağırlar ve daha düşük MAPE ve RMSE değerleri ile tahminler üretmiştir. BR algoritmasında en düşük MAPE değeri %4.77 ve GDX algoritmasında bu değer %4.75 olarak belirlenmiştir. Bu değer GDX algoritmasının bütün denemelerinden elde edilen en düşük MAPE değeridir.

GDX algoritması MATLAB varsayılan değerleri kullanıldığında en düşük MAPE değeri ile sonuç vermektedir.

Tablo 4.41. Rastgele ayırmada varsayılan parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	2.6150	5	0.7640	90	0.7440	923
	5	1.8890	8	7.1810	922	0.7400	992
	8	1.9190	7	1.3810	177	0.7400	999
2	3	4.3720	282	8.2360	967	0.7700	947
	5	2.5320	48	2.3600	254	0.6760	589
	8	2.3200	10	6.0790	564	0.9490	702
3	3	2.3680	15	3.3740	315	0.8650	999
	5	2.8590	42	4.4810	397	0.9020	945
	8	3.0350	13	0.2310	12	0.7330	523
Ort.		2.6566	48	3.7874	411	0.7910	847

LM (en düşük %4.56) ve BR (en düşük %4.72) algoritmaları da tüm parametre kombinasyonları denenerek elde edilen minimum MAPE değerlerine çok yakın sonuçlar üretmiştir.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.41’de verilmiştir. Tablo 4.41 incelendiğinde LM algoritmasında iki gizli katman ve üç nöron kullanıldığı durum hariç az sayıda döngüyle eğitimin tamamlandığı görülmektedir.

4.1.5.2 Rastgele ayırmada birinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece MU değerinin 0.005 olarak alınmasıyla elde edilmiştir. Yapılan analizler sonucunda Tablo 4.42’deki bulgulara ulaşılmıştır. Tablo 4.42’den bu parametre grubunda en düşük MAPE değeri yine LM algoritmasında ancak iki gizli katman ve beş nöronun kullanıldığı durumda %4.6 olarak tespit edildiği görülmektedir.

Tablo 4.42. Rastgele ayırmada birinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.85	35.41	0.9335	4.93	36.74	0.9280	4.99	37.01	0.9273
	5	4.78	35.57	0.9324	4.93	36.84	0.9276	5.35	38.62	0.9209
	8	4.76	35.39	0.9340	4.95	36.95	0.9273	5.29	38.79	0.9207
2	3	7.11	50.46	0.7978	4.90	36.39	0.9294	7.82	55.40	0.7799
	5	4.60	33.89	0.9391	4.94	36.82	0.9278	5.28	39.93	0.9107
	8	4.62	34.54	0.9368	4.85	36.01	0.9307	5.07	37.15	0.9279
3	3	6.33	46.76	0.8628	4.85	35.96	0.9309	7.71	53.60	0.7859
	5	4.62	33.91	0.9390	4.86	36.07	0.9305	5.00	36.34	0.9302
	8	4.74	36.85	0.9284	4.85	36.00	0.9308	5.87	41.74	0.9022
Ortalama		5.16	38.09	0.9115	4.90	36.42	0.9292	5.82	42.06	0.8895

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.005, 0.01, 10^{-7} ve 1000 olarak alınmıştır.

BR algoritması katman ve nöron sayılarından etkilenmezken LM ve GDX algoritması iki gizli katmanda üç nöron olduğu durumda yüksek MAPE değeri ile (%7.11, %7.82) tahmin üretmiştir. Bunun sebebi momentum güncelleme katsayısının 0.005 olarak alınması nedeniyle ağırlık lokal minimum'a takılması olarak değerlendirilebilir.

Ağırlık eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.43'de verilmiştir. Tablo 4.43 incelendiğinde LM algoritmasında üç gizli katmanda beş nöronun kullanıldığı durum hariç az sayıda döngü ile eğitimin tamamlandığı görülmektedir.

Tablo 4.43. Rastgele ayırmada birinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
KS	NS	LM		BR		GDX	
		Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	1.9370	12	1.1720	156	0.6420	613
	5	2.6070	98	1.1180	148	1.4860	376
	8	2.2100	8	2.4170	302	0.7740	924
2	3	2.1110	12	1.1800	118	0.6010	442
	5	2.2540	12	1.8060	128	0.2650	64
	8	2.5070	22	1.5210	139	0.4790	282
3	3	2.5340	13	6.6600	639	0.8710	992
	5	5.4710	269	1.6040	148	0.7640	500
	8	3.3220	6	15.4090	1000	1.0170	941
Ort.		2.7726	50	3.6541	309	0.7666	570

4.1.5.3 Rastgele ayırmada ikinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece MU değerinin 0.01 olarak alınmasıyla elde edilmiştir. Yapılan analizler sonucunda Tablo 4.44'deki bulgulara ulaşılmıştır.

Tablo 4.44. Rastgele ayırmada ikinci grup parametrelerle yapılan analiz bulguları.

KS	NS	ALGORİTMALAR								
		LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.88	36.23	0.9304	4.92	36.82	0.9279	5.17	38.21	0.9233
	5	4.75	35.47	0.9329	4.93	36.72	0.9282	5.28	38.63	0.9210
	8	4.67	34.31	0.9374	4.97	36.84	0.9276	5.22	38.06	0.9234
2	3	4.95	36.07	0.9313	4.88	36.35	0.9295	6.50	46.34	0.8577
	5	4.93	36.24	0.9309	4.92	36.77	0.9279	5.04	37.32	0.9263
	8	4.76	37.32	0.9299	4.93	36.69	0.9282	7.41	51.99	0.7990
3	3	4.80	34.75	0.9363	4.84	36.07	0.9305	5.21	38.96	0.9208
	5	4.73	34.63	0.9366	4.93	36.70	0.9282	5.29	38.09	0.9235
	8	4.86	36.11	0.9310	6.97	49.54	0.8012	5.12	37.13	0.9275
Ortalama		4.81	35.68	0.9330	5.14	38.06	0.9144	5.58	40.53	0.9025

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.01, 0.01, 10^{-7} ve 1000 olarak alınmıştır.

Bu parametre grubunda LM algoritması önceki iki MU değerinden daha istikrarlı sonuçlar üretmiş, Tablo 4.45'deki en iyi döngü sayılarında incelenecek olursa en fazla 31 döngüde eğitimi tamamlamasına karşın hiçbir denemede %5 üzerinde MAPE değeri elde edilmemiştir. BR algoritması üç katman ve sekiz nöronun kullanıldığı durumda aşırı eğitimden etkilenmiş bunun dışındaki denemelerde %5'in altında MAPE değeri ile ağırlar üretmiştir.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.45'de verilmiştir. Tablo 4.45 incelendiğinde bir önceki parametre grubunda LM algoritmasında ortalama 50 olan en iyi döngü sayısının bu parametre grubunda ortalama 16 olduğu görülmektedir. MU parametresini 10 kat büyütmek LM algoritmasında döngü sayısının azalmasına neden olmuştur.

Tablo 4.45. Rastgele ayırmada ikinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	2.0560	28	0.7190	82	0.5700	476
	5	2.3790	16	2.5150	285	0.8250	923
	8	2.2860	31	5.3030	638	0.3860	212
2	3	2.4140	13	1.8580	198	0.8030	1000
	5	2.9610	2	3.8540	363	0.8840	989
	8	3.1380	12	1.3190	111	0.9190	940
3	3	2.6620	26	2.2910	244	0.8960	995
	5	2.9420	6	1.9580	144	0.3800	103
	8	3.8180	9	14.2180	1000	0.9870	963
Ort.		2.7396	16	3.7817	341	0.7389	733

4.1.5.4 Rastgele ayırmada üçüncü grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece λ değerinin 0.05 olarak alınmasıyla elde edilmiştir. Yapılan analizler sonucunda Tablo 4.46'daki bulgulara ulaşılmıştır. Tablo 4.46 incelendiğinde bu parametre grubunda yine en düşük MAPE değerinin (%4.72) LM algoritmasıyla elde edildiği görülmektedir.

Tablo 4.46. Rastgele ayırmada üçüncü grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.94	36.26	0.9305	4.92	36.83	0.9279	5.07	37.73	0.9251
	5	4.73	35.06	0.9345	4.89	36.25	0.9299	5.28	39.18	0.9184
	8	4.72	34.80	0.9355	4.97	36.93	0.9273	5.19	37.58	0.9257
2	3	4.95	36.07	0.9313	4.85	36.09	0.9305	10.97	74.77	0.6584
	5	4.93	36.24	0.9309	4.92	36.77	0.9279	5.01	37.09	0.9273
	8	4.76	37.32	0.9299	4.92	36.62	0.9285	4.99	36.40	0.9296
3	3	4.80	34.75	0.9363	4.84	36.01	0.9307	5.09	37.13	0.9269
	5	4.73	34.63	0.9366	4.92	36.64	0.9285	5.58	41.33	0.9069
	8	4.86	36.11	0.9310	9.33	64.22	0.6671	5.59	41.30	0.9069
Ortalama		4.82	35.69	0.9329	5.40	39.60	0.8998	5.86	42.50	0.8917

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.05, 10^{-7} ve 1000 olarak alınmıştır.

LM algoritması öğrenme oranının beş kat büyütülmesinden etkilenmemiş tüm denemelerde %5'in altında MAPE değerleriyle tahminler elde edilmiştir. BR

algoritmasında üç katmanda sekiz nöronun bulunduğu durum hariç %5'in altında MAPE değerleri ile tahminler üretilmiş, GDX algoritması iki katmanda üç nöronun bulunduğu yapıda lokal minimum'a takılmış, yüksek bir MAPE değeri (%10.97) ile tahmin üretmiştir.

Ağın eğitimi esnasında geçen süreler ve en iyi döngü sayıları Tablo 4.47'de verilmiştir. Tablo 4.47 incelendiğinde tüm algoritmalarda en iyi döngü sayılarında azalma olduğu görülmektedir.

Tablo 4.47. Rastgele ayırmada üçüncü grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	1.9200	28	0.6640	81	0.5990	465
	5	1.8900	16	2.6380	268	0.3550	152
	8	2.2060	31	6.6520	765	0.9030	912
2	3	2.1250	13	0.8960	92	0.7170	639
	5	2.6250	2	4.0790	371	0.8560	916
	8	2.6300	12	10.7260	1000	0.3540	145
3	3	2.9710	26	1.9620	209	0.8870	912
	5	2.7610	6	1.7000	148	0.9090	980
	8	3.7610	9	14.4530	1000	0.9490	822
Ort.		2.5432	16	4.8633	437	0.7254	660

4.1.5.5 Rastgele ayırmada dördüncü grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece λ değerinin 0.1 olarak alınmasıyla elde edilmiştir. Yapılan analizler sonucunda Tablo 4.48'daki bulgulara ulaşılmıştır. Tablo 4.48'den görüldüğü gibi bu parametre grubunda yine en düşük MAPE değeri (%4.72) ile tahmin LM algoritmasından elde edilmiştir. LM algoritması öğrenme oranının beş kat büyütülmesinden etkilenmemiş tüm denemelerde %5'in altında MAPE değerleri ile tahminler üretmiştir. BR algoritması katman ve nöron sayısının artışından etkilenmiş üç katman ve sekiz nöron bulunan yapıda yüksek bir MAPE değeri (%9.33) ile tahmin üretmiştir.

Tablo 4.48. Rastgele ayırmada dördüncü grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.94	36.26	0.9305	4.92	36.83	0.9279	5.13	38.05	0.9240
	5	4.73	35.06	0.9345	4.89	36.25	0.9299	5.22	38.36	0.9222
	8	4.72	34.80	0.9355	4.97	36.93	0.9273	5.20	37.99	0.9235
2	3	4.95	36.07	0.9313	4.85	36.10	0.9305	5.63	43.21	0.8977
	5	4.93	36.24	0.9309	4.92	36.77	0.9279	5.09	37.67	0.9250
	8	4.76	37.32	0.9299	4.92	36.62	0.9285	7.37	51.73	0.8365
3	3	4.80	34.75	0.9363	4.84	36.01	0.9307	5.12	38.08	0.9237
	5	4.73	34.63	0.9366	4.92	36.64	0.9285	5.25	37.75	0.9249
	8	4.86	36.11	0.9310	9.33	64.22	0.6671	5.09	36.95	0.9284
Ortalama		4.82	35.69	0.9329	5.40	39.60	0.8998	5.46	39.98	0.9118

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.1, 10^{-7} ve 1000 olarak alınmıştır.

Tablo 4.49’da en iyi döngü sayıları incelendiğinde diğer durumlarda ortalama 400 döngü ile eğitimi tamamlayan BR algoritmasında üç katman ve sekiz nöron bulunan yapıda 1000 döngü ile eğitimi tamamladığı görülmektedir. Yüksek MAPE değerinin sebebi aşırı eğitim olabilir.

Tablo 4.49. Rastgele ayırmada dördüncü grup parametrelerle eğitimde süre ve iterasyon sayıları.

		ALGORİTMALAR					
KS	NS	LM		BR		GDX	
		Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	2.0530	28	0.6240	81	0.6700	591
	5	2.2280	16	2.0360	268	0.8020	959
	8	2.2180	31	5.8560	765	0.4010	221
2	3	2.1520	13	0.8020	92	0.7870	1000
	5	2.4290	2	3.6890	371	0.8020	932
	8	2.3360	12	10.1500	1000	0.8490	990
3	3	2.4660	26	2.1590	209	0.8720	995
	5	2.5380	6	1.6410	148	0.4010	105
	8	3.1140	9	14.5840	1000	0.9870	967
Ort.		2.3927	16	4.6157	437	0.7301	751

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.49’da verilmiştir. Tablo 4.49 incelendiğinde GDX algoritmasında en iyi döngü sayılarında bir düşüş olduğu ancak Tablo 4.48’de bu azalmanın MAPE değerlerinde bir yükselmeye sebebiyet verdiği görülmektedir.

4.1.5.6 Rastgele ayırmada beşinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece gradyan değerinin 10^{-6} olarak alınmasıyla elde edilmiştir.

Yapılan analiz sonucunda Tablo 4.50'deki bulgulara ulaşılmıştır. Tablo 4.50 incelendiğinde bu parametre grubunda yine en düşük MAPE değerinin (%4.56) LM algoritmasından elde edildiği görülmektedir. Bu değer tüm denemelerden elde edilen üç minimum MAPE değerinden biridir.

Tablo 4.50. Rastgele ayırmada beşinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.33	39.24	0.9191	4.93	36.76	0.9280	5.28	39.25	0.9185
	5	4.87	35.90	0.9315	4.93	36.39	0.9293	5.20	38.35	0.9223
	8	4.68	35.87	0.9311	4.96	36.86	0.9275	4.98	36.42	0.9302
2	3	4.82	35.92	0.9319	4.94	36.65	0.9283	5.06	37.06	0.9273
	5	4.78	35.05	0.9348	4.94	36.72	0.9281	7.34	52.27	0.8061
	8	4.56	34.21	0.9377	4.92	36.66	0.9285	5.18	37.53	0.9260
3	3	4.82	35.35	0.9335	4.87	36.24	0.9300	5.37	39.44	0.9171
	5	4.81	36.83	0.9290	4.83	35.94	0.9310	5.16	37.99	0.9239
	8	4.74	34.75	0.9367	4.88	36.49	0.9290	5.04	36.86	0.9277
Ortalama		4.82	35.90	0.9317	4.91	36.52	0.9289	5.40	39.46	0.9110

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-6} ve 1000 olarak alınmıştır.

*** Siyah dolgulu hücre tüm denemeler sonucunda LM için en düşük MAPE değeridir.

GDX algoritması katman ve nöron sayısının artışından olumsuz yönde etkilenmiş, iki katman ve sekiz nöron bulunan yapıda yüksek bir MAPE değeri (%7.34) ile tahmin üretmiştir.

BR algoritmasının daha önceki parametre gruplarıyla yapılan denemelerde de katman ve nöron sayılarının artmasından olumsuz etkilendiği gözlemlenmiştir. Gradyan değerinin 10 kat azaltıldığı bu parametre grubunda BR algoritmasıyla üç katman ve sekiz nöron bulunduğu yapıda da düşük MAPE değerleri elde edildiği Tablo 4.50'den görülmektedir.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.51’de verilmiştir. Tablo 4.51 incelendiğinde GDX algoritmasında üç katmanda beş ve sekiz nöron bulunan yapıda yaklaşık 100 döngüde eğitimin sona erdiği görülmektedir.

Tablo 4.51. Rastgele ayırmada beşinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	1.4040	17	1.1340	163	0.7330	999
	5	1.8250	13	3.9100	413	0.7640	967
	8	2.0570	21	2.5380	331	0.7870	932
2	3	2.8080	52	3.5040	395	0.7410	687
	5	2.2600	15	3.0390	305	0.3190	116
	8	2.3430	7	1.4350	124	0.8960	943
3	3	3.9610	169	8.7140	809	0.8570	944
	5	2.5440	10	7.6530	659	0.5080	102
	8	2.9850	6	14.9920	1000	0.5390	105
Ort.		2.4652	34	5.2132	467	0.6827	644

4.1.5.7 Rastgele ayırmada altıncı grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece gradyan değerinin 10^{-5} olarak alınmasıyla elde edilmiştir. Yapılan analizler sonucunda Tablo 4.52’deki bulgulara ulaşılmıştır.

Tablo 4.52 incelendiğinde bu parametre grubunda yine en düşük MAPE değerinin (%4.6) LM algoritmasından elde edildiği görülmektedir. LM algoritması gradyan değerinin 100 kat büyütülmesinden en iyi döngü sayıları bakımından olumlu etkilenmiş (ortalama 12 döngü) ayrıca tüm denemelerde %5’in altında MAPE değerleri ile tahminler ürettiği görülmektedir. BR algoritması gradyan değerinin büyütülmesinden pozitif yönde etkilenmiş, tüm denemelerden elde edilen minimum MAPE değeri bu parametre grubunda elde edilmiştir. Ayrıca GDX algoritmasında lokal minimum probleminin oluşmadığı, MAPE değerlerinin yükselmediği görülmektedir.

Tablo 4.52. Rastgele ayırmada altıncı grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.96	36.42	0.9294	4.94	36.78	0.9279	5.11	38.21	0.9225
	5	4.83	35.97	0.9316	4.99	36.87	0.9275	5.27	38.38	0.9220
	8	4.72	34.54	0.9368	4.93	36.77	0.9279	5.20	38.53	0.9211
2	3	4.91	36.35	0.9306	4.87	35.88	0.9313	5.10	38.35	0.9230
	5	4.65	34.31	0.9377	4.92	36.66	0.9284	5.46	40.70	0.9114
	8	4.77	35.03	0.9351	4.79	35.49	0.9327	4.89	36.25	0.9311
3	3	4.97	36.03	0.9309	4.81	35.46	0.9328	5.48	39.96	0.9153
	5	4.60	34.03	0.9386	4.85	36.02	0.9307	5.00	36.70	0.9287
	8	4.65	34.93	0.9351	4.70	34.85	0.9351	5.22	38.73	0.9213
Ortalama		4.78	35.29	0.9340	4.87	36.09	0.9305	5.19	38.42	0.9218

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-5} ve 1000 olarak alınmıştır.

*** Siyah dolgulu hücre tüm denemeler sonucunda BR için en düşük MAPE değeridir.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.53’de verilmiştir.

Tablo 4.53. Rastgele ayırmada altıncı grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	2.7420	16	0.6240	86	0.7330	721
	5	2.9650	14	1.6570	202	0.7640	1000
	8	3.0060	7	4.0580	467	0.7240	573
2	3	2.7670	12	4.2120	453	0.8560	948
	5	8.0210	9	2.0210	181	0.8870	921
	8	2.5750	10	3.9210	344	0.9260	831
3	3	2.5110	15	1.7130	179	0.9640	924
	5	2.5760	17	7.9790	666	1.0180	958
	8	3.3720	5	4.1300	269	0.9870	934
Ort.		3.3928	12	3.3683	316	0.8732	868

Tablo 4.53 incelendiğinde BR ve LM algoritmalarının eğitim süreleri bakımından yine birbirlerine benzer oldukları GDX algoritmasının bu parametre grubunda da yine çok daha kısa sürede eğitimi tamamladığı görülmektedir.

4.1.5.8 Rastgele ayırmada yedinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden

sadece döngü sayısının 500 olarak alınmasıyla elde edilmiştir. Yapılan analizler sonucunda Tablo 4.54'deki bulgulara ulaşılmıştır.

Tablo 4.54. Rastgele ayırmada yedinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.86	36.26	0.9305	4.92	36.83	0.9279	5.23	38.70	0.9213
	5	4.73	35.06	0.9345	4.89	36.25	0.9299	5.40	39.42	0.9176
	8	4.72	34.80	0.9355	4.97	36.93	0.9273	5.27	38.79	0.9200
2	3	4.95	36.07	0.9313	4.85	36.09	0.9305	6.69	47.89	0.8511
	5	4.93	36.24	0.9309	4.92	36.77	0.9279	5.26	38.99	0.9195
	8	4.76	37.32	0.9299	4.92	36.62	0.9285	7.54	52.94	0.8131
3	3	4.80	34.75	0.9363	4.84	36.01	0.9307	5.63	41.58	0.9078
	5	4.73	34.63	0.9366	4.92	36.64	0.9284	5.45	39.65	0.9180
	8	4.86	36.11	0.9310	9.33	64.23	0.6671	5.53	40.39	0.9122
Ortalama		4.82	35.69	0.9329	5.40	39.60	0.8998	5.78	42.04	0.8978

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-7} ve 500 olarak alınmıştır.

Tablo 4.54 incelendiğinde döngü sayısının azaltılması GDX algoritmasında hataların bir miktar yükselmesine neden olmuştur. Bunun dışında diğer iki algoritmaya bir etkisi olmamıştır. Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.55'te verilmiştir. Tablo 4.55 incelendiğinde döngü sayısının azaltılmasının BR ve LM algoritmalarına bir etkisi olmazken GDX algoritmasında ise en iyi döngü sayıları ve süreler bakımından döngü sayısının sınırlandırılması nedeniyle doğal olarak yarı yarıya bir azalma olduğu görülmektedir.

Tablo 4.55. Rastgele ayırmada yedinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
KS	NS	LM		BR		GDX	
		Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	2.0940	28	0.6250	81	0.4150	476
	5	2.1090	16	2.0160	268	0.4710	460
	8	2.2180	31	3.8890	500	0.3880	212
2	3	2.2340	13	1.0460	92	0.4360	500
	5	2.1560	2	3.5620	371	0.4730	483
	8	2.3740	12	5.2330	500	0.5310	394
3	3	2.9700	26	2.1080	209	0.5510	459
	5	2.9370	6	2.2010	148	0.4910	438
	8	3.3240	9	7.4520	500	0.5490	463
Ort.		2.4907	16	3.1258	297	0.4783	432

4.1.5.9 Rastgele ayırmada sekizinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece döngü sayısının 100 olarak alınmasıyla elde edilmiştir.

Tablo 4.56. Rastgele ayırmada sekizinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.91	36.05	0.9313	4.92	36.78	0.9279	6.64	49.45	0.8671
	5	4.85	35.73	0.9338	4.94	36.73	0.9280	7.19	51.43	0.8552
	8	4.65	34.56	0.9371	4.98	36.91	0.9274	6.91	51.19	0.8624
2	3	4.77	34.86	0.9352	4.92	36.73	0.9281	7.73	57.55	0.8203
	5	4.78	34.56	0.9367	4.82	35.63	0.9323	7.06	51.89	0.8538
	8	4.80	37.72	0.9267	4.90	36.71	0.9282	6.94	50.99	0.8601
3	3	6.97	48.82	0.8261	4.9	36.24	0.9299	8.50	61.67	0.7892
	5	4.62	34.21	0.9379	7.19	50.98	0.7958	7.78	58.51	0.8118
	8	5.40	40.03	0.9225	4.85	36.11	0.9304	7.79	56.72	0.8163
Ortalama		5.08	37.39	0.9208	5.16	38.09	0.9142	7.39	54.38	0.8374

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-7} ve 100 olarak alınmıştır.

Yapılan analizler sonucunda Tablo 4.56'daki bulgulara ulaşılmıştır. Tablo 4.56'dan döngü sayısından en fazla etkilenen algoritmanın GDX algoritması olduğu görülmektedir. Bu parametre grubunda yine en düşük MAPE değeri (%4.62) LM algoritması ile yapılan tahmin elde edilmiştir.

Tablo 4.57. Rastgele ayırmada sekizinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.5550	20	0.6240	85	0.1040	100
	5	0.7970	6	0.7670	100	0.1060	100
	8	0.7810	62	0.8120	100	0.1070	100
2	3	0.8280	44	0.8810	100	0.1210	100
	5	0.9210	5	0.9660	100	0.1160	87
	8	0.9370	10	1.0720	100	0.1470	100
3	3	0.9210	99	1.1390	100	0.1630	98
	5	1.0630	11	1.4140	100	0.1300	100
	8	1.1770	4	1.4980	100	0.1270	95
Ort.		0.8867	29	1.0192	98	0.1246	98

Ağ eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.57'de verilmiştir. Tablo 4.57'den döngü sayısının azaltılmasının

beklenen bir durum olarak tüm algoritmalarda eğitimde geçen süreleri azalttığı görülmektedir.

4.1.6 Rastgele ayırmada normalize edilmiş verilerle deneme

Çalışmanın bu kısmında sığırın görüntülerinden çeşitli görüntü işleme yöntemleriyle elde edilen ham veriler z-skora dönüştürülerek standartlaştırma uygulanmış, rastgele olarak %70 eğitim, %15 test ve %15 geçerlilik setine ayrılan veriler EK1d'deki MATLAB komutları yardımıyla yapay sinir ağlarında eğitim yapılmıştır. Aşağıda dokuz farklı parametre grubu için yapılan deneme sonuçları verilmiştir.

4.1.6.1 Rastgele ayırmada normalize edilmiş veriler ve MATLAB varsayılan parametreleriyle eğitim

Bu gruptaki parametreler Tablo 4.58'de görülen ve MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerleridir.

Tablo 4.58. MATLAB varsayılan parametreleri

MATLAB varsayılan parametreleri			
MU	λ	GD	Döngü
0.001	0.01	10^{-7}	1000

Verilerin rastgele olarak alt setlere ayrıldığı ve verilere normalizasyon uygulanarak MATLAB varsayılan parametreleriyle yapılan analizler sonucunda Tablo 4.59'daki bulgulara ulaşılmıştır. Tablo 4.59 incelendiğinde varsayılan parametrelerle yapılan eğitimde GDX algoritmasının daha önce varsayılan grubunda olduğu gibi düşük MAPE ve RMSE değeri ile tahmin ürettiği görülmektedir. Diğer iki algoritmada yine minimum hata değerlerine çok yakın sonuçlarla eğitimi tamamlamıştır.

Tablo 4.59. Normalize veriler ve varsayılan parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.88	35.95	0.9313	4.94	36.79	0.9278	5.37	39.77	0.9163
	5	4.78	35.27	0.9339	4.94	36.70	0.9281	5.33	40.07	0.9134
	8	4.69	34.62	0.9372	4.96	36.87	0.9276	5.23	38.21	0.9225
2	3	4.85	35.31	0.9336	4.89	36.60	0.9287	5.03	36.78	0.9285
	5	4.61	34.36	0.9371	4.86	36.01	0.9307	5.11	37.34	0.9263
	8	4.58	34.01	0.9385	4.93	36.73	0.9281	4.88	36.29	0.9309
3	3	4.82	35.84	0.9318	4.75	34.72	0.9355	5.52	41.09	0.9104
	5	4.70	34.11	0.9383	4.78	35.91	0.9312	5.32	38.96	0.9200
	8	4.80	35.29	0.9355	7.20	51.34	0.7944	4.94	38.09	0.9281
Ortalama		4.75	34.97	0.9352	5.14	37.96	0.9147	5.19	38.51	0.9218

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-7} ve 1000 olarak alınmıştır.

Ağların eğitimi tamamladıkları süreler ve en iyi döngü sayıları Tablo 4.60'da verilmiştir.

Tablo 4.60. Normalize veriler ve varsayılan parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
KS	NS	LM		BR		GDX	
		Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.2690	5	0.9550	136	0.7550	920
	5	1.0450	12	1.9920	268	0.2830	92
	8	2.2720	19	0.9020	111	0.8070	978
2	3	2.7910	18	6.8210	733	0.8070	1000
	5	2.7370	71	9.6560	965	0.5150	255
	8	2.4370	15	3.8410	356	0.8560	932
3	3	2.8220	22	6.7190	682	0.8820	1000
	5	2.8170	32	2.2580	196	0.8990	914
	8	3.4550	18	8.4990	601	0.9180	584
Ort.		2.2939	24	4.6270	450	0.7469	742

Tablo 4.60 incelendiğinde bir katman üç nöron bulunan yapıda LM algoritmasıyla beş döngü yaparak dahi %4.88'lik bir MAPE değeri ile tahmin yapılabildiği görülmektedir. Bu durum, veriler ayrılırken doğrulama setinin bulunması ve verilerin normalize edilmesinden dolayı, LM algoritması kullanıldığında diğer algoritmalarla göre daha az döngü ve daha yüksek doğrulukla tahmin yapılabildiğini göstermektedir.

4.1.6.2 Normalize edilmiş veriler ve birinci grup parametrelerle eğitim sonucu hatalar

Bu gruptaki parametreler MATLAB varsayılan parametrelerinden sadece MU değerinin 0.005 olarak alınmasıyla elde edilmiştir. Yapılan analiz sonucunda Tablo 4.61'deki bulgulara ulaşılmıştır. Tablo 4.61'den bu parametre grubunda en düşük MAPE değerinin (%4.64) önceki parametre gruplarında olduğu gibi LM algoritmasından elde edildiği görülmektedir. Genel ortalamala MAPE değeri (%4.79) dikkate alındığında yine LM algoritması tüm denemelerde ortalama olarak daha düşük MAPE değeri tahminlerin yapıldığı algoritma olmuştur.

Tablo 4.61. Rastgele ayırmada normalize veriler ve birinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.00	36.60	0.9308	4.94	36.74	0.9280	5.12	37.59	0.9257
	5	4.71	36.51	0.9277	4.96	36.84	0.9277	5.38	39.08	0.9184
	8	4.67	35.13	0.9348	4.97	36.81	0.9277	5.14	37.76	0.9241
2	3	4.90	35.97	0.9312	4.88	36.25	0.9299	4.99	37.06	0.9271
	5	4.64	33.90	0.9389	4.85	36.08	0.9305	5.21	37.94	0.9238
	8	4.80	36.80	0.9294	4.97	36.86	0.9275	5.08	37.85	0.9245
3	3	4.70	34.35	0.9373	4.77	35.33	0.9334	5.47	40.33	0.9149
	5	4.83	35.10	0.9360	4.85	36.02	0.9308	5.14	37.15	0.9275
	8	4.89	36.15	0.9321	4.86	33.02	0.9300	4.90	35.95	0.9313
Ortalama		4.79	35.61	0.9331	4.89	35.99	0.9295	5.16	37.86	0.9241

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.005, 0.01, 10^{-7} ve 1000 olarak alınmıştır.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.62'de verilmiştir. Tüm algoritmalarda döngü sayıları ve süreler bakımından kayda değer bir değişim gözlenmediği Tablo 4.62'den görülmektedir.

Tablo 4.62. Normalize veriler ve 1. grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	1.8580	24	2.0600	301	0.9030	973
	5	1.9360	19	0.8800	113	0.7530	937
	8	1.9420	8	0.9640	95	0.2840	95
2	3	2.1340	4	3.7300	306	0.9440	476
	5	2.8230	51	1.1110	114	0.9030	997
	8	2.6850	5	1.2590	119	0.9200	1000
3	3	2.4770	26	2.1040	185	0.8970	926
	5	2.8690	9	1.6060	136	0.9620	905
	8	3.4280	5	15.5950	1000	0.9500	917
Ort.		2.4613	17	3.2566	263	0.8351	803

4.1.6.3 Normalize veriler ve ikinci grup parametrelerle eğitim sonucu hatalar

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece MU değeri 0.01 olarak alınmasıyla elde edilmiştir.

Tablo 4.63. Rastgele ayırmada normalize veriler ve ikinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.85	35.70	0.9323	4.94	36.84	0.9278	5.08	38.34	0.9224
	5	4.88	40.68	0.9081	4.95	36.80	0.9278	5.20	38.90	0.9200
	8	4.58	33.74	0.9402	4.94	36.85	0.9277	5.36	39.78	0.9155
2	3	4.96	35.61	0.9334	4.92	36.57	0.9287	5.37	39.07	0.9196
	5	4.72	34.61	0.9362	4.92	36.71	0.9281	5.34	39.32	0.9188
	8	4.70	35.52	0.9332	4.92	36.78	0.9278	5.24	38.33	0.9217
3	3	4.76	34.77	0.9357	4.91	36.60	0.9285	5.23	37.89	0.9241
	5	5.12	36.64	0.9274	4.80	35.63	0.9324	5.17	37.71	0.9242
	8	4.56	33.81	0.9399	4.83	35.96	0.9310	5.12	37.17	0.9274
Ortalama		4.79	35.68	0.9318	4.90	36.53	0.9289	5.23	38.50	0.9215

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.01, 0.01, 10^{-7} ve 1000 olarak alınmıştır.

*** Siyah dolgulu hücre tüm denemeler sonucunda LM için elde edilen en düşük MAPE değeridir.

Yapılan analizler sonucunda Tablo 4.63'teki bulgulara ulaşılmıştır. Tablo 4.63 incelendiğinde bu parametre grubunda LM algoritmasıyla tüm denemelerden

elde edilen en düşük MAPE değerlerinden birisiyle (%4.56) eğitimin tamamlandığı görülmektedir.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.64'te verilmiştir. Tablo 4.64 incelendiğinde MU parametresinin 10 kat büyütülmesi GDX algoritmasında daha az döngülerle eğitimin sonlandırılmasına neden olmuştur. BR algoritmasında da en iyi döngü sayılarında bir miktar azalma olduğu söylenebilir.

Tablo 4.64. Normalize veriler ve 2. grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	1.4740	20	1.1030	155	0.5010	386
	5	2.5040	5	3.0540	411	0.7490	940
	8	2.1020	13	1.0380	129	0.2890	92
2	3	2.2240	5	4.3550	296	0.7870	997
	5	2.2840	15	3.8860	404	0.3160	95
	8	2.6690	9	0.9770	91	0.6330	453
3	3	2.3520	12	8.2060	814	0.8740	1000
	5	2.7080	26	1.4500	136	0.9490	997
	8	3.0550	10	2.7920	183	0.4020	113
Ort.		2.3747	13	2.9846	291	0.6111	564

4.1.6.4 Normalize veriler ve üçüncü grup parametrelerle eğitimde hatalar

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece λ değeri 0.05 olarak alınmasıyla elde edilmiştir. Yapılan analizler sonucunda Tablo 4.65'deki bulgulara ulaşılmıştır. Tablo 4.65 incelendiğinde LM algoritması yine en düşük MAPE değeri (%4.59) ile tahmin üretmiştir.

BR algoritması ile eğitilen yedi ağdan ortalama %4.85, GDX algoritması ile ise %5.01 MAPE değerleri elde edilmiştir. Ayrıca tüm algoritmalarda verilerin rastgele ayrıldığı ve normalize edildiği diğer parametre gruplarında olduğu gibi çok yüksek hataların oluşmadığı gözlemlenmiştir.

Tablo 4.65. Rastgele ayırmada normalize veriler ve 3. grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.94	36.74	0.9285	4.96	36.81	0.9278	5.26	39.08	0.9181
	5	4.82	35.32	0.9337	4.92	36.79	0.9279	5.31	38.95	0.9202
	8	4.62	33.69	0.9395	4.91	36.43	0.9292	5.30	39.23	0.9190
2	3	4.75	34.90	0.9352	4.92	36.69	0.9281	5.26	38.62	0.9212
	5	4.66	34.32	0.9375	4.89	36.29	0.9297	5.67	41.77	0.9071
	8	4.71	34.49	0.9367	4.92	36.69	0.9282	5.11	37.12	0.9282
3	3	4.86	35.50	0.9331	4.85	36.06	0.9306	5.17	38.09	0.9235
	5	4.59	33.89	0.9388	4.92	36.72	0.9281	5.65	41.68	0.9071
	8	4.73	35.37	0.9336	4.9	36.62	0.9286	5.01	36.60	0.9292
Ortalama		4.74	34.91	0.9352	4.91	36.57	0.9287	5.30	39.02	0.9193

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.05, 10^{-7} ve 1000 olarak alınmıştır.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.66’da verilmiştir. Tablo 4.66’dan bir önceki parametre grubuna nazaran tüm algoritmalarda en iyi döngü sayıları ve sürelerin arttığı görülmektedir.

Tablo 4.66. Normalize veriler ve 3. grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	1.9070	32	2.1330	313	0.3000	92
	5	1.8880	3	1.4050	180	0.8020	946
	8	1.8900	8	0.6390	74	0.8020	946
2	3	2.1550	21	5.4730	603	0.4010	188
	5	2.7610	78	1.2040	130	0.3390	101
	8	3.9370	7	10.6750	1000	0.5340	346
3	3	2.4870	30	6.4010	630	0.9180	995
	5	2.5290	8	5.9990	522	1.1240	992
	8	3.3390	30	5.0130	352	1.0250	932
Ort.		2.5437	24	4.3269	423	0.6939	615

4.1.6.5 Normalize veriler ve dördüncü grup parametrelerle eğitimde hatalar

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece λ değeri 0.1 olarak alınmasıyla elde edilmiştir. Yapılan analizler sonucunda Tablo 4.67’deki bulgulara ulaşılmıştır.

Tablo 4.67. Rastgele ayırmada normalize veriler ve 4. grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.85	35.70	0.9323	4.92	36.83	0.9278	5.13	37.61	0.9256
	5	4.88	40.68	0.9081	4.93	36.72	0.9282	5.44	39.43	0.9163
	8	4.58	33.52	0.9402	4.97	36.84	0.9276	5.15	37.84	0.9237
2	3	4.96	35.61	0.9336	4.88	36.35	0.9295	4.97	36.90	0.9277
	5	4.72	34.61	0.9362	4.92	36.77	0.9279	5.20	37.78	0.9246
	8	4.70	35.52	0.9332	4.93	36.69	0.9282	5.05	37.61	0.9251
3	3	4.83	35.11	0.9343	4.84	36.07	0.9305	5.53	40.73	0.9126
	5	4.69	34.60	0.9362	4.93	36.70	0.9282	5.12	37.38	0.9259
	8	4.79	37.33	0.9283	4.78	35.38	0.9331	5.04	36.44	0.9295
Ortalama		4.78	35.85	0.9314	4.90	36.48	0.9290	5.18	37.97	0.9234

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.05, 10^{-7} ve 1000 olarak alınmıştır.

*** Siyah dolgu ile gösterilen değer tüm analizler sonucunda elde edilen en düşük RMSE değeridir.

Tablo 4.67 incelendiğinde bu parametre grubunda yine en düşük MAPE değerinin (%4.58) LM algoritmasından elde edildiği görülmektedir. Ayrıca yapılan bütün analizlerden elde edilen en düşük RMSE değeri (33.52kg) bu parametre grubunda tespit edilmiştir. Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.68’de verilmiştir.

Tablo 4.68. Normalize veriler ve 4. grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
KS	NS	LM		BR		GDX	
		Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	1.1030	20	0.6170	83	0.7260	965
	5	1.8940	5	2.3570	305	0.8550	935
	8	2.1600	13	6.3820	656	0.2850	101
2	3	2.3370	5	2.1100	197	0.5640	462
	5	2.3380	15	3.9340	373	0.8020	1000
	8	2.5970	9	1.4840	108	0.8930	1000
3	3	4.0050	128	3.0600	248	0.8930	934
	5	2.6340	5	2.1360	154	0.9490	899
	8	3.0760	6	16.83	1000	0.5820	307
Ort.		2.4604	23	4.3233	347	0.7277	734

4.1.6.6 Normalize edilmiş veriler ve beşinci grup parametrelerle eğitim

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece gradyan değerinin 10^{-6} olarak alınmasıyla elde edilmiştir. Yapılan analiz sonucunda Tablo 4.69’daki bulgulara ulaşılmıştır.

Tablo 4.69. Rastgele ayırmada nrm. veriler ve beşinci grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.86	35.74	0.9321	4.91	36.83	0.9279	5.12	37.59	0.9257
	5	4.71	34.63	0.9364	4.89	36.25	0.9299	5.38	39.08	0.9184
	8	4.71	34.76	0.9361	4.97	36.93	0.9273	5.14	37.76	0.9241
2	3	4.94	35.89	0.9327	4.85	36.09	0.9305	4.99	37.06	0.9271
	5	4.60	34.50	0.9372	4.92	36.77	0.9279	5.21	37.94	0.9238
	8	4.76	36.20	0.9329	4.92	36.62	0.9285	5.08	37.84	0.9245
3	3	4.85	35.17	0.9340	4.84	36.01	0.9307	5.47	40.33	0.9149
	5	4.73	35.48	0.9331	4.92	36.64	0.9285	5.14	37.15	0.9275
	8	4.92	37.26	0.9271	6.99	49.20	0.8023	4.90	35.95	0.9313
Ortalama		4.79	35.51	0.9335	5.13	37.93	0.9148	5.16	37.86	0.9241

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-6} ve 1000 olarak alınmıştır.

Tablo 4.69 incelendiğinde bu parametre grubunda LM algoritmasının yine en düşük MAPE değeri (%4.6) ile eğitimi sonlandırdığı görülmektedir. BR algoritması üç katman ve sekiz nöron olduğu durumda yüksek MAPE değeri ile (%6.99) sonuçlar üretmiştir.

Tablo 4.70. Normalize veriler ve beşinci grup parametrelerle eğitimde süreler ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	1.8610	20	0.6030	83	0.7330	973
	5	1.8650	4	2.0740	263	0.7860	937
	8	2.3290	2	6.5020	782	0.3230	95
2	3	2.1370	8	0.8020	91	0.5860	476
	5	2.7220	20	3.6230	374	0.8490	997
	8	2.7780	9	11.2180	1000	0.8640	1000
3	3	4.1500	157	2.2150	204	0.8760	926
	5	2.5560	5	2.0660	146	0.9180	905
	8	3.3460	5	15.0400	996	0.9880	917
Ort.		2.6382	26	4.9048	438	0.7692	803

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.70’da verilmiştir. Tablo 4.70 incelendiğinde bir önceki parametre grubunda üç katman ve üç nöron bulunan yapıda LM algoritması ile eğitimde yüksek bulunan en iyi döngü sayısının bu parametre grubu için de geçerli olduğu görülmektedir.

4.1.6.7 Normalize veriler ve altıncı grup parametrelerle eğitim sonucu hatalar

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece gradyan değerinin 10^{-5} olarak alınmasıyla elde edilmiştir.

Yapılan analizlerin sonucunda Tablo 4.71'deki bulgulara ulaşılmıştır. Tablo 4.71'den LM algoritması ile en düşük MAPE değerinin (%4.6) iki katman ve beş nöronun olduğu yapıda elde edildiği görülmektedir.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.72'de verilmiştir. Tablo 4.72 incelendiğinde bu grupta yine LM algoritması ile eğitimde üç katman ve üç nöron bulunan yapıda en iyi döngü sayısı yüksek bulunmuştur.

Tablo 4.71. Rastgele ayırmada nrm. veriler ve altıncı grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
KS	NS	LM			BR			GDX		
		MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.86	35.75	0.9321	4.92	36.83	0.9279	5.15	37.92	0.9248
	5	4.71	34.63	0.9364	4.89	36.25	0.9299	5.20	38.01	0.9232
	8	4.71	34.76	0.9361	4.97	36.93	0.9273	5.31	39.15	0.9174
2	3	4.94	35.89	0.9327	4.85	36.09	0.9305	5.38	39.62	0.9157
	5	4.60	34.50	0.9372	4.92	36.77	0.9279	5.15	37.76	0.9246
	8	4.76	36.20	0.9329	4.92	36.62	0.9285	5.04	37.03	0.9269
3	3	4.85	35.17	0.9340	4.76	35.17	0.9339	5.20	37.73	0.9248
	5	4.73	35.48	0.9331	4.77	35.63	0.9322	5.23	38.80	0.9201
	8	4.92	37.26	0.9271	4.87	36.78	0.9279	5.21	38.28	0.9220
Ortalama		4.79	35.52	0.9335	4.87	36.34	0.9296	5.21	38.26	0.9222

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-5} ve 1000 olarak alınmıştır.

Bunun dışında BR algoritmasında sekiz nöron, iki ve üç katman olduğu yapılarda ağın eğitimi 1000 döngüye kadar devam etmiştir. Nöron sayısının gereksiz artırılması BR algoritmasında eğitimin erken durdurulmamasında etkili olmuş olabilir.

Tablo 4.72. Normalize veriler ve altıncı grup parametrelerle eğitimde süre ve iterasyon sayıları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	1.8830	20	0.6330	83	0.7270	904
	5	1.8350	4	1.9620	263	0.7870	973
	8	1.9420	2	6.4490	782	0.7640	937
2	3	2.0660	8	0.8160	91	0.5860	507
	5	2.7110	20	3.5630	374	0.5460	321
	8	2.3340	9	10.3320	1000	0.3690	89
3	3	4.2780	157	2.1040	221	0.3850	168
	5	2.9290	5	5.0500	434	0.7600	570
	8	3.0240	5	14.3560	1000	1.0810	998
Ort.		2.5558	26	5.0294	472	0.6672	607

4.1.6.8 Normalize edilmiş veriler ve yedinci grup parametrelerle eğitim sonucu hatalar

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece döngü sayısının 500 olarak alınmasıyla elde edilmiştir. Yapılan analizler sonucunda Tablo 4.73'teki bulgulara ulaşılmıştır. Tablo 4.73 incelendiğinde bu parametre grubunda LM algoritmasının yine tüm denemelerden elde edilen en düşük MAPE değerlerinden biri (%4.56) ile sonuç verdiği görülmektedir.

Tablo 4.73. Rastgele ayırmada normalize veriler ve 7. grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	5.14	43.47	0.9007	4.95	36.79	0.9279	5.19	38.50	0.9220
	5	4.86	35.92	0.9318	4.97	36.97	0.9272	5.22	38.62	0.9208
	8	4.61	34.04	0.9384	4.97	36.89	0.9275	5.27	38.49	0.9216
2	3	4.71	34.26	0.9377	4.94	36.70	0.9281	5.38	39.65	0.9161
	5	4.72	35.02	0.9352	4.93	36.70	0.9282	5.32	38.71	0.9207
	8	4.56	33.79	0.9393	4.91	36.75	0.9280	5.27	39.39	0.9173
3	3	4.78	35.23	0.9344	4.86	36.10	0.9304	5.81	43.07	0.9013
	5	4.65	37.04	0.9265	4.82	35.93	0.9310	5.78	42.73	0.9038
	8	4.75	35.53	0.9344	7.13	50.95	0.7959	5.48	41.06	0.9101
Ortalama		4.75	36.03	0.9309	5.16	38.20	0.9138	5.41	40.02	0.9149

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-7} ve 500 olarak alınmıştır.

*** Sarı dolgulu hücre tüm denemeler sonucunda LM için elde edilen en düşük MAPE değeridir.

BR algoritmasıyla yapılan eğitimde üç katman ve sekiz nöron bulunan yapıda lokal minimuma takılma söz konusu olabilir. Çünkü bu yapıda BR algoritmasının ortalama MAPE değerinin oldukça üzerinde bir MAPE değeri (%7.13) ile tahmin yapıldığı görülmektedir.

Tablo 4.74. Rastgele ayırmada normalize veriler ve 7. grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.6480	14	0.8330	103	0.4700	353
	5	0.5230	11	4.3880	500	0.4640	461
	8	2.9170	54	1.9800	162	0.4170	480
2	3	3.2340	45	4.8290	500	0.4010	224
	5	2.6800	9	4.9820	500	0.4160	498
	8	3.1510	12	4.5830	425	0.5010	496
3	3	3.5110	63	5.0840	500	0.5700	398
	5	4.5990	13	5.6970	500	0.4710	499
	8	3.4960	4	7.2960	477	0.6230	449
Ort.		2.7510	25	4.4080	407	0.4814	429

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.74’de verilmiştir. Tablo 4.74’ten döngü sayısını 500 ile sınırlandırmanın GDX algoritmasında eğitim sürelerinde doğal olarak yarı yarıya düşüşe sebep olduğu görülmektedir.

4.1.6.9 Normalize veriler ve sekizinci grup parametrelerle eğitimde hatalar

Bu gruptaki parametreler MATLAB paket programı tarafından ileri beslemeli geri yayımlımlı ağ için otomatik olarak atanan varsayılan parametre değerlerinden sadece döngü sayısının 100 olarak alınmasıyla elde edilmiştir. Yapılan analizler sonucunda Tablo 4.75’teki bulgulara ulaşılmıştır. Tablo 4.75 incelendiğinde bu parametre grubunda LM algoritmasının yine en düşük MAPE değerini (%4.58) verdiği görülmektedir. Döngü sayısının 100 ile sınırlandırıldığı diğer denemelerde olduğu gibi bu parametre grubunda da GDX algoritmasında MAPE değerlerinde yükselme görülmüştür. Böylece verilerin normalize edildiği durumdaki bütün denemelerde en düşük MAPE değeri daima LM algoritmasından elde edilmiştir.

Tablo 4.75. Rastgele ayırma, normalize veriler ve 8. grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR								
		LM			BR			GDX		
KS	NS	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
1	3	4.86	36.01	0.9312	4.96	36.85	0.9276	6.68	49.43	0.8690
	5	4.81	35.16	0.9342	4.94	36.77	0.9279	6.01	44.84	0.8916
	8	4.69	34.52	0.9365	4.96	36.85	0.9276	5.80	43.39	0.9003
2	3	4.91	35.59	0.9330	4.90	36.40	0.9293	6.57	48.96	0.8710
	5	4.58	33.57	0.9401	4.93	36.73	0.9281	6.02	45.06	0.8904
	8	4.61	34.83	0.9354	4.93	36.68	0.9283	5.52	40.47	0.9133
3	3	4.70	34.23	0.9377	4.72	35.02	0.9345	6.92	50.44	0.8624
	5	4.70	34.84	0.9361	4.90	36.79	0.9279	6.83	49.99	0.8649
	8	4.76	34.77	0.9379	4.90	36.53	0.9288	5.43	39.56	0.9168
Ortalama		4.74	34.84	0.9358	4.90	36.51	0.9289	6.20	45.79	0.8866

* Kalın rakamlarla gösterilen değerler her bir algoritmayla elde edilen en düşük MAPE değerleridir.

** MU, λ , GD ve Döngü sayıları sırasıyla 0.001, 0.01, 10^{-7} ve 100 olarak alınmıştır.

Ağın eğitimi esnasında geçen süreler ve eğitimin sona erdirildiği en iyi döngü sayıları Tablo 4.76’de verilmiştir. Tablo 4.76 incelendiğinde bu parametre grubunda en iyi döngü sayılarında ve sürelerde bir azalma olduğu görülmektedir. Ancak GDX algoritmasında döngü sayısının azaltılması döngü sayısının azaltıldığı diğer tüm denemelerde olduğu gibi bu grup parametrelerle yapılan denemede de hataların yükselmesine neden olmuştur.

Tablo 4.76. Rastgele ayırma, normalize veriler ve 8. grup parametrelerle yapılan analiz bulguları.

		ALGORİTMALAR					
		LM		BR		GDX	
KS	NS	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
1	3	0.4390	21	0.7340	100	0.1000	100
	5	0.7170	6	0.7860	100	0.0970	100
	8	0.7740	14	0.8040	100	0.1010	93
2	3	0.8330	88	0.8510	100	0.1380	100
	5	0.9720	16	0.9490	100	0.1160	100
	8	1.0340	8	1.0340	100	0.1160	96
3	3	0.9020	10	0.9750	100	0.1200	100
	5	1.0040	10	1.1190	100	0.1320	100
	8	1.1730	5	1.4600	100	0.1320	92
Ort.		0.8720	20	0.9680	100	0.1169	98

4.1.7 Rastgele ayırma için tüm durumlara ait özet

Verilerin rastgele olarak %70 eğitim, %15 test, %15 doğrulama setine ayrıldığı durum için MATLAB tarafından otomatik olarak verilen parametre grupları dahil olmak üzere dokuz farklı parametre gurubu, normalize edilmiş ve

ham veriler olarak iki farklı veri grubu için toplam 36 adet tablo elde edilmiştir (Tablo 4.40-Tablo 4.76). Bu tablolardan her bir algoritma için en az hata ile sonuç veren durumlar ve bunlara ait en iyi döngü sayıları, eğitimde geçen süreler ile ortalama hataları gösteren özet tablolar EK 2’de verilmiştir. Ek 2a incelendiğinde LM algoritması ile en düşük MAPE değerinin (%4.56) 3 kez elde edildiği görülmektedir. Çalışma sonunda en düşük MAPE değerlerinin elde edildiği durumlar kombine edilerek yapılan denemelerden Şekil 4.1’deki sonuçlar elde edilmiştir. Şekil 4.1’den en düşük MAPE değerinin (%4.15), (2-8-8-1) ağ mimarisi ile LM algoritmasıyla eğitimde 4. ağda (MAPE4) elde edildiği anlaşılmaktadır.



4.2 Çok Değişkenli Uyarlanabilir Regresyon Eğrileri (MARS) Bulguları

Sığırların üstten sırt (dorsal) görüntülerinden hesaplanan alan ve yükseklik değerlerinin girdi verisi, ağırlık değerlerinin ise çıktı verisi olarak kullanıldığı MARS analizinde Tablo 4.77’de verildiği gibi MAPE, RMSE ve R^2 değerleri sırasıyla %4.75, 32.95kg ve 0.9334 olarak tespit edilmiştir.

Tablo 4.77. MARS analizi ile yapılan tahmin için performans ölçütü değerleri

PERFORMANS DEĞERLENDİRME ÖLÇÜTLERİ		
MAPE (%)	RMSE (Kg)	R^2
4.75	32.95	0.9334

Analiz Salford Predictive Modeler programı demo versiyonunda ve programın MARS yöntemi için önerdiği varsayılan ayarlar kullanılarak yapılmıştır. Program çıktıları Ekler kısmında Ek 7a ve Ek 7b’de verilmiştir. Analiz sonucunda temel fonksiyonlar:

$$BF1 = \max\{0, X_1 - 0.209354\}$$

$$BF3 = \max\{0, X_2 - 138.49\}$$

$$BF5 = \max\{0, X_1 - 0.360215\}$$

$$BF15 = \max\{0, X_2 - 153.662\}$$

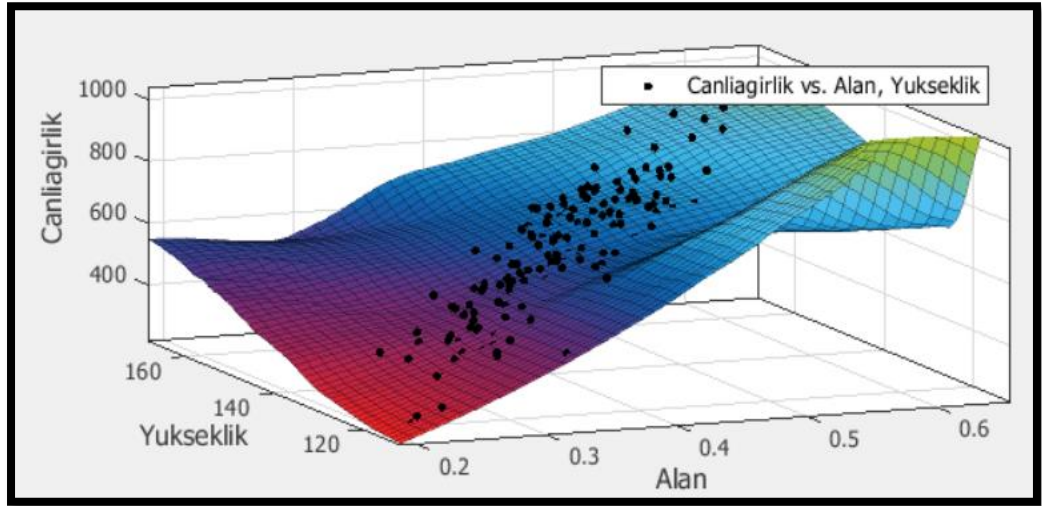
$$BF17 = \max\{0, X_2 - 152.146\}$$

olarak elde edilmiştir. Yöntemin modeli Eşitlik 4.1’de verilmiştir:

$$\hat{Y} = 268.021 + 1690.46 \times BF1 + 3.22929 \times BF3 - 461.836 \times BF5 - 25.7811 \times BF15 + 226744 \times BF17 \quad (4.1)$$

4.3 Çoklu Doğrusal Regresyon Analizi Bulguları

Görüntü işleme kısmında anlatılan komutlar yardımıyla hesaplanan sığırların görüntüde kapladığı alan yüzdeleri, sağrı yükseklikleri ile canlı ağırlıkları arasındaki ilişkileri gösteren grafik Şekil 4.2’de verilmiştir.



Şekil 4.2. Alan yüzdesi, yükseklik ve canlı ağırlıklar arasındaki ilişkilere ait yanıt yüzey grafiği.

Canlı ağırlık ile alan yüzdeleri arasında doğrusal bir ilişki söz konusudur. Bu ilişkiye ait korelasyon değeri 0.9587 olarak tespit edilmiştir. Yine yükseklik ve alan yüzdeleri arasındaki korelasyon değeri 0.8710 olarak hesaplanmıştır. Değişkenler arasındaki bu yüksek korelasyonlar, buna bağlı olarak yüksek belirleme katsayısı nedeniyle öncelikle Eşitlik 4.2’de modeli verilen çoklu doğrusal regresyon eşitliği yardımıyla hayvanların canlı ağırlıkları tahmin edilmiştir.

$$\hat{Y} = -297.74 + 1304.72X_1 + 2.54X_2 \quad (4.2)$$

Burada;

$\hat{Y}$: Canlı ağırlık bağımlı değişkeninin tahmin değeri (kg),

X_1 : Alan bağımsız değişkeni (%),

X_2 : Yükseklik bağımsız değişkeni (cm).

Çoklu doğrusal regresyon analizi SPSS v25 paket programı kullanılarak yapılmıştır. Analizin varsayımlarından normallik varsayımı Kolmogorov-Smirnov test istatistiği ile yapılmış ve tüm değişkenlerin normal dağıldığı tespit edilmiştir ($P > 0.01$). Hataların normal dağılıp dağılmadığına da Kolmogorov-Smirnov test istatistiği ile bakılmış, test değeri $K=0.076$ olarak tespit edildiğinden hataların normal dağıldığına karar verilmiştir ($P > 0.01$). Bağımsız değişkenler arasında çoklu bağlantılılık varsayımı için Durbin-Watson testi yapılmış test değeri $d=1.667$ olarak

bulunmuş ve bu değer yaklaşık olarak 2 olduğu için çoklu bağlantılılığın olmadığına karar verilmiştir. Otokorelasyon için ise VIF değerine bakılmış ve 3.916 olarak tespit edilen bu değer 10'dan küçük olduğu için bağımsız değişkenler arasında otokorelasyon olmadığına karar verilmiş, varsayımlar yerine getirildiğinden çoklu doğrusal regresyon analizi yapılmıştır.

Tablo 4.78. Çoklu doğrusal regresyon analizi ile yapılan tahmin için performans ölçütü değerleri

PERFORMANS DEĞERLENDİRME ÖLÇÜTLERİ		
MAPE (%)	RMSE (Kg)	R ²
5.02	37.06	0.9266

Tablo 4.78'de çoklu doğrusal regresyon ile yapılan tahmine ait performans değerlendirme ölçütleri verilmiştir. Et sığırlarının kuşbakışı görüntüsünden hesaplanan sığır alanı ve yüksekliği bağımsız değişkenler, et sığırlarının canlı ağırlığı ise bağımlı değişken olarak ele alınmış ve tahmine ait MAPE değeri %5.02, RMSE değeri 37.06kg ve R² değeri ise 0.9266 olarak hesaplanmıştır.

4.4 CHAID Analizi Bulguları

Çalışmanın bu kısmında et sığırlarının alan ve yükseklik bilgileri kullanarak canlı ağırlıklarını tahmin etmek amacıyla CHAID analizi uygulanmış ve Şekil 4.3'teki bulgulara ulaşılmıştır. Ayrıca CHAID analizi ile yapılan tahmine ait performans değerlendirme ölçütleri hesaplanarak Tablo 4.79'da verilmiştir.

Tablo 4.79. CHAID analizi ile yapılan tahmin için performans ölçütü değerleri

PERFORMANS DEĞERLENDİRME ÖLÇÜTLERİ		
MAPE (%)	RMSE (Kg)	R ²
5.36	38.42	0.9211

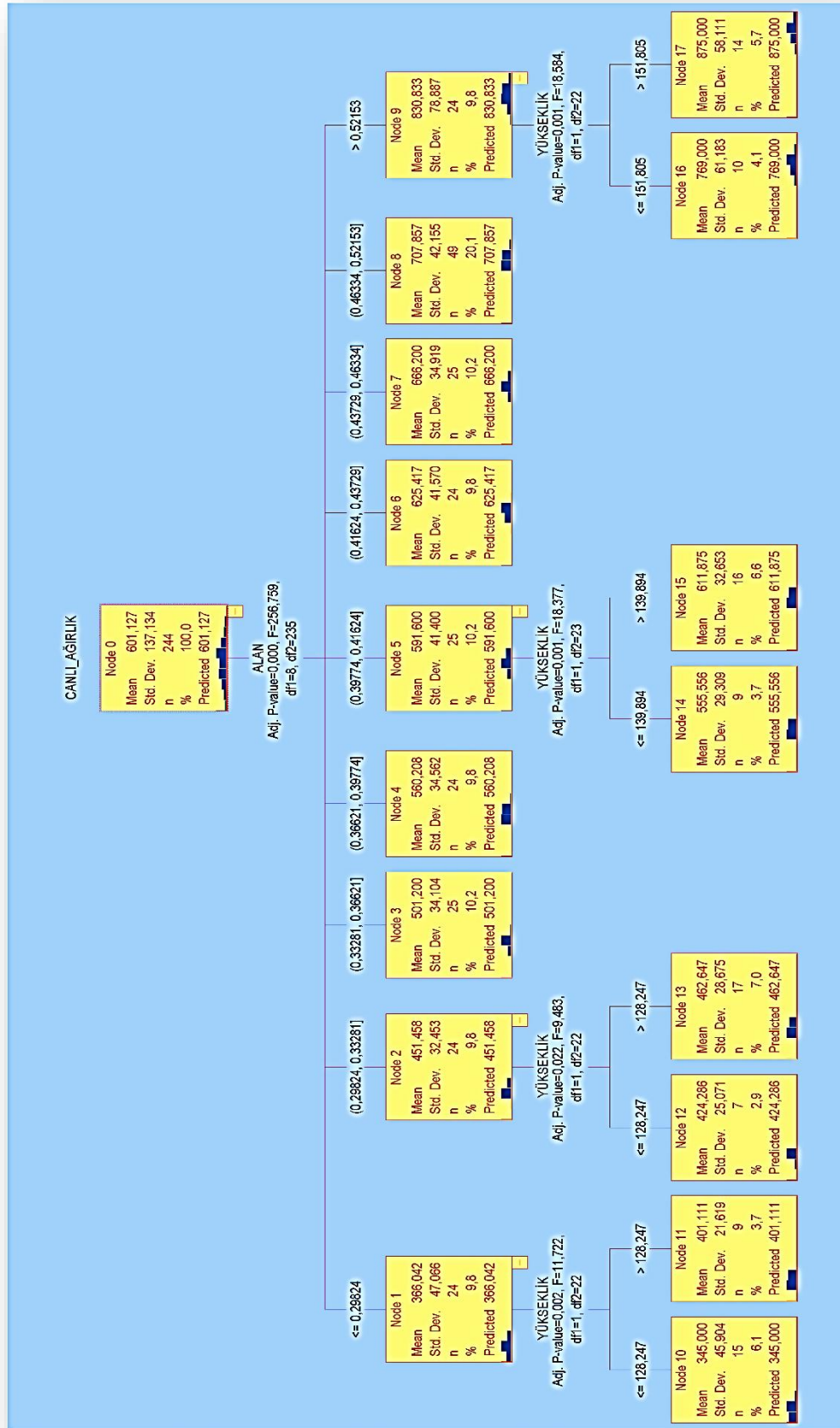
Şekil 4.3 incelendiğinde canlı ağırlık üzerine alan değişkeninin, yükseklik değişkeninden daha etkili olduğu görülecektir. CHAID analizi algoritması ile veri seti, alan bakımından kendi içinde homojen birbirleri arasında heterojen olan dokuz farklı gruba (Düğüm 1-9) ayrılmıştır. Kök düğüm olan Düğüm 0, veri setine ait genel tanımlayıcı istatistikleri içermektedir. Canlı ağırlıklara ait genel ortalama

601.13 $\pm$ 137.13kg olarak hesaplanmıştır. Düğüm 1, Düğüm 5 ve Düğüm 9’da alan değişkeninden sonra yükseklik de etkili bir faktördür.

Düğüm 1’de tüm görüntünün %29.82’sinden daha az alan kaplayan et sığırlarının ortalaması 366.042 $\pm$ 47.066kg, bu grupta olup yüksekliği 128.25cm’den fazla olanların ortalama 401.111 $\pm$ 21.619kg, 128.25cm’den daha az olanların ise ortalama olarak 345 $\pm$ 45.904kg olduğu görülmektedir.

Düğüm 5’te %39.77 ile %41.62 arasında alana sahip olan sığırlardan, yüksekliği 139.89cm’den az olanlarda ağırlık 555.56 $\pm$ 29.309kg iken yüksekliği 139.89cm’den fazla olanlar ortalama olarak yaklaşık 611.875 $\pm$ 32.653kg olarak tespit edilmiştir.

Düğüm 9’da toplam görüntü alanının %52.15’inden daha fazla alana sahip et sığırlarının (24 et sığırı) ortalaması 830.83 $\pm$ 78.887 kg olarak belirlenmiştir. Bu gruptaki sığırlardan yüksekliği 151.81cm’den daha düşük olanlarda canlı ağırlık 769 $\pm$ 61.183kg iken 151.81cm’den daha fazla olan sığırlarda 875 $\pm$ 58.111kg olarak hesaplanmıştır.



Şekil 4.3. Et sığırlarında canlı ağırlık tahmini için CHAID analizi diyagramı

Tez çalışması kapsamında ayrıca yapay sinir ağlarında kullanılan parametre, algoritma, katman sayısı ve nöron sayısı ile en iyi döngü sayıları ve eğitimde geçen sürelerin MAPE, RMSE, R^2 ve EDS ile süreler bakımından karşılaştırılması da CHAID analizi ile yapılmıştır. Elde edilen ağaç diyagramları EK 4, EK 5 ve EK 6'da verilmiştir.

EK 4'te iki farklı veri ayırma yönteminin beraber etkisinin incelendiği bir CHAID analizi diyagramı verilmiştir. EK 5'de sadece çapraz doğrulama ile verinin ayrıldığı durumda MAPE, EDS ve süreler bakımından karşılaştırmalar sunulmuştur. Son olarak EK 6'da rastgele olarak verinin %70 eğitim, %15 doğrulama ve %15 test setine ayrıldığı durumdaki MAPE, RMSE, R^2 , EDS ve süreler bakımından bütün faktörlerin etkisi karşılaştırılmıştır.

EK 4 incelendiğinde MAPE değerini azaltma bakımından en önemli faktörün algoritmalar olduğu görülmektedir. Bütün denemelere ait genel ortalama MAPE için %6.478'dir. Düğüm 1, 2 ve 3 incelendiğinde çapraz doğrulama ile verinin ayrıldığı durumda, en yüksek hata ile (ortalama %8.69) LM algoritmasının en düşük MAPE değeri (%5.17) ile BR algoritmasının sonuç verdiği görülmektedir. LM algoritmasını en çok etkileyen faktör verinin ayrılma biçimidir ve veri çapraz doğrulama ile ayrıldığında ortalama %12.60 olan MAPE değeri (Düğüm 5), verinin rastgele ayrıldığı durumda ortalama %4.83 (Düğüm 4) olmaktadır. Bunun nedeni çapraz doğrulamada veri ayrılırken %85 eğitim %15 test seti olacak şekilde doğrulama seti olmadan ayrılması ve buna bağlı olarak genellikle ortalama 25 döngü ile eğitimi tamamlayan LM algoritmasının 1000 döngüye kadar eğitime devam etmesidir. Ancak Düğüm 28 incelendiğinde bu durumda MAPE'nin %5.91 olduğu görülecektir. Bu da LM algoritmasında doğrulama seti olmasa bile katman ve nöron sayısının düşük tutulması ile yine hatanın azalacağını göstermektedir.

Düğüm 13 katman sayısı farketmeksizin sadece üç nöron kullanarak MAPE değerinin azaltılabileceğini (%5.90) göstermektedir.

Nöron sayısı rastgele ayırma durumunda LM algoritmasında yine önemli bir faktör olarak kendisini göstermektedir. Ancak bu durumda LM algoritmasından üç nöron kullanıldığı durumda (Düğüm 11) daha yüksek (%4.99), beş ve sekiz nöron

kullanıldığı durumda (Düğüm 12) daha düşük MAPE değeri (%4.75) elde edilmiştir. Verinin z-skorlarına dönüştürülerek normalize edilmesine bağlı olarak verilerin rastgele ayrıldığı (Düğüm 24, 25, 26, 27) tüm denemelerde MAPE değerlerinde bir miktar azalma olmuştur. Katman sayısı MAPE ve RMSE değerlerinin azalmasında bir başka faktör olarak karşımıza çıkmaktadır. Verilerin rastgele ayrıldığı denemelerde LM algoritmasında iki katman (Düğüm 44) bulunması durumunda en düşük MAPE değerleri (18 ağın ortalaması %4.67) elde edilmiştir. Çapraz doğrulamada ise tam tersi bir durum söz konusudur, katman sayısının birden fazla alınması (Düğüm 29) iki kattan daha yüksek MAPE değerine (%12.89) neden olmuştur. Özellikle 3 katman ve 8 nöron kullanıldığı durumda (Düğüm 32) LM algoritmasında MAPE değerleri yaklaşık 8 kat (%34.62) artmıştır.

Tüm veri ayırma yöntemlerinin analize dahil edildiği durumda en düşük hata ile tahminler BR algoritması kullanılarak yapılmıştır. BR algoritmasında zaten doğrulama setinin bulunmayışı nedeniyle LM algoritmasında olduğu gibi çok yüksek MAPE değerleri elde edilmemiş, tüm denemeler için genel ortalama MAPE değeri %5.17 olarak (Düğüm 2) tespit edilmiştir. Bu algoritma için katman sayısı önemli bir faktör olmuş 1 ve 2 katman kullanılması durumunda %5.02 MAPE değeri (Düğüm 6) tespit edilirken, 3 katman kullanılması durumunda MAPE değeri %5.48'e yükselmiştir. Katman sayısından sonra en önemli faktör LM algoritmasında olduğu gibi verinin ayrılma biçimidir. Rastgele ayırmada (Düğüm 16) %4.92 olarak tespit edilen MAPE değeri, çapraz doğrulamada (Düğüm 17) %5.12 olarak belirlenmiştir. Ancak yinede LM algoritmasında olduğu gibi 8 kat bir farklılık söz konusu olmamıştır. Standartlaştırma yine bu algoritma için de önemli bir faktör olmuş (Düğüm 49 ve 50), verinin standartlaştırılması MAPE değerini bir miktar (%0.04) azaltmıştır. BR algoritması ile en yüksek hata 3 katman ve 8 nöronun kullanıldığı durumda (Düğüm 19) %6.35 olarak hesaplanmıştır.

GDX algoritmasında MAPE değerleri en çok döngü sayısından etkilenmiştir. En yüksek hata döngü sayısının 100 ile sınırlandırıldığı ve standartlaştırmanın yapılmadığı durumda (Düğüm 23) %7.54 olarak tespit edilmiştir. GDX algoritması verinin ayrılma biçiminden LM algoritması kadar etkilenmemiş (Düğüm 39 ve 40) ancak yinede çapraz doğrulama ile verinin ayrılması MAPE değerinde bir miktar (%0.13) yükselmeye neden olmuştur. Bu algoritma ile döngü sayısının 1000, MU

parametresinin 0.005 ve 0.01, öğrenme oranının 0.1 ve gradyan değerinin 10^{-6} , 10^{-5} olarak alındığı parametre gruplarında, veriler standardize edildiğinde, veri rastgele olarak ayrıldığında ve 8 nöron kullanıldığında (Düğüm 53) en düşük MAPE değeri (%5.19) elde edilmiştir.

Çapraz doğrulama ile verilerin ayrıldığı duruma ait CHAID analizi diyagramı EK 5'te verilmiştir. EK 5 incelendiğinde çapraz doğrulamanın MAPE değeri üzerine etkisi daha net bir şekilde görülecektir. Sonuçlar EK 4'te izah edilen bulgulara benzerlik teşkil ettiğinden yeniden açıklanmayacaktır. Bunun yerine algoritmaların EDS ve süreler bakımından çapraz doğrulama ile veri ayırma durumunda nasıl davrandığı üzerinde durulacaktır. Bu anlamda EK 5b incelenecek olursa genel olarak (Kök düğüm) tüm algoritmalar için ortalama süre 3.861 sn'dir. GDX algoritması her durumda 1sn'nin altında bir sürede eğitimi tamamlamıştır. LM ve GDX algoritmalarında döngü sayısının azaltılması süreleri düşürmüştür, BR algoritması bu faktörden etkilenmemiştir. GDX algoritmasında nöron sayısı etkili olmazken katman sayısının artması çok az bir miktar (0.751sn-0.822sn-0.887sn) süreyi artırmıştır. BR algoritmasında süre bakımından tek etkili faktör katman sayısı olmuş, katman sayısının birden fazla alınması durumunda eğitim süreleri yaklaşık 3 kat artmıştır.

EK 5c incelendiğinde çapraz doğrulama ile verinin ayrıldığı durumda en iyi döngü sayısı için en önemli faktör müsaade edilen en çok döngü sayısı olarak (Düğüm 1, 2 ve 3) tespit edilmiştir. Döngü sayısının 1000 olarak alındığı durumda algoritmalar ikinci bir faktör olarak karşımıza çıkmaktadır. LM algoritmasında hataların yüksek çıkmasının en önemli nedeninin EK 5c incelendiğinde en iyi döngü sayıları olduğu görülecektir. Çünkü rastgele ayırmada ortalama olarak 25 döngü ile eğitimi tamamlayan LM algoritması burada en iyi döngüye ulaşmak için ortalama olarak 850 döngü yapmıştır. Çalışmanın yöntem kısmında verilen Şekil 3.26'da LM algoritmasının performans grafiği incelenecek olursa döngü sayısı artırıldığında LM algoritmasında yüksek hatalar oluştuğu görülecektir. Özetle LM algoritmasında eğitimin erken durdurulmaması yüksek hatalara neden olmaktadır. BR algoritması için katman sayısı önemli bir faktör olarak tespit edilmiş (Düğüm 12 ve 13), iki ve üç katman alınması durumunda (515) bir katman kullanıldığı durumdan (238) daha fazla döngü ile eğitim tamamlanmıştır.

EK 6a'da verilerin rastgele ayrıldığı durumda MAPE'yi etkileyen faktörlerin incelendiği CHAID analizi diyagramı verilmiştir. Diyagram incelendiğinde en düşük MAPE (%4.82) ile LM algoritması kullanıldığı durumda (Düğüm 1) tahmin yapıldığı görülecektir. BR algoritması hatayı azaltmadaki başarısı bakımından ortalama olarak %5.064 MAPE değeriyle LM algoritmasından sonra (Düğüm 2) gelmektedir. Üç algoritma için en yüksek hata ortalaması ile (%5.552) tahminin yapıldığı algoritma GDX algoritması olarak (Düğüm 3) tespit edilmiştir. GDX algoritması için nöron ve katman sayısı etkili olmazken BR algoritmasında standartlaştırmanın hatayı azaltmada bir etkisi bulunmamıştır.

LM algoritmasında beş ve sekiz nöron bulunduğu durumda (Düğüm 5, %4.745) üç nöron olduğu durumdan (Düğüm 4, %4.987) daha düşük hata ile tahminler elde edilmiştir. LM algoritmasında üç nöron kullanıldığı durumda üç katmanla düşük MAPE değeri (Düğüm 23, %4.794) elde edilirken, beş ve sekiz nöronun kullanıldığı durumda iki katman kullanılması MAPE değerini (Düğüm 25, %4.668) azaltmıştır. LM algoritması ile yapılan 162 deneme sonucunda iki gizli katmanda beş ve sekiz nöron bulunması ve standartlaştırma yapılması durumunda en düşük MAPE değeri %4.668 (18 denemenin ortalaması, Düğüm 25) olarak elde edilmiştir. LM algoritmasında en yüksek MAPE değeri (%5.111) ile katman sayısı önemli olmaksızın 3 nöron kullanılması ve verinin standardize edilmemesi durumunda elde edilmiştir.

BR algoritmasında nöron sayısının üç ve beş alındığı durumda MAPE değeri (Düğüm 6, %4.914), sekiz alındığı durumdan (Düğüm 7, %5.363) daha düşük bulunmuştur. Gizli katman sayısı BR algoritmasında MAPE değerinin azalmasında önemli bir faktör olarak belirlenmiş, katman sayısının iki alınması durumunda MAPE değeri (Düğüm 16, %4.914) bir katmanda (Düğüm 15, %4.962) ve üç katmanda (Düğüm 17, %6.214) olarak tespit edilmiştir.

GDX algoritmasının hatayı azaltma başarısını en fazla etkileyen faktör döngü sayısı olmaktadır ve döngü sayısının düşük tutulması durumunda yüksek MAPE değerleri (Düğüm 9 ve 10) elde edilmektedir. Döngü sayısı 1000, öğrenme oranı 0.1 ve gradyan değerinin 10^{-6} ile 10^{-5} alınması durumunda (Düğüm 8), GDX algoritmasının katman sayısı, nöron sayısı ve normalizasyondan etkilenmediği, bu

grupta GDX algoritmasıyla en düşük MAPE değerinin (%5.274) elde edildiği EK 6a'dan görülmektedir.

EK 6b incelendiğinde RMSE değerleri için, EK 6c incelendiğinde ise R^2 değerleri için MAPE değerlerine benzer sonuçların elde edildiği görülmektedir. EK 6b'de en düşük ortalama RMSE değerinin LM algoritmasından (Düğüm 5, 35.418 kg) EK 6c'de en yüksek ortalamaya sahip belirleme katsayısının ise (Düğüm 5, $R^2=0.934$) yine LM algoritması ile elde edildiği görülmektedir. EK 5d incelendiğinde LM algoritmasının döngü sayısı 100 ile sınırlandırıldığında (Düğüm 6, 0.879sn) GDX algoritmasının ortalama süre değerine (Düğüm 3, 0.642sn) yaklaştığı görülmektedir. BR algoritmasında döngü sayısı etkili olmazken döngü sayısının genel ortalaması (Düğüm 3.792sn) diğer iki algoritmadan daha yüksek olduğu görülmektedir. EK 6e incelendiğinde LM algoritmasının en düşük RMSE değerini yakaladığı en iyi döngü sayısı bakımından (24 döngü) diğer iki algoritmaya göre (BR=350 döngü ve GDX=610 döngü) çok daha üstün olduğu söylenebilir. Nöron sayısının 5 ve 8 olarak alınması LM algoritmasında çok daha az döngü ile (Düğüm 5, 17 döngü) eğitimin tamamlanmasına neden olmuştur.

4.5 Makine Öğrenmesi Yaklaşımlarının Performans Değerlendirme Ölçütleri.

Tez çalışması kapsamında yapay sinir ağları, çok değişkenli uyarlanabilir regresyon eğrileri, çoklu doğrusal regresyon ve Ki-kare otomatik etkileşim belirleme analizleri kullanılarak yapılan tahminlere ait performans değerlendirme ölçütleri Tablo 4.80'de verilmiştir.

Tablo 4.80. Performans değerlendirme ölçütleri bakımından makine öğrenmesi yaklaşımları.

MAKİNE ÖĞRENMESİ YAKLAŞIMI	MAPE (%)	RMSE (Kg)	R^2
Yapay Sinir Ağları	4.15	31.63	0.9467
Çok Değişkenli Uyarlanabilir Regresyon Eğrileri	4.75	32.95	0.9334
Çoklu Doğrusal Regresyon	5.02	37.06	0.9266
Ki-Kare Otomatik Etkileşim Belirleme	5.36	38.42	0.9211

Makine öğrenmesi yaklaşımlarından yapay sinir ağlarında en düşük MAPE (%4.15) ve RMSE değeri (31.63kg) ile en yüksek belirleme katsayısı ($R^2=0.9467$) olarak belirlenmiştir. Yapay sinir ağlarında elde edilen en düşük MAPE değeri, verilerin normalize edildiği, %70 eğitim, %15 test ve %15 doğrulama setine ayrıldığı durumda, Levenberg-Marquardt algoritması kullanılarak, iki gizli katmanda sekiz nöronun kullanıldığı yapıda (2-8-8-1), MU değeri 0.01, GD değeri 10^{-7} , LR değerinin 0.01 olarak alındığı ve döngü sayısının 500 ile sınırlandırıldığı eğitimden elde edilmiştir. Analizlerin yapılmasında MATLAB paket programı kullanılmıştır.

Çok değişkenli uyarlanabilir regresyon eğrileri yönteminde Salford Predictive Modeler 7.0 programı demo versiyonunda ve *penalty* değeri üç olarak alınmış, MAPE değeri %4.75, RMSE değeri 32.95kg ve R^2 değeri 0.9334 olarak belirlenmiştir.

Çoklu doğrusal regresyon analizinde çalışmada çok fazla bağımsız değişken kullanılmadığından değişken eleme ve ekleme için herhangi bir yöntem kullanılmamış, SPSS v25 programı kullanılarak elde edilen denklem yardımıyla yapılan tahminler ve gerçek canlı ağırlıklar arasındaki MAPE değeri %5.02, RMSE değeri 37.06kg ve R^2 değeri 0.9266 olarak hesaplanmıştır.

Son olarak SPSS v25 paket programı kullanılarak uygulanan CHAID yönteminde bütün analizlerde Ebeveyn: Çocuk (Parent Node: Child Node) düğüm oranı 50:25 olarak ayarlanmış, ağaç derinliği 7 olarak alınmış ve MAPE değeri %5.36, RMSE değeri 38.42kg ve R^2 değeri 0.9211 olarak tespit edilmiştir.

5. TARTIŞMA

Sığırlarda canlı ağırlığın tahmin edilmesiyle ilgili ilk çalışmalara 20. yüzyılın başlarında rastlanmaktadır. Ölçüm cetveli ya da mezura gibi ölçüm araçlarıyla hayvanlardan alınan göğüs çevresi, vücut uzunluğu, vücut derinliği, sağrı yüksekliği, kalça genişliği ve karın çevresi gibi çeşitli vücut ölçümlerini ve çoklu doğrusal regresyon gibi istatistiksel yöntemleri kullanarak bilim insanları hayvanlarda canlı ağırlığı tahmin etmeye çalışmıştır. 21. yüzyılın başlarında ise bu yöntemler yerini hayvanlarda çeşitli vücut özelliklerinin ve canlı ağırlığın görüntüleme ile tahmin edildiği çalışmalara bırakmıştır.

Bu tez çalışmasında klasik uygulamanın aksine et sığırlarına rahatsızlık vermeksizin üç boyutlu bir görüntüleme aracı olan structure sensör kullanılarak uzaktan elde edilen görüntülerinden, görüntü işleme yöntemleri yardımıyla hesaplanan vücut alanı ve yükseklik bilgilerini kullanarak et sığırlarının canlı ağırlıkları makine öğrenmesi yaklaşımları kullanılarak tahminlenmiştir. Literatürde üç boyutlu görüntülerle, görüntü işleme ve makine öğrenmesi yaklaşımlarını birlikte kullanarak sığırlarda canlı ağırlığın tahmin edildiği çalışmalara rastlanmamıştır. Bu nedenle tartışma kısmında geleneksel yöntemlerle canlı ağırlığın tahmin edildiği çalışmalar, görüntü işleme yöntemlerini kullanmaksızın makine öğrenmesinde kullanılan bazı yaklaşımlarla yapılmış çalışmalar, görüntü işleme yöntemiyle yapılan çalışmalar ile çalışma bulguları karşılaştırılacaktır. Ayrıca yapay sinir ağlarında kullanılan algoritmaların etkinliği bu alanda yapılmış çalışmalarla karşılaştırılacaktır.

Tez çalışmasında et sığırlarının canlı ağırlıkları ile sığır görüntülerinden tahmin edilen vücut alanları arasındaki belirleme katsayısı $R^2=0.919$, vücut alanı ve yükseklik birlikte kullanıldığında ise $R^2=0.926$ olarak hesaplanmıştır. Bu değer, çeşitli vücut ölçümlerinin elle ölçüldüğü, çoklu doğrusal regresyon yöntemi yardımıyla canlı ağırlığın geleneksel yöntemlerle tahmin edildiği Tüzemen et al. (1995), Koç ve Akman (2007) ve Gruber et al. (2018) tarafından yapılan çalışmalarda elde edilen sırasıyla $R^2=0.812$, $R^2=0.848$ ve $R^2=0.821$ belirleme katsayılarından yüksektir.

Tez çalışmasında sığırların üç boyutlu görüntüleri ile canlı ağırlıkları arasında hesaplanan $R^2=0.926$ değeri, Brunton and Salisbury (1946) tarafından çeşitli vücut özelliklerinin elle ölçülerek canlı ağırlıkların çoklu doğrusal regresyon analizi yardımıyla tahmin edildiği çalışmada, Jersey ve Siyah Alacalar birlikte ele alındığında hesaplanan belirleme katsayısı $R^2=0.953$ 'ten daha düşüktür. Ancak aynı çalışmada Siyah Alacalar için hesaplanan $R^2=0.910$ ve Jersey ırkı için hesaplanan $R^2=0.918$ değeri ve Bozkurt (2006) tarafından yapılan çalışmada polinomiyal regresyon analizi yardımıyla göğüs çevresi ve vücut uzunluğu kullanılarak elde edilen $R^2=0.932$ değeri ile yakındır.

Çalışma kapsamında sığır görüntülerinden tahmin edilen vücut alanı ve yükseklik değerleri kullanılarak çoklu doğrusal regresyon yardımıyla belirleme katsayısı $R^2=0.926$ olarak hesaplanmıştır. Bu değer Özlütürk vd (2006) tarafından vücut ölçümlerinin elle alınarak yürütüldüğü araştırmada 15-18-24 aylık erkek esmer sığırlar için çoklu doğrusal regresyon kullanılarak klasik yöntemlerle hesaplanan göğüs çevresi ve canlı ağırlık arasındaki belirleme katsayısı değerlerinden ($R^2=0.941$, $R^2=0.924$ ve $R^2=0.924$) daha düşük ancak ergin sığırlar için hesaplanan $R^2=0.815$ değerinden daha yüksektir. 15-18-24 aylık sığırlarda aynı yaşta sığırların seçilmesi nedeniyle verinin daha homojen yapıda olmasına bağlı olarak daha yüksek belirleme katsayısı elde edilmiş olabilir.

Tez çalışması kapsamında ayrıca makine öğrenmesinde kullanılan yaklaşımlardan yapay sinir ağları (YSA), çok değişkenli uyarlanabilir regresyon eğrileri (MARS), çoklu doğrusal regresyon (ÇDR) ve Ki-kare otomatik etkileşim belirleme (CHAID) analizleri karşılaştırmalı olarak incelenmiştir. Bu yöntemlerle yapılan tahminlerden elde edilen MAPE değerleri sırasıyla %4.15, %4.75, %5.02 ve %5.36 olarak, RMSE değerleri sırasıyla 31.63kg, 32.95kg, 37.06kg ve 38.42kg olarak, belirleme katsayıları ise yine aynı sırayla 0.9467, 0.9334, 0.9266 ve 0.9211 olarak hesaplanmıştır. Çalışma bulguları, Ali et al. (2015) tarafından koyunların canlı ağırlıklarının tahmin edilmesi amacıyla yürütülen çalışma bulgularından farklılık göstermektedir. Araştırmacılar Ki-kare otomatik etkileşim belirleme analizi ile yapılan tahminin belirleme katsayısının $R^2=0.8381$, yapay sinir ağları ile yapılan tahminden $R^2=0.8154$ daha yüksek olduğunu tespit etmişlerdir. Oysa araştırma bulgularımız Ki-kare otomatik etkileşim belirleme ile yapılan tahminin yapay sinir

ağları ile yapılan tahminden daha düşük belirleme katsayısı elde edildiğini göstermektedir. Araştırmacıların Ki-kare otomatik etkileşim belirleme analizi ile daha yüksek belirleme katsayısı elde etmesinin sebebi çalışmalarında kullanılan bağımsız değişken sayısının fazla olması olabilir.

Çeşitli vücut özelliklerinin elle ölçülerek koyunlarda canlı ağırlığın tahmin edildiği Karadaş et al. (2017) tarafından yürütülen çalışmada Ki-kare otomatik etkileşim belirleme analizinin tahmin başarısının, yapay sinir ağlarından daha yüksek olduğu belirtilmiştir. Araştırmacıların iki yöntem için tespit ettikleri RMSE değerleri birbirine yakın olsa da sığır canlı ağırlıklarının tahmin edildiği bu tez çalışmasında ($RMSE_{CHAID}=38.42\text{kg}$, $RMSE_{YSA}=31.63\text{kg}$) RMSE değerlerinin yapay sinir ağlarında daha düşük olduğu tespit edilmiştir. Yapılan çalışmada Ki kare otomatik etkileşim belirleme analizinin tahmin başarısının daha yüksek olduğu bilgisi çalışma bulgularımızla örtüşmemektedir.

Çelik et al. (2017) tarafından koyun ağırlıklarını tahmin etmek için yapay sinir ağları (YSA), çok değişkenli uyarlanabilir regresyon eğrileri (MARS) ve Ki kare otomatik etkileşim belirleme analizleri (CHAID) kullanılarak yürütülen çalışmada MAPE değerleri bakımından yaklaşımları iyiden kötüye doğru CHAID ($MAPE=\%5.797$) < MARS ($MAPE=\%7.145$) < YSA ($MAPE=\%7.856$) olarak sıralamışlardır, bu bilgi de sığırlarda canlı ağırlığın aynı yöntemlerle tahmin edildiği çalışma bulgularımızdan ($MAPE_{CHAID}=\%5.36$ > $MAPE_{MARS}=\%4.75$ > $MAPE_{YSA}=\%4.15$) farklılık göstermektedir.

Çalışmamızda yapay sinir ağları ile yapılan tahmin sonuçları çoklu doğrusal regresyon analizinden daha iyi bulunmuş ve yapay sinir ağlarının, parametrik bir test olması nedeniyle çeşitli varsayımları bulunan çoklu doğrusal regresyon analizine alternatif olarak kullanılabileceği tespit edilmiştir.

Grzesiak et al. (2003) süt verimini tahmin etmek için yürüttükleri araştırmada yapay sinir ağları ile tahminden elde ettikleri RMSE değerini, çoklu doğrusal regresyon analizi ile tahminde elde ettikleri RMSE değerinden daha düşük olarak belirlemişlerdir. Sığırlarda canlı ağırlığın tahmin edildiği çalışma bulgularımız yapay sinir ağları ile tahminde RMSE değerinin 31.63kg, çoklu doğrusal regresyon

analizinde bu değerin 37.06kg olduğunu ve yapay sinir ağları ile daha düşük RMSE değeri ile tahmin yapılabildiğini göstermekte, Grzesiak et al. (2003) tarafından sunulan bulguları destekler niteliktedir.

Araştırma bulgularımız, kanatlı beslemesi alanında çalışan Perai et al. (2010) tarafından yürütülen, bu tez çalışmasına benzer şekilde yapay sinir ağları ile çoklu doğrusal regresyon analizinin karşılaştırıldığı araştırmanın bulgularını (yapay sinir ağlarıyla $R^2=0.94$, çoklu doğrusal regresyonla $R^2=0.38$) desteklemekle birlikte çalışmamızda yöntemlerle elde edilen belirleme katsayıları ($R^2_{YSA}=0.9467$ ve $R^2_{ÇDR}=0.9266$) arasında araştırmacıların elde ettiği kadar yüksek farklılık bulunmamıştır.

Takma vd. (2012) sığırlarda süt verimini tahmin ettikleri çalışmada bu tez çalışmasına benzer şekilde yapay sinir ağları ($R^2=0.85$) ile çoklu doğrusal regresyon analizini ($R^2=0.75$) karşılaştırmışlardır. Araştırmacılar bu tez çalışmasını destekler nitelikte yapay sinir ağlarının ($R^2=0.9467$) çoklu doğrusal regresyondan ($R^2=0.9266$) daha iyi sonuç verdiğini bildirmişlerdir.

Dongre et al. (2012) sığırlarda ilk laktasyon süt verimini tahmin etmek amacıyla bu tez çalışmasına benzer şekilde yapay sinir ağları ve çoklu doğrusal regresyon analizi yöntemlerinin tahmin başarısını karşılaştırmışlardır. Araştırmacıların yapay sinir ağlarının ($R^2=0.86$) çoklu doğrusal regresyon ($R^2=0.85$) analizinden daha iyi sonuç verdiğini belirttiği bulguları çalışma bulgularımızı ($R^2_{YSA}=0.9467$ ve $R^2_{ÇDR}=0.9266$) destekler niteliktedir.

Tez çalışmasına benzer şekilde görüntü işleme yöntemleri kullanılarak hayvancılıkta canlı ağırlığın tahmin edildiği çalışmalardan Kmet et al. (2000) tarafından elde edilen $R^2=0.490$ değerinden, Stajko et al. (2008) tarafından hesaplanan $R^2=0.548$ ve Pradana et al. (2016) tarafından belirlenen $R^2=0.7321$ değerinden, Goyache et al. (2001) tarafından hesaplanan 0.325 ile 0.810 R^2 değerlerinden, Özkaya (2006) tarafından tesbit edilen $R^2=0.81$ değerinden ve Önal (2011) tarafından elde edilen $R^2=0.921$ değerinden bu tez çalışmasında daha yüksek belirleme katsayısı ($R^2=0.9467$) elde edilmiştir.

Bununla birlikte piliçlerde görüntü işleme yöntemlerini kullanılarak canlı ağırlığın çoklu doğrusal regresyon analizi yardımıyla tahmin edildiği Mollah et al. (2010) tarafından yürütülen çalışmada hesaplanan $R^2=0.999$ değerinden daha düşük belirleme katsayısı elde edilmiştir.

Taşdemir (2010) tarafından sığır görüntülerinden canlı ağırlığın bulanık mantık ve regresyon analizi yardımıyla tahmin edildiği çalışmada elde edilen $R^2=0.981$ değeri çalışma bulgularımızda elde ettiğimiz $R^2_{\text{ÇDR}}=0.9266$ değerinden daha yüksektir.

Kashiha tarafından yürütülen domuzlarda görüntü işleme yöntemleri ve çeşitli matematiksel modeller kullanılarak elde edilen $R^2=0.975$ değerinden çalışmamızda daha düşük belirleme katsayısı ($R^2=0.9467$) hesaplanmıştır.

Tez çalışmasında Kongsro (2014) tarafından domuz görüntülerinden canlı ağırlığın tahmin edildiği çalışmada elde edilen $R^2=0.99$ değerinden daha düşük belirleme katsayısı elde edilmesine karşın benzer MAPE (%4-%5) değerleri elde edilmiştir.

Görüntü işleme ve yapay sinir ağlarının birlikte hayvancılıkta kullanıldığı sadece dört çalışmaya rastlanmış bunlardan Khojastehkey et al. (2015) tarafından yürütülen çalışmada kuzuların post desenine göre sınıflandırılması yapılmıştır. İkinci çalışmada ise Wongsriworaphon et al. (2015) domuzların canlı ağırlıklarını yapay sinir ağları yardımıyla %2.94'lük bir hata ile tahmin etmiştir. Bu değer çalışmamızda elde edilen %4.15'lik hata değerinden biraz düşük bir değerdir.

Tez çalışmasına benzer şekilde yapay sinir ağları ve görüntü işleme yöntemlerini birlikte kullanan çalışmalardan üçüncüsü Bhatt et al. (2018) tarafından yürütülmüştür. Koyunların görüntülerinden canlı ağırlıklarının tahmin edildiği bu çalışmada elde edilen $R^2=0.81$ değerinden çalışma bulgularımızda elde edilen $R^2=0.9467$ değeri daha yüksektir.

Son olarak Amraei et al. (2017) tarafından etlik piliçlerin canlı ağırlıklarının görüntü işleme yöntemleri ve yapay sinir ağlarının birlikte kullanılarak tahmin

edildiği çalışmada elde edilen $R^2=0.984$ değeri çalışmamızda elde edilen $R^2=0.9467$ değerinden bir miktar yüksektir. Araştırmacılar çalışma bulgularında Bayesian Regularization algoritmasının Levenberg-Marquardt algoritmasından daha başarılı sonuçlar ürettiğini belirtmektedir. Ancak çalışmamızda Levenberg-Marquardt algoritması en yüksek doğruluğa sahip eğitim algoritması olarak belirlenmiştir. Araştırmacılar veri setini test ve eğitim seti olarak iki alt sete ayırmıştır. Çalışmamızda Levenberg-Marquardt algoritmasında doğrulama seti olmadığına başarının azaldığı belirlenmiştir. Çalışmada Bayesian Regularization algoritmasının daha iyi sonuç vermesinin nedeni araştırmacıların doğrulama seti kullanmaması olabilir.

Literatürde hayvancılık dışındaki alanlarda çok değişkenli uyarlanabilir regresyon eğrileri (MARS) ve yapay sinir ağlarının (YSA) karşılaştırmalı olarak incelendiği çalışmalardan Zhag and Goh (2016) tarafından yürütülen araştırmanın bulguları (yapay sinir ağları kullanıldığında $R^2=0.970$ ve çok değişkenli uyarlanabilir regresyon eğrileri kullanıldığında $R^2=0.957$) çalışma bulgularımıza paralellik göstermektedir. Araştırmacı yapay sinir ağlarının çok değişkenli uyarlanabilir regresyon eğrilerinden biraz daha yüksek doğruluğa sahip olduğunu belirtmiştir.

Tayyebi and Pijanowski (2014) araştırmalarında çalışma bulgularımıza benzer şekilde yapay sinir ağlarının çok değişkenli uyarlanabilir regresyon eğrilerinden daha yüksek doğruluk sağladığını bildirmişlerdir. Bunun dışında, De veaux et al. (1993) tarafından çok değişkenli uyarlanabilir regresyon eğrilerinin çoğu durumda yapay sinir ağlarına göre daha yüksek doğrulukla ve daha hızlı sonuçlar ürettiğini belirttiği çalışma bulgularıyla araştırma bulgularımız örtüşmemektedir.

Yapay sinir ağlarında kullanılan 3 farklı öğrenme algoritmasının karşılaştırıldığı çalışmamızda Levenberg-Marquardt (LM) algoritması ile %4.56 MAPE değeri ile Bayesian Regularization (BR) algoritması kullanıldığında %4.70 ve değişken öğrenme oranlı iniş azaltma (GDX) algoritması kullanıldığında %4.75 MAPE değerleri ile tahminler yapılmıştır. Çalışma bulgularımız Eren et al. (2016) ve Rahimi et al. (2018) tarafından önerilen Levenberg-Marquardt algoritmasının

hatayı azaltma bakımından Bayesian Regularization algoritmasından daha iyi sonuçlar ürettiğine dair bulgularla benzerdir.

Moosavizadeh et al. (2011), Bui et al. (2012), Dongre et al. (2012), Akkol vd. (2015), Kayri (2016) ve Mukherjee and Rajanikanth (2019), Bayesian Regularization algoritmasının daha yüksek R^2 değerleri ile tahminler ürettiğini bildirmektedir. Bu çalışmalarda verinin ayrılma şekli incelendiğinde veri genellikle test ve eğitim setine ayrılmıştır. Bu durum çalışmamızda çapraz doğrulama ile verinin ayrıldığı durumdaki sonuçlara benzerlik teşkil etmektedir. Zira Levenberg-Marquardt algoritması kullanıldığında, doğrulama setinin olmayışı nedeniyle Bayesian Regularization algoritmasına göre daha yüksek hatalar ile tahmin yapılmıştır. Bayesian Regularization algoritmasında zaten doğrulama seti bulunmamaktadır. Bu nedenle Bayesian Regularization ve Levenberg-Marquardt algoritmaları açısından doğru bir karşılaştırma yapabilmek için Levenberg-Marquardt algoritmasının kullanımında verinin belirli oranlarda eğitim, test ve doğrulama setlerine ayrılması gerekmektedir.

Tez çalışmasında Levenberg-Marquardt algoritması için 3 nöron kullanıldığı durumda 3 gizli katman, 5 ve 8 nöron kullanıldığı durumda 2 gizli katman alınması önerilmektedir. Benzer şekilde Dongre et al. (2012) tarafından yürütülen araştırma sonucunda 2 gizli katman, 3 ve 5 nöron alınması durumunda Scaled Conjugate Gradient (SCG), Bayesian Regularization ve Levenberg-Marquardt algoritmalarında en iyi tahmin sonucuna ulaşıldığı bildirilmektedir.

Cozler et al. (2019) süt sığırlarında çeşitli vücut özelliklerini en yüksek $R^2=0.79$ belirleme katsayısı değeri elde ederek tahmin etmişlerdir. Çalışmada lazer projektör ile desteklenen 5 adet kamera kullanılmış ve araştırmacılar bu cihazlarla yapılan ölçümün yaklaşık olarak 15 dakika sürdüğünü bildirmişlerdir. Çalışmamızda sığır vücut özellikleri tahmin edilmese de kullanılan görüntü işleme yöntemi ve yapay sinir ağları ile hayvan alanının hesaplanması ve canlı ağırlığın tahmin edilmesi işlemleri bir dakikanın altında bir zamanda tamamlanmıştır. Bu bakımdan gelecekte yapılacak benzer bir çalışmada üç boyutlu iki kameranın (yandan ve üstten) kullanılması ve çalışmamızda önerdiğimiz yöntemlerin seçilmesi tahmin başarısını artırabilir ve çok daha kısa sürede sonuç verebilir.

Amraei et al. (2018) çeşitli görüntü işleme yöntemlerini kullanarak etlik piliçlerde canlı ağırlığı $R^2=0.98$ gibi yüksek bir belirleme katsayısı ile tahmin etmişlerdir. Bu değer çalışma bulgularımızdan ($R^2=0.9467$) yüksek olmasına karşın çalışmada görüntüden pilicin ayrılması için yapılan işlemler (histogram eşikleme, görüntü bölütleme vs.), üç boyutlu bir görüntüleme aracı kullanılan çalışmamızda görüntünün ayıklanması için yapılan işlemlerden çok daha fazla adım gerektirmektedir.

Yan et al. (2019) tarafından Yak (Tibet) sığırlarının iki boyutlu görüntülerinden canlı ağırlıklarının tahmin edildiği çalışmada elde edilen $R^2=0.972$ değeri çalışma bulgularımızdan ($R^2=0.9467$) bir miktar yüksektir. Yine çalışmada belirlenen $MAPE=\%3.19$ değeri çalışmamızda elde edilen $MAPE=\%4.16$ değerinden biraz düşük ve $RMSE=7.52\text{kg}$ değeri çalışma bulgularımızdan ($RMSE=31.63\text{kg}$) oldukça düşüktür. Yak (Tibet) sığırlarında ağırlık ortalama olarak yaklaşık 260-300kg civarındadır. Değişim genişliğinin düşük olması ve çalışmada sadece Yak sığırının kullanılması canlı ağırlık için yapılan tahminin doğruluğunu artırmış olabilir.

Shi et al. (2016) domuzlarda cidago yüksekliği, vücut uzunluğu ve canlı ağırlığı tahmin ettikleri araştırmada, vücut özellikleri için yapılan tahminlerle ölçüm değerleri arasındaki belirleme katsayılarını sırasıyla $R^2=0.94$, $R^2=0.99$ ve $R^2=0.9931$ olarak tespit etmişlerdir. Canlı ağırlık için araştırmacılar tarafından tespit edilen $R^2=0.9931$ değeri çalışma bulgularımızdan ($R^2=0.9467$) oldukça yüksek olmasına karşın, araştırma bulgularımız sığırlarda da yüksekliğin, görüntüdeki derinlik bilgisini kullanarak yüksek doğrulukla ($R^2=0.9966$) ölçülebildiğini göstermekte ve araştırmacılar tarafından tespit edilen $R^2=0.99$ değerini destekler niteliktedir.

Hayvancılık dışındaki alanlarda yapılan bazı çalışma sonuçları da çok değişkenli uyarlanabilir regresyon eğrileri yönteminin çoklu doğrusal regresyon analizinden daha yüksek doğrulukla tahminde bulunduğu sonucuna ulaşılan çalışmamızın bulgularını desteklemektedir.

Mansouri et al. (2016) çoklu doğrusal regresyon (ÇDR) ve çok değişkenli uyarlanabilir regresyon eğrileri (MARS) yöntemlerini karşılaştırmalı olarak incelediği araştırmada MARS yöntemi ile elde edilen $RMSE=0.127$ değerinin ÇDR analizinden elde edilen $RMSE=0.221$ değerinden düşük olduğunu belirtmiştir. Çalışma bulgularımızda MARS analizinde çoklu doğrusal regresyon analizine göre daha düşük MAPE değerleri elde edildiğini göstermektedir. Araştırma sonuçları çalışma bulgularımızla paraleldir.

Goh and Zhang (2013) çok değişkenli uyarlanabilir regresyon eğrileri yönteminin ($R^2=0.8559$) çoklu doğrusal regresyon analizinden ($R^2=0.8358$) daha yüksek doğruluğa sahip olduğunu bildirmiştir. Bu çalışmanın sonuçları da bulgularımızla örtüşmektedir.

Lu et al. (2009) çok değişkenli uyarlanabilir regresyon eğrileri (MARS), destek vektörü regresyon (SVR), geri yayımlı (BP) yapay sinir ağı ve çoklu doğrusal regresyon analizi yöntemlerini karşılaştırmalı olarak inceledikleri araştırmada MAPE değerlerini sırasıyla %1.15, %2.18, %3.55 ve %4.25 olarak tespit etmişlerdir. Çalışmamızda çok değişkenli uyarlanabilir regresyon eğrileri yöntemiyle yapılan tahminlerde çoklu doğrusal regresyon analizi yöntemine göre daha yüksek başarı elde edilmiştir. Ancak bulgularımızda iki yöntem arasında bu kadar yüksek bir farklılık oluşmamış, ayrıca yapay sinir ağları çok değişkenli uyarlanabilir regresyon eğrileri yönteminden daha düşük MAPE değeri ile tahmin vermiştir. Bu anlamda çalışma bulgularımızla örtüşmemektedir.

Emamgolizadeh et al. (2015) hidroloji alanında yapmış olduğu çalışmada yapay sinir ağları, çok değişkenli uyarlanabilir regresyon eğrileri ve çoklu doğrusal regresyonu karşılaştırmış ve sonuç olarak çalışma bulgularımıza çok benzer şu sonuçları elde etmişlerdir. Çalışmalarında yapay sinir ağları ile yapılan tahminde $RMSE=0.252$, $R^2=0.892$, MARS yönteminde $RMSE=0.318$, $R^2=0.864$ ve çoklu doğrusal regresyon analizinde $RMSE=0.408$, $R^2=0.768$ olarak tespit etmişlerdir. Araştırma bulgularımız da üç yöntemin tahmin doğruluğu bakımından aynı sıralamada olduğunu göstermektedir.

Kiran and Ravi (2007) yapay sinir ağıları, çok değişkenli uyarlanabilir regresyon eğrileri ve çoklu doğrusal regresyon yöntemlerini beş farklı durum için, normalleştirilmiş hata kareler ortalamasının karekökü (Normalized Root Mean Square Error - NRMSE) performans değerlendirme ölçütünü kullanarak karşılaştırmışlardır. İlk durumda çok değişkenli uyarlanabilir regresyon eğrileri yöntemi ile en düşük NRMSE=0.170584 değerini elde eden araştırmacılar geri yayımlı yapay sinir ağlarında (BP) bu değeri NRMSE= 0.171375 olarak, çoklu doğrusal regresyonda (ÇDR) ise NRMSE=0.171448 olarak tespit etmişlerdir. Çalışmada incelenen diğer dört durumda ise geri yayımlı yapay sinir ağları ile elde edilen NRMSE değerlerinin (sırasıyla 0.166086, 0.151429, 0.144949 ve 0.145541) MARS yönteminden elde edilen NRMSE değerlerinden (sırasıyla 0.17091, 0.161343, 0.154821 ve 0.15267) daha düşük olduğunu bildirmişlerdir. Çalışmada ayrıca yapay sinir ağlarından elde edilen NRMSE değerlerinin çoklu doğrusal regresyon yönteminden elde edilen NRMSE değerlerinden (sırasıyla 0.167776, 0.156537, 0.151152 ve 0.147881) daha düşük olduğu belirtilmiştir. Çalışma sonuçları ilk durum hariç çalışma bulgularımızla benzerlik göstermektedir.

6. SONUÇ VE ÖNERİLER

Sığır işletmelerinde canlı ağırlığın bilinmesi işletme karlılığı açısından elzem konular arasındadır. Klasik yöntemler kullanarak canlı ağırlığı tespit veya tahmin etmek zor, zahmetli ve hayvanlarda strese sebep veren bir süreçtir. Bu sebeple uzaktan canlı ağırlığın tahmin edildiği, böylece insan yaralanma riskinin de minimize edildiği, makine öğrenmesi ve görüntü işleme yöntemlerini kullanarak yürütülen bu çalışmanın yetiştiricilere büyük avantajlar sağlayacağı düşünülmektedir. Çalışma bulguları yeni yöntemle, klasik yöntemlerle yapılan tahminlerden daha az vücut özelliği kullanarak daha yüksek doğrulukla tahmin yapılabileceğini göstermektedir. Gelecekte bu verilerle eğitilmiş yapay sinir ağlarını kullanarak geliştirilen yazılımlar sayesinde, üç boyutlu bir kameraya sahip, küçük bir cihaz yardımıyla sığırların canlı ağırlığını tahmin etmek mümkün olabilecektir. Bu cihaz ile elde edilen veriler kablosuz olarak bir bilgisayara aktarıldığında hayvanların canlı ağırlıkları kaydedilecek ve bu şekilde canlı ağırlık takibi yapan sığircılık işletmelerinde daha efektif üretim yapılabilecektir.

Hayvancılıkta YSA kullanımının çoklu doğrusal regresyon analizi, ayırma analizi, kümeleme analizi gibi geleneksel istatistiksel yöntemlerden daha iyi sonuçlar verdiği birçok çalışmada belirtilmiş, et sığırlarında canlı ağırlığın tahminlendiği bu çalışmada da YSA ile daha düşük tahmin hataları elde edilmiştir.

Yapay sinir ağlarında, çoklu doğrusal regresyonda olduğu gibi tek bir modelle canlı ağırlığı tahmin etmek pek mümkün olmamaktadır. En uygun ağ yapısının oluşturulması için çeşitli denemeler yapılması gerekmektedir. Bu da yapay sinir ağlarının dezavantajlarından biridir. Bununla birlikte daha önce benzer verilerle yapılmış çalışmalardaki nöron ve katman sayıları ile parametreler dikkate alınırsa ağın öğrenmesi hızlı bir şekilde gerçekleştirilebilir. Bu çalışmada YSA'da kullanılan ağ parametrelerinin etkinliği ile nöron ve katman sayılarının hatayı azaltmadaki etkisini incelemek amacıyla pek çok eğitim durumu denenmiştir. Bu nedenle çok fazla eğitim kombinasyonu ortaya çıkmış ve analiz sayısı artmıştır. Gerçekte bir çalışmanın yürütülmesinde bu kadar denemenin yapılmasına ihtiyaç duyulmamaktadır. Birkaç deneme yapılarak ya da varsayılan MATLAB

parametreleri kullanılarak kabul edilebilir bir hata seviyesinde tahminler elde edilebilmektedir.

Çoklu doğrusal regresyon ve çok değişkenli uyarlanabilir regresyon eğrileri yöntemlerinde bir tahmin denkleminin olması bu yöntemleri diğer yöntemlere nazaran avantajlı kılmaktadır. Ancak çoklu doğrusal regresyon analizi dışındaki yöntemlerin ise parametrik testlerin gerektirdiği bazı önşart ve varsayımları gerektirmemesi yönüyle avantajı büyüktür. Özellikle birçok alanda karşılaşılan bağımsız değişkenler arasındaki çoklu doğrusal bağlantı (multicollinearity) problemi, çoklu doğrusal regresyon analizinin kullanımını sınırlandırmaktadır. Bunun dışında araştırmacılar tarafından varsayımlar kontrol edilmeden ve ön şartlar sağlanmadan analizin yapılması durumunda sonuçların geçerliliği de tartışmalı olmaktadır.

Araştırmada kullanılan CHAID analizi diğer yöntemlerin aksine sunduğu ağaç diyagramı sayesinde sonuçların yorumlanmasına büyük kolaylık sağlamaktadır. Yine araştırmada kullanılan bir diğer yaklaşım olan çok değişkenli uyarlanabilir regresyon eğrileri ise temel fonksiyonları kullanarak çoklu doğrusal regresyon yönteminden daha esnek bir model oluşturması yönüyle avantaj sağlamaktadır.

Bu çalışmada gelecekte yapılması planlanan benzer çalışmalara yol göstermek amacıyla yapay sinir ağlarında üç farklı eğitim algoritması için üç farklı gizli katman ve nöron kullanıldığında hangi durumda iyi sonuç alındığı, veri ayırma yöntemlerinden algoritmaların nasıl etkilendiği ile ilgili bilgiler verilmiştir. LM ve GDX algoritmalarında veriler standartlaştırıldığı zaman daha düşük MAPE değerleriyle tahmin yapılabileceği, eğitim süreleri bakımından da önemli bir azalma görüldüğü saptanmıştır. LM algoritması için 3 nöron kullanıldığı durumda 3 katman, 5 ve 8 nöron kullanıldığı durumda 2 katman alınması önerilmektedir. Çalışmada en az döngü ile eğitimi tamamlayan algoritma LM algoritması olarak belirlenmiştir. Döngü sayısı ve maksimum başarısız döngü sayısı yüksek tutulduğunda GDX algoritmasının çok daha kısa sürelerde eğitimi tamamladığı belirlenmiştir. Hataların azaltılması bakımından en etkin algoritmanın LM

algoritması olduđu tespit edilmiştir. Çalışmadan elde edilen bu bilgilerin de araştırmacılara yol göstereceğı düşünülmektedir.

Gelecekte benzer çalışma yapmayı planlayan araştırmacılar görüntüleme yapılırken hayvanları aynı pozisyonda sabitleyerek görüntü almaya odaklanmalıdır. Sığırların bir yürüme bandında görüntülenmesi vücut uzunluğu, kalça genişliği gibi birkaç parametrenin de otomatik olarak hesaplanmasına olanak verebilir. İkinci bir kamera kullanılarak göğüs derinliğinin belirlenmesi de yine daha yüksek doğrulukla tahmin yapılmasını sağlayacaktır. Ayrıca aynı ırktan olan sığırların çalışmada kullanılması durumunda hatanın bir miktar azalacağı düşünülmektedir. Son olarak kullanımı son birkaç yılda yaygınlaşan derin öğrenme yöntemlerinin kullanılması da araştırmacılara tavsiye edilmektedir.

KAYNAKLAR DİZİNİ

- Adamczyk, K., Molenda, K., Szarek, J. A. N. and Skrzyński, G., 2005,** Prediction of bulls' slaughter value from growth data using artificial neural network. *Journal of Central European Agriculture*, 6(2), 133-142pp.
- Akıllı, A. and Atıl, H., 2014,** Süt sığırcılığında yapay zeka teknolojisi: Bulanık mantık ve yapay sinir ağları, *Hayvansal Üretim*, 55(1): 39-45 s.
- Akkol, S., Akıllı, A. ve Cemal, İ., 2017,** Comparison of Artificial Neural Network and Multiple Linear Regression for Prediction of Live Weight in Hair Goats, *Yyü Tar Bil Derg:* 27(1): 21-29pp.
- Albayrak, A. S., 2012,** Çoklu doğrusal bağlantı halinde enküçük kareler tekniğinin alternatifi yanlı tahmin teknikleri ve bir uygulama. *Uluslararası Yönetim İktisat ve İşletme Dergisi*, 1(1), 105-126s.
- Albayrak, A. S. ve Yılmaz, S. K., 2009,** Veri Madenciliği: Karar Ağacı Algoritmaları Ve İmkb Verileri Üzerine Bir Uygulama, *Suleyman Demirel University Journal of Faculty of Economics & Administrative Sciences*, 14(1)s.
- Ali, M., Eyduran, E., Tariq, M.M., Tirink, C., Abbas, F., Bajwa, M.A., Baloch, M.H., Nizamani, A.H., Waheed, A. and Awan, M.A., 2015,** Comparison of artificial neural network and decision tree algorithms used for predicting live weight at post weaning period from some biometrical characteristics in Harnai sheep, *Pakistan Journal of Zoology*, 47(6)p.
- Alikhanov, D., Penchev, S., Georgieva, T., Moldajanov, A., Shynybaj, Z. and Daskalov, P., 2015,** Indirect Method for Egg Weight Measurement Using Image Processing, *International Journal of Emerging Technology and Advanced Engineering*, 5, 30-34p.
- Alpar, R., 2013,** Uygulamalı çok değişkenli istatistiksel yöntemler, Detay yayıncılık, Ankara, 888s.
- Amraei, S., Abdanan Mehdizadeh, S., and Salari, S., 2017,** Broiler weight estimation based on machine vision and artificial neural network. *British poultry science*, 58(2), 200-205pp.

KAYNAKLAR DİZİNİ (Devamı)

- Amraei, S., Mehdizadeh, S. A. and Nääs, I. D. A.,** 2018, Development of a transfer function for weight prediction of live broiler chicken using machine vision. *Engenharia Agricola*, 38(5), 776-782pp.
- Arı, A., ve Berberler, M. E.,** 2017, Yapay Sinir Ağları ile Tahmin ve Sınıflandırma Problemlerinin Çözümü İçin Arayüz Tasarımı. *Acta Infologica*, 1(2), 55-73s.
- Ataseven, B .,** 2013, Yapay Sinir Ağları İle Öngörü Modellemesi. *Öneri Dergisi*, 10 (39), 101-115s.
- Atil, H. and Akilli, A.,** 2015, *Investigation of Dairy Cattle Traits by Using Artificial Neural Networks and Cluster Analysis*. Paper presented at the HAICTA.
- Atıl, H. and Akıllı, A.,** 2016, Comparison of artificial neural network and K-means for clustering dairy cattle, *International Journal of Sustainable Agricultural Management and Informatics*, 2(1): 40-52 pp.
- Babalık, A. ve Botsalı, F. M.,** 2010, Yapay sinir ağı ve görüntü işleme teknikleri kullanarak durum buğdayının camsılığının belirlenmesi, *Teknik-Online Dergi*, 9(2): 163-174 s.
- Bahreini B, M. R., and Aslaminejad, A. A.,** 2010, A comparison of neural network and nonlinear regression predictions of sheep growth, *Journal of animal and veterinary advances*, 9.
- Bektaş, S.,** 2016, Kırşehir İlinde Yetiştirilen Esmer Ve Esmer Melezi Sığırlarda Bazı Morfolojik Özelliklerin Görüntü İşleme Yöntemi İle Belirlenmesi. Ahi Evran Üniversitesi Fen Bilimleri Enstitüsü, Kırşehir, 44s.
- Bewley, J., Peacock, A., Lewis, O., Boyce, R., Roberts, D., Coffey, M., Schutz, M.,** 2008, Potential for estimation of body condition scores in dairy cattle from digital images, *Journal of dairy science*, 91(9), 3439-3453pp.
- Bhatt, C., Hassanien, A. E., Shah, N. A., & Thik, J.,** 2018, Barqi Breed Sheep Weight Estimation based on Neural Network with Regression. *arXiv preprint arXiv:1807.10568*.

KAYNAKLAR DİZİNİ (Devamı)

- Bozağaç, C.D.**, 2014, Metamodeling atomic models in discrete event system specification (DEVS) formalism using multivariate adaptive regression splines (MARS), Ortadoğu Teknik Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 116p.
- Bozkurt, Y.**, 2006, Prediction of body weight from body size measurements in Brown Swiss feedlot cattle fed under small-scale farming conditions, *Journal of Applied Animal Research*, **29**(1): 29-32p.
- Branton, C. and Salisbury, G.**, 1946, The estimation of the weight of bulls from heart girth measurements, *Journal of dairy science*, **29**(3): 141-143pp.
- Bui, D.T., Pradhan, B., Lofman, O., Revhaug, I. and Dick, O.B.**, 2012, Landslide susceptibility assessment in the Hoa Binh province of Vietnam: a comparison of the Levenberg–Marquardt and Bayesian regularized neural networks, *Geomorphology*, **171**: 12-29pp.
- Cebeci, Z.**, 2020, R uygulamalı yeniden örnekleme teknikleri, Pagem Akademi Yayıncılık, Ankara.
- Celik, S., Eyduran, E., Karadas, K. and Tariq, M. M.**, 2017, Comparison of predictive performance of data mining algorithms in predicting body weight in Mengali rams of Pakistan. *Revista Brasileira de Zootecnia*, **46**(11), 863-872pp.
- Chou, S. M., Lee, T. S., Shao, Y. E. and Chen, I. F.**, 2004, Mining the breast cancer pattern using artificial neural networks and multivariate adaptive regression splines. *Expert systems with applications*, **27**(1), 133-142pp.
- Cozler, Y., Allain, C., Caillot, A., Delouard, J. M., Delattre, L., Luginbuhl, T. and Faverdin, P.**, 2019, High-precision scanning system for complete 3D cow body shape imaging and analysis of morphological traits. *Computers and electronics in agriculture*, **157**, 447-453pp.
- Çankaya, G., Arslan, M. H., ve Ceylan, M.**, 2013, Görüntü İşleme Ve Yapay Sinir Ağları Yöntemleri İle Betonun Basınç Dayanımının Belirlenmesi. *Selçuk Üniversitesi Mühendislik, Bilim ve Teknoloji Dergisi*, **1**(1), 1-12 s.

KAYNAKLAR DİZİNİ (Devamı)

- Çayiroğlu, İ., “İleri algoritma analizi-5 Yapay sinir ağları”, <http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.Hafta-YapaySinirAglari.pdf> (Erişim tarihi: 4 Temmuz 2019)
- Celik, S., Eydurhan, E., Karadas, K. ve Tariq, M. M., 2017, Comparison of predictive performance of data mining algorithms in predicting body weight in Mengali rams of Pakistan, *Revista Brasileira de Zootecnia*, 46(11), 863-872pp.
- Çerçi, İ., 2010 Çok değişkenli regresyon analizi (Gsm sektöründe bir uygulama) Gazi Üniversitesi Sosyal Bilimler Enstitüsü, Ankara, 153s.
- De Veaux, R. D., Psychogios, D. C., and Ungar, L. H., 1993, A comparison of two nonparametric estimation schemes: MARS and neural networks. *Computers & chemical engineering*, 17(8), 819-837pp.
- Daliakopoulos, I. N., Coulibaly, P. and Tsanis, I. K., 2005, Groundwater level forecasting using artificial neural networks, *Journal of hydrology*, 309(1-4), 229-240pp.
- Dongre, V., Gandhi, R., Singh, A. and Ruhil, A., 2012, Comparative efficiency of artificial neural networks and multiple linear regression analysis for prediction of first lactation 305-day milk yield in Sahiwal cattle, *Livestock science*, 147(1-3): 192-197pp.
- Efe, E., Bek, Y. ve Şahin, M., 2000, SPSS'de Çözümleri ile İstatistik Yöntemler II, Kahramanmaraş Sütçüimam Üniversitesi Rektörlüğü Yayın No:10, Kahramanmaraş, 280s.
- Emamgolizadeh, S., Bateni, S. M., Shahsavani, D., Ashrafi, T. and Ghorbani, H., 2015, Estimation of soil cation exchange capacity using genetic expression programming (GEP) and multivariate adaptive regression splines (MARS). *Journal of Hydrology*, 529, 1590-1600pp.

KAYNAKLAR DİZİNİ (Devamı)

- Eren, B., Yaqub, M. and Eyüpoğlu, V.**, 2016, Polimer içerikli membran verimi tahmininde yapay sinir ağları öğrenme algoritmalarının değerlendirilmesi, *Sakarya University Journal of Science*, **20**(3): 533-542s.
- Eyduran, E., Zaborski, D., Waheed, A., Celik, S., Karadas, K., & Grzesiak, W.**, 2017, Comparison of the predictive capabilities of several data mining algorithms and multiple linear regression in the prediction of body weight by means of body measurements in the indigenous beetal goat of Pakistan. *Pakistan Journal of Zoology*, 49(1)p.
- Fernandez, C., Soria, E., Sanchez-Seiquer, P., Gómez-Chova, L., Magdalena, R., Martín-Guerrero, J.D., Navarro, M. and Serrano, A.**, 2007, Weekly milk prediction on dairy goats using neural networks, *Neural Computing and Applications*, **16**(4-5): 373-381pp.
- Friedman, J. H. and Roosen, C. B.**, 1995, An introduction to multivariate adaptive regression splines. *The Annals of Statistics*, 19:1-67pp.
- Goh, A. T. and Zhang, W. G.**, 2014, An improvement to MLR model for predicting liquefaction-induced lateral spread using multivariate adaptive regression splines. *Engineering geology*, 170, 1-10pp.
- Ghosh-Dastidar, S., ve Adeli, H.**, 2009, Spiking neural networks. *International journal of neural systems*, 19(04), 295-308pp.
- Goyache, F., Del Coz, J., Quevedo, J. d., López, S., Alonso, J., Ranilla, J., . . . Bahamonde, A.**, 2001, Using artificial intelligence to design and implement a morphological assessment system in beef cattle, *Animal Science*, **73**(1), 49-60pp.
- Gruber, L., Ledinek, M., Steininger, F., Fuerst-Waltl, B., Zottl, K., Royer, M., Krimberger, K., Mayerhofer, M. and Egger-Danner, C.**, 2018, Body weight prediction using body size measurements in Fleckvieh, Holstein, and Brown Swiss dairy cows in lactation and dry periods, *Archives Animal Breeding*, **61**(4): 413-424pp.

KAYNAKLAR DİZİNİ (Devamı)

- Grzesiak, W., Lacroix, R., Wójcik, J. and Blaszczyk, P.**, 2003, A comparison of neural network and multiple regression predictions for 305-day lactation yield using partial lactation records, *Canadian journal of animal science*, **83**(2): 307-310pp.
- Hassan, K., Samarasinghe, S. and Lopez-Benavides, M.**, 2009, Use of neural networks to detect minor and major pathogens that cause bovine mastitis, *Journal of dairy science*, **92**(4): 1493-1499pp.
- Haykin, S. S.**, 2009, Neural networks and learning machines/Simon Haykin. New York: Prentice Hall.
- Hssina, B., Merbouha, A., Ezzikouri, H., and Erritali, M.**, 2014, A comparative study of decision tree ID3 and C4. 5. *International Journal of Advanced Computer Science and Applications*, **4**(2), 13-19pp.
- Huang, L., Li, S., Zhu, A., Fan, X., Zhang, C. and Wang, H.**, 2018, Non-Contact Body Measurement for Qinchuan Cattle with LiDAR Sensor, *Sensors*, **18**(9), 3014p.
- Karadas, K., Tariq, M., Tariq, M.M. and Eyduran, E.**, 2017, Measuring Predictive Performance of Data Mining and Artificial Neural Network Algorithms for Predicting Lactation Milk Yield in Indigenous Akkaraman Sheep, *Pakistan Journal of Zoology*, **49**(1)p.
- Kass, Gordon V.**, 1980, An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **29**(2): 119-127pp.
- Kashiha, M., Bahr, C., Ott, S., Moons, C. P., Niewold, T. A., Ödberg, F. O., and Berckmans, D.**, 2014, Automatic weight estimation of individual pigs using image analysis. *Computers and electronics in agriculture*, **107**, 38-44pp.
- Kayaalp, G. T., Güney, M. Ç. ve Cebeci, Z.**, 2015, Çoklu Doğrusal Regresyon Modelinde Değişken Seçiminin Zootečniye Uygulanışı, *Çukurova Üniversitesi Ziraat Fakültesi Dergisi*, **30**(1), 1-8s.

KAYNAKLAR DİZİNİ (Devamı)

- Kayri, M.**, 2016, Predictive abilities of bayesian regularization and Levenberg–Marquardt algorithms in artificial neural networks: a comparative empirical study on social data, *Mathematical and Computational Applications*, 21(2), 20pp.
- Khojastehkey, M., Aslaminejad, A. A., & Dianat, R.**, 2015, Pelt Pattern Classification of New Born Lambs Using Image Processing and Artificial Neural Network, *Global Journal of Animal Scientific Research*, 3(2), 321-328pp.
- Kiran, N. R. and Ravi, V.**, 2007, Software reliability prediction by soft computing techniques. *Journal of Systems and Software*, 81(4), 576-583pp.
- Kmet, J., Sakowski, T., Huba, J., Peškovičova, D., Chrenek, J., and Polak, P.**, 2000, Application of video image analysis in the slaughter value estimation of live Simmental bulls. *Archives Animal Breeding*, 43(4), 411-416pp.
- Koc, E. K. and Bozdogan, H.**, 2015, Model selection in multivariate adaptive regression splines (MARS) using information complexity as the fitness function. *Machine Learning*, 101(1-3), 35-58pp.
- Koç, A. and Akman, N.**, 2007, Siyah-alaca tosunların değişik dönemlerdeki vücut ölçüleri ve vücut ölçülerinden canlı ağırlığın tahmini, *ADÜ Ziraat Fakültesi Dergisi* 4(1-2) : 21 - 25s.
- Kongsro, J.**, 2014, Estimation of pig weight using a Microsoft Kinect prototype imaging system. *Computers and electronics in agriculture*, 109, 32-35pp.
- Köksal, E.**, 2017, Modeling of exchange rates by multivariate adaptive regression splines and comparison with classical statistical methods, Ortadoğu Teknik Üniversitesi Uygulamalı Matematik Enstitüsü, Ankara, 98p.
- Krukowski, M.**, 2009, *Automatic determination of body condition score of dairy cows from 3D images*: Skolan för datavetenskap och kommunikation, Kungliga Tekniska högskolan.

KAYNAKLAR DİZİNİ (Devamı)

- Küçükönder, H.**, 2011, Yapay Sinir Ağları ve Tarımda Bir Uygulama, Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü, Kahramanmaraş, 147s.
- Lu, C. J., Chang, C. H., Chen, C. Y., Chiu, C. C., & Lee, T. S.**, December 2009, Stock index prediction: A comparison of MARS, BPN and SVR in an emerging market, In 2009 IEEE International conference on Industrial Engineering and Engineering Management, 2343-2347pp.
- Lines, J., Tillett, R., Ross, L., Chan, D., Hockaday, S., and McFarlane, N.**, 2001, An automatic image-based system for estimating the mass of free-swimming fish, *Computers and electronics in agriculture*, 31(2), 151-168pp.
- Mansouri, I., Safa, M., Ibrahim, Z., Kisi, O., Tahir, M. M., Baharom, S. and Azimi, M.**, 2016, Strength prediction of rotary brace damper using MLR and MARS. *Structural Engineering and Mechanics*, 60(3), 471-488pp.
- Mann, N. J.**, 2018, A brief history of meat in the human diet and current health implications. *Meat science*, 144:169-179pp.
- MATLAB**, MATLAB R2013a for Windows, MathWorks Inc, 2013.
- McCulloch, W. S., and Pitts, W.**, 1943, A logical calculus of the ideas immanent in nervous activity, *The bulletin of mathematical biophysics*, 5(4), 115-133pp.
- Minsky, M. and Papert, S. A.**, 1969, Perceptrons: An introduction to computational geometry. *MIT press*.
- Mollah, M. B. R., Hasan, M. A., Salam, M. A. and Ali, M. A.**, 2010, Digital image analysis to estimate the live weight of broiler, *Computers and electronics in agriculture*, 72(1), 48-52pp.
- Moosavizadeh-Mojarrad, R. and Sepaskhah, A.R.**, 2011, Predicting soil water retention curve by artificial neural networks, *Archives of Agronomy and Soil Science*, 57(1): 3-13pp.

KAYNAKLAR DİZİNİ (Devamı)

- Mukherjee, D. and Rajanikanth, B.**, 2019, Prediction of variation of oxides of nitrogen in plasma-based diesel exhaust treatment using artificial neural network, *International Journal of Environmental Science and Technology*: 1-14pp.
- Nouir, Z., Sayrac, B., Fourestié, B., Tabbara, W., & Brouaye, F.**, 2007, *Comparison of neural network learning algorithms for prediction enhancement of a planning tool*, Paper presented at the 13th European Wireless Conference, Paris, France.
- Orhan, H., Teke, E. Ç., ve Karıcı, Z.**, 2018, Laktasyon Eğrileri Modellemesinde Çok Değişkenli Uyarlanabilir Regresyon Eğrileri (Mars) Yönteminin Uygulanması. *Kahramanmaraş Sütçü İmam Üniversitesi Tarım ve Doğa Dergisi*, 21(3), 363-373s.
- Osborne, J. W. and Waters, E.**, 2002, Four assumptions of multiple regression that researchers should always test. *Practical Assessment, Research, and Evaluation*, 8(1), 2p.
- Ozluturk, A., Kopuzlu, S., Guler, O. and Yanar, M.**, 2006, Determination of linear regression models for estimation of body weights of Eastern Anatolian Red cattle, *Atatürk Üniv. Ziraat Fak. Derg.* **37**(2): 169-175 pp.
- Ölmez, İ.**, 2017, Exhaustive CHAID analysis of factors that affect detention duration of vessels in Paris MoU Region. Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü, İzmir, 99p.
- Önal, A. R.**, 2011, Görüntü işleme teknolojilerinden yararlanarak sığır ve mandalarda morfolometrik parametrelerin tahmininde kullanılan farklı metodların karşılaştırılması. Namık Kemal Üniversitesi Fen Bilimleri Enstitüsü, Tekirdağ, 143s
- Özbalcı, Y.**, 2008, Çok değişkenli uyarlanabilir regresyon kesitleri, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 58s.
- Özkan, D.**, 2016, " Göbekli tepe" örneğinde insani eşgüdümlü eylem, din ve iletişim. *TUBA-AR*, 15(1): 9-20s.

KAYNAKLAR DİZİNİ (Devamı)

- Özkaya, S.**, 2006, Besi Sığırlarında Sayısal Görüntü Analizi İle Canlı Ağırlık ve Karkas Performansının Tahmin Edilmesi ve. Tahmin Modelleri İle Karşılaştırılması. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Isparta, 73s.
- Öztemel, E.**, 2003, Yapay Sinir Ağları. *PapatyaYayincilik, İstanbul*.
- Peraı, A., Nassiri Moghaddam, H., Asadpour, S., Bahrampour, J. and Mansoori, G.**, 2010, A comparison of artificial neural networks with other statistical approaches for the prediction of true metabolizable energy of meat and bone meal, *Poultry Science*, **89**(7): 1562-1568pp.
- Pradana, Z. H., Hidayat, B., and Darana, S.**, 2016, Beef cattle weight determine by using digital image processing. In 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC) - September- (pp. 179-184). IEEE.
- Rahimi, Z., Shafri, H.Z.M. and Norman, M.**, 2018, A GNSS-based weather forecasting approach using Nonlinear Auto Regressive Approach with Exogenous Input (NARX), *Journal of Atmospheric and Solar-Terrestrial Physics*, **178**: 74-84pp.
- Sabancı, K., Aydın, C. ve Ünlerşen, M. F.**, 2012, Görüntü İşleme ve Yapay Sinir Ağları Yardımıyla Patates Sınıflandırma Parametrelerinin Belirlenmesi. *Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 2(2 Sp: A), 59-62 s.
- Sabancı, K.**, 2013, Şeker Pancarı Tarımında Yabancı Ot Mücadelesi İçin Değişken Düzeyli Herbisit Uygulama Parametrelerinin Yapay Sinir Ağlarıyla Belirlenmesi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya, 156s.
- Sapmaz, K., ve Yercan, M.**, 2017, Yapay Sinir Ağı ve Lojistik Regresyon Modeli Kullanılarak Market Markalı Gıda Ürünleri Tüketimini Etkileyen Faktörlerin Belirlenmesi: İzmir İli Örneği, *Tarım Ekonomisi Dergisi*, 23(2), 311-322s.
- Sevimli, Y.**, 2009, Çok değişkenli uyarlanabilir regresyon uzanımlarının bir split-mouth çalışmasında uygulaması, Marmara Üniversitesi Sağlık Bilimleri Enstitüsü, İstanbul, 87s.

KAYNAKLAR DİZİNİ (Devamı)

- Seong, B.**, 2009, Bonferroni correction for seasonal cointegrating ranks. *Economics Letters*, 103(1), 42-44pp.
- Shi, C., Teng, G. and Li, Z.**, 2016, An approach of pig weight estimation using binocular stereo system based on LabVIEW. *Computers and Electronics in Agriculture*, 129, 37-43pp.
- Shi, C., Zhang, J. and Teng, G.**, 2019, Mobile measuring system based on LabVIEW for pig body components estimation in a large-scale farm. *Computers and electronics in agriculture*, 156, 399-405s.
- SPSS**, SPSS v25 for Windows, IBM Inc, New York 10504-1722 United States, 2019.
- SPM**, Salford Predictive Modeler demo versiyon 8.3.0, Minitab Company, 2019.
- Stajanko, D., Brus, M. and Hočevan, M.**, 2008, Estimation of bull live weight through thermographically measured body dimensions, *Computers and electronics in agriculture*, 61(2), 233-240pp.
- Subaşı, N.**, 2011, Gerçek zamanlı insan yüzü belirleme, Trakya Üniversitesi Fen Bilimleri Enstitüsü, Edirne, 70s.
- Şahin, A., Buluş, E. ve Sakalli, M. T.**, 2006, 24-bit renkli resimler üzerinde en önemsiz bite ekleme yöntemini kullanarak bilgi gizleme. *Trakya Üniversitesi Fen Bilimleri Dergisi*, 7(1), 17-22s.
- Şahin M.**, 2009, Parçalı regresyonlar ve tarımsal alanlarda kullanımı, Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü, Kahramanmaraş, 130s.
- Şeker, S., ve Şeker, F.**, 2019, Rakamların icadında muhasebenin itici gücü: Sümer sayı sistemi örneği. *Uluslararası Muhasebe ve Finans Araştırmaları Dergisi*, 1(1), 80-90 s.
- Takma, Ç., Atıl, H. and Aksakal, V.**, 2012, Çoklu doğrusal regresyon ve yapay sinir ağı modellerinin laktasyon süt verimlerine uyum yeteneklerinin karşılaştırılması, *Kafkas Üniversitesi, Veterinerlik Fakültesi Dergisi*, **18**(6): 941-944s.

KAYNAKLAR DİZİNİ (Devamı)

- Takma, Ç., and Gevrekçi, Y.,** 2018, Use of neural network model to predict of egg yield. *Gaziosmanpaşa Üniversitesi Ziraat Fakültesi Dergisi*, 35(2), 147-151s.
- Taşdemir, Ş.,** 2010, Dijital görüntü analiz yöntemi ile siyah alaca ineklerde vücut ölçülerinin belirlenmesi ve canlı ağırlığının tahmin edilmesi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya, 163s.
- Taşova, O.,** 2011, Yapay Sinir Ağları ile Yüz Tanıma, Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü, İzmir, 62s.
- Tayyebi, A. and Pijanowski, B. C.,** 2014, Modeling multiple land use changes using ANN, CART and MARS: Comparing tradeoffs in goodness of fit and explanatory power of data mining tools. *International Journal of Applied Earth Observation and Geoinformation*, 28, 102-116pp.
- Temel, G. O., Ankarali, H. ve Yazici, A. C.,** 2010, Regresyon Modellerine Alternatif Bir Yaklaşım: MARS/An Alternative Approach to Regression Models: MARS. *Türkiye Klinikleri Biyoistatistik*, 2(2), 58p.
- Toprak, S.,** 2015, Çok değişkenli uyarlamalı regresyon eğrileri ve konik programlama ile zaman serilerinin modellenmesi, Dicle Üniversitesi Fen Bilimleri Enstitüsü, Diyarbakır, 120s.
- Tüzemen, N., Yanar, M., Akbulut, Ö., Uğur, F. and Aydın, R.,** 1995, Prediction of body weights from body measurements in Holstein-Friesian calves, *Atatürk Üniversitesi Ziraat Fakültesi Dergisi*, 26(2)p.
- Ugur, A. and Kinaci, A. C.,** 2010, A web-based tool for teaching neural network concepts. *Computer Applications in Engineering Education*, 18(3), 449-457pp.
- Williams, M. N., Grajales, C. A. G. and Kurkiewicz, D.,** 2013, Assumptions of multiple regression: Correcting two misconceptions. *Practical Assessment, Research, and Evaluation*, 18(1), 11p.

KAYNAKLAR DİZİNİ (Devamı)

- Winston, L. R.**, 2011, Human instinct. Random House 493p.
- Wongsriworaphon, A., Arnonkijpanich, B., & Pathumnakul, S.**, 2015, An approach based on digital image analysis to estimate the live weights of pigs in farm environments. *Computers and electronics in agriculture*, 115, 26-33pp.
- Yan, Q., Ding, L., Wei, H., Wang, X., Jiang, C. and Degen, A.**, 2019, Body weight estimation of yaks using body measurements from image analysis. *Measurement*, 140, 76-80pp.
- Yavuz, S. ve Deveci, M.**, 2012, İstatiksel Normalizasyon Tekniklerinin Yapay Sinir Ağı Performansına Etkisi. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, (40), 167-187s.
- Yetkin, M.**, 2015, Tanker-şamandıra Bağlama Sistemlerinin Yapay Sinir Ağları Tekniğiyle Optimizasyonu, Doctoral dissertation, Fen Bilimleri Enstitüsü, İstanbul, 53s.
- Yılmaz, E.**, 2012, İstatiksel analiz yöntemi olarak veri madenciliğinde CHAID algoritması ve Türkiye'de işgücü piyasasının durumunun ve bunun nedenlerinin belirlenmesine ilişkin bir uygulama, Yıldız Teknik Üniversitesi Sosyal Bilimleri Enstitüsü, İstanbul, 149s.
- Zhang, W. and Goh, A. T.**, 2016, Multivariate adaptive regression splines and neural network models for prediction of pile drivability, *Geoscience Frontiers*, 7(1), 45-52pp.

TEŞEKKÜR

Tez çalışmam ve lisansüstü eğitimim boyunca engin bilgi ve tecrübelerinden yararlandığım danışman hocam Prof. Dr. Hülya ATIL'a teşekkürü bir borç bilirim. Tez çalışmam boyunca değerli bilgilerinden yararlandığım beni yönlendiren Tez İzleme Komitesi üyeleri Prof. Dr. Aybars UĞUR ve Prof. Dr. Çiğdem TAKMA'ya şükranlarımı sunarım. Yine verdikleri değerli katkılardan dolayı Prof. Dr. Yavuz AKBAŞ ve Doç. Dr. Muhammed Gökhan CİNSDİKİCİ'ye teşekkürlerimi sunarım.

Erzurum Tarım ve Kırsal Kalkınmayı Destekleme kurum koordinatörü Dr. Atilla ÖZLÜ beyefendiye ve kurum personellerinden değerli dostum Uzman Ziraat Yüksek Mühendisi Erkan EFEKAN'a verilerin toplanmasında sağladıkları kolaylıktan dolayı teşekkür ederim. Dr.Mehmet Ali ÇAKAL'a saha çalışması için kıymetli fikirlerinden dolayı, değerli ağabeyim Bilal ÇAKAL'a ve Yunus ŞAHİN'e malzeme teminindeki desteklerinden dolayı teşekkür ederim.

Görüntüleme için müsaadeleri ve ayrıca güler yüzlü personelleri ile görüntülemenin yapıldığı yaklaşık bir aylık süre boyunca misafirperverlikleri için Erzurum İli Damızlık Koyun Keçi Yetiştiricileri Birliği başkanı Emrullah AKPUNAR ve özellikle bilgisayar operatörleri Murat ve Fatih Bey başta olmak üzere tüm birlik personeline teşekkürü bir borç bilirim. Verileri toplarken büyük emek harcarayarak hep yanımda ve destekçim olan kardeşim Melih KARAHAN ve Recep KARAHAN'a teşekkür ederim.

Eğitim hayatım boyunca maddi ve manevi her konuda yanımda olan canım annem ve babama teşekkür ederim. Lisanüstü eğitimim boyunca, bu uzun yolda sabırla, her daim destek olan kıymetli eşim Zuhal'e ve uslu yavrularım Elzem Berra ile Mehmet Buğra'ya sonsuz teşekkür ederim.

Bu doktora tezi, Öğretim Üyesi Yetiştirme Programı (ÖYP) kapsamında Ege Üniversitesi ÖYP koordinasyon birimi tarafından desteklenmiştir.



ÖZGEÇMİŞ

1982 yılında Erzurum’da doğdu. İlk, orta ve lise öğrenimini Erzurum’da tamamladı. 2005 yılında Atatürk Üniversitesi Ziraat Fakültesinden mezun oldu. 2005-2009 yılları arasında Erzurum İli Damızlık Sığır Yetiştiricileri Birliğinde Sorumlu Müdür olarak görev yaptı. 2010 yılında Harran Üniversitesi Ziraat Fakültesinde, 2011 yılında Iğdır Üniversitesi Ziraat Fakültesinde Araştırma Görevlisi olarak görev yaptı. 2011 yılında Ege Üniversitesinde Araştırma Görevlisi olarak çalışmaya ve Fen Bilimleri Enstitüsü Zootekni Anabilim dalında yüksek lisans öğrenimi görmeye başladı. 2014 yılında aynı enstitüde doktora öğrenimine başladı. Şu anda Iğdır Üniversitesi Ziraat Fakültesinde Araştırma Görevlisi olarak çalışmaya devam etmekte olup evli ve iki çocuk babasıdır.

EKLER

Açıklama	EK
Yapay sinir ağlarında çapraz doğrulama ve tek gizli katman için MATLAB komutları	1a
Yapay sinir ağlarında rastgele veri ayırma ve iki gizli katman için MATLAB komutları	1b
Yapay sinir ağlarında veriler normalize edilmeden yapılan eğitim ve üç gizli katman için MATLAB komutları	1c
Yapay sinir ağlarında üç gizli katman, normalize veriler ve döngü sayısı 100 olması durumunda MATLAB komutları	1d
Çapraz Doğrulama ile verinin ayrıldığı durumdaki eğitimler sonucu en düşük MAPE değerlerinin dikkate alındığı özet tablo	2a
Çapraz Doğrulama ile verinin ayrıldığı durumdaki eğitimler sonucu ortalama MAPE değerlerinin dikkate alındığı özet tablo	2b
Çapraz Doğrulama ile verinin ayrıldığı durumdaki eğitimler sonucu süreler ve en iyi döngü sayılarına ait özet tablo	2c
Rastgele olarak verilerin ayrıldığı durumdaki eğitimler sonucu en düşük MAPE değerlerinin dikkate alındığı özet tablo	3a
Rastgele olarak verilerin ayrıldığı durumdaki eğitimler sonucu ortalama MAPE değerlerinin dikkate alındığı özet tablo	3b
Rastgele olarak verilerin ayrıldığı durumdaki eğitimler sonucu süreler ve en iyi döngü sayılarına ait özet tablo	3c
Yapay sinir ağlarında yapılan bütün eğitimlere ait MAPE değerleri bakımından karşılaştırmalar için CHAID analizi bulguları.	4
Çapraz doğrulama ile veri ayırma durumunda MAPE değerleri bakımından karşılaştırmalara ait CHAID analizi bulguları	5a

Çapraz doğrulama ile veri ayırma durumunda eğitimde geçen süreler bakımından karşılaştırmalara ait CHAID analizi bulguları 5b

Çapraz doğrulama ile veri ayırma durumunda en iyi döngü sayıları bakımından karşılaştırmalara ait CHAID analizi bulguları5c

Verilerin rastgele olarak %70 eğitim %15 test %15 doğrulama setine ayrıldığı durumda MAPE değerlerine ait karşılaştırmalar için CHAID analizi bulguları6a

Verilerin rastgele olarak %70 eğitim %15 test %15 doğrulama setine ayrıldığı durumda RMSE değerlerine ait karşılaştırmalar için CHAID analizi bulguları 6b

Verilerin rastgele olarak %70 eğitim %15 test %15 doğrulama setine ayrıldığı durumda R^2 değerleri bakımından karşılaştırmalar için CHAID analizi bulguları6c

Verilerin rastgele olarak %70 eğitim %15 test %15 doğrulama setine ayrıldığı durumda süreler bakımından karşılaştırmalara ait CHAID analizi bulguları..... 6d

Verilerin rastgele olarak %70 eğitim %15 test %15 doğrulama setine ayrıldığı durumda en iyi döngü sayıları bakımından karşılaştırmalara ait CHAID analizi bulguları6e

Salford Predictive Modeler Programı Çok Değişkenli Uyarlanabilir Regresyon Eğrileri Analiz Sonucu..... 7a

Salford Predictive Modeler Programı Çok Değişkenli Uyarlanabilir Regresyon Eğrileri Analizi Temel Fonksiyonlar Çıktısı..... 7b

EK 1a

```
clc
clear all
load('C:\Users\veri.mat')

k = 7;
cv = cvpartition(length(input1),'kfold',k);

for i=1:k
    trainIdxs{i} = find(training(cv,i));
    testIdxs{i} = find(test(cv,i));
    trainMatrix{i} = [input1(trainIdxs{i}) input2(trainIdxs{i}) target(trainIdxs{i})];
    testMatrix{i} = [input1(testIdxs{i}) input2(testIdxs{i}) target(testIdxs{i})];

    net{i} = feedforwardnet(8);
    net{i}.trainFcn='trainlm';
    net{i}.divideFcn = '';
    net{i}.trainParam.epochs = 1000;
    %net{i}.trainParam.epochs = 500;
    %net{i}.trainParam.epochs = 100;
    net{i}.trainParam.max_fail = 250;
    net{i}.trainParam.min_grad = 1e-7;
    %net{i}.trainParam.min_grad = 1e-6;
    %net{i}.trainParam.min_grad = 1e-5;
    net{i}.trainParam.mu= 0.001;
    % net{i}.trainParam.mu= 0.005;
    % net{i}.trainParam.mu= 0.01;
    net{i}.trainParam.lr= 0.01;
    % net{i}.trainParam.lr= 0.05;
    % net{i}.trainParam.lr= 0.1;
    net{i}.layers{1}.transferFcn = 'tansig';
    [net{i},tr]= train(net{i},trainMatrix{i}(:,1:2),trainMatrix{i}(:,3));
    save lm18cv;
end

sim1=sim(net{1},testMatrix{1}(:,1:2));
sim2=sim(net{2},testMatrix{2}(:,1:2));
sim3=sim(net{3},testMatrix{3}(:,1:2));
sim4=sim(net{4},testMatrix{4}(:,1:2));
sim5=sim(net{5},testMatrix{5}(:,1:2));
sim6=sim(net{6},testMatrix{6}(:,1:2));
sim7=sim(net{7},testMatrix{7}(:,1:2));

testMatrix{1}(:,3)=testMatrix{1}(:,3);
a1=testMatrix{1}(:,3)-sim1';
c1=abs(a1./(testMatrix{1}(:,3)));
MAPE1=mean(c1);
ymape1=100*MAPE1;
a1=a1.^2;
```

```

mse1=mean(a1);
RMSE1=sqrt(mse1);
R1=corr(testMatrix{1}(:,3),sim1');
save lm18cv1;

testMatrix{2}(:,3)=testMatrix{2}(:,3);
a2=testMatrix{2}(:,3)-sim2';
c2=abs(a2./(testMatrix{2}(:,3)));
MAPE2=mean(c2);
ymape2=100*MAPE2;
a2=a2.^2;
mse2=mean(a2);
RMSE2=sqrt(mse2);
R2=corr(testMatrix{2}(:,3),sim2');
save lm18cv2;

testMatrix{3}(:,3)=testMatrix{3}(:,3);
a3=testMatrix{3}(:,3)-sim3';
c3=abs(a3./(testMatrix{3}(:,3)));
MAPE3=mean(c3);
ymape3=100*MAPE3;
a3=a3.^2;
mse3=mean(a3);
RMSE3=sqrt(mse3);
R3=corr(testMatrix{3}(:,3),sim3');
save lm18cv3;

testMatrix{4}(:,3)=testMatrix{4}(:,3);
a4=testMatrix{4}(:,3)-sim4';
c4=abs(a4./(testMatrix{4}(:,3)));
MAPE4=mean(c4);
ymape4=100*MAPE4;
a4=a4.^2;
mse4=mean(a4);
RMSE4=sqrt(mse4);
R4=corr(testMatrix{4}(:,3),sim4');
save lm18cv4;

testMatrix{5}(:,3)=testMatrix{5}(:,3);
a5=testMatrix{5}(:,3)-sim5';
c5=abs(a5./(testMatrix{5}(:,3)));
MAPE5=mean(c5);
ymape5=100*MAPE5;
a5=a5.^2;
mse5=mean(a5);
RMSE5=sqrt(mse5);
R5=corr(testMatrix{5}(:,3),sim5');
save lm18cv5;

```

```

testMatrix{6}(:,3)=testMatrix{6}(:,3);
a6=testMatrix{6}(:,3)-sim6';
c6=abs(a6./(testMatrix{6}(:,3)));
MAPE6=mean(c6);
ymape6=100*MAPE6;
a6=a6.^2;
mse6=mean(a6);
RMSE6=sqrt(mse6);
R6=corr(testMatrix{6}(:,3),sim6');
save lm18cv6;

testMatrix{7}(:,3)=testMatrix{7}(:,3);
a7=testMatrix{7}(:,3)-sim7';
c7=abs(a7./(testMatrix{7}(:,3)));
MAPE7=mean(c7);
ymape7=100*MAPE7;
a7=a7.^2;
mse7=mean(a7);
RMSE7=sqrt(mse7);
R7=corr(testMatrix{7}(:,3),sim7');
MAPEORT=(MAPE1+MAPE2+MAPE3+MAPE4+MAPE5+MAPE6+MAPE7)/7;
RORT=(R1+R2+R3+R4+R5+R6+R7)/7;
RKARE=RORT^2;
RMSEORT=(RMSE1+RMSE2+RMSE3+RMSE4+RMSE5+RMSE6+RMSE7)/7;
save lm18cv7;

```

EK 1b

```
clc
clear all
load('C:\Users\veri.mat')
input1=input1';target=target';
[input1n,meaninput1,stdinput1,targetn,meantarget,stdtarget] = prestd(input1,target);
k=7;
input1n=input1n';targetn=targetn';
for i=1:k
net{i} = feedforwardnet([8 8]);
[trainInd,valInd,testInd] = dividerand(length(input1n),0.7,0.15,0.15);
net{i}.trainFcn='traingdx';
% net{i}.trainParam.epochs = 1000;
% net{i}.trainParam.epochs = 500;
net{i}.trainParam.epochs = 100;
net{i}.trainParam.max_fail = 250;
net{i}.trainParam.min_grad = 1e-7;
% net{i}.trainParam.min_grad = 1e-6;
% net{i}.trainParam.min_grad = 1e-5;
net{i}.trainParam.mu= 0.001;
% net{i}.trainParam.mu= 0.005;
% net{i}.trainParam.mu= 0.01;
net{i}.trainParam.lr= 0.01;
% net{i}.trainParam.lr= 0.05;
% net{i}.trainParam.lr= 0.1;
net{i}.layers{1}.transferFcn = 'tansig';
[net{i},tr] = train(net{i},input1n',targetn');
save gdx28mnozet
end

sim1=sim(net{1},input1n');
sim2=sim(net{2},input1n');
sim3=sim(net{3},input1n');
sim4=sim(net{4},input1n');
sim5=sim(net{5},input1n');
sim6=sim(net{6},input1n');
sim7=sim(net{7},input1n');

sim1=stdtarget*sim1 + meantarget;
sim2=stdtarget*sim2 + meantarget;
sim3=stdtarget*sim3 + meantarget;
sim4=stdtarget*sim4 + meantarget;
sim5=stdtarget*sim5 + meantarget;
sim6=stdtarget*sim6 + meantarget;
sim7=stdtarget*sim7 + meantarget;

sim1=sim1';
sim2=sim2';
sim3=sim3';
```

```

sim4=sim4';
sim5=sim5';
sim6=sim6';
sim7=sim7';

a1=target-sim1';
c1=abs(a1./(target));
MAPE1=mean(c1);
ymape1=100*MAPE1;
a1=a1.^2;
mse1=mean(a1);
RMSE1=sqrt(mse1);
R1=corr(target',sim1);
R1KARE=R1^2;
save gdx28mnozet1;

a2=target-sim2';
c2=abs(a2./(target));
MAPE2=mean(c2);
ymape2=100*MAPE2;
a2=a2.^2;
mse2=mean(a2);
RMSE2=sqrt(mse2);
R2=corr(target',sim2);
R2KARE=R2^2;
save gdx28mnozet2;

a3=target-sim3';
c3=abs(a3./(target));
MAPE3=mean(c3);
ymape3=100*MAPE3;
a3=a3.^2;
mse3=mean(a3);
RMSE3=sqrt(mse3);
R3=corr(target',sim3);
R3KARE=R3^2;
save gdx28mnozet3;

a4=target-sim4';
c4=abs(a4./(target));
MAPE4=mean(c4);
ymape4=100*MAPE4;
a4=a4.^2;
mse4=mean(a4);
RMSE4=sqrt(mse4);
R4=corr(target',sim4);
R4KARE=R4^2;
save gdx28mnozet4;

```

```

a5=target-sim5';
c5=abs(a5./(target));
MAPE5=mean(c5);
ymape5=100*MAPE5;
a5=a5.^2;
mse5=mean(a5);
RMSE5=sqrt(mse5);
R5=corr(target',sim5);
R5KARE=R5^2;
save gdx28mnozet5;

a6=target-sim6';
c6=abs(a6./(target));
MAPE6=mean(c6);
ymape6=100*MAPE6;
a6=a6.^2;
mse6=mean(a6);
RMSE6=sqrt(mse6);
R6=corr(target',sim6);
R6KARE=R6^2;
save gdx28mnozet6;

a7=target-sim7';
c7=abs(a7./(target));
MAPE7=mean(c7);
ymape7=100*MAPE7;
a7=a7.^2;
mse7=mean(a7);
RMSE7=sqrt(mse7);
R7=corr(target',sim7);
R7KARE=R7^2;

MAPEORT=(MAPE1+MAPE2+MAPE3+MAPE4+MAPE5+MAPE6+MAPE7)/7;
RKAREORT=(R1KARE+R2KARE+R3KARE+R4KARE+R5KARE+R6KARE+R7KARE)/7;
RMSEORT=(RMSE1+RMSE2+RMSE3+RMSE4+RMSE5+RMSE6+RMSE7)/7;
save gdx28mnozet7;

```

EK1c

```
clc
clear all
load('C:\Users\veri.mat')
input1=input1';target=target';
k=7;

for i=1:k
net{i} = feedforwardnet([8 8 8]);
[trainInd,valInd,testInd] = dividerand(length(input1),0.7,0.15,0.15);
net{i}.trainFcn='traingdx';
% net{i}.trainParam.epochs = 1000;
% net{i}.trainParam.epochs = 500;
net{i}.trainParam.epochs = 100;
net{i}.trainParam.max_fail = 250;
net{i}.trainParam.min_grad = 1e-7;
%net{i}.trainParam.min_grad = 1e-6;
%net{i}.trainParam.min_grad = 1e-5;
net{i}.trainParam.mu= 0.001;
% net{i}.trainParam.mu= 0.005;
% net{i}.trainParam.mu= 0.01;
% net{i}.trainParam.lr= 0.01;
net{i}.trainParam.lr= 0.01;
% net{i}.trainParam.lr= 0.1;
net{i}.layers{1}.transferFcn = 'tansig';
[net{i},tr] = train(net{i},input1,target);
save gdx38rnstdyokep100
end

sim1=sim(net{1},input1);
sim2=sim(net{2},input1);
sim3=sim(net{3},input1);
sim4=sim(net{4},input1);
sim5=sim(net{5},input1);
sim6=sim(net{6},input1);
sim7=sim(net{7},input1);

sim1=sim1';
sim2=sim2';
sim3=sim3';
sim4=sim4';
sim5=sim5';
sim6=sim6';
sim7=sim7';

a1=target-sim1';
c1=abs(a1./(target));
MAPE1=mean(c1);
ymape1=100*MAPE1;
```

```
a1=a1.^2;  
mse1=mean(a1);  
RMSE1=sqrt(mse1);  
R1=corr(target',sim1);  
R1KARE=R1^2;  
save gdx38rnstdyokep1001;
```

```
a2=target-sim2';  
c2=abs(a2./(target));  
MAPE2=mean(c2);  
ymape2=100*MAPE2;  
a2=a2.^2;  
mse2=mean(a2);  
RMSE2=sqrt(mse2);  
R2=corr(target',sim2);  
R2KARE=R2^2;  
save gdx38rnstdyokep1002;
```

```
a3=target-sim3';  
c3=abs(a3./(target));  
MAPE3=mean(c3);  
ymape3=100*MAPE3;  
a3=a3.^2;  
mse3=mean(a3);  
RMSE3=sqrt(mse3);  
R3=corr(target',sim3);  
R3KARE=R3^2;  
save gdx38rnstdyokep1003;
```

```
a4=target-sim4';  
c4=abs(a4./(target));  
MAPE4=mean(c4);  
ymape4=100*MAPE4;  
a4=a4.^2;  
mse4=mean(a4);  
RMSE4=sqrt(mse4);  
R4=corr(target',sim4);  
R4KARE=R4^2;  
save gdx38rnstdyokep1004;
```

```
a5=target-sim5';  
c5=abs(a5./(target));  
MAPE5=mean(c5);  
ymape5=100*MAPE5;  
a5=a5.^2;  
mse5=mean(a5);  
RMSE5=sqrt(mse5);  
R5=corr(target',sim5);  
R5KARE=R5^2;
```

```
save gdx38rnstdyokep1005;
```

```
a6=target-sim6';  
c6=abs(a6./(target));  
MAPE6=mean(c6);  
ymape6=100*MAPE6;  
a6=a6.^2;  
mse6=mean(a6);  
RMSE6=sqrt(mse6);  
R6=corr(target',sim6);  
R6KARE=R6^2;  
save gdx38rnstdyokep1006;
```

```
a7=target-sim7';  
c7=abs(a7./(target));  
MAPE7=mean(c7);  
ymape7=100*MAPE7;  
a7=a7.^2;  
mse7=mean(a7);  
RMSE7=sqrt(mse7);  
R7=corr(target',sim7);  
R7KARE=R7^2;
```

```
MAPEORT=(MAPE1+MAPE2+MAPE3+MAPE4+MAPE5+MAPE6+MAPE7)/7;  
RKAREORT=(R1KARE+R2KARE+R3KARE+R4KARE+R5KARE+R6KARE+R7KARE)/7;  
RMSEORT=(RMSE1+RMSE2+RMSE3+RMSE4+RMSE5+RMSE6+RMSE7)/7;  
save gdx38rnstdyokep1007
```

EK 1d

```
clc
clear all
load('C:\Users\veri.mat')
input1=input1';target=target';
[input1n,meaninput1,stdinput1,targetn,meantarget,stdtarget] = prestd(input1,target);
k=7;
input1n=input1n';targetn=targetn';
for i=1:k
net{i} = feedforwardnet([8 8 8]);
[trainInd,valInd,testInd] = dividerand(length(input1n),0.7,0.15,0.15);
net{i}.trainFcn='traingdx';
% net{i}.trainParam.epochs = 1000;
% net{i}.trainParam.epochs = 500;
net{i}.trainParam.epochs = 100;
net{i}.trainParam.max_fail = 250;
net{i}.trainParam.min_grad = 1e-7;
% net{i}.trainParam.min_grad = 1e-6;
% net{i}.trainParam.min_grad = 1e-5;
net{i}.trainParam.mu= 0.001;
% net{i}.trainParam.mu= 0.005;
% net{i}.trainParam.mu= 0.01;
net{i}.trainParam.lr= 0.01;
% net{i}.trainParam.lr= 0.05;
% net{i}.trainParam.lr= 0.1;
net{i}.layers{1}.transferFcn = 'tansig';
[net{i},tr] = train(net{i},input1n',targetn');
save gdx38rnep100
end

sim1=sim(net{1},input1n');
sim2=sim(net{2},input1n');
sim3=sim(net{3},input1n');
sim4=sim(net{4},input1n');
sim5=sim(net{5},input1n');
sim6=sim(net{6},input1n');
sim7=sim(net{7},input1n');

sim1=stdtarget*sim1 + meantarget;
sim2=stdtarget*sim2 + meantarget;
sim3=stdtarget*sim3 + meantarget;
sim4=stdtarget*sim4 + meantarget;
sim5=stdtarget*sim5 + meantarget;
sim6=stdtarget*sim6 + meantarget;
sim7=stdtarget*sim7 + meantarget;

sim1=sim1';
sim2=sim2';
sim3=sim3';
```

```
sim4=sim4';
sim5=sim5';
sim6=sim6';
sim7=sim7';

a1=target-sim1';
c1=abs(a1./(target));
MAPE1=mean(c1);
ymape1=100*MAPE1;
a1=a1.^2;
mse1=mean(a1);
RMSE1=sqrt(mse1);
R1=corr(target',sim1);
R1KARE=R1^2;
save gdx38nep1001;

a2=target-sim2';
c2=abs(a2./(target));
MAPE2=mean(c2);
ymape2=100*MAPE2;
a2=a2.^2;
mse2=mean(a2);
RMSE2=sqrt(mse2);
R2=corr(target',sim2);
R2KARE=R2^2;
save gdx38nep1002;

a3=target-sim3';
c3=abs(a3./(target));
MAPE3=mean(c3);
ymape3=100*MAPE3;
a3=a3.^2;
mse3=mean(a3);
RMSE3=sqrt(mse3);
R3=corr(target',sim3);
R3KARE=R3^2;
save gdx38nep1003;

a4=target-sim4';
c4=abs(a4./(target));
MAPE4=mean(c4);
ymape4=100*MAPE4;
a4=a4.^2;
mse4=mean(a4);
RMSE4=sqrt(mse4);
R4=corr(target',sim4);
R4KARE=R4^2;
save gdx38nep1004;
```

```

a5=target-sim5';
c5=abs(a5./(target));
MAPE5=mean(c5);
ymape5=100*MAPE5;
a5=a5.^2;
mse5=mean(a5);
RMSE5=sqrt(mse5);
R5=corr(target',sim5);
R5KARE=R5^2;
save gdx38nep1005;

a6=target-sim6';
c6=abs(a6./(target));
MAPE6=mean(c6);
ymape6=100*MAPE6;
a6=a6.^2;
mse6=mean(a6);
RMSE6=sqrt(mse6);
R6=corr(target',sim6);
R6KARE=R6^2;
save gdx38nep1006;

a7=target-sim7';
c7=abs(a7./(target));
MAPE7=mean(c7);
ymape7=100*MAPE7;
a7=a7.^2;
mse7=mean(a7);
RMSE7=sqrt(mse7);
R7=corr(target',sim7);
R7KARE=R7^2;

MAPEORT=(MAPE1+MAPE2+MAPE3+MAPE4+MAPE5+MAPE6+MAPE7)/7;
RKAREORT=(R1KARE+R2KARE+R3KARE+R4KARE+R5KARE+R6KARE+R7KARE)/7;
RMSEORT=(RMSE1+RMSE2+RMSE3+RMSE4+RMSE5+RMSE6+RMSE7)/7;
save gdx38nep1007;

```

EK 2a

Çapraz Doğrulama ile verinin ayrıldığı durumdaki eğitimler sonucu en düşük MAPE değerlerinin dikkate alındığı özet tablo.

PARAMETRELER					ALGORİTMALAR														
					Levenberg Marquardt					Bayesian Regulation					GDX				
Normalizasyon var	MU	λ	GD	Döngü	KS	NS	MAPE (%)	RMSE (Kg)	R ²	KS	NS	MAPE (%)	RMSE (Kg)	R ²	KS	NS	MAPE (%)	RMSE (Kg)	R ²
	0.001	0.01	10 ⁻⁷	1000	1	3	5.25	39.90	0.9153	1	8	5.07	37.65	0.9268	1	3	5.10	37.41	0.9262
	0.005	0.01	10 ⁻⁷	1000	1	5	5.19	36.71	0.9274	2	8	5.05	37.33	0.9257	3	5	5.08	37.28	0.9294
	0.01	0.01	10 ⁻⁷	1000	1	8	5.12	36.52	0.9252	1	3	5.03	37.50	0.9272	2	8	5.09	37.59	0.9236
	0.001	0.05	10 ⁻⁷	1000	1	3	5.47	46.02	0.8987	3	3	4.98	37.21	0.9276	2	3	5.07	36.89	0.9253
	0.001	0.1	10 ⁻⁷	1000	3	3	5.29	38.97	0.9210	3	8	5.03	37.29	0.9238	1	8	4.96	36.79	0.9334
	0.001	0.01	10⁻⁶	1000	3	3	5.29	38.97	0.9210	1	3	5.03	37.50	0.9272	2	8	5.09	37.60	0.9236
	0.001	0.01	10⁻⁵	1000	3	3	5.29	38.97	0.9210	1	3	5.03	37.50	0.9272	2	8	5.09	37.59	0.9236
	0.001	0.01	10 ⁻⁷	500	3	3	5.16	37.86	0.9269	1	3	5.03	37.50	0.9272	1	5	5.23	38.64	0.9251
	0.001	0.01	10 ⁻⁷	100	3	3	5.04	36.94	0.9281	2	3	4.99	37.09	0.9270	3	8	5.79	44.13	0.9037
Normalizasyon yok	0.001	0.01	10 ⁻⁷	1000	1	5	4.95	36.74	0.9277	2	5	4.99	37.74	0.9241	3	3	5.06	37.01	0.9272
	0.005	0.01	10 ⁻⁷	1000	1	3	5.12	37.99	0.9255	1	5	5.00	36.81	0.9230	1	8	5.01	36.78	0.9259
	0.01	0.01	10 ⁻⁷	1000	2	3	5.12	38.05	0.9229	2	5	5.06	37.74	0.9253	1	3	4.96	37.50	0.9234
	0.001	0.05	10 ⁻⁷	1000	1	3	5.21	38.53	0.9234	1	3	5.03	37.50	0.9272	2	3	5.15	38.14	0.9245
	0.001	0.1	10 ⁻⁷	1000	1	3	5.20	37.79	0.9255	2	5	5.08	37.78	0.9252	2	8	5.14	37.87	0.9272
	0.001	0.01	10⁻⁶	1000	1	3	5.18	38.57	0.9290	1	5	5.10	37.95	0.9234	1	8	5.28	39.58	0.9141
	0.001	0.01	10⁻⁵	1000	1	3	5.21	38.53	0.9234	2	3	5.07	37.84	0.9290	1	8	5.05	36.99	0.9263
	0.001	0.01	10 ⁻⁷	500	1	3	5.12	38.57	0.9193	2	8	5.04	37.57	0.9220	1	8	5.20	38.38	0.9177
	0.001	0.01	10 ⁻⁷	100	1	5	4.85	35.19	0.9291	3	3	5.03	37.02	0.9219	1	3	6.42	47.85	0.8823

-MU: Momentum güncelleme katsayısı (Momentum update) - λ : Öğrenme oranı (Learning rate) - GD: Minimum gradyan değeri - Döngü: Müsaade edilen en çok döngü sayısı (epoch) - KS: Katman sayısı (Hidden layer number) - NS: Nöron sayısı (Hidden layer neuron number) - MAPE: Ortalama mutlak hata yüzdesi (Mean absolute percentage error) - RMSE: Hata kareler ortalamasının karekökü (Root mean square error) - R²: Belirleme katsayısı (Coefficient of determination)

EK 2b

Çapraz doğrulama ile verinin ayrıldığı durumdaki eğitimler sonucu ortalama MAPE değerlerinin dikkate alındığı özet tablo.

PARAMETRELER					ALGORİTMALAR								
					Levenberg Marquardt			Bayesian Regulation			GDX		
	MU	λ	GD	Döngü	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
Normalizasyon var	0.001	0.01	10 ⁻⁷	1000	77.07	872.58	0.6747	5.11	41.22	0.8949	5.31	38.74	0.9220
	0.005	0.01	10 ⁻⁷	1000	12.78	182.36	0.6800	5.38	39.22	0.9098	5.34	39.11	0.9193
	0.01	0.01	10 ⁻⁷	1000	16.70	315.00	0.7022	5.16	38.49	0.9245	5.29	38.74	0.9196
	0.001	0.05	10 ⁻⁷	1000	15.02	267.15	0.6527	5.09	37.76	0.9249	5.27	38.18	0.9223
	0.001	0.1	10 ⁻⁷	1000	15.23	264.52	0.6427	5.10	37.86	0.9253	5.36	38.82	0.9202
	0.001	0.01	10⁻⁶	1000	15.23	234.13	0.6427	5.37	39.28	0.9109	5.29	38.75	0.9196
	0.001	0.01	10⁻⁵	1000	15.24	264.90	0.6418	5.37	39.28	0.9109	5.29	38.74	0.9196
	0.001	0.01	10 ⁻⁷	500	11.80	179.76	0.6627	5.37	39.28	0.9109	5.49	40.18	0.9131
	0.001	0.01	10 ⁻⁷	100	6.00	52.12	0.8607	5.09	37.76	0.9245	6.55	48.12	0.8792
Normalizasyon yok	0.001	0.01	10 ⁻⁷	1000	13.60	202.69	0.6904	5.63	41.43	0.8927	5.52	40.12	0.9128
	0.005	0.01	10 ⁻⁷	1000	10.70	136.65	0.7019	5.40	39.83	0.9079	5.25	38.38	0.9201
	0.01	0.01	10 ⁻⁷	1000	13.76	229.56	0.7143	5.14	38.07	0.9230	5.41	39.40	0.9175
	0.001	0.05	10 ⁻⁷	1000	15.87	233.39	0.6949	5.37	39.27	0.9110	5.87	42.26	0.8998
	0.001	0.1	10 ⁻⁷	1000	11.88	162.01	0.6768	5.33	39.24	0.9084	5.51	40.20	0.9125
	0.001	0.01	10⁻⁶	1000	12.49	170.65	0.7269	5.41	39.89	0.9085	5.40	39.53	0.9188
	0.001	0.01	10⁻⁵	1000	11.38	181.25	0.6934	5.33	39.63	0.9071	5.23	38.32	0.9210
	0.001	0.01	10 ⁻⁷	500	8.84	82.86	0.7749	5.11	37.87	0.9242	5.49	40.90	0.9105
	0.001	0.01	10 ⁻⁷	100	5.92	52.01	0.8754	5.34	39.34	0.9102	7.69	55.18	0.8427

-**MU**: Momentum güncelleme katsayısı (Momentum update) - λ : Öğrenme oranı (Learning rate) - **GD**: Minimum gradyan değeri - **Döngü**: Müsaade edilen en çok döngü sayısı (epoch) - **KS**: Katman sayısı (Hidden layer number) - **NS**: Nöron sayısı (Hidden layer neuron number) - **MAPE**: Ortalam mutlak hata yüzdesi (Mean absolute percentage error) - **RMSE**: Hata kareler ortalamasının karekökü (Root mean square error) - **R²**: Belirleme katsayısı (coefficient of determination)

EK 2c

Çapraz Doğrulama ile verinin ayrıldığı durumdaki eğitimler sonucu süreler ve en iyi döngü sayılarına ait özet tablo.

PARAMETRELER					Levenberg Marquardt		Bayesian Regulation		GDX	
Normalizasyon var	MU	λ	GD	Döngü	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
	0.001	0.01	10^{-7}	1000	8.8987	897	3.6364	279	0.9401	999
	0.005	0.01	10^{-7}	1000	9.2079	1000	6.0441	580	0.8166	999
	0.01	0.01	10^{-7}	1000	8.4380	893	5.3727	464	0.8120	999
	0.001	0.05	10^{-7}	1000	8.5701	811	2.9456	340	0.8180	996
	0.001	0.1	10^{-7}	1000	7.6313	811	4.7597	441	0.8262	999
	0.001	0.01	10^{-6}	1000	7.8981	807	4.4454	426	0.8319	999
	0.001	0.01	10^{-5}	1000	6.9882	741	4.5156	426	0.8150	999
	0.001	0.01	10^{-7}	500	3.8220	423	3.2193	325	0.4757	498
	0.001	0.01	10^{-7}	100	0.8569	98	0.9350	99	0.1158	98
Normalizasyon yok	0.001	0.01	10^{-7}	1000	8.3894	909	4.8452	417	0.8349	1000
	0.005	0.01	10^{-7}	1000	9.3480	896	2.1043	179	0.8990	998
	0.01	0.01	10^{-7}	1000	8.0507	898	4.1670	405	0.8522	999
	0.001	0.05	10^{-7}	1000	9.7999	1000	5.1930	420	0.8853	997
	0.001	0.1	10^{-7}	1000	7.1744	792	2.6207	269	1.0138	998
	0.001	0.01	10^{-6}	1000	8.5497	1000	3.8328	351	0.8776	1000
	0.001	0.01	10^{-5}	1000	11.0658	1000	2.3748	230	0.9024	998
	0.001	0.01	10^{-7}	500	3.7660	445	2.6171	246	0.4561	497
	0.001	0.01	10^{-7}	100	0.7498	100	0.9212	100	0.1348	100

-**MU**: Momentum güncelleme katsayısı (Momentum update) - λ : Öğrenme oranı (Learning rate) -**Gd**: Minimum gradyan değeri - **Döngü**: Müsaade edilen en çok döngü sayısı (epoch) -**Süre**: Eğitimde geçen süre – **EDS**: Eğitimin durdurulduğu en iyi döngü sayısı (best epoch)

EK 3a

Rastgele olarak verilerin ayrıldığı durumdaki eğitimler sonucu en düşük MAPE değerlerinin dikkate alındığı özet tablo.

PARAMETRELER					ALGORİTMALAR														
					Levenberg Marquardt					Bayesian Regulation					GDX				
Normalizasyon var	MU	λ	GD	Döngü	KS	NS	MAPE (%)	RMSE (Kg)	R ²	KS	NS	MAPE (%)	RMSE (Kg)	R ²	KS	NS	MAPE (%)	RMSE (Kg)	R ²
	0.001	0.01	10 ⁻⁷	1000	2	8	4.58	34.01	0.9385	3	3	4.75	34.72	0.9355	2	8	4.88	36.29	0.9309
	0.005	0.01	10 ⁻⁷	1000	2	5	4.64	33.90	0.9389	3	3	4.77	35.33	0.9334	3	8	4.90	35.95	0.9313
	0.01	0.01	10 ⁻⁷	1000	3	8	4.56	33.81	0.9399	3	5	4.80	36.53	0.9289	1	3	5.08	38.34	0.9224
	0.001	0.05	10 ⁻⁷	1000	3	5	4.59	33.89	0.9388	3	3	4.85	36.06	0.9306	3	8	5.01	36.60	0.9292
	0.001	0.1	10 ⁻⁷	1000	1	8	4.58	33.52	0.9402	3	8	4.78	35.38	0.9331	2	3	4.97	36.90	0.9277
	0.001	0.01	10⁻⁶	1000	2	5	4.60	34.50	0.9372	3	3	4.84	36.01	0.9307	3	8	4.90	35.95	0.9313
	0.001	0.01	10⁻⁵	1000	2	5	4.60	34.50	0.9372	3	3	4.76	35.17	0.9339	2	8	5.04	37.03	0.9269
	0.001	0.01	10 ⁻⁷	500	2	8	4.56	33.79	0.9393	3	5	4.82	35.93	0.9310	1	3	5.19	38.50	0.9220
	0.001	0.01	10 ⁻⁷	100	2	5	4.58	33.57	0.9401	3	3	4.72	35.02	0.9345	3	8	5.43	39.56	0.9168
Normalizasyon yok	0.001	0.01	10 ⁻⁷	1000	1	8	4.66	34.65	0.9372	3	5	4.77	35.76	0.9318	2	8	4.88	36.06	0.9309
	0.005	0.01	10 ⁻⁷	1000	2	5	4.60	33.89	0.9391	2	3	4.90	36.39	0.9294	1	3	4.99	37.01	0.9273
	0.01	0.01	10 ⁻⁷	1000	1	8	4.67	34.31	0.9374	3	3	4.84	36.07	0.9305	2	5	5.04	37.32	0.9263
	0.001	0.05	10 ⁻⁷	1000	1	8	4.72	34.80	0.9355	3	3	4.84	36.01	0.9307	2	8	4.99	36.40	0.9296
	0.001	0.1	10 ⁻⁷	1000	1	8	4.72	34.80	0.9355	3	3	4.84	36.01	0.9307	3	8	5.09	36.95	0.9284
	0.001	0.01	10⁻⁶	1000	2	8	4.56	34.21	0.9377	3	5	4.83	35.94	0.9310	1	8	4.98	36.42	0.9302
	0.001	0.01	10⁻⁵	1000	3	5	4.60	34.03	0.9386	3	8	4.70	34.85	0.9351	2	8	4.89	36.25	0.9311
	0.001	0.01	10 ⁻⁷	500	1	8	4.72	34.80	0.9355	3	3	4.84	36.01	0.9307	1	3	5.23	38.70	0.9213
	0.001	0.01	10 ⁻⁷	100	3	5	4.62	34.21	0.9379	2	5	4.82	35.63	0.9323	1	3	6.64	49.45	0.8671

-**MU**: Momentum güncelleme katsayısı (Momentum update) - λ : Öğrenme oranı (Learning rate) - **GD**: Minimum gradyan değeri - **Döngü**: Müsaade edilen en çok döngü sayısı (epoch) - **KS**: Katman sayısı (Hidden layer number) - **NS**: Nöron sayısı (Hidden layer neuron number) - **MAPE**: Ortalam mutlak hata yüzdesi (Mean absolute percentage error) - **RMSE**: Hata kareler ortalamasının karekökü (Root mean square error) - **R²**: Belirleme katsayısı (coefficient of determination)

EK 3b

Rastgele olarak verilerin ayrıldığı durumdaki eğitimler sonucu ortalama MAPE değerlerinin dikkate alındığı özet tablo.

PARAMETRELER					ALGORİTMALAR								
					Levenberg Marquardt			Bayesian Regulation			GDX		
	MU	λ	GD	Döngü	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²	MAPE (%)	RMSE (Kg)	R ²
Normalizasyon var	0.001	0.01	10 ⁻⁷	1000	4.75	34.97	0.9352	5.14	37.96	0.9147	5.19	38.51	0.9218
	0.005	0.01	10 ⁻⁷	1000	4.79	35.61	0.9331	4.89	35.99	0.9295	5.16	37.86	0.9241
	0.01	0.01	10 ⁻⁷	1000	4.79	35.68	0.9318	4.90	36.53	0.9289	5.23	38.50	0.9215
	0.001	0.05	10 ⁻⁷	1000	4.74	34.91	0.9352	4.91	36.57	0.9287	5.30	39.02	0.9193
	0.001	0.1	10 ⁻⁷	1000	4.78	35.85	0.9314	4.90	36.48	0.9290	5.18	37.97	0.9234
	0.001	0.01	10⁻⁶	1000	4.79	35.51	0.9335	5.13	37.93	0.9148	5.16	37.86	0.9241
	0.001	0.01	10⁻⁵	1000	4.79	35.52	0.9335	4.87	36.34	0.9296	5.21	38.26	0.9222
	0.001	0.01	10 ⁻⁷	500	4.75	36.03	0.9309	5.16	38.20	0.9138	5.41	40.02	0.9149
	0.001	0.01	10 ⁻⁷	100	4.74	34.84	0.9358	4.90	36.51	0.9289	6.20	45.79	0.8866
Normalizasyon yok	0.001	0.01	10 ⁻⁷	1000	4.82	36.30	0.9303	5.16	38.18	0.9139	5.41	39.59	0.9113
	0.005	0.01	10 ⁻⁷	1000	5.16	38.09	0.9115	4.90	36.42	0.9292	5.82	42.06	0.8895
	0.01	0.01	10 ⁻⁷	1000	4.81	35.68	0.9330	5.14	38.06	0.9144	5.58	40.53	0.9025
	0.001	0.05	10 ⁻⁷	1000	4.82	35.69	0.9329	5.40	39.60	0.8998	5.86	42.50	0.8917
	0.001	0.1	10 ⁻⁷	1000	4.82	35.69	0.9329	5.40	39.60	0.8998	5.46	39.98	0.9118
	0.001	0.01	10⁻⁶	1000	4.82	35.90	0.9317	4.91	36.52	0.9289	5.40	39.46	0.9110
	0.001	0.01	10⁻⁵	1000	4.78	35.29	0.9340	4.87	36.09	0.9305	5.19	38.42	0.9218
	0.001	0.01	10 ⁻⁷	500	4.82	35.69	0.9329	5.40	39.60	0.8998	5.78	42.04	0.8978
	0.001	0.01	10 ⁻⁷	100	5.08	37.39	0.9208	5.16	38.09	0.9142	7.39	54.38	0.8374

-**MU**: Momentum güncelleme katsayısı (Momentum update) - λ : Öğrenme oranı (Learning rate) - **GD**: Minimum gradyan değeri - **Döngü**: Müsaade edilen en çok döngü sayısı (epoch) - **KS**: Katman sayısı (Hidden layer number) - **NS**: Nöron sayısı (Hidden layer neuron number) - **MAPE**: Ortalam mutlak hata yüzdesi (Mean absolute percentage error) - **RMSE**: Hata kareler ortalamasının karekökü (Root mean square error) - **R²**: Belirleme katsayısı (coefficient of determination)

EK 3c

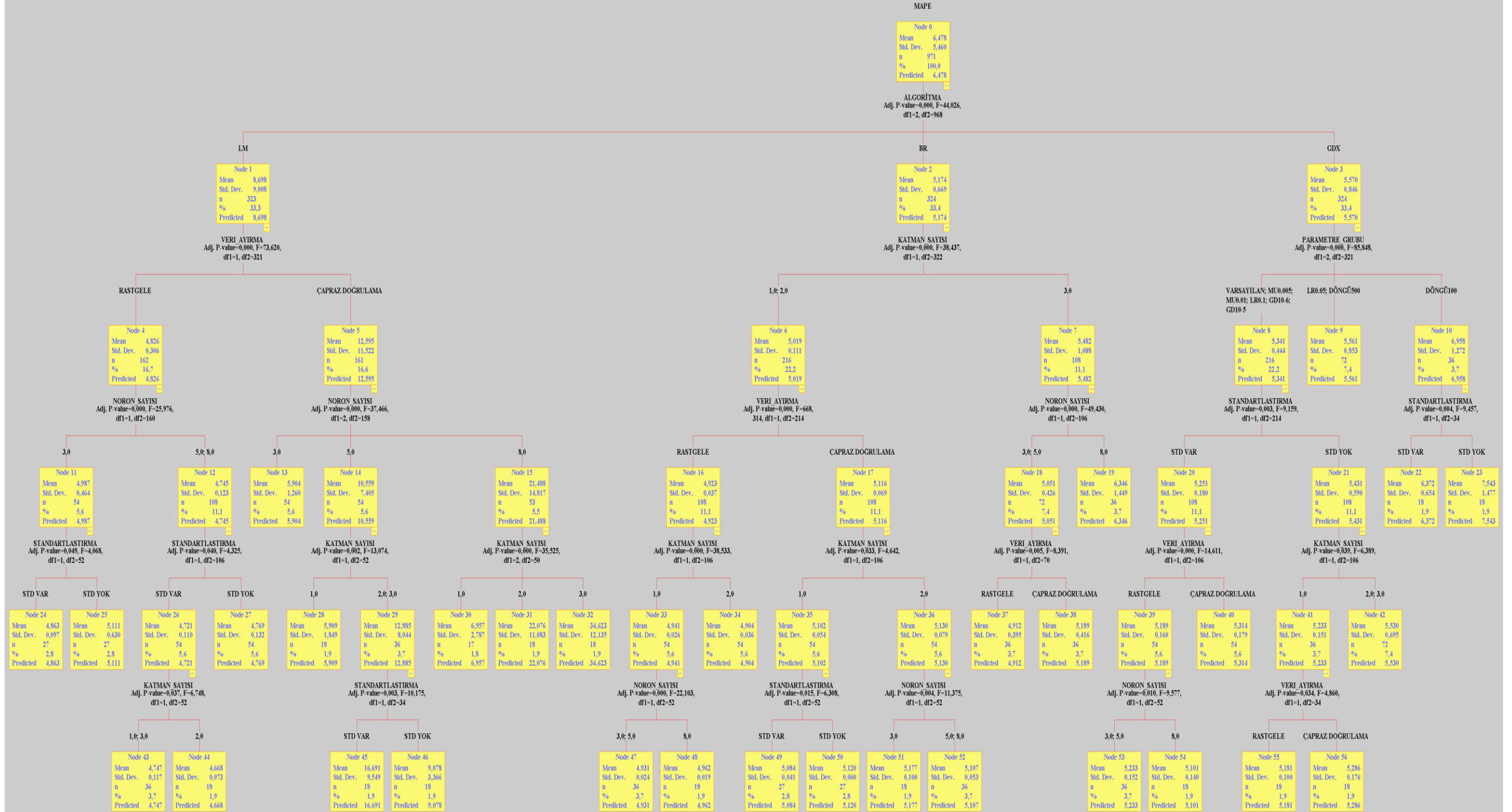
Rastgele olarak verilerin ayrıldığı durumdaki eğitimler sonucu süreler ve en iyi döngü sayılarına ait özet tablo.

PARAMETRELER					Levenberg Marquardt		Bayesian Regulation		GDX	
Normalizasyon var	MU	λ	GD	Döngü	Süre(sn)	EDS	Süre(sn)	EDS	Süre(sn)	EDS
	0.001	0.01	10^{-7}	1000	2.2939	24	4.6270	450	0.7469	742
	0.005	0.01	10^{-7}	1000	2.4613	17	3.2566	263	0.8351	803
	0.01	0.01	10^{-7}	1000	2.3747	13	2.9846	291	0.6111	564
	0.001	0.05	10^{-7}	1000	2.5437	24	4.3269	423	0.6939	615
	0.001	0.1	10^{-7}	1000	2.4604	23	4.3233	347	0.7277	734
	0.001	0.01	10^{-6}	1000	2.6382	26	4.9048	438	0.7692	803
	0.001	0.01	10^{-5}	1000	2.5558	26	5.0294	472	0.6672	607
	0.001	0.01	10^{-7}	500	2.7510	25	4.4080	407	0.4814	429
	0.001	0.01	10^{-7}	100	0.8720	20	0.9680	100	0.1169	98
Normalizasyon yok	0.001	0.01	10^{-7}	1000	2.6566	48	3.7874	411	0.7910	847
	0.005	0.01	10^{-7}	1000	2.7726	50	3.6541	309	0.7666	570
	0.01	0.01	10^{-7}	1000	2.7396	16	3.7817	341	0.7389	733
	0.001	0.05	10^{-7}	1000	2.5432	16	4.8633	437	0.7254	660
	0.001	0.1	10^{-7}	1000	2.3927	16	4.6157	437	0.7301	751
	0.001	0.01	10^{-6}	1000	2.4652	34	5.2132	467	0.6827	644
	0.001	0.01	10^{-5}	1000	3.3928	12	3.3683	316	0.8732	868
	0.001	0.01	10^{-7}	500	2.4907	16	3.1258	297	0.4783	432
	0.001	0.01	10^{-7}	100	0.8867	29	1.0192	98	0.1246	98

-**MU**: Momentum güncelleme katsayısı (Momentum update) - λ : Öğrenme oranı (Learning rate) -**GD**: Minimum gradyan değeri -**Döngü**: Müsaade edilen en çok döngü sayısı (epoch) -**Süre**: Eğitimde geçen süre – **EDS**: Eğitimin durdurulduğu en iyi döngü sayısı (best epoch)

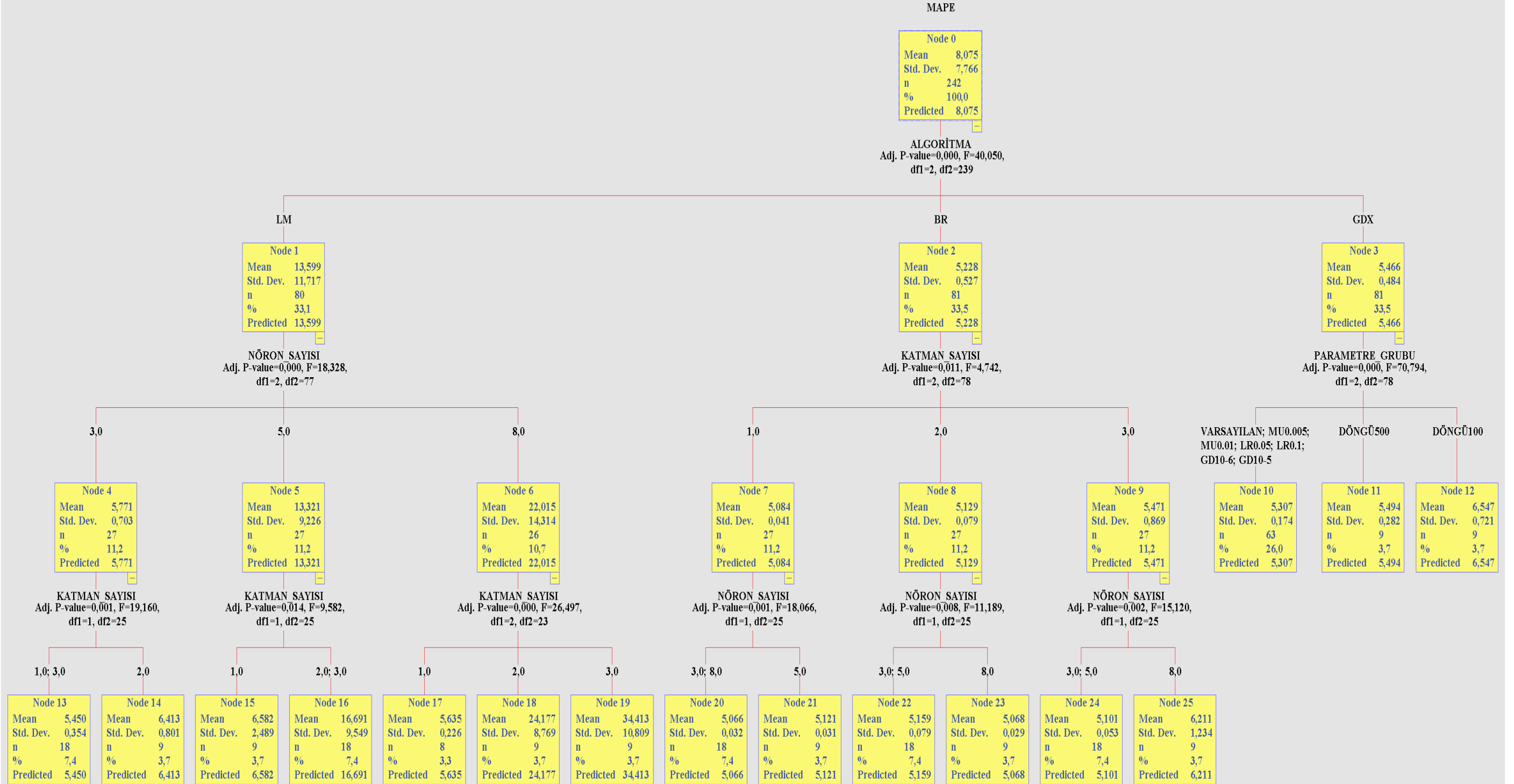
EK-4

Yapay sinir ağlarında yapılan bütün eğitimlere ait MAPE değerleri bakımından karşılaştırmalar için CHAID analizi bulguları.



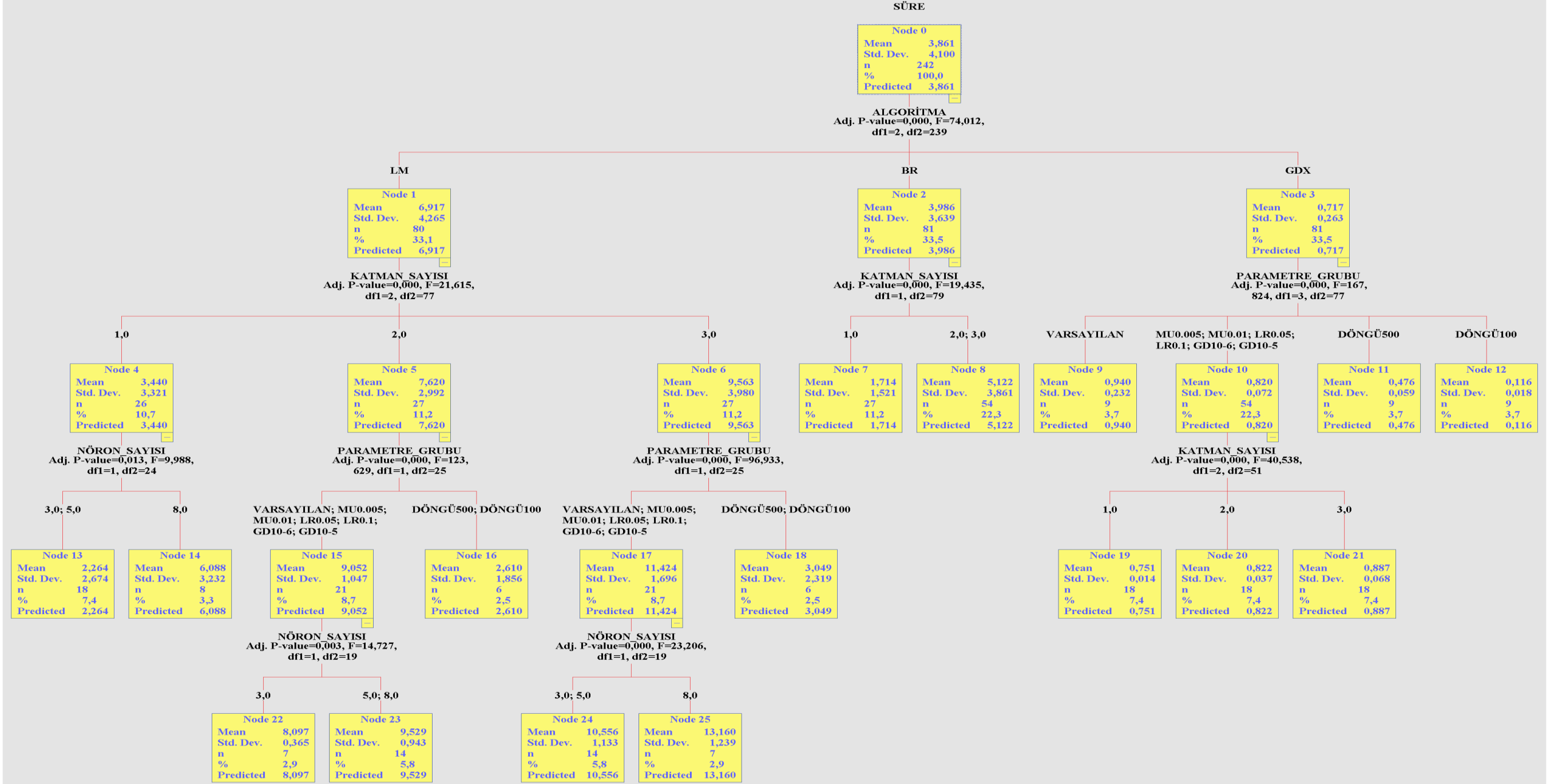
EK 5a

Çapraz doğrulama ile veri ayırma durumunda MAPE değerleri bakımından karşılaştırmalara ait CHAID analizi bulguları



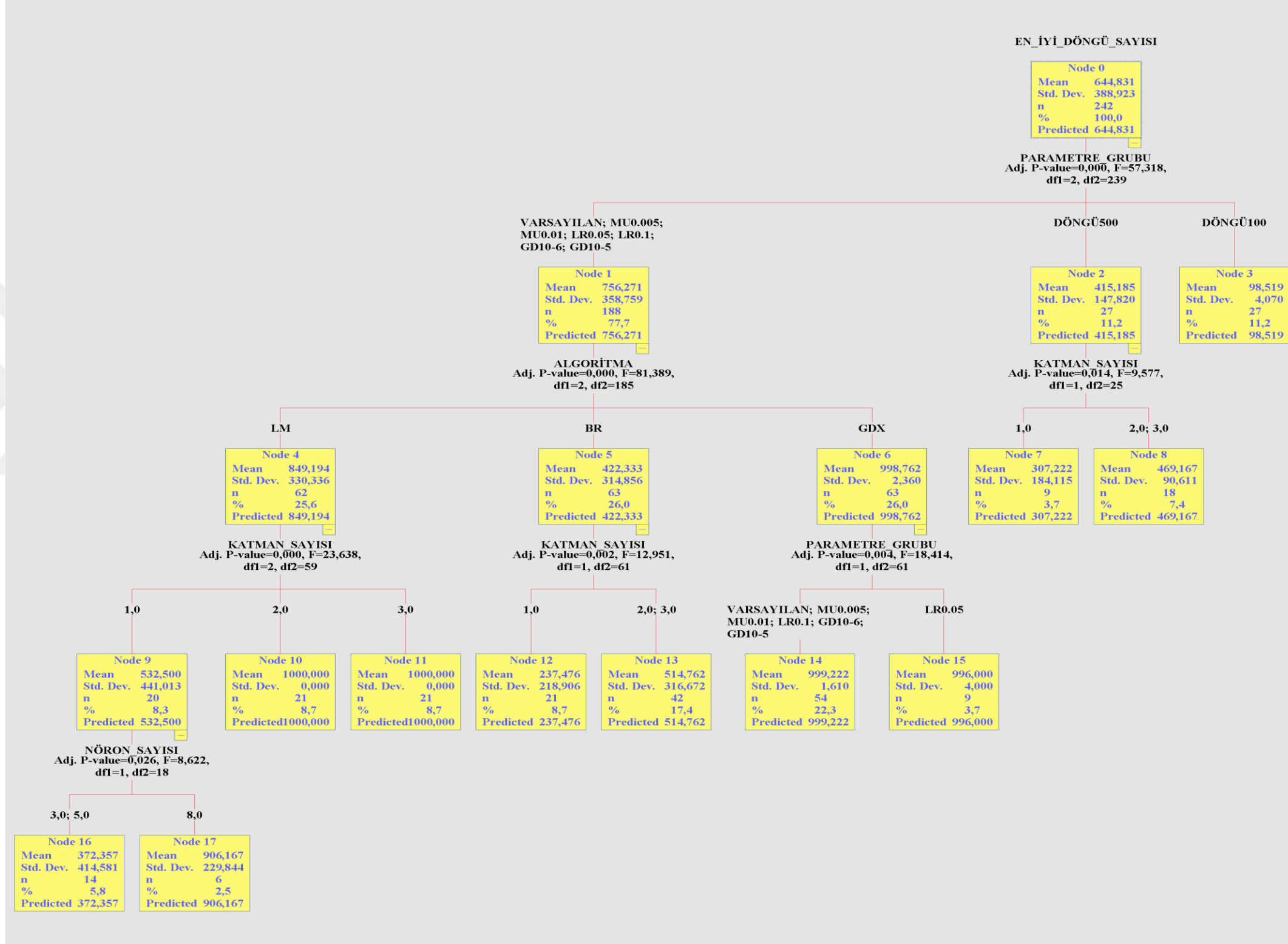
EK 5b

Çapraz doğrulama ile veri ayırma durumunda eğitimde geçen süreler bakımından karşılaştırmalara ait CHAID analizi bulguları



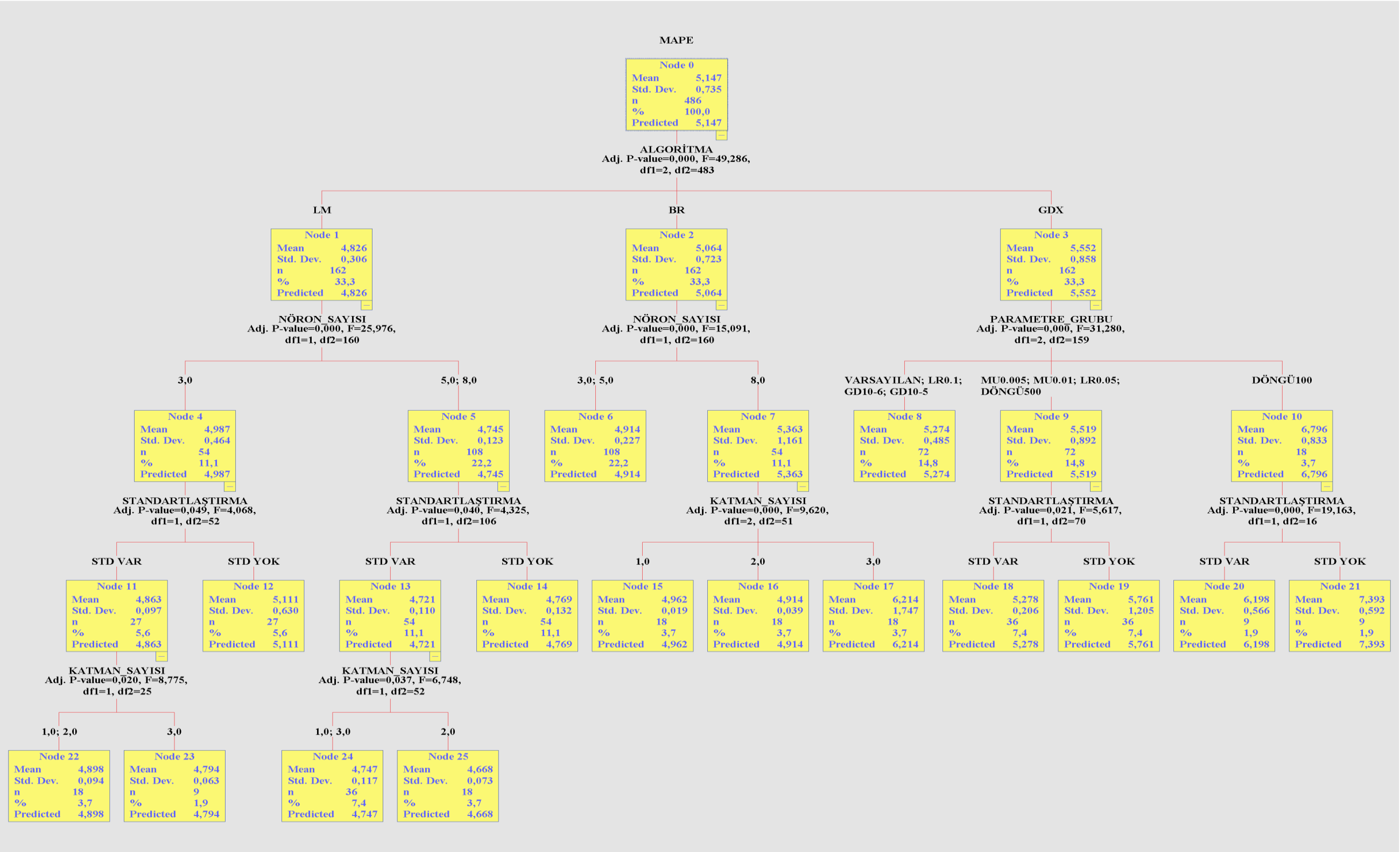
EK 5c

Çapraz doğrulama ile veri ayırma durumunda en iyi döngü sayıları bakımından karşılaştırmalara ait CHAID analizi bulguları



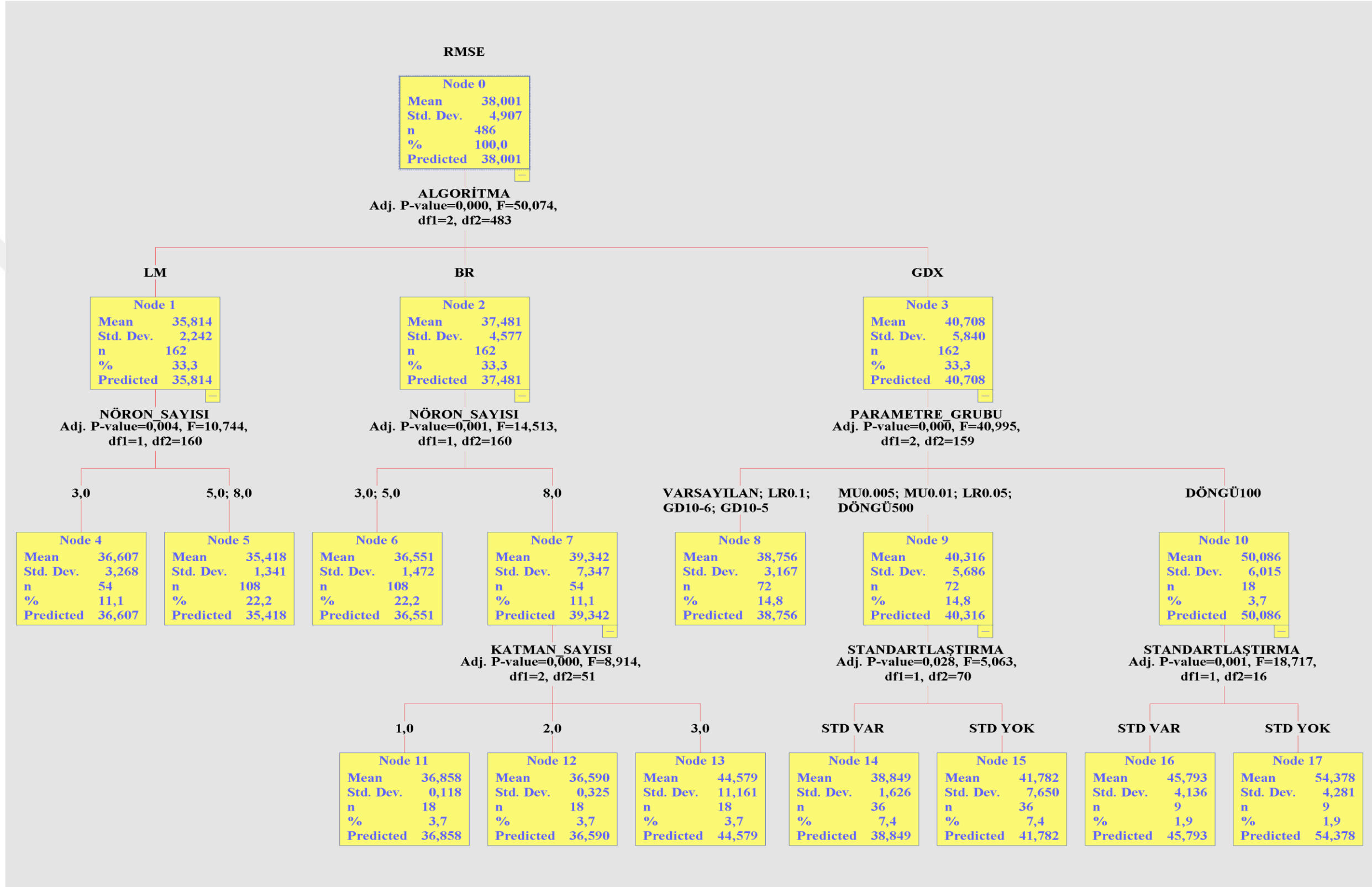
EK 6a

Verilerin rastgele olarak %70 eğitim %15 test %15 doğrulama setine ayrıldığı durumda MAPE değerlerine ait karşılaştırmalar için CHAID analizi bulguları



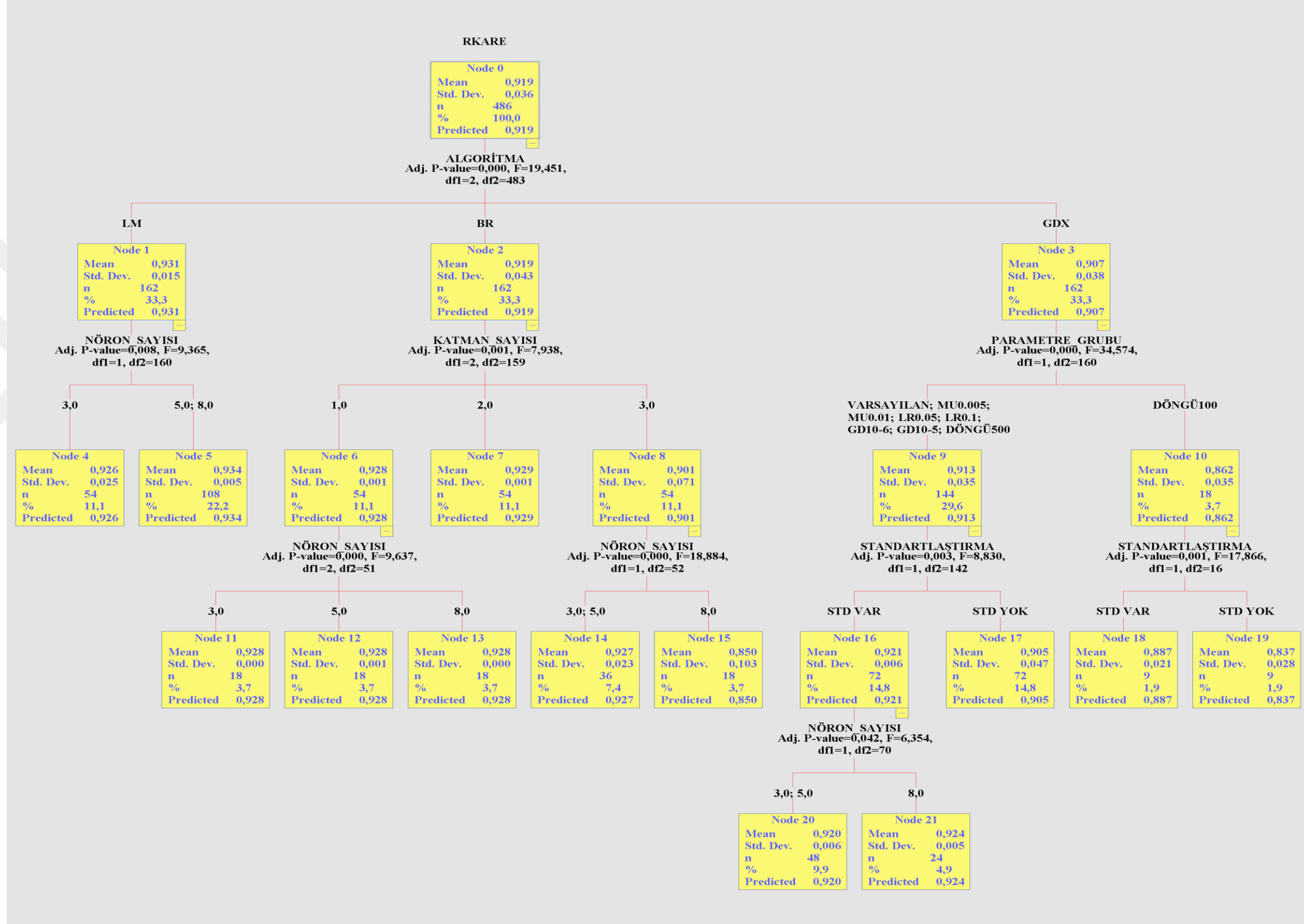
EK 6b

Verilerin rastgele olarak %70 eğitim %15 test %15 doğrulama setine ayrıldığı durumda RMSE değerlerine ait karşılaştırmalar için CHAID analizi bulguları



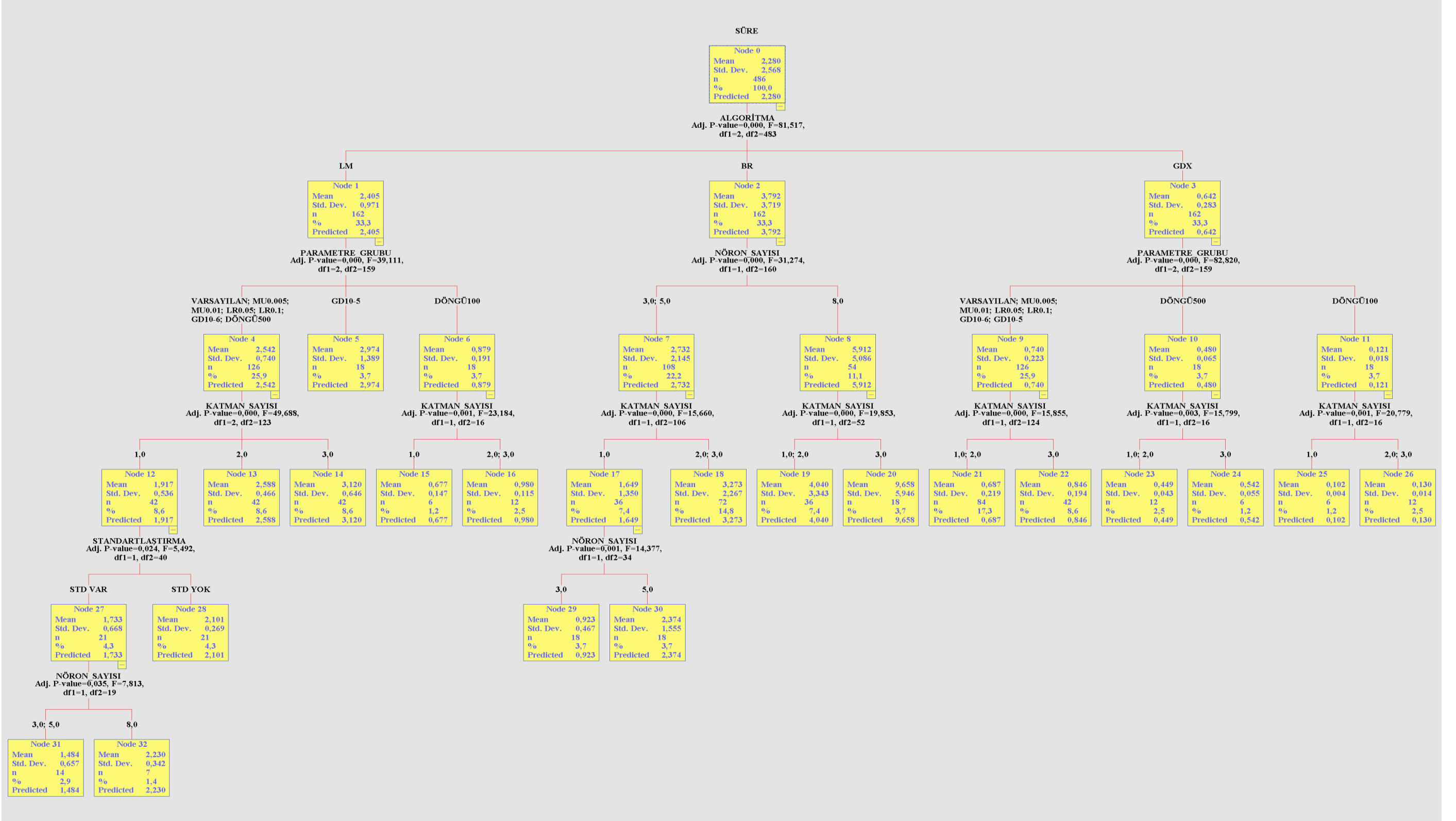
EK 6c

Verilerin rastgele olarak %70 eğitim %15 test %15 doğrulama setine ayrıldığı durumda R^2 değerleri bakımından karşılaştırmalar için CHAID analizi bulguları



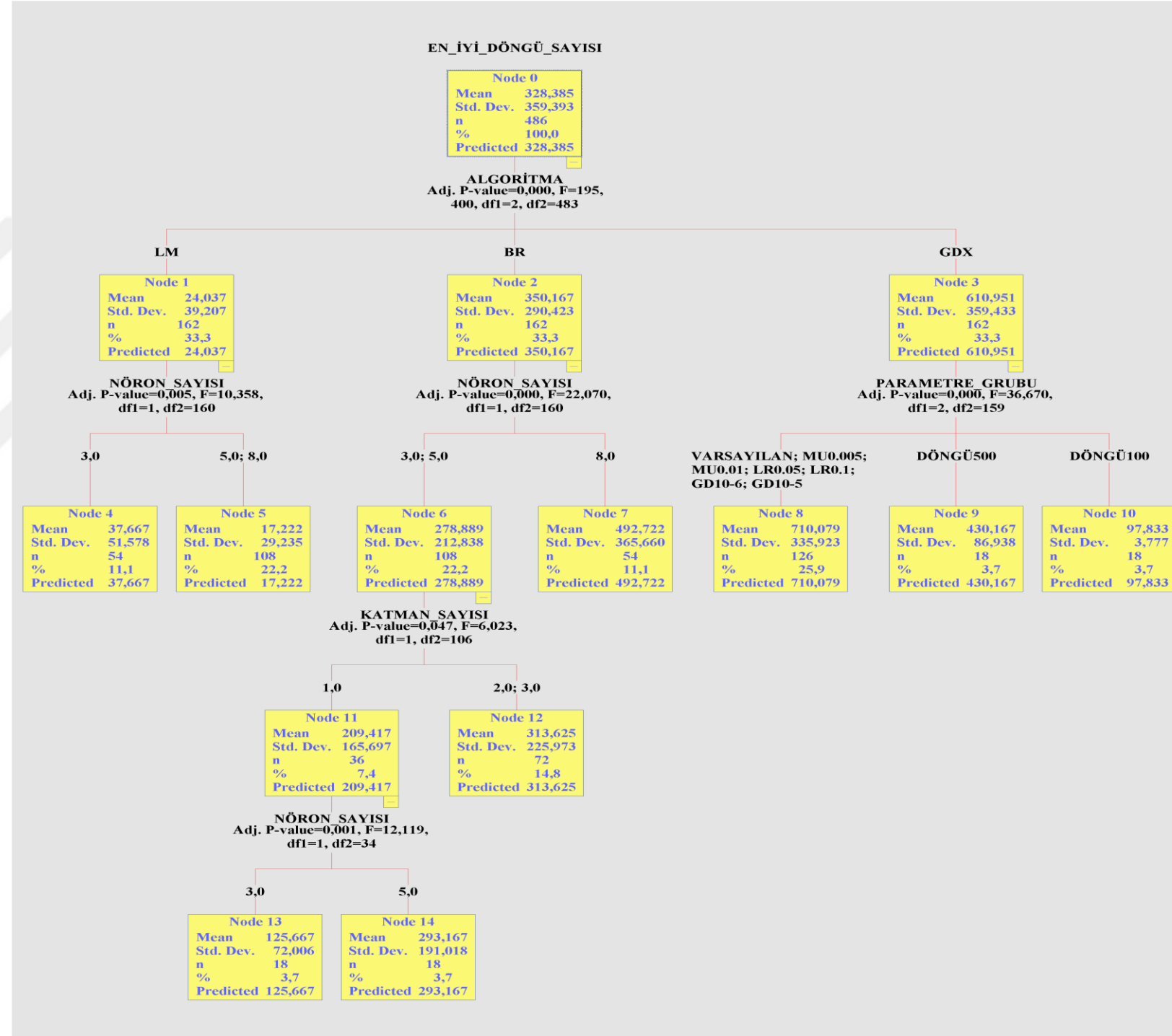
EK 6d

Verilerin rastgele olarak %70 eğitim %15 test %15 doğrulama setine ayrıldığı durumda süreler bakımından karşılaştırmalara ait CHAID analizi bulguları



EK 6e

Verilerin rastgele olarak %70 eğitim %15 test %15 doğrulama setine ayrıldığı durumda en iyi döngü sayıları bakımından karşılatırmalara ait CHAID analizi bulguları



EK 7a

Salford Predictive Modeler Programı Çok Değişkenli Uyarlanabilir Regresyon Eğrileri Analiz Sonucu.

Salford Predictive Modeler v8.3 64 bit - [Classic Output (Ctrl+Alt+C)]

File Edit View Explore Model Report Window Help

[Diagnostics - 5-BF Model](#)
[Test Sample Residual Fit Diagnostics - 5-BF Model](#)
[Timing Information](#)

MARS 2

[Start](#)
[MARS Results](#)
[Distribution of Y](#)
[Forward Stepwise Knot Placement](#)
[Final Model](#)
[ANOVA Decomposition](#)
[Variable Importance](#)
[MARS Regression: Training Data](#)
[Basis Functions](#)
[Selector Info](#)
[Regression Performance Summary](#)
[Performance By Outlier Trimming](#)
[% of Error Stats Due To Outliers \(Lrn\)](#)
[% of Error Stats Due To Outliers \(Tst\)](#)
[Learn Sample Residual Fit](#)
[Diagnostics - 5-BF Model](#)
[Test Sample Residual Fit Diagnostics - 5-BF Model](#)
[Timing Information](#)

DOF Penalty = 3

BasFn	TotVar	DirVar	EffPar	GCV	Learn MSE	Test MSE	Adj MSE	RMSE
11	2	2	34.00001	1729.18305	1231.86453	1430.47497	1164.05547	35.09793
10	2	2	31.00001	1678.16550	1234.82174	1458.77865	1172.51422	35.14003
9	2	2	28.00001	1635.15999	1242.09202	1506.66793	1185.11532	35.24333
8	2	2	25.00001	1593.60902	1249.05828	1527.84864	1197.49166	35.34202
7	2	2	22.00001	1554.22634	1256.35607	1514.96511	1210.25126	35.44511
6	2	2	19.00001	1535.20904	1279.26473	1243.15969	1238.18742	35.76681
** 5	2	2	16.00001	1545.24350	1326.74225	1085.43022	1290.22641	36.42447
4	2	2	13.00001	1533.46841	1356.03079	1257.54154	1324.92916	36.82432
3	2	2	10.00001	1492.94840	1359.12206	1185.35835	1334.18404	36.86627
2	2	2	7.00001	1517.96079	1422.04216	1332.25173	1402.47277	37.70997
1	1	1	4.00001	1583.76276	1526.17639	1441.97195	1512.17478	39.06631
0	0	0	1.00000	19153.60547	18978.28676	16673.78628	.	137.76170

=====

Regression Performance Summary

=====

Sample	Joint N	Wgt	Joint N	Mean(Score)	Mean(Target)	R-Sq	RMSE	MSE	MAD	MAPE	Norm R-Sq	SSY	SSE
Lrn	218	218.00	599.08257	599.08257	0.93009	36.42447	1326.74224	29.67326	0.05017	0.93009	.413727E+07	.289230E+06	
Tst	26	26.00	626.02884	618.26923	0.93343	32.94587	1085.43047	28.54764	0.04746	0.94340	.423947E+06	28221.19223	

=====

Performance By Abs(Deviation) Outlier Trimming

=====

Percentile	Joint N	Wgt	Joint N	Mean(Score)	Mean(Target)	R-Sq	RMSE	MSE	MAD	MAPE	Norm R-Sq	SSY	SSE
Lrn 100%	218	218.00	599.08257	599.08257	0.93009	36.42447	1326.74224	29.67326	0.05017	0.93009	.413727E+07	.289230E+06	
99%	216	216.00	597.70124	596.73611	0.93256	35.17992	1237.62680	28.98289	0.04949	0.93269	.396412E+07	.267327E+06	
98%	214	214.00	596.21444	594.41589	0.93433	34.29804	1176.35569	28.42935	0.04897	0.93479	.383365E+07	.251740E+06	
97.5%	213	213.00	595.43557	594.01408	0.93597	33.91493	1150.22280	28.17731	0.04863	0.93625	.382629E+07	.244997E+06	
97%	212	212.00	594.28731	592.47642	0.93590	33.53504	1124.59893	27.92752	0.04844	0.93644	.371952E+07	.238415E+06	
96%	210	210.00	591.70958	590.61905	0.93764	32.83509	1078.14319	27.45589	0.04796	0.93777	.363072E+07	.226410E+06	
95%	208	208.00	591.18771	590.07212	0.94046	32.20683	1037.27967	27.01819	0.04734	0.94055	.362357E+07	.215754E+06	
90%	197	197.00	585.27705	583.78173	0.94932	29.46627	868.26118	24.97220	0.04485	0.94948	.337501E+07	.171047E+06	
80%	175	175.00	587.06199	584.62857	0.96171	25.52336	651.44169	21.72214	0.03879	0.96209	.297705E+07	.114002E+06	

Estimated Number of Records: 245

C:\Users\AhmetErhan\Desktop\Kitap1.xlsx

16:23
17.06.2019

EK 7b

Salford Predictive Modeler Programı Çok Değişkenli Uyarlanabilir Regresyon Eğrileri Analizi Temel Fonksiyonlar Çıktısı.

Salford Predictive Modeler v8.3 64 bit - [Classic Output (Ctrl+Alt+C)]

File Edit View Explore Model Report Window Help

[Diagnostics - 5-BF Model](#)
[Test Sample Residual Fit Diagnostics - 5-BF Model](#)
[Timing Information](#)

MARS 2

[Start](#)
[MARS Results](#)
[Distribution of Y](#)
[Forward Stepwise Knot Placement](#)
[Final Model](#)
[ANOVA Decomposition](#)
[Variable Importance](#)
[MARS Regression: Training Data](#)
[Basis Functions](#)
[Selector Info](#)
[Regression Performance Summary](#)
[Performance By Outlier Trimming](#)
[% of Error Stats Due To Outliers \(Lrn\)](#)
[% of Error Stats Due To Outliers \(Tst\)](#)
[Learn Sample Residual Fit](#)
[Diagnostics - 5-BF Model](#)
[Test Sample Residual Fit Diagnostics - 5-BF Model](#)
[Timing Information](#)

Variable	Importance	-gcv
X1	100.00000	7249.20911
X2	11.10885	1615.63417

=====

MARS Regression: Training Data

=====

W: 218.00 R-SQUARED: 0.93009
 MEAN DEP VAR: 599.08257 ADJ R-SQUARED: 0.92844
 UNCENTERED R-SQUARED = R-0 SQUARED: 0.99649

Parameter	Estimate	S.E.	T-Value	P-Value
Constant	268.02099	11.82282	22.66981	0.00000
Basis Function 1	1690.45999	95.33554	17.73169	0.00000
Basis Function 3	3.22929	0.86761	3.72206	0.00025
Basis Function 5	-461.83925	132.73775	-3.47934	0.00061
Basis Function 15	-25.78141	11.91744	-2.16333	0.03163
Basis Function 17	22.67461	10.00038	2.26737	0.02438

F-STATISTIC = 564.10768 S.E. OF REGRESSION = 36.93632
 P-VALUE = 0.00000 RESIDUAL SUM OF SQUARES = .289230E+06
 (MDF,NDF) = [5, 212] REGRESSION SUM OF SQUARES = .384804E+07

=====

Basis Functions

=====

BF1 = max(0, X1 - 0.209354);
 BF3 = max(0, X2 - 138.49);
 BF5 = max(0, X1 - 0.360215);
 BF15 = max(0, X2 - 153.662);
 BF17 = max(0, X2 - 152.146);

Y = 268.021 + 1690.46 * BF1 + 3.22929 * BF3 - 461.836 * BF5
 - 25.7811 * BF15 + 22.6744 * BF17;

MODEL Y = BF1 BF3 BF5 BF15 BF17;