

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

**COVID 19 SÜRECİ İÇİN YAPAY SİNİR AĞLARI İLE
AMORTİSÖR SATIŞ TAHMİNİ**

YÜKSEK LİSANS TEZİ

Sena BALKIŞLI

Enstitü Anabilim Dalı	: ENDÜSTRİ MÜHENDİSLİĞİ
Enstitü Bilim Dalı	: MÜHENDİSLİK YÖNETİMİ
Tez Danışmanı	: Dr. Öğr. Üyesi Berrin DENİZHAN

Ocak 2022

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

**COVID 19 SÜRECİ İÇİN YAPAY SİNİR AĞLARI İLE
AMORTİSÖR SATIŞ TAHMİNİ**

YÜKSEK LİSANS TEZİ

Sena BALKIŞLI

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Enstitü Bilim Dalı : MÜHENDİSLİK YÖNETİMİ

Bu tez 25/01/2022 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

Jüri Başkanı

Üye

Üye

BEYAN

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

Sena BALKIŞLI

25.12.2021

TEŞEKKÜR

Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Dr. Öğr. Üyesi Berrin Denizhan’a teşekkürlerimi sunarım.

İÇİNDEKİLER

TEŞEKKÜR.....	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ	iv
ŞEKİLLER LİSTESİ	v
TABLOLAR LİSTESİ.....	viii
ÖZET.....	ix
SUMMARY	x
BÖLÜM 1.	
GİRİŞ	1
1.1. Araştırmanın Amacı.....	3
1.2. Araştırmanın Yöntemi	4
1.3. Çalışmanın Önemi	22
1.4. Çalışmanın Varsayımları	23
1.5. Çalışmanın Kapsam ve Sınırlılıkları.....	23
BÖLÜM 2.	
KAYNAK ARAŞTIRMASI	24
2.1. Amortisörler Hakkında Yapılan Araştırmalar	24
2.2. Yapay Zeka ile Talep Tahminleme Araştırmaları (2016-2021 Arası)....	24
2.3. WEKA Kullanımı ile Tahmin ve Analiz Araştırmaları.....	29
2.4. Matlab Kullanımı ile Tahmin ve Analiz Araştırmaları.....	30

BÖLÜM 3.

MATERYAL VE YÖNTEM	34
3.1. WEKA Analiz Programı (Waikato Environment for Knowledge Analysis).....	34
3.2. MATLAB Analiz Programı	55
3.3. Veri Setinin Oluşturulması	56
3.4. Veri Setinin WEKA'ya İşlenmesi	57
3.5. WEKA ile Tahmin Çözümü	69
3.6. Veri Setinin MATLAB'a İşlenmesi.....	69

BÖLÜM 4.

ARAŞTIRMA BULGULARI	89
4.1. WEKA ile Elde Edilen Tahmin Sonuçları.....	91
4.2. MATLAB ile Elde Edilen Tahmin Sonuçları.....	92
4.3. WEKA ve MATLAB Programları ile Ulaşılan Tahminlerin Kıyaslanması	94

BÖLÜM 5.

TARTIŞMA VE SONUÇ	100
KAYNAKLAR	104
ÖZGEÇMİŞ	108

SİMGELER VE KISALTMALAR LİSTESİ

GSYİH	: Gayri Safi Yurt içi Hasıla
K-NN	: K En Yakın Komşu Sınıflandırma Algoritması
LSTM	: Long/Short Term Memory (Uzun/Kısa Dönem Bellek)
MAPE	: Ortalama Mutlak Hata Yüzdesi
MLP	: Hata Kareleri Ortalaması
MSE	: Temel Bileşenler Analizi
PCA	: Çok Katmanlı Algılayıcı
SAÜ	: Sanayi Üretim Endeksi
SVM	: Destek Vektör Makinesi
TÜFE	: Tüketici Fiyatları Endeksi
XOR	: Eşdeğerliğin Olumsuzlanması Sorunu
Yİ-ÜFE	: Yurt içi Üretici Fiyat Endeksi
YSA	: Yapay Sinir Ağı
YZ	: Yapay Zeka

ŞEKİLLER LİSTESİ

Şekil 1.1. Biyolojik Beyin Nöronu Yapısı	8
Şekil 1.2. Çok Katmanlı Yapay Sinir Ağı Nöronu Çalışma Fonksiyonu (F. Rosenblatt'ın 1958 Yılında Tanıttığı Model)	9
Şekil 1.3. İleri Beslemeli Yapay Sinir Ağı Katmanları.....	10
Şekil 1.4. Sigmoid Fonksiyonu Grafiği	11
Şekil 1.5. Derin sinir ağlarını genel yapısı (Mira,1995:11)	14
Şekil 1.6. Derin Öğrenme ve Geleneksel Makine Öğrenmesi Yaklaşımının Karşılaştırılması.....	16
Şekil 1.7. Katmanlar, kayıp fonksiyonu ve eniyileme algoritması arasındaki ilişki	17
Şekil 3.1. WEKA programının ana penceresi.	35
Şekil 3.2. WEKA'da desteklenen dosya biçimleri.....	36
Şekil 3.3. WEKA veri Kaydetme	37
Şekil 3.4. WEKA'da yer alan Düzenlenebilir Yapay Sinir Ağının (YSA) görünümü.....	40
Şekil 3.5.WEKA Programının Sınıflandırma Sekmesinde Eğitim Verileri ve Test Bileşenlerinin Ekran Görüntüsü	41
Şekil 3.6. Veri Setinden Eğitim Sınıfı Olarak Tanımlanan Küme ve Modelin Eğitim Çıktısı Ekran Görüntüsü	41
Şekil 3.7. Ağaç görselleştirmenin sonucu	43
Şekil 3.8. WEKA Programında Görselleştir sekmesi ve PlotSize Aracının K Değeri Tespiti	47
Şekil 3.9. K-NN temelli çift özelliği ve Yakın Değerlerin Görüntülenmesi.....	48
Şekil 3.10.WEKA programı Nitelikler sekmesi ekran görüntüsü.....	49
Şekil 3.11. WEKA Programında 2 Veriseti, 4 Algoritma, 3 parçalık verisetleri ve 2 Yinelemeli Kontrol Seçimini Gösterir Ekran Görüntüsü	50

Şekil 3.12. Experimenter modülünün Analiz sekmesi (deneyler hakkında TeX raporu görüntüsü).	51
Şekil 3.13. Veri Setindeki Test Grubunun Rapor Çıktısı.....	52
Şekil 3.14. WEKA Bilgi Yığını Tarama ve Sınıflandırması.....	53
Şekil 3.15. WEKA’da üretilebilecek örnek K-NN geri çağırma kodu	54
Şekil 3.16. Veri Setinin .arff formatına dönüştürülmesi	57
Şekil 3.17. Verisetinin Weka Preprocess sekmesinde açılması	58
Şekil 3.18. Normalize edilmiş veriseti	59
Şekil 3.19. PCA ekran görüntüsü.....	60
Şekil 3.20. Preprocess sekmesinde PCA filtresinin uygulanması.....	61
Şekil 3.21. Görselleştirme ekran görüntüsü	61
Şekil 3.22. YSA modeline ait ayarlar.....	62
Şekil 3.23. YSA tahmin sonuçlarının yazdırılması komutu.....	63
Şekil 3.24. Kurulan YSA modeli ekran görüntüsü	63
Şekil 3.25. YSA tahmin sonuçları ve hataları	64
Şekil 3.25. YSA tahmin sonuçları ve hataları (Devam).....	65
Şekil 3.26. YSA tahminlerine ait hatanın görselleştirilmesi.....	65
Şekil 3.27. Kurulan YSA’ya ait korelasyon katsayısı ve hata oranları	65
Şekil 3.28. Aylık satış tahminleri ile gerçekleşen satışlar tablosu	66
Şekil 3.29. Tahmini yapılacak pandemi 3 ayına ait verilerin .arff formatına dönüştürülmesi	67
Şekil 3.30. Kaydedilen YSA modelinin Weka’ya tekrar çağırılması	67
Şekil 3.31. InputMappedClassifier seçeneği.....	68
Şekil 3.32. YSA tahmin sonucu	68
Şekil 3.33. Verinin Okunması ve Normalleştirilmesi (Ekran Görüntüsü).....	70
Şekil 3.34. Normalleştirilen Verinin Korelasyon Matris Hesabı ve Temel Bileşen Analizi	70
Şekil 3.35. Korelasyon Matrisi	71
Şekil 3.36. Eigen Değerleri ve Eigen Vektörleri.....	71
Şekil 3.37. Aşırı Uyum Örneği	72
Şekil 3.38. Sinir Ağının Başlangıç Değerlerinin ve Parametrelerinin Ayarlanması.....	73

Şekil 3.39. Network Veri Tipinin Parametreleri	74
Şekil 3.40. Ağın Eğitilmesi ve Verilerin Sınıflarına Ayrılması	76
Şekil 3.41. Hedef ile Çıktı Arasındaki Farkın Grafiğe Yazdırılması	77
Şekil 3.42. Yapay Sinir Ağının Diyagramı	78
Şekil 3.43. İstenen Değerlerin ve Sonuç Değerlerinin Grafikselsel Olarak Karşılaştırılması.....	78
Şekil 3.44. Performans, Eğitim Durumu, Histogram ve Regresyon Grafiklerinin Çizdirilmesi	79
Şekil 3.45. Performans Grafiği	80
Şekil 3.46. Eğitim Parametrelerinin Son Durumu	81
Şekil 3.47. Hata Histogramı	82
Şekil 3.48. Regresyon Grafikleri.....	83
Şekil 3.49. Yeni Verinin MATLAB'a Tanıtımı, Normalizasyonu ve Temel Bileşen Analizi	85
Şekil 3.50. Yeni Verinin Yapay Sinir Ağına Uygulanması ve Hata Hesabı.....	86
Şekil 3.51. Yeni Verinin Hata, Histogram ve Regresyon Grafiklerinin Yazdırılması	86
Şekil 3.52. Yeni Veri ile Tahmin Edilen Değer ile Gerçek Değerin Grafiği	87
Şekil 3.53. Yeni Veri ile Tahmin Edilen Değerin Gerçek Değere Göre Regresyonu	87
Şekil 3.54. Yeni Veri ile Tahmin Edilen Değerin Hata Histogramı	88
Şekil 4.1. MATLAB çıktı ekranı görselleri	93
Şekil 4.2. MatLab Çıktı Grafiği	93

TABLolar LİSTESİ

Tablo 3.1. WEKA YSA tahmin sonucu ile gerek satıř deęerlerinin karřılařtırılması.....	69
Tablo 3.2. Toplam Veri İstatistięi.....	82
Tablo 3.3. Veri Seti Hata Deęerleri	82
Tablo 3.4. Tm Veri Seti İin Regresyon Deęerleri	83
Tablo 3.5. Veri Sınıflarının mutlak ortalama hatası ve hatanın standart sapması...	84
Tablo 3.6. Ortalama Hata Deęeri ve Hatanın Standart Sapmasının Sunumu	88
Tablo 4.1. Gerekleřtirilen Amortisr Satıřları ve Baęımsız Deęiřkenler	90
Tablo 4.2. MatLab YSA Tahmin Sonucu ve Gerek Satıř Deęerleri ile Karřılařtırılması.....	94

ÖZET

Anahtar kelimeler: Amortisör Satışları, WEKA, MatLab, YSA ve Tahmin Parametreleri, Covid 19

Günümüzde çevik yaklaşımla yönetilen birçok firma verilerini doğru şekilde değerlendirmek ve karar verme sürecinde yardımcı birer olarak kullanmak durumundadır.

Covid 19 pandemisi ile birlikte birçok sektörde olduğu gibi otomotiv endüstrisi de olumsuz etkilenmiştir. Bu çalışmada, bir amortisör firmasına ait, Türkiye için erken pandemi dönemi olan 2020 yılı Mart, Nisan ve Mayıs ayları satışlarının ne olacağı tahminlemesi yapılmış ve elde edilen tahmin değerleri ile gerçekleşen satışlar arasındaki fark kıyaslanarak pandeminin olumsuz etkisi yansıtılmak istenmiştir.

WEKA ve MatLab programlarının YSA yapıları, tahmin parametreleri ve model eğitimleri 2016 yılı Ocak ayı ile 2020 yılı Şubat ayı arası aylık satış dataları bir amortisör firmasından elde edilerek kullanılmıştır. Pandemi dönemini kapsayan 3 aylık gerçekleşen satış rakamları ise, firmadan alınarak gerçekleşen satış olarak adlandırılmış ve WEKA ile MatLab satış tahmini sonuçlarıyla karşılaştırılmıştır.

Sonuç olarak, amortisör satışlarının pandemi dönemi öncesinde artış gösteren bir eğilimde olduğu ancak pandemi dönemine ait yapılan tahminler ile gerçekleşen satışlar kıyaslandığında amortisör satışlarının pandemiden oldukça olumsuz etkilendiği gözlenmiştir. Araştırmada elde edilen bulgulara göre, WEKA yapısının tahminleme sonuçlarında daha büyük bir hata payı olduğu ve kurulan modelde daha fazla YSA barındıran MatLab'ın daha başarılı tahmin ürettiği sonucuna varılmıştır. Ancak her iki programın, tahmini yapılan pandemi dönemi için hata oranlarının bir tahmin çalışması kapsamında yüksek oluşunun temel sebebi olarak, programlarda kriz dönemlerine ait bir tanım bulunmamasının, dolayısı ile kurulan YSA modellerine etkisinin yansıtılamamış olmasının sebep olmuş olacağı öngörülmüştür.

SHOCK ABSORBER SALES FORECAST WITH ARTIFICIAL NEURAL NETWORKS FOR THE COVID 19 PROCESS

SUMMARY

Keywords: Shock Absorber Sales, WEKA, MatLab, ANN and Forecast Parameters, Covid 19

Today, many companies managed with an agile approach have to evaluate their data correctly and use it as an additional resource in the decision-making process.

With the Covid-19 pandemic, the automotive industry has been adversely affected, as in many other sectors. In this study, an estimate of the sales of a shock absorber company for March, April and May 2020, which is the early pandemic period for Turkey, has been made and the negative impact of the pandemic has been tried to be reflected by comparing the difference between the estimated values obtained and the actual sales.

ANN structures, estimation parameters and model trainings of WEKA and MatLab programs were used by obtaining monthly sales data from January 2016 to February 2020 from a shock absorber company. On the other hand, the actual 3 months sales figures covering the pandemic period, were taken from the company and named as actual sales and compared with the sales forecast results of WEKA and MatLab.

As a result, it was observed that the sales of shock absorbers had an increasing trend before pandemic period but when the forecasts for the pandemic period and actual sales were compared, the sales of shock absorbers were adversely affected by the pandemic. According to the findings obtained in the research, it was concluded that the WEKA structure has larger margin of error in the estimation results and MatLab, which includes more ANNs in the established model, produces more successful predictions. However, it was predicted that the main reason for the high error rates of both programs for the predicted pandemic period within the scope of an estimation study was that there was no definition of crisis periods in the programs, and therefore the effect could not be reflected on the established ANN models.

BÖLÜM 1. GİRİŞ

Amortisör makinalarda çalışırken ortaya çıkan sarsıntı ve titreşimin şiddetini ve ortaya çıkabilecek olan etkiyi azaltmak adına kullanılmakta olan elemanlar olarak tanımlanmaktadır. Amortisörler, hareket yönüne karşı hız ile orantılı bir direnç sağlamak üzere kullanılmaktadır. Her türlü vurmali makinalarda (tekstil makinaları, presler, iş makinaları, kaldırma makinaları, arabalar...) kullanılsa da en sık kullanıldığı yer araçlar olarak bilinmektedir. Amortisörlerin genel çalışma prensibinde, titreşimleri absorbe ederek, titreşimlere neden olan enerjiyi ısıya dönüştürmek bulunmaktadır. Genel kabulde en yaygın kullanılan makineler olarak araçların, amortisörlerin çalışma prensibi açıklamasında bulunması bu sebeple anlamlı görülmektedir. Amortisör, araçlarda, yoldaki titreşimleri aracın şasisine iletmeden emen, aracın süspansiyon sisteminde bulunan yaylar ile birlikte kullanılmasının yanı sıra, amortisörün rolü yaylardan başka özellikler sergilemektedir (Zheng ve ark., 2020).

Araç sistemlerinde amortisörlerin rolü, depolanan yay enerjisinin, titreşimler şeklinde şasiye iletilmeden amortisör emilmesi ile ilişkilidir. Şöyle ki; araçlardaki yaylar, yolun düzensizliğinden kaynaklanan darbe ve dalgalanmaların araç gövdesine iletilmesini engellemektedirler. Amortisör ise hem başlangıçta tekerlekten gelen enerjiyi hem de yayda depolanan enerjiyi ters yönde dirençle ısıya dönüştürmekte görev almaktadır. Böylece titreşimi absorbe ederek ortadan kaldırmaktadırlar.

Amortisörlerin yolu iyi tutmada önemli bir rolü bulunmaktadır. Yani araç kabiliyetinin bir parçası olarak görülmektedirler. İyi bir amortisörün, dönerken kaymayı önleme özelliği ile birlikte, tekerleklerin yere iyi temas etmesi ve sürüş sırasında zıplamamasını sağlaması özelliklerini barındırması beklenmektedir. Ayrıca, amortisörlerin araçlarda çekiş kabiliyetini arttırma ve fren yaparken durma mesafesini kısaltma görevleri de bulunmaktadır.

Tüm bunların yanı sıra, araçlarda kullanılan akışkan tipleriyle ilişkili olarak, farklı biçimlerde aynı görevi üstlenmek üzere kullanımları söz konusu olabilmektedir. Şöyle ki; araç akışkanları, sıvı veya gaz esaslı olarak sınıflandırılmaktadır. Bu akışkanlardan sıvı olanlarda, gaz tipindeki akışlara nazaran daha çok yağ kullanıldığı bilinmektedir.

Araçlarda kullanılan yağlar, iç moleküler yapısı gereği; yüksek viskozitesinden (kalınlık) kaynaklı olarak basınç altında, yağın dar bir kanaldan geçirilmesiyle ısıya dönüştürülmesi eylemini sergilemektedirler. Bu prensibe bağlı olarak enerji, toplanan moleküller arasındaki sürtünme tarafından emilmesiyle sonuçlanmaktadır. Amortisör seçimi de doğru yönlendirmeyi yapabilmek üzere, bu yapıya uygun olarak yapılmaktadır.

Gaz bazlı tipler aynı prensipte çalışmak üzere kurgulansa da gaz formundaki akışkanlar için daha fazla hava kullanılmaktadır. Bu da gaz bazlı amortisör tipinin yağ bazlı amortisör tipinden kullanımından farklı olarak seçilmesini gerekli kılmaktadır.

Her iki temel ilkeye dayalı olarak, imalat ve otomobillerde farklı amortisör tiplerinin kullanılmasını anlaşılabilir kılmaktadır. Araçlarda kullanılan amortisör tiplerinden en yaygın olanı teleskop türü olarak bilinmektedir.

Araçlarda amortisörün rolleri aynı olsa da araç kullanım alanı ve gereksinimlerine göre amortisör tipleri değişmektedir. Bununla birlikte gelişen araç pazarı için amortisörler halen teknik bileşenler arasında önemini korumaktadırlar. Araç bileşenleri olarak amortisörler, insanların kullanım alışkanlıkları ve amaçlarına göre kendi pazarını oluşturmuş, araçların üretimleri konusunda da fikir yürütmeye yardımcı malzemeler olarak kabul edilmektedir. Bilindiği üzere, yerli üretim olarak binek araçlarda henüz bir ivmelenmenin olduğu yurdumuzda, bu teknik araç parçasının satış grafiği kısmen yurtdışı kısmen de ihracat yurdumuzdaki başarımını da gösterecektir. Bu temel amaçla, bu çalışmada amortisör satış grafikleri kullanılarak, dünya çapında etkili olan pandemi sürecinde araç pazarının bir temsili sunulacaktır. Bunun için gerçek bir amortisör üretim firmasının satış verisi kullanılarak gelecek için satış tahmini yapılacak olup,

talep tahminlemede WEKA ve MATLAB yapay zeka programları ile bu tahminlerin sunumu sağlanarak, programların tahminlemedeki başarımı karşılaştırmalı olarak değerlendirilecektir.

Gerçek satış rakamları kullanılarak sonraki ayın tahminlemesi yapılarak doğrulama sağlanacak ve doğrulaması yapılmış verilerle gelecek 3 aylık pandemi dönemi için de talep tahminleri sergilenecektir.

1.1. Araştırmanın Amacı

Talebi tahmin etmenin ve doğruluğunun bir firmanın başarısı ve müşterisinin memnuniyeti ile doğrudan bağlantısı bulunmaktadır. Ürünün talepteki değişkenliği ve talebi etkileyen faktörlerin zaman içindeki dinamizmi, talebin tahmin edilmesini ve planlanmasını zorlaştırmaktadır. Talebi tahmin etmek, bu zorluklara rağmen, ayrıntılı talep tahminleriyle şirkete en fazla kârı getirecek stratejik hedefleri belirlemede en önemli rolü üstlenmektedir.

Talep tahmini aynı zamanda, işletmenin stratejik hedeflerine ulaşmak için kaynaklara ulaşma konusunda ve planlamanın sağlanmasında da önemli bir etken olarak kabul edilmektedir. Bu sebep tahminlerin doğruluğu, işletme kazançları ile doğru orantılı bir yaklaşımı tanımlamaktadır (Kuo ve ark., 2016).

İşletme kazançlarının planlaması, talep tahminlerinin işin ruhuna uygun olarak doğru biçimde sağlanmasına yönelik tüm eylemsel hareketlilik firmanın varlığını devam ettirmedeki önemini göstermektedir. Bu önemden hareketle bu çalışmada, talep tahminlerinin sağlanmasına yönelik bir yöntem geliştirmek amaçlanmaktadır. Tahmin modelinin geliştirilmesine yönelik olarak bir amortisör firmasına ait gerçek satış verileri ile bir paket program olan WEKA kullanılmış olup, daha sonra aynı verilerin MATLAB ortamında yapay zeka yaklaşımı ile işlenmesi sağlanmış ve her iki programın çıktıları değerlendirilmiştir.

Bu çalışma ile birlikte son dönemde popüler olan yapay sinir ağları ile otomobillerin önemli bir parçası olan amortisörlerin satış tahminlemesi yapılması ve pandemi döneminin ilgili pazar üzerindeki etkisinin ölçülmesi amaçlanmıştır.

1.2. Araştırmanın Yöntemi

Literatürde talebi tahmin etmenin birçok farklı yolu bulunmaktadır. Araştırma yöntemlerinden nitel ve nicel yöntemler, talep tahminleri için de kullanılmaktadır (Acito ve ark., 2021).

Karaatlı ve ark., (2012)'ye göre, talep tahmin değerlendirmeleri için kullanılan nitel ve nicel değerlendirme yöntemlerinin tanımı ile saha araştırmasını kapsamaktadır. Bu yöntem; biçim itibariyle, karma (Creswell ve Plano, 2014) yöntemde tanımlanan araştırmalarda olduğu üzere, alanında uzman kişilerin görüş ve deneyimlerine dayandırılarak açıklanmaktadır.

Ancak bu araştırmacıların kullandığı yöntem Yapay Sinir Ağının eğitimi için planlanmış bir eylem araştırmasını da çağrıştırmaktadır (Jarvinen, 2007).

Araştırmacılar, çalışmalarında nitel ve nicel araştırma yöntemlerini tanımlamanın yanı sıra, bir araba satış distribütörünün verilerini kullanarak, araç talep miktarının tahminlemesine dair bir anlayış sunmuşlardır (Karaatlı ve ark., 2012).

Başka bir tahminleme çalışmasında, sayısal tahmin yöntemi matematiksel bir modele dayalı olarak sunulmuştur. Örneğin bir talep tahmin yöntemi geliştirilmek istenen, Bolt (1994) tarafından yapılan çalışmada, nitel ve nicel araştırma yöntemi yerine öznel ve nesnel yöntemler tanımlarını kullanmıştır. Araştırmacıya göre, sübjektif yöntem, matematiksel verilerden ziyade deneyime, sonuçlara ve entelektüel kullanıma dayanan literatüre dayalı bilgilerin işlenmesine dair tanımlamasını kapsamaktadır. Yine araştırmacı tarafından objektif yöntem olarak tanımlanan bilgilerin ise, istatistiksel ve matematiksel verilere dayanan nicel araştırma yöntemini tanımladığı görülmektedir.

Tahminleme çalışmaları için hem nitel hem de nicel yöntemlerin birlikte kullanılmasının daha uygun olacağı yönündeki görüşler, literatürde yer almaktadır. Örneğin araştırma yöntemlerini inceleyen Tekin (1996)'ya göre bir tahminleme modelinin geliştirilmesi için en uygun araştırma tekniği, her iki yöntemin birleştirilmesinden geçmektedir.

Bu nedenle bu çalışmada nitel ve nicel araştırma yöntemin birleştirilmesine dayalı olan ve yapay zekâ ile modellenmiş bir tahmin biçiminin oluşturulması planlanmaktadır.

Çalışmada kullanılacak argümanlardan biri olan yapay zekaya dayalı hesaplama programlarının çalışma mantığını sergilemek üzere, yapay zekanın parçalarına ve işlem tekniklerine çalışmanın arka planı başlığında yer verilmesi uygun görülmektedir. Böylelikle, hesaplama tekniklerinde yapay zekaya dayalı programların kullanılması ve tahminleme güvenilirliğinin açıklığa kavuşturularak anlamlı kılınması hedeflenmektedir.

- Çalışmanın Arka Planı: Yapay Zekanın Tanımı, Gelişimi ve Önemi

Yapay zekanın başlangıcı ve tarihçesine dair bilgiler arasında, Cambridge Üniversitesi Profesörlerinden, Alfred North Whitehead ve Bertrand Russell tarafından 1925 yılında yazılan Principia Mathematica adlı üç ciltlik eser ile teorik mantığın temelleri atıldığı bilgisi yer almaktadır. 1936 yılında, Viyana Syntactic Structures'in yazarı Rudolf Canap mantıksal sözdizimi ile destek sunması ile geliştirilmesine katkı sunduğu görülmekte ise de çalışmaların o dönem için çokça ilgi gördüğünün bir yansıması olarak kabul edilmemektedir. Ancak yapay zekâ çalışmalarını kapsayan ilginin 1954 yılı soğuk savaş sırasında oluşturulduğu iddiası güçlü görülmektedir. 1954 yılından itibaren ABD ordusu Rus belgelerini ve akademik makalelerini otomatik olarak çevirmek için bir program istemesi, yapay zekâ çalışmalarının hız kazandığı yıl olarak görülmesini de anlamlı kılmaktadır.

Yapay zeka tanımlamaları daha geniş perspektiften ele alındığında ise, sibernetik tanımları ile karşılaşılmaktadır. Sibernetiklerin kâşifi sayılan Walter Pitts ve Warren

Mcculloch 1947 yılında yapay zeka ile kendini kontrol eden bir robot yaptıklarında ilk izlenimlerin oluştuğu varsayılmaktadır. Ancak 1951 yılında elektrik kontrolleri bilimi olan sibernetiklerin henüz ilgi uyandırmış olması, 1954 yılında tam çeviri işleminin sağlanamamış olması, 1969 yılında Marvin Minsky ve Seymour Papert tarafından yazılan An Introduction to Computational Geometry adlı kitaplarında XOR sorunu yani eşdeğerliğin olumsuzlanması sorunu nedeniyle yapay zeka teknolojilerine olan ilginin de azaldığını söylemeyi mümkün kılmaktadır.

Yapay zeka tarihçesinde XOR sorunu kısaca modern sinir ağlarındaki gibi işleyişin henüz olmadığı zamanlarda tüm grafiklerin doğrusal olarak verilmesi sorunu olarak tanımlanmıştır. XOR sorunu özetle; “İki boyutlu bir algılayıcı sisteminde sonucun doğrusal olması ve grafiğin herhangi bir eğri içermemesi sorunu” olarak bilinmektedir. XOR problemi doğrusal olmayan ve türevi alınabilen Sigmoid fonksiyonunun kullanılması ile çözüme kavuşmuş ve XOR sorununun çözülmesi ile birlikte çok katmanlı bir algılayıcı fikrinin ortaya çıktığı varsayımını geçerli kılmaktadır (Vapnik, 1963:771).

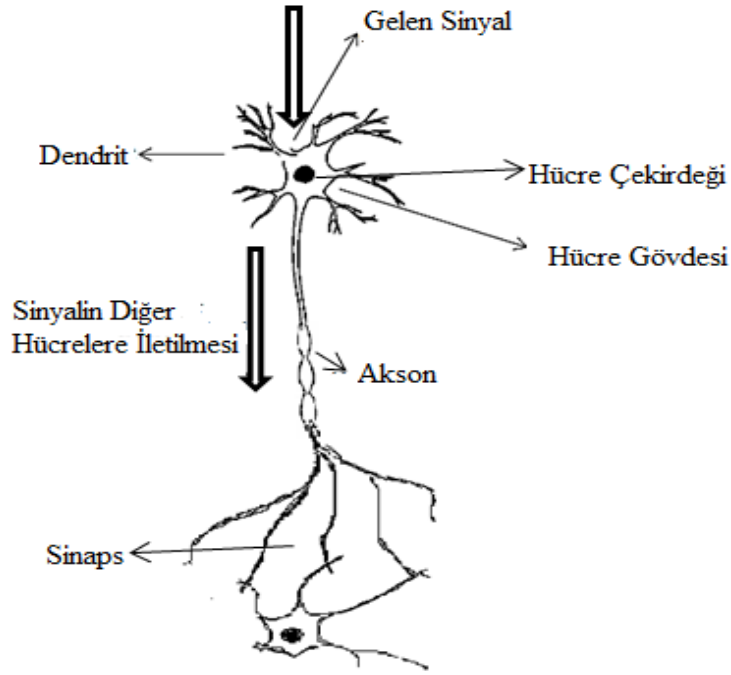
Vapnik(1963) çalışmasında çok katmanlı algılayıcılar olarak bilinen destek vektör makinelerinin temelini (DVM) 1963 yılında kendisi (Vladimir Vapnik) ve Alexey Chervonenkis tarafından ortaya atılmış olduğunu ileri sürmektedir. İstatistiksel öğrenmeyi esas alan, bir çeşit denetimli öğrenme algoritması olarak tanımlanan bu makineler 60’lı yılların ilk yapay zeka ürünü olarak kabul edilmektedir (DVM’ler 60’lı yıllarda çok fazla ilgi görmedi ise de, 1995 yılında Vladir Vapnik, Bernhard Boser ve Isabelle Guyon tarafından geliştirilmiştir).

İlk olarak 1965’ te Ivakhnenko ve Lapa tarafından geliştirilen ileri yayımlı perceptron da çok ilgi görmemiş, yine aynı yıl Nilssons “Öğrenen Makineler” adlı bir kitap yayımlamış ancak bilim kurgu sinemasında aranan ilgi karşılık bulmuş, akademik gelişimde ise beklenen ilgiyi uyandıramamıştır. 1969 yılında, ABD’de yapay zekâ araştırma çalışmalarını yürüten organizasyon olan DARPA yapay zeka ve makine öğrenmesi çalışmalarını desteklemeyeceğini açıklaması üzerine, yapay zekanın gelişimi bir süre daha beklenen ilgiyi canlandıramamıştır. Werbos tarafından 1974

yılında geliştirilen geriye yayılma tekniği ile çok katmanlı porseptron, 1986 yılında Rumelhart'ın çalışmaları ile ünlenene dek yapay zekâ ve makine öğrenmesi çalışmaları için duraklama devri devam etmiş, ancak Rumelhart'ın çalışmaları tekrar hız kazanan yapay zekâ teknolojileriyle bu günkü derin sinir ağlarının temeli atılmıştır (Huang ve ark., 2017:43).

Yapay sinir ağlarını barındıran zeka, insan beyni ve öğrenme işlevi dikkate alınarak kurgulanan bir çeşit geliştirilmiş makine öğrenmesi olarak tanımlanmaktadır. Bu tanımda, yapay sinir ağları, biyolojik insan beyni nöronlarını taklit ederek öğrenmeyi sağlamaktadır. Şekil 1.1.'de insan beynindeki biyolojik nöron yapısına yer verilmektedir.

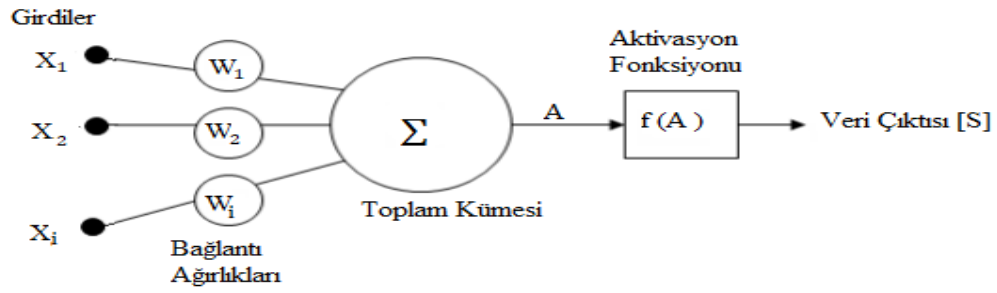
Biyolojik nöronda bulunan dendritlerin görevi, gelen uyarı sinyallerini sinir hücresinin gövdesine iletip, hücredeki kararlı halin bozulmasına sinir hücresinde bir dizi kimyasal sürecin başlamasına neden olmaktır. Sinir hücresinde vuku bulan bu kimyasal süreçte tüm girdiler bir sonraki sinir hücresine aksonlar aracılığı ile iletilir. Sinir hücresindeki akson ucu ile kendisinden sonra gelen dendrit arasındaki boşluk sinaptik boşluk olarak tanımlanır. Sinaptik boşluk içerisine bilgiyi ileten her dal iletken madde olarak kabul edilir ve bu bilgi iletimi sırasında dolarak bilginin geçişini sağlar. Sinaptik boşluktaki değişkenlik öğrenme süreci olarak tanımlanır.



Şekil 1.1. Biyolojik Beyin Nöronu Yapısı

Yapısı gereği matematiksel algoritmalarla ibaret olan, yapay sinir ağları ise temelde üç ana bölümden oluşmaktadır. Giriş katmanı sisteme gelen verilerden oluşur. Veri seti bilgileri bu katmana işlenir. Özetle bu katman girdi vektörüdür. Orta katman olarak görülen gizli katman ise nöron görevini üstlenen algılayıcıların bulunduğu YSA'nın özü olarak tanımlanabilmektedir. Biyolojik nörondaki tüm görevleri burada gizli katmanda yer alan yapay sinir ağlarını çalıştıran algoritmalar tarafından yönetilmektedir. Gizli katman nöronları bir önceki katmandan alınan her veriyi bir dizi işlemde geçirerek bir sonraki katmana ileterek, anlamlı çıkışlar sunmaya çalışmaktadır.

Giriş katmanındaki bilgiler bu katmanda işlenerek anlamlı çıktıların alınmasını sağlayarak giriş katmanındaki YSA nöronları $X_1, W_1; X_2, W_2; X_i, W_i$ gibi girdi verilerinin aktivasyon fonksiyonundan geçerek algoritmik kararı çıktı olarak sunar. $X_1, X_2, \dots, X_i$ girdi kümesidir. $W_1, W_2, \dots, W_i$ ise YSA nöronları ağırlık vektörü kümesinin elemanlarıdır. Algoritmik fonksiyon $X_i W_i$ girdilerinin çarpımlarının toplam kümesinden aktivasyon fonksiyonuna geçirilerek çıktı olarak sunulmaktadır. Bu çalışmanın mantıksal çizimine Şekil 1.2.'de yer verilmektedir.



Şekil 1.2. Çok Katmanlı Yapay Sinir Ağı Nöronu Çalışma Fonksiyonu (F. Rosenblatt'ın 1958 Yılında Tanıttığı Model) (Hornik ve ark., 1989:362)

$$S = f(A) = f(X_1.W_1 + X_2.W_2 + \dots + X_i.W_i) = f(\sum_{p=1}^i X_i W_i) \quad (1.1)$$

$$S = f(A) = f(X_1.W_1 + X_2.W_2 + \dots + X_i.W_i) = f(\sum_{p=1}^i X_i W_i + b) \quad (1.2)$$

Burada;

S = Verilerin Çıktısını,

A = Nöronun Aktivasyonunu,

$f(A)$ = Aktivasyon Fonksiyonunu,

$X_1, X_2, \dots, X_i$ = Sisteme Yüklenen Girdileri,

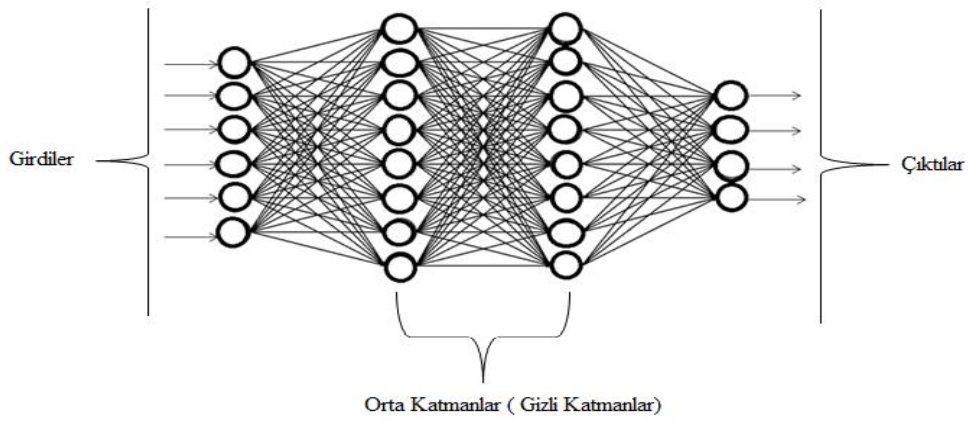
$W_1, W_2, \dots, W_i$ = Bağlantının Vektörel Ağırlıklarını

$X_p = (p = 1, \dots, i)$ Girdi Kümesi Olan X 'in P Sırasında Bulunan Elemanını,

$W_p = (P = 1, \dots, i)$ Ağırlık Vektörü W 'nin P Sırasındaki Elemanını,

b = Bias önyargı değerini, ifade etmektedir.

1989 yılında K. Hornik ve ark. tarafından yazılan Neural Network adlı eserin 2. Cilt 5. sayfasında ileri beslemeli yapay sinir ağını Şekil 1.3.'te gösterildiği gibi tanımlanmıştır. Şekil 1.3.'te beyin aksonlarına benzer şekilde görülen çizgiler ($\Leftarrow$) ağırlık vektörlerini ifade etmektedir. Ağırlık vektörünün ataması düzgün şekilde atanmamış ise bilgi diğer katmana geçmez yahut eksik geçer (Hornik ve ark., 1989:365).



Şekil 1.3. İleri Beslemeli Yapay Sinir Ağı Katmanları

Burada veri girişi ile giriş katmanında yer alan her yapay sinir ağı nöronu, giriş vektöründeki değerleri ayrı ayrı üstlenerek bir değer elde eder. Giriş vektörü değerleri; RGB bandındaki JPEG formatında bir resme ait piksel değeri olabileceği gibi, ses aralığındaki titreşim değeri veya sayısallaştırılmış herhangi bir değeri ifade etmektedir. Giriş katmanı nöronları, kendilerine atanmış ağırlık değeri vektörleri ile bağlı gizli katman nöronlarına bilgiyi taşır. Gizli katman nöronları $S = f(A)$ başlangıcı ile (1.2) de bağlantısı verilen fonksiyonu tekrar tekrar işler ve her işleymde bias (b) adı verilen bir önyargı değeri ekleyerek ilerleme sağlamaktadır. Burada elde edilen her değer, kendisinden sonra gelen diğer katmana taşınarak ilerleme ve öğrenme sağlanmaktadır. Diğer gizli katmanlarda işlenen her değer bir sonraki katmana taşınmakta, değerlerin sürekli olarak sonraki katmana aktarılması şeklinde ilerlemesinden dolayı ileri beslemeli yapay sinir ağı adını almaktadır. Sistemdeki her gizli katman aynı fonksiyonu işleyerek sonucunu sonraki katmana iletilmektedir. Son katman çıktı katmanıdır ki burada girdi katmanında yer alan veri sayısı ile farklılık göstermesi mümkün görülmektedir. Bu giriş verilerinde değer kaybı yahut sayısallaştırmada sorun olması sebebi ile sınıflandırılmamış olması yahut aynı değer birden fazla kez yüklenmiş olması sebebi ile birlikte öznelite ataması yapıldığı algısını çağrıştırmaktadır.

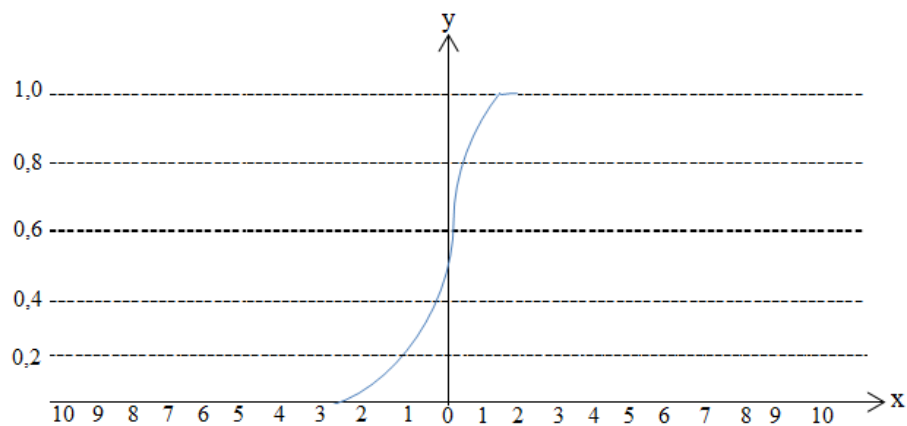
K.Hornik ve ark.; a.g.e.'de, evrensel yaklaşımlarla, standart çok katmanlı ileri beslemeli sinir ağlarının verileri, yakınlık derecesi niteliği ile sınıflandırılabilirliğini,

yahut derecesi fark etmeksizin bir fonksiyonun sonucunu kabul edilebilir bir sonuç ile tahmin edebileceğini ileri sürmektedir. Başarısız bir tahmin sonucunun oluşması hâlinde bu sorunun; girdi verilerinin eksik olmasında, gizli katmanların yetersiz olması ile öğrenme eksikliğinin olmasında yahut girdi verileri ile hedeflenen çıktı verileri arasında yeterli derecede ilişki kurulamamasına bağlı olduğunu vurgulamaktadır (Hornik ve ark, 1989:368).

Yapay sinir ağlarına (YSA) yönelik bir tarih yazılacak olursa 1980 öncesi ve sonrası olarak ikiye ayırma konusunun hiçbir yazar tarafından göz ardı edilemeyecek kadar büyük bir ayrım barındırmaktadır. YSA kullanımının, yapay zeka tarihçesinde kısaca bahsetmiş olduğumuz gibi doğrusal olmayan Sigmoid fonksiyonunu kullanan çok katmanlı algılayıcı modelin tanınması ile birlikte hız kazandığı görülmektedir. Sigmoid fonksiyonu, temelde bulanık mantık kümesi olan $[0,1]$ aralığındaki değerlerden oluşan kümeyi kullanır. Yapay sinir ağları oluşturulurken sıklıkla başvuru alan aktivasyon fonksiyonu olarak bilinir. Doğrusal ve doğrusal olmayan davranışlar arasında denge sağlayan ve sürekli artan bir fonksiyon olarak tanımlanmaktadır (Chih-Min ve ark.,2014:373).

$$y = f(x) = \sigma(x) = \frac{1}{1+e^{-x}}, 0 \leq f(x) \leq 1 \quad (1.3)$$

Sigmoid fonksiyonuna ilişkin grafik ise Şekil 1.4.'te yer aldığı gibidir:



Şekil 1.4. Sigmoid Fonksiyonu Grafiği

Sigmoid Fonksiyonu, nöral kompresyonda genellikle sadece ağırlıklandırılmış toplamaların katmanlı ağırları ve ardından harici programlamanın kısmen öğrenme ile sübvansede edildiği bir karar fonksiyonu olarak, 1980'lerde bağlantıcılığın güçlü bir canlanmasını sağlayan fonksiyon olarak kullanılması ile karşımıza çıkar. Bu vizyon, sonunda takip problemlerini ortaya koymaktadır (Srivastava ve Lekha, 2010:163).

Sigmoid fonksiyonunu kullanan aktivasyon kodlarına ilişkin problemleri şu şekilde özetleyebiliriz (Mira,1995:13).

- Gerçek nöronların bildiklerini açıklamak için, açıkça yetersiz olan hesaplama modüllerinin kullanımı, yapay nöronların biyolojik gerçeklikten geri dönüşü olmayan bir açıyla uzaklaşmasına yol açmaktadır.
- Sentez süreçlerinde metodoloji eksikliği ve aşırı ampirizm vardır.
- Kendini, yapay zekanın sembolik bakış açısıyla bütünleştirilemeyen bir alternatif olarak sunar.

Yapay Sinir Ağlarına ilişkin problem her ne kadar çözülmüşse de 2000'li yılların başlarına kadar hayatın içinde ve hemen her kademesinde varolduğunu söylememiz de gerçeği tam olarak yansıtmayacaktır (Ruder,2016).

1980 yılında çalışmaları desteklenmeyen ve sıkıntılı sürece karşın çalışmalarına devam eden çok az bilim adamı çalışmalarının karşılığını almaya başlamışlardır (Lantz, 2015).

Yukarıda yer alan problemlere ilişkin yapay zekâ ve YSA'lar için farklı aktivasyon kodları kullanılmaya çalışılmıştır. En bilinen aktivasyon kodları: Hiperbolik Tanjant Fonksiyonu, ReLU, Softmax fonksiyonunu kullanan aktivasyon kodlarıdır. Bu aktivasyon kodlarına ilişkin bilgilere derin öğrenme konusunda değinilecektir. Rumelhard V.d. tarafından 1986 ve 1988 tarihlerinde yayımlanan, "Parallel distributed processing" ve "Explorations in the microstructure of cognition" adlı eserleri ile bugünkü yapay sinir ağlarının temelleri atılmıştır (Rumelhart ve ark., 1986:43).

1988 yılında Broomhead ve Lowe tarafından “Radyal Tabanlı Fonksiyonlar” tanıtılmıştır. 1997 yılında Hochreiter ve Schmidhuber tarafından Tekrarlayan Sinir Ağları (TSA) çalışması tanımlanmış ve tanıtılmıştır (Pratt, 1994:12).

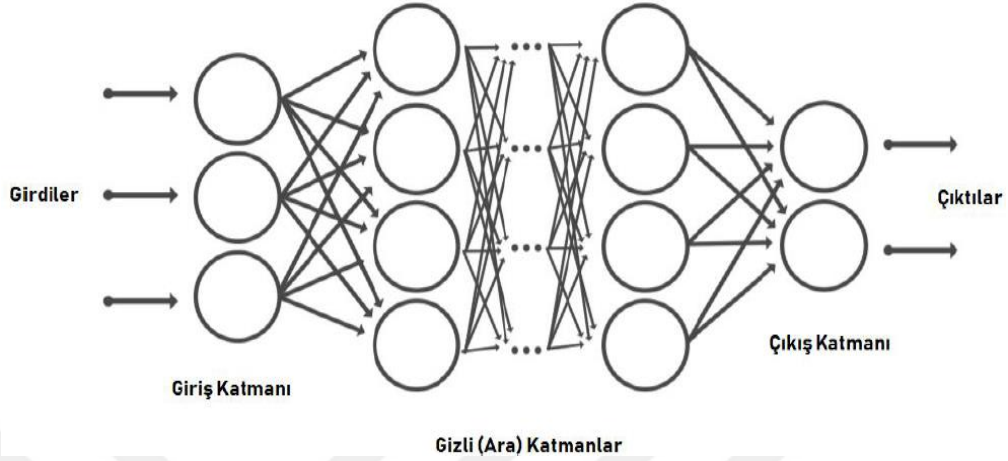
Ancak 2000 yılına dek YSA algoritmaları makine öğrenmesine göre çok daha fazla avantaj sunmasına rağmen maliyeti fazla görüldüğünden, model eğitim süresinin uzun olması sebepleri ile tercih edilmeyerek bunun yerine probleme özel tasarlanan destek vektör makineleri tercih edilmiştir. Yapay sinir ağlarının tercih edilebilirliği bilgisayar hız kapasitelerinin artması, grafik işlemcileri (GPU)’un hesaplamalarda kullanılabilir olması ile modellerin daha hızlı eğitilebilmesine imkân tanınması ile olmuştur (Bramer,2016:11),

2000 yılında YSA için “Derin Öğrenme” metodu Igor Aizenberg tarafından tanıtılmış ve bu doğru öğrenme metodu ilgi uyandırmıştır (Bramer, 2016:18).

Günümüzde yapay sinir ağlarının en gelişmiş öğrenme biçimi derin öğrenme olarak tanımlanmaktadır. Derin öğrenme Derin Sinir Ağları ile gerçekleştirilmektedir. İlk kez 2006 yılında Hinton vd. tarafından çok katmanlı yapay sinir ağlarının daha verimli eğitebileceğinin öne sürülmesiyle ortaya çıkmış olsa da derin öğrenme terimi ilk olarak 1987 yılında Rina Dechter tarafında kullanılmış ve 2000 yılında Igor Aizenberg ve arkadaşları yaptıkları çalışmalarda bir yapay sinir ağı modeli olarak derin öğrenme kavramını ortaya koymuştur.

Buradaki derin kelimesi ağıdaki katman sayısını ifade eder. Katman sayısı artıkça ağ yapısı derinleşir. Yapay sinir ağları genelde bir iki katmandan oluşurken, derin ağlar yüzlerce katmanlardan oluşabilmektedir. Ancak Derin öğrenme 1990 ile 2000’ler arası veri yetersizliği, ana darboğaz ve donanım yetersizliğinden dolayı parlak bir geçmişe sahip olarak görülmemiştir. Çünkü sinir ağ modeli derinleştikçe işlem yükü artar ve donanımsal yetersizlikler ortaya çıkmaktadır. Ayrıca ağı geriye doğru eğitilmesi sırasında katman sayısı artıkça geri bildirim sinyali yok olup gidebilmektedir. Veri yetersizliği de derin öğrenmenin en büyük problemlerinden olan aşırı uydurmaya

sebebi olmaktadır. Şekil 1.5.'te bir derin sinir ağı yapısına yer verilmektedir (Mira,1995).



Şekil 1.5. Derin sinir ağlarının genel yapısı (Mira,1995:11)

Yapay sinir ağlarının derin öğrenme ile eğitimleri, danışmansız öğrenme, yarı danışmanlı öğrenme ve danışmanlı öğrenme olarak gruplandırılmaktadır. Danışmansız öğrenme; veri setinde etiketlemesi bulunmayan dolayısı ile girdilere yönelik sistem çıktısının alınmasının görece daha zor olduğu öğrenme biçimidir. Kümeleme işlemi bu tarz öğrenme biçimine örnektir (Jaderberg ve diğerleri, 2015).

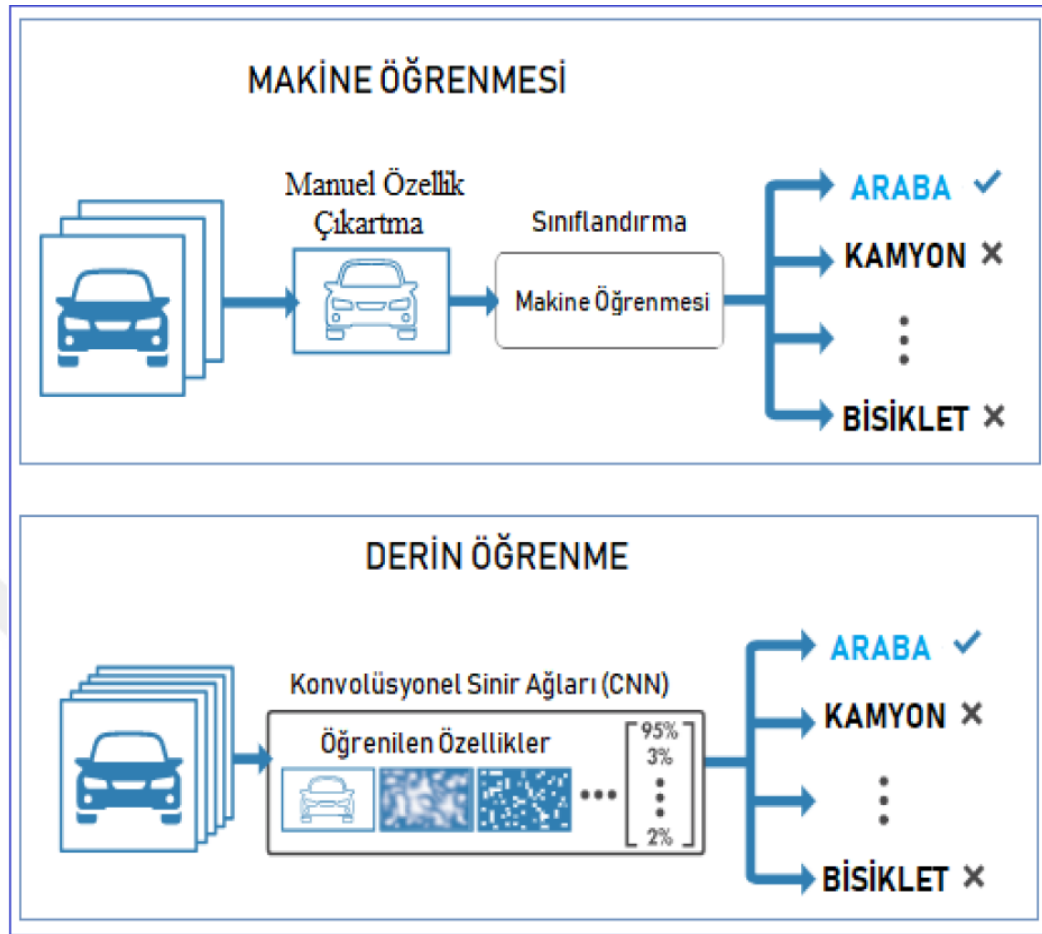
- İnternetin yaygınlaşması ile makine öğrenmesinin ihtiyaç duyduğu büyük veri setlerinin toplanması ve dağıtımının mümkün kılınması.
- 2009-2010 civarında basit ama önemli birkaç algoritmanın keşfi ile hata sinyalinin geriye daha iyi iletilmesi mümkün hale gelmiştir. Bu algoritmalar:
- Katmanlar için daha uygun aktivasyon fonksiyonları (Relu, Softmax gibi),
- Daha uygun ilk ağırlık başlatma yöntemleri,
- Daha iyi eniyileme (optimizasyon) yöntemleri (RMSProp ve Adam gibi).
- 2014,2015, 2016 yıllarında yığın normalizasyonu (batch normalization), artık bağlantılar (residual connections) ve derinlemesine evrişim (depthwise separable convolution) gibi hata sinyalin geriye daha iyi yayılımını sağlayan bu yöntemler ile günümüzde, yüzlerce katmandan oluşan ağları sıfırdan eğitebilir duruma geldi.

- Bilgisayar oyunları pazarı için geliştirilen ekran kartları ile daha verimli paralel hesaplamaları gerçekleştirebilmesidir.

Sonuç olarak derin öğrenme bugün sağlık, eğitim, iletişim gibi sektörlerde doğal dil işleme, örüntü tanıma, konuşma tanıma gibi birçok alanlarda yaygın bir biçimde kullanılmaktadır. Bu çalışmada da derin öğrenme verilerin bir kısmını görerek gelecek tahminlemede bir rota oluşturan Matlab programında kullanılmaktadır. WEKA için ise daha büyük miktarda girdilere ihtiyaç duyulmaktadır.

Derin öğrenme, tüm katmanlarının gösterimlerini sıralı olarak öğrenmek yerine bütünlük olarak öğrenir. Her şey tek bir geri bildirim sinyali tarafından kontrol edilir ve modeldeki her değişim hedefe ulaşmaya hizmet eder. Böylece sıg modelleri üst üste sıralama yerine gösterimleri ara katmanlardan birbirinden soyut olarak öğrenebilmek mümkün olmaktadır.

Derin öğrenmeyi farklı kılan en önemli özelliklerinden biri; Klasik makine öğrenme tekniklerinde veriyi kullanmadan önce ön işlem den geçirerek veriyi model için anlamlı kılmak gerekmektedir. Bunun için problemin çözümünü kolaylaştırmak amacıyla girdilerin daha ayırt edici olmasını sağlayan parametrelerin atanması sağlanmaktadır. Derin öğrenme ile yapılan kurgulamalar veya benzetim yollu simülasyonlar, makine öğrenmesinin bu kritik adımını tamamen otomatikleştirmiş olarak kendi içinde yapması nedeniyle problemlerin çözümünü kolaylaştırdığından tercih edilmektedir. Şekil 1.6.'da girdilerin tahmin edilmesinde hem klasik makine öğrenmesi hem de derin öğrenme modellerinin nasıl bir yol izlediği gösterilmiştir.



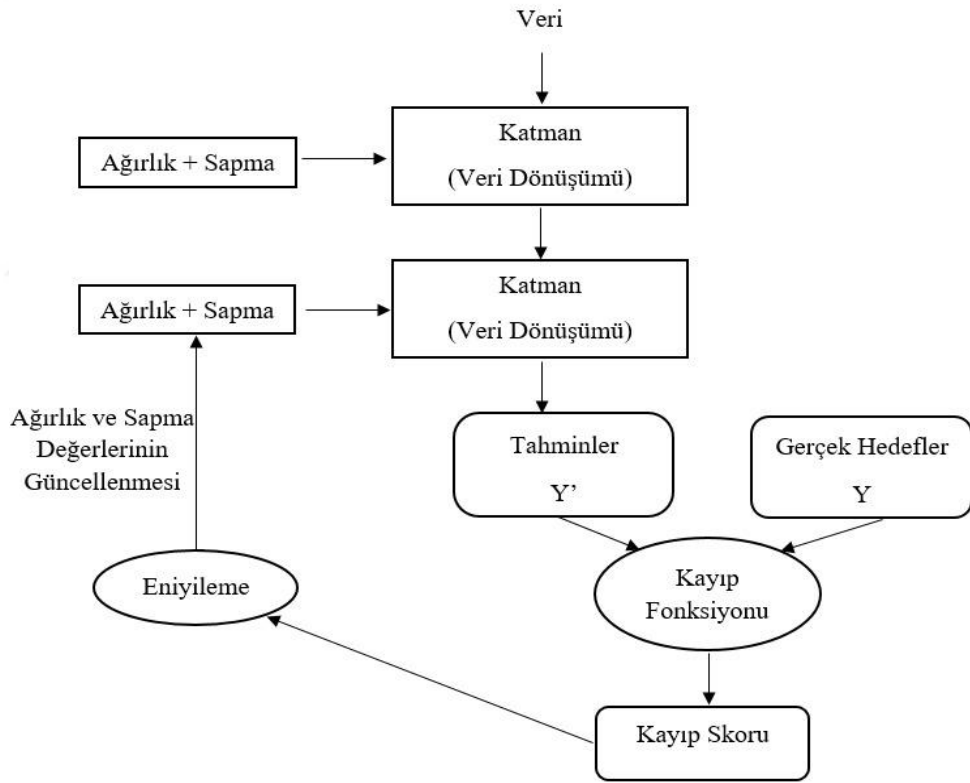
Şekil 1.6. Derin Öğrenme ve Geleneksel Makine Öğrenmesi Yaklaşımının Karşılaştırılması

Geleneksel makine öğrenmesi tekniklerinde eğitim verilerinin artırılması her zaman başarıyı olumlu etkilemezken; derin öğrenmede eğitim verisi çeşitlenip artıkça başarı oranı da arttığını söylemek mümkün olmaktadır. Ancak aynı veri setinin tekrarlı halini derin öğrenme ile eğitilmiş sinir ağı algılamaktadır ve yinelenen verileri eğitimde tekrar dikkate almamakta, verilerin farklılaştığı alanları dikkate almaktadır.

Derin sinir ağlarıyla uygulama yapılırken genelde 3 farklı yöntem uygulanır. Birinci olarak problemin çözümüne uygun derin ağ modeli oluşturularak eğitimin gerçekleştirilmesi, ikinci olarak önceden eğitilmiş hazır bir ağ yapısını yeni verilerle eğitilmesi ve son olarak da derin sinir ağı yapısından öznelilik çıkarımı yapılarak diğer makine öğrenme yöntemleri ile kullanılmasıdır. Gerçekleştirilen bu çalışmada eğitilmiş bir ağ yapısı olan programlar kullanılarak verilerin girişi sağlanmıştır. WEKA için daha iyi sonuç verdiği için veri çoğaltma yöntemi kullanılmış olup

analizler çoğaltılan veri girişi ile sağlanırken Matlab için veri girişlerinin tekrarını görmemesi sebebiyle çoğaltma işleminin yapılamadığını vurgulamak yerinde olacaktır ki yapısı gereği Matlab'ta çoğaltılmış veriler yukarıda açıklanan öğrenme biçimi nedeniyle gerçekleştirilmemiştir.

Derin ağ yapısı, zincirleme eklenen katmanlardan oluşmaktadır ve girdi verisini tahminlere eşleyerek çalışmaktadır. Sonrasında kayıp fonksiyonu vasıtasıyla tahminler ile hedefler arasındaki fark olan kayıp değeri bulunarak ağırlık tahminleri ile hedeflerin ne kadar eşleştiği ölçülür. Eniyileme algoritması kayıp değerini kullanarak ağırlık ve sapma değerleri güncellenmektedir. Katmanlar, kayıp fonksiyonu ve eniyileme algoritması arasındaki ilişki Şekil 1.7.'deki gibidir.



Şekil 1.7. Katmanlar, kayıp fonksiyonu ve eniyileme algoritması arasındaki ilişki

Buradaki “Ağırlık” eğitilen parametrelerden bir tanesi İngilizcede weight anlamına geldiği için genellikle W harfi ile gösterilir. Sinir ağı tarafından eğitim esnasından daha önemli olduğu düşünülen girdilere daha yüksek ağırlık değeri verilmektedir.

“Sapma” ise, eğitilen parametrelerden bir tanesi İngilizcede bias anlamına geldiği için genellikle b harfi ile gösterilir. Ağırlıktan farklı ek olarak uygulanan doğrusal bileşene sapma denilmektedir. Bu değer; ağırlıklı katsayılı girişin aralığını temel olarak değiştirmek için eklenmektedir.

“Aktivasyon fonksiyonu”, tensörlere etki eden doğrusal olmayan operasyonlar olarak tanımlanmaktadır. Amaç ağırlık ve sapma değerlerin ayarlanmasıdır. Bunu neronların tetiklenmesi olarak düşünebiliriz. Yüksek bir değer geldiği zaman neron daha yüksek bir sinyal verecektir.

Yapay zekâ ve YSA’da kullanılması ile neredeyse devrim yaratan Sigmoid Fonksiyonu hakkında bilgiler çalışma kapsamında ilgili bilgiler verildiğinden diğer en sık kullanılan fonksiyonlara ilişkin bilgilere değinilecektir. Bunlar sırası ile; Hiperbolik Tanjant Fonksiyonu, ReLU Fonksiyonu ve Softmax Fonksiyonlarıdır.

- Hiperbolik Tanjant Fonksiyonu (Tanh); Sigmoid Fonksiyonunun $[-1$ ve $1]$ aralığını kapsayan fonksiyondur. Eşikleme $0,5$ te değil 0 da sağlanır. Bu sebeple 0 dan küçük olan sayılar 0 (sıfır) olarak kabul edilmeyerek -1 ‘e yaklaşan değerler alır ve fonksiyon denklemi şu şekilde formüle edilir:

$$\text{Tanh}; -1 \leq f(x) \leq 1 \quad (1.4)$$

- ReLU Fonksiyonu; Düzeltilebilir doğrusal birim aktivasyon fonksiyonudur. Fonksiyon aynı Sigmoid fonksiyonunda yer alan sorunu yansıtır. Bu sorun 0 ’dan küçük değerleri 0 değeri gibi kabul etmektedir. Pozitif değerler x değerini alırken negatif değerler 0 olarak kabul edilir ve şu şekilde formülize edilir:

$$y = f(x) = \max(0, x), \quad 0 \leq f(x) \quad (1.5)$$

- ReLU fonksiyonu, Sigmoid ve Hiperbolik Tanjant fonksiyonlarını kullanan aktivasyon kodlarına nazaran daha YSA eğitimi sırasında daha hızlı

çalıştığının tecrübe edilmiş olması nedeniyle, ağın öğrenmesi sürecinde sıkça kullanılan bir fonksiyon olmasına rağmen tıpkı Sigmoid Fonksiyonu aktivasyon kodunda olduğu gibi dezavantajlara sahiptir. Bu dezavantaj negatif değerlerin 0 olarak kabul edilmesidir. Bu durum 0'dan küçük ağırlık değerlerini alan perseptronların diğer değişle nöronların yok olmasını ve bir daha eğitim sürecinde kullanılamaması sonucunu beraberinde getirmektedir. Bu dezavantajın önüne geçilmek için sızdırmalı (leaky) ReLU fonksiyonu ileri sürülmüştür. Bu fonksiyonun formülasyonu şu şekildedir:

$$y = f(x) = 1(x < 0)(\alpha x) + 1(x \geq 0)(x), \quad (1.6)$$

Fonksiyondaki α sabiti genellikle küçük bir sayı olarak eklenir ve 0'a yakın bir değer elde edilir (0,1) gibi ve bu değer negatif ağırlık değeri ile çarpılmaktadır.

- Softmax Fonksiyonu; Çoklu sınıflandırma işlemleri için kullanımına en çok başvurulmuş fonksiyon olmakla birlikte, girdiler için bir sınıflandırma olasılığına bağlı kayıp değerleri üreten fonksiyondur. Sınıflandırıcı, girdilerin gerçek sınıfa ait dağılımı ile ait olma olasılığı bulunan kümenin dağılımı (en çok olabilirlik fonksiyonunu) normalize eder ve iki değer arasındaki çarpaz entropi değerinin en aza indirgenmesi mantığı ile hareket ederek çıktıları sunar. Formülü ise şu şekildedir:

$$\sigma: \mathbb{R}^n \rightarrow \text{int}(\Delta^{n-1}), \quad (1.7)$$

$$, \lambda > 0 \quad (1.8)$$

softmax işlevi $\sigma: \mathbb{R}^n \rightarrow$ ile verilir

$\Delta = 1$ olduğunda, (1) 'e standart softmax işlevinde ise $\Delta \rightarrow 0$ olarak, σ çıkışı noktası merkezinde akıllıca, yani tekdüze bir olasılık dağılımı uygular. Öte yandan, $\lambda \rightarrow \infty$ olarak, σ çıkışı noktasal olarak e, $j \in \mathbb{R}^n$ 'ye yakınsar, burada $j = \arg\max_{1 \leq i \leq n} e_i$, z,

değerleri ile ifade edilir. İki veya daha fazla bileşen arasındaki fark z'nin çok küçük olmaması gerekmektedir.

Bu fonksiyonlara ek olarak Derin Öğrenmede kullanılan farklı aktivasyon kodları da bulunmaktadır. Derin öğrenmede en sık başvurulan aktivasyon fonksiyonlarına yer verilmiştir ki WEKA ve Matlab için kullanılan kodlarda en iyileme ile benzetim için de bu fonksiyonların kullanımı muhtemeldir.

Öğrenme katsayısı ve eniyileme algoritmasını ise şu şekilde açıklayabiliriz: Öğrenme katsayısı, başkaca bir söylem ile hata düzeltme katsayısıdır. YSA'da akson görevi gören taşıyıcı sinaptikleri eğim düşümü algoritmalarında kullanılan bir katsayıdır. Öğrenme katsayısı, eğim düşümüne göre tanımlanmış algoritmalarının yaklaşmasını sağlamaktadır. Öğrenme katsayısı büyüdükçe sinaptiklerin bir sonraki katmana bilgiyi taşıyamamasına neden olur. Bu katsayı için çok küçük bir değer atandığında ise öğrenme sürecinin uzamasına etki etmektedir.

Öğrenme katsayıları için çeşitli eniyileme algoritmalar kullanılır, amaç Eniyileme algoritması, öğrenmenin nasıl gerçekleşeceğini belirlemektir. En sık kullanılan algoritmalar ise, eğimine göre, Adagrad, Adamax, Nadam, Adadelata, RMSprop ve Momentum algoritmalarıdır.

Kayıp fonksiyonu da burada tahminleme için bir ağı doğruluk değerini hesaplamak üzere kullanılmaktadır. Bu fonksiyon, ağı, gerçek değerlerinin tahmininde hata oluşursa cezalandırmaya çalışmaktadır.

Geri yayılım ise tek bir iterasyon ile elde edilen çıktı için ağı hatasının hesaplanabilmesi için kullanılan bir kavramdır ve elde edilen hata değeri, maliyet fonksiyonun eğimi ile birlikte ağı ağırlıklarının, güncellenebilmesi için ağa geri beslenmesi olarak tanımlanmaktadır. Sonraki iterasyonlar da hatanın minimize edilmesi için bu ağırlıkların güncellenmesini gerekli kılmaktadır. Kayıp fonksiyonun eğimi kullanılarak ağırlıkların güncellenmesi geri yayılım olarak ifade edilmektedir.

Yığın, epok ve tekrar sayısı, tanımlamaları ise şu şekilde yapılabilmektedir. Yığın, model tarafından işlenmesi için bir defada alınan miktarı ifade etmektedir. Tüm girdinin tek seferde gönderilmesi yerine girdiler rastgele seçilmiş eşit büyüklükte birbirinden bağımsız yığınlara bölünerek ağda eğitilmektedir. Her bir yığın kümesi işlendiğinde bir tekrar, tüm girdi yığınlarının bir defada işlenmesine ise epok denilmektedir.

Derin öğrenmenin birden fazla mimarisi bulunmaktadır ancak en çok kullanılanlar;

- Tekrarlayan Sinir Ağı -Recurrent Neural Networks (RNN): Ara katman çıkışını hem bir sonraki katmana hem de tekrar aynı katmana giriş olarak vermesinden dolayı tekrarlayan sinir ağları denir. Sınıflandırma problemlerinde kullanılabilen bu yapının ses tanıma çalışmalarında da başarılı olduğu görülmüştür.
- Kısıtlanmış Boltzmann Makineleri (Restricted Boltzmann Machines) (RBM): Sınıflandırma, bağlanım ve özellik öğrenimi işlemlerinde kullanılan boltzman makinesi giriş veri seti üzerinde olasılık dağılımlarını öğrenebilen bir sinir ağıdır.
- Uzun Kısa Dönemli Bellek (Long / Short Term Memory) (LSTM): RNN'lerin dezavantajı olarak bilinen bağlam boşluklarının tahmin edilmesi ihtiyacından dolayı Hochreiter ve Schmidhuber tarafından 1997 yılında LSTM'ler ortaya çıkmıştır.

Gizli durumları hesaplamak için LSTM ağlarında bir yapı kullanılmaktadır. LSTM içerisinde hafıza hücreleri yer almaktadır. Bu hafıza hücreleri önceki durum ile girdi bilgisini tutan hücreler olarak bilinmektedir. Ağ mimarisi içerisinde yer alan bu hücreler hangi verinin tutulacağına ya da hangi verinin silceğine karar vermekte etkin bir rol oynamaktadırlar. Sonraki aşamada ise önceki durumu mevcut bellek ile giriş verisini birleştirilmektedirler. Böyle bir yaklaşımla uzun vadeli bağımlılıkların ortadan kaldırılarak veri dizilerinin devam ettirilmesi mümkün kılınmaktadır. Böylelikle dışarıdan işlenen verinin simüle edilmesi için modelin eğitilmesi daha kısa

sürebilmektedir. Bilinen benzetim programları için tekrar modelleme çalışmalarında da bu işleyiş biçimi dikkate alınmaktadır.

Yukarıda YSA ve Derin Öğrenme kavramlarıyla sunulan arka planda yer alan bilgilere dayalı olarak iki adet (WEKA ve MATLAB) simülasyon programlarının çalışma prensibi gereği sadece tahminleme raporlaması için oluşturulmadığını, Yapay sinir ağları kullanan programlar olmaları sebebiyle modellemeler için kullanılacak programlar olduğunu vurgulamak üzere sunulmaktadır. Yani bir tahminleme modelini oluşturmak üzere uygun parametrelerin seçilmesiyle ve aktivasyon kodlarının üretilmesiyle problem çözümlerinde kullanılabileceği zaten bilinen benzetim programlarının, satış tahmini raporlamalarındaki başarımlarını ölçerek, çözümlerinin yapılması tamamen bu sezgisel planı işletmek, çalışma bütününe yayarak soruları çözmek ve üretim de dahil olmak üzere talep tahminlemesine göre planlamaların yapılması için bir model sunmak üzere çalışmada yer almaktadırlar.

1.3. Çalışmanın Önemi

Bu çalışma kapsamında sunulan geniş arka plana bağlı olarak çalışma kapsamında araçların teknik parçası olan amortisörlerin talep tahminleri için gerçek verilerin kullanılmasıyla gelecek talep tahminlerini doğru sunmak üzere iki YSA tabanlı program kullanılarak tahmin modellemesini sunmak ve başarımlar durumlarını kıyaslamak, gelecek çalışmalar için de bir tahminleme rotası oluşturması açısından önemli görülmektedir. Bununla birlikte, üretim planlama bölümleri bağlamında burada seçilen veriler amortisörler için talep tahminleri özelinde olsa da diğer ürün ve hizmetler için de kullanılacak rehber bir çalışma olması açısından da önemli görülmektedir.

Üretim planlama çalışmaları özelinde, bu çalışma, literatüre katkı sunması açısından ayrıca önemli görülmektedir. Ayrıca pandemi dönemine ait bir tahmin çalışması olması sebebiyle bu sürecin üretim sektörüne olan etkisine bir perspektif kazandırmış ve çalışma özelindeki şirketin maddi anlamda yaşayacağı tahmini kaybı yansıtmıştır. Simülasyon programları özelinde ise, erişilecek bulgularla, çalışma kapsamında bir

mukayese sunması da piyasadaki firmalar için fikir elde etmede yol gösterici niteliğe sahip bir çalışma niteliği barındıracaktır. Bu yönüyle bir mukayese çalışması olması açısından da önemli bir çalışmadır.

1.4. Çalışmanın Varsayımları

Bu çalışma kapsamında gerçek bir amortisör firmasının satış raporları kullanılmış olup WEKA için veri setinin işlenmesi kısmında açıklamasının yer aldığı üzere, veri seti 2 kez büyütülerek, 1 aylık verinin 3 aylık olarak işlendiği varsayılmıştır. MATLAB için ise verinin tekrarlanarak büyütülmesi uygulamanın eğitim yöntemine olumsuz etki yarattığından tekrarlayan girişlerin varlığı mümkün olmamıştır.

1.5. Çalışmanın Kapsam ve Sınırlılıkları

Bu çalışma kapsamında, “amortisörler” özelinde ürün sınırlaması yapılmıştır. Amortisörler için talep tahminlemesinde ise gerçek satış rakamları ile elde edilen veriler, bir firma verisi ile sınırlı tutulmuş olup, çalışma boyunca sadece bu firmanın verileri kullanılmıştır. Veri setinin oluşturulması kısmında anlatılacağı üzere, aylık satış raporlarında yer alan adetlerle çalışma sınırlandırılmıştır.

Ayrıca amortisör yahut genel anlamda araç satışlarını etkilediği düşünülen bağımsız değişkenler problemde 14 genel başlıkla sınırlandırılmıştır. Bunlardan bazıları literatürdeki çalışmalarda değinilmiş, bazıları ise nitel bir yaklaşımla problemi etkileyeceği düşünüldüğünden probleme eklenmiştir.

BÖLÜM 2. KAYNAK ARAŞTIRMASI

2.1. Amortisörler Hakkında Yapılan Araştırmalar

Amortisörler hakkında yapılan araştırmalarda, amortisörlerin yeniden modellenmesini ve analizini içeren çalışmalar; Ninness(2000) tarafından yapılan analiz çalışması ve simülasyonun sağlandığı MATLAB çalışması süspansiyonun diferansiyel denklemlerle nasıl sembolize edilebileceğine yer vermiştir. Ninness yaptığı çalışmada, aracın tüm fonksiyonlarını varsayımsal olarak iki araç için tanımlamış tekerlek ve şase bağlantılarını sunduğu denklem tanımlarını MATLAB verisini çalışmasında sunmuş ve son olarak simülasyona yer vermiştir. Bu çalışmadaki amaç simülasyon verisinin gerçek değerlerle uyumundan çok, MATLAB için denklem oluşturma amacı doğrultusunda şekillendirmiştir. Çalışmanın sonunda MATLAB denklemini sunmuştur.

Kryze (2002) ise amortisörleri debriyaj mekanizmasının statik tepkisinin modellenmesini sağlamıştır. Amortisörler hakkındaki dinamik tepkinin modellenmesini araç sisteminin vibrasyon etkisinin azaltılmasını motor volanı ile dişli kutusu arasına bir titreşim damperi yerleştirerek çözmektedir. Araştırmanın sonucunda, modelleme ANYS olarak bilinen sonlu elemanlar metodunu kullanarak sağlamıştır.

2.2. Yapay Zeka ile Talep Tahminleme Araştırmaları (2016-2021 Arası)

Lalou ve ark., (2020). Perakende satış ağının talebini tahmin etmenin, fiziksel mağaza siparişlerinin entegrasyonunun verimli bir şekilde depolanmasının, analiz edilmesinin, veri oluşturmanın oldukça zorlu bir iş olarak tanımlanmaktadır. Araştırmanın temel amacı, veri analitiğinin ağ depolarını yenileme konusunda 3PL karar verme sürecini

nasıl destekleyebileceğini göstermek, böylece hem Dağıtım Merkezi (DC) hem de perakende satış noktalarında envanter yönetimini ve insan iş yükü planlamasını iyileştirmektir. Bunu yapmak için bu makalede, beş farklı ülkede 129 mağazadan oluşan bir ağ işleten büyük bir spor malzemeleri ithalatçısı adına fiziksel mağaza ve çevrimiçi siparişleri yerine getiren bir Yunan 3PL sağlayıcısının verileri işlenerek, talep tahmin yöntemlerini karşılaştırmak ve en küçük tahmin hatasını üreteni belirlemek için, çok sayıda kitaplıkla iyi donatılmış bir istatistiksel programlama dili olan döngü durumundan faydalanmaktadır. Sonuç olarak talep tahminleri ile gerçek arz rakamlarını karşılaştırmaktadır.

Lakshmanan ve ark., (2020). Bu makale, pazardaki satış tahminine model sunar ve gelecekteki talebi tahmin etmek için farklı makine öğrenimi modellerini karşılaştırmaktadır. Derin öğrenme mimarisindeki son gelişmelerle, büyük hacimli piyasa verilerini işleyerek ürünlerin geçmiş satış verilerini girdi olarak alan ve her bir ürünün sonraki üç zaman serisi için talebini tahmin eden uzun kısa süreli bellek (LSTM) ağı önerilmektedir. Ham satış verileri önce çeşitli teknikler kullanılarak ön işleme tabi tutulur ve ardından işlenen girdi modele beslenerek, satış verileri, gerçek zamanlı satış verileri ile karşılaştırılmaktadır.

Dellino ve ark., (2018). Tedarik zinciri yönetimi genel bağlamında, taze ve çabuk bozulan ürünler için satış tahmini sorununu ele almışlardır. Tahmin faaliyeti, belirli bir mağazadaki tek bir öğeye atıfta bulunur ve veri analizi ve normalleştirme için bir ön işleme aşamasından başlamaktadır. Daha sonra veriler, kullanıcı etkileşimli olacak şekilde tasarlanmış bir tahmin algoritması için girdi olarak kullanılmaktadır. Çalışmada üç tahmin yöntemi uygulanmıştır: ARIMA, ARIMAX ve transfer fonksiyonu modelleri ile tahmin modellerinin dışsal bileşenleri fiyatların etkisini dikkate alarak modellerin en iyi konfigürasyonu bulmaya odaklanmıştır. Çalışmanın sonucunda 19 küçük ve orta ölçekli mağaza ve 156 taze ürüne dayanan bir dizi gerçek veri, hem tahmin sonuçlarının kalitesini hem de bunların sipariş planlama faaliyeti üzerindeki etkilerini değerlendirmek için kullanılmış; satış tahmini, beklenen talebin bir temsilcisi olarak sunulmaktadır.

Khan ve ark., (2020). Modern çağda iş zekası (BI), ile çalışmışlardır. Talep tahmini için önce piyasadan ham satış verileri toplanarak bu verilere göre gelecekteki satış-ürün talepleri tahmin edilmek üzere kullanılmıştır. Talep tahmini, makine öğrenimi ile, haftalık, aylık ve üç aylık taleplerini belirlemek üzere kullanılmıştır. Talep tahmini daha sonra gerçek verilerle karşılaştırılarak test edilmiştir. Çalışmanın sonucunda BI simülasyon sonuçları, mağaza için %92,38'e varan doğruluk payı ile sunulmuştur.

Wang ve Yun, (2020). Talep planlaması (TP) ve satış tahmini (ST), başarılı tedarik zinciri analitiği elde etmek için iki kritik konudur. Genel olarak, TP, çeşitli bitmiş ürünler için gerekli olan ortak bir bileşen veya alt montaj için toplam talebin belirlenmesini ifade etmektedir. Talep belirsizliklerinin ve dinamik etkileşimlerin etkilerini vurgulamak için yapılan araştırmada, zorlukların üstesinden gelmek için yeni bir çerçeve önerilmektedir. Bu çerçevede; bilgisayar ürünlerindeki mevsimsel değişikliklerden ve döngüsel eğilimlerden kaynaklanan talep belirsizlikleri yakalanmakta, TP ve ST, ürün oynaklığındaki değişikliği dikkate almaktadır. DeneySEL sonuçlar, sunulan çerçevenin satış tahmini doğruluğunda başarı sunduğunu göstermektedir.

Abolghasemi ve ark.,(2020). Güvenilir talep tahminleri, çeşitli içsel ve dışsal değişkenlerden etkilenmektedir. Uygulamada, temel istatistiksel modeller tarafından oluşturulan tahminlerin, temel modellere dahil edilmeyen faktörleri ve bilgileri dahil etmek için genellikle tahminciler tarafından kullanılan değişkenlere göre ayarlanması tavsiye edilmektedir. Temel tahminler ayarlanırken insan müdahalesinin en aza indirilmesinde yarar görülmektedir. Çalışma kapsamında sunulan model ile, iki Avustralyalı şirketin satış ve promosyon verileri kullanılarak, karşılaştırmalar gerçek veriler ve tahmin rakamlarının karşılaştırılması biçiminde sunulmaktadır.

Sengar ve Ahmed, (2019). Tedarik Zinciri Yönetimi (TZY), ticari sektörde uygulamaya konan yeni kavramlardan biri olarak tanımlanmaktadır. Talep tahmini, TZY'nin ve onları destekleyen ve dijital dönüşümün ilk aşamasında olan simülasyon programları hakkında geniş bir literatür sunarak, müşterilerin talebini yeterince tahmin etme yeteneği açısından her programın başarı oranları hakkında bilgi verilmektedir.

Söz konusu simülasyon programları BI iş zekası adı verilen paket programa dayanmaktadır. Sunulan modelleme ise Micro Soft'un (MS) VIS kütüphanesine dayanmaktadır. Bu literatür taramasına göre MS tabanlı kütüphanelerin başarısının %70 oranında olduğu belirtilmektedir.

Mulyani ve ark., (2021). Honda Motosikletlerinin verileri kullanılarak satış tahmini planlamasına dair modellemenin yapılması amaçlanmıştır. Çalışmanın sonucunda, 2020 yılında satış tahmininin önceki yıllara göre 3250 adede kadar artacağı ön görülmektedir. Satış tahmini ve kazanılacak tutar hakkında da bir tahminleme yapılarak bu çalışma kapsamında araştırmacılar tarafından sunulan modelin ticari faaliyetlerde karar vermede bir rehber olarak kullanılabileceği ileri sürülmüştür.

Arslankaya ve Öz, (2018). Bu çalışmada otomobil satışlarının talep tahminini yürütmek için zaman serisi analizi, nedensel yöntemler ve yapay sinir ağları (YSA) kullanılmıştır. Çalışmada kullanılan zaman serisi ve YSA tahmin çıktılarının karşılaştırılması ile en iyi satış tahmin yönteminin hangisi olduğuna dair bir kıyaslama yapılmaya çalışılmıştır. Veriler gerçek verilerle karşılaştırılmış ve YSA sonuçlarının gerçeğe daha yakın olması nedeniyle modelleme daha büyük çapta yeniden kurularak büyük bir otomobil firmasında kullanılmak üzere dizayn edilmiştir.

Yang, (2016). Çalışmasında, bayileri talep tahmini, satış gücü ve envanter kontrolü dahil olmak üzere iç süreçlerine vurgu yaparak değerlendirmek için kontrol edilemeyen değişkenlere sahip bir ağ performans değerlendirme modeli önermektedir. Önerilen model, kurumsal verimliliği, dahili süreçlerle ilişkili ağırlıkların veri odaklı olduğu, ağırlıklı bir ortalama dahili süreç verimliliğine ayrıştırmaktadır. Model, Tayvan'daki 27 otomobil parçası satıcısından oluşan ampirik bir duruma uygulanarak test edilmektedir. Analiz sonuçlarında modelin %92 oranında başarılı olduğu gösterilmektedir.

Scheele ve ark., (2018). Satış bölümünün talep tahmini satış teşvik sistemi ve parasal ödemeleri ve bir de tahmin hatası içeren bir model oluşturmuştur. YSA tabanlı bu model ile davranışsal faktörler de dahil edilmiştir. Çalışmanın sonucunda insan

davranışları ile paralellik gösteren bir satış modeli kurulduğunda ürün satış tahmininin net sonuçlarına yaklaşılabildiği bilgisine yer verilmiştir.

Lasek ve ark., (2016). Çalışmada, talep tahmini, restoran satışları ve tüketici talep tekniklerinin kapsamlı bir literatür taraması ve sınıflandırılması sunulmaktadır. Literatürde tahmin için bir dizi metodoloji ve model yedi kategori altında incelenmektedir. Talep tahmini, restoran yönetimi için operasyonların planlanmasında çok önemli bir rol oynadığı öne sürülmektedir. Literatürde verilen bir dizi metodoloji ve modelden gelen teknikler (1) çoklu regresyon, (2) Poisson regresyon, (3) üstel düzeltme ve Holt-Winters modeli, (4) AR, MA ve Box -Jenkins modelleri, (5) sinir ağları, (6) Bayes ağı ve (7) hibrit yöntemler başlıkları altında her biri ayrı ayrı değerlendirilerek sunulmaktadır.

Xie ve ark., (2017). Otomobil ve ağır ekipman gibi sermaye yoğun ürünler, hem yaş hem de kullanım göz önüne alındığında genellikle iki boyutlu bir garanti politikası kapsamında satılmaktadır. Ürünler müşterilere aralıklı olarak satıldığından, satış süreci genellikle belirsizlik altında gerçekleştirilir. Bu stokastik yapı, zaman içinde garanti talebini tahmin etmede zorluklar sunar. İki boyutlu garantiye ilişkin mevcut literatür, yalnızca satış dinamiklerinin satılan birimlerin toplam garanti talepleri üzerindeki etkisini göz ardı eden tek bir birimin garanti talebine odaklanmaktadır. Bu sorunu ele almak için, yeni bir garanti analizi perspektifinden bu çalışma, belirli bir zaman dilimine kadar satılan tüm birimlerin toplam onarım talebi için genel bir tahmin tekniği geliştirmektedir. Stokastik satış süreci homojen olmayan bir Poisson süreci ile modellenmektedir. Zamana göre değişen satış süreçleri ile garanti taleplerinin tahmin edilmesinde modelin uygulanabilirliğini ve esnekliğini göstermek için sayısal deneyler sunulmaktadır.

Benhamida ve ark., (2021). Çalışmada, talep tahmini, gelecek dönemler için zaman serisi verileri için talep tahmin yöntemleri hakkında kapsamlı bir literatür taraması sunulmaktadır. Literatür taramasının ardından analiz sonuçları ve bulgularına dayalı olarak, envanter kontrolü için yeni bir talep tahmin aracı önerilmektedir. Çalışmada, ilk olarak, en doğru talep tahmin yönteminin seçilmesine izin vermek için bir tahmin

hattı tasarlanmaktadır. Önerilen çözümün doğrulaması, büyüyen bir çevrimiçi perakende platformu olan Stock&Buy vaka incelemesiyle sağlanmaktadır. Değerlendirme sürecinde, önerilen aracın her bir ürün seçimi için uygulanacak tahmin yöntemini tanımlarken en uygun seçimi yaparak iyi bir tahmin doğruluğu sağladığını göstermiştir.

Bakri ve ark.,(2019). Bu çalışmanın amacı, satış işlem verilerini tahmin etmek için Auto Sales Forecasting yazılımını uygulamaktır. Oto Satış Tahmini yazılımı, görselleştirmesi ile birlikte tanımlayıcı analiz ve tahmin özellikleri olmak üzere iki ana özellikten oluşmaktadır. Otomatik Satış Tahmini uygulamasında yer alan tahmin yöntemleri, Basit Hareketli Ortalama, Sağlam Üstel Düzeltme, Otomatik ARIMA, Yapay Sinir Ağı, Holt-Winters ve Hibrit Tahmin yöntemleri olarak sunulmaktadır. Araştırma sonucunda, Otomatik Satış Tahmini yazılımının satış işlem verilerini başarıyla analiz ettiğini gösterilmektedir.

2.3. WEKA Kullanımı ile Tahmin ve Analiz Araştırmaları

Farid ve ark., (2021). Araştırmada, hisse senedi fiyat tahmini, finansal zaman serilerinin, gelişmekte olan bir piyasada veri madenciliği teknikleri kullanılarak hisse senedi fiyatları tahmin edilmeye çalışılmaktadır. Çalışmada, WEKA yazılımı kullanılarak analiz için karar ağacı modeli, CRISP-DM (Veri madenciliği için Endüstriler Arası Standart Süreç) kullanılmaktadır. Çalışma örneklemini, Pakistan Menkul Kıymetler Borsası'nda (PSX) işlem gören beş farklı sektörden on firma oluşturmaktadır. Bulgular doğruluk oranlarının %50 ile %60 arasında değiştiğini göstermektedir. Bulgular, piyasa katılımcılarının önceki hisse senedi fiyatlarına gömülü bilgileri göz önünde bulundurarak daha yüksek getirileri açıklayabileceklerini göstermektedir. Ayrıca yatırımcıların, bulgular ışığında daha ihtiyatlı alım-satım kararları alabileceğini göstermektedir.

Zontul ve ark., (2013). Çalışmada Türkiye'nin batısında yer alan Kırklareli ilinin rüzgar enerjisi potansiyeli incelenerek bir analizi yapılmıştır. 2001-2007 yılları arasındaki istatistiksel veriler kullanılmıştır. Veriler Devlet Meteoroloji İşleri'nin

Kırklareli şubesinden elde edilmiştir. Kıbrıseli bölgesinde, yapılması planlanan rüzgar santralleri için 2013 yılına ilişkin rüzgar hızı tahminleri yapılmıştır. Yapılan analizler için WEKA aracı kullanılmıştır. Tahmin için kullanılan algoritma, karar ağacı algoritması olan REPTree'dir. Eğitim seti üzerinde çapraz doğrulama yöntemi uygulanarak gerçek ve tahmin edilen değerler arasındaki korelasyon katsayısı değeri 0,8154 olarak elde edilmiştir. Bu, REPTree'nin 2013 yılı rüzgar hızı tahmini için Torbalama yöntemiyle birlikte kullanılarak daha iyi bir sonuç alınabileceğini göstermektedir.

Massaro ve ark., (2018). Araştırmada, süpermarket ürün kaplamasını içeren Küresel Dağıtım (GDO) uygulamalarında görsel mağazacılık için uygun yapay sinir ağlarının (YSA) farklı Çok Katmanlı Algı Modelleri (MLP) önerilmektedir. Çalışma, bir test rafına tahsis edilen farklı ürünleri analiz ederek satış tahmininin aralık geçerliliğini vurgular. Çalışma, test ve eğitim veri kümelerinin nasıl işlenebileceğini açıklayarak en garanti sonuçları analiz edebilen doğru prosedürleri gösterir. Tahmin sonuçları, raf tahsislerini, genel satış trendini ve promosyon faaliyetlerini dikkate alarak aylık bir planogram tasarlamak için faydalıdır. Ön korelasyon analizi, öngörülen çıktıların yenilikçi bir anahtar okumasını sağlamıştır. Test, MLP ANN tarafından deneysel veri setinin her bir niteliğini tahmin edebilen Weka ve RapidMiner araçları tarafından gerçekleştirilmiştir. Son olarak, MLP ANN RapidMiner algoritmasının girdisi olarak Weka tahmin çıktılarını birleştiren yenilikçi bir hibrit model formüle edilmiştir. Uygulamada, deneysel veri kümeleri birkaç veriden oluştuğunda yararlı olan yapay bir test veri kümesinin kullanılmasına izin verilerek, modelin kendi kendine öğrenme süreci hızlandırılmaktadır. Çalışmanın sonucunda, MLP ANN RapidMiner algoritmasının girdisi olarak Weka tahmin çıktılarını birleştiren yenilikçi bir hibrit modelle formüle edilmektedir.

2.4. Matlab Kullanımı ile Tahmin ve Analiz Araştırmaları

Jayanthi ve Florence, (2019). Yazılım hatası tahmini için çeşitli teknikler sunulmuştur. Araştırmada yazılım hata tahmini için önemli bir paradigma olarak makine öğrenimini kullanan veri madenciliği önerilmektedir. Önerilen yaklaşım, öznel azaltmanın iyi

bilinen ilke bileşen analizi (PCA) şemasıyla gerçekleştirildiği ve PCA veri yeniden yapılandırmasında hata azaltma için maksimum olasılık tahmininin dahil edilmesiyle daha da geliştirildiği bir özellik azaltma ve yapay zeka konseptini içermektedir. Araştırma sonucunda, MATLAB simülasyon aracı kullanılarak performans analizi için KC1, PC3, PC4 ve JM1 gibi dört veri kümesinin dikkate alındığı NASA yazılım veri kümesinde bir çerçeve formüle edilerek uygulanmaktadır.

Kshatri ve ark., (2021). Araştırmada, topluluk öğrenme yöntemi, yeni örnekler üretmek için öğrenilen sınıflandırıcıların tahminlerini toplamak için uygulayan işbirlikçi bir karar verme mekanizması olarak tanımlanmaktadır. Suçların dinamik doğasını belirlemek amacıyla, MATLAB kullanarak öğrenmeye dayalı yöntemleri uygulayarak uygun suç tahminlerini belirlemek için SVM algoritmalarına dayalı birleştirme-yığıma tabanlı suç tahmin yöntemi (SBCPM) adı verilen verimli bir otantik yöntem önerilmektedir. SVM algoritması, başka bir makine öğrenimi modeli J48, SMO Naïve byes bagging ve Rastgele Orman ile karşılaştırıldığında alana özgü konfigürasyonları elde etmek için uygulanmaktadır. Araştırma sonucunda, önerilen yöntem, test verilerinde %99,5 sınıflandırma doğruluğu elde edilmiştir.

Scherbak,(2020). Araştırmada, bilimsel araştırmaların bilinen sonuçlarının analizine ve genelleştirilmesine dayanarak, nesne - model - ölçü - deney - sonuç baskısında bilimsel araştırmaların yeni metodolojisi sunulmaktadır. Önerilen metodoloji ile ilgili her bir araştırma konusunun açıklaması verilmektedir. Zaman aralıklarını ve sınırlı alan alanlarını sınırlamak için gösterge fonksiyonlarını kullanan kaynak tüketim süreçlerinin çok boyutlu matematiksel modelleri göz önünde bulundurulmaktadır. Araştırma sonuçlarının nicel bilgilerini elde etmek için fiziksel olasılık ölçütleri kısaca açıklanmaktadır. Teorik araştırma, simülasyon ve saha deneyleri yapılırken, uygun bilgi desteği, dağıtık hesaplama için yazılım ortamları (Matlab, Mathcad, Statistica, Anaconda vb.) ve ağlar kullanılarak kullanılmaktadır.

Abraham ve Shrivastava, (2018). Araştırmada, derin öğrenme algoritmaları kullanılarak kontrolör tasarımı ve ayarlanması alanında birçok araştırma yapılmıştır. Çalışmada, Matlab/Simulink paketindeki mevcut araçlar ve kütüphaneler kullanılarak

tasarlanmış bir orantılı integral türev denetleyicisi ile bir sinir ağı denetleyicisi arasındaki etkinlik ve performans üzerine bir karşılaştırma sunulmuştur. Başlangıçta orantısal integral türev denetleyici, sağlamlık ve daha hızlı yanıt arasında bir denge olan bir DC motor modeli için ayarlanmıştır. Daha sonra aynı model için sinir ağı denetleyicisi eğitilmiş ve ardından denetleyicinin optimizasyonu yapılmıştır. Daha sonra her iki kontrolör de aynı model DC motor kullanılarak Matlab/Simulink'te simülasyon ile denenmiştir. Bunu takiben, motorun eylemsizlik momenti değerlerinin değiştirilmesi ve ardından denetleyiciyi daha yeni model için yeniden ayarlamadan denetleyici yanıtının simüle edilmesiyle başka bir deney yapılmıştır. Sonuç analizi, hem çizilen grafikten kontrolör yanıtı hem de ortalama kare hata hesaplaması incelenerek yapılmıştır. Simülasyon deneyleri ve kontrolörler arası karşılaştırma, makine öğrenmesi algoritmalarına sahip kontrolörün kontrol sistemlerinde daha uzun süre daha iyi olduğu gösterilmiştir.

Jingwei ve ark., (2016). Çalışmada, PR1400 kaynak robotu araştırma nesnesi olarak alınarak yapı ve bağlantı parametreleri analiz edilerek ve koordinat sistemini kurmak üzere standart Denavit-Hartenberg yöntemi uygulanmıştır. Daha sonra robot kinematiği analiz edilir. Matlab Robotics Toolbox, model ve yörünge planlama simülasyonunu oluşturmak için kullanılır. Literatürde, altı bağlantı için çevrimdışı yörünge planlamasının matematiksel simülasyon modeli oluşturulmuş ve simülasyon sonuçları, SolidWorks kullanılarak kurulmuş bir kaynak Robotunun katı modeli için MATLAB kullanılarak sezgisel ve dinamik olarak gösterilebilmiştir.

Khamila ve ark., (2020). Çalışmada, makinelerden, sistemlerden ve altyapıdan üretilen atık enerji, hem Peltier hem de Seebeck modülü için analitik modellemeyi, kullanıcının uygulamasına bağlı olarak hızlı değerlendirme için gereken ana parametreler açısından, örneğin voltaj, akım, performans katsayısı ve verimlilik, termal öz direnç, toplam iç direnç ve Seebeck katsayısı açısından incelemektedir. Bu parametreler normalde modülün üreticisi tarafından veri sayfası aracılığıyla verilmektedir. Temel denklemlerden simülasyon yapmak için MATLAB/Simulink kullanılmaktadır. Çıktının grafik gösterimi, Matlab komut penceresinde çeşitli kodlar kullanılarak oluşturulabilmektedir. Çalışmadan elde edilen sonuçlar, MATLAB/Simulink'in

modellemesini kanıtlayan üretici tarafından sağlanan veri sayfasıyla uyumlu görülmektedir.

Abatan ve Olayemı, (2014). İstatistiksel yazılımın veri analizindeki nicel araştırma büyüdükçe, istatistiksel yazılım (SS) uygulaması veri analizinin daha önemli bir parçası haline gelmektedir. İki fakültedeki 8 bölümden her biri 5 öğretim üyesinden oluşan bir kesitsel anket seçilmiş ve uygunlukları ve ilgilerine göre anket uygulanmaktadır. %80 ile (1-5) Likert ölçeğinde Ortalama (M) =4,80 ve Standart Sapma (SD)=0,41 verdiğini ortaya koyulmaktadır. Yanıt verenler kategorisi ve SS'siz çalışan analiz, ortalama (M) = 2,27 ve Standart sapma (SD) = 1,37'yi (1-5) Likert ölçeğinde gösterilmektedir ve %66'sı SS olmadan analiz yapamayacaklarını kabul edilmektedir.

Venkata ve Narayana, (2020). Bu makale kalp sesi sinyallerinin yani Fonokardiyoloji (PCG) sinyallerinin işlenmesiyle ilgilidir. Elde edilen kalp sesi sinyalleri Seviye 1'den 10'a kadar ayırık dalgacık değişimi kullanılarak alt gruplarına izole edilmiştir. Bu strateji, sinyalleri düzenlemek için kaldırılan ve başka bir alana değiştirilen Kalp Sinyali Özellikleri için en iyi bileşeni önermiştir. Önerilen stratejide, vurgu çıkarma için Dalgacık kullanılmıştır ve farklı İstatistiksel stratejiler kullanılmıştır. MATLAB & WEKA Yazılımları analiz amaçlı kullanılmaktadır. Bu yazıda, sentetik PCG'yi analiz etmek için coiffelet tekniği kullanılmış ve sınıflandırıcı parametreleri birbirleriyle karşılaştırılmıştır.

BÖLÜM 3. MATERYAL VE YÖNTEM

Nitel ve nicel yöntemlerin birlikte kullanıldığı çalışmalar, birlikte kullanıldığı düzenler karma çalışmalar olarak bilinmektedir. Ancak bu çalışmada nitel çalışmalar daha çok literatür dahilinde kullanıldığı gibi, sayısal girdiler, bir amortisör firmasının 50 aylık satış raporundan elde edilen verilerle düzenlenmiştir. WEKA için veriler 2 kat büyütülmüş olarak kullanılmıştır. Bu çalışmada WEKA ve Matlab programları için aynı veri seti kullanılmış olup, COVID 19 dönemini kapsayan 2020 yılı Mart, Nisan ve Mayıs aylarındaki satış adetlerinin ne olacağı tahmin edilmiştir. İki program ile elde edilen çıktılar, Mart, Nisan ve Mayıs ayında gerçekleşen satış adetleri ile karşılaştırılarak hangi tahmin yönteminin daha gerçekçi olduğu tespit edilerek bulgular bölümünde sunulmuştur. Bu çalışmanın yöntemi yarı deneysel ampirik çalışma olarak kabul edilmektedir. Literatür taraması ve arka plan tamamen yapay zeka ve simülasyon programlarının kullanımını içermektedir. Modellemeler ise WEKA ve Matlab başlıkları altında ayrı ayrı incelemektedir.

3.1. WEKA Analiz Programı (Waikato Environment for Knowledge Analysis)

WEKA programı ([Weka], [Weka Wiki], [Weka, 2009]) Yeni Zelanda'daki Waikato Üniversitesi'nde (programın adı da buradan gelmektedir ve Türkçe deyişle “bilgi analizi için Waikato ortamı” gibi bir anlam taşımaktadır), Java ile geliştirilmiş bir kod yapısını kullanmaktadır. WEKA, GNU Genel Kamu Lisansı sürüm 21 (GPLv2) altında lisanslanmış ve veri ön işleme, sınıflandırma, regresyon, kümeleme ve birliktelik kurallarını arama problemlerini çözmek ve veri ve sonuç görselleştirme amaçlarıyla kullanılmaktadır.

WEKA programı, etkili analiz biçimlerini sunmanın yanı sıra, kullanıcılar tarafından kolay öğrenebilen bir paket yapısına sahip olma avantajını sunmaktadır. Paket

simülasyon programları içerisinde benzer programlar arasında iyileştirilmiş bir arayüze sahiptir. Ücretsiz bir program olmasının yanı sıra ön işleme ve veri görselleştirme konusunda yeni algoritmalara izin veren bir kullanıcı deneyimi sunmaktadır. Program aynı zamanda, araçlar ile desteklenmektedir.

Programın iyi yorumlanmış kodlama sistemi weka-src.jar dosyasında bulunmakta bu da java programlamaya aşina olmayan kullanıcıların normal bir zip arşiviyle çalışmasına imkan tanımaktadır. Başlangıç aşamasında RunWeka.bat dosyasını çalıştırmak yeterli olmaktadır. Dosya açıldığında, kullanıcıdan, dört program modülünden birini seçmesi istenen Weka GUI Seçici penceresi açılmaktadır.

Şekil 3.1.'de WEKA programının ana penceresine yer verilmekte olup, bu panelin girişi sunulmaktadır.



Şekil 3.1. WEKA programının ana penceresi.

Programın ana modülü “Gezgin Modülü” olarak adlandırılan kısmı kapsamaktadır. Bu kısımda, verilerin yüklenmesi işlemleri yapılmaktadır. Ön İşlem sekmesi, Sınıflandırma, Kümeleme, İlişkilendirme işlemlerine izin verilmektedir. Görsel işleme için ise, Özniteliklerin seçilmesine izin veren modül ve YSA için Makine öğrenmesinde eğitilecek modelin taslağını biçimlendirmesine izin veren kullanılabilecek görselden yeni görsel yaratma veya görselden veri arttırmaya izin veren sekmeler de yine gezgin modülü olarak adlandırılan kısımdan yapılmaktadır.

Genel pencerede her görevin kendi sekmesi bulunmaktadır. İlk başta, görevlerin geri kalanı veri gerektirdiğinden yalnızca Önışleme sekmesi kullanımı etkin görülmektedir. Veri yüklendikçe ilerleme sağlanabilmektedir.

Başlangıçta, ön işleme sekmesinde, pencerenin üst kısmında, Dosya Aç, İnternette URL Aç, bir veritabanından Open DB, veri indirme için Open DB ve model verisinin oluşturulmasına “Oluştur” komutlarını tanıyan dört aktif düğme bulunmaktadır. Çoğu zaman, bir dosyadaki verileri kullanmanız gerekir. "Dosyayı Aç" komutunu içeren düğmeyi tıklayarak, dosya uzantılarının listesini görüntülemek mümkün olmaktadır. Böylelikle WEKA'nın hangi formatları anladığını görebilmek mümkün olmaktadır. Bu formatlar; arff, CSV, C4.5, binary, libsvm biçimlerini içermektedir.

Şekil 3.2.'de WEKA'da desteklenen örnek dosya gruplarına yer verilmektedir.

```

example.arff

% комментарий
@RELATION myproblem

@ATTRIBUTE firstfeature REAL
@ATTRIBUTE secondfeature REAL
@ATTRIBUTE noise REAL
@ATTRIBUTE class {A,B}

@DATA
0,0,1.1,A
0,0,3.2,A
0,0,4.7,A
0,0,3.1,A
0,1,2.2,B
0,1,4.3,B
1,0,1.1,B
1,0,2.5,B
1,1,3.7,A
1,1,4.2,A
1,1,3.9,A
1,1,5.1,A

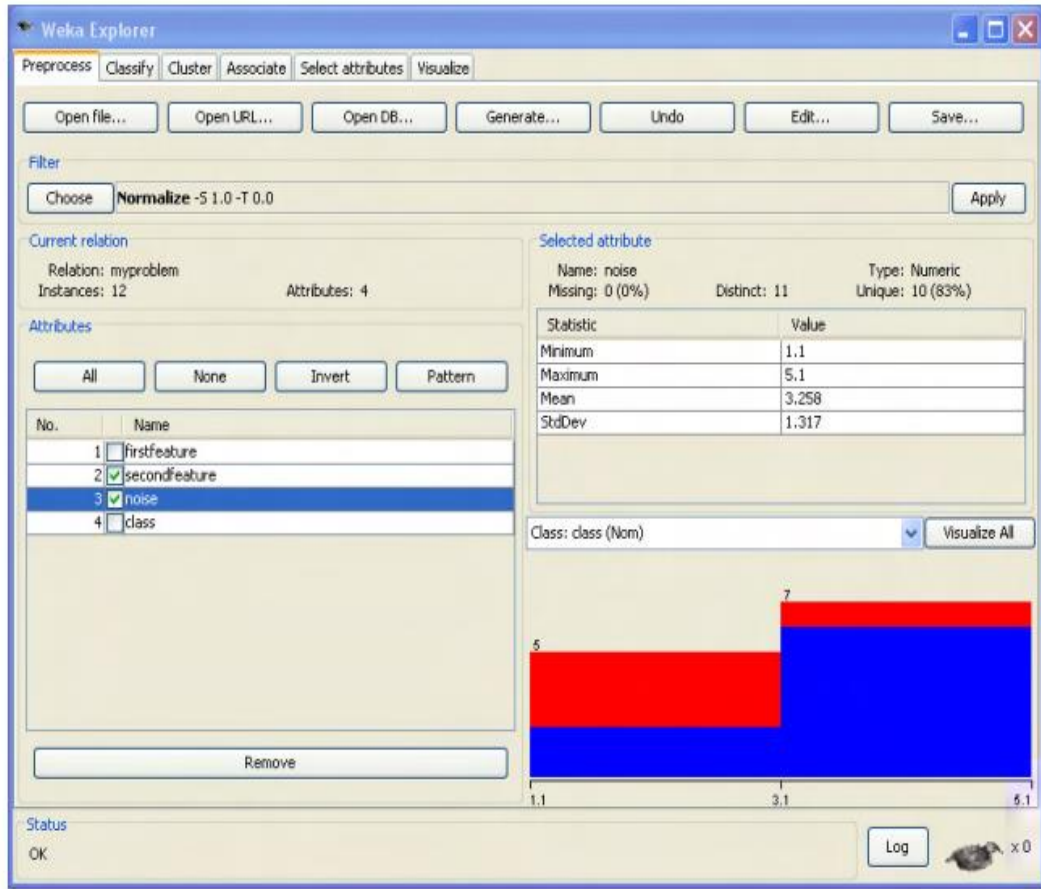
```

Şekil 3.2. WEKA'da desteklenen dosya biçimleri

Ana biçim arff olarak görünmekte, arff uzantılı dosyaların veri dizininde, arff dosyalarının tanımlanma ve okuma biçimlerinde RELATION bölümü görev adını, ATTRIBUTE sözcüğü dosyayı geri çağırmak için atanabilecek anahtar kelimeleri, DATA ise verilerin kendisini ifade etmektedir. Verilerin tamamı @DATA yazılarak çağrılabilir. Nominal ve sayısal işaretlere ek olarak, dize, tarih, ilişki türü işaretleri de tanımlanma sırasında görev başlığına tanımlanabilmektedir. Verilerde bir

boşluk söz konusu ise “?” (soru işareti)” konularak işleme devam etme özelliği de bulunmaktadır.

WEKA’da verileri kaydetmek için ise, dosyayı sol alt panelden seçerek “Nitelikler” sekmesine yüklemek mümkün olmaktadır. Nitelik seçimini ve veri kaydetmeyi gösterir ekrana Şekil 3.3.’te yer verilmektedir.



Şekil 3.3. WEKA veri Kaydetme

Şekil 3.3.’te nitelik seçiminin yanı sıra veri kaydetmeyi gösteren histogramda, seçilmiş nitelikler, panelin sağ tarafında görüntülenmektedir. Burada “Tümünü Görselleştir” seçeneğini tıklayarak tüm özelliklerin histogramlarını görüntüleyebilen bir seçenek de bulunmaktadır.

Bu düğmenin solunda, hedef özelliği seçme bileşeni yer almaktadır. Görselleştirme sırasında hangi özellik bir sınıf olarak kabul edilecekse seçim yapmaya imkan

tanınmaktadır. Bununla birlikte “NoClass” seçimi de bulunmaktadır. Bu sınıflandırmanın istenmediği durumlarda seçilmektedir. Böylelikle eğer bir görüntü işleme yapılmıyorsa ve sadece veriler arasında bir ilişik kurulmaya çalışılıyorsa bu seçeneğin işaretlenmesi imkanını tanımaktadır.

Panelin üst kısmında “Explorer modülünün” Önişleme sekmesi bulunmaktadır. Bu sekme ile veri ön işleme için, iki özellik vurgulanmaktadır.

Bu özellikler; öznitelikler ve basit istatistikleri kapsayan bir görüntüleri işleyen özniteliğin maksimum, minimum ve ortalama değerlerini, örnek varyansını, ihmal sayısını, öznitelik türünü ve benzersiz değerlerin sayısını esas alan nitelik seçilimi ile ilişkili durumda görülmektedir. Burada istenilen öznitelik seçilimi için, bir sayısal değer girişi yapılacaksa, görüntü işleme alanlarını kapsayan niteliklerin “Kaldır” seçeneği ile kaldırılması, onaylanacak alanların ise, “Nitelikler” panelinde “onay işaretleriyle” süzülmesi gerekmektedir.

Nitelikler seçildikten sonra, düzenle düğmesine basarak, niteliklerin değerlerinin düzenlenmesi olanağı bulunmaktadır. Tüm düzenlenmeler sağlandıktan sonra, “Kaydet” seçeneği ile verilerin kaydedilmesi sağlanmaktadır.

WEKA’da bulunan üst filtre panelinde, veri dönüşümleri, hiyerarşik liste, filtreler denetimsiz; nitelik; normalize biçiminde verilerin düzenlenmesi seçeneklerini barındırmaktadır. Panelde görülen filtrenin adı görülmekte, WEKA sisteminde veri ön işleme prosedürleri biçiminde adlandırılmaktadır. Seçilen bir niteliğin veri seti üzerinden kaldırılması arzu edildiğinde, seçimin iptali için, panel üzerine sağ fare tuşu ile tıklayarak parametrelerin değiştirilmesi sağlanabilmektedir. Ancak, bazı filtreler, ilgili veriye mutlaka uyarlanması gereken ve filtrelerin hangi özelliklere uygulanması gerektiğini belirleyen parametrelere sahip olarak sabit kalmaktadır. Bu özelliklerin değiştirilmemesi tavsiye edilmektedir.

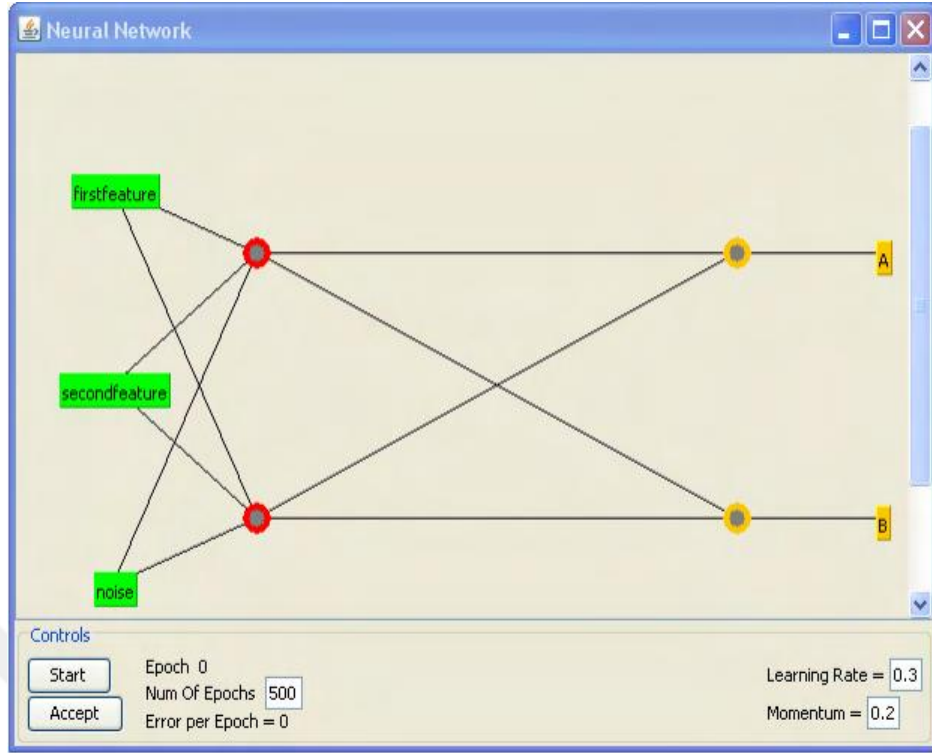
Tavsiye edilen parametrelerin haricindeki diğer seçenekler işaretlenerek, “Uygula” komutuna tıklayarak filtenin uygulanması başlatılmaktadır. WEKA’da yer alan yapay

zeka katmanının etkisi yalnızca üçüncü özellik seçenekleri için normalleştirme işlemini gerçekleştirmektedir. Buradaki değerler doğrusal olarak (gelişmiş YZ için sigmoid fonksiyonunu görmezden gelerek) değerleri $[0,1]$ segmentine aktararak sunmaktadır. Yani YZ ilk dönemlerinde olduğu gibi XOR sorunun bir parçasını WEKA barındırmaktadır ve program katmanları arasında $[-1,1]$ aralığını kullanmamaktadır. Bu durum, yeni model geliştirse de gerçek zamanlı hareket içeren modellerin kurulması ve işlenmesinde çok elverişli değildir. Ancak bu tez konusundaki tahminleme ve iki boyutlu görseller için belirgin işlevleri nedeniyle tercih edilmektedir. WEKA'nın bir paket program olması ve özellikleri ekleme çıkarma işlevleri iki katmanlı bir yapı olarak işlevlilik sunmaktadır. Bu işlevler; hesaplamalar için parametrelere göre hareket eder ve çalışan işlemlerin çıktısının alınmasıyla program durur ve “x” simgesi ile yapılan işlemlerin tamamı gösterilir biçiminde bir kurguya sahiptir.

WEKA'nın “sınıflandırıcı paneli” kullanıcının, bir sınıflandırıcı seçmesine olanak tanımaktadır. Yapay sinir ağları bu panelde, koordinat düzleminin X,Y,Z boyutlarını alt durum (eksi değer) olmaksızın ele almaktadır.

WEKA GUI olarak adlandırılan seçici pencerede, sınıflandırma seçeneği için “Seç” komutu kullanılmaktadır. Bu komut tıklandığında, hiyerarşik bir liste açılmaktadır ve kullanıcının ilgili seçimleri yapmasına yani parametreleri belirlemesine olanak tanımaktadır. Örneğin seçimler için izlenecek hiyerarşik düzenlemede “weka—sınıflandırıcılar—işlevler--MultilayerPerceptron (çok katmanlı Yapay Sinir Ağı)” sıralaması takip edilmektedir.

WEKA'da filtrelerin seçiminde, “Sınıflandırıcı” panelinde görünen isme sağ tıklamak, “sınıflandırıcı parametrelerinin” seçilmesine izin vermektedir. Çok katmanlı bir yapay sinir ağı için GUI parametresi değeri “true” olarak belirlendiğinde, yapay sinir ağının yeniden düzenlenmesine imkan tanınmaktadır. Şekil 3.4.'te bu MultilayerPerceptron'un (Çok Katmanlı YSA'nın) düzenlenmesine ilişkin ekran çıktısına yer verilmektedir.

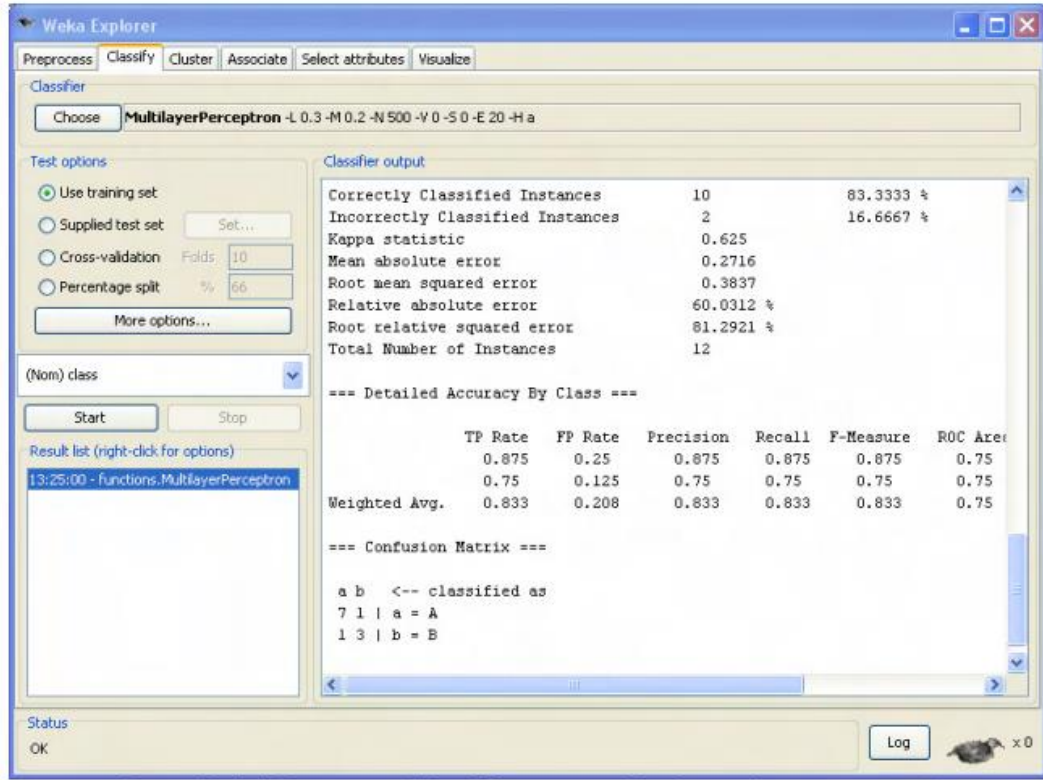


Şekil 3.4. WEKA’da yer alan Düzenlenebilir Yapay Sinir Ağının (YSA) görünümü

Düzenlenebilir YSA yapısının test edilmesi için test yöntemi seçenekleri bir panelde sunulmaktadır ve burada düzenlenmiş YSA’nın nasıl test edileceği tanımlanmaktadır. Bu panelde, “sınıflandırıcı” verilerin işleneceği eğitim seti içerisinde bir kısmını eğitim, bir kısmını test ve bir kısmını doğrulama olarak kullanılmasını içermektedir. Buradaki sınıflandırıcı ifadesi, bir öznitelik çıkarmaktan çok eğitim setine dayalı bir model kurulmasını içermektedir ve “eğitim, test, doğruma” olarak verilerin sınıflandırılması anlamını taşımaktadır.

“Sınıflandırma” işlemi için ayrı bir dosyada (veri test seti), bloklarla çapraz doğrulama, ilk örneği eğitim ve kontrole bölen “yüzde bölünmesi” gibi test parametrelerinin belirlenmesi gerekmektedir. Burada örneğin, çapraz doğrulama seçildiğinde, numunenin kaç bloğa bölüneceğini ayrıca belirtmek gerekmektedir.

Sınıflandırmayı sağlayan çok katmanlı YSA yani MultilayerPerceptron ekranı ve seçilecek birleşenleri gösteren ekran görüntüsüne Şekil 3.5.’te yer verilmektedir.



Şekil 3.5.WEKA Programının Sınıflandırma Sekmesinde Eğitim Verileri ve Test Bileşenlerinin Ekran Görüntüsü

Şekil 3.5.'te görüleceği üzere, “başlat” komutu ile “sınıflandırıcı” verileri okumaya ve hazır modeli eğitmeye başlamaktadır. Bu panelin çıktısı, eğitim raporunu görüntüleme amacıyla kullanılmaktadır. Sınıflandırıcı eğitim çıktısında, görev ve sınıflandırma algoritması hakkındaki her bilgi görüntülenebilmektedir. Şekil 3.6.’da eğitim modelinin çıktı görünümüne yer verilmektedir.

```

=== Run information ===

Scheme:      weka.classifiers.functions.MultilayerPerceptron -L
0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation:      myproblem-
weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0
Instances:    12
Attributes:   4
              firstfeature
              secondfeature
              noise
              class
Test mode:    evaluate on training data

```

Şekil 3.6. Veri Setinden Eğitim Sınıfı Olarak Tanımlanan Küme ve Modelin Eğitim Çıktısı Ekran Görüntüsü

WEKA, model eğitiminden sonra, sınıflandırma sonuçları ve çıktı tahminleri sunabilecek bir akışa izin vermektedir. Bu durumda, eğitim sınıflandırmaları

kullanımda, herhangi bir değerin birinci basamak eğitim sınıfında tanınmasını sağlamaktadır.

WEKA sınıflandırıcının çalışma düzeninde, test verileri ile karşılaştırma yaptığını ve olasılık tahminlerini de doğrulama cevaplarıyla karşılaştırdığını ve cevapların yüzdelik dilimi halinde $l \times l$ boyutunda bir matris ile hesapladığını burada vurgulamak gerekmektedir. Yani WEKA'ya tanımlanan veriler, eğitimde ayrılmış doğrulama verileri ve eğitim modelindeki verilerle bir ilişki kurmakta ve olasılıklar bağlamını yaratmaktadır. Şöyle ki; WEKA'ya bir sınıf tanımlandığında örneğin bu sınıf eğitim verisi (EV) olarak gruplandırılmış olsun, $l \times l$ matrisin, x değeri EV'nin bir elemanıdır ve x ögesi, x^i -EV sınıfına atanan nesnelerin sayısına eşit olarak dağılmaktadır. Doğru sınıflandırılmış nesnelerin sayısı, ana köşegen üzerindeki ögelerin toplamına eşit olduğu anlamına gelmektedir. Çıktı istatistiklerinde her değer değer True Positive (TP) yani pozitif bir tam sayı ile sunulmaktadır.

“Geri Çağırma” komutu ise, sınıflar için, sınıfın doğru sınıflandırılmış nesnelerinin yüzdesine eşit olarak dağılım göstermekte ve matrisin bir köşegen ögesinin satırındaki ögelerin toplamına bölünmesiyle elde edilir. Eksi bir değer WEKA'da yer almadığından (XOR sorunu), bu cins bir veri, Yanlış Pozitif (FP) değeri olarak kabul edilmektedir. FP söz konusu ise sınıfa yanlışlıkla girilen herhangi bir değer, diğer sınıfların nesnelerinin yüzdesine eşittir ve söz konusu sınıfın bir satırını matristen silerseniz, değer toplamını yeniden eşitlemektedir. Bir sınıfın sütununun ögeleri, tüm ögelerin toplamına bölünerek kesinlik değeri elde edilmektedir.

Kesinlik değeri, dikkate alınan sınıfa algoritma tarafından atanan nesnelerden doğru sınıflandırılmış nesnelerin yüzdesine eşit bir değer sunmaktadır (köşegen elemanın sütun elemanlarının toplamına oranı biçiminde bir matris değeri ile hesaplamaktadır).

$$\frac{F\text{--Ölçü değeri} \times 2 \times \text{Kesinlik} \times \text{Geri Çağırma}}{\text{Hassas} + \text{Geri Çağırma}}$$

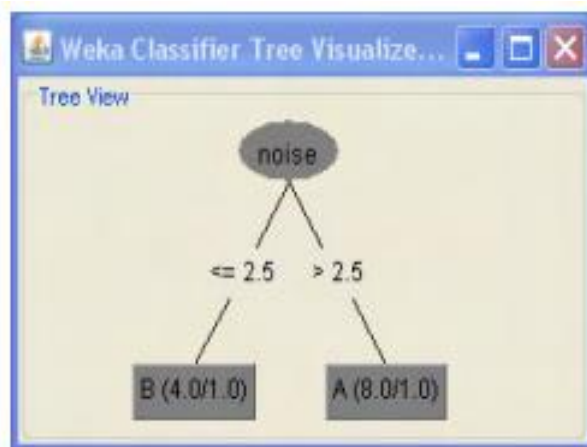
Şeklinde basit bir hesaplama biçiminde otomatik olarak yapmaktadır.

WEKA programı, burada ifade edilen “Kesinlik sayısı”nı, “Geri Çağırma sayısı”nın harmonik ortalaması şeklinde almaktadır.

Sınıflandırma çalışmasının “Sonuç Listesi”nin sol alt bölümü, tüm veri sınıflandırma çalışmalarının bir listesini sunmaktadır. Bu liste, sınıflandırıcının “Başlat” komutu tuşuna tıklayarak görüntülenmektedir. Konfigürasyon ve test işleminin tamamlanmasını takiben, listeye yeni bir öğe eklenmiş olarak görüntülenmektedir. Sınıflandırmanın bitiş zamanını ve sınıflandırıcının adını göstermektedir.

Buradaki liste öğesi, farenin “sağ tık” hareketi kullanılarak raporun ayrı bir pencerede görüntülenmesi sağlanmaktadır. Bu görüntüleme biçimi, üzerinde çalışılan ve eğitim verisi ile eğitilen modelin, yüklenerek kaydedebilmesini sağlamaktadır. Buradaki işlem listesi; “sınıflandırıcının türü ve parametreleri seçimi, sağ tık görüntüle, modeli yükle ve modeli kaydet” akışını takip etmektedir. Sınıflandırıcı hatalarının örneğin; girinti ve hata eğrileri gibi hataların görselleştirilmesi için sadece “görselleştir” komutu kullanılmaktadır.

Sınıflandırmada en yakın komşu (k-NN) değerleri ile Sınıflandırmanın Ağaç Görünümü de elde edilebilmektedir. Şekil 3.7.’de sınıflandırmanın ağaç görüntülenmesine ait ekran görüntüsüne yer verilmektedir.



Şekil 3.7. Ağaç görselleştirme sonucunu

Gürültü özelliğine sahip XOR probleminde, ağaç bu özellik ile tek ile oluşturulmuş olarak görülmektedir. Bunun sebebi WEKA'nın sigmoid fonksiyonu yerine temel bileşen kümesini kullanmasıdır.

Hataların görselleştirilmesi kısmında, özel işaret yanlış özellik çiftleri üzerine izdüşümlerdeki sınıflandırılmış nesneler üzerinde ve yalnızca sınıflandırıcı hatalarını görselleştirilmesi seçeneği belirlendiği durumlarda kullanılabilir olmaktadır. Görselleştir sekmesinde, tüm verileri ve verilere ait gerçek sınıflandırmalarını görebilme imkanı sunulmaktadır. Ancak bu sekmede “sınıflandırıcı hatalarını görselleştir” komutu özel olarak seçilmemiş ise “sınıflandırma hataları” görüntülenememektedir.

Görselleştirme Sekmesi; seçimi görselleştirmenizi sağlamaktadır. Seçki imgesi ise onaylanan tikleri görme imkanı tanımaktadır. Nitelikler, görselleştirme için özelliklerin seçilmesine izin vermektedir. Bu alan daha çok “görüntü işleme” kümelerinde “öznitelik çıkarma” ve “gürültünün temizlenmesi” işlemleri ile bağlaşıklık durumunda hareket etmektedir. Dolayısıyla görselleştirme sekmesi daha çok “Yapay Sinir Ağı” na geçişten önce en yakın komşu yani k-NN işlevini yürütmektedir.

Bu YSA kısmında Makine Öğrenmesi kısmı ile bağdaşık bir yapı olarak literatürde ele alınmaktadır. Bu noktada K-NN'nin kısa bir biçimde tanıtımı gerekmektedir.

Öğrenme aktarımı kullanılarak öznitelik çıkarımı ile birlikte destek vektör makineleri (SVM) ve K-NN (K-Nearest Neighbor) en yakın komşu sınıflandırması algoritmaları WEKA programında, kullanılan YSA biçimleri olarak kullanılmaktadır.

Vapnik, Destek Vektör Makinesi algoritmalarını 1963 yılında tanıtmıştır (Vapnik, 1963: 774–778). Destek vektör makineleri (SVM), denetimli öğrenme modellerinden biri olarak sınıflandırmanın temel YSA eğitiminde kullanılan sinir ağının eğitilmesini sağlayan modellermeyi de göstermektedir. SVM'lerde girdi olarak verilen örnekler göre çıktı olarak neyin beklendiği etiketlenmektedir. SVM' nin amacı, bir veri setini

iki sınıfa ayıran bir hiper düzlem veya bir karar yüzeyi bulmaktır (Akinsola, J.E.T. , 2017: 132).

Hiper düzlemi belirleyen veri noktaları "destek vektörleri" olarak adlandırılmaktadır (Huang ve ark., 2017: 47). Veri noktalarını iki sınıfa ayıran birçok olası hiper düzlemde asıl amaç, n-boyutlu bir alanda "azami verim" e sahip bir köprü bulabilmektir. Buradaki verim, iki sınıf arasındaki mesafe için hiper düzlemin boyutuna bağlı olarak, artmakta veya azalmaktadır. Bu da giriş özelliklerinin sayısı ile ilişkili bir bağlantı bulunmasını kolaylaştırmaktadır. SVM, doğrusal olmayan modellerde iki sınıf arasındaki mesafeyi hesaplamak için "çekirdek" adı verilen bir mekanizma kullanmaktadır. Çekirdek seçimi, yerleşik SVM modelinin performansına etki etmektedir.

Doğru çekirdeğe karar vermek için en iyi yol, veri üzerindeki çekirdek işlevlerinin ayarlanarak denetlenmesini sağlamaktan geçer. Doğrusal destek vektör makinesi, karar düzlemi sınıfları mükemmel bir şekilde +1 ve -1 düzleminde ayırır(Srivastava ve Lekha, 2010: 24).

K-NN (K-Nearest Neighbor) algoritması en basit ve en sık kullanılan sınıflandırma algoritmasından biridir. K-NN eğitim aşaması olmayan bir yöntemdir. Yani tembel tabir edilen bir öğrenme algoritmasını sunmaktadır. Bu yöntemde, eğitim verilerinin öğrenilmesi yerine, “eğitim veri kümesi “ezberlenerek ilerleme sağlanmaktadır” (Chih-Min ve ark., 2014:172).

Bu bağlamda, K-NN’de bir tahmin yapmak istenildiğinde, K-NN en yakın komşu öğrenmesi temelinde, tüm veri setinde bulunan en yakın komşuları aramaktadır. Bu öğrenme biçiminin dezavantajı uzaklık hesaplanırken bütün uzaklıkları model gereği saklandığı için çok büyük veride yüksek bellek alanına ihtiyaç duyması biçiminde açığa çıkmaktadır.

K-NN, algoritmanın çalışmasında bir K değeri belirleyerek bir akış sunmaktadır. Bu K değeri, değer verilen noktaya, en yakın uzaklıktaki komşu sayısına göre

belirlenmektedir. Kullanılan algoritma, uzaklık fonksiyonları ile yeni noktanın mevcut noktalara göre tek tek uzaklığını hesaplanmakta ve en yakın k-komşuların özniteliklerine göre komşuların sınıfına yerleştirilmekte olduğu sonucunu mantıksal çıkarımını sunmaktadır. Uzaklık hesaplama işleminde genelde Öklid fonksiyonu kullanılmaktadır. Ancak Öklid fonksiyonuna alternatif olarak Hamming, Manhattan ve Minkowski fonksiyonlarına da başvurulduğu görülmektedir (Özarı ve Eren, 2019: 1125-1126).

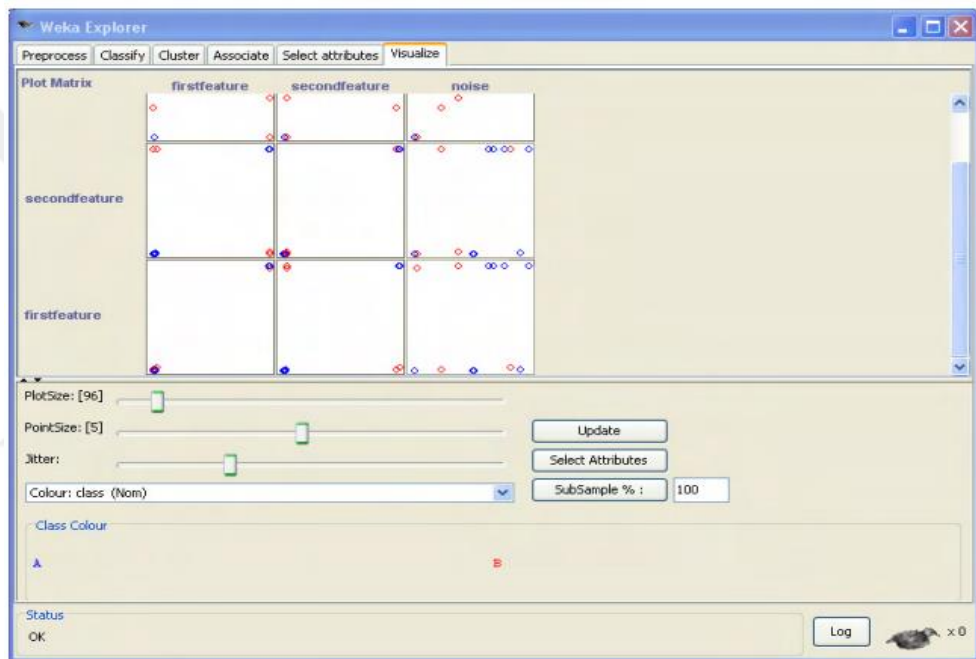
WEKA için K-NN kullanımı ve ilerleyen bölümde ele alınacak Destek vektör makineleri (SVM) öğrenmeleri tamamıyla, bu tez çalışmasında arka plan olarak sunulan YSA'nın bu iki temel öğrenme düzleminin pratik çıktıları ile işlem yapan hızlı programlama olarak tanımlanmaktadır.

Bu kısa bilgi notunu takiben WEKA kullanımında, projeksiyon görüntülerinin, tembel öğrenme olarak adlandırılan ezberleme temelinde şekillendirilmiş olduğunu söylemek mümkün olmaktadır.

Bu özellikler, modellemenin her türlü çifti üzerine inşasına imkan tanırken, özel modelleme durumunda, daha net sonuçların elde edilebileceğini belirtmeyi kaçınılmaz kılmaktadır. Çünkü; K-NN ve SVM'ler, özel modelleme kurgularında daha fazla katman ile kullanılırken paket program olan simülasyon çalışmalarında, imkan tanıdığı modelleme biçimleriyle hareket etmektedir.

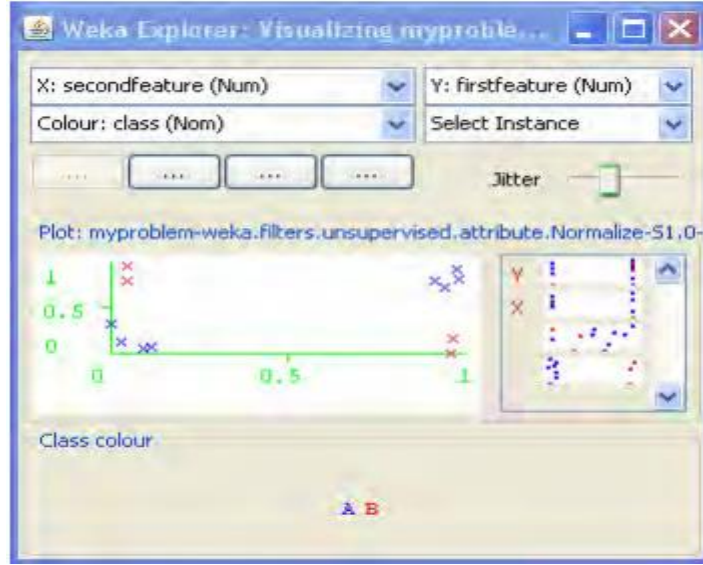
WEKA programının "PlotSize" olarak adlandırılan kaydırıcısı da bu kısımda açıklanan K-NN algoritmasının temel akışını basit mantıksal çıkarımlarla sunmaktadır. WEKA'nın görselleştirme sekmesinin bir aracı olan PlotSize resimlerin boyutunu, PointSize ise nesneleri temsil eden noktaların boyutunu, Jitter ise özelliklere atanmış (daha çok WEKA'da ek özellik seçiminde imkan tanındığı boyutta, seçeneklerin belirlenmesi kısmını içermektedir) eklenen gürültü seviyesini seçmek üzere kullanılmaktadır.

PlotSize olarak adlandırılan ve “resim kaydırıcı” olarak dilimize uyarlayabileceğimiz araç, WEKA modeli için oldukça önemli bir görselleştirme bileşeni aracıdır. Bu araç WEKA’da “az sayıda değer alan bir çift özellik için k değerli olanlar” biçiminde bir gruplandırma yapmaktadır. Bu gruplandırma ile tüm nesne grubu bir noktada birleşir ve kaydırıcı bunu dağıtma işlemine izin vermektedir. Değer üzerinde bir K değeri seçildiğinde bu değere ait olan hesaplamaları ve “K” değerini otomatik olarak sunma işlemi bu araç ile sağlanmaktadır. Şekil 3.8.’de PlotSize aracının K değeri hesaplamasına ilişkin bir görsele yer verilmektedir.



Şekil 3.8. WEKA Programında Görselleştir sekmesi ve PlotSize Aracının K Değeri Tespiti

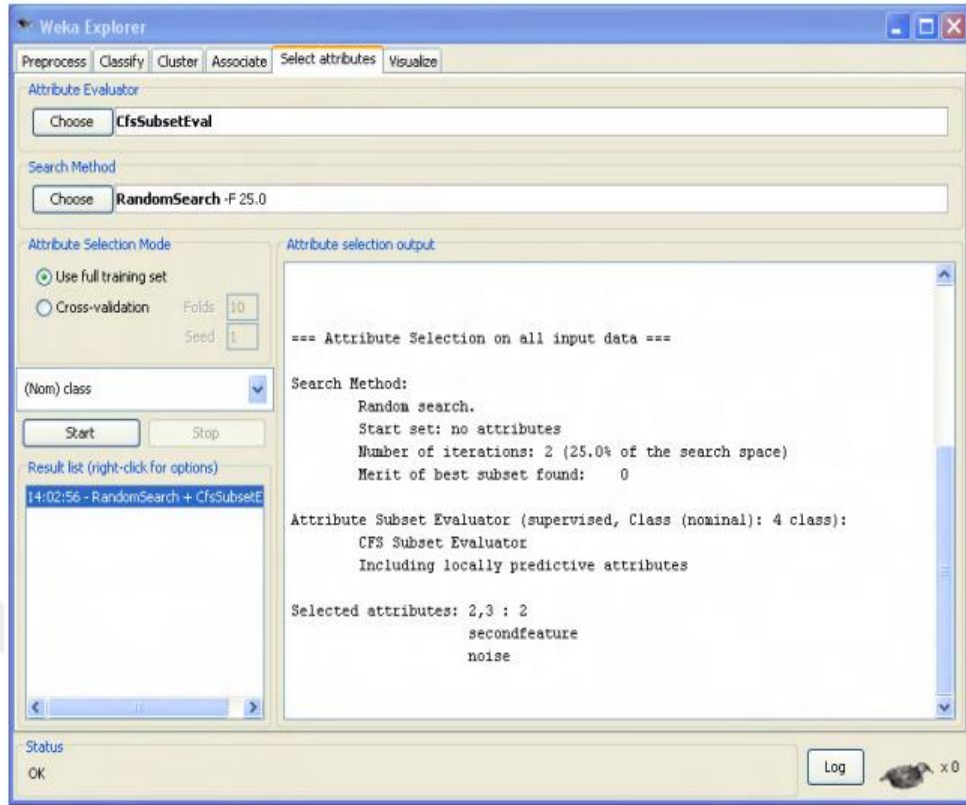
K değeri tespitini takiben herhangi bir nokta seçilimi yapıldığında, çift özelliğine göre kümelenme ve etkilenen sonuçlara ait görsele ise Şekil 3.9.’da yer verilmektedir.



Şekil 3.9. K-NN temelli çift özelliği ve Yakın Değerlerin Görüntülenmesi

WEKA niteliklerin seçilmesinin gerçekleştirildiği sekme, nesnelerin sonraki sınıflandırması veya kümelenmesi için niteliklerin seçilmesine olanak tanımaktadır. Nitelik Değerlendirici panelinde, özelliklerin kalitesini değerlendirme ve Arama Yöntemi panelinde en uygun özellik alanını bulma yöntemi örneğin genetik algoritma, rastgele arama veya kaba kümeleme, yahut “Küresel optimizasyon yöntemleri” ile iyileştirme aramalarının yapılacağı otomatik şablonlar sunmaktadır. Bu şablonlardan herhangi birini seçmek model hatalarının iyileştirilmesi için kullanılmaktadır. Bu panellerle çalışma şeması, bir filtre veya sınıflandırıcı seçme şemasına benzer biçimde, “Nitelik Seçimi Modu” eğitim veri seti ile çalıştırılabilmektedir. Eğitim veri setinde optimizasyon yanı sıra hata giderme veya iyileştirme denilebilen işlem adımları için belli bir seviyede en iyileme sunmaktadır. Model için oluşturulan matriste blok kontrolü kullanarak çapraz doğrulama gibi opsiyonlarla modelin iyileştirilmesini sağlamaktadır.

Nitelikler seçme için kullanılan modülün bir görseline Şekil 3.10.’da yer verilmektedir.



Şekil 3.10.WEKA programı Nitelikler sekmesi ekran görüntüsü

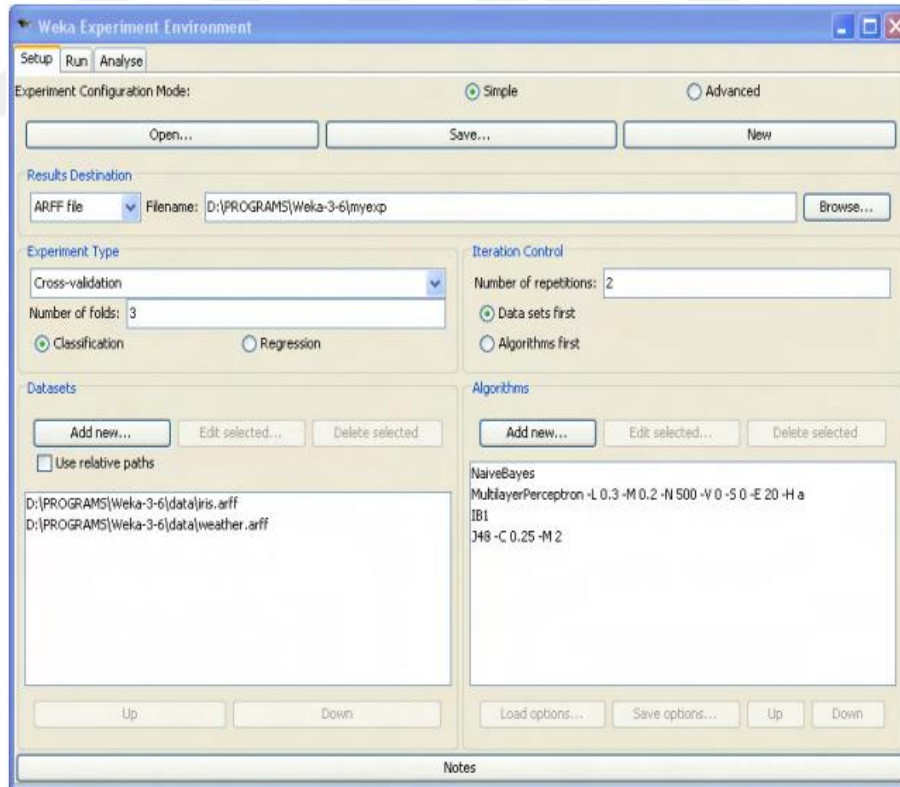
Nitelik özelliklerinin seçim prosedürlerinin raporu bu sekmeden görüntülenebilmektedir

WEKA’da yer alan sekmelerle “Sınıflandırma ve İlişkilendirme” işlemleri, Kümeleme ve ilişki arama sekmeleri üzerinde çalışma imkanı sunmaktadır. Sınıflandırma ile çalışmaya benzer bir biçimde, ilişkilendirme işlemleri de regresyon ve konfigürasyonlarla sağlanması yolları izlenmektedir.

WEKA’da ilişkilendirme işlemleri için, mevcut algoritmaları seçerek çalıştırma imkanı sunan bir arayüze sahip durumdadır. Bu arayüz, Weka ilk panelinde seçilecek olan “Experimenter” modülüdür. Yine bu arayüzde bir veri dosyası için çalışılacaksa veri dosyasının bir kural olarak arff uzantısına uyumlu hale getirilmesini gerekli kılmaktadır. Dosya üzerine yazmak yerine, uyumlu veri dosyası biçimleri listenebilmektedir. Deneme amaçlı demo panellerinde yer alan yeni deney tipinin belirlenmesi seçimiyle, veri dosyasının bölünmesi ve eğitim, deney alanı ve hedefler

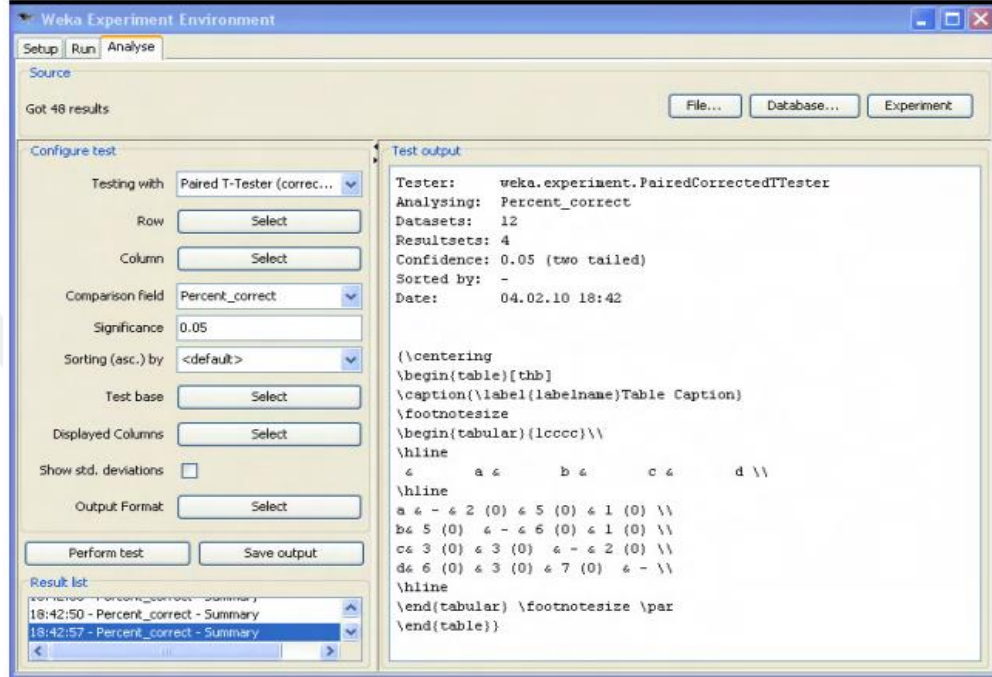
olarak bölümlenerek sınıflandırılması sağlanmaktadır. Bu sınıflandırma kısmında anlatıldığı üzere gerçekleştirildikten sonra, deney tipi seçilmektedir.

Bu panelde, bloklar ve kıvrımlar tarafından kontrol yöntemiyle test edilmesiyle doğrulama işlemi yapılmaktadır. Doğrulama işlemi bir şekilde tekrarlayan YSA ağına benzerlik göstermektedir. Bu panel “Yinelemeli Kontrol” olarak da tanımlanmaktadır. Yineleme sayısı, deneylerin tekrar sayısı ile eşit durumdadır. Tekrarlamalar yeni öğrenme ve kontrol bölünmeleriyle kıvrımlar oluşturmaktadır. Öğrenme, deneyin sayısı ile bu panelde belirtilerek işaretlenmektedir. Bu bölümde kontrol ve doğrulamayla önce tüm problemler ve tüm algoritmalar birlikte gözden geçirilmektedir. Algoritmalar, bu panelde seçilen görevler doğrultusunda belirlenmektedir. Panelde seçebilecek veri kümeleri arasında yer alan algoritmaların atanan görevle belirlenmesi ve yapılacak işlem parametrelerine ilişkin görsele Şekil 3.11.’de yer verilmektedir.



Şekil 3.11. WEKA Programında 2 Veriseti, 4 Algoritma, 3 parçalık verisetleri ve 2 Yinelemeli Kontrol Seçimini Gösterir Ekran Görüntüsü

Seçilimlerin gerçekleştirilmesi ve doğrulamanın hemen takibinde, “Analiz Sekmesi” çalıştırılmaktadır. Veri doğrulanmasının ardından hata çıktısı ve analizleri sergileyen ekran TeX formatında bir çıktı sunmaktadır. Bu formatı gösterir ekran görüntüsüne Şekil 3.12.’de yer verilmektedir.



Şekil 3.12. Experimenter modülünün Analiz sekmesi (deneyler hakkında TeX raporu görüntüsü).

Analiz sekmesi, tüm deneylerin sonuçlarını analiz etmeyi sağlayan sekme olarak bilinmektedir. Yani sadece deneysel modelin doğrulamasını sağlamakla kalmayıp, tüm çalışma çıktılarının da alınacağı sekme anlamına gelmektedir.

Analiz sekmesi, satır düğmesi, görüntülenen matrisin satırları tarafından ne yazılacağı ve nereye varılmak istendiğine ilişkin bilgileri alarak, uygun seçimleri yaparak, çıktıların sunulduğu sekmedir ve atanan “Veri Kümesi” üzerinde doğrulamadan sonra, “Çalıştır”, “Katla” gibi komutlarla veri kümesi sütunlarında sütunlara göre ne kaydedilecek ise bir şema seçimine izin vererek, “Karşılaştırma” alanında, tablonun neyle doldurulacağının belirlenmesi konusunda kullanıcıyı yönlendirmektedir. Örneğin bu sırada çıktılar yüzdelik dilim olarak verilmesi isteniyorsa “Yüzde_Doğru” seçimi yapılmakta ve doğru sınıflandırmanın, modelin öğrenmesi ve testi sırasındaki tüm çıktıların ilgili sütunları tanıyarak yüzdelilerle doldurulmasını sağlamaktadır. Şekil

3.13.'te çıktı formatına ilişkin Latex seçiminde rapor görünümüne ilişkin görsele yer verilmektedir.

Tester:	weka.experiment.PairedCorrectedTTester			
Analysing:	Percent_correct			
Datasets:	12			
Resultsets:	4			
Confidence:	0.05 (two tailed)			
Sorted by:	-			
Date:	04.02.10 18:24			
Dataset	(1) bayes.Nai	(2) funct	(3) lazy.I	(4) trees
iris 1 1	(1) 98.00	98.00	96.00	96.00
iris 1 2	(1) 92.00	92.00	92.00	92.00
iris 1 3	(1) 94.00	92.00	96.00	92.00
iris 2 1	(1) 96.00	98.00	96.00	88.00
iris 2 2	(1) 96.00	98.00	96.00	98.00
iris 2 3	(1) 96.00	92.00	94.00	96.00
weather 1 1	(1) 60.00	60.00	100.00	60.00
weather 1 2	(1) 100.00	80.00	80.00	60.00
weather 1 3	(1) 75.00	75.00	100.00	75.00
weather 2 1	(1) 80.00	60.00	100.00	60.00
weather 2 2	(1) 80.00	60.00	100.00	60.00
weather 2 3	(1) 50.00	50.00	50.00	50.00
Average	84.75	79.58	91.67	77.25
	(v/ /*)	(0/12/0)	(0/12/0)	(0/12/0)
Key:				
(1) bayes.NaiveBayes				
(2) functions.MultilayerPerceptron				
(3) lazy.IB1				
(4) trees.J48				

Şekil 3.13. Veri Setindeki Test Grubunun Rapor Çıktısı

Çok bilindik bir tahmin yöntemi olarak, örneğin meteoroloji verilerinin WEKA'ya işlenmesiyle gelecek durum tahmin şablonu oluşturulmak isteniyorsa, WEKA yine sadece dört adet algoritmayı çalıştıracaktır. Bu algoritmalarla, önce veriyi tanıtmamızı isteyecek, bunu karar ağacı algoritmasından geçirerek, Naive Bayes sınıflandırıcısını harekete geçirecek, buradan bir şablon oluşturmak üzere çok katmanlı perceptronu çalıştıracak ve son olarak K-NN yani en yakın komşu algoritmasını kullanarak her gruptan tek bir ortalama değeri çıktı olarak sunacaktır.

Şekil 3.13.'de yer aldığı üzere 0/12/0 olarak okunan tek çıktı, tüm durumların ortalamasını sağlayacaktır. Şu halde, WEKA sadece ortalama değeri tek çıktı ile sunan bir parametrik harmonik yaklaşımla temellendirilmiş bir programı ifade etmektedir. Bu da WEKA'nın aylık veya haftalık tahmin ortalamasını yapması için ayrı ayrı çıktılar yerine yaklaşık ortalama değeri çıktı olarak sunduğunu bilmemizi gerekli kılmaktadır.

Son olarak, WEKA programının “KnowledgeFlow” modülü olarak adlandırılan seçenek penceresi, bir problemi çözme sürecinin tamamını bir grafik şeklinde temsil edilmesine izin veren WEKA aracını tanımlamaktadır.

Bu modülde, çözümün aşamaları köşe matris biçiminde belirtilmektedir. Burada WEKA programının çalışması ve daha büyük bir yığınla öğrenmesi için yeni değerler üretilebilmektedir. Burada bulunan CLI modülü, komutları girmek için bir metin satırı arabirimi sağlamaktadır. Komut satırını kullanmak, kullanıcıya diğer modüllerde bulunmayan özellikleri vermesinin yanı sıra, bellekten tasarruf edilmesini de sağlamaktadır. İlave komut satırlarını kullanmak da WEKA’nın tek çıktı sunma sorununu ortadan kaldırmamakta, ancak daha iyi süzölmüş bir modelin tasarlanmasına imkan tanıma avantajını getirmektedir. Şekil 3.14.’te WEKA’nın bilgi yığını süzme ve eğitim için sınıflandırma düzenine yer verilmektedir.

```
> java weka.classifiers.trees.J48 -t data\weather.arff -p 0
```

=== Predictions under cross-validation ===

inst#	actual	predicted	error	prediction
1	2:no	1:yes	+	0.75
2	1:yes	2:no	+	0.75
1	2:no	1:yes	+	0.75
2	1:yes	1:yes		1
1	2:no	2:no		1
2	1:yes	1:yes		1
1	2:no	1:yes	+	1
2	1:yes	1:yes		1
1	2:no	2:no		1
1	1:yes	2:no	+	0.75
1	1:yes	1:yes		1
1	1:yes	1:yes		1

Şekil 3.14. WEKA Bilgi Yığını Tarama ve Sınıflandırması

Şekil 3.14.’te de görüldüğü üzere ortalama değerler değişmeksizin daha fazla satır sunmakta ancak gruplandırma sabit kalmaktadır.

Özet bir CLI modülünde yer alan sınıflandırıcıların özeti ise şu şekilde karşımıza çıkmaktadır: Karar Ağacı Algoritması, NaiveBayes sınıflandırıcısı, K-NN sayısal özellikler için parametrik olmayan yoğunluk yeniden yapılandırma işleminin uygulanması, Destek Vektör Makinesi (SVM) yöntemi (Burada, WEKA’nın

varsayılan araçlarında yer alan SVM, doğrusal bir çekirdek ile, 10 faktörlü bir derece 5 polinom çekirdeği için E-5 ve C-10 düzenini otomatik sunmaktadır) ve yeniden K-NN algoritmasını yürütmektedir. K-NN algoritması,

- t; eğitim örneği içeren arff formatındaki dosyayı,
- T; test örneği içeren ve atanmış bir parametre yoksa bloklarla kontrol yapmayı varsayan olarak 10 bloklu matrisi yürüterek test işlemi yürütücüsünü, ki bu test yürütücüsü, 1x1 matris akış düzenini 10 kez tekrar eden bir mantık kurgusuna dayalı olarak çalışmaktadır.
- x; Matris bloklarının kat sayısını,
- c; hedef özelliği sınıf belirtecini,
- d; algoritma türü ve tüm eğitim setinde yapılandırılan parametrelerle modeli kaydeden işlevi,
- l; model şemasının çıkarılması işlevini,
- p; sınıflandırmayı ve bir test dosyası belirtilmişse ve “?” işareti gibi bilinmeyen değerler için bir işaret belirlenmişse burada belirtilen işaret sayısını,
- i; sınıflandırma kalitesinin ayrıntılı bir açıklamasını,
- out; modelle ilgili bilgilerin görüntülenmesinin devre dışı bırakıldığı anı,
- Xmx1024m (1GB için) Çağrı başına 1 “harcanan bellek” alanını

ifade etmektedir.

K-NN’nin geri çağırılması için yazılacak kod; bu sıralama düzenini takip eden ifadeyi beraberinde getirmektedir. Şekil 3.15.’te bu örnek K-NN çağırmasına ait kod üretimine yer verilmektedir.

```
java -Xmx1024m weka.classifiers.trees.J48 -t data.arff -i -k \
-d J48-data.model >&! J48-data.out &
```

Şekil 3.15. WEKA’da üretilebilecek örnek K-NN geri çağırma kodu

Şekil 3.15.’te de görüleceği üzere WEKA kendi takip ettiği düzende kod üretimine izin veren bir paket program düzenine sahip bir ortamı sunmaktadır.

WEKA'daki takip düzeninde, yığın arttırılarak, karar ağacı sınıflandırıcısı başlatılmakta ve tüm sınıflar (-i ve -k) için kesinlik ve geri çağırma değerleri görüntülenebilmektedir. Modelin kaydedilmiş hali (-d) yazılarak çağırılabilir. Ters eğik çizgi "\" işareti ile ifade edilen ise satırın aramayı tamamladığı yani verilen komutun yerine getirildiğini göstermektedir.

Weka GUI “Seçici pencere” menüsünde yer alan diğer araçlar ve anlamları ise şu şekilde tanımlanmaktadır; Araçlar sekmesi bir arff dosyası “ArffViewer” ile düzenlenmektedir. SqlViewer veritabanı görüntüleyicisi işlevini üstlenmektedir. Bayes ağları oluşturmak ve eğitmek için “Bayes net düzenleyicisi” kullanılmaktadır. Görselleştirme sekmesinde yer alan araçlar için, “Plot” çalıştırma görevini yürütmektedir. Kaydedilen ROC eğrilerinin görselleştirilmesi “ROC” komutu ile sağlanmaktadır. “TreeVisualizer” karar ağaçlarının görselleştirilmesi için kullanılmaktadır. XML BIF veya DOT uzantılı çıktılarda grafiklerinin görselleştirilmesi ise “GraphVisualizer” komutu ile sağlanmaktadır.

3.2. MATLAB Analiz Programı

MatLab, The MathWorks, Inc. Şirketi tarafından üretilen bir yazılım ürünüdür. [MathWorks], mühendislik, bilimsel ve uygulamalı hesaplamaların yanı sıra sonuçlarının görselleştirilmesi ve analizine yönelik bir yapıya sahip olmasıyla bilinmektedir. MatLab, güçlü bir C dili ile geliştirilmiş, sayısal yöntemlerin uygulanmasına yönelik işlevsel kabul edilen C++ ve FORTRAN'dan daha uyumlu bir yapıya sahip olan bir dinamik ortam sağlayıcı işlevselliğini sergilemektedir. MatLab ortamı, birçok kütüphane ile çalışmasının yanı sıra, makine öğrenimi problemlerini, sinyal ve görüntü işlemeyi vb. çözmek için ideal araçlar dizinini barındırmaktadır.

Matlab pogramı, kullanıcıya verileri indirmenin ve önceden işlemenin mümkün olduğu bir dizi araç sunmaktadır. Matlab WEKA'ya nazaran daha karmaşık bir yapıya ve daha çok örüntü dizini tanımaya yönelik işlem sağlamaktadır. MatLab, matris laboratuvarı anlamına gelmektedir. Dolayısıyla 1x1 matris ortalamalarında daha çok

örüntü bloğunu tanımaktadır. Buradaki tüm hesaplamalar matris olarak kabul edilmekte, bir anlamda, yalnızca yararlı veri türü ayrımını sağlamaktadır.

Bu da Matlab'a aynı veriyi arttırarak yüklenmesinin algılanmasına ve yığın veriyi temizlemesine yol açmaktadır. MatLab zaman serisi, değer akışı, dinamik modelleme gibi birçok araçla farklı işlemleri gerçekleştirmek üzere kullanılmakla birlikte Simülink ile gerçek zamanlı tanımlamaları da simüle edebilen bir yapı olarak karşımıza çıkmaktadır. Dolayısıyla MatLab amaçları doğrultusunda, farklı kullanım alanları ve kütüphaneleriyle, temel araçlarının inceleme konusuna göre güncellenebilmesiyle öne çıkan karmaşık yapısı yerine ancak işlem adımlarıyla ortaya konmasını gerektirmektedir. Çünkü araçları, dinamiklerine ve seçilen parametrelerine göre değişiklik göstermektedir. Bu sebeple MatLab kullanımı Zaman Tahmin Serisi parametreleriyle Veri işleme bölümünde ayrıntılarıyla anlatılması planlamasını beraberinde getirmektedir.

MatLab programlamanın temelleri, "doğru kodlama" ilkeleri, standart işlevlerin etkin kullanımına dayanmaktadır. Bu çalışmaya özel olarak seçilen parametreler ve tahminleme çıktılarının oluşturulmasına yönelik atılan adımlara bu çalışmanın 3.6' da yer alan ana metninde yer verilmektedir.

3.3. Veri Setinin Oluşturulması

Veri analizi için 50 aylık satış rakamları gerçek bir amortisör firmasından elde edilmiştir. Covid dönemi tahmini için veriler MATLAB'de doğrudan alınırken WEKA'da iki kat kadar büyütülerek eğrinin düzleştirilmesi prensibine uyulup 3 aylık satışların tahmin değerleri WEKA ve Matlab ile gerçekleştirilmiştir. 50 aylık gerçek satış rakamları bir excel tablosuna kaydedilmiştir.

Tahmin işlemi için kullanılacak WEKA programı tahmin parametreleri bu biçimde kullanırken, bu çalışma kapsamında WEKA ile elde edilen tahmin değerinin MATLAB ile elde edilen tahmin değerleri ile karşılaştırılmasına yönelik bir yöntem izlenmesi söz konusu olmaktadır. WEKA çıktılarının MATLAB çıktıları ile

karşılaştırılması, bir ölçüye kadar ilişkili bulunan ve temelinde YSA'nın katman sayılarına göre ve yürüttükleri işlemlere göre hangi programın daha başarılı olabileceği merakıyla şekillenmektedir.

3.4. Veri Setinin WEKA'ya İşlenmesi

Bu çalışmada Weka Version 3.8.5 sürümü kullanılmıştır. Excel'e aktarılan satış adetleri ile bu adetleri etkileyeceği düşünülen 14 adet girdi (bağımsız değişken) çoğaltılarak 50 aylık veri 150 aylığa çıkarılmıştır. Ardından excel önce .csv formatına; sonrasında Şekil 3.16.'da görüleceği üzere Notepad'de tanımlamaları yapılarak .arff formatına (Weka'da işlenebilir veri formatı) dönüştürülmüştür. Problemden bağımlı değişken tahmini yapılan amortisör satışları olurken; bağımsız değişkenler ilgili amortisörün kullanıldığı araç grubu olan hafif ticari araçlarının satış adedi, motorin fiyatı, döviz kuru, aylık dış ticaret, taşıt kredileri toplamı, bankalarca açılan kredilere uygulanan ağırlıklı ortalama faiz oranları, GSYİH, Sanayi Üretim Endeksi, TÜFE, Yi-ÜFE, Tüketici Güven Endeksi, Reel Kesim Güven Endeksi, Ekonomik Güven Endeksi ve hane halkı maddi durum beklentisi verilerinden oluşmaktadır.

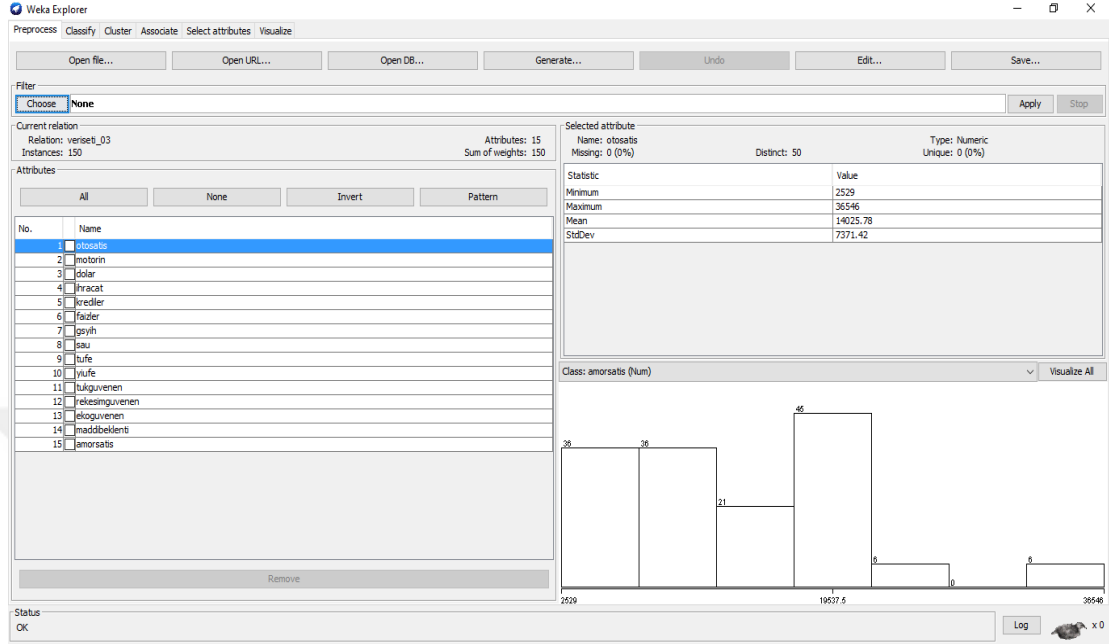
```
@attributes_03.arff - Notepad
File Edit Format View Help
@relation veriseti_03

@attribute otosatis numeric
@attribute motorin real
@attribute dolar real
@attribute ihracat real
@attribute krediler numeric
@attribute faizler real
@attribute gsyih real
@attribute sau real
@attribute tufe real
@attribute yiupe real
@attribute tukguvenen real
@attribute rekesimguvenen real
@attribute ekoguvenen real
@attribute maddibeklenti real
@attribute amorsatis numeric

@data
9355, 3.762842581, 3.00909, 10170052.05, 6356, 15.822, 499315, 102.7642103, 274.44, 250.67, 92.23526304, 106.5, 100.7636748, 92.16780138, 89714
12237, 3.383816207, 2.94067619, 12356684.82, 6302, 16.315, 499315, 104.0694061, 274.38, 250.16, 90.04438591, 105.1, 97.76280445, 89.31293074, 98858
18973, 3.513942903, 2.885013043, 12762853.96, 6320, 16.3325, 499315, 102.9291586, 274.27, 251.17, 90.083096, 104.6, 99.71953468, 89.76912934, 104264
19269, 3.536373667, 2.833619048, 12374495.04, 6358, 15.916, 525932, 102.6546117, 276.42, 252.47, 91.31028861, 105.3, 98.44262391, 89.97106029, 93209
20072, 3.730303226, 2.933742857, 12473674.41, 6403, 15.5425, 525932, 104.7876372, 278.02, 256.21, 91.52763129, 105.100, 2277802, 89.94075971, 85615
20429, 3.81824, 2.913954545, 12963716.94, 6435, 15.3, 525932, 104.0259644, 279.33, 257.27, 91.63303526, 104.1, 100.5093791, 91.20881292, 91225
12967, 3.752598065, 2.969358824, 11991572.92, 6340, 15.308, 655400, 97.49182766, 282.58, 257.81, 88.90190725, 105.7, 102.9744292, 88.35545166, 83459
17579, 3.707142581, 2.960222727, 11769830.53, 6340, 15.4475, 655400, 105.0056903, 281.76, 258.01, 94.71843367, 101.7, 98.20740018, 94.19836216, 79500
16253, 3.957527333, 2.962655556, 11788034.17, 6304, 15.008, 655400, 102.0552077, 282.27, 258.77, 94.80558329, 106.6, 102.6300996, 92.06592099, 76189
19254, 4.177003871, 3.07145, 13710399.74, 6296, 14.735, 734393, 106.3237192, 286.33, 260.94, 95.29323885, 104.9, 101.1331331, 94.09409363, 94681
26526, 4.160811667, 3.281768182, 12884161.9, 6502, 13.7125, 734393, 105.273589, 287.81, 266.16, 91.56271824, 108.2, 102.1196237, 89.01402554, 92500
33868, 4.418639032, 3.493554545, 13201866.47, 6719, 13.864, 734393, 104.839411, 292.54, 274.09, 87.37332055, 103.9, 97.44038061, 85.40220661, 98708
9634, 4.628841935, 3.744354545, 11561022.22, 6630, 15.21, 641584, 105.7438058, 299.74, 284.99, 88.72077054, 100.6, 94.04992794, 87.86493178, 93221
12307, 4.63584, 3.66338, 12502550.83, 6558, 15.6175, 641584, 107.0736631, 302.17, 288.59, 88.03296822, 106.2, 98.86087881, 85.94818642, 94158
18186, 4.520100645, 3.668208696, 14354430.49, 6541, 15.382, 641584, 108.6851313, 305.24, 291.58, 90.10925194, 106.5, 101.7665029, 87.76594697, 109494
17990, 4.561650333, 3.650095, 13768705.76, 6536, 15.68, 734211, 111.8565972, 309.23, 293.79, 92.01931419, 106.3, 103.2477129, 90.73987929, 103294
19623, 4.453890323, 3.560380952, 13549743.33, 6560, 15.6125, 734211, 110.3423539, 310.61, 295.31, 93.99177382, 104.8, 104.0883101, 91.98471481, 109967
```

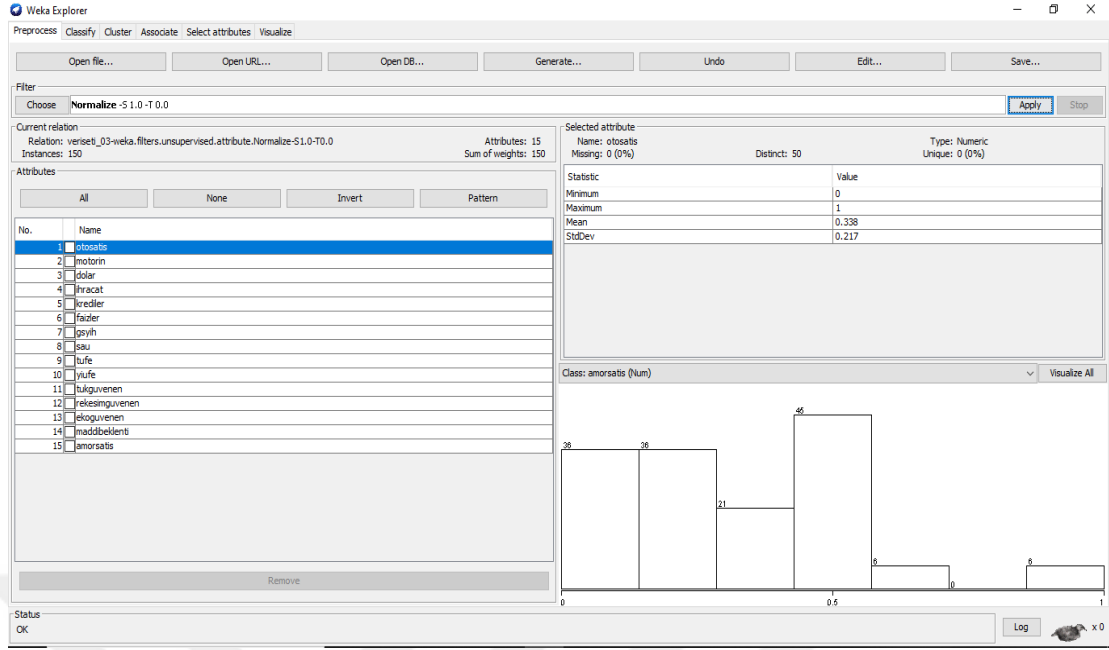
Şekil 3.16. Veri Setinin .arff formatına dönüştürülmesi

Weka'nın Explorer ekranından ilgili veriler programa yüklenmiştir ve Şekil 3.17.'deki Preprocess sekmesinde çalışılmaya başlanmıştır.



Şekil 3.17. Verisetinin Weka Preprocess sekmesinde açılması

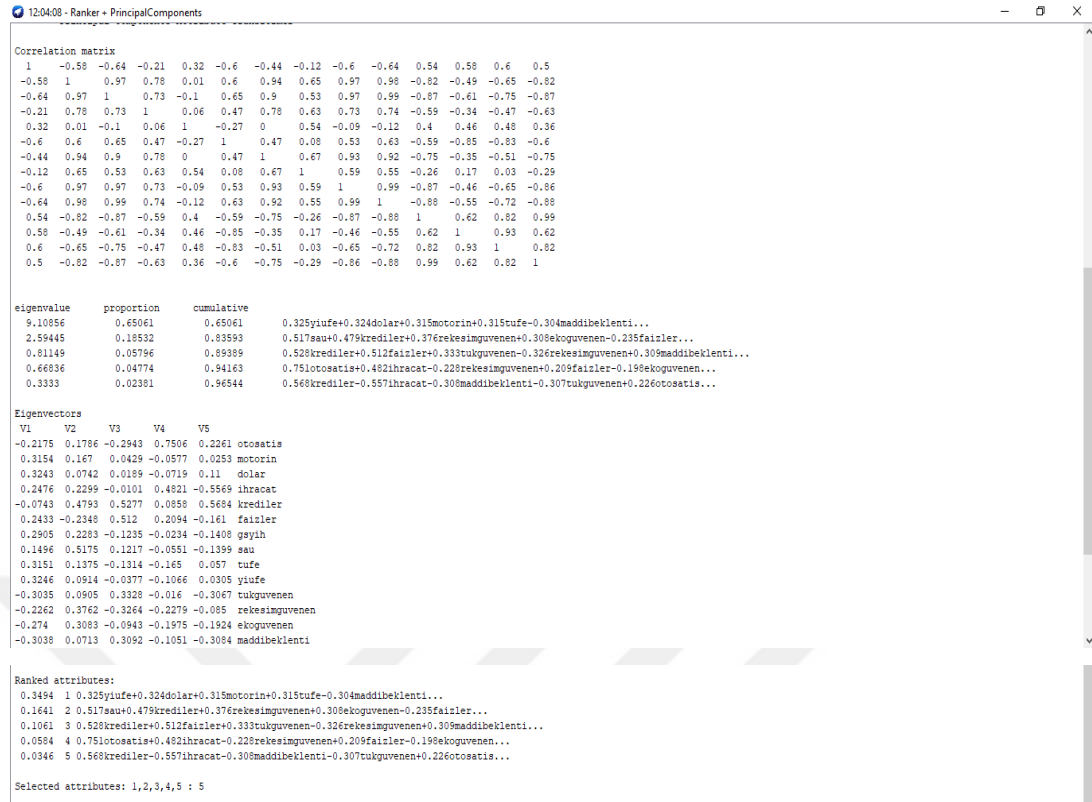
Preprocess sekmesinde verilerin üzerinde filtreleme ekranı sayesinde çalışmak mümkündür. Çalışmada Filters'tan Unsupervised-Attribute-Normalize filtresi seçilerek amortisör satışı (tahmini yapılacak bağımlı değişken) hariç girdiler (bağımsız değişkenler) normalize edilmiştir. Daha önce bahsedilen selected attributes alanında verilerin minimum ve maksimumu buna göre değişmiştir (Şekil 3.18.).



Şekil 3.18. Normalize edilmiş veriseti

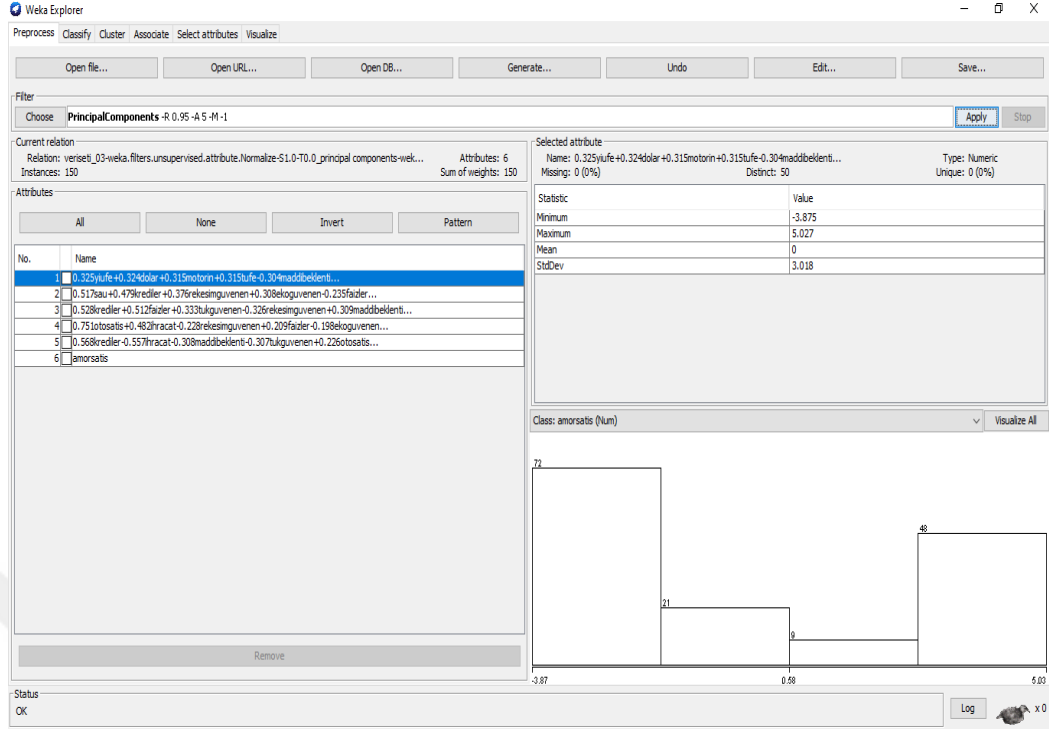
Select attributes sekmesinde bağımsız değişkenlerin arasındaki ilişki incelenmiştir. Attribute Evaluator'dan Temel Bileşenler Analizi yani Principal Components filtresi uygulanarak ilgili korelasyona bakılmıştır. Bu korelasyona bakarken amortisör satışları tahmini sınıfına göre tüm veriler hem eğitim hem test seti olarak seçilmiştir ve Şekil 3.19.'daki matris elde edilmiştir.

Bu anlamda Temel Bileşenler Analizi'ne kısaca değinmek gerekir ki, Temel bileşen analizi (PCA), temel bileşenleri hesaplama ve bunları, bazen yalnızca ilk birkaç temel bileşeni kullanarak ve gerisini göz ardı ederek veriler üzerinde bir temel değişikliği gerçekleştirmek için kullanma sürecidir. PCA, keşifsel veri analizinde ve tahmine dayalı modeller yapmak için kullanılır. Genellikle, veri varyasyonunun mümkün olduğu kadar çoğunu korurken daha düşük boyutlu veriler elde etmek için her veri noktasını yalnızca ilk birkaç temel bileşene yansıtarak boyutluluğu azaltmak için kullanılır.

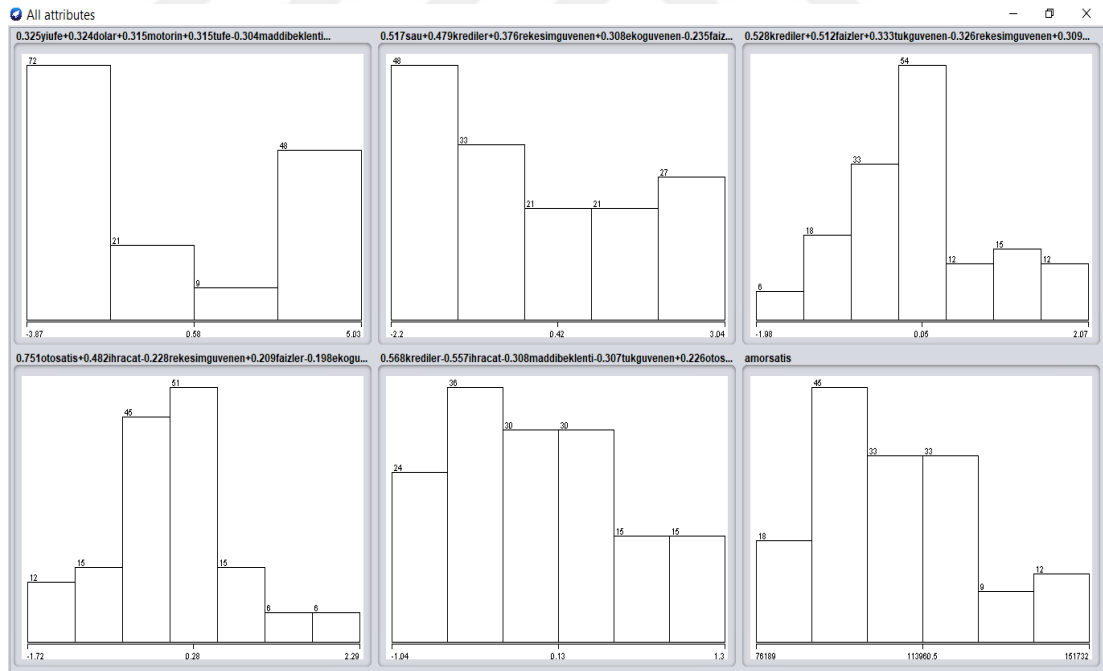


Şekil 3.19. PCA ekran görüntüsü

Şekil 3.19.'daki vektör matrisi ve Ranked Attributes sonucundan görüleceği üzere problemdeki 14 adet bağımsız değişken aralarındaki korelasyona göre incelenmiş ve 5 adet değişkene indirgenmiştir. Bu 5 adet değişken ise tahmini yapılacak amortisör satış adetleri üzerinde toplam %71,26 oranında etkili olduğu gözlemlenmiştir. Bu araştırmayı burada yaptıktan sonra veriyi buna uygun olacak şekilde işlemek için tekrar Preprocess sekmesinden filter olarak Princial Components seçilerek girdiler Şekil 3.20.'deki gibi indirgenmiştir ve Şekil 3.21.'deki gibi görselleştirilmiştir;



Şekil 3.20. Preprocess sekmesinde PCA filtresinin uygulanması

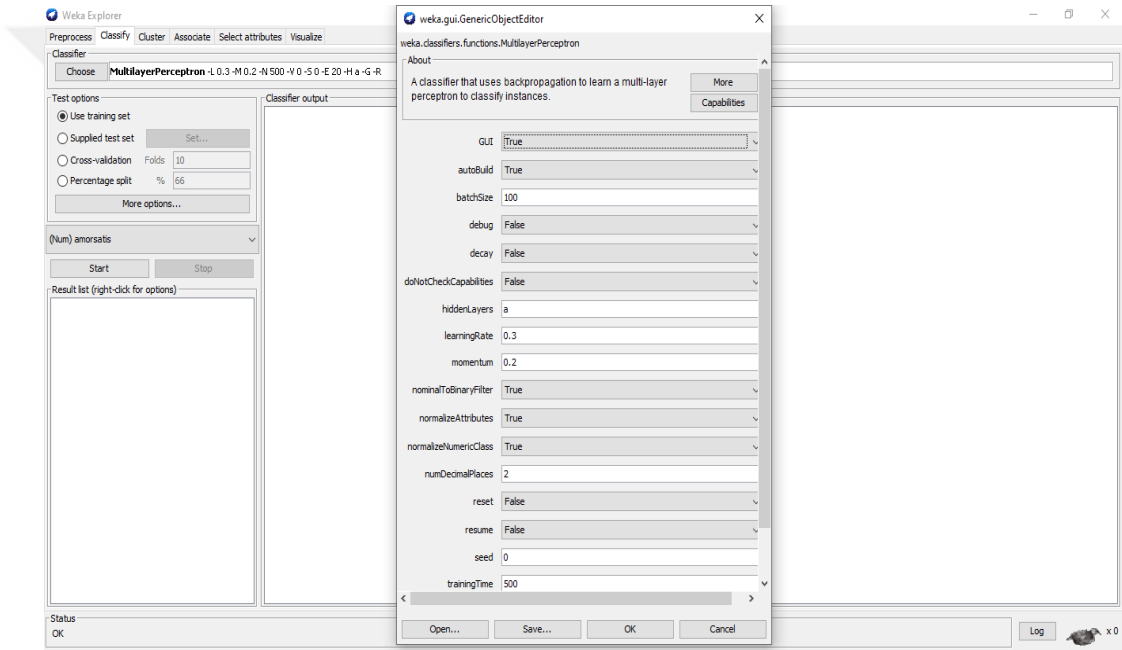


Şekil 3.21. Görselleştirme ekran görüntüsü

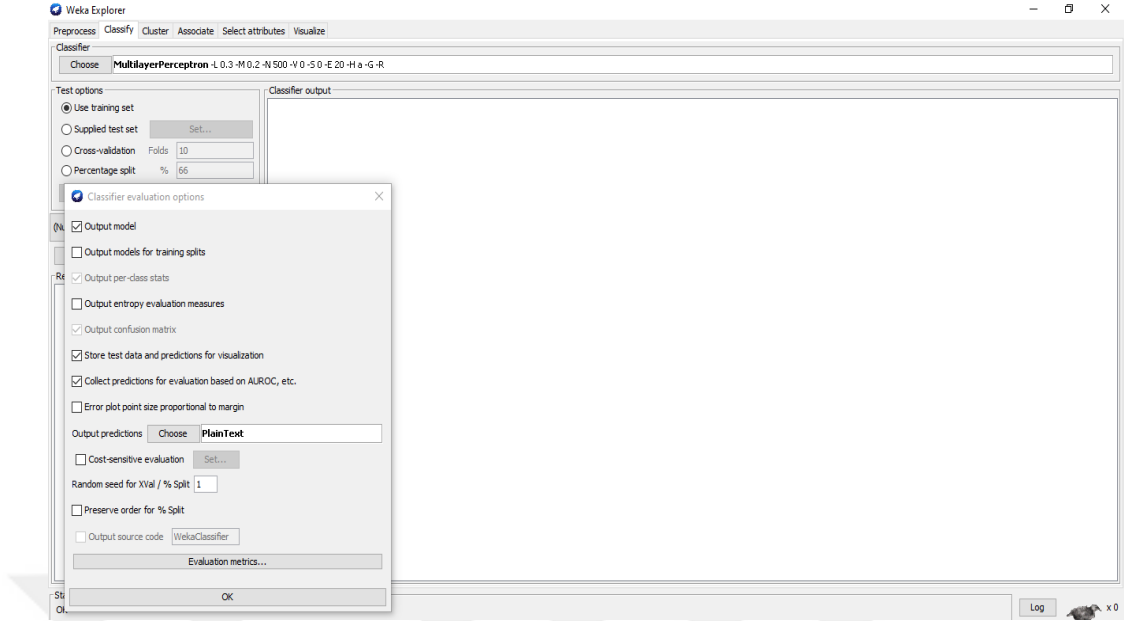
Bağımsız değişkenler indirgendiği için artık YSA'ya başlanacaktır. Bu çalışmada önce çoğaltılarak 150 aylık olan satış adetlerinin ilgili 5 bağımsız değişken ile en iyi tahmin sonucunu veren YSA kurularak kaydedilmiştir. Ardından bu başarılı sonucu veren

YSA üzerine tahmini yapılmak istenen Mart, Nisan ve Mayıs 2020, erken pandemi dönemi ayları “supplied test set” ile YSA’ya tanıtılarak ilgili aylar için YSA kurulan modelin üzerinde tekrar hesaplatılacak ve tahmin sonucu bulunacaktır.

Öncelikle Classify’den Classifier olarak Functions altındaki Multilayer Perceptron seçilerek YSA çalıştırılmıştır. (Daha önceki başlıklarda da bahsedildiği gibi YSA’da manuel olarak değişiklik yapabilmek ve görseline ulaşmak için Şekil 3.22.’deki gibi GUI; True yapılmıştır.) Ayrıca tahmin sonuçlarının ekrana yansımaları için Plaintext olarak yazdır komutu verilmiştir (Şekil 3.23.).

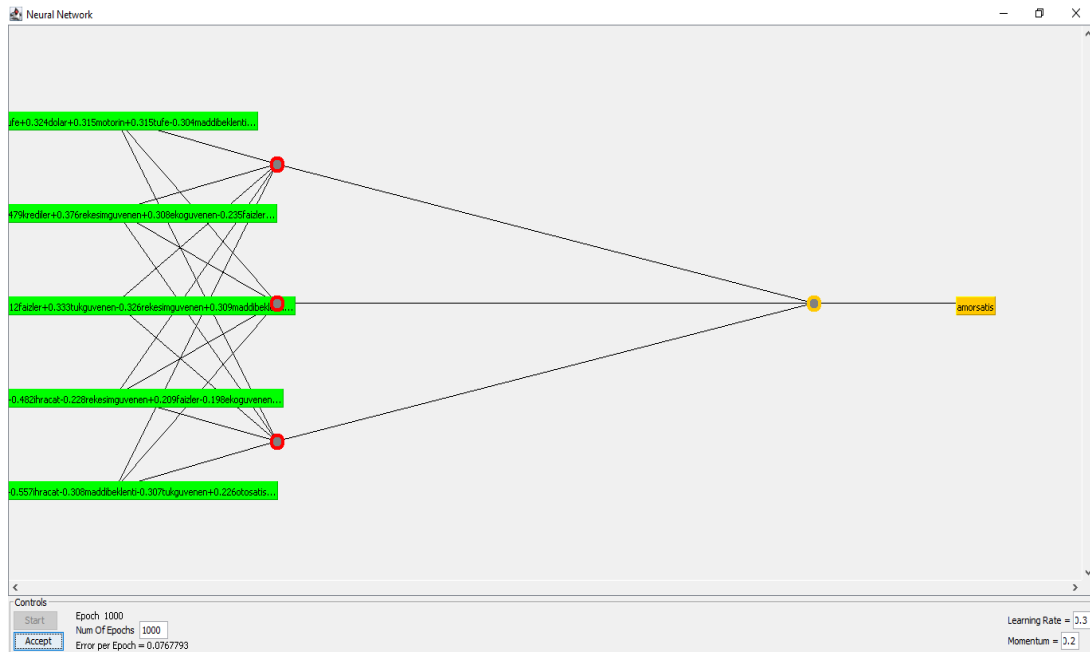


Şekil 3.22. YSA modeline ait ayarlar



Şekil 3.23. YSA tahmin sonuçlarının yazdırılması komutu

En iyi sonuca ulaşmak için YSA’larda epok sayısı, eğitim/test setinin Percentage Split ile %66 / %70 / %80 olarak ayrılması, yine eğitim/test setinin Cross Validation ile 10 Folds iken denemeler yapılmış ve Şekil 3.24.’teki YSA ile learning rate (öğrenme oranı) 0,3; momentum 0,2; epok 1000, eğitim/test setinin Percentage Split %66 iken en iyi YSA modeli kurulmuştur.



Şekil 3.24. Kurulan YSA modeli ekran görüntüsü

Buna göre kurulan YSA’da elde edilen tahmin sonuçları, gerçekleşen değerler ile farklarını yansıtan “error” (hataları) Şekil 3.25.’teki gibi olmuştur;

```

21:41:32 - functions.MultilayerPerceptron
=== Run information ===

Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R
Relation: veriseti_05-weka.filters.unsupervised.attribute.Normalize-SL.0-70.0_principal components-weka.filters.unsupervised.attribute.PrincipalComponents-R0.95-A5-M-1
Instances: 150
Attributes: 6
  0.325yiufo+0.324dolar+0.315motorin+0.315tufe-0.304maddibeklenti...
  0.517sau+0.479krediler+0.376rekesinguvnen+0.308ekoguvnen+0.235faisler...
  0.528krediler+0.512faisler+0.333tukuvnen+0.326rekesinguvnen+0.309maddibeklenti...
  0.751otocatis+0.482ihracat+0.228rekesinguvnen+0.209faisler+0.198ekoguvnen...
  0.568krediler+0.557ihracat+0.308maddibeklenti+0.307tukuvnen+0.226otocatis...
  amorasatis
Test mode: split 66.0% train, remainder test

=== Classifier model (full training set) ===

Linear Node 0
  Inputs Weights
  Threshold 0.06677287910437778
  Node 1 -3.7475011656622805
  Node 2 -0.6257109127361339
  Node 3 1.598459493378378

Sigmoid Node 1
  Inputs Weights
  Threshold -5.172105213529338
  Attrib 0.325yiufo+0.324dolar+0.315motorin+0.315tufe-0.304maddibeklenti... -0.7101021444334986
  Attrib 0.517sau+0.479krediler+0.376rekesinguvnen+0.308ekoguvnen+0.235faisler... 4.637307738745063
  Attrib 0.528krediler+0.512faisler+0.333tukuvnen+0.326rekesinguvnen+0.309maddibeklenti... -2.5046256600481542
  Attrib 0.751otocatis+0.482ihracat+0.228rekesinguvnen+0.209faisler+0.198ekoguvnen... -0.7417859973451473
  Attrib 0.568krediler+0.557ihracat+0.308maddibeklenti+0.307tukuvnen+0.226otocatis... -0.9919770849235739

Sigmoid Node 2
  Inputs Weights
  Threshold -3.519308248894166
  Attrib 0.325yiufo+0.324dolar+0.315motorin+0.315tufe-0.304maddibeklenti... -3.2973341043909214
  Attrib 0.517sau+0.479krediler+0.376rekesinguvnen+0.308ekoguvnen+0.235faisler... -5.27818001486644
  Attrib 0.528krediler+0.512faisler+0.333tukuvnen+0.326rekesinguvnen+0.309maddibeklenti... -7.44337256413083
  Attrib 0.751otocatis+0.482ihracat+0.228rekesinguvnen+0.209faisler+0.198ekoguvnen... -2.7942775549549481
  Attrib 0.568krediler+0.557ihracat+0.308maddibeklenti+0.307tukuvnen+0.226otocatis... 1.3746663060465885

Sigmoid Node 3
  Inputs Weights
  Threshold -7.53738813557357
  Attrib 0.325yiufo+0.324dolar+0.315motorin+0.315tufe-0.304maddibeklenti... 1.566208349868404
  Attrib 0.517sau+0.479krediler+0.376rekesinguvnen+0.308ekoguvnen+0.235faisler... 5.337589302475958
  Attrib 0.528krediler+0.512faisler+0.333tukuvnen+0.326rekesinguvnen+0.309maddibeklenti... -14.974724530067162
  Attrib 0.751otocatis+0.482ihracat+0.228rekesinguvnen+0.209faisler+0.198ekoguvnen... 5.124676066103494
  Attrib 0.568krediler+0.557ihracat+0.308maddibeklenti+0.307tukuvnen+0.226otocatis... -4.4339599595811726

Class
  Input
  Node 0

Time taken to build model: 4.66 seconds

=== Predictions on test split ===

inst# actual predicted error
1 98624 115471.758 16847.758
2 139200 143666.745 4466.745
3 103150 99648.06 -3501.14
4 98708 115247.27 16539.27
5 116327 116266.549 -60.451
6 103341 117487.931 14146.931
7 94158 111800.351 17642.351
8 104264 105421 1157
9 116815 116259.026 -555.974
10 121117 132051.2 10934.2
11 112396 110574.938 -1821.062
12 127533 123075.4 -4457.6
13 94681 99055.625 4374.625
14 116815 116259.026 -555.974
15 125894 121004.11 -4879.89
16 120126 135381.025 15255.025
17 141073 144345.574 3272.574
18 145680 143289.825 -2390.175
19 76895 79014.074 2119.074
20 114219 116370.28 2151.28
21 117471 119678.63 2207.63

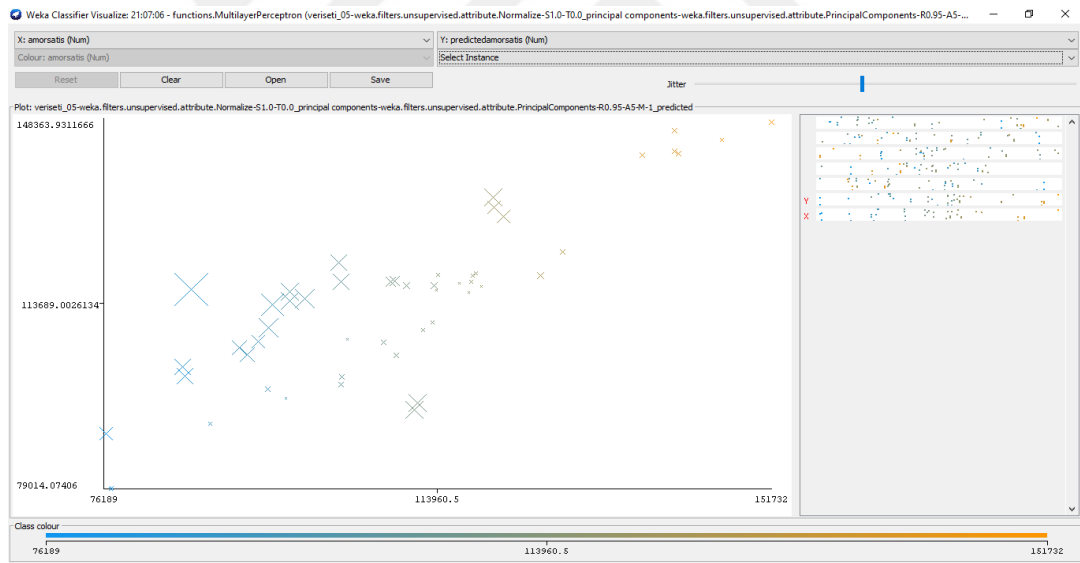
```

Şekil 3.25. YSA tahmin sonuçları ve hataları

22	76895	79014.074	2119.074
23	103150	99648.86	-3501.14
24	118100	119770.658	1670.658
25	109494	118235.243	8741.243
26	139200	143666.745	4466.745
27	92500	105607.343	13107.343
28	94940	96097.061	1157.061
29	109494	118235.243	8741.243
30	93221	113329.742	20106.742
31	112396	110574.938	-1821.062
32	107578	111144.742	3566.742
33	117471	119678.63	2207.63
34	85615	100413.154	14798.154
35	92500	105607.343	13107.343
36	86070	116413.573	30343.573
37	76189	87455.604	11266.604
38	93209	104549.09	11340.09
39	151732	148363.931	-3368.069
40	98708	115247.27	16539.27
41	103341	117487.931	14146.931
42	85615	100413.154	14798.154
43	121117	132051.2	10934.2
44	110767	116245.707	5478.707
45	109967	106417.377	-3549.623
46	110227	93976.619	-16250.381
47	119929	118992.016	-936.984
48	89714	92070.378	2356.378
49	110767	116245.707	5478.707
50	110227	93976.619	-16250.381
51	141073	144345.574	3272.574

Şekil 3.265. YSA tahmin sonuçları ve hataları (Devam)

Weka'daki bu YSA modeline ait hataları görselleştirmek için Visualize classifier errors ile Şekil 3.26.'daki görsel elde edilmiştir;

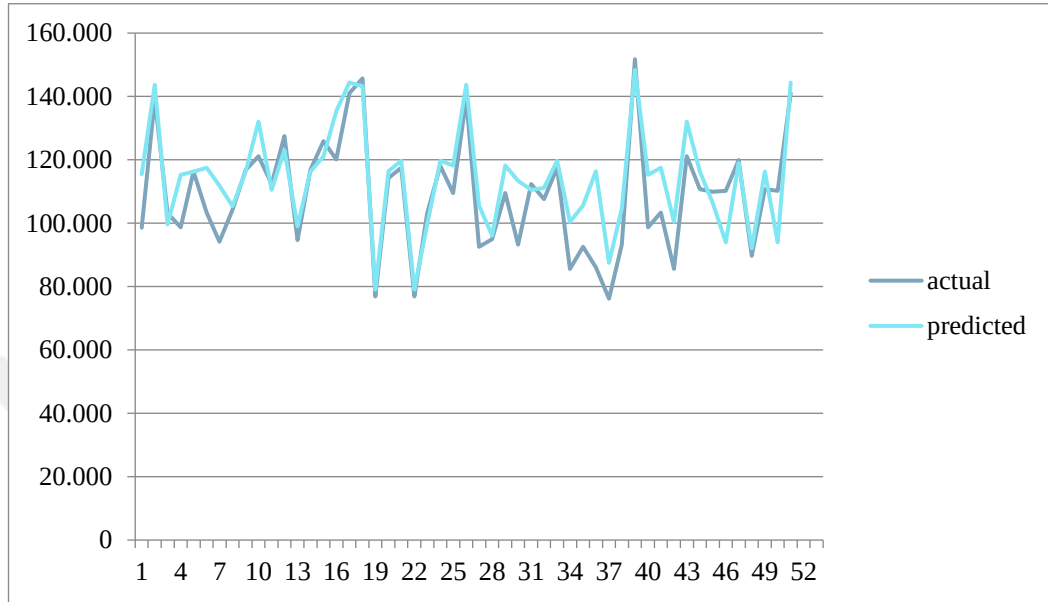


Şekil 3.26. YSA tahminlerine ait hatanın görselleştirilmesi

Correlation coefficient	0.8684
Mean absolute error	7740.4222
Root mean squared error	10243.1682
Relative absolute error	54.4714 %
Root relative squared error	57.2384 %
Total Number of Instances	51

Şekil 3.27. Kurulan YSA'ya ait korelasyon katsayısı ve hata oranları

Şekil 3.27.'de de görüldüğü gibi YSA sonucunda korelasyon katsayısı %86,6 iken tahminler yapılmış ve hata oranları belirtilmiştir. Buna göre YSA'nın yaptığı tahminler ile gerçekleşen satışlara ait görsel sonuç Şekil 3.28.'de belirtildiği gibi olmuştur.



Şekil 3.28. Aylık satış tahminleri ile gerçekleşen satışlar tablosu

Kurulan bu YSA modeli .model uzantısı ile Weka'da tekrar çağrılabilir şekilde kaydedilmiştir. Çalışmanın bundan sonrasında tahmini yapılmak istenen 2020 yılı pandemi dönemine ait Mart, Nisan ve Mayıs aylarına ait veri setlerinin Weka'ya yüklenip YSA ile tahmin edilmesi aşamasına geçilmiştir.

Bu maksatla ilk önce ilgili 3 aya ait 14 girdiden ve «?» belirsiz satış adetinden oluşan yeni bir test seti .arff dökümanı Şekil 3.29.'daki gibi oluşturulmuştur.

```
*Adsız - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
@relation veriseti_05_pandemi_2

@attribute otosatis numeric
@attribute motorin real
@attribute dolar real
@attribute ihracat real
@attribute krediler numeric
@attribute faizler real
@attribute gsyih real
@attribute sau real
@attribute tufe real
@attribute yiufe real
@attribute tukguvenen real
@attribute rekesimguvenen real
@attribute ekoguvenen real
@attribute maddibeklenti real
@attribute amorsatis numeric

@data
10121,5.677613548,6.330054545,13188396.08,7080,13.35,1071098,113.0113041,450.58,468.69,8111938243,98.6,93.12529561,78.58403041,?
4632,5.132681,6.838109524,8636027.98,7000,13.91,1041643,76.47981408,454.43,474.69,78.14239289,62.3,52.39652788,72.48901342,?
7162,5.25394129,6.942647059,11065738.93,7147,13.12,1041643,92.86566002,460.62,482.02,82.66690665,73.5,62.73441563,79.24337877,?
```

Şekil 3.28. Tahmini yapılacak pandemi 3 aya ait verilerin .arff formatına dönüştürülmesi

Yukarıda anlatılan ve .model uzantısı ile kaydedilen YSA modeli Şekil 3.30.'daki gibi Weka'ya yüklenmiştir.

```
=== Model information ===

Filename:      result from last5 for make predictions.model
Scheme:        weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 1000 -V 0 -S 0 -E 20 -H a -G -R
Relation:       veriseti_05-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0 principal components-weka.filters.unsupervised.attribute.PrincipalComponents-R0.95-A5-M-1
Attributes:     6
               0.325yiufe+0.324dolar+0.315motorin+0.315tufe-0.304maddibeklenti...
               0.517sau+0.479krediler+0.376rekesimguvenen+0.308ekoguvenen-0.235faizler...
               0.528krediler+0.512faizler+0.333tukguvenen-0.326rekesimguvenen+0.309maddibeklenti...
               0.751otosatis+0.482ihracat-0.228rekesimguvenen+0.209faizler-0.198ekoguvenen...
               0.568krediler-0.557ihracat-0.308maddibeklenti-0.307tukguvenen+0.226otosatis...
               amorsatis

=== Classifier model ===

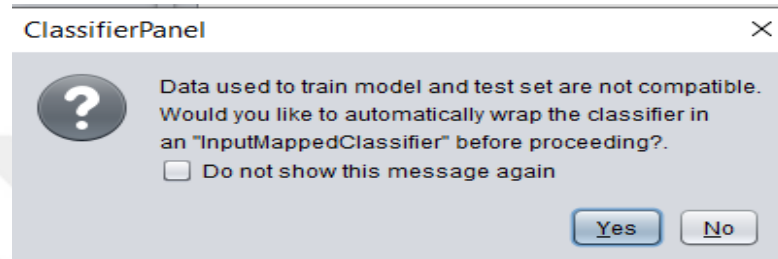
Linear Node 0
  Inputs  Weights
  Threshold 0.08021199109578689
  Node 1 -4.315502919057884
  Node 2 -0.6277746823124817
  Node 3 1.6516675482395273
Sigmoid Node 1
  Inputs  Weights
  Threshold -5.169440206211096
  Attrib 0.325yiufe+0.324dolar+0.315motorin+0.315tufe-0.304maddibeklenti... -0.7453534490850259
  Attrib 0.517sau+0.479krediler+0.376rekesimguvenen+0.308ekoguvenen-0.235faizler... 4.3995865540189
  Attrib 0.528krediler+0.512faizler+0.333tukguvenen-0.326rekesimguvenen+0.309maddibeklenti... -2.4935771480343187
  Attrib 0.751otosatis+0.482ihracat-0.228rekesimguvenen+0.209faizler-0.198ekoguvenen... -0.6908067517457525
  Attrib 0.568krediler-0.557ihracat-0.308maddibeklenti-0.307tukguvenen+0.226otosatis... -0.9536204619722345
Sigmoid Node 2
  Inputs  Weights
  Threshold -4.442709538364706
  Attrib 0.325yiufe+0.324dolar+0.315motorin+0.315tufe-0.304maddibeklenti... -3.8002507528552463
  Attrib 0.517sau+0.479krediler+0.376rekesimguvenen+0.308ekoguvenen-0.235faizler... -6.633731379069671
  Attrib 0.528krediler+0.512faizler+0.333tukguvenen-0.326rekesimguvenen+0.309maddibeklenti... -9.317845177924228
  Attrib 0.751otosatis+0.482ihracat-0.228rekesimguvenen+0.209faizler-0.198ekoguvenen... -3.6576011083929813
  Attrib 0.568krediler-0.557ihracat-0.308maddibeklenti-0.307tukguvenen+0.226otosatis... 1.8069056449370096
Sigmoid Node 3
  Inputs  Weights
  Threshold -8.553479972103919
  Attrib 0.325yiufe+0.324dolar+0.315motorin+0.315tufe-0.304maddibeklenti... 1.8861121513558976
  Attrib 0.517sau+0.479krediler+0.376rekesimguvenen+0.308ekoguvenen-0.235faizler... 5.815817656229351
  Attrib 0.528krediler+0.512faizler+0.333tukguvenen-0.326rekesimguvenen+0.309maddibeklenti... -17.125140831310148
  Attrib 0.751otosatis+0.482ihracat-0.228rekesimguvenen+0.209faizler-0.198ekoguvenen... 6.066281012306319
  Attrib 0.568krediler-0.557ihracat-0.308maddibeklenti-0.307tukguvenen+0.226otosatis... -4.813816422066185
Class
  Input
  Node 0
```

Şekil 3.29. Kaydedilen YSA modelinin Weka'ya tekrar çağırılması

Bu YSA modelinin üzerine tahmini yapılmak istenen Mart, Nisan ve Mayıs ayına ait verilerin olduğu .arff formatına dönüştürülmüş veriler Supplied Test Set ile seçilerek yüklenmiştir. 14 bağımsız değişken sayısı, kurulan bu YSA'da PCA ile 5'e düşmüştür.

olduğundan, yeni aylara ait yüklenen test setinde yine aynı başlıklar altında 14 bağımsız değişken olduğundan program uyumsuzluğu yok etmek amacıyla InputMappedClassifier seçeneğini sunmaktadır (Şekil 3.31.).

InputMappedClassifier; Bir sınıflandırıcının oluşturulduğu eğitim verileri ile gelen test örneklerinin yapısı arasında bir eşleme oluşturarak uyumsuz eğitim ve test verilerini ele alan sarmalayıcı sınıflandırıcı olarak tanımlanmaktadır.



Şekil 3.31. InputMappedClassifier seçeneği

Bu işlem basamakları tamamlandığında .model ile kaydedilmiş olan YSA üzerine tahmini yapılmak istenen veriler yüklenmiş ve YSA'yı tekrar hesapla seçeneği ile tahminler Şekil 3.32.'daki gibi gerçekleştirilmiştir.

```

=== Re-evaluation on test set ===

User supplied test set
Relation:      veriseti_05_pandemi_2
Instances:     unknown (yet). Reading incrementally
Attributes:    15

=== Predictions on user test set ===

    inst#    actual  predicted    error
      1         ?  115804.675         ?
      2         ?  115804.675         ?
      3         ?  115804.675         ?

=== Summary ===

Total Number of Instances          0
Ignored Class Unknown Instances    3

```

Şekil 3.3230. YSA tahmin sonucu

3.5. WEKA ile Tahmin Çözümü

WEKA programına 50 aylık satış rakamlarının iki kat büyütülmesiyle elde edilen yığınla ulaşılan tahmin sonuçları, %86,6 hata payı ile sunmaktadır. Bunun nedeni WEKA'nın ortalama tahmin değerlerini K-NN prensibi gereği hesaplamasıdır.

WEKA çalışması sonucunda elde edilen değerlerin gösterimine Tablo 3.1.'de yer verilmektedir.

Tablo 3.1. WEKA YSA tahmin sonucu ile gerçek satış değerlerinin karşılaştırılması

	Weka Tahmini	Gerçekleşen Satış	Fark
Mart	115805	18900	96905
Nisan	115805	22	115783
Mayıs	115805	42159	73646

3.6. Veri Setinin MATLAB'a İşlenmesi

MATLAB bir programlama dili ve nümerik işlem ortamıdır ve komut satırı temelli çalışmaktadır. Ayrıca grafik kullanıcı arayüzünü ve “script” programlamayı da desteklemektedir. Bu yapay sinir ağı eğitimi için bir “script” yazılarak program başlatılmıştır.

MATLAB'da eğitilecek bir yapay sinir ağı için birden fazla yaklaşım izlenilebilmektedir. MATLAB'ın kendisi bir programlama dili olduğu için en baştan tamamen el ile bir yapay sinir ağı yazılabilmesinin yanı sıra, yazılımın arayüzünde bulunan “İstatistik ve Makine Öğrenimi Araç Kutusu” ile gelen hazır fonksiyonlar da kullanılabilmektedir. Bu çalışmada, pratiklik açısından gelişmiş ve esnek bir kullanım sunduğu için hazır fonksiyonların kullanımı tercih edilmiştir.

Bu çalışmada MATLAB R2017a (9.2.0.538062) February 23, 2017 lisans sürümü kullanılmıştır. Uygulamaya başlangıç aşamasında, öncelikle bağımlı ve bağımsız değişkenleri tanıtmak üzere Excel ortamında hazırlanan “.xlsx” uzantılı veri tablosu MATLAB'a tanıtılmıştır ve veriler normalleştirilmiştir. Çalışmaya ait ekran görüntüsüne Şekil 3.33.'te yer verilmektedir.

```

clear
close
clc

%% Data Acquisition and Normalization

data = xlsread('attributes.xlsx');

norm_data = zeros(50, 14);

for i = 1:14
    norm_data(:, i) = (data(:, i) - min(data(:, i)))/( max(data(:, i)) ...
        - min(data(:, i)));
end

```

Şekil 3.33. Verinin Okunması ve Normalleştirilmesi (Ekran Görüntüsü)

Bu işlem için ilk olarak “xlsread” fonksiyonu ile veriler “data” değişkenine aktarılmıştır ve ilk 14 sütunda bulunan bağımsız değişkenler için denklem 3.1’ de yer verilen, normalleştirme formülü kullanılmış ve norm_data değişkenine bu normalleştirilmiş veri tanıtılmıştır.

$$data_{norm} = \frac{data_i - data_{min}}{data_{max} - data_{min}} \quad (3.1)$$

Bu işlemi takiben, normalleştirilen verinin Korelasyon Matrisi hesaplanmıştır. Bu aşamada “Temel Bileşen Analizi” yapılmak üzere, bağımsız değişkenler 14 boyuttan 5 boyuta indirilmiştir; bu çalışmayı gösterir ekran çıktısına Şekil 3.34.’te yer verilmektedir.

```

%% Correlation Matrix and Principal Components Calculation

correlationMatrix = corrcoef(norm_data);

[coeff, PCA_Data, eigenvalues] = pca(norm_data, 'VariableWeights', ...
    'variance', 'NumComponents', 5);

coeff = inv(diag(std(norm_data(:, 1:14))))*coeff;

```

Şekil 3.34. Normalleştirilen Verinin Korelasyon Matris Hesabı ve Temel Bileşen Analizi

Bu işlem adımını takiben hesaplanan korelasyon matrisi “correlationMatrix” değişkeninde saklanmıştır. Temel Bileşen Analizi yapan “pca()” fonksiyonuna ise “analiz yöntemi” ve “bileşen boyutu” argümanları girilmiştir. Fonksiyon çıktısı olarak;

- coeff: Eigen vektörleri
- PCA_Data: 5 boyuta indirilmiş veri
- eigenvalues: Eigen değerleri

verileri değişkenlere sırası ile tanıtılmıştır.

Bu çalışmalara ait ekran görüntülerine Şekil 3.35. ve Şekil 3.36.'da sırası ile yer verilmektedir.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	-0.5787	-0.6355	-0.2108	0.3157	-0.6046	-0.4373	-0.1197	-0.6026	-0.6362	0.5438	0.5773	0.5983	0.4991
2	-0.5787	1	0.9692	0.7827	0.0129	0.6031	0.9359	0.6476	0.9685	0.9769	-0.8151	-0.4894	-0.6537	-0.8203
3	-0.6355	0.9692	1	0.7320	-0.1022	0.6519	0.9041	0.5287	0.9677	0.9859	-0.8744	-0.6053	-0.7466	-0.8728
4	-0.2108	0.7827	0.7320	1	0.0589	0.4719	0.7762	0.6295	0.7310	0.7445	-0.5904	-0.3392	-0.4658	-0.6308
5	0.3157	0.0129	-0.1022	0.0589	1	-0.2667	0.0015	0.5386	-0.0946	-0.1221	0.3987	0.4559	0.4830	0.3597
6	-0.6046	0.6031	0.6519	0.4719	-0.2667	1	0.4696	0.0762	0.5297	0.6345	-0.5868	-0.8534	-0.8318	-0.6034
7	-0.4373	0.9359	0.9041	0.7762	0.0015	0.4696	1	0.6702	0.9311	0.9190	-0.7483	-0.3473	-0.5088	-0.7533
8	-0.1197	0.6476	0.5287	0.6295	0.5386	0.0762	0.6702	1	0.5883	0.5510	-0.2571	0.1677	0.0339	-0.2864
9	-0.6026	0.9685	0.9677	0.7310	-0.0946	0.5297	0.9311	0.5883	1	0.9906	-0.8704	-0.4577	-0.6490	-0.8620
10	-0.6362	0.9769	0.9859	0.7445	-0.1221	0.6345	0.9190	0.5510	0.9906	1	-0.8823	-0.5510	-0.7201	-0.8770
11	0.5438	-0.8151	-0.8744	-0.5904	0.3987	-0.5868	-0.7483	-0.2571	-0.8704	-0.8823	1	0.6246	0.8231	0.9883
12	0.5773	-0.4894	-0.6053	-0.3392	0.4559	-0.8534	-0.3473	0.1677	-0.4577	-0.5510	0.6246	1	0.9316	0.6226
13	0.5983	-0.6537	-0.7466	-0.4658	0.4830	-0.8318	-0.5088	0.0339	-0.6490	-0.7201	0.8231	0.9316	1	0.8184
14	0.4991	-0.8203	-0.8728	-0.6308	0.3597	-0.6034	-0.7533	-0.2864	-0.8620	-0.8770	0.9883	0.6226	0.8184	1

Şekil 3.31. Korelasyon Matrisi

	1
1	9.1086
2	2.5944
3	0.8115
4	0.6684
5	0.3333
6	0.1480
7	0.1384
8	0.1024
9	0.0382
10	0.0238
11	0.0151
12	0.0106
13	0.0070
14	2.7077e-04

	1	2	3	4	5
1	-0.2175	0.1786	-0.2943	0.7506	0.2261
2	0.3154	0.1670	0.0429	-0.0577	0.0253
3	0.3243	0.0742	0.0189	-0.0719	0.1100
4	0.2476	0.2299	-0.0101	0.4821	-0.5569
5	-0.0743	0.4793	0.5277	0.0858	0.5684
6	0.2433	-0.2348	0.5120	0.2094	-0.1610
7	0.2905	0.2283	-0.1235	-0.0234	-0.1408
8	0.1496	0.5175	0.1217	-0.0551	-0.1399
9	0.3151	0.1375	-0.1314	-0.1650	0.0570
10	0.3246	0.0914	-0.0377	-0.1066	0.0305
11	-0.3035	0.0905	0.3328	-0.0160	-0.3067
12	-0.2262	0.3762	-0.3264	-0.2279	-0.0850
13	-0.2740	0.3083	-0.0943	-0.1975	-0.1924
14	-0.3038	0.0713	0.3092	-0.1051	-0.3084

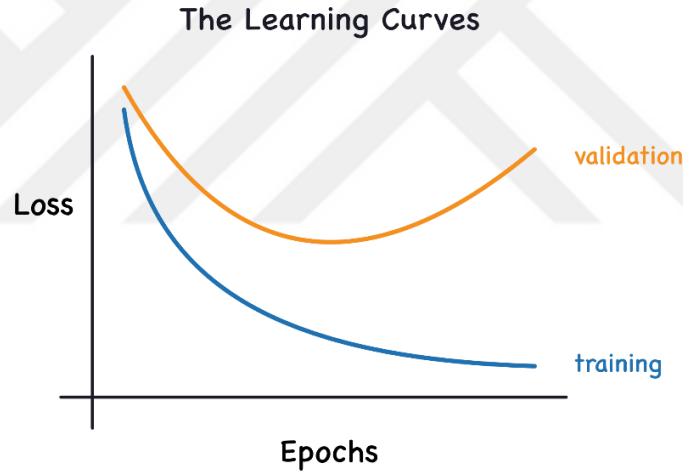
Şekil 3.36. Eigen Değerleri ve Eigen Vektörleri

Tablolardan da görüldüğü üzere değerler, WEKA uygulamasında hesaplanan değerler ile benzerlik sergilemekte ve 1x1 matris değerinin bölümlenmesinde bu değerleri

kesiştirme alanı olarak sunmaktadır. Bu durum, yapay sinir ağını eğitmek için kullanılan PCA verisi her iki uygulama için de aynı olmasından kaynaklanmaktadır.

Tahmin amaçlı eğitilen yapay sinir ağlarında görülen tipik bir problem, eğitilen ağı eğitim verisi ile aşırı uyum, yani “overfitting” sağlamasından ötürü ileriye dönük tahminlerin gerçek değerlerden uzaklaşma sorunudur. Çalışmada şayet karmaşık, yüksek varyanslı çıktıya sahip ve çok değişkenli bir yapay sinir ağı eğitiliyor ise hata oranı düşerken bir süre sonra aşırı uyum gerçekleşecektir.

Aşırı uyumlu bir ağı, eğitim verisi uygulandığında düşük hata verirken eğitim verisinde olmayan, farklı bir girdi uygulandığında hata oranı yüksek olacaktır. Aşırı uyum sonucunu sergileyen bir örnek grafiğe Şekil 3.37.’de yer verilmektedir.



Şekil 3.37. Aşırı Uyum Örneği

Grafikteki örnekten de görüleceği gibi adım sayısı ilerledikçe eğitim verisinin hata oranı düşerken “validation” yani doğrulama verisinin hata oranı artmaya başlamaktadır. Bu artış aşırı uyum sebebi ile gerçekleşmektedir.

Yapay sinir ağının gizli katmanının büyüklüğü de aşırı uyum etkisini artırmaktadır. Çok büyük bir sinir ağı oluşturmaktan bu sebeple kaçınılmıştır. “Makine Öğrenimi Araç Kutusu”nda bulunan yapay sinir ağları dört farklı kategoriye ayrılmaktadır;

- Eğri Uydurma (Curve Fitting)
- Örüntü Tanıma (Pattern Recognition)
- Kümeleme (Clustering)
- Zaman Serisi Analizi (Time Series Analysis)

Bu uygulamada kullanılan yöntem “Eğri Uydurma” yöntemidir. Bu yöntemin kullanıldığı yapay sinir ağları birden fazla bağımsız değişkenden girdi alıp geniş bir varyansta çıktı verebilmektedir. Bu özelliği sebebiyle bu yöntem çalışmanın ruhuna uygun olarak seçilmiştir.

Eğri uydurma yöntemi için yapay sinir ağının kurulumunda gerekli başlangıç parametrelerinin düzenlenmesi çalışmanın üçüncü adımını oluşturmaktadır. Bu düzenlemeleri gösterir ekran görüntüsüne Şekil 3.38.’de yer verilmektedir.

```
%% Initialization Parameters for Neural Network

targetTrue = data(:, 15)';
input = PCA_Data';

hiddenLayerSize = [4 2];
net = fitnet(hiddenLayerSize);

net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 10/100;
net.trainParam.max_fail = 30;
```

Şekil 3.38. Sinir Ağının Başlangıç Değerlerinin ve Parametrelerinin Ayarlanması

Parametrelerin düzenlenmesini takiben, yapay sinir ağının eğitileceği çıktı değerleri “targetTrue” değişkenine atanmıştır. Ardından “PCA_Data” değişkeninde bulunan 5 değişkenli girdilerin transpozu alınıp “input” değişkenine atanmıştır.

Girdilerin transpozunun alınmasının sebebi “PCA_Data”daki bağımsız değişkenlerin sütunlarla ayrılmış olmasıdır. Araç kutusunda bulunan fonksiyonlar değişkenleri satırlarla ayrılmış şekilde okumaktadırlar.

Transpoze işleminin yapay sinir ağını kuran “fitnet()” fonksiyonuna argüman olarak girilmesi gereken gizli katmanların sayısı “hiddenLayerSize” değişkeninde

tanımlanmıştır. Birçok farklı kombinasyonun denenmesinin ardından gizli katman sayısı 2 olarak seçilmiştir. Birinci katmanda 4, ikinci katmanda ise 2 sinir hücresi bulunmaktadır.

Yapay sinir ağları MATLAB’da bir “network” veri tipi olarak saklanabilmektedir. “fitnet()” fonksiyonu ise bu veri tipini kurmaktadır. Bu fonksiyon ile ağ “net” değişkenine atanmıştır. “network” veri tipinin birçok parametresi bulunmaktadır; bu parametreleri gösterir ekran görüntüsüne Şekil 3.39.’da yer verilmektedir.

Field ▲	Value
trainFcn	'trainlm'
trainParam	1x1 struct
performFcn	'mse'
performParam	1x1 struct
derivFcn	'defaultderiv'
divideFcn	'dividerand'
divideMode	'sample'
divideParam	1x1 struct
trainInd	1x40 double
valInd	[4 22 32 36 39]
testInd	[1 5 11 16 38]
stop	'Validation stop.'
num_epochs	50
trainMask	1x1 cell
valMask	1x1 cell
testMask	1x1 cell
best_epoch	20
goal	0
states	1x8 cell
epoch	1x51 double
time	1x51 double
perf	1x51 double
vperf	1x51 double
tperf	1x51 double
mu	1x51 double
gradient	1x51 double
val_fail	1x51 double
best_perf	6.5625e+07
best_vperf	3.3020e+07
best_tperf	3.0793e+07

Şekil 3.39. Network Veri Tipinin Parametreleri

Veri tipinin çoğu parametresi varsayılan değerleri ile iyi sonuç vermektedir. Değiştirilen dört parametre, “trainRatio, valRatio, testRatio ve max_fail” parametreleridir.

MATLAB’da kurulan yapay sinir ağları eğitilirken üç farklı veri girdi olarak uygulanmaktadır. Bu veriler eğitim verisi, doğrulama verisi ve test verisi olarak tanımlanmaktadır.

Eğitim verisi: Yapay sinir ağına girdi olarak uygulanıp çıktısının hatasından ağırlık değerlerinin değiştirilmesini sağlayan, yapay sinir ağını eğitmek için kullanılan veridir.

Doğrulama verisi: Her adımda eğitilen ağın aşırı uyum sağlayıp sağlamadığını kontrol amaçlı girilen veridir. Eğer doğrulama girdisinin sonucu, her bir adımda hedef değerden uzaklaşıyorsa yapay sinir ağı aşırı uyum sağlamıştır.

Test verisi: Bu verinin yapay sinir ağına eğitimi veya doğrulanması ile bir alakası yoktur. Eğitilen ağı farklı girdilere performansını ölçmek amaçlı kullanılan veriler olarak tanımlanmaktadır.

Değiştirilen ilk üç parametre (trainRatio, valRatio ve testRatio), örnek verinin hangi oranda ne için kullanılacağını belirleyen parametrelerdir. Birçok denemenin ardından;

Eğitim verisi = %80 Doğrulama verisi = %10 Test verisi = %10

oranlarının en doğru sonuçları verdiği görülmüştür. MATLAB bağımsız değişkenlerden bu oranlara göre rastgele örneklem seçecektir.

Yapay sinir ağı eğitilirken her bir adımda eğitim verisi ile ayrıca doğrulama verisi ve test verisi de ağı girdilerine uygulanmaktadır. MATLAB en düşük doğrulama hatasının bulunduğu adımı en başarılı adım olarak kaydetmektedir. Bununla birlikte, “max_fail” parametresi, en başarılı doğrulama hatasının bulunduğu adımdan sonra kaç adım daha eğitimin devam etmesine izin verileceğini belirlemek üzere kullanılmaktadır. Yapılan denemelerde eğitilen verinin ilk 100 adımın içerisinde en iyi doğrulama sonucunu verdiği, ilk 100 adımın ardından aşırı uyuma gittiği görülmüştür. Bu sebeple çalışma sırasında, “max_fail” parametresi düşük bir değer olan 30’a ayarlanarak kullanılmıştır.

Yapay sinir ağı, çalışma ortamına tanıtıldıktan ve parametreler belirlendikten sonra geriye bu ağı eğitilmesi kalmıştır. Bir sonraki adımda ağı eğitilip veriler kendi

sınıflarına ayrılmıştır. Bu ayrımı gösteren ekran görüntüsüne Şekil 3.40.'ta yer verilmektedir.

```
%% Training Neural Network and Seperating Validation and Test Data

[net, tr] = train(net, input, targetTrue);

valInput  = input(:, tr.valInd);
testInput = input(:, tr.testInd);
trainInput = input(:, tr.trainInd);

valTarget = net(valInput);
testTarget = net(testInput);
trainTarget = net(trainInput);
totalOutput = net(input);

valTargetTrue = targetTrue(tr.valInd);
testTargetTrue = targetTrue(tr.testInd);
trainTargetTrue = targetTrue(tr.trainInd);

xVal = 1:length(tr.valInd);
xTest = 1:length(tr.testInd);
xTotal = 1:length(targetTrue);
xTrain = 1:length(tr.trainInd);

rmseTested = sqrt(mean(testTarget - testTargetTrue).^2);
rmseTrained = sqrt(mean(trainTarget - trainTargetTrue).^2);
rmseValidated = sqrt(mean(valTarget - valTargetTrue).^2);

errorTrain = trainTargetTrue - trainTarget;
errorVal = valTargetTrue - valTarget;
errorTest = testTargetTrue - testTarget;
errorTotal = [errorTrain errorVal errorTest];
```

Şekil 3.40. Ağın Eğitilmesi ve Verilerin Sınıflarına Ayrılması

Ayrıca ağın eğitilmesi sırasında, ağ performansının analizi için hata değerleri ve kök ortalama kare hatası hesaplamasının yapıldığı da Şekil 3.40.'da yer alan diğer bir bilgi olarak görülmektedir.

Şekil 3.40.'da yer alan ifadeleri kısaca özetlemek gerekirse, “train()”ağ eğitimi sırasında kullanılan fonksiyondur. Ağ, girdileri ve çıktıları argüman olarak alıp, ağı eğitip eğitilen ağ ve parametrelerini çıktı olarak veren bir fonksiyonu tanımlamak üzere kullanılmaktadır. Ağın kendisi “net” değişkeni ile ifade edilirken, eğitilen ağın “tr” değişkeninde, ağın ve eğitim durumunun bilgileri bulunmaktadır.

Ağın eğilmesinin ardından rastgele seçilen doğrulama, eğitim ve test verileri kendi sınıflarına ayrıştırılmıştır. Ayrıştırılan bu verilerin teker teker girdileri, sonuçları,

hedef çıktıları, hataları ve kök ortalama kare hataları kendi sınıflarına tanıtılmıştır. Bu veriler, eğitimin performansını ve hata oranını ölçmek için önemlidir.

MATLAB’da kurulan ağıın diyagram olarak gösterimi mümkündür. Ağ eğitilip, gerekli sınıflandırılmalar yapıldıktan sonra ağıın diyagramı görselleştirilmiştir ve ağ performansının nasıl sonuç verdiğinin anlaşılması için ağ çıktısı ile istenen değer arasındaki fark grafik olarak oluşturulmuştur. Şekil 3.41.’de bu grafiği gösterir ekran çıktısına yer verilmektedir.

```
%% Plotting Network Structure and Error Between Output and Target Data
view(net);

figure('Name', 'Error Between Target and Output Data');

subplot(2, 2, 1);
plot(xTrain, trainTarget, xTrain, trainTargetTrue, 'LineWidth', 1.5);
ylim([min(targetTrue) max(targetTrue)]);
title('Training');
legend('Trained', 'True');

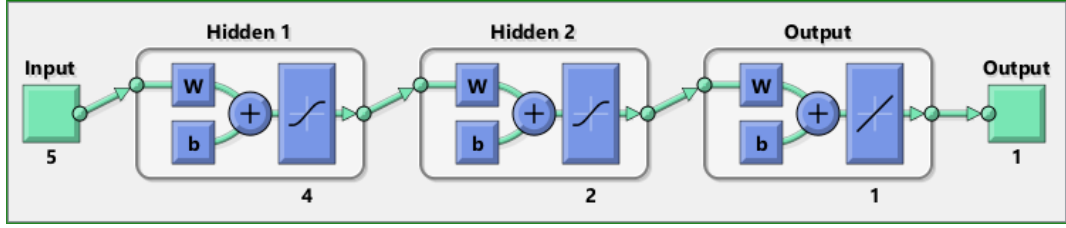
subplot(2, 2, 2);
plot(xVal, valTarget, xVal, valTargetTrue, 'LineWidth', 1.5);
ylim([min(targetTrue) max(targetTrue)]);
title('Validation');
legend('Validated', 'True');

subplot(2, 2, 3);
plot(xTest, testTarget, xTest, testTargetTrue, 'LineWidth', 1.5);
ylim([min(targetTrue) max(targetTrue)]);
title('Testing');
legend('Tested', 'True');

subplot(2, 2, 4);
plot(xTotal, totalOutput, xTotal, targetTrue, 'LineWidth', 1.5);
ylim([min(targetTrue) max(targetTrue)]); xlim([1 50]);
title('Total');
legend('Output', 'Target');
```

Şekil 3.41. Hedef ile Çıktı Arasındaki Farkın Grafiğe Yazdırılması

“view()” fonksiyonu “Makine Öğrenimi Araç Kutusu”na özel, kurulan ağların diyagram olarak görselleştirilmesi için kullanılan bir fonksiyondur. Diyagram ağıın yapısının anlaşılması için iyi bir seçenek olarak tanımlanmaktadır. Şekil 3.42.’de bu çalışma kapsamında oluşturulan diyagrama yer verilmektedir.

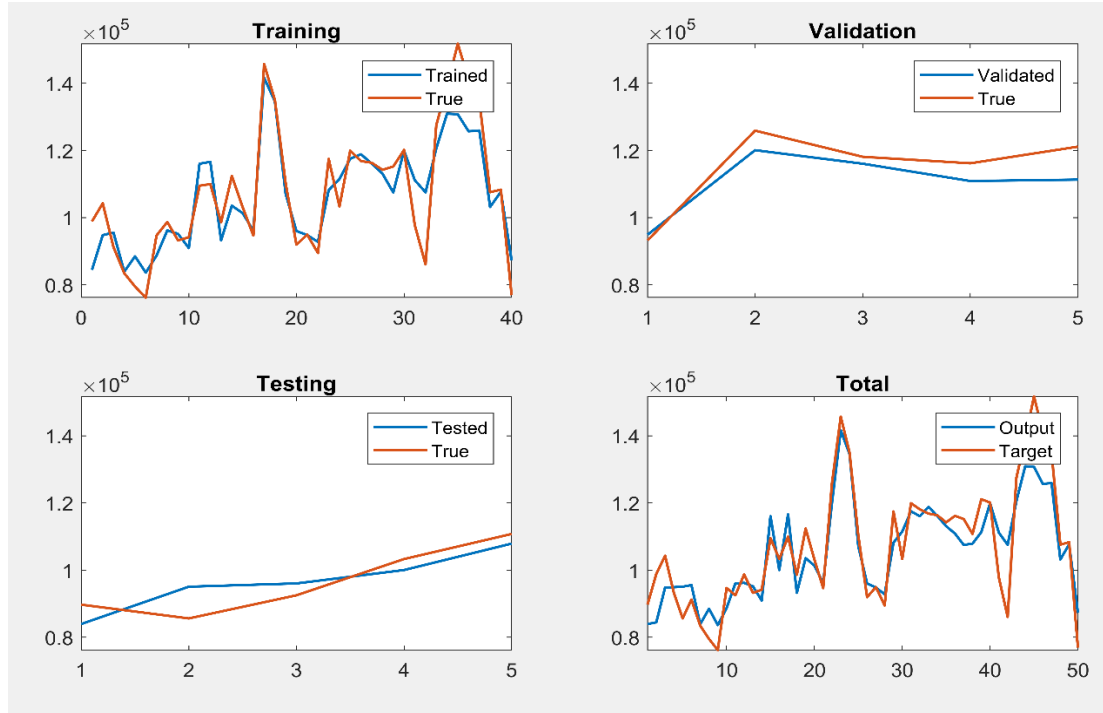


Şekil 3.42. Yapay Sinir Ağı'nın Diyagramı

Diyagramdan görüldüğü üzere bütün ağ parametreleri istendiği gibidir. 5 sinir hücresinin bulunduğu girdi katmanı; Biri 4, diğeri 2 sinir hücresinden oluşan iki adet gizli katman ve sonuç katmanı bulunmaktadır.

MatLab'ta, ağı kurulumunda kullanılan “fitnet()” fonksiyonu varsayılan aktivasyon fonksiyonu olarak sigmoid kullanmaktadır.

Diyagram görselleştirmesinin ardından “plot()” fonksiyonu ile eğitim verisi, doğrulama verisi, test verisi ve bütün verinin istenen değerleri ve çıktıları karşılaştırılmıştır; Şekil 3.43.'te karşılaştırılan çıktıların ekran görüntülerine yer verilmektedir.



Şekil 3.43. İstenen Değerlerin ve Sonuç Değerlerinin Grafikselsel Olarak Karşılaştırılması

Şekil 3.43.'te yer alan mavi çizgi yapay sinir ağının çıktısını, turuncu çizgi ise istenen değeri ifade etmektedir. Bütün grafiklerin y eksenindeki sınırları, istenen değerin minimum ve maksimum değerine göre sınırlandırılmıştır.

Grafiklerden kolaylıkla anlaşılmaktadır ki, mükemmel bir çıktı elde edilmese bile amortisör satışlarının yaklaşık değerinin ve satışların artış veya azalış yaşayacağını tahminini yapmak gayet mümkündür. Yalnızca eğitim verisinin değil, doğrulama ve test verisinin de yakın değerler ve doğru yönde yönelim göstermesi bunun bir kanıtıdır.

Hata grafiklerinin çiziminin ardından yapay sinir ağının performansını gösteren diğer grafiklerin de çizimi yapılmıştır. Çizime dair ekran görüntüsüne Şekil 3.44.'te yer verilmektedir.

```
%% Plotting Performance, Training State, Error Histogram and Regression
figure('Name', 'Performance Data');
plotperform(tr);

figure('Name', 'Training State');
plottrainstate(tr);

figure('Name', 'Error Histogram');
ploterrhist(errorTrain, 'Train Error', errorVal, 'ValidationError',...
            errorTest, 'Test Error');

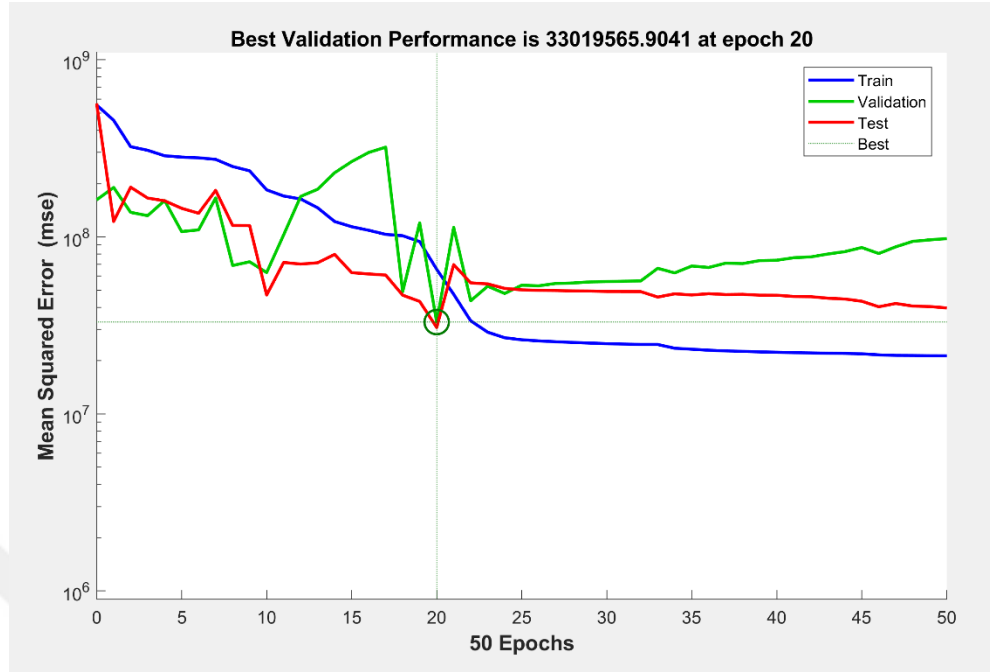
figure('Name', 'Regression Data');
plotregression(trainTargetTrue, trainTarget, 'Training', ...
               valTargetTrue, valTarget, 'Validation', ...
               testTargetTrue, testTarget, 'Test', ...
               targetTrue, totalOutput, 'Total');
```

Şekil 3.44. Performans, Eğitim Durumu, Histogram ve Regresyon Grafiklerinin Çizdirilmesi

Matlab model eğitiminde son olarak araç kutusu ile gelen performans değerlerini gösteren fonksiyonlar kullanılarak,

- Yapay sinir ağının performansı
- Eğitim Durumu
- Hata Histogramı
- Regresyon Değerleri

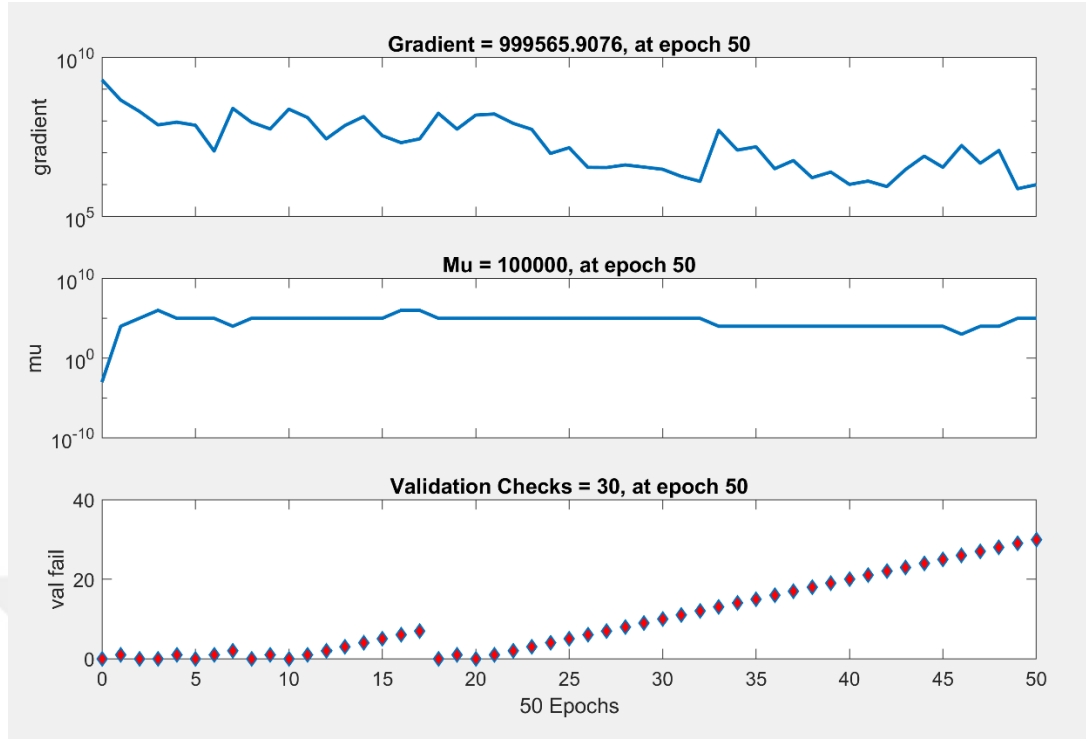
görüntülenmesi sağlanmış olup, bu grafiklere dair ekran görüntülerine Şekil 3.45. ile Şekil 3.48. aralığında yer verilmektedir.



Şekil 3.45. Performans Grafiği

Yapay Sinir Ağı eğitilirken her bir adımda ortalama karesel hata (MSE) hesaplanarak kaydedilmiştir. En iyi değerler hem test verisinin hem de doğrulama verisinin minimum görüldüğü 20. adımdır. Bu adımın ardından eğitim verisinin hatası düşerken aşırı uyum gerçekleşmiştir ve doğrulama verisinin hata değerinin ilerleyen adımlarda yükseldiği görülmektedir. “max_fail” parametresi 30 belirlendiği için en iyi değerin ardından 30 adım daha eğitim gerçekleşmiş ve 50. adımda son bulmuştur.

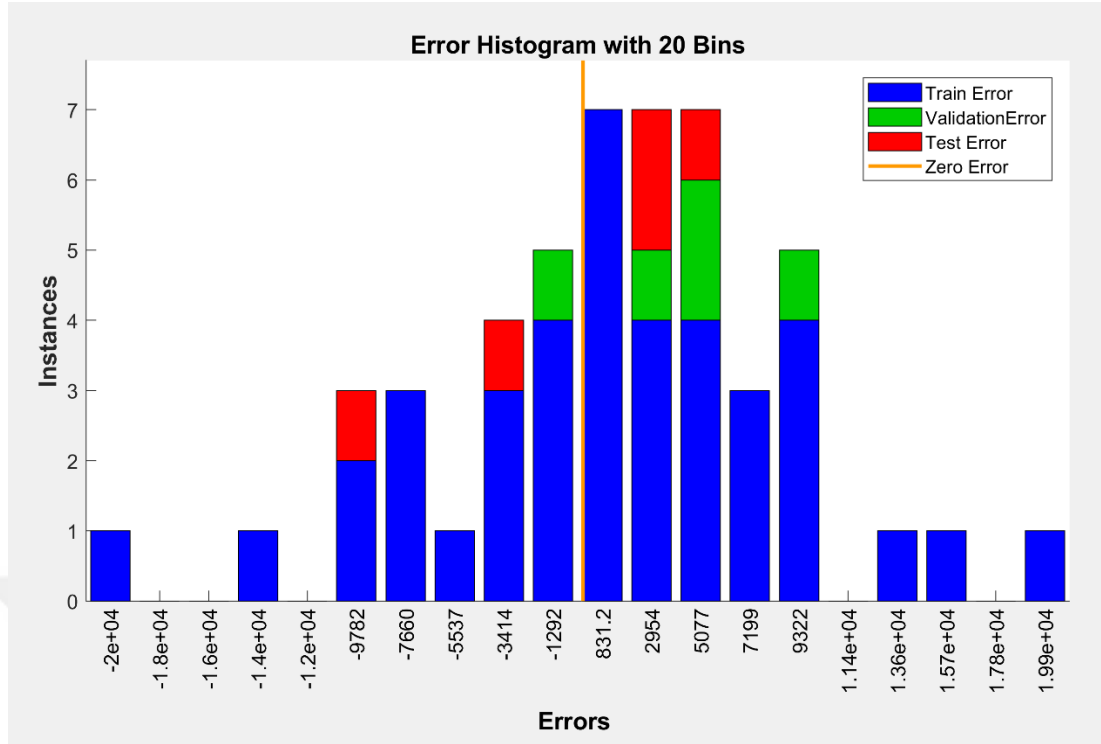
Algoritma yapay sinir ağının ağırlık değerlerini, en iyi doğrulama değerlerinin olduğu adıma göre ayarlamaktadır.



Şekil 3.46. Eğitim Parametrelerinin Son Durumu

- “Gradient” istenen değere göre çıktının toplam hata oranını
- “Mu” öğrenme katsayısını
- “Validation Checks” doğrulama verisinin hatasının kaç adımda arttığını göstermektedir.

MATLAB, varsayılan öğrenme algoritması olarak “Levenberg-Marquardt” algoritmasını kullanmaktadır. “Mu” değerinin yüksek olmasının sebebi de çıktı değerlerinin normalize edilmemesinden ötürüdür.



Şekil 3.47. Hata Histogramı

Hata histogramı, hata değerlerinin sıfıra doğru bir yaklaşımı olduğunu göstermektedir. Tek modlu dağılıma yakın bir hata dağılımı vardır. Sola doğru çarpık bir görünüm sergilemektedir. Toplam veriye dair istatistik Tablo 3.2.'de yer almaktadır.

Tablo 3.2. Toplam Veri İstatistiği

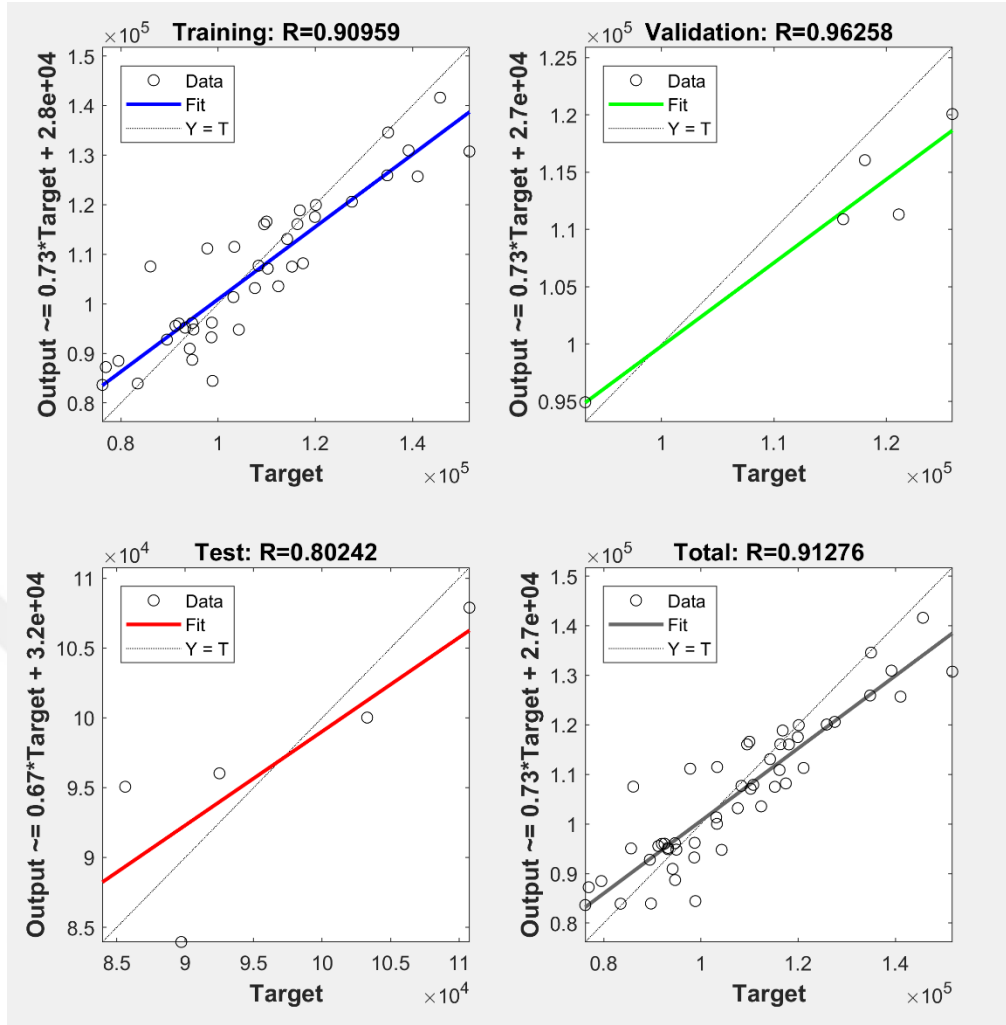
Standart Sapma	Medyan	Ortalama
7637,4	1930,2	1310,8

Her bir veri seti için hata değerlerinin sunumuna ise Tablo 3.3.'te yer verilmektedir.

Tablo 3.3. Veri Seti Hata Değerleri

Veri Seti	Minimum Hata	Maksimum Hata	Ortalama*
Eğitim Verisi	118,85	21457	1133,8
Doğrulama Verisi	1706	9811	4251,2
Test Verisi	2872,1	9450,9	-213,73

* Bu ortalama, istenen değer ile sonucun farkının ortalamasıdır



Şekil 3.48. Regresyon Grafikleri

Bütün veri setleri için regresyon değerlerinin sunumuna Tablo 3.4.'te yer verilmektedir.

Tablo 3.4. Tüm Veri Seti İçin Regresyon Değerleri

Veri Seti	R değeri	(m) Eğim	(b) Kayma
Eğitim Verisi	0,9096	0,7303	$2,789 \times 10^4$
Doğrulama Verisi	0,9626	0,7269	$2,713 \times 10^4$
Test Verisi	0,8024	0,6724	$2,179 \times 10^4$
Toplam Veri	0,9128	0,7315	$2,747 \times 10^4$

Görüldüğü üzere özellikle doğrulama verisinde gayet yüksek bir korelasyon söz konusudur.

Uygulamanın son adımı olarak ise bütün veri tiplerinin mutlak ortalama hataları ve standart sapmaları hesaplanmış ve çıktı olarak MATLAB'ın komut satırına yazdırılmıştır. Bu çıktıları Tablo 3.5.'te yer verilmiştir.

Tablo 3.5. Veri Sınıflarının mutlak ortalama hatası ve hatanın standart sapması

Veri Seti	Mutlak Ortalama Hata	Hatanın Standart Sapması
Eğitim Verisi	6160	8123
Doğrulama Verisi	4933	4322
Test Verisi	4977	6199
Toplam Veri	5919	7637

MATLAB uygulaması ile eğitilen yapay sinir ağının çıktıları ve hata değerlerinden de anlaşılacağı üzere satışların 70 bin ile 120 bin arasında olduğu örneklemeler üzerine yapay sinir ağı toplamda yalnızca 5919 gibi küçük bir ortalama hata oranı sunmuştur. Üstelik bu orana yapay sinir ağını eğitmek için kullanılmamış yeni veri girdisi de dahildir.

Bütün hesaplamaların ardından son olarak uygulamanın doğrulama kısmına gelinmiştir. 50 aylık satış tahminleri ve diğer bağımsız değişkenlerin verisine ek olarak yapay sinir ağının yeni verilere tahmin becerisini ölçmek üzere önceki verilerden farklı bir dönemden alınmış üç aylık amortisör satışı verisi ve bağımsız değişken listesi yapay sinir ağına uygulanmıştır.

Ek verinin de hesaplanması için ayrı bir script hazırlanmıştır. İlk olarak yeni veriler bir Excel dosyası içerisinde önceki veri formatı ile aynı olacak şekilde hazırlanmıştır. Ardından bu veri de eğitim verisi gibi MATLAB'a tanıtılmıştır ve onay verisi olarak "valData" değişkenine atanmıştır.

"valData" verisi ise önce normalize edilmiştir. Ardından "temel bileşen analizi" uygulanarak eğitim sırasında kullanılan veri tipine dönüştürülmüştür ve "val_norm_data" değişkenine atanmıştır. Bu adıma dair ekran görüntüsüne Şekil 3.49.'da yer verilmiştir.

```

%% Acquisition and Normalisation of New Data
valData = xlsread('ValidationData.xlsx');
val_norm_data = zeros(9, 14);
for i = 1:14
    val_norm_data(:, i) = (valData(:, i) - min(valData(:, i)))/( max(valData(:, i)) ...
        - min(valData(:, i)));
end
[~, PCA_valData, ~] = pca(val_norm_data, 'VariableWeights', 'variance', ...
    'NumComponents', 5);

```

Şekil 3.49. Yeni Verinin MATLAB’a Tanıtımı, Normalizasyonu ve Temel Bileşen Analizi

Temel bileşen analizinin uygulanması ile veri istenen formata dönüştürüldükten sonra istatistiksel analiz ve karşılaştırma için gerekli olan değişkenler hesaplanmıştır. Bu değişkenler istenen gerçek değerlerin bulunduğu “val_targetTrue” değişkeni ile yapay sinir ağına uygulanması için PCA verisinin transpozu alınarak hesaplanmış “val_input” değişkenidir.

“net()” fonksiyonu eğitim sırasında eğitilmiş yapay sinir ağının fonksiyonudur. Kendisine argüman olarak bağımsız değişkenlerin girilmesi ile hesaplanan çıktı değerini geri döndürmektedir.

Yeni verilerin hesaplanması için de bu eğitilmiş yapay sinir ağı fonksiyonuna “val_input” değeri girdi olarak gönderilip tahmin edilen amortisör satışlarının değerleri “val_output” değişkenine atanmıştır. Bu hesaplamadan sonra kök ortalama kare hatası ve istenen gerçek değerlerin yapay sinir ağı tahminine olan fark hesaplanmıştır. Kök ortalama kare hatası “val_rmse”, Hata değerleri ise “val_error” değişkenine atanmıştır. Bu hesaplamaların bulunduğu ekran görüntüsüne Şekil 3.50.’de yer verilmiştir.

%% Initialization and Application of New Data to Trained Network

```
val_targetTrue = valData(1:3:9, 15)';
val_input = PCA_valData';

val_output = net(val_input);
val_output = val_output(1, 1:3:9);
xValOutput = 1:3;

val_rmse = sqrt(mean(val_output - val_targetTrue).^2);
val_error = val_targetTrue - val_output;
```

Şekil 3.50. Yeni Verinin Yapay Sinir Ağına Uygulanması ve Hata Hesabı

Son olarak hesaplanan değerlerin analizi için istenen değer ile hesaplanan değer gösterildiği hata grafiği, hatanın histogram grafiği, Regresyon grafiği, Mutlak ortalama hata değeri ve hatanın standart sapması hesaplanmıştır. Söz konusu hesaplamaların yazdırıldığı kodun ekran görüntüsüne Şekil 3.51.'de yer verilmiştir.

%% Plotting Error, Histogram and Regression Graphs and % Displaying Mean and Standard Deviation of New Data

```
figure('Name', 'Error Between Target and Output Data');
plot(xValOutput, val_output, xValOutput, val_targetTrue, 'LineWidth', 1.5);
ylim([min(val_targetTrue) max(val_output)]);
title('Validation With New Data');
legend('Tested', 'True');

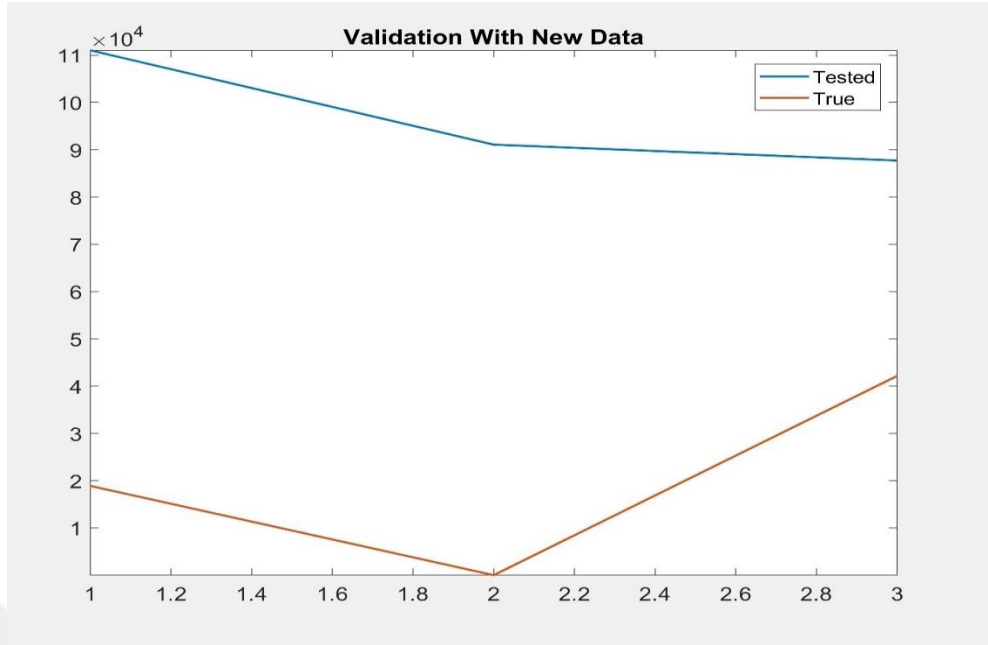
figure('Name', 'Validation Error Histogram');
ploterrhist(val_error, 'New Data Validation Error');

figure('Name', 'Regression Data');
plotregression(val_targetTrue, val_output, 'New Data Validation');

disp(['New Data Mean Validation Error: ', num2str(mean(abs(val_error))), newline]);
disp(['Standart Deviation of New Data Validation Error: ', ...
    num2str(std(errorTrain))]);
```

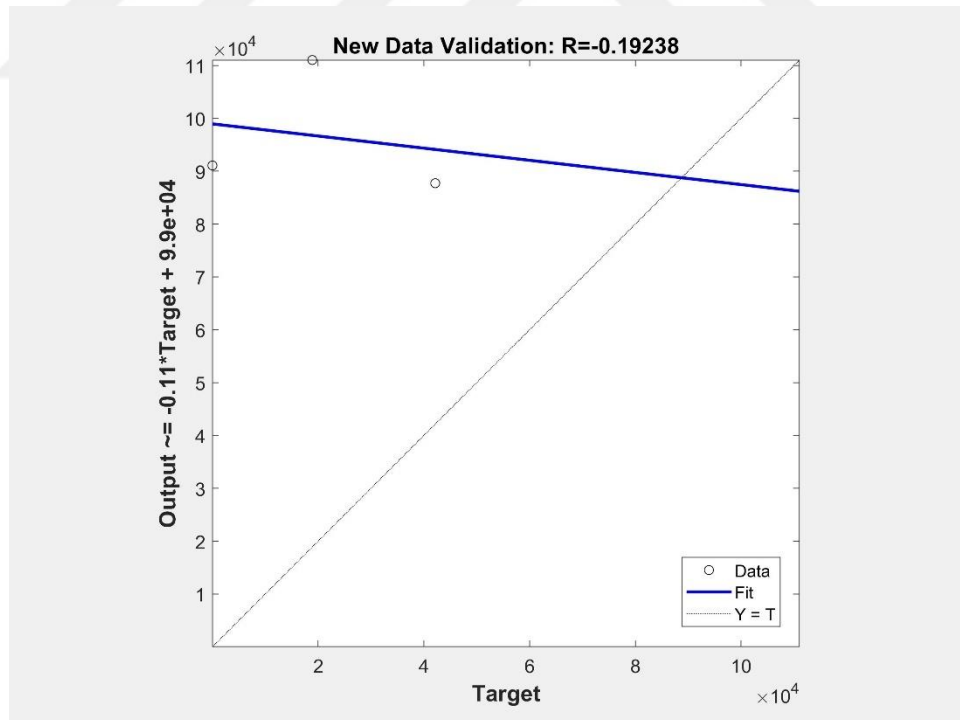
Şekil 3.51. Yeni Verinin Hata, Histogram ve Regresyon Grafiklerinin Yazdırılması

İstenen değerler ile yapay sinir ağının tahmin ettiği verilerin farkı Şekil 3.52.'deki grafikte görülmektedir;



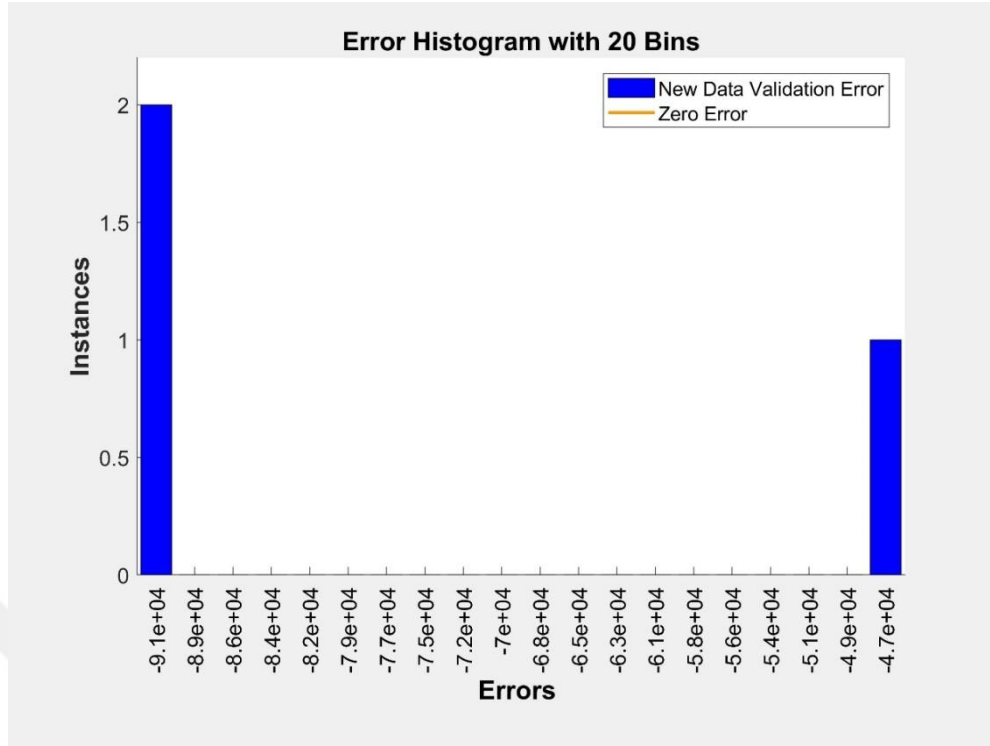
Şekil 3.52. Yeni Veri ile Tahmin Edilen Değer ile Gerçek Değerin Grafiği

Regresyon grafiğine ise Şekil 3.53.'te yer verilmiştir;



Şekil 3.53. Yeni Veri ile Tahmin Edilen Değerin Gerçek Değere Göre Regresyonu

Son olarak hata histogramının grafiğine Şekil 3.54.'te yer verilmiştir;



Şekil 3.54. Yeni Veri ile Tahmin Edilen Değerin Hata Histogramı

Ortalama hata değeri ve hatanın standart sapması ise Tablo 3.6.'da bulunmaktadır.

Tablo 3.6. Ortalama Hata Değeri ve Hatanın Standart Sapmasının Sunumu

Veri Seti	Mutlak Ortalama Hata	Hatanın Standart Sapması
Yeni Test Verisi	76242	26581

BÖLÜM 4. ARAŞTIRMA BULGULARI

Verilerle şekillendirilmiş WEKA ve MatLab programlarının tahmin modellerinden elde edilen rakamlar ve gerçek satış rakamlarının karşılaştırılması bu bölümde sağlanmaktadır. Öncelikle bağımsız değişkenlerin birbirleri ile arasında ciddi bir ilişki gözlenmiş, bu korelasyonlar dikkate alınarak Temel Bileşenler Analizi uygulanan girdilerle YSA'lar kurulmuştur. 2016 Ocak - 2020 Şubat ayları satış grafiğine bakılarak firmanın duruşta olduğu aylarda satışın oldukça düşük olduğu gözlemlenmişken Aralık aylarında yaşanan satış artışının firmalar tarafından yapılan kampanyalar ve indirimlerle bağlantılı olabileceği yorumu yapılmıştır. Sonuç olarak, amortisör satışlarının pandemi dönemi öncesinde artış gösteren bir eğilimde olduğu anlaşılmış ancak yapılan tahminler ile gerçekleşen satışlar kıyaslandığında satışların pandemiden oldukça olumsuz etkilendiği gözlenmiştir.

Tahmini yapılan döneme ait güncel satış rakamlarının sunumu Tablo 4.1.'deki gibi olmuştur. Buna göre ilgili periyodun satış rakamları; Mart ayında 18.900; Nisan ayında, 22 ve Mayıs ayında 42.159 olarak firma tarafından bildirilmiştir.

Tablo 4.1. Gerçekleştirilen Amortisör Satışları ve Bağımsız Değişkenler

Otosatis	Motorin	Dolar	Ihracat	Krediler	Faizler	GSYİH	SAU	TUFE	YIUFE	Tuk GüvenEn	ReKesim GüvenEn	Eko GüvenEn	Maddi Beklenti	Amor Satis
10121	5,677614	6,330055	13188396	7080	13,35	1071098	113,0113	450,58	468,69	81,11938	98,6	93,1253	78,58403	18900
4632	5,132681	6,83811	8636028	7000	13,91	1041643	76,47981	454,43	474,69	78,14239	62,3	52,39653	72,48901	22
7162	5,253941	6,942647	11065739	7147	13,12	1041643	92,86566	460,62	482,02	82,66691	73,5	62,73442	79,24338	42159

Güncel satış rakamlarından görüleceği üzere, bu satışların yapılmasında etkili parametreler, bağımsız değişkenler bulunmaktadır. Literatürdeki çalışmalar ve piyasadaki gözlemler sonucu amortisör satışını etkileyebilecek bağımsız değişkenler; hafif ticari aracın satışı, motorin fiyatı, döviz kuru, aylık dış ticaret, taşıt kredileri toplamı, bankalarca açılan kredilere uygulanan ağırlıklı ortalama faiz oranları, GSYİH, Sanayi Üretim Endeksi, TÜFE, Yi-ÜFE, Tüketici Güven Endeksi, Reel Kesim Güven Endeksi, Ekonomik Güven Endeksi ve hane halkı maddi durum beklentisi bir ilişki parametresi düzeni izlediği görülmektedir.

Gerçek Satış verileri WEKA ve MatLab'daki sonuçlarla karşılaştırılarak bu bölümde incelenecektir. Önce WEKA programındaki satış tahmini rakamlarını içeren değerler daha sonra MatLab programındaki satış tahmini rakamlarını içeren değerler ayrı ayrı sunulmaktadır. Bu değerlerin gerçek verilerle karşılaştırılması için gerçek satış değerleri de “gerçekleşen satış” başlığının altındaki sütunda sergilenmektedir.

4.1. WEKA ile Elde Edilen Tahmin Sonuçları

WEKA gelecek simülasyonu ve satış tahmini eldesine ait veri seti üzerinde yapılan çalışmalar Bölüm 3.5'te ele alınmıştır. Burada WEKA işlev ve araçları kullanılarak veri seti üzerinde yapılan çalışmanın tüm aşamaları ayrıntılarıyla ele alınmıştır.

Çalışmada 50 aylık çoğaltılmış veriye ait YSA modeli, gizli katman sayısı modele uygun olacak şekilde otomatik atanması komutu verildikten sonra çeşitli denemeler sonucu hata oranı %86,6 olacak şekilde kurulmuştur. Bu hata oranı tahmin süreçlerinde kullanılabilecek düzeyde olduğu kabul edilmiş ve model kaydedilmiştir.

Model üzerine eklenen pandemi dönemine ait 3 aylık veri YSA'ya eklenerek tahminleme çalışılmıştır. WEKA sonuç ekranında tabloya yazdırma özelliği bulunmaktadır. Sonuçlarda tek bir çıktı katmanı üzerinde tek bir değer sunan WEKA ortalama değer tespitini 115.805 olarak sunmuştur. Bu sunuma ait tabloya ve gerçek satış değerleriyle karşılaştırmasına Tablo 3.1.'de yer verilmektedir.

Tablo 3.1. WEKA YSA Tahmin Sonucu ile Gerçek Satış Değerlerinin Karşılaştırılması

	Weka Tahmini	Gerçekleşen Satış	Fark
Mart	115805	18900	96905
Nisan	115805	22	115783
Mayıs	115805	42159	73646

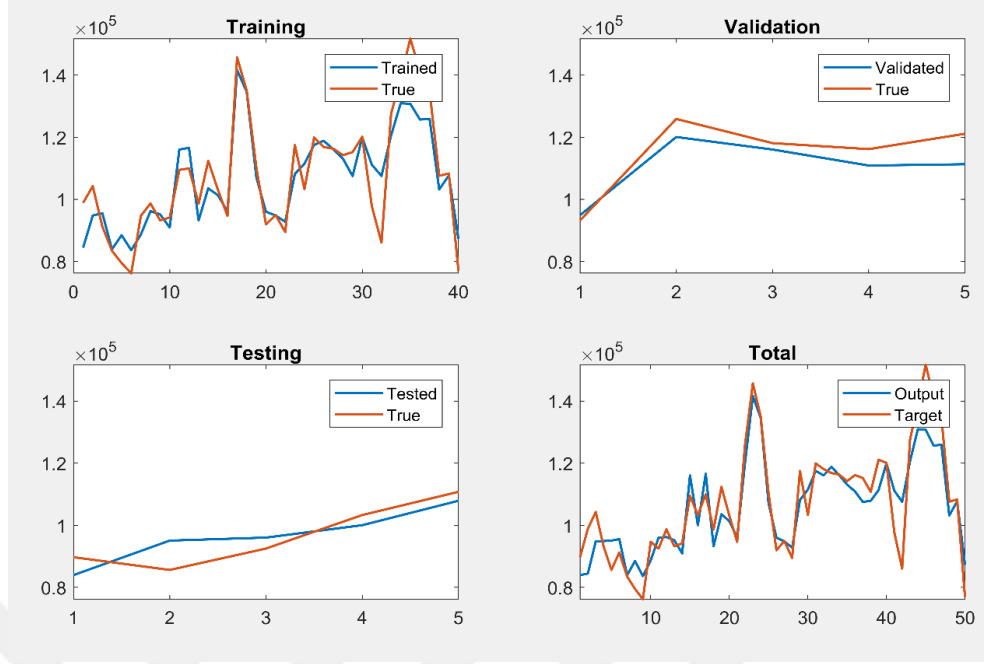
Tablo 3.1.'de görüldüğü üzere, K-NN tabanlı en yakın komşu kullanımının ortak değeri olan sayıyı WEKA çıktı olarak sunmaktadır. Bu da veri dasetinin ortalama değeri olan 115.805 değeri ile ifade edilmektedir. Hemen yan sütunda yer alan gerçek satış rakamları Mart, Nisan ve Mayıs ayı değerleri ile karşılaştırmalı olarak sunulmaktadır.

Tablo 3.1.'de WEKA tahmin değeri olan 115.805 rakamı gerçek satış değerleri olan Mart satış değeri 18.900 rakamından, 96.905 adet fazla olarak görünmektedir. Bu durum WEKA'nın %512 oranında yanlıgısını göstermektedir. Bu da gerçek satış rakamlarının WEKA tahmin parametresi ile sergilenerek gelecek simülasyonu oluşturulmak üzere kullanılmasında yüksek oranda hata miktarı içerdiğini göstermektedir. WEKA tahmin değerlerinin çalışma biçimi gereği en yakın nokta tespiti ve veri kümesini ezberlemeye yönelik çalışma biçimini yansıtmıştır. Bu büyük hata oranı Nisan ayında, %5262'lik sapmaya, Mayıs ayında ise %174 oranında sapmanın oluşmasına yol açmaktadır.

4.2. MATLAB ile Elde Edilen Tahmin Sonuçları

MatLab'de kurulan YSA 2 gizli katmanının ilkinde 4 ikincisinde 2 sinir hücresi katman çıktısını süzmesiyle ve son çıktı katmanı ile değerleri harmanlama konusunda aktif olarak çalışmaktadır.

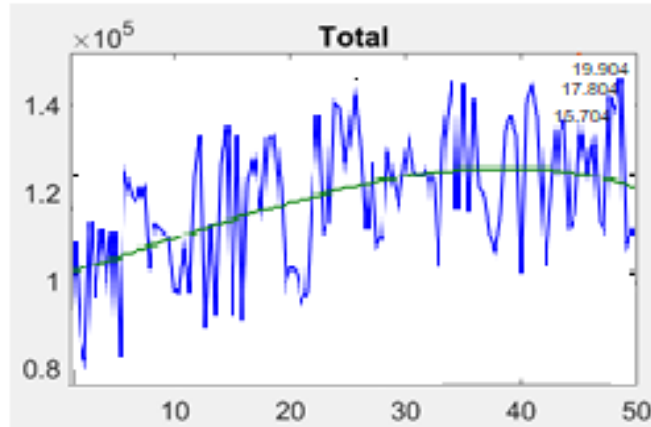
MatLab değerlerinde veri büyütülmeden matris düzenini tam olarak işlevli kullandığından köşe matrisi değerlerinin frekanslarıyla sunmaktadır. Çıktıları tam bir liste biçiminden çok görsel grafik düzeniyle elde etmeyi sağlamaktadır. Şekil 4.1.'de sonuçların görseline yer verilmektedir. Bu tabloda Total olarak adlandırılan grafik üzerinde "Target" çıktısı işaretlenmediği takdirde çıktı (Output) tek total grafik sunmaktadır.



Şekil 4.1. MATLAB çıktı ekranı görselleri

Son çıktı diyagramında “Target” diyagramı kaldırılıp sadece “Output” seçeneği işaretlendiğinde grafik değeri, üzerine fare imlecinin getirilmesiyle okunmaktadır. Değerlerin sunumuna ait “Total Output “ grafiğine Şekil 4.2.’de yer verilmektedir.

Değerler grafik çıktıda farenin hareketi ile (1. değerden sonraki değere geçişte) kaybolmaktadır. Bu nedenle Mart, Nisan ve Mayıs ayları için imlecin gösterdiği değerler, gösterimin sağladığı alanların üzerine yazılması işlemi “paint” programı ile sağlanmıştır.



Şekil 4.2. MatLab Çıktı Grafiği

Değerlerin okunması ancak şablon üzerinden sağlanabildiğinden çıktıların sunumuna ait okunan değerlerin tablolaştırılması manuel olarak sağlanmış olup, okunan değerler tabloya aktarılmıştır. Bu değerlere Tablo 4.2.'de yer verilmektedir.

Tablo 4.2. MatLab YSA Tahmin Sonucu ve Gerçek Satış Değerleri ile Karşılaştırılması

	Matlab Tahmini	Gerçekleşen Satış	Fark
Mart	15704	18900	-3196
Nisan	17804	22	17782
Mayıs	19904	42159	-22255

Tablo 4.2.'de yer alan MatLab grafiğinden okunan değerler, gerçek satış değerleri ile birlikte verilmiş ve MatLab tahmin değerinin, gerçek satış değeri arasında yaratmış olduğu farklar "Fark" başlığı ile adlandırılan sütuna yazılmış halde sunulmaktadır.

MatLab, satış tahminleri Mart ayı çıktılarında % 20 hata ile tespit sunarken Nisan ayı gerçek satışlarının çok düşük seyrini tahmin etmede %99 oranında hata ile sunmuş, satışlardaki ani düşüşü tahmin edemeyerek büyük bir yanlış payı sunmuş olarak görülmektedir. Son olarak Mayıs ayı satış rakamlarında, Nisan düşüşünü öngörmediği için büyük bir artışı da öngörememiş ve %111 oranında yanlışlığı sunmakla sınırlı kaldığı görülmektedir.

4.3. WEKA ve MATLAB Programları ile Ulaşılan Tahminlerin Kıyaslanması

MatLab tahmin sonuçlarının, gerçek satış değerleri ile uyumu aynı zamanda WEKA'nın şablon akışındaki uyumu söz konusudur. Ancak WEKA çıktıları aylık bazda olmayan sadece ortalama kendi algoritmaları doğrultusunda, ortalama veya en yakın komşu değerlerinin ortalaması biçiminde sunması bu değerleri yakalamasına engel teşkil etmemektedir. Çünkü her iki simülasyon programı için çalışılan model aynı veri datasında eğitilmiştir. Yalnız sonuçları WEKA tek çıktı ile MatLab bir grafikte sunmaktadır.

Gelecek simülasyonu değerlerinde grafik akışını kullanan MatLab son üç zaman serisinin Output grafiği değerlerinin üzerine gelindiğinde okumayı mümkün kılmaktadır.

WEKA'nın hata oranı, genel anlamda görsel işleme biçimine yoğunlaşan kısmının daha yoğun olarak düşünüldüğünü göstermektedir. Bir görüntü işleme sonucu bu düzende daha iyi sonuç verebilirken, Veri Madenciliği, çıktı değerlerinin katmanlar halinde sunulmaması ve oluşturduğu YSA yapısının öğrenmeden çok kümeyi ezberleme yöntemiyle çalışma düzeni, veriye dayalı hata olasılığının yanlış tahliline yol açan etkenler olarak düşünülmelerini sağlamaktadır.

MatLab ise daha girift yapısıyla gerçek zamanlı çalışmada, görüntü işlemeye, dinamik modellemeden, matematik algoritmalarının ve lineer denklemlerin çözülmesine dek farklı matris düzeni laboratuvarları sunmaktadır. Bu da daha karmaşık sorunların çözülmesinde daha başarılı sonuçların eldesine yol açmaktadır.

MatLab 'da gerçek satışlara WEKA'ya göre daha iyi sonuçlar sunduğu görülse de, Covid-19 Pandemi Dönemi ekonomik etkisinin özellikle Nisan ayındaki düşmesi konusunda ne yazık ki gerçek değerlerin oldukça üzerinde kaldığı görülmektedir. Bu da simülasyon programlarına “kriz” parametrelerinin de işlenmesi gerektiği yorumunu doğrulamaktadır.

BÖLÜM 5. TARTIŞMA VE SONUÇ

Bu çalışmada öncelikle YSA yapıları incelenmiş, bunu takiben YSA tabanının farklı yapılarını kullanan WEKA ve MatLab ile, Covid 19 kriz dönemi hafif ticari bir araç modeli için üretilmiş olan amortisör satışlarının tahmin edilmesine yönelik modeller geliştirilerek tahmin sonuçlarının sergilenmesi sağlanmıştır.

Kurulan YSA modellerinde bağımsız değişkenlerin korelasyonları kullanılarak pandemi dönemi öncesi daha başarılı tahminlemeler elde edilirken pandemi krizinin yoğun olarak yaşandığı ve Türkiye'nin erken pandemi dönemi olarak adlandırabileceğimiz 2020 yılı Mart, Nisan ve Mayıs ayları, ülkedeki dinamik dengelere, tüm araç satışları gibi piyasaya ve özellikle karantina gibi süreçlerle pandeminin en etkin hissedildiği Nisan ayı dönemindeki satışların fazlaca düşmesi gibi şartlar, iki simülasyon programında da bu parametreye ilişkin “kriz” tanımı bulunmadığından öngörülemediği, simülasyon sonuçlarında açıkça görülmüştür.

WEKA'da 2016 yılı Ocak ayından başlayarak 2020 yılı Şubat ayına dek gerçekleşen satışlara göre bir YSA tahminlemesi çalıştırılmış, burada elde edilen YSA'nın üzerine pandemi dönemi 3 aylık tahmini K-NN algoritmasının dahil olduğu bir ortalama çıktısı ile elde edilmiş ve bu ortalama çıktıyı, kriz dönemlerini içermeyen Ocak 2016 ila Şubat 2020 aralığındaki satışlarını baz alarak yaptığını programın çalışma mantığının açıklandığı 3. bölümde geniş olarak ele alınmış olan WEKA XOR sorunu sunumuna dayanarak anlaşılmasını sağlamaktadır.

Covid-19 pandemi krizinin ülkemizdeki ilk dönemleri yani erken pandemi dönemi gereğince eğitim seti ve tüm datada “var olmayan” bir öngörüü gerektirmekteyken bu “öngörülü datanın bulunmaması” veya benzer bir düşüşün daha önceki satış zamanlarında yaşanmaması nedeniyle olduğunu düşünmeyi gerektirmektedir.

Yapay zeka yani YZ'nin basit modellemeleri ve algoritmalarına dayanan yapısının elverişli parametrelerini işleyen WEKA parametreleri ve MatLab parametrelerinin kıyaslanması, gerçek kriz dönemi satışlarıyla kıyaslandığında yüksek hatalı çıktılar ürettikleri görülmüştür. Ancak bu hatalar, iki simülasyon programının da kriz bileşenlerini içermemesi nedeniyle olduğunu düşündürmektedir. Bununla birlikte, WEKA yapısı gereği K-NN algoritmasına dayanan ortak çıktı yorumu, ortak değerli simülasyon çıktısının tümüyle uzağında gerçekleşen satış rakamlarına ulaşıldığını göstermektedir.

MatLab ise bir matris laboratuvarı olduğunu sonuç çıktılarında yeniden kanıtlamakta ancak datada bir düşüşün tahminini gerektiren satış rakamı bulunmadığından artış eğilimli bir grafik çıktısı elde edilmesi sonucuna varmamızı sağlamıştır. MatLab satış tahmini rakamlarında özellikle Mart ayına ilişkin satışları neredeyse hatasız (3.196 adet eksik) sunması bunun bir göstergesi olarak karşımıza çıkmaktadır. Ancak Nisan ayındaki hatalı tahmin 17.782 adet fazla satışın gerçekleşmesinden çok, Nisan ayı itibarıyla pandeminin maddi tüm etkilerinden kaynaklı ilgili hafif ticari aracın kullanım ihtiyacının hatta pazarının neredeyse son bulması halini yansıtmaktadır. Dolayısıyla bu hatalı tahmin değerinden çok, kriz dönemi “öngörü” aracının eksikliği ile ifade edilebileceği gibi, datanın bu biçimde bir “kriz” dönemini içermemesinden de kaynaklanmaktadır. Son MatLab çıktısına sahip Mayıs dönemi tahmin ve gerçekleşen satış değerlerinin arasındaki büyük farkın, Nisan dönemi belirsizliğinden kaynaklandığını anlamamızı sağlamaktadır. Şöyle ki; Nisan ve Mayıs ayı satışlarının basit bir aritmetik ortalamasını alacak olursak yani $22+42129=42.151 / 2 = 21.075,5$ gibi bir rakam elde etmiş oluruz. Bu ortalama değer aslında MatLab tahmin çıktılarıyla ± 3.000 değeri kadar yarıldığını göstermektedir. Bu ortalama değerle tahmin çözümü esas alınacak olursa MatLab'ın ilk değer tahmin başarısı gibi $\cong \%20$ yanılğı payını devam ettirdiğini görmeyi mümkün kılmaktadır.

Benzer bir biçimde WEKA tek sonuç değeri olan 115.804 sayısı da tahmin yanılığının oldukça hatalı olduğunu gösterecektir. MatLab'da olduğu gibi Nisan ayındaki satışların çok düşük olacağı tahmini, data'nın bu tahmin değerini içermemesi ve WEKA'nın da “kriz dönemi” için ayrı parametre sunmamasının asıl yanılğı

düşüncesini doğurduğu görülmüş olacaktır. “data”nın herhangi bir biçimde “kriz döneminde yaşanan satışlardaki ani düşüşü barındıran bir veri içermemesi” temeline dayandırılmaktadır.

WEKA ve MatLab temelli sonuçlar ve bu programların temellendirilmesinde kullanılan YSA katmanlarının incelenmesi çıktıları MatLab ve WEKA modelinin çok katmanlı yapılarına dayandırılacak olursa, daha fazla YSA barındıran MatLab’ın daha iyi bir tahmin sunduğu gerçekliğini görmemizi sağlamaktadır.

Tüm bu varsayımsal çıkarımlar, çalışmanın aynı zamanda pandemi dönemi özelinde yaşanan kaybı yansıtmak amaçlı yapıldığını da göstermektedir. Gerçek satış rakamları, “kriz dönemi” verilerinden yoksun olarak kriz dönemi verilerinin tahmin edilmesinden uzaklaşılmasında rol oynamaktadır. Böylece YSA’da yansıtılan hata oranlarına göre tahminlemeler gerçekleştirilen amortisör firması için tahmini satış kaybını yansıtmaktadır. Dolayısıyla en düşük satış olan Nisan ayında gerçekleşen 22 amortisör satışı, data’da bulunmadığından bir eğitim seti ile öğrenmeyi sağlayan YSA tarafından görülememiş ve tahmin değerleri bu miktarın çok ötesinde çıktıların elde edilmesi sonuçlarını doğurmuştur. Bununla birlikte yüksek yanılğı payları için, Nisan ayı satışlarının dolayısıyla sonraki ay olan Mayıs ayı satışlarıyla birlikte yorumlanmasını neredeyse gerekli kılmaktadır.

2 gizli katmanındaki sinir hücreleri ile daha fazla YSA yapısına sahip çözümleme sunan MatLab’ın gerçek satış verilerine daha yakın çıktılar sunmakta başarılı olduğu ayrıca bu çalışmada yansıtılmıştır. Bu çalışmanın bulgular bölümüne dayanarak, Tablo 3.1. WEKA sonuçları ve Tablo 4.2. MatLab sonuçlarıyla söylemek mümkündür.

Sonuç olarak, YSA katmanları, girdi bölümü, öğrenme bölümü ve çıktı bölümü olmak üzere basit yapısını sunarken WEKA’da tek çıktı, MatLab’ta grafiksel çıktı ile sunulması dolayısıyla ihtiyaç duyulan sonuçların okunmasını sağlayan alanların fazla olması, bu çalışmada kurulan YSA modellerinin yapısında MatLab’in WEKA’dan daha fazla katmana sahip olması ve tek öğrenme biçiminin “toplam kümeyi

ezberleme” öğrenmesi yaptığı düşünülecek olursa MatLab’ın tahminlemelerde WEKA’ya nazaran daha iyi sonuçlar ürettiğini göstermiştir.

Çalışmanın öğrenme verilerine programlarda öngörülemeyen pandemi dönemine ait daha fazla aylık data eklenerek çok katmanlı yeni YSA’lar kurulabilir ve birçok sektör bu ve benzeri çalışmaları hem krizin maddi etkilerini anlamak, hem planlama süreçlerinde yardımcı kaynak olarak kendilerine rehber olarak alabilirler.



KAYNAKLAR

- Abatan, S., & Olayemi, M. (2014). The Roles of Statistical Software in Data Analysis. *International Journal of Applied Research and Studies*, 3(8) 64-89.
- Abolghasemi, M., Hurley, J., Eshragh, A., & Fahimnia, B. (2020). Demand forecasting in the presence of systematic events: Cases in capturing sales promotions. *International Journal of Production Economics*, 230.
- Abraham, J., & Shrivastava, S. (2018). Dc Motor Speed Control Using Machine Learning Algorithm. *International Journal of Engineering Research & Technology*, 7(04), 456–470.
- Acito, A. A., Folsom, D., & Zhao, R. (2021). Management Sales Forecasts and Firm Market Power. *Journal of Accounting, Auditing and Finance*, 36(2), 278–303.
- Arslankaya, S., & Öz, V. (2018). Time Series Analysis of Sales Quantity in an Automotive Company and Estimation by Artificial Neural Networks. *Sakarya University Journal of Science*, 1–18.
- Bakri, R., Data, U., & Saputra, A. (2019). Marketing Research : The Application of Auto Sales Forecasting Software to Optimize Product Marketing Strategies. *Journal of Applied Science, Engineering, Technology, and Education*, 1(1), 6–12.
- Benhamida, F. Z., Kaddouri, O., Ouhrouche, T., Benaichouche, M., Casado-Mansilla, D., & López-De-Ipiña, D. (2021). Demand forecasting tool for inventory control smart systems. *Journal of Communications Software and Systems*, 17(2), 185–196.
- Bolt, G., (1994), *Market and sales forecasting: a total approach*, Kogan Page, Londra.
- Chih-Min, M., Wei-Shui, Y. and Bor-Wen, C. (2014). How the Parameters of K-nearest Neighbor Algorithm Impact on the Best Classification Accuracy: In Case of Parkinson Dataset, *Journal of Applied Sciences*, 171–176.
- Creswell JW, Plano Clark VL. *Designing and conducting mixed methods research*. 2nd ed. Thousand Oaks, CA: Sage; 2014.
- Dellino, G., Laudadio, T., Mari, R., Mastronardi, N., & Meloni, C. (2018). Microforecasting methods for fresh food supply chain management: A computational study. *Mathematics and Computers in Simulation*, 147, 100–120.

- Farid, S., Tashfeen, R., Mohsan, T., & Burhan, A. (2021). Forecasting stock prices using a data mining method: Evidence from emerging market. *International Journal of Finance and Economics*.
- Hornik, K., Stinchcombe, M., White, H., (1989), Multilayer feedforward networks are universal approximators, *Neural networks*, vol. 2, no. 5, pp. 359-366,
- Huang, S., Cai, N., Pacheco, P.P., Narrandes, S., Wang, Y. and Xu, W. (2017). Applications of Support Vector Machine (SVM) Learning in Cancer Genomics, *Cancer Genomics Proteomics*, 15, 41 – 51.
- Jarvinen, P. (2007). Action research is similar to design science. *Quality and Quantity*, 41(1), 37–54.
- Jayanthi, R., & Florence, L. (2019). Software defect prediction techniques using metrics based on neural network classifier. *Cluster Computing*, 22, 77–88.
- Jingwei, L., Yifei, T., Shaofeng, W., Qingmeng, T., & Dongbo, L. (2016). Welding Robot Kinematics Analysis and Trajectory Planning. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 14(2A), 92.
- K., D. Srivastava, Lekha, B. (2010). Data Classification using Support Vector Machine, *Journal of Theoretical and Applied Information Technology*.
- Karaatlı, M., Helvacıoğlu, Ö. C., Ömürbek, N., & Tokgöz, G. (2012). Yapay Sinir Ağları Yöntemi İle Otomobil Satış Tahmini. *Uluslararası Yönetim Ekonomisi ve İşletme Dergisi* , 8 (17) 1-14.
- Khamila, K.N. , Sabria, M.F.M., Yusop , A.M., Mohamed, R., & Sharuddin, M.S. (2020). Modelling and Simulation of the Performance Analysis for Peltier Module and Seebeck Module using MATLAB/Simulink. *Jurnal Kejuruteraan*, 32(2), 231–238.
- Khan, M. A., Saqib, S., Alyas, T., Ur Rehman, A., Saeed, Y., Zeb, A., ... Mohamed, E. M. (2020). Effective Demand Forecasting Model Using Business Intelligence Empowered with Machine Learning. *IEEE Access*, 8, 116013–116023.
- Kryze, J., Burchila, B., & Woimbee, V. (2002). Embrayages à sec Modelisation de la reponse dynamique. *Techniques de l'ingénieur Transmission de Puissance Mecanique: Accouplement, Embrayage, Freinage*, base docum(ref. article : bm5856).
- Kshatri, S. S., Singh, D., Narain, B., Bhatia, S., Quasim, M. T., & Sinha, G. R. (2021). An Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using Stacked Generalization: An Ensemble Approach. *IEEE Access*, 9, 67488–67500.

- Kuo, R. J., Tseng, Y. S., & Chen, Z. Y. (2016). Integration of fuzzy neural network and artificial immune system-based back-propagation neural network for sales forecasting using qualitative and quantitative data. *Journal of Intelligent Manufacturing*, 27(6), 1191–1207.
- Lakshmanan, B., Vivek Raja, P. S. N., & Kalathiappan, V. (2020). Sales Demand Forecasting Using LSTM Network. In *Advances in Intelligent Systems and Computing* (Vol. 1056, pp. 125–132). Springer.
- Lalou, P., Ponis, S. T., & Efthymiou, O. K. (2020). Demand Forecasting of Retail Sales Using Data Analytics and Statistical Programming. *Management and Marketing*, 15(2), 186–202.
- Lantz, B., (2015), *Machine learning with R*, Packt Publishing, Birmingham.
- Lasek, A., Cercone, N., & Saunders, J. (2016). Sales and customer demand forecasting: Literature survey and categorization of methods. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST* (Vol. 166, pp. 479–491). Springer Verlag.
- Massaro, A., Vitti, V., & Galiano, A. (2018). Model of Multiple Artificial Neural Networks Oriented on Sales Prediction and Product Shelf Design. *International Journal on Soft Computing, Artificial Intelligence and Applications*, 7(3), 01–19.
- Mira, J. ,(1995) , *Neural Computation Lecture Notes In Computer*, International Workshop on Artificial Neural Networks Malaga- Torremolinos ,Spain June 1995, Proceeding, IWANN '95. Sprenger ISBN 3 1741 102 204 973 pp. 4-15.
- Mulyani, S., Hayati, D., & Sari, A. N. (2021). Analisis Metode Peramalan (Forecasting) Penjualan Sepeda Motor Honda Dalam Menyusun Anggaran Penjualan Pada Pt Trio Motor Martadinata Banjarmasin. *Dinamika Ekonomi - Jurnal Ekonomi Dan Bisnis*, 14(1), 178–188.
- Ninness, B. (2000). Using MATLAB to simulate systems governed by Linear Ordinary Differential Equations (LODE ' s) 2 Example of Car Shock Absorber Simulation. *Computer Engineering*, 1(2), 1–10.
- Rumelhart, D. E., & the PDP Research Group, (1986), *Parallel distributed processing: Explorations in the microstructure of cognition*, Vol. 1. Psychological and biological models. Cambridge, Massachusetts: MIT Press.
- Rumelhart, D. E., McClelland, J. L., & the PDP Research Group, (1986b), *Parallel distributed processing Explorations in the microstructure of cognition*, Vol. 2. Foundations, Cambridge, Massachusetts, and MIT Press.
- S. Ruder,(2016), An overview of gradient descent optimization algorithms, ArXiv e-prints, vol. 1609, <http://adsabs.harvard.edu/abs/2016arXiv160904747R> Eriřim Tarihi:12.12.2020, Ankara.

- Scheele, L. M., Thonemann, U. W., & Slikker, M. (2018). Designing incentive systems for truthful forecast information sharing within a firm. *Management Science*, 64(8), 3690–3713.
- Scherbak, L. M. (2020). The Methodology of Research on Resource Consumption Processes. *Thermophysics and Thermal Power Engineering*, 42(3), 99–105.
- Sengar, R. S., & Ahmed, Dr. F. (2019). Review on Trends in Machine Learning Applied to Demand & Sales Forecasting. *SMART MOVES JOURNAL IJOSCIENCE*, 5(6), 4.
- Tekin, M., (1996), *Üretim Yönetimi*, Arı Ofset Matbaacılık, Konya.
- Vapnik, V., (1963). Pattern Recognition Using Generalized Portrait Method, pp.774–780.
- Venkata Hari Prasad G., Narayana Thalluri, L., (2020). Enhanced Performance of PCG Signal using Effective Feature Extraction Method. *International Journal of Recent Technology and Engineering*, 9(2), 809–813.
- Wang, C. H., & Yun, Y. (2020). Demand planning and sales forecasting for motherboard manufacturers considering dynamic interactions of computer products. *Computers and Industrial Engineering*, 149.
- Xie, W., Shen, L., & Zhong, Y. (2017). Two-dimensional aggregate warranty demand forecasting under sales uncertainty. *IIE Transactions*, 49(5), 553–565.
- Yang, F. C. (2016). Efficiency decomposition in dealers from the perspectives of demand forecasting, sales force, and inventory control: a case study. *Production Planning and Control*, 27(16), 1334–1343.
- Zheng, P., Wang, R., & Gao, J., (2020). A Comprehensive Review on Regenerative Shock Absorber Systems. *Journal of Vibration Engineering & Technologies* 2(8):225–246.
- Zontul, M., Aydin, F., Dogan, G., Sener, S., & Kaynar, O. (2013). Wind speed forecasting using reptree and bagging methods in Kırklareli-Turkey. *Journal of Theoretical and Applied Information Technology*, 56(1), 17–29.

ÖZGEÇMİŞ

Adı Soyadı : Sena BALKIŞLI

ÖĞRENİM DURUMU

Derece	Eğitim Birimi	Mezuniyet Yılı
Yüksek Lisans	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü/ Endüstri Mühendisliği / Mühendislik Yönetimi	Devam ediyor
Lisans	Sakarya Üniversitesi/Mühendislik Fakültesi/Endüstri Mühendisliği	2017
Lise	Behiye Dr. Nevhiz Işıl Anadolu Lisesi	2013

İŞ DENEYİMİ

Yıl	Yer	Görev
2018-Halen	ZF Sachs Süspansiyon Sistemleri	Endüstri Mühendisi

YABANCI DİL

İngilizce

ESERLER (makale, bildiri, proje vb.)

ISPEC 11th International Conference on Engineering&Natural Science (September 18-19,2021 / Muş, Turkey)

HOBİLER

Dans etmek