

T.C
İSTANBUL KÜLTÜR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

GERÇEK ZAMANLI KAÇAK ELEKTRİK KULLANIMININ DERİN
ÖĞRENME İLE TESPİT EDİLMESİ

YÜKSEK LİSANS TEZİ
BARAN BEREKET YILDIZ
2000006397

Anabilim Dalı: Bilgisayar Mühendisliği
Programı: Bilgisayar Mühendisliği

Tez Danışmanı: Dr. Öğr. Üyesi Fatma Patlar AKBULUT

HAZİRAN 2022

T.C
İSTANBUL KÜLTÜR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

GERÇEK ZAMANLI KAÇAK ELEKTRİK KULLANIMININ DERİN
ÖĞRENME İLE TESPİT EDİLMESİ

YÜKSEK LİSANS TEZİ
BARAN BEREKET YILDIZ
2000006397

Anabilim Dalı: Bilgisayar Mühendisliği

Programı: Bilgisayar Mühendisliği

Tez Danışmanı: Dr. Öğr. Üyesi Fatma Patlar AKBULUT

Juriler : Doç.Dr.Akhan AKBULUT

Prof.Dr.Özgür Koray ŞAHİNGÖZ

HAZİRAN

İÇİNDEKİLER

ŞEKİLLER	iii
ÖZET	vi
ABSTRACT	vii
ÖNSÖZ	viii
1. GİRİŞ	1
2. MATERYAL VE METOD	4
2.1. Türkiye’de ve Dünya’da Kaçak Elektrik Kullanımı	4
2.2.1. Veri Ön İşleme Yöntemleri	6
2.2.3. Aykırı Değer Belirleme Yöntemi	8
2.2.4. Değişken Tipleri ve Dönüşüm Yöntemleri	9
2.2.5. Veri Normalleştirme İşlemleri	11
2.2.6. Varsayım ve Ki Kare Testi	12
2.3. Derin Öğrenme	13
2.3.1. Derin Öğrenme Mimarileri	13
2.4. Keras ile Model Oluşturulması	18
2.4.1. Keras Model ile Metriklerin Belirlenmesi	19
2.4.2. Aktivasyon ve Optimizasyon Fonksiyonlarının Belirlenmesi	20
3. BULGULAR	24
3.1. Derin Öğrenme ve Gerçek Zamanlı Kaçak Elektrik Tüketim Analizi	24
3.1.1. Veri Ön İşleme Yöntemi	25
3.1.2. Veri Normalleştirme İşlemleri	25
3.2. Derin Öğrenme Model Sonuçları	29
4. SONUÇ	32
5. KAYNAKLAR	33

KISALTMALAR

CNN	: Konvolüsyonel – Evrişimli Sinir Ağı
GPU	: Grafik İşlemci Birimleri
LSTM	: Uzun Kısa Vadeli Bellek – Long Short – Term Memory
RNN	: Tekrarlayan Sinir Ağı – Recurrent Neural Network
GRU	: Özyinelemeli Geçit Ağ Mimarisi – Gated Recurrent Unit
RELU	: Reaktifiyeli Doğrusal Ünite – Rectified Linear Unit

ŞEKİLLER

Şekil 2.1	:	Değişken Veri Tipleri	10
Şekil 2.2	:	Tekrarlayan Sinir Ağı Hücre Yapısı	15
Şekil 2.3	:	LSTM Hücre Yapısı	16
Şekil 2.4	:	GRU Hücre Yapısı	17
Şekil 2.5	:	Keras Adımları	18
Şekil 2.6	:	Sigmoid Fonksiyonu	21
Şekil 2.7	:	Tanh Fonksiyonu	22
Şekil 2.8	:	RELU Aktivasyon Fonksiyonu	22
Şekil 3.1	:	Faktörize Karne No Değişken Histogramı	25
Şekil 3.2	:	Faktörize Karne No Değişkeni Box – Cox Dönüşümü	26
Şekil 3.3	:	Faktörize Tarife Adı Değişken Histogramı	27
Şekil 3.4	:	Abone Trafo Tüketim Oranı Histogramı	27
Şekil 3.5	:	Abone Trafo Tüketim Oranı Normalleştirme Dağılımı	28
Şekil 3.6	:	COV Değişkeni Box-Cox Dönüşümü	28
Şekil 3.7	:	Abone Trafo Tüketim Oranı Normalleştirme Dağılımı	29
Şekil 3.8	:	Sequential (Sıralı) Model Deneme Sonuçları	30
Şekil 3.9	:	Sequential (Sıralı) Model Doğruluk Sonuçları	30
Şekil 3.10	:	Sequential (Sıralı) Model Doğruluk Grafiği	31

TABLÖLAR

Tablo 1.1	:	2018 ve 2019 Yıllarına ait Türkiye’deki Bölgelerin Kaçak Tüketim Oranları	2
Tablo 2.1	:	Türkiye'deki Bölgelere göre Kayıp Miktarları	4
Tablo 2.2	:	Güneydoğu Anadolu Bölgesine ait Kayıp Miktarları (EPDK)	5
Tablo 2.3	:	Doğu Anadolu Bölgesine ait Kayıp Miktarları (EPDK)	5



SİMGELER LİSTESİ

A	:	Min – Mak Kümesi	11
$\hat{a}$	:	Küme Aralığı	12
μ	:	Ortalama	12
x_1, x_2	:	Giriş Katman Değerleri	15
y	:	Çıkış Katman Değeri	15
A	:	LSTM Katmanı	16
h_{t-1}	:	LSTM Giriş 1	17
h_t	:	LSTM Çıkış Katmanı	17
x_t	:	LSTM Giriş 2	17
z	:	Güncelleme Kapısı	17
w	:	Katman Ağırlığı	17
z	:	Güncelleme Kapısı	17
w	:	Katman Ağırlığı	17
h	:	Aktivasyon Formülü	17
r	:	Sıfırlama Kapısı	17
z_t	:	t Anında Aktivasyon Formülü	17
h_t	:	t Anında Sıfırlama Kapısı	17
r_t	:	t Anında Sıfırlama Kapısı	17
$sigm$	:	Sigmoid Aktivasyon	17
$tanh$	:	Tanh Aktivasyon	17

GENEL BİLGİLER

Üniversite	: İstanbul Kültür Üniversitesi
Enstitü	: Lisansüstü Eğitim Enstitüsü
Dalı	: Bilgisayar Mühendisliği
Programı	: Bilgisayar Mühendisliği
Proje Danışmanı	: Dr. Öğr. Üyesi Fatma PATLAR AKBULUT
Proje Türü ve Tarihi	: Yüksek Lisans HAZİRAN 2022

ÖZET

GERÇEK ZAMANLI KAÇAK ELEKTRİK KULLANIMININ DERİN ÖĞRENME İLE TESPİT EDİLMESİ

Baran BEREKET YILDIZ

Teknolojinin gelişimiyle elektrik enerjisine duyulan ihtiyaç günden güne hızla artmaktadır. Elektrik enerjisine artan talep ile paralel olarak kaçak enerji tüketimleri de başta Güneydoğu Anadolu Bölgesi olmak üzere yaygın şekilde kullanılmaya devam etmektedir. Bu durumun sonucunda literatürde kaçak elektrik tüketim tespiti analizlerine olan ihtiyaç artmaktadır.

Bu çalışmada abonelere ait gerçek zamanlı kaçak elektrik tüketim durumu olan analiz başarı sayısı ve gerçek zamanlı anlık olarak elde edilen bir çok değişken ile derin öğrenme kütüphanelerinden biri olan Keras kütüphanesi Sıralı model kullanılarak eğitilmiş olup uygun hiper parameterler ile katmanlara ayrılarak açıklanmıştır. Proje sonucunda abonelerin kaçak tüketim gerçekleştirip gerçekleştirmediği durumu en yüksek doğru sınıflandırma ile oluşturulmuştur.

Anahtar Sözcükler: Kaçak elektrik, derin öğrenme, Keras kütüphanesi, gerçek zamanlı modelleme

University	:	İstanbul Kültür University
Institute	:	Postgraduate Education
Department	:	Computer Engineering
Programme	:	Computer Engineering
Supervisor	:	Assoc. Prof. Fatma PATLAR
		AKBULUT
Degree Awarded and Date	:	MASTER DEGREE JUN 2022

ABSTRACT

REAL TIME DETECTION OF ILLEGAL ELECTRICITY USAGE WITH DEEP LEARNING

Baran BEREKET YILDIZ

With the development of technology, the need for electrical energy is increasing day by day. In parallel with the increasing demand for electrical energy, illegal energy consumption continues to be widely used, especially in the Southeastern Anatolia Region. As a result of this situation, the need for leakage electricity consumption detection analyzes is increasing in the literature.

In this study, Keras library, which is one of the deep learning libraries with the number of analysis successes and many variables, which is the leakage electricity consumption status of the subscribers, is trained using the Sequential model and is explained by dividing into layers with the appropriate hyper parameters. As a result of the project, the status of whether or not the subscribers realized illegal consumption was created with the highest correct classification.

Keywords: Leakage electricity, deep learning, Keras library, real-time modeling

ÖNSÖZ

Tez yazımı süresince değerli bilgilerini benimle paylaşan danışmanım Sayın Dr. Öğr. Üyesi Fatma PATLAR AKBULUT Hocam’a minnet duyar ve teşekkürü bir borç bilirim. Yüksek lisans öğrenimim boyunca yeni teknolojilere adapte olmamı sağlayan bölüm başkanımız Sayın Doç. Dr. Akhan AKBULUT Hocam’a ve diğer bölüm hocalarıma minnet duyarım.

Ayrıca eğitim ve öğrenim hayatım boyunca maddi ve manevi katkılarını benden hiçbir zaman esirgemeyen, daima en büyük destekçilerim olan aileme ve çalışmış olduğum Dicle Elektrik Dağıtım A.Ş. teşekkürlerimi sunarım.

Baran BEREKET YILDIZ

HAZİRAN, 2022

1. GİRİŞ

İnsan yaşamının temel ihtiyaçlarından biri olan elektrik enerjisi, teknolojinin gelişimi ve kullanım alanlarının artmasıyla günlük hayatın kaçınılmaz unsurları arasında yer almıştır. Özellikle gelişmekte olan ülkelerde nüfus artışı, sanayileşme gibi faktörlerin etkisiyle beraber elektrik enerjisine duyulan ihtiyaç daha yoğun hale gelmiştir. İnsan hayatında önemli bir yer edinen bu enerji çeşidinin, üretim ve tüketim aşamaları yer almaktadır. Elektrik enerjisinin söz konusu üretim ve tüketim aşamaları esnasında meydana gelen bazı sorunlar da yer alabilmektedir. Elektrik enerji sistemlerinde meydana gelen bu sorunlar enerji kaybı olarak adlandırılır [1].

Enerji kayıpları, teknik ve teknik olmayan kayıplar olarak iki kısma ayrılır. Teknik kayıplar, elektrik enerji sistemlerinde kullanılan teknik malzeme ve donanımlar sonucu oluşan kayıplardır. Bu kayıplar, enerji üretiminin yapıldığı santrallerden başlayarak iletim ve dağıtım hatlarından tüketiciye ulaşıncaya kadar sürmektedir. Teknik olmayan kayıplar ise tüketiciden kaynaklanan kaçak tüketim olarak adlandırılır. Bu kaçak tüketimlerin başlıca nedenleri; sayaçlara yapılan müdahaleler ile kaçak enerji kullanımı, bilinçsiz enerji tüketimi ve dağıtım şirketlerinde enerjiyi faturalandırma sırasında yapılan hatalar olarak gösterilebilir.

Tüketici kaynaklı olan kaçak elektrik kullanımı, gün geçtikçe artmaya devam etmektedir. Bu tür kullanımların ülkemiz ve Dünya açısından sebep olduğu ekonomik zararlar önemli maliyetlere sebep olmuştur. Dünya’da kaçak elektrik tüketiminin ülkelere göre oranları Finlandiya’da 2018 yılında %3,3, Hırvatistan’da %7,7, Hollanda’da %4,4, İrlanda’da %6,7 ve Yunanistan’da %9,3 olarak yayımlanmıştır. Yapılan araştırmalarda Türkiye’de ki kaçak tüketim oranı ise %12,7 olarak tespit edilmiştir [2]. Türkiye’de 2018 ve 2019 yılına ait kaçak tüketim oranları Tablo 1’de yer almaktadır [3].

Tablo 1.1: 2018 ve 2019 Yıllarına ait Türkiye’deki Bölgelerin Kaçak Tüketim Oranları

Dağıtım Şirketi	GERÇEKLEŞEN			HEDEF	HEDEF- GERÇEKLEŞEN
					KAYIP ORANI
					FARKI
	2018	2019	2018-2019 Değişim	2019	2019
DİCLE	54,94	51,32	3,62	65,99	14,67
VANGÖLÜ	49,16	47,56	1,6	54,74	7,18
ARAS	23,55	21,64	1,91	24,97	3,33
ÇAMLIBEL	5,08	4,75	0,33	7,31	2,56
TRAKYA	4,37	4,49	-0,12	6,87	2,38
İST.AND.YAK.	6,04	5,05	0,99	7,31	2,26
ULUDAĞ	4,2	4,82	-0,62	6,73	1,91
BAŞKENT	6,11	5,89	0,22	7,33	1,44
AKDENİZ	5,78	5,91	-0,13	7,31	1,4
SAKARYA	6,52	5,96	0,56	7,28	1,32
ÇORUH	7,85	7,38	0,47	8,68	1,3
ADM	5,53	5,63	-0,1	6,72	1,09
MERAM	6,69	6,22	0,47	7,3	1,08
KAYSERİ VE CİVARI	6,57	6,02	0,55	6,94	0,92
OSMANGAZİ	6,41	6,61	-0,2	7,43	0,82
YEŞİLIRMAK	7,63	7,13	0,5	7,94	0,81
AKEDAŞ	5,51	6,46	-0,95	7,08	0,62
FIRAT	10,32	9,93	0,39	10,53	0,6
GEDİZ	6,63	7,55	-0,92	7,66	0,11
TOROSLAR	11,85	11,76	0,09	11,74	-0,02

Ekonomik ve sosyal olarak bir çok kayba neden olan kaçak elektrik kullanımlarının önüne geçilebilmesi açısından literatürde bir çok analiz ve yöntem geliştirilmiştir. Bu yöntemlerin başında derin öğrenme yöntemi yer almaktadır. Derin öğrenme yöntemi, veri seti sonuçlarının tahminlenmesini sağlayan ve birden fazla katman ile veri setine ait özelliklerin öğrenimini sağlayan bir yöntem olarak kullanılmaktadır [4]. Bunun yanısıra yapay zekaya kıyasla çok fazla değişken olması durumunda dahi yapının katmanlar ile daha fazla öğretilmesini sağlayarak güçlü bir sınıflandırma yapabilmeyi sunmaktadır. Derin öğrenmede katmanlı yapının yüksek doğruluklu bir sonuç üretebilmesi verimli veri ön işleme adımları sonucunda oluşmaktadır.

Önerilen bu çalışma kapsamında, Dünya ve Türkiye ekonomisinde kayıplara yol açan ve giderek artmakta olan kayıp kaçak elektrik oranlarının azaltılması ve buna yönelik olarak uygulanan derin öğrenme yöntemleri ile gerçek zamanlı kaçak elektrik tüketiminin tespiti yer almaktadır.

Bu çalışmanın ikinci bölümünde, çalışmaya ait yararlanılan kaynakların literatür taraması yer almaktadır.

Üçüncü bölümde, kaçak elektrik tüketiminin, Türkiye ve Dünya'daki son durumu ele alınmıştır.

Dördüncü bölümde, analizde kullanılan değişkenlerin derin öğrenmede en iyi sonucu vermesi için yapılması gereken veri ön işleme süreci ele alınmıştır. Bu süreçte ilk olarak kullanılan yöntemlerden biri olan eksik gözlem analizi, Label Encoding, veri tiplerinin kontrol edilmesi, Shapiro Wilk normallik testi, Varsayım ve Ki kare testi uygulanmıştır.

Beşinci bölümde, abonelere ait tüketimlerin kaçak veya kaçak olmaması analizi derin öğrenme ile ele alınmıştır. Derin öğrenme yöntemi kullanılırken oluşturulan katmanların hızlı ve verimli açıklanabilirliği açısından ele alınan Keras yöntemi kullanılmıştır.

Altıncı bölümde, Derin öğrenme ile gerçek zamanlı kaçak elektrik tüketimi analizi yapılarak sonuçlandırılmıştır.

Sonuç bölümü olan altıncı bölümde, geliştirilen yöntemlerin literatürden farklı olarak gerçek zamanlı değişkenler ile abonelerin kaçak elektrik tüketimlerinin Keras kütüphanesine ait Sıralı modeli ile hangi sınıfa ait olduğu sonucuna varılmıştır.

2. MATERİYAL VE METOD

2.1. Türkiye’de ve Dünya’da Kaçak Elektrik Kullanımı

Ülkemiz ve Dünya’da elektrik tüketimlerinde yaşanan kayıp – kaçak miktarlarının azımsanmayacak düzeyde olması Dünya ve Türkiye ekonomisinde kayıplara sebep olmaktadır [30]. Elektrik tüketimindeki kayıp – kaçak miktarlarının, tüketici faturalarına yansıtılmasıyla da bir çok problem meydana gelmektedir. Dünya üzerinde belirlenmiş kayıp – kaçak oranı %7 olarak belirlenen bu oran detaylı olarak bölgelere göre ayrıldığında şu sonuçlar Tablo 2.1’de gösterilmiştir [3].

Tablo 2.1: Türkiye’deki Bölgelere göre Kayıp Miktarları

BÖLGE İSİMLERİ	Kayıp Miktarı (MWh)
İÇ ANADOLU BÖLGESİ	1.875.533,00
GÜNEYDOĞU ANADOLU BÖLGESİ	6.378.530,46
DOĞU ANADOLU BÖLGESİ	2.140.649,89
MARMARA BÖLGESİ	5.038.984,46
EGE BÖLGESİ	2.367.520,18
KARADENİZ BÖLGESİ	1.043.230,58
AKDENİZ BÖLGESİ	3.040.994,48
TOPLAM	21.885.443,05

Enerji Piyasaları Denetleme Kurulu (EPDK) verilerine göre kayıp miktarı 21.885.443,05 (MWh) olan ve yüzdelik olarak kayıp oranının %10.49 civarında olduğu ve Dünya da ise bu oranın %7-8 civarında olduğu bu nedenle ülke genellemesinde çok yüksek kayıp oranının bulunmadığı açıklanmakta olsa da ekonomik anlamda olumsuzluk yaratmaktadır. Bölgesel toplam tutarların illere indirgenmiş hali Tablo 2.2’de gösterilmektedir [3].

Tablo 2.2: Güneydoğu Anadolu Bölgesine ait Kayıp Miktarları (EPDK)

Güneydoğu Anadolu Bölgesi Bağlı illeri	Faturalanan Tüketimin İllere Göre Dağılımı (MWh)	Bağlı Olduğu Dağıtım Şirketi	Kayıp Oranı (%)	Kayıp Miktarı (MWh)
GAZİANTEP	7.957.756,26	TOROSLAR	11,76	935.832,14
KİLİS	304.252,72	TOROSLAR	11,76	35.780,12
DIYARBAKIR	2.596.430,55	DİCLE	51,32	1.332.488,16
MARDİN	1.731.907,03	DİCLE	51,32	888.814,69
BATMAN	876.769,16	DİCLE	51,32	449.957,93
SİİRT	491.799,77	DİCLE	51,32	252.391,64
ŞANLIURFA	4.696.184,10	DİCLE	51,32	2.410.081,68
ADİYAMAN	1.132.880,80	AKEDAŞ	6,46	73.184,10
TOPLAM	19.787.980,39			6.378.530,46

Tablo 2.2 de görüldüğü gibi 19.878.980,39 (MWh) miktarında faturalanan tüketimden kayıp miktarı 6.378.530,46 (MWh) miktarında bir kayıp miktarı söz konusu olmaktadır [3].

Tablo 2.3: Doğu Anadolu Bölgesine ait Kayıp Miktarları (EPDK)

Doğu Anadolu Bölgesi Bağlı illeri	Faturalanan Tüketimin İllere Göre Dağılımı (MWh)	Bağlı Olduğu Dağıtım Şirketi	Kayıp Oranı (%)	Kayıp Miktarı (MWh)
MALATYA	1.630.978,43	FIRAT	9,93	161.956,16
ELAZIĞ	1.178.418,62	FIRAT	9,93	117.016,97
TUNCELİ	126.236,69	FIRAT	9,93	12.535,30
BİNGÖL	305.989,05	FIRAT	9,93	30.384,71
ARDAHAN	128.263,76	ARAS	21,64	27.756,28
ERZURUM	890.031,38	ARAS	21,64	192.602,79
ERZİNCAN	612.004,17	ARAS	21,64	132.437,70
KARS	308.077,29	ARAS	21,64	66.667,93
İĞDIR	199.596,15	ARAS	21,64	43.192,61
AĞRI	448.173,46	ARAS	21,64	96.984,74
MUŞ	391.385,07	VANGÖLÜ	47,56	186.142,74
BİTLİS	360.460,45	VANGÖLÜ	47,56	171.434,99
VAN	995.734,65	VANGÖLÜ	47,56	473.571,40
HAKKARİ	260.686,65	VANGÖLÜ	47,56	123.982,57
ŞIRNAK	592.328,55	DİCLE	51,32	303.983,01
TOPLAM	7.836.035,82			2.140.649,89

Tablo 2.3 de görüldüğü gibi 7.836.035,82 (MWh) miktarında faturalanan tüketimden kayıp miktarı 2.140.649,89 (MWh) miktarında bir kayıp miktarı söz konusu olmaktadır [3]. Sonuç olarak EPDK raporlarında da görüldüğü gibi Türkiye’de bölge ve illere ait kayıp kaçak elektrik tüketimleri teknolojinin de gelişimiyle azaltılmaya ve

kaçak tüketimlerin üstesinden gelmeye çalışılmaktadır. Bu durumlara çözüm bulmak amacıyla yapay zeka ile makine öğrenimi ve derin öğrenmeye yöntemleri geliştirilmektedir [30],[31].

2.2. Kaçak Elektrik Enerji Sistemlerinde Kullanılan Yöntemler

2.2.1. Veri Ön İşleme Yöntemleri

Teknolojik kaynakların artmasıyla büyüyen veri hacmi, kaliteli veri kullanımının önemini ortaya çıkarmıştır. Doğru, güvenilir veri elde edebilmek için veri ön işleme yöntemi ortaya çıkmıştır. Veri ön işleme, makine öğrenimi ve veri madenciliği gibi bir çok alanda model kurulma aşamasından önceki işlemleri ifade eden performans geliştirme ve düzeltme işlemleri olarak tanımlanmaktadır [5].

Doğru veri elde edebilmek için gerekli veri ön işleme adımları bir çok yöntem içermektedir. Literatürde yaygın yöntemler arasında veri temizleme, normalizasyon, veri dönüşümü, veri ağırlıklandırma, özellik çıkarımı ve seçme yöntemleri gibi birçok yöntem bulunmaktadır. Bu çalışmada kullanılan veri ön işleme yöntemleri arasından kayıp gözlem analizi, aykırı değer tespiti,, kategorik sınıf atama yöntemi (Label Encoding), Shapiro Wilk Normalik testi, varsayım ve Ki Kare yöntemleri kullanılmıştır [6].

2.2.2. Kayıp Gözlem Analizi

Bir çok alanda olduğu gibi elektrik alanında yapılan uygulamalar sonucu elde edilen verilerde mevcut sorunlar yer almaktadır. Kayıp gözlem sorunu da bu sorunlar arasında yer almaktadır. Kayıp gözlem sorunu, değişken veya girdi olarak adlandırılan veri seti özelliklerinin eksik, boş veri varlığını ifade etmektedir. Kayıp gözlemlerin bir çok sebebi olabilmektedir [8]. Bu durumlardan bazıları;

- Kayıp gözlem varlığının en önemli sorunu bilinmezlik sorunudur. Değişken içerisinde herhangi bir sebepten kaynaklı olarak gerçekleşebilmektedir. Bu durumlar veri girişinin unutulması veya kaybolmasından kaynaklanabilmektedir.

- Diğer bir sebep ise değişkenin sahip olduğu bir özelliğin bulunmamasından kaynaklı kayıp gözlem oluşabilmektedir.

yukarıdaki şekilde ele alınmıştır [6].

Doğru ve güvenilir veri seti elde edebilmek amacıyla literatürde bir çok kayıp gözlem analizi yer almaktadır. Bu çalışmada belirtilen sebeplerden kaynaklanan kayıp gözlem analizi bazı yöntemler ile giderilmiştir.

2.2.2.1. Ortalama – Ortanca – Mod eksik veri işleme yöntemleri

Veri setinde bilinmeyen veya istenmeyen özelliklerden kaynaklı olarak meydana gelen kayıp gözlem verilerini işlemek için bir çok yöntem kullanılmaktadır. Bunlardan biri olan ortalama, ortanca, mod ile kayıp veri giderme işlemleridir [5].

Ortalama, ortanca, mod ile kayıp veri giderme işlemi, istatistiksel metodlar kullanılarak yapılan bir veri ön işleme yöntemidir. Bu yöntemde veri setinde bulunan tüm değerler analize dahil edilmektedir. Ortalama ile işlenen kayıp veriler, her zaman doğru sonuçlar vermemektedir. Bunun sebebi, istatistiksel ortalama değer veri setinin ana yapısını teösil etmektedir. Kayıp değerlerin varlığı ise mevcutta bulunan veri yapısını bozma eğilimindedir. Bu durumda değişkenlerin ortalama değer ile doldurulması doğru sonuçlar doğurmamaktadır. Bu durum mod yönteminde de aynı sonuçlar doğurmaktadır. Değişkenler arasındaki ilişki ve bağımlılığın bilinmemesi güçlü tahmin oranlarını da düşürmektedir. Sonuç olarak, ortalama ve mod ile kayıp gözlem doldurma işlemleri gerçek veri setlerinde doğru sonuçlar doğurmamaktadır [8].

Genellikle gerçek hayat çalışmalarında yapılan analizlerde, çeşitli sebeplerden kaynaklanan kayıp gözlem verileri bulunduğu değişken yapısını bozmaktadır. Bozulmanın meydana geldiği değişken yapılarında ortalama ve mod ile işleme yönteminin uygun olmadığı gözlemlenmiştir. Ortanca ile kayıp veri işleme yönteminde ise veri setinin çarpıklaşma durumundan etkilenme derecesinin ortalama ve mod değerlerine göre daha düşük miktarda olduğu sonucuna varılmaktadır [8].

2.2.3. Aykırı Değer Belirleme Yöntemi

Uygulamalı veri setlerinde sıklıkla görülen aykırı değerler, veri setlerinin genel yapısından farklı davranışlar sergileyen gözlemlerdir. Bu değerler tüm veri setini etkileyen değerlerdir. Özellikle makine öğreniminde doğru analiz yapılabilmesi için bu anomalliklerin giderilmesi doğru bir analiz sonucunu ortaya çıkarmaktadır [9].

Aykırı Değerler, aynı zamanda istatistiksel analizlerde sorunlara neden olabilecek diğer gözlemlerden uzak olan bir gözlem noktası olarak da tanımlanmaktadır. Doğru bir analiz yapabilmek amacıyla gerçekleştirilen ve veri ön işleme adımlarından biri olan aykırı değer analizi, veri yapısına göre değişkenlik göstermektedir [10]. Bu yöntemlerin bazıları şu şekildedir:

- Mesafe tabanlı algoritmik veri setlerinde, nesneler arasındaki uzaklıkların geometrik yorumlaması ile hesaplanmaktadır [12].
- Yoğunluk baz alınarak tahminlenen aykırı değerler, düşük yoğunluğa uygun hale getirilerek veri setine dahil edilmektedir [12].
- Minimum kovaryans determinant yöntemi, aykırı değerlerin konum ve şeklinin tahminini sağlayan aykırı değer algoritması olarak tanımlanmaktadır. Bu algoritmaya göre aykırı olan noktalar hesaplamalara dahil edilmemektedir [11].
- Aykırı değer analizinde kullanılan bir diğer yöntem ise tek sınıfa sahip sınıflandırma algoritmasıdır (one class svm). Tek sınıflı sınıflandırma algoritmaları, aykırı değerlerin azınlık olarak yer aldığı uç ile aykırı değerlerin yoğunluk olarak yer aldığı bir uç kısmı ayırabilen bir yapı sağlamaktadır. Bu algoritmanın diğer algoritmalarından farkı, test edilecek yapının yanlış sınıflandırma olasılığını en aza indirebilmesidir [9].

2.2.4. Değişken Tipleri ve Dönüşüm Yöntemleri

2.2.4.1. Değişken tipleri

Veri tipi ve dönüşümleri, veri değişkenlerinin türünü veya dağılımını değiştirmek için kullanılan veri ön işleme yöntemlerinden biridir. Bu, yöntemler her bir veri yapısı için farklı teknikler ile elde edilmektedir [13].

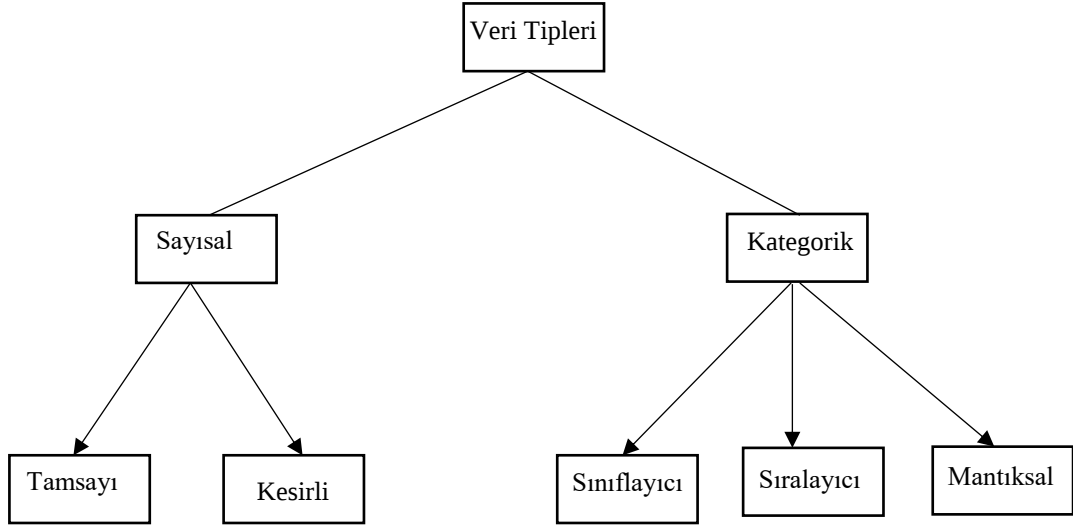
Bir çok alanda kullanıldığı gibi makine öğrenmesinde kullanılan veri setlerinde de genel olarak sayısal veya kategorik türde veri yapıları bulunmaktadır. Sayısal veri yapısına sahip değişkenlerin tamsayı ve yüzdellik yapıda iki türü bulunmaktadır. Kategorik veri türünde, sınıflayıcı(nominal), sıralı(ordinal), cebirsel(boole) yapıda alt türler bulunmaktadır. Sayısal veri türleri aşağıdaki şekilde yer almaktadır [13].

- Tamsayı: Ondalık kısmı bulunmayan sayılar olarak adlandırılır.
- Yüzdellik: Kesirli sayı değerleri mevcut olan sayılardır.

Kategorik Veri türleri ise;

- Sıralayıcı (Ordinal) Veri Tipi: Sıralı sınıf etiketleri içeren veri tipleridir.
- Sınıflayıcı (Nominal) Veri Tipi: Sınıf etiketleri bir sıralama içermeyen veri tiplerini içermektedir.
- Cebirsel (Boolean) Veri Tipi: Doğru ve yanlış şeklindeki mantıksal ifadeler içeren veri tiplerini ifade etmektedir.

Analiz aşamasının verimli bir şekilde gerçekleşebilmesi için doğru bir ön işleme aşamasının önemli bir kısmını oluşturan veri tiplerinin kontrolü ele alınmıştır. Bu işlemlere ait veri tiplerinin özeti Şekil 2.1’de gösterilmektedir.



Şekil 2.1: Değişken Veri Tipleri

2.2.4.2. Veri dönüşümleri

Veri dönüştürme, işlenmemiş verilerin analiz edilebilmesi ve anlaşılabilir forma dönüştürülmesinde önemli bir rol oynamaktadır. Dönüşüm ile elde edilen yeni yapılar ile doğru sınıflayıcıların oluşturulması sağlanmaktadır. Aynı zamanda dönüşümler ile kavramların daha iyi öğrenilmesi sağlanır [14].

Veri dönüşüm işlemlerinin gerçekleştirilmesi amacıyla ele alınan bazı yöntemler bulunmaktadır. Bu yöntemler [13]:

- Ayrıklaştırma dönüşümü (Discretization Transform): Sayısal bir değişkenin sıralı bir değişkene dönüşümünü sağlamaktadır.
- Sıralı Dönüşüm (Ordinal Transform): Kategorik değişkenin bir tamsayı değişkenine dönüşümünün sağlandığı bir veri dönüşüm yöntemidir.
- Tek Dönüşüm (One Hot Encoding): Makine öğrenmesinde sıklıkla kullanılan kategorik değişkenlerin ikili sınıf grubuna (0,1) dönüştürülmesini sağlayan bir yöntem olarak kullanılmaktadır. Bu yöntem kategorik verilerin sebep olduğu karmaşıklığı da ortadan kaldırarak kirli veri sorununu minimize etmektedir.

2.2.5. Veri Normalleştirme İşlemleri

Veri normalleştirme, sistem algoritmalarını besleyerek veri kalitesini artırmayı amaçlayan bir yöntemdir. Veriler arasındaki sapmayı en aza indirerek verileri standart bir değer aralığına dönüştürmektedir. Bu dönüşüm 0 ile 1 veya -1 ile 1 gibi belirli aralıklarda alma işlemi olarak da bilinmektedir. Bu ölçekleme yöntemleri aykırı değer sayısının minimum seviyede olduğu durumlarda daha kullanışlı olmaktadır [18].

Normalleştirme işleminde kullanılan değişkenlerin aynı ölçek yapısına sahip olması makine öğrenmesi gibi yöntemlerde yer alan eğitim sürecini hızlandırmaktadır. Aynı zamanda farklı ölçeklere sahip değişkenlerde ise modelleme uygulamalarında yarar sağlamaktadır [15]. Literatürde normalleştirme yöntemine ait farklı teknikler yer almaktadır. Bu çalışmada kullanılan normalleştirme teknikleri ise min- mak normalizasyonu, Z skoru normalizasyonu şeklindedir.

2.2.5.1. Min - mak normalizasyonu

Min-mak normalizasyonu, [0,1] veya [-1,1] aralığında tanımlayarak normalleştirme işlemi gerçekleştirmektedir. Bu yöntem minimum ve maksimum değer aralığında bir A kümesi oluşturmaktadır. Oluşan bu A küme aralığı ise $\hat{a}$ ile temsil edilmektedir. Min- mak normalizasyon yöntemi ise aşağıdaki şekilde gösterilmektedir [15]:

$$\hat{a} = \min + \frac{(\max - \min) * (a - \min A)}{(\max A - \min A)} \quad (2.1)$$

Min-mak normalizasyon yönteminin dezavantajı da bulunmaktadır. Ölçeklendirme ile önemli veri kayıpları gerçekleşebilmektedir. Bu çalışmada ver setlerinin normalleşmesinde bazı değişkenler üzerinde avantaj sağladığı gözlemlenmiştir [18].

2.2.5.2. Z-Skoru Yöntemi

Z skoru ölçekleme yöntemi, normal dağılan veri setlerinde kullanışlı bir yöntem olarak kullanılmaktadır. Normal dağılmayan veri setlerinde gerekli dönüşümler ile doğru bir yöntem olarak kullanılabilir. Z skoru ölçekleme yöntemi, verileri ortalaması 0 ve standart sapma değeri 1 olan veri yapısına dönüştürmektedir [32]. Normalleştirilmiş A değerinin standartlaştırılmış bir $\hat{a}$ formülü ise:

$$\hat{a} = \frac{a - \mu(a)}{\sigma(a)} \quad (2.2)$$

şeklindedir. Bu yöntem bilinmeyen en küçük ve en büyük değerlerin olduğu durumlarda kullanışlı olmaktadır. Bu yöntem zaman serisi gibi durağanlığın olmadığı durumlarda iyi sonuçlar vermemektedir. Normale yakın veri setlerinde daha iyi sonuçlar elde edebilmek için bu yöntem avantaj sunmaktadır [33].

2.2.6. Varsayım ve Ki Kare Testi

Hipotez testi kavramı, ilk olarak 1900 yılında Karl Pearson tarafından ortaya atılmıştır. Aynı zamanda bağımsızlık testi olarak bilinen Ki-Kare testini ortaya çıkararak İstatistik bilimine önemli bir yöntem kazandırmıştır. Pearson Ki-Kare testi, normal dağılmayan veri setlerinde dahi test edilme ve istatistiki yorumlamaya olanak sağlamaktadır [17]. Ki-Kare testi, iki durum söz konusu olduğunda kullanılabilir. Bu durumlar aşağıda yer almaktadır:

1. İki veya daha fazla grup, kitle veya kriter arasında ilişki olup olmadığı durumlarını hipotez test etmek amacıyla kullanılmaktadır (Değişkenler arasındaki bağımsızlığı test etmektedir.)
2. Gözlemlenen verilere ait dağılımın beklenen dağılım ile ne derece uygun olduğu Ki-kare testi ile test edilebilmektedir.

Gerçek hayat verilerinde genellikle veriler normal dağılmamaktadır. Bu sayede Ki-kare testi ile normal dağılmayan veriler arasındaki ilişkilerin anlamlı olup olmadığı test edilebilmektedir [17].

2.3. Derin Öğrenme

2.3.1. Derin Öğrenme Mimarileri

Artan ve karmaşıklaşan veri yapılarını temsil etme amacıyla ortaya çıkan ve belirli insan davranışlarını taklit etmeye yarayan bir yapay zekâ ve makine öğrenme yöntemi olarak ortaya atılmıştır. Derin öğrenme sayesinde veriler üzerinde yapılan her bir işlem deneyim ile öğretim işlemlerine dayanmaktadır [19].

Derin öğrenme, tahminlemeye dayalı modelleme işlemlerinde sıklıkla kullanılmaktadır. Klasik tahminleme modellerinde kullanılan algoritmalar doğrusal iken derin öğrenme yöntemlerinde doğrusal olmayan, parametrik, karmaşık algoritmalar çözümlenebilmektedir. Derin öğrenme yöntemleri, ham girdi verilerinin referans alınmasıyla diğer katmanlardaki üst düzey bilgilerin çözümlenmesini sinir ağı algoritmalarının katmanlarını kullanarak gerçekleştirmektedir [20].

Sinir ağı algoritmaları, insan beyninden esinlenerek veri modellerini tanımlamak için tasarlanmıştır. Sinir ağları benzerliklere göre verileri kümeleyerek verilerdeki etiketlere göre sınıflandırma yapmaktadır. Buna bağlı olarak verilerden özellikler elde etmektedir. Ağlar ile tanınan nicel değişkenler ise, gerçek uygulama verilerinde vektörler olarak tanımlanmaktadır. Bu verilerin açıklanabilirliği için derin öğrenme yöntemlerinin mimari yapıları bulunmaktadır [39].

Ivakhnenko ve Lapa tarafından 1965 yılında yapılmış olan temel derin öğrenme mimarisinde, her katmanda en iyi özellikler istatistiksel yöntemlerle seçilip bir sonraki katmana iletilmektedir. Ağların eğitmek için geri yayılım (backpropagation) kullanılmayıp önceki katmanlardan sonraki katmanlara en küçük kareler yöntemi kullanılmıştır [21]. İlk derin öğrenme mimari yapısı örnek alınarak geliştirilen bir çok yapı bulunmaktadır. Canlı yapılar önek alınarak oluşturulan kendiliğinden gerçekleştirilen “denetimsiz öğrenme” yapı mimarisi yapılmıştır. İlk başarılı derin sinir ağı uygulaması ise ve Yann LeCun ile arkadaşları tarafından posta kutusu yazıları üzerinde geliştirmişlerdir. Bir diğer örnek ise 1997 yılında Hochreiter ve Schmidhuber'in ortaya çıkardığı tekrarlayan sinir ağları (recurrent neural network) için Uzun Kısa Vadeli Bellek (Long Short-Term Memory) yöntemidir. Teknolojinin henüz gelişmemiş bu dönemlerinde derin öğrenmeye bağlı yapay sinir

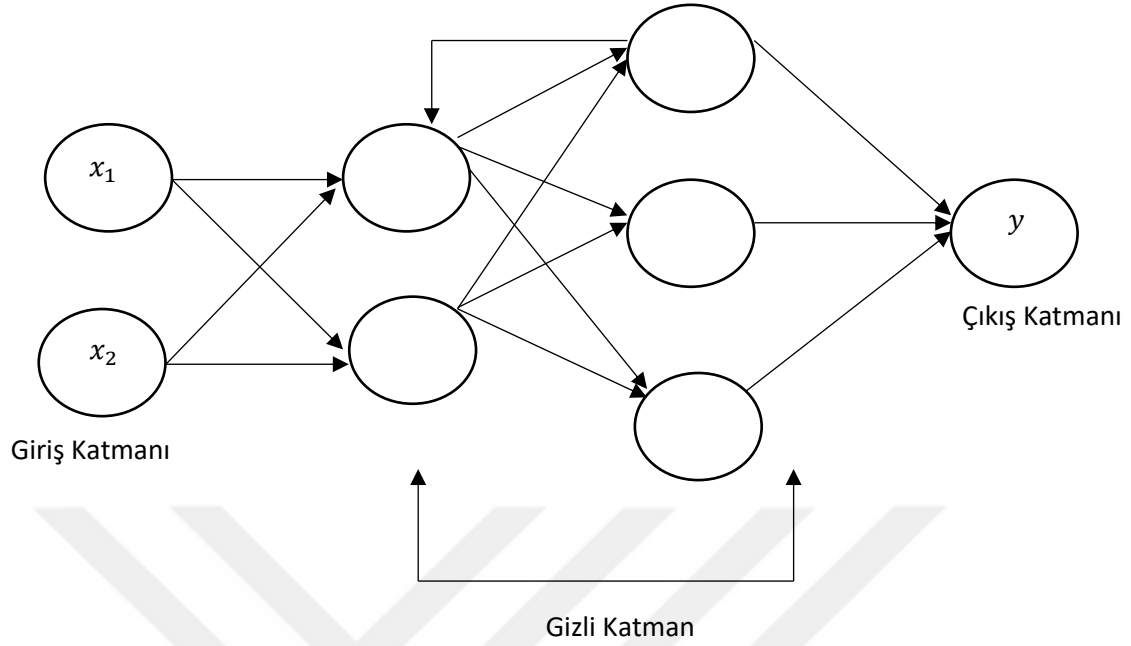
ağları algoritmaları bir çok avantaj bulundurmasına rağmen hesaplama zorluğundan dolayı kullanılamamış ve eğitim süreçlerinde kalmış olup uygulamaya geçirilememiştir. Teknolojinin gelişimi, bilgisayarların çalışma hızlarının artmasıyla ve grafik işlemci birimleri (GPU) kullanımıyla yapay sinir ağı hesaplamalarında büyük gelişimler meydana gelmiştir [22].

2.3.1.1. Konvolüsyonel - Evrişimli sinir ağı (Convolutional neural network - CNN)

Konvolüsyonel Sinir Ağları, (Convolution Neural Network-CNN) çok katmanlı algılayıcı türlerinden biridir. Sınıflandırma, doğal dil işleme, öneri sistemleri, görüntü ve video tanıma gibi görüntü tekniklerinde kullanılan derin öğrenme yöntemidir. Bu yöntem, girdi görüntüsü olarak görüntüdeki çeşitli nesnelere ağırlık atayıp görüntüler arası ayrımı sağlayan bir yöntem olarak kullanılmaktadır. CNN, bir veya daha fazla konvolüsyonel katman, alt örnekleme katmanı ve standart çok katmanlı sinir ağı gibi katmanlara sahiptir. Bu yönleri ile diğer sinir ağlarına nazaran daha az parametre ve eğitime bağlı olarak çalışabilmektedir [23].

2.3.1.2. Tekrarlayan sinir ağı (Recurrent neural network)

Tekrarlayan sinir ağı (RNN), birimler ve düğümler arasındaki bağların zamansal bir döngü boyunca yönlendirilmiş ve yönlendirilmemiş bir grafik oluşturduğu yapay sinir ağıdır. RNN, bir veri kümesinin sahip olduğu sıralı özellikleri tanımak ve meydana gelebilecek olası durumları tahmin etmek için tasarlanmış sinir ağı yöntemidir. İleri beslemeli sinir ağlarından türetilen tekrarlayan sinir ağları, diğer ağlardan farklı olarak girdi belleklerinin rasgele dizilerini işlemek için kullanılmaktadır. Bu özelliklerden kaynaklı olarak çoğunlukla zaman serileri, doğal dil işleme gibi sıralı verileri işlemede kullanılmaktadır. Tekrarlayan sinir ağı yapısı Şekil 2.2’de gösterilmektedir.



Şekil 2.2: Tekrarlayan Sinir Ağı Hücre Yapısı

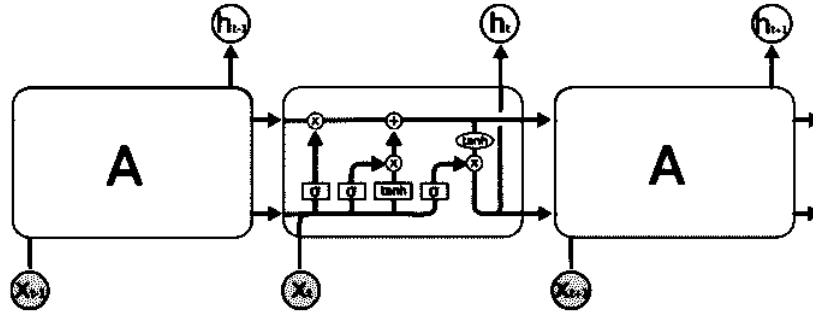
Diğer sinir ağı yapılarından farklı olarak giriş çıkış katmanlarının birbirlerinden bağımsız olma durumu bu ağı yapısında bulunmamaktadır. Tekrarlayan sinir ağı yapısında, gizli katman önceki katmanlarda bulunan yapıyı kullanıp bilgileri korumaktadır. RNN yapısında, önceki adım çıktılarından elde edilen ağırlık ve yanlılıklar tahminlenmek amacıyla bir sonraki katmanlara aktarılmaktadır. Meydana gelen bir dizi katman birleşerek tekrarlayan katman oluşturmaktadır. Tüm bu durumlar sonucunda tekrarlanan katmanlar tahminledikleri bilgileri hafızada tutarak çıktı bilgisi vermektedir.

2.3.1.3. Uzun kısa vadeli hafıza ağı (Long – short term memory network)

Uzun kısa süreli bellek ağı (LSTM), uzun süreli bağımlılıkları öğrenebilen ve bağımlılık sorununu çözmek için tasarlanmış bir ağı yapısıdır. LSTM yapıları, tekrarlanan sinir ağıları gibi varsayılan davranışları uzun süre bellekte tutmaktadır. Klasik RNN yapılarında yinelenen modül, tek bir katman olup basit bir yapıya sahiptir. LSTM yapısında bulunan ve yinelenen modül ise farklı ve karmaşık bir yapı

içermektedir. Bir diğer farklılık ise tek bir sinir ağı katmanını yerine özel bir etkileşim yapısına sahip olan dört katman bulunmaktadır [25].

LSTM katman yapısı Şekil 2.3'te gösterilmektedir [24].

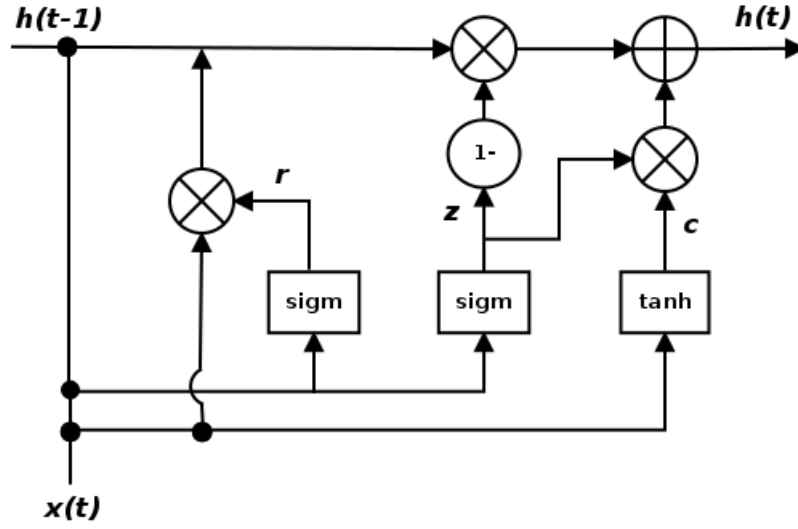


Şekil 2.3: LSTM Hücre Yapısı

LSTM yapısında satırlar, düğüm çıktısından diğer girdilere kadar varolan tüm vektörleri taşımaktadır. Daireler, noktasal işlemleri temsil etmektedir. Kare kutular ise, “sinir ağı katmanını öğrencileri” olarak adlandırılmaktadır. Yapı içerisindeki iki giriş ise h_{t-1} ve x_t ’dir. Hücre içerisindeki doğrusal çizgiler, küçük doğrusal etkileşimleri ifade etmekte olup doğruca çalışmaktadır. LSTM kapıları, hücre durumlarına bilgi ekleme veya çıkarma durumları ile bilgi geçişini sağlamaktadır. LSTM kapıları, sigmoid sinir ağı katmanından ve çarpımsal işlemlerden meydana gelmektedir. Sigmoid sinir katmanı, bileşenlerin geçme durumlarının yeterliliğini açıklayan ve $[0,1]$ aralığından gerçekleşen izin sistemi olarak adlandırılmaktadır. Üç kapı durumu, LSTM yapısını kontrol edip korumaktadır. LSTM ağlarının sahip olduğu bu yapı ile sorun çözme ve ayrıntılı değişken elde etme yöntemleri ile ön plana çıktığı ispatlanmıştır.

2.3.1.4. Kapı özyinelemeli geçitler (Gated recurrent units - GRU)

Klasik RNN mimarilerinde, kayıp gradyan probleminden kaynaklı olarak optimizasyon yaşanmaktadır. Bu durum uzun vadeli öğrenme bağımlılık öğrenimine engel olmaktadır. Bu sorunun çözülebilmesi için çarpımsal model olan kapı özyinelemeli geçit ağ mimarisi (GRU) geliştirilmiştir [27].



Şekil 2.4: GRU Hücre Yapısı

Şekil 2.4’ te ele alınan GRU yapısı, uzun kısa dönem ağ çeşidine benzer özellikler göstermektedir. GRU ağ yapısı LSTM yapısından farklı olarak gizli katman yapısını güncelleyerek daha az işlem gerektirmektedir. Bu durum sonucunda hesaplama kolaylığı sağlamaktadır [27]. GRU yapısında bulunan kapı formülleri aşağıda yer almaktadır.

Güncelleme (update) kapı formülü z;

$$z_t = \sigma (w_z x_t + U_z h_{t-1}) \quad (2.3)$$

t zamanında aktivasyon formül yapısı h;

$$h_t = (1 - z)h_{t-1} + z_t * h_t \quad (2.4)$$

Sıfırlama kapısı formül yapısı r;

$$r_t = \sigma (w_r x_t + U_r h_{t-1}) \quad (2.5)$$

ile temsil edilmektedir.

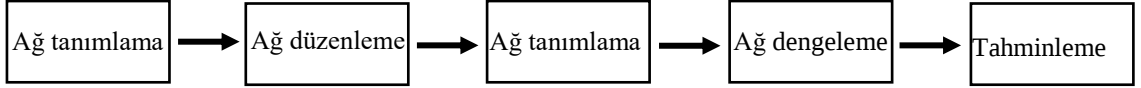
GRU ağ yapısında bulunan ve güncellenip eklenen her bir aday aktivasyonu, geçmiş aktivasyonlardan öğrenerek bilgi akışı düzenlenir. Bu durum diğer sinir yapılarına göre avantaj sağlamaktadır. Dahil edilen aktivasyon miktarı ise güncelleme kapısı ile belirlenmektedir. Sıfırlama kapısı ise mevcut bilgilerin ne kadarının unutulması gerektiğine karar vermek için kullanılmaktadır [27].

2.4. Keras ile Model Oluşturulması

Derin öğrenme yöntemleri ile teori ve uygulamada karmaşık sistem yapıları, sinir ağları yardımıyla katmanlara bölünerek çözülmek üzere tasarlanmıştır. Karmaşık sistem yapılarının oluşturduğu bazı durumlar sinir ağları ile de çözülememektedir. Bu tür sorunların varlığı derin öğrenme kütüphanesi olan Keras Kütüphanesini ortaya çıkarmaktadır. Google tarafından geliştirilmiş olan Keras, önemlilik seviyesi yüksek derin öğrenme API olarak bilinmektedir. Yagın olarak kullanılmakta olan Keras, hesaplama açısından da bir kolaylık sağlamaktadır. Keras kütüphanesine ait bazı özellikler aşağıda yer almaktadır [28].

- Yaygın olarak kullanılan çoğu sinir ağ modellerini destekleme özelliğine sahiptir.
- Modüller bir yapıya sahip olma özelliği esnek, yenilikçi kolay ulaşılabilir kütüphane özelliği katmaktadır.

Sinir ağ modellerini oluşturma da yarar sağlayan Keras Kütüphanesi bazı adımlar içermektedir. Bu adımlar Şekil 2.5'te gösterilmektedir[27]:



Şekil 2.5: Keras Adımları

1. Ağ tanımlama adımında sahip olunan model katmanları ve ağ bağlantı ilişkileri tanımlanmaktadır. Bu Keras ana modelleri sıralı ve fonksiyonel model olmak üzere ikiye ayrılmaktadır. Seçilen model ile veri akışı bu aşamada gerçekleşmektedir.
2. Belirlenen model yapısına uygun kod seçimi burada gerçekleşmektedir. Kod seçimi ile derin öğrenme ile uyumlu hale getirilir. Aynı zamanda model derlenmesi, model kayıp fonksiyonu, optimizasyon, model doğruluk metrikleri bu aşamada belirlenmektedir.
3. Derlenen model ile veriler eğitilmektedir.

4. Bu aşamada eğitilmiş modellerin hata oranları ve hata değerlendirme işlemleri değerlendirilmeye alınmaktadır.
5. Son aşama olan tahminleme işleme ile model tahmin işlemleri gerçekleştirilmektedir.

Bu aşamalar ile gerçekleştirilen model tanımlamanın iki çeşidi bulunmaktadır [27].

- Sıralı model (Sequential): Sinir ağı katmanlarını yığmak için kullanılan metoddur [34].
- Fonksiyonel API: Çok fazla çıktıya sahip modellerde veya karmaşık modellerde kullanılan model tanımlama yöntemidir [35].

2.4.1. Keras Model ile Metriklerin Belirlenmesi

Metrikler, model performans ölçümünü değerlendirmek amacıyla kullanılan işlevsel bir yöntemdir. Bu yöntem, bir çok farklı işlevi gerçekleştirmek için kullanılabilir.

Doğruluk metriği için kullanılan yöntem olan;

- Doğruluk sınıf metriği (Accuracy class metric): Model tahminlemelerinde atanan etiketlerin eşit olup olmadığını ne derece tutarlı olduğunu göstermektedir. İki değişken sınıfı tanımlayarak oluşturulan bu metrik, doğru eşleşme ve veri sıklığını (frekans) hesaplamayı sağlamaktadır [36].

Olasılık metriği için kullanılan yöntemler ise;

- İkili Çapraz Entropi Sınıfı (Binary Crossentropy Class): Modelde tahmin edilen değerler ile gerçek değerler arasındaki çapraz entropi kayıp değerlerini bulmak için kullanılmaktadır. Bu metrik sınıflandırma çeşidinde iki etiket sınıfı (0, 1) ile temsil edilmektedir [37].

Regresyon metriği için kullanılan yöntemler;

- Regresyon performansını ölçmek için skor, kayıp öğrenmek ve fayda sağlamak için kullanılan işlevsel bir yöntemdir. Aynı zamanda birden fazla çıktı durumunu sağlamak için kullanılmaktadır [29].

2.4.2. Aktivasyon ve Optimizasyon Fonksiyonlarının Belirlenmesi

Derin öğrenme yöntemlerinin modelleme aşamasında kullanılan aktivasyon fonksiyonları, var olan girdilerin bir sonraki nöron katmanından çıkan çıktı verileri üzerine uygulanan doğrusal olmayan işlevler bütünüdür. Bir diğer ifade ile “eşikleme kapısı” olarak bilinmektedir. Derin öğrenme yöntemleri için bazı aktivasyon fonksiyonları aşağıda ele alınmıştır [29], [45].

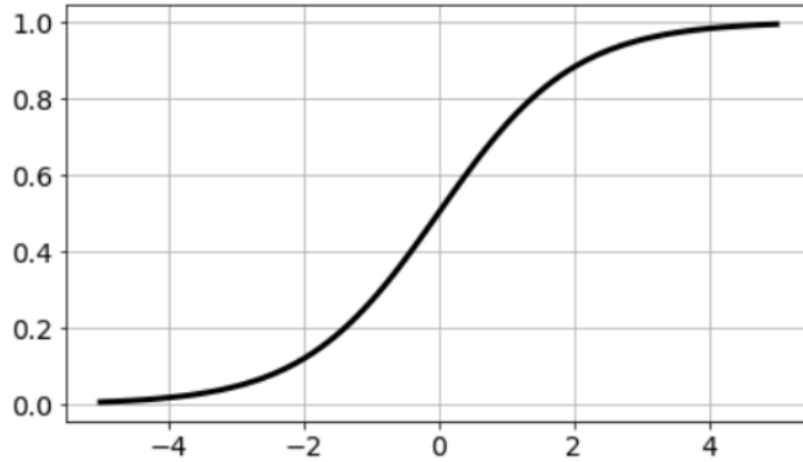
2.4.1.2. Sigmoid (Logistic - Lojistik) aktivasyon fonksiyonu

Girdi katmanı gerçek değerli sayılardan meydana gelen ve çıktı değerleri [0,1] aralığında bulunan doğrusal olmayan aktivasyon fonksiyonlarından biridir. Sigmoid fonksiyonunda, girdideki minimal değişiklikler çıktıda da minimal değişikliklere sebep olmaktadır. Bu tür durumlar adım fonksiyonlarından daha toleranslı sonuçlar ortaya çıkarmaktadır [40].

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.6)$$

$$f'(x) = \frac{\partial f(x)}{\partial x} = f(x)(1 - f(x)) \quad (2.7)$$

Sigmoid fonksiyonuna ait şekil, Şekil 2.6’da gösterilmektedir.



Şekil 2.6: Sigmoid Fonksiyonu

Sigmoid fonksiyonuna nazaran, az sayıda yineleme işlemiyle aynı performansı gösteren diğer aktivasyon fonksiyonlarının kullanımıyla daha az tercih edilmeye başlanmıştır [29], [44].

2.4.1.3. Tanh (HyperbolicTangent - Hiperbolik Tanjant) fonksiyonu

Sıfır merkezli tanh fonksiyonu, $[-1,1]$ aralığında yer alan doğrusal olmayan aktivasyon fonksiyonudur. Büyük miktarlarda pozitif ve negatif değerlerde görülmektedir [29], [41].

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.8)$$

$$f'(x) = \frac{\partial f(x)}{\partial x} = f(x)(1 - f(x))^2 \quad (2.9)$$

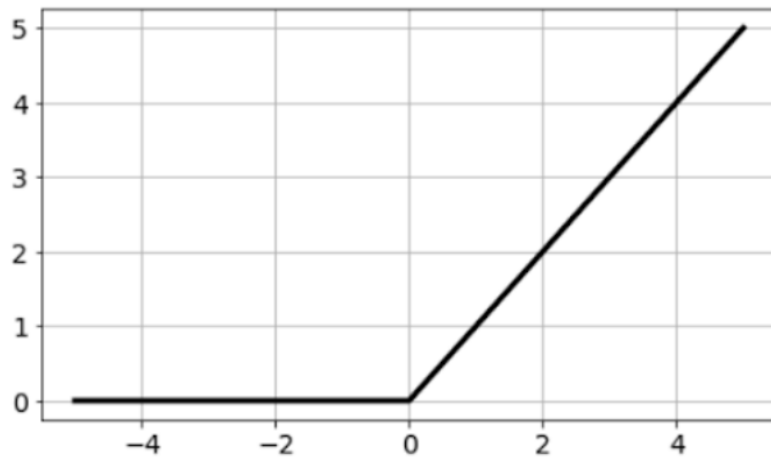
Tanh fonksiyonuna ait şekil, Şekil 2.7’de gösterilmektedir.



Şekil 2.7: Tanh Fonksiyonu

2.4.1.4. RELU (Rectified Linear Unit - Reaktifli Doğrusal Ünite) fonksiyonu

CNN (evrişimli sinir ağı) modellerinde çoğunlukla kullanılmakta olan RELU fonksiyonu, doğrusal olmayan aktivasyon fonksiyonudur. Bu aktivasyon fonksiyonunda pozitif değerli alanlarda istenilen doyum sağlanmamaktadır. Bunun sonucunda kayıp gradyan sorunundan kaçınılmaktadır [38]. Hesaplama kolaylığı bulunan RELU fonksiyonu diğer aktivasyonlara göre de daha hızlı performans göstermektedir [29], [42]. RELU fonksiyonuna ait şekil, Şekil 2.8’de gösterilmektedir.



Şekil 2.8: RELU Aktivasyon Fonksiyonu

Uyarlanabilir moment tahmini Adam ise, modelde belirlenen parametreler için uyarlanabilen öğrenme oranlarını hesaplayabilen fonksiyon çeşididir. Sonuçta derin öğrenme modellerinde kullanılmakta olan aktivasyon ve optimizasyon fonksiyonları sayesinde büyük ve karmaşık veri kümelerinden önemli derecede öğrenme sağlanmaktadır. Aynı zamanda doğrusal olmayan model özelliklerinden daha hızlı ve verimli şekilde sonuçlar elde edilmektedir [29], [43].



3. BULGULAR

3.1. Derin Öğrenme ve Gerçek Zamanlı Kaçak Elektrik Tüketim Analizi

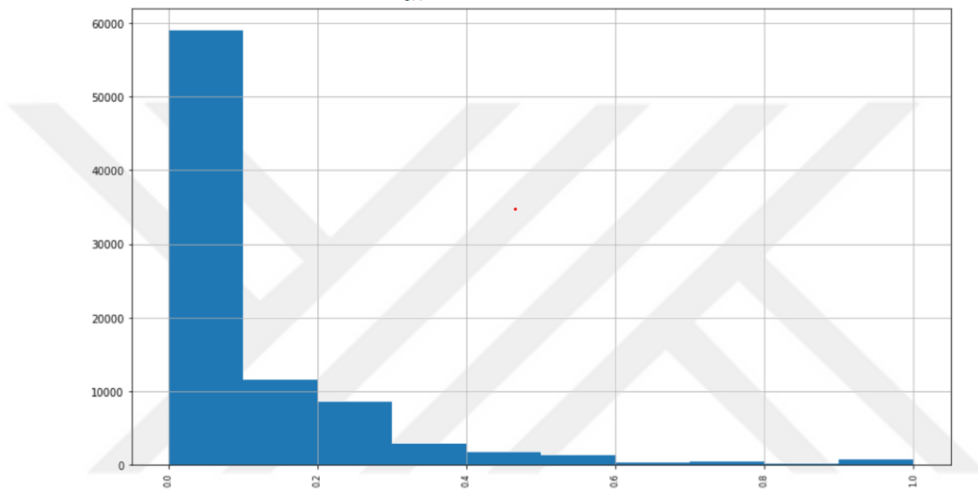
Bu çalışmada bölgelere ait kaçak elektrik tüketimlerinin azaltılması açısından yapılan derin öğrenme ile gerçek zamanlı tahminleme çalışmasına ait uygulama ve bulgular bu bölümde yer almaktadır. Kayıp – kaçak elektrik tüketim tahminlemesinde kullanılan değişkenler şu şekilde ele alınmıştır: Std_Consumption_Value (Standartlaştırılmış tüketim değeri), Avg_Consumption_Value (Ortalama tüketim değeri), COV (Coefficient of Variation – Değişken Değişim Katsayıları), Trafo_Carpan, Abone_Trafo_Tuketim_Oran, Factorized_Tariff_Name (Faktörize edilmiş Tarifeler), Factorized_Karne_No (Faktörize edilmiş karneler), İl_Adi, Factorized_Ilce_Adi, Factorized_Transformer_Number (Faktörize edilmiş trafolar), Installed_Power_Grup (Şebeke maksimum kurulu güç sayısı), Application_Status (Uygulama durumları), Abonelik_Suresi, Abone_Grup_Adi, Subscriber_Kind (Abone Tipi), Person_State (Abonelik durumları), Meter_Type (Sayaç Tipi), Meter_Kind (Sayaç çeşidi), Abone_Yasi_Yil, Gecikmeye_Dusen_Fatura_Adet, Zamaninda_Odenen, Ortalama_Toplam_Tutar, Odenmemis, Ortalama_Odeme_Suresi, Toplam_Tahakkuk_Tutar, Toplam_Gecikme_Tutar, Toplam_Borc_Tutar, Toplam_Tahsilat_Tutar, Toplam_Tahakkuk_Kwh, Analiz_Basari_Sayisi (Kaçak olup olmama durumu). Elektrik tüketimlerine ait değişkenlerin (x_i) ise analiz başarı sayısı olan kaçak ve kaçak olmama durumları üzerine etkileri tahminlemeye alınmaktadır.

3.1.1. Veri Ön İşleme Yöntemi

Makine ve derin öğrenme yöntemlerinde öncelikli olarak verilerin tahmin edilme aşamasına getirilmesi ve en güvenilir sonuçlar alınabilmesi açısından veri ön işleme adımları yapılmaktadır. Bu çalışmada da kategorik verilere nicelleştirmeye yönelik olarak label encoding (etiket kodlama) işlemleri yapılmıştır. Burada Subscriber_Kind (Abone Tipi), Person_state (Abone, kişi durumu), meter_type (sayaç tipi), meter_kind (sayaç çeşidi) değişkenlerine label encoding (etiket kodlama) işlemleri uygulanmıştır.

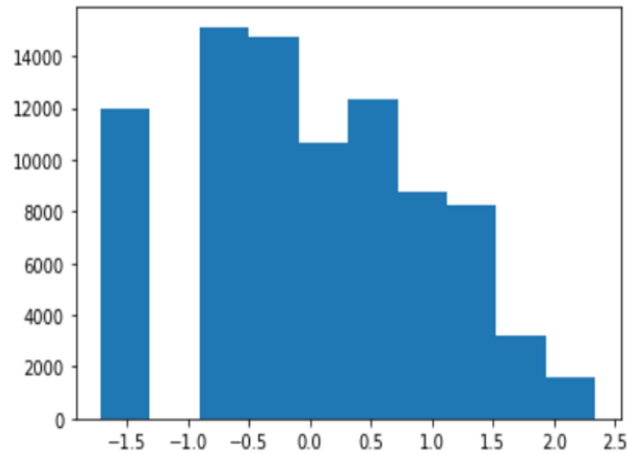
3.1.2. Veri Normalleştirme İşlemleri

Bu çalışmada dönüştürülen veri tiplerinden sonra ele alınan bir diğer işlem ise normalleştirme ve standartlaştırma işlemleridir. Analiz başarı sayısı olan kayıp – kaçak elektrik kullanımının belirlenmesi açısından kullanılan bu değişkenlerde standartlaştırma, Box-Cox normalleştirme ve min-mak dönüşüm işlemleri denenip en uygun normalleştirme işlemlerine karar verilmiştir. İşlem sonuç grafiği Şekil 3.1’de gösterilmektedir.

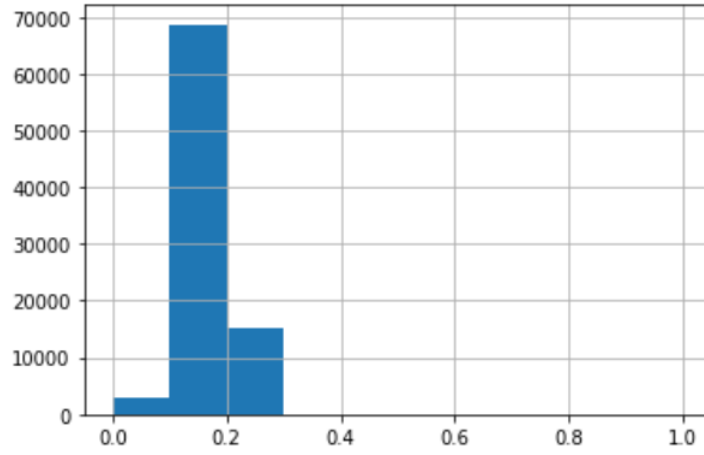


Şekil 3.1: Faktörize Karne No Değişkeni Histogramı

Faktörize edilmiş karne no değişkenin histogramına bakılmış olup değişkenin sağdan çarpık bir yapıya sahip olduğuna ve değerlerin 0 ile 1 arasında dağılım gösterdiğinden dolayı Box – Cox dönüşümü uygulanacağına karar verilmiştir. Sonuçlar Şekil 3.2 ve 3.3 ‘ te gösterilmektedir.



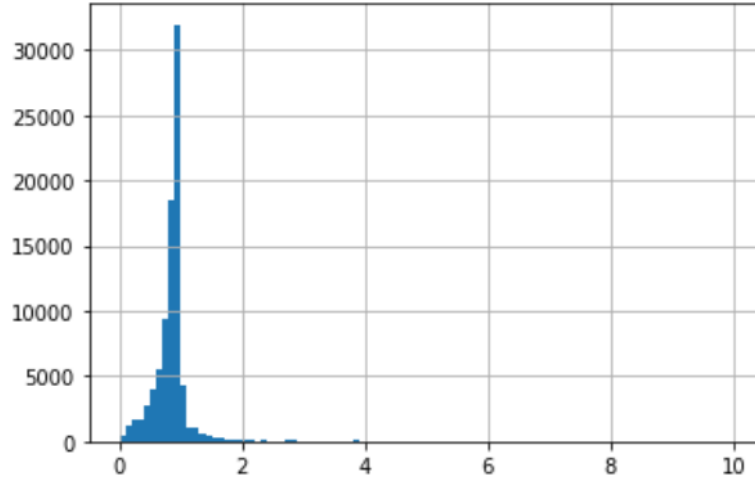
Şekil 3.2: Faktörize Karne No Değişkeni Box – Cox Dönüşümü



Şekil 3.3: Faktörize Tarif Adı Değişkeni Histogramı

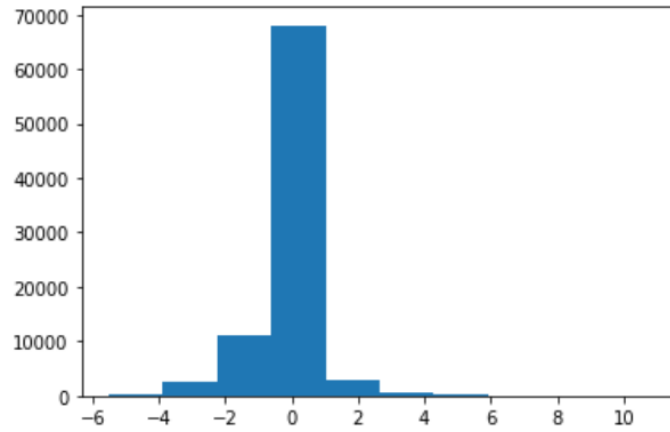
Faktörize edilmiş tarif adı değişkeninin histogramına bakılmış olup değişkenin normal dağılmadığından dolayı ve düzensiz bir çarpıklıktan kaynaklı olarak standartlaştırma işlemi uygulanmıştır.

Faktörize edilmiş karne no değişkeninin histogramına bakılmış olup değişkenin sağdan çarpık bir yapıya sahip olduğuna ve değerlerin 0 ile 1 arasında dağılım gösterdiğinden dolayı Box – Cox dönüşümü uygulanacağına karar verilmiştir.

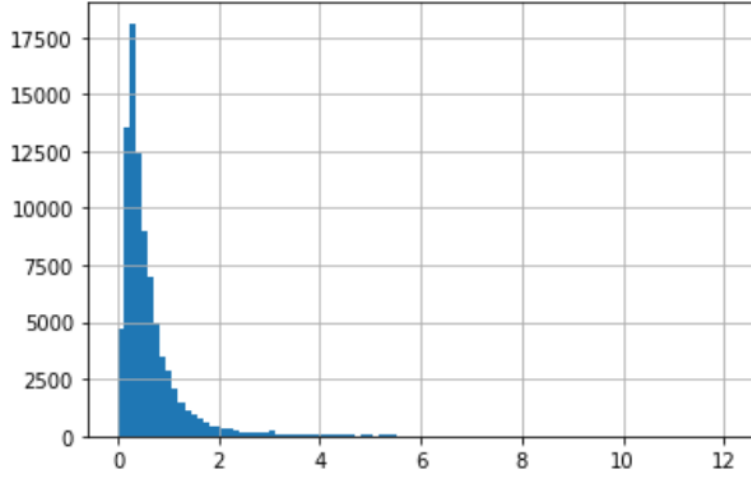


Şekil 3.4: Abone Trafo Tüketim Oranı Değişken Histogramı

Abone trafo tüketim oranı değişkenin Şekil 3.4’ te histogramına bakılmış olup değişkenin çarpık ve üstel bir yapıya sahip olduğundan dolayı Box – Cox dönüşümü uygulanacağına karar verilmiştir. Sonuç ise Şekil 3.5’te gösterilmektedir.

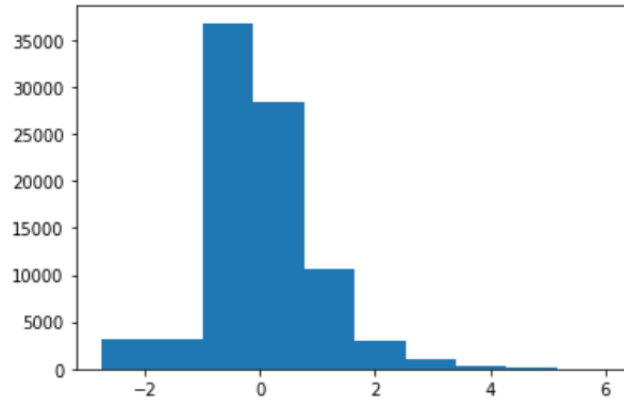


Şekil 3.5: Abone Trafo Tüketim Oranı Değişkeni Normalleştirme Dağılımı



Şekil 3.6: COV Değişkeni Normalleştirme Dağılımı

Bir diğer değişkeni olan coefficient of variation (değişim katsayısı) değişkeninin Şekil 3.6'da histogramına bakılmış olup değişkenin çarpık ve üstel bir yapıya sahip olduğundan dolayı Box – Cox dönüşümü uygulanacağına karar verilmiştir. Sonuç ise Şekil 3.7'de gösterilmektedir.



Şekil 3.7: COV Değişkeni Box-Cox Dönüşümü

Normal dağılmayan verilere uygulanan dönüşüm işlemleriyle değişkenler normale yakınlştırılmış olup modele hazır hale getirilmiştir. Diğer kategorik değişkenler ise nicel (süreklİ) değişken hale dönüştürülmüştür.

Analiz aşamasındaki veri ön işleme adımlarından biri olan outlier(aykırı değEr) tespiti bu çalışma açısından da büyük önem arz etmekte olup aykırı değErler yöntemlerinden biri olan One Class SVM tek sınıflı aykırı değEr temizleme yöntemi ile analiz edilmiştir.

Kişi durumu değişkenlerinden olan PERSON_STATE parametresi 0 ile kodlanarak çalışma boyunca kullanılan Python programa dili ile kodlanarak örneklem sayısı alt parametresi olan “nu” ile içerisindeki aykırı değErler tahminlenerek silinmiştir.

Sonuçta veri ön işleme adımları öncelikle değişken tip ve dönüşümleri, normalleştirme işlemleri ve aykırı değEr ayıklama yöntemleri ile temizlenerek normal veri haline getirilmiştir. Bu çalışmada analiz başarı sayısı olan bağımlı değişken parametresi kayıp-kaçak olması (1) ve olmaması (0) değişkenin en uygun doğruluğa ulaşması adına ilk yöntemler uygulanmış olup modele geçilmiştir. Analiz için gerekli olan veri ön işlem adımları veri setini derin öğrenmeye hazır hale getirmektedir. Bu durumda verim artırılarak doğru sonuçlara ulaşmaya hazır hale getirilmiştir.

3.2. Derin Öğrenme Model Sonuçları

Güneydoğu Anadolu Bölgesi’nin altı iline ait olan kayıp-kaçak elektrik verilerinin veri ön işleme adımları ile elde edilen sonuçlar derin öğrenme modelleri ile eğitilmiştir. İlk olarak Python programına ait olan Keras ve gerekli diğer kütüphaneler yüklenerek model oluşturulmuştur. Gerekli kütüphaneler aşağıda bulunmaktadır.

Derin öğrenme modeli olan LSTM ve Keras kütüphanesi grafikler ve gerekli tabaka kütüphaneleri dahil edilmiştir. Son olarak elde edilen normalleştirilmiş veri seti df adı verilen değişken ile tanımlanmıştır. Bağımlı değişken olan Analiz Başarı Sayısı değişkeni y, diğer değişkenler ise X olarak tanımlanmıştır. Veri setini eğitmek amacıyla X ve y eğitim ve test verileri oluşturulmuştur. Eğitim ve test boyutu, en iyi doğruluğu verecek tabaka sayısı Sequential (Sıralı) model ile denenerek en iyi doğruluk katsayısı bulunmaya çalışılmıştır. Model eğitim verisi ve boyut sayısı ise (53459, 62) şeklindedir.

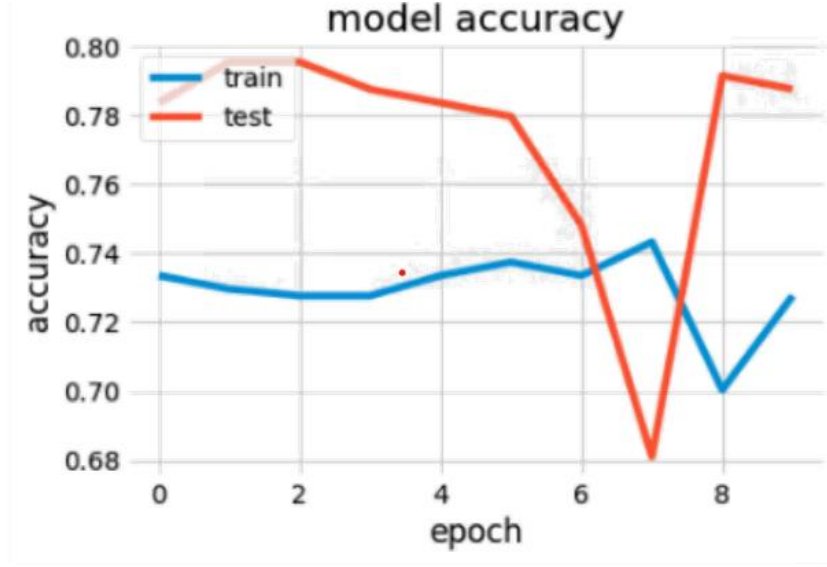
Layer (type)	Output Shape	Param #
lstm_23 (LSTM)	(None, 8, 32)	4352
lstm_24 (LSTM)	(None, 8, 64)	24832
lstm_25 (LSTM)	(None, 128)	98816
dense_23 (Dense)	(None, 256)	33024
dense_24 (Dense)	(None, 128)	32896
dense_25 (Dense)	(None, 64)	8256
dense_26 (Dense)	(None, 16)	1040
dense_27 (Dense)	(None, 1)	17
Total params: 203,233		
Trainable params: 203,233		
Non-trainable params: 0		

Şekil 3.8: Sequential (Sıralı) Model Deneme Sonuçları

Sıralı (Sequential) model ile oluşturulan ve optimal tabakası sayısı Şekil 3.8’de elde edildiği gibi sonuçlanmıştır. Eğitim, test verileri ve uygun tabaka sayısı ile elde edilen doğruluk sonuçları ise Şekil 3.9 ‘da gösterilmektedir.

```
Epoch 1/10
514/514 [=====] - 0s 689us/step - loss: 0.5185 - accuracy: 0.7335 - val_loss: 0.4541 - val_accuracy: 0.7835
Epoch 2/10
514/514 [=====] - 0s 641us/step - loss: 0.5125 - accuracy: 0.7296 - val_loss: 0.5254 - val_accuracy: 0.7953
Epoch 3/10
514/514 [=====] - 0s 654us/step - loss: 0.5140 - accuracy: 0.7276 - val_loss: 0.4460 - val_accuracy: 0.7953
Epoch 4/10
514/514 [=====] - 0s 656us/step - loss: 0.5060 - accuracy: 0.7276 - val_loss: 0.4695 - val_accuracy: 0.7874
Epoch 5/10
514/514 [=====] - 0s 648us/step - loss: 0.5052 - accuracy: 0.7335 - val_loss: 0.4789 - val_accuracy: 0.7835
Epoch 6/10
514/514 [=====] - 0s 635us/step - loss: 0.5083 - accuracy: 0.7374 - val_loss: 0.4445 - val_accuracy: 0.7795
Epoch 7/10
514/514 [=====] - 0s 609us/step - loss: 0.5083 - accuracy: 0.7335 - val_loss: 0.4572 - val_accuracy: 0.7480
Epoch 8/10
514/514 [=====] - 0s 611us/step - loss: 0.5186 - accuracy: 0.7432 - val_loss: 0.5521 - val_accuracy: 0.6811
Epoch 9/10
514/514 [=====] - 0s 621us/step - loss: 0.5544 - accuracy: 0.7004 - val_loss: 0.4559 - val_accuracy: 0.7913
Epoch 10/10
514/514 [=====] - 0s 646us/step - loss: 0.5106 - accuracy: 0.7276 - val_loss: 0.4600 - val_accuracy: 0.7874
```

Şekil 3.9: Sequential (Sıralı) Model Doğruluk Sonuçları



Şekil 3.10: Sequential (Sıralı) Model Doğruluk Grafiği

Derin öğrenme yöntemi olan LSTM sonuçlarına göre 53459 eğitim verisi ile %33'lük test verisi ($\text{validation_split} = 0.33$) alınarak 654 adımda, en ideal optimizasyon sonucunu veren 27 tabaka ile en iyi doğruluk sonucuna (0.7953) varılmıştır. 654 adımda Sıralı model ile binary_ crossentropy (0,1) olan gerçek ile tahmin değeri arasındaki değeri bulan sınıflandırma çeşidi ile ADAM optimum parametresi kullanılarak derin öğrenme sonuçları elde edilmiş olup sonuçlar Şekil 3.10'da gösterilmektedir.

4. SONUÇ

Günümüz teknolojisinde elektrik, savunma sanayi, finans gibi birçok alanda makine öğrenimi ve derin öğrenme yöntemleri kullanılmaktadır. Bu teknolojiler sayesinde geçmiş verilerden gelecek verileri tahmin etme, doğru görüntü sınıflandırma, kolay iletişim ağları, ses tanıma gibi birçok uygulama gerçekleştirilmektedir.

Bu çalışmada da Güneydoğu Anadolu Bölgesi'nin 6 iline ait elektrik abonelerinin kayıp – kaçak elektrik kullanımları tahmin edilerek kaçak tüketimlerin önüne geçilmesi amaçlanmıştır. Tüketimlerin analiz başarı sayısı değişkeni olan kaçak elektrik kullanımı ve kaçak olmayan elektrik abonesinin doğru sınıflandırılması yapılmıştır. Yapılan ön işleme ve derin öğrenme yöntemleriyle kaçak abonelere ait tüketimler belirlenmiş olup bu duruma göre “iş emri” adı verilen tüketiciye ait talimat çıkarılmış olup kaçak elektrik tüketimleri yüksek oranda engellenmiştir.

Sonuç olarak Keras kütüphanesi kullanılarak oluşturulan Sıralı Modelde, binary crossentropy ve ADAM en uygun optimal parametre seçilerek 27 tabakada 0.7953 doğruluk ile model sonuçlanmıştır. Bu durum analiz başarı sayısı olan kayıp – kaçak elektrik olup olmama durumu 0.7953 doğruluk ile doğru sınıflandırılmıştır. Güneydoğu Anadolu Bölgesi’de yapılan kaçak elektrik tüketimleri elde edilen model ile tespit edilerek müdahale şansı ortaya çıkarmış olup bu kayıplar önemli ölçüde azaltılmaya çalışacaktır.

5. KAYNAKLAR

¹ Mutlu, Y., “Türkiye’nin enerji potansiyeli ve yenilenebilir enerji kaynaklarının elektrik enerjisi üretimi açısından önemi,” *Ankara Üniversitesi Çevre Bilimleri Dergisi*, 4(2), pp. 33-54, 2012.

² Türkiye Elektrik Dağıtım A.Ş. Genel Müdürlüğü, Online Türkiye elektrik dağıtım sektör raporu, 2020.

³ Türkiye Enerji Piyasası Düzenleme Kurumu Online, 2020, Strateji Geliştirme Dairesi Başkanlığı, <<https://www.epdk.gov.tr/Detay/Icerik/3-0-24/elektrik-yillik-sektor-raporu>>

⁴ Küçükaslan, B., (2019). Keras Kullanarak Türkiye Gün Öncesi Elektrik Piyasası Piyasa Takas Fiyatı Tahmini. İstanbul Kültür Üniversitesi Fen Bilimleri Enstitüsü.

⁵ Zhang, J. (2018). Automatic Data Cleaning, (Master Thesis, Eindhoven University of Technology, pp.81.

⁶ Kotsiantis S. B., Kanellopoulos D., Pintelas P.E., (2006), “ Data processing for supervised learning” , International Journal of Computer Science, 1(1), 1306-4428.

⁷ Agarwal, V., (2015). “Research on data processing and categorization technique for smartphone review analysis”, International Journal of Computer Applications, 131(4), pp.8.

⁸ Xu, P. (2004). The Analysis of Missing Data in Public Use Survey Databases: A Survey of Statistical Methods (Master Thesis, Capital University of Medical Sciences, China), s. 79.

⁹ Sohrab, F. (2016). *Railway vehicle detection from audio recordings using one-class classification* (Doctoral dissertation).

- ¹⁰ Ok, E. (2020). *Detection of structural damage using one-class support vector machine* (Doctoral dissertation, Institute of Science And Technology).
- ¹¹ Hubert, M., Debruyne, M., & Rousseeuw, P. J. (2018). Minimum covariance determinant and extensions. *Wiley Interdisciplinary Reviews: Computational Statistics*, 10(3), e1421.
- ¹² Sağlamlar, H. (2021). Multi center polyhedral conic classifiers that can classify complex data. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 36(4), 1817-1830.
- ¹³ Brownlee, J. (2020). *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python*. Machine Learning Mastery.
- ¹⁴ Agarwal, V. (2015). Research on data preprocessing and categorization technique for smartphone review analysis. *International Journal of Computer Applications*, 131(4), 30-36.
- ¹⁵ Nayak, S. C., Misra, B. B., & Behera, H. S. (2014). Impact of data normalization on stock index forecasting. *International Journal of Computer Information Systems and Industrial Management Applications*, 6(2014), 257-269.
- ¹⁶ Jayalakshmi, T., & Santhakumaran, A. (2011). Statistical normalization and back propagation for classification. *International Journal of Computer Theory and Engineering*, 3(1), 1793-8201.
- ¹⁷ Rana, R. ve Singhal, R. (2015). Chi-square test and its application in hypothesis testing. *Journal of the Practice of Cardiovascular Sciences*, 1 (1), 69.
- ¹⁸ Ali, P. J. M., Faraj, R. H., Koya, E., Ali, P. J. M., & Faraj, R. H. (2014). Data normalization and standardization: a technical report. *Mach Learn Tech Rep*, 1(1), 1-6.

¹⁹ Mehdiyev, N., Lahann, J., Emrich, A., Enke, D., Fettke, P., & Loos, P. (2017). Time series classification using deep learning for process planning: A case from the process industry. *Procedia Computer Science*, 114, 242-249.

²⁰ LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.

²¹ A. G. Ivakhnenko and V. G. Lapa, "Cybernetic Predicting Devices," 1966.

²² Şeker, A., Diri, B., & Balık, H. H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. *Gazi Mühendislik Bilimleri Dergisi (GMBD)*, 3(3), 47-64.

²³ Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4), 611-629.

²⁴ Hochreiter, J. S., & Schmidhuber, J. (1997). Long short-term memory (Neural Computation 9 (8): 1735 {1780, 1997).

²⁵ DiPietro, R., & Hager, G. D. (2020). Deep learning: RNNs and LSTM. In *Handbook of medical image computing and computer assisted intervention* (pp. 503-519). Academic Press.

²⁶ Chen, G., Tang, B., Zeng, X., Zhou, P., Kang, P., & Long, H. (2022). Short-term wind speed forecasting based on long short-term memory and improved BP neural network. *International Journal of Electrical Power & Energy Systems*, 134, 107365.

²⁷ Bansal, T., Belanger, D., & McCallum, A. (2016, September). Ask the gru: Multi-task learning for deep text recommendations. In *proceedings of the 10th ACM Conference on Recommender Systems* (pp. 107-114).

²⁸ Moolayil, J., Moolayil, J., & John, S. (2019). *Learn Keras for deep neural networks* (pp. 1-192). Birmingham: Apress.

- ²⁹ Lee, H., & Song, J. (2019). Introduction to convolutional neural network using Keras; an understanding from a statistician. *Communications for Statistical Applications and Methods*, 26(6), 591-610.
- ³⁰ Güngör, O. (2015). Kaçak Elektrik Kullanımının GSM Aracılığıyla Takibi/Follow the Electricity Leakes Through the GSM Signals. *EMO Bilimsel Dergi*, 4(8), 29-34.
- ³¹ Meier, A., & Huber, W. (1999). Results from the investigations on leaking electricity in the USA. In *Energy Efficiency in Household Appliances* (pp. 342-352). Springer, Berlin, Heidelberg.
- ³² Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.
- ³³ Imron, M. A., & Prasetyo, B. (2020). Improving algorithm accuracy k-nearest neighbor using z-score normalization and particle swarm optimization to predict customer churn. *Journal of Soft Computing Exploration*, 1(1), 56-62.
- ³⁴ Denoyer, L., & Gallinari, P. (2014). Deep sequential neural network. *arXiv preprint arXiv:1410.0510*.
- ³⁵ Gulli, A., & Pal, S. (2017). *Deep learning with Keras*. Packt Publishing Ltd.
- ³⁶ Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., & Ghogho, M. (2016, October). Deep learning approach for network intrusion detection in software defined networking. In *2016 international conference on wireless networks and mobile communications (WINCOM)* (pp. 258-263). IEEE.
- ³⁷ Saxe, J., & Berlin, K. (2015, October). Deep neural network based malware detection using two dimensional binary program features. In *2015 10th international conference on malicious and unwanted software (MALWARE)* (pp. 11-20). IEEE.
- ³⁸ Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4), 1875-1897.

- ³⁹ Abraham, A. (2005). Artificial neural networks. *Handbook of measuring system design*.
- ⁴⁰ Harrington, P. D. B. (1993). Sigmoid transfer functions in backpropagation neural networks. *Analytical Chemistry*, 65(15), 2167-2168.
- ⁴¹ Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *towards data science*, 6(12), 310-316.
- ⁴² Bak, S., Tran, H. D., Hobbs, K., & Johnson, T. T. (2020, July). Improved geometric path enumeration for verifying relu neural networks. In *International Conference on Computer Aided Verification* (pp. 66-96). Springer, Cham.
- ⁴³ Salem, H., Kabeel, A. E., El-Said, E. M., & Elzeki, O. M. (2022). Predictive modelling for solar power-driven hybrid desalination system using artificial neural network regression with Adam optimization. *Desalination*, 522, 115411.
- ⁴⁴ Menon, A., Mehrotra, K., Mohan, C. K., & Ranka, S. (1996). Characterization of a class of sigmoid functions with applications to neural networks. *Neural Networks*, 9(5), 819-835.
- ⁴⁵ Agostinelli, F., Hoffman, M., Sadowski, P., & Baldi, P. (2014). Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830*.