

**T.C.
İZMİR DEMOKRASİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

DERİN YAPAY SİNİR AĞLARI İLE ELEKTRİK TÜKETİM TAHMİNİ

Fatih ÖZCAN

**Danışman
Prof. Dr. Barış BOZKURT**



İZMİR-2022

Fatih ÖZCAN

DERİN YAPAY SINIR AĞLARI İLE ELEKTRİK TÜKETİM TAHMİNİ

2022

TAAHHÜTNAME

Bu tezin İzmir Demokrasi Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü'nde, akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Fatih ÖZCAN



İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER	I
SİMGELER VE KISALTMALAR DİZİNİ	III
ŞEKİLLER DİZİNİ.....	IV
TABLO DİZİNİ	V
TEŞEKKÜR.....	VI
ÖZET.....	VII
1.GİRİŞ	1
1.1. Elektrik Enerjisi Tüketim Tahmininin Önemi.....	1
1.2. Türkiye'nin Elektrik Enerjisi Tüketimi	2
1.3. Türkiye'de Kullanılan Enerji Tahmin Modelleri	5
1.4. Zaman Serisi Problemi Olarak Elektrik Enerjisi Tüketim Tahmini.....	6
1.5. STLF Modelini Oluştururken Karşılaşılan Zorluklar.....	8
1.6. Derin Öğrenme Modelleri İle Elektrik Enerjisi Tüketim Tahmini.....	8
2. LİTERATÜR ÖZETİ.....	10
2.1. Türkiye'nin Verileri İle Sunulmuş STLF Çalışmaları	10
2.2. Türkiye'nin Verileri Sunulmuş İle MTLF ve LTMF Çalışmaları.....	15
2.3. Farklı Ülke ve Şehirlerin Verileri İle Sunulmuş STLF Çalışmaları.....	18
2.4. RNN, CNN ve TCN İle Gerçekleştirilmiş Diğer STLF Çalışmaları.....	19
2.5. Tezin Amacı Ve Katkıları	21
3. ZAMAN SERİSİ TAHMİNİ İÇİN DERİN ÖĞRENME MİMARİLERİ	22
3.1. Çok Katmanlı Algılayıcı (Multi-Layer Perceptron).....	22
3.1.1. Geri Yayılım (Backpropagation) Algoritması	24
3.2. Tekrarlayan Sinir Ağları (Recurrent Neural Networks).....	26
3.2.1. Basit Bir RNN Modelinin Mimarisi	27
3.2.2. Elman Tekrarlayan Sinir Ağları (Elman Recurrent Neural Networks)	29
3.2.3. Uzun Kısa Süreli Bellek Ağları (Long Short-Term Memory Networks) .	30
3.2.4. Yankı Durumu Ağları (Echo State Networks).....	34
3.2.5. Kapılı Tekrarlayan Birimler (Gated Recurrent Units).....	36
3.2.6. Kodlayıcı-Kod Çözücü İstiflenmiş Uzun Kısa Süreli Bellek (Encoder-Decoder Stacked LSTM)	38
3.3. Evrişimli Sinir Ağları (Convolutional Neural Networks)	39

3.3.1. CNN Mimarisi	40
3.4. Zamansal Evrişimli Ağ (Temporal Convolutional Network).....	41
3.4.1. TCN Mimarisi.....	42
4. MATERYAL VE YÖNTEM	44
4.1. Veri Setine Genel Bir Bakış	44
4.2. Derin Öğrenme Mimarilerinin Deneysel Kurulumu	51
4.3. Değerlendirme Prosedürü.....	54
4.3.1. Veri Setinin Bölünmesi.....	55
4.3.2. Ön İşleme (Pre-Processing) ve Eğitimin (Training) Gerçekleştirilmesi... 56	
4.3.3. Hareketli Pencere Prosedürü (Moving Window Procedure)	59
4.3.4. Değerlendirme Metrikleri	61
4.3.5. İstatistiksel Analiz	63
5. ARAŞTIRMA BULGULARI VE TARTIŞMA	64
5.1. Modellerin Tahmin Doğruluğuna Etkisi	64
5.2. Özniteliklerin Tahmin Doğruluğuna Etkisi	67
5.3. Modellerin Hesaplama Zamanına Etkisi	68
5.4. Modellerin Mimarisi Yapısının Tahmin Doğruluğuna Etkisi	69
5.5. Model Hiperparametrelerinin Eğitim Dinamikleri Üzerindeki Etkisi.....	73
5.6. Geçmiş Değerlerin ve Tahmin Ufkunun Tahmin Doğruluğuna Etkisi	77
5.7. En İyi Model İle Gerçekleştirilen Tahminler	79
5.8. Hata ve MAPE Dağılımı (Loss and MAPE Distribution).....	81
6. SONUÇ VE ÖNERİLER	83
KAYNAKLAR	85
ÖZGEÇMİŞ	Hata! Yer işareti tanımlanmamış.

SİMGELER VE KISALTMALAR DİZİNİ

GSYİH	Gayri Safi Yurtiçi Hâsıla
ETKB	Türkiye Cumhuriyeti Enerji ve Tabii Kaynaklar Bakanlığı
IAEA	Uluslararası Atom Enerjisi Kurumu (International Atomic Energy Agency)
MAED	Enerji Talep Analizi Modeli (Model for Analysis of Energy Demand)
WASP	Viyana Otomatik Sistem Planlama Paketi (Wien Automatic System Planning Package)
DPT	Devlet Planlama Teşkilatı
TEİAŞ	Türkiye Elektrik İletim A.Ş.
EPIAŞ	Enerji Piyasaları İşletme A.Ş.
VSTLF	Çok Kısa Vadeli Yük Tahmini (Very Short-Term Load Forecasting)
STLF	Kısa Vadeli Yük Tahmini (Short-Term Load Forecasting)
MTLF	Orta Vadeli Yük Tahmini (Medium-Term Load Forecasting)
LTLF	Uzun Vadeli Yük Tahmini (Long-Term Load Forecasting)
ARIMA	Bütünleşik Otoresgresif Hareketli Ortalama
RNN	Tekrarlayan Sinir Ağı (Recurrent Neural Network)
MLP	Çok Katmanlı Algılayıcı (Multi-Layer Perceptron)
ERNN	Elman Tekrarlayan Sinir Ağı (Elman Recurrent Neural Network)
LSTM	Uzun Kısa Süreli Bellek Ağı (Long Short-Term Memory Network)
ESN	Yankı Durumu Ağı (Echo State Network)
GRU	Kapılı Tekrarlayan Birimler (Gated Recurrent Units)
SLSTM	İstiflenmiş Uzun Kısa Süreli Bellek (Stacked LSTM)
CNN	Evrişimli Sinir Ağı (Convolutional Neural Network)
TCN	Zamansal Evrişimli Ağ (Temporal Convolutional Network)
BP	Geri Yayılım (Backpropagation)
GA	Genetik Algoritma
MAPE	Ortalama Mutlak Yüzde Hata (Mean Absolute Percentage Error)
YSA	Yapay Sinir Ağı (ANN)
GWh	Gigawatt-Saat (Gigawatt-Hour)
MWh	Megawatt-Saat (Megawatt-Hour)

ŞEKİLLER DİZİNİ

Şekil 1.1. Elektrik tüketimi (MWh) [7] ve GSYİH (Milyon Dolar) değerleri [8]	4
Şekil 1.2. Elektrik enerjisi tüketimi (MWh) [7] ve Türkiye nüfusu [9]	5
Şekil 1.3. Tez çalışmasında kullanılan özellikler	7
Şekil 2.1. Referans çalışmadan [37] elde edilmiş tüketim tahmin grafiği	17
Şekil 2.2. Referans çalışmadaki [42] modellerin MAPE değerleri	20
Şekil 3.1. Çok katmanlı bir algılayıcı [53]	23
Şekil 3.2. Basit bir RNN yapısı	28
Şekil 3.3. Elman tekrarlayan sinir ağının (ERNN) yapısı [73]	30
Şekil 3.4. Uzun kısa süreli belleğin (LSTM) yapısı [76]	33
Şekil 3.5. Yankı durumu ağının (ESN) yapısı [84]	35
Şekil 3.6. GRU'nun mimari grafiği [87]	36
Şekil 3.7. Encoder-Decoder Stacked LSTM'nin yapısı	38
Şekil 3.8. Evrişimli sinir ağının (CNN) yapısı	40
Şekil 3.9. Farklı çekirdek boyutlu ve genişleme oranlı, 2 katmanlı TCN mimarisi ..	42
Şekil 4.1. Gün bazında ortalama saatlik elektrik enerjisi tüketimi (MWh)	46
Şekil 4.2. Gün bazında ortalama saatlik elektrik enerjisi tüketimi (MWh)	47
Şekil 4.3. 32 hafta için ortalama saatlik elektrik enerjisi tüketimi (MWh)	47
Şekil 4.4. 32 gün için ortalama saatlik elektrik enerjisi tüketimi (MWh)	48
Şekil 4.5. Türkiye'nin gün bazında ortalama hissedilen sıcaklık değeri (°C)	49
Şekil 4.6. Hissedilen sıcaklığa (°C) göre elektrik enerjisi tüketimi (MWh)	50
Şekil 4.7. Takvimsel özelliklere göre ortalama saatlik elektrik tüketimi (MWh)	50
Şekil 4.8. Tez çalışmasındaki eğitim, test ve doğrulama setlerinin ayrılması	55
Şekil 4.9. RNN'ler için farklı giriş-çıkış konfigürasyonları.	57
Şekil 4.10. Tez çalışmasındaki hareketli pencere yönteminin gösterimi [51]	60
Şekil 4.11. Her eğitimin sonunda kaydedilen örnek grafikler	63
Şekil 5.1. Modeller için MAPE sonuçlarının dağılımını gösteren kutu grafikleri	65
Şekil 5.2. Eğitim ve test süresi açısından tüm modellerin dağılımı	68
Şekil 5.3. Modellerin normalize edilmiş eğitim sürelerine göre performansları	69
Şekil 5.4. Farklı konfigürasyonlara sahip farklı RNN'lerin MAPE değerleri	71
Şekil 5.5. MLP modeli için kullanılan batch size değerlerine göre ölçümler	74
Şekil 5.6. MLP için öğrenme oranı (learning rate) değerlerine göre ölçümler	75
Şekil 5.7. MLP için optimize edicilere (optimizer) göre eğitim ölçümleri	76
Şekil 5.8. 72, 168 ve 336 saatlik geçmiş değere göre 1 saatlik tahmin grafikleri	78
Şekil 5.9. 72, 168 ve 336 saatlik geçmiş değere göre 6 saatlik tahmin grafikleri	78
Şekil 5.10. 72, 168 ve 336 saatlik geçmiş değere göre 24 saatlik tahmin grafikleri ..	79
Şekil 5.11. En iyi model için 48 saatlik tahmin (1 saatlik zaman ufku)	79
Şekil 5.12. En iyi model için 1 haftalık tahmin (24 saatlik zaman ufku)	80
Şekil 5.13. En iyi model ile gerçekleştirilen 1 yıllık tahmin (1 saatlik zaman ufku). ..	80
Şekil 5.14. En iyi model için hata dağılım grafiği	81
Şekil 5.15. Gerçekleştirilen tüm eğitimler için MAPE dağılım grafiği	82

TABLO DİZİNİ

Tablo 1.1. Tüketici türü bazında faturalanan elektrik enerjisi tüketimi (MWh) [3]	3
Tablo 2.1. Referanstaki [34] gelecek ve gerçek değerler [35] (TWh)	17
Tablo 4.1. Tez çalışmasında kullanılan veri setindeki sütunlar ve özellikleri	45
Tablo 4.2. Derin öğrenme mimarileri için kullanılan parametreler	52
Tablo 4.3. Tez çalışmasına dâhil edilen eğitim parametreleri	59
Tablo 5.1. Modeller ve öznitelikler için elde edilen en iyi MAPE ölçümleri.....	66
Tablo 5.2. Modellerin ortalama, en düşük ve en yüksek MAPE değerleri	66
Tablo 5.3. Özniteliklerin ortalama MAPE ve süre ölçümleri	67
Tablo 5.4. Farklı MLP konfigürasyonları için ölçüm değerleri	70
Tablo 5.5. Farklı LSTM konfigürasyonları için ölçüm değerleri	70
Tablo 5.6. Farklı CNN konfigürasyonları için ölçüm değerleri.....	72
Tablo 5.7. Farklı TCN konfigürasyonları için ölçüm değerleri	73
Tablo 5.8. Eğitimdeki geçmiş değer ve tahmin ufkuna göre ortalama MAPE	77

TEŞEKKÜR

Çalışmamın her aşamasında bana destek olan, bilgi ve deneyimleri ile yol gösteren danışman hocam Sayın Prof. Dr. Barış BOZKURT' a ve öğrenim hayatım boyunca beni maddi ve manevi olarak destekleyen ve hep yanımda olan aileme yürekten teşekkür ederim. Çalışmada yer alan hususlar bireysel görüşümü yansıtmakta olup, Türk Silahlı Kuvvetlerinin görüşlerini yansıtmaz.

Fatih ÖZCAN
İzmir, 2022



ÖZET

Yüksek Lisans Tezi

Derin Yapay Sinir Ağları İle Elektrik Tüketim Tahmini

Fatih ÖZCAN

İzmir Demokrasi Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Barış BOZKURT

Türkiye'nin gelişmekte olan yapısı sebebiyle özellikle elektrik enerjisine olan ihtiyacı sürekli artmaktadır. Belirli dönemler için elektrik enerjisi talebinde yüksek farklar oluşmaktadır. Elektrik enerjisi büyük oranlarda depolanamaması sebebiyle söz konusu enerji talebini eş zamanlı ve kesintisiz olarak karşılamak gerekmektedir. Aksi durumda, bölgesel elektrik kesintileri veya elektrik enerjisi israfı gerçekleşmektedir. Bu problemleri en aza indirmek için belirli bir zaman ufku da elektrik enerjisi talebinin tahmin edilmesi gerekmektedir.

Geleneksel modeller ile kıyaslandığında derin öğrenme yöntemleri, birçok makine öğrenimi problemi için daha iyi performans göstermektedir. Ayrıca, zamansal bağımlılıklar içeren zaman serisi problemleri için derin öğrenme yöntemleri başarıyla uygulanmaktadır. Derin öğrenme yöntemleri, zaman serisi problemleri için ne kadar etkili olsa da, en uygun parametreleri seçmek uzmanlık gerektiren karmaşık bir konudur. Gerçekleştirilen bu tez çalışmasında, 8 adet farklı derin öğrenme mimarisinin kapsamlı bir araştırması gerçekleştirilmiştir. Daha sonra, söz konusu mimarilerin çeşitli mimari konfigürasyon ve eğitim hiperparametreleri kullanılarak eğitimler gerçekleştirilmiştir. Deneyler gerçekleştirilirken, Türkiye'nin 01.01.2010 tarihi saat 00:00'dan 31.12.2020 tarihi saat 23:00'a kadar saatlik olarak örneklenmiş olan elektrik enerjisi tüketim, sıcaklık ve ekonomik verileri kullanılmıştır. Ağırlıklandırılmış toplam hissedilen sıcaklık olarak adlandırılan yeni bir öznitelik, deneylere dâhil edilmiştir. Ayrıca, 2020 yılı test verisi olarak kullanılmış olup, test işlemi gerçekleştirirken hareketli pencere yaklaşımı kullanılmıştır.

Gerçekleştirilen deneylerin sonucunda, zaman serisi problemi için en iyi tahmin performansını LSTM ve ardından GRU mimarisi göstermiştir. CNN'lerin ise kısa eğitim süresinde, tatmin edici performansa ulaştığı görülmüştür. Hareketli pencere prosedürü ile test işlemi daha verimli hale gelmiştir. Ayrıca, önerilen ağırlıklandırılmış toplam hissedilen sıcaklık kavramının tahmin doğruluğunu arttırdığı tespit edilmiştir. Sonuç olarak, optimal bir mimari konfigürasyon bulmanın önemi deneyimlenmiştir.

Anahtar Kelimeler: derin öğrenme, zaman serisi

2022, 97 sayfa

ABSTRACT

M.Sc. Thesis

Forecasting Electricity Consumption with Deep Artificial Neural Networks

Fatih ÖZCAN

**İzmir Demokrasi University
Graduate School of Applied and Natural Sciences
Department of Electrical-Electronics Engineering**

Supervisor: Prof. Dr. Barış BOZKURT

Due to the developing structure of Turkey, especially the need for electrical energy is constantly increasing. Electrical energy demand varies according to certain periods. Since electrical energy cannot be stored in large quantities, it is necessary to meet the energy demand simultaneously and uninterruptedly. Otherwise, regional power outages or wastage of electricity occur. In order to minimize these problems, it is necessary to estimate the electrical energy demand over a certain time horizon.

Compared to traditional models, deep learning methods perform better for many machine learning problems. In addition, deep learning methods have been successfully applied for time series problems involving temporal dependencies. Although deep learning methods are effective for time series problems, choosing the most appropriate parameters is a complex subject that requires expertise. In this thesis, a comprehensive research of 8 different deep learning architectures was carried out. Then, trainings were carried out using various architectural configurations and training hyperparameters of the mentioned architectures. While performing the experiments, Turkey's electrical energy consumption, temperature and economic data sampled hourly from 01.01.2010 at 00:00 to 31.12.2020 at 23:00 were used. A new feature called weighted total felt temperature was included in the experiments. In addition, the year 2020 was used as the test data, and the moving window approach was used while performing the test.

As a result of the experiments, LSTM and then GRU architecture showed the best prediction performance for the time series problem. CNNs, on the other hand, have been found to achieve satisfactory performance in a short training period. With the moving window procedure, the testing process is more efficient. It has been found that the weighted total felt temperature increases the prediction accuracy. The importance of finding an optimal architectural configuration is experienced.

Keywords: deep learning, time series

2022, 97 pages

1.GİRİŞ

Yunanca bir kelime olan enerji, bir sistemdeki iş yapma yeteneği olarak ifade edilmektedir. Çeşitli kaynaklardan elde edilen enerji, sosyal, ekonomik ve çevresel olarak herhangi bir ülkenin sürdürülebilir şekilde kalkınması için hayati öneme sahiptir. 2022 yılı itibariyle 84 milyon olan Türkiye nüfusunun 2040 yılında 100 milyonu geçmesi beklenmektedir [1]. Söz konusu nüfus artışı nedeniyle Türkiye'nin enerji gereksinimi ve tüketimi artacaktır. Ayrıca Türkiye'nin sanayileşen ve gelişmekte olan yapısı, enerji ihtiyacını da devamlı olarak arttırmaktadır. Enerji tüketimini doğru bir şekilde tahmin etmek, enerji planlaması, enerji politikalarının oluşturulması ve stratejilerin geliştirilmesi açısından gereklidir [2].

İkincil enerji kaynaklarından biri olan elektrik enerjisi ise petrol, kömür, doğal gaz, nükleer güç, hidrolik, biokütle, gelgit, güneş ve rüzgâr gibi birincil enerji kaynaklarının dönüştürülmesiyle üretilir. Elektrik enerjisi aydınlatma, sanayi, iletişim, ulaşım, tıp, güvenlik, havacılık gibi alanlarda da yaygın olarak kullanılmaktadır. Elektrik enerjisi, uzak mesafelere hızla ve kolayca iletilebilen, diğer enerji formlarına dönüştürülebilen, çevreye zararı olmayan, verimli ve güvenilir bir enerji kaynağıdır. Gerekli donanımlar sağlandığı sürece her ortamda kullanılabilmesi, koku ve atık bırakmaması, teknolojik gelişmelere doğrudan etki etmesi sebebiyle elektrik enerjisine olan talep artarak devam etmektedir.

1.1. Elektrik Enerjisi Tüketim Tahmininin Önemi

Yıl içerisinde Türkiye'de bölgesel, mevsimsel ve zamansal faktörlere dayalı %200'e varan elektrik enerjisi talep farkı oluşmaktadır. Elektrik santrallerinin büyük bir kısmının anlık olarak açılıp kapatılarak arz ve talebi dengelemesi mümkün değildir. Elektrik enerjisini yüksek miktarlarda depolamanın günümüz koşullarında mümkün olmaması ve elektrik enerjisi talebinin değişkenlik gösteren karakteristiği sebebiyle Türkiye geneli elektrik enerjisinin talebinin doğru bir şekilde tahmin edilerek kesintisiz, eş zamanlı ve anlık olarak arzın gerçekleştirilmesi gerekmektedir. Elektrik enerjisi arzının talepten düşük olması durumunda zorunlu bölgesel elektrik kesintilerinin yanı sıra zorlama elektrik tasarrufları meydana gelecek ve özellikle üretim gibi alanlarda problemler ortaya çıkacaktır. Söz konusu problemler ekonomik kriz riskini arttırıp ülkenin refah seviyesinin düşmesine yol açabilmektedir [10].

Elektrik enerjisi arzının talepten yüksek olması durumunda ise elektrik enerjisi israfı meydana gelerek finansal zararlar ortaya çıkmaktadır. Uzun vadede hatalı yapılan elektrik enerjisi talep tahminleri, yüksek maliyetli olan enerji yatırımlarının yanlış yönlendirilmesine sebep olarak kaynak israfına yol açmaktadır.

Elektrik enerjisi üretim sistemleri bölgesel, mevsimsel ve zamansal talep değişimleri göz önünde bulundurularak en düşük maliyetle sürekli ve kesintisiz bir şekilde talebi karşılayabilecek düzeyde planlanmalıdır. Bu yüzden gerekli stratejik kararlar alınarak enerji yatırımlarının yapılması önemli bir role sahiptir. Gelecekteki elektrik enerjisi talebinin doğru ve güvenilir bir şekilde tahmin edilmesi ve kaynak planlamasında temel bir hedef haline gelmesi, geliştirilecek politikaların etkinliğini belirleyen en önemli unsurdur [11].

Türkiye'nin kısa vadeli elektrik enerjisi tüketimini tahmin etmek, tez konusunun odak noktasını oluşturmaktadır. Bu görevi doğru bir şekilde gerçekleştirmek, enerji sektöründeki sistem yöneticilerinin arz talep dengesini yerine getirmesi için yardımcı olacaktır. Ayrıca, santral bakım planlamalarında önlemler alınarak, oluşacak problemlerin önceden önüne geçilmesi için olanak sağlanacaktır. Doğru elektrik enerjisi tüketim tahmini, sistemi en uygun hale getirerek kaynakların etkin ve verimli bir şekilde kullanılmasına imkân oluşturacaktır.

1.2. Türkiye'nin Elektrik Enerjisi Tüketimi

Türkiye'nin elektrik enerjisi talebi olağan dışı ekonomik ve sosyal olayların gerçekleştiği dönemler dışında sürekli artmaktadır. Elektrik enerjisi, üretildiği anda tüketilmesi gereken bir enerji türüdür. Bu yüzden tüketimi üretimine paralel bir gelişme göstermektedir. Elektrik enerjisine olan talebin artması kurulu güç ve üretim değerlerinin de artmasına sebep olmaktadır [4].

Türkiye'de elektrik enerjisi tüketimi tüketici türü bazında beş alt sınıfa ayrılmaktadır. Bunlar sırasıyla; sanayi, mesken, ticarethane, tarımsal sulama ve aydınlatmadır. Türkiye'nin 2020 yılında tüketici türlerine göre faturalanan elektrik enerjisi tüketim oranları Tablo 1.1.'de verilmiştir.

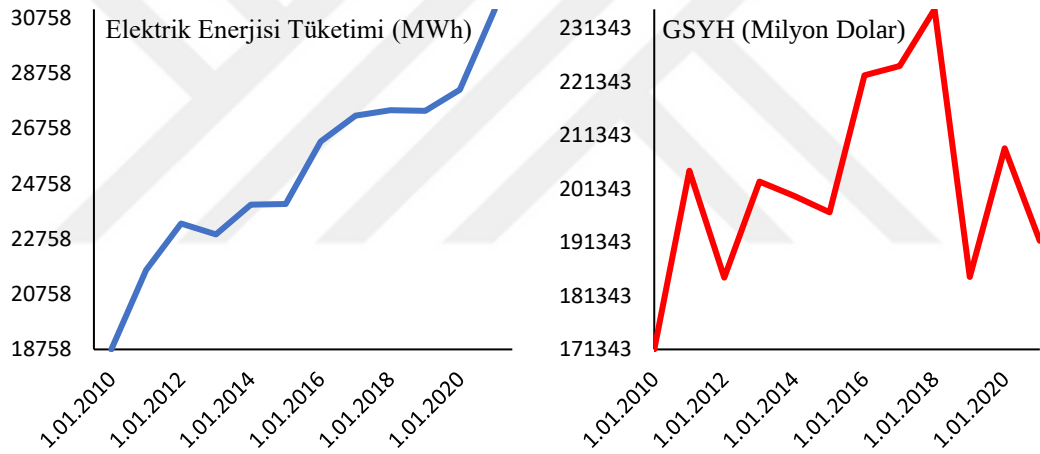
Tablo 1.1. Tüketici türü bazında faturalanan elektrik enerjisi tüketimi (MWh) [3]

Tüketici Türü	Tüketim Miktarı (MWh)	
	Toplam	Oran (%)
Sanayi	99.767.340,58	42,74
Mesken	60.133.987,76	25,76
Ticarethane	57.633.481,86	24,69
Tarımsal Sulama	10.805.968,48	4,63
Aydınlatma	5.095.836,13	2,18
Genel Toplam	233.436.614,81	100,00

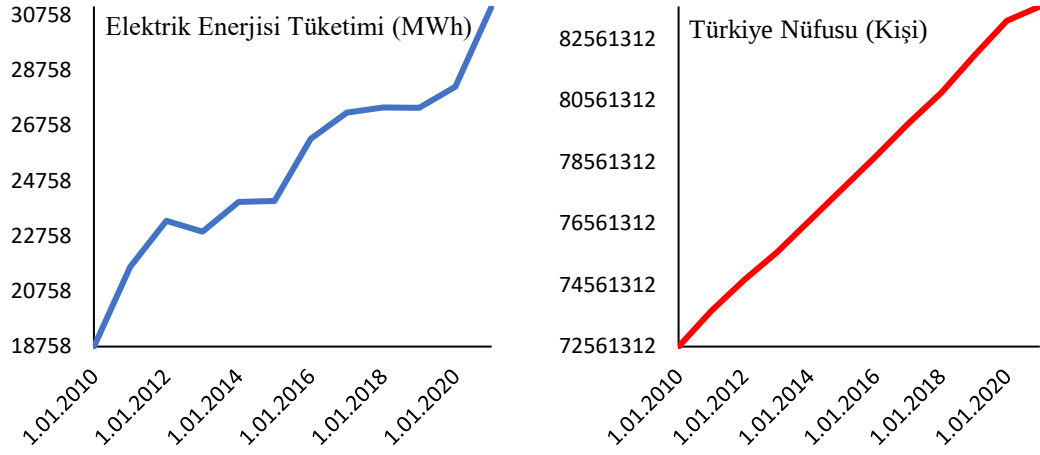
Türkiye'nin faturalanan elektrik enerjisi tüketiminde en yüksek paya %42,74'lük oranla sanayi türü sahip iken sanayi türünü %25,76'lık oranla mesken türü takip etmektedir. Ticarethane türü ise %24,69'luk oranla mesken türü ile yakın bir orana sahiptir. Tarımsal sulama türü %4,63'lük oranla dördüncü sırada ve aydınlatma türü ise %2,18'lik oranla beşinci sırada yer almaktadır.

Elektrik enerjisine olan talep, uzun vadede artan nüfustan, ekonomik olaylardan, sanayi ve teknolojiye gelişmelerden etkilenirken kısa vadede sıcaklık, nem, güneşlenme süresi gibi meteorolojik etkenlerden ve iş günleri, resmi bayramlar, resmi tatiller gibi takvime dayalı faktörlerden etkilenmektedir. Söz konusu özelliklerden her bir tüketici türü farklı seviyede etkilenmektedir. Sanayi türünde elektrik enerjisi tüketimine büyük ölçüde ekonomi, teknolojiye meydana gelen değişimler ve takvime dayalı faktörler yön vermektedir. Takvime dayalı çalışma koşulları nedeniyle, resmi tatiller ve bayramlarda sanayi türünde düşük elektrik kullanımı gözlemlenmektedir. Mesken, ticarethane ve tarımsal sulama türleri için elektrik enerjisi tüketimindeki değişimler büyük oranda mevsimsellik kaynaklıdır. Yaz aylarında soğutma amacıyla çoğunlukla klimalar kullanılırken kış aylarında ısıtıcı ve klimalarla birlikte kömür, doğalgaz gibi alternatif kaynaklarda kullanılmaktadır. Bu sebeple Türkiye'nin mevsime dayalı elektrik tüketimi incelendiğinde yaz aylarında elektrik enerjisi tüketimi kış aylarına göre daha yüksek seviyede olduğu görülmektedir. Yaz aylarında gün ışığından diğer mevsimlere kıyasla daha çok yararlanıldığından özellikle ticarethane ve tarımsal sulama türleri için elektrik enerjisi tüketiminde azaltıcı etki söz konusu olmaktadır. Ayrıca tasarruflu elektrikli cihazların kullanılması mesken türü tüketimde düşürücü bir etki oluşturmaktadır [5].

Türkiye'nin 2010 – 2021 yılları arasındaki elektrik enerjisi tüketim ve GSYİH değerleri Şekil 1.1.'de verilmiştir. Bu değerler incelendiğinde, 2019 yılına kadar iki değişken arasında ilişki gözlemlenirken 2019 yılından sonra söz konusu iki değişkenin doğrudan birbiri ile bağlantılı olmadığı görülmektedir. Ekonomik büyüme ile elektrik enerjisi tüketimi arasındaki ilişkiyi açıklayan temel bir ekonomik teori bulunmamaktadır fakat 1950 - 2000 yılları arasında Türkiye'nin elektrik enerjisi tüketimi ve GSYİH ile yapılan çalışmada elektrik enerjisi tüketimindeki artışın büyüyen ekonominin öncü göstergesi olarak kabul edilebileceği sonucuna varılmıştır [4]. Ayrıca artan nüfus ile daha fazla insan elektrikli cihazlara erişim sağlamaktadır ve bu da Türkiye geneli elektrik enerjisi tüketiminde uzun vadede artışa sebep olmaktadır. 2010 – 2021 yılları arasında elektrik enerjisi tüketimi ile nüfus arasındaki ilişki ise Şekil 1.2.'de gösterilmektedir.



Şekil 1.1. Elektrik tüketimi (MWh) [7] ve GSYİH (Milyon Dolar) değerleri [8]



Şekil 1.2. Elektrik enerjisi tüketimi (MWh) [7] ve Türkiye nüfusu [9]

1.3. Türkiye’de Kullanılan Enerji Tahmin Modelleri

1970’lerin sonlarında Türkiye’nin gelecekteki enerji talebini tahmin etmek amacıyla matematiksel modellemenin uygulanması için bazı çabalar gösterilmiş olsa da, Enerji ve Tabii Kaynaklar Bakanlığı (ETKB) tarafından bu tür yöntemlerin enerji planlamasında resmi olarak kullanımı ancak 1984’ten sonra gerçekleştirilmiştir. 1984 yılı, Türkiye’de enerji planlaması ve gelecekteki enerji taleplerinin tahmini için bir dönüm noktası olmuştur. 1984 yılında Dünya Bankası, ETKB’na genel enerji talebinin belirlenmesi için Uluslararası Atom Enerjisi Ajansı (IAEA) tarafından geliştirilen “Enerji Talep Analizi Modeli” (MAED) ve “Viyana Otomatik Sistem Planlama Paketi” (WASP) benzetim modellerini kullanmasını tavsiye etmiştir [112].

MAED, sosyal, ekonomik ve teknolojik altyapılar üzerinde etkisi olan çeşitli etkenleri dikkate alarak, bir ülkenin orta ve uzun zaman diliminde elektrik enerjisi talebini tahmin etmek için faydalanan bir öngörü modelidir. Yani MAED modeli kullanılarak Türkiye’nin genel enerji ve elektrik enerjisi talepleri tespit edilmektedir. MAED modeli yakın gelecekte gerçek değerlerle benzer sonuçlar vermekte, daha uzun zaman dilimlerinde ise büyük farklılıklar göstermektedir. Bu sapmaların sebebi, MAED tarafından yapılan talep tahminlerinde Devlet Planlama Teşkilatı’nın (DPT) geleceğe yönelik demografik ve ekonomik parametre tahminleri de bulunmasıdır. WASP ise, Türkiye Elektrik İletim A.Ş. (TEİAŞ) tarafından uzun dönemli üretim planlamalarını gerçekleştirmek amacıyla kullanılan bir tahmin modelidir [6].

Enerji piyasalarının etkin, şeffaf, güvenilir ve sürdürülebilir biçimde işletilmesi ve geliştirilmesi amacıyla 18 Mart 2015 tarihinde Enerji Piyasaları İşletme A.Ş. (EPIAŞ) kurulmuştur. TEİAŞ tarafından kullanılan elektrik enerjisi talep tahmin algoritması ile her gün, bir sonraki gün için saatlik bazda Türkiye'nin kısa vadeli elektrik enerjisi yük tahmini yapılmakta ve EPIAŞ'a iletilmektedir. EPIAŞ, söz konusu tahminleri her gün çevrim içi şeffaflık raporu olarak yayınlamaktadır [18]. Elektrik enerjisi üreticileri ise bu tahminleri değerlendirerek bir sonraki gün için üreteceği elektrik enerjisine hangi saatlerde ihtiyaç duyulacağını tespit etmektedir. Bu sayede elektrik enerjisi üreticileri, yayınlanan yük tahminine göre üretim planlaması yapmaktadır.

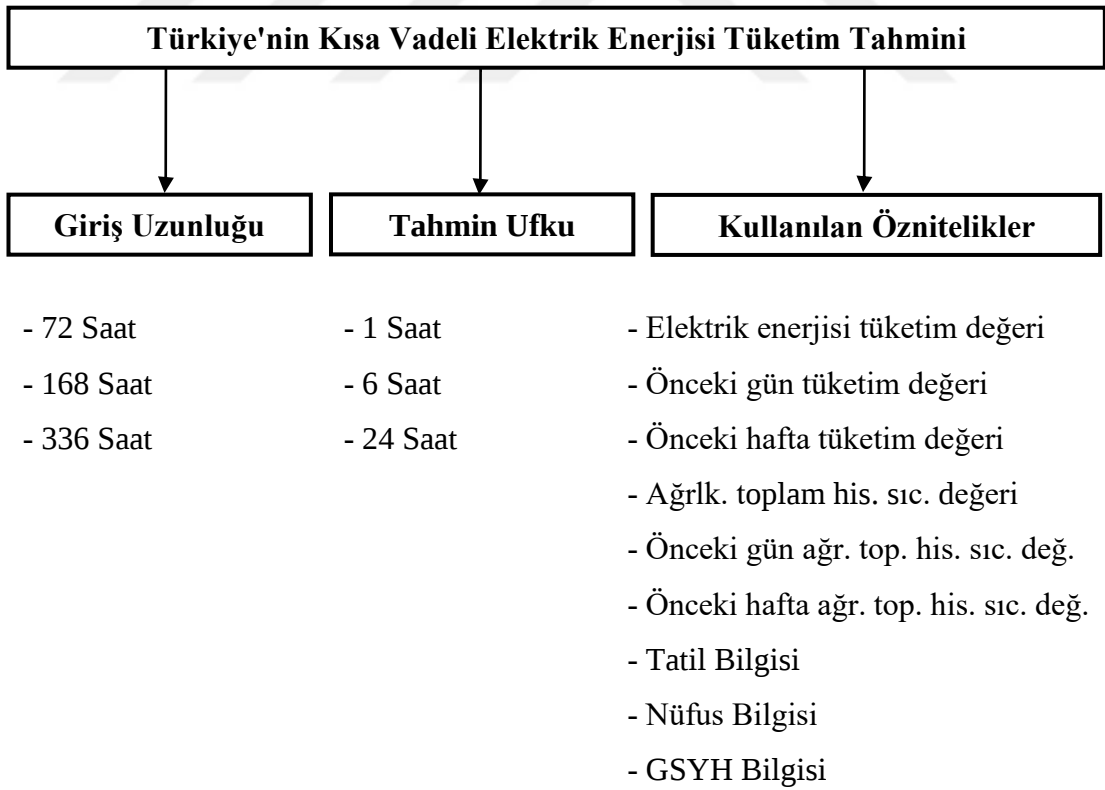
1.4. Zaman Serisi Problemi Olarak Elektrik Enerjisi Tüketim Tahmini

Zaman serisi, eşit zaman aralıklarında kaydedilen sıralı değerler dizisidir. Zaman serisi analizi, iki alt bileşenden oluşmaktadır. İlk kısım, gözlemlenen verileri ve temel modeli elde etmektir. İkinci kısım ise geleceğe yönelik tahminler yapmak için bir modelin oluşturulmasıdır. Zaman serilerinin tahmini, zamansal bir bileşeni olan hava durumu tahmini, elektrik enerjisi tüketim tahmini, anormallik tespiti ve trafik tahmini gibi çok çeşitli gerçek hayat problemlerinde önemli bir rol oynar. Elektrik enerjisi tüketim değerleri zamana dayalı bileşenlere sahiptir bu nedenle elektrik enerjisi tüketim tahmini bir zaman serisi problemi olarak ele alınmaktadır. Elektrik enerjisi tüketiminin, geçmiş ve şimdiki zaman değerleri arasında karmaşık bir korelasyon olması ve altında yatan dinamiklerin fazlalığı, elektrik enerjisi tüketim tahminini zorlu bir problem haline getirmektedir [12]. Elektrik enerjisi tüketim tahmini doğrusal olmayan bir problem olarak sınıflandırılmaktadır. Sonuç olarak, doğrusal regresyon analizi gibi parametrelerin modellenmesi üzerine kurulu teknikler kullanarak, gelecekteki elektrik enerjisi tüketimini tahmin etmek zor bir süreçtir.

Elektrik enerjisi tüketim tahmini, tahmin vadesi ve tahmin amacı olmak üzere iki farklı kritere göre sınıflandırılır. Tahmin vadesi, tahmin edilecek zaman periyodu olarak tanımlanmaktadır ve çoğu kaynakta üç farklı kategoriye ayrılmıştır. Çok kısa vadeli yük tahmini (VSTLF), saniye veya dakikadan birkaç saate kadar periyoda sahiptir ve akışı kontrol etmek için kullanılır. Kısa vadeli yük tahmini (STLF), saatlerden haftalara kadar uzanan zaman ufkundan oluşmaktadır. Bu sayede gerçekleştirilen tahminler ile talep ayarlanarak, elektrik piyasasına teklifler sunma

amacıyla kullanılmaktadır. Orta vadeli ve uzun vadeli yük tahmini (MTLF ve LTLF) ise aylar ve yıllar arası periyotlardan oluşmaktadır. Genellikle yeni tesisler kurulması planlanırken kullanılmaktadır [13]. Yapılan tez çalışmasında: 1,6 ve 24 saatlik tahmin vadeleri kullanılmıştır ve söz konusu tahmin vadeleri kısa vadeli elektrik enerjisi tüketim tahmin sınıfını kapsamaktadır.

Tahmin amacı, tahmin edilecek değerin sayısı olarak tanımlanmaktadır. Tahmin amacında iki çeşit çıktı hedeflenir: ilk çeşit, bir sonraki saatin yükü, bir sonraki günün pik yükü, bir sonraki günün toplam yükü gibi tekil değerden oluşur; ikinci çeşit ise bir sonraki günün saatlik yükü veya birden fazla tahmin edilecek parametre gibi çoklu değere sahip tahminlerden oluşur. Yapılan tez çalışmasında: 1 saatlik tahminin gerçekleştirildiği modeller yalnızca bir değerden oluşan gruba dâhil olurken, 6 ve 24 saatlik tahminin gerçekleştirildiği modeller ise çoklu değerden oluşan gruba dâhil olmaktadır. Tez çalışmasında kullanılan giriş uzunluğu, tahmin ufku ve öznitelikler Şekil 1.3.'de verilmiştir.



Şekil 1.3. Tez çalışmasında kullanılan özellikler

1.5. STLF Modelini Oluştururken Karşılaşılan Zorluklar

Kısa vadeli bir yükün, geçmiş yük verileri, rüzgar hızı, yağış, atmosfer basıncı, sıcaklık ve nem gibi birçok faktörden etkilenen bir değişken olması sebebiyle yüksek performanslı bir STLF modeli oluşturmak için bazı hususların dikkate alınması gerekmektedir. Verilerdeki çeşitli doğrusal olmayan özellikler karmaşıklığa sebep olmakta ve bu yüzden sağlam bir STLF modeli oluştururken sorunlara yol açmaktadır. Bu karmaşıklık, ilk önce tahmin için doğru modeli bulmanın ve ikinci olarak bu modelin gerekli parametrelerini optimal bir şekilde ayarlamamanın önünde bir engeldir. Örneğin, SARIMA, ARIMA ve SARFIMA gibi klasik doğrusal modellerin kullanılması uygun hiperparametre kümelerini bulmak için büyük veri ön işleme gerektirmektedir. Bu yüzden, en iyi parametre kombinasyonlarını seçmek zor olmaktadır [48].

1.6. Derin Öğrenme Modelleri İle Elektrik Enerjisi Tüketim Tahmini

Elektrik enerjisi tüketim tahmini için önerilen metotlar; regresyon tabanlı yöntemler, Box-Jenkins metodolojisi olarak bilinen ARIMA modelleri, zaman serisi yaklaşımları, kalman filtreleri, uzman sistem teknikleri, sinir ağı ve bulanık mantık modellerine dayanmaktadır. ARIMA modelleri, zaman serisi tahmin tekniklerinin temel ve en genel şeklidir. Bunlar, zaman serilerini fark alma işlemi ile durağan hale getirme (düşük frekanslı bileşeni kaldırma) fikrine dayanmaktadır. Bu modeller, geçmişteki gözlemlerden doğrusal fonksiyonlar oluşturarak gelecek tahminlerinin yapılmasına olanak sağlar [14]. Zaman serisi için ARIMA denklemi, girdi değişkeninin (y) gecikmeleri ile tahmin hatasının (e) gecikmelerinden oluşan bir denklemdir ve Denklem (1.1)'de gösterilmektedir [17].

Çıkış 'y'

$$\begin{aligned} &= \text{sabit} + 'y' \text{değişkeninin geçmiş değerlerinin ağırlıklı toplamı} \\ &+ 'e' \text{değişkenine ait geçmiş değerlerin ağırlıklı olarak toplamı.} \end{aligned} \quad (1.1)$$

Gerekli ön işleme yapılmadan uygulamaya geçirildiğinde, istatistiksel teknikler genel olarak gerekli performansı gösterememektedirler. Örnek olarak, ARIMA modelinde durağan olmayan bir zaman serisi eğilim ve mevsimsellikten arındırılarak durağan hale getirilmesi gerekliliği verilebilir. Diğer bir konu ise, söz konusu modeller büyük miktarda veri ile başa çıkmada zorlanırken veri azlığında ise

özellik ve kalıpları çıkarmada başarısız olmaktadır. Ayrıca, güncel çalışmalar incelendiğinde, doğrusal yöntemlerin performans kıyaslaması amacıyla kullanılmaya devam ettiği görülmektedir.

Geleneksel istatistiksel yöntemlere dayalı modeller, veri kümesi içindeki yalnızca doğrusal ilişkiye sahip yapıları değerlendirmektedirler. Bunun aksine, derin öğrenme modelleri ise doğrusal olmayan ilişkiye sahip yapıları da ele alabilmektedirler. Bu yüzden derin öğrenme teknikleri, elektrik enerjisi tüketim tahmini gibi birçok makine öğrenimi problemi için popüler yaklaşım haline gelmiştir [15]. Derin öğrenme teknikleri, formülasyon olarak kolay bir problemdir fakat birden fazla katman istiflenerek oluşturulan derin yapıları sebebiyle karmaşıktır. Bilgi sistem teknolojilerinin gelişmesiyle birlikte daha derin modeller oluşturulabilmekte ve derin öğrenme modellerinin başarıları giderek artmaktadır. Klasik modellerin aksine doğrudan veriye adapte olma ve önceki değerlere ihtiyaç duymama kapasiteleri sebebiyle derin öğrenme modelleri, zaman serisi problemleri için büyük avantaj sağlamaktadır [16]. Yapılan tez çalışmasında, literatürde kullanılan en ilgili derin öğrenme ağ türleri incelenmektedir. Söz konusu ağ türleri üç kategoriye ayrılmaktadır:

- Tam bağlantılı sinir ağları.
 - Çok Katmanlı Algılayıcı (MLP)
- Tekrarlayan sinir ağları.
 - Elman Tekrarlayan Sinir Ağı (ERNN)
 - Uzun Kısa Süreli Bellek (LSTM)
 - Yankı Durumu Ağı (ESN)
 - Kapılı Tekrarlayan Birimler (GRU)
 - Kodlayıcı-Kod Çözücü İstiflenmiş Uzun Kısa Süreli Bellek (Encoder-Decoder Stacked LSTM)
- Evrişimli ağlar.
 - Evrişimli Sinir Ağı (CNN)
 - Zamansal Evrişimli Ağ (TCN)

Yukarıda verilen derin öğrenme ağ türleri, farklı mimari varyantlarla tezde kullanılan veri setine başarılı bir şekilde uygulanmıştır. Söz konusu uygulamalar gerçekleştirilirken uygulanan yöntem ve elde edilen sonuçlar detaylı olarak ileriki bölümlerde bahsedilmektedir.

2. LİTERATÜR ÖZETİ

Kısa vadeli yük tahmini, enerji sistemlerinin hem ekonomik hem de güvenli bir şekilde işletilmesinde, orta ve uzun vadeli yük tahmini ise enerji sistemlerini doğrudan veya dolaylı yoldan ilgilendiren planlamalarda önemli bir rol oynar. Söz konusu her iki kategori de araştırmacıların ilgisini çekmiştir. Bu yüzden literatürde elektrik enerjisi tüketim tahmini ile ilgili birçok çalışma bulunmaktadır. Ayrıca, çözülen problem hakkında çok az bilgi gerektirmesi, kolay uygulanması ve matematiksel bir modele gerek olmaması gibi avantajlarından dolayı elektrik enerjisi tüketim tahmini problemi için derin öğrenme temelli yaklaşımların sayısı giderek artmaktadır. Bu bölümde, ilgili çalışmalar dört gruba ayrılmaktadır. Birinci gruptaki çalışmalar, Türkiye'nin kısa vadeli elektrik enerjisi tüketim tahminine, ikinci gruptaki çalışmalar Türkiye'nin orta ve uzun vadedeki elektrik enerjisi tüketim tahminine, üçüncü gruptaki çalışmalar da Türkiye'den farklı olarak ele alınan kısa vadeli elektrik enerjisi tüketim tahmini üzerine odaklanmaktadır. Dördüncü gruptaki çalışmalar ise tekrarlayan sinir ağı modelleri ile gerçekleştirilen STLF temelli çalışmaları genel kapsamlı olarak ele almaktadır.

2.1. Türkiye'nin Verileri İle Sunulmuş STLF Çalışmaları

Erkmen ve Özdoğan (1997), Türkiye'de ilk olarak YSA yaklaşımı kullanarak Türkiye'nin saatlik yük tahminini gerçekleştiren STLF sistemini geliştirmiştir. Bu çalışmada giriş değişkeni olarak saatlik yük ve takvim verileri kullanılmış ancak sıcaklık verileri giriş değişkeni olarak dahil edilmemiştir. Teknik, Modifiye Kohonen algoritması (MKA), genetik algoritma (GA) ve üç katmanlı geri yayılım (BP) ağı gibi üç temel modülden oluşmaktadır. Önerilen tekniğin amacı, STLF için kümeleme işlemi ve genetik algoritmanın avantajlarını birleştirmektir. Geleneksel geri yayımlı modelin MAPE (ortalama mutlak yüzde hata) ölçümü %2.76, genetik olarak bulunmuştur. Optimize edilmiş geri yayımlı modelin MAPE ölçümü %2.64, kümeleme sonrası geri yayımlı modelin MAPE ölçümü %1.89 ve kümeleme işleminden sonra genetik olarak optimize edilmiş geri yayımlı modelin MAPE ölçümü ise %0.96 olarak elde edilmiştir. GA'nın kümeleme işleminden önce kullanılmasının, kümeleme işleminden sonra kullanılmasından daha az etkili olduğu, büyük verilerde GA'nın yetersiz kaldığı, etkili rasgele sayı üreticilerinin

kullanılmasının özellikle büyük veri tabanları için GA'nın arama kabiliyetini geliştireceği gibi sonuçlar elde edilmiştir [21].

Topalli ve ark. (2006), Türkiye'nin toplam elektrik yükünü bir gün önceden tahmin etmek için geleneksel regresyon temelli yaklaşımlara alternatif olarak yapay zeka yaklaşımı sunmuştur. Çalışmada, hem ileri besleme hem de geri beslemeye sahip bağlantılarıyla öğrenmeyi kolaylaştıran ve tekrarlayan sinir ağlarının bir alt sınıfı olan Elman ağı kullanılmıştır. Geçmiş verilerden yararlanmak ve anlık tahminler yapmak için çevrimdışı eğitim ile gerçek zamanlı öğrenmeyi birleştiren bir hibrit öğrenme algoritması önerilmiştir. Önerilen hibrit öğrenme ile model, önce mevcut çevrimdışı verilerle eğitilir ve gerçek zamanlı uygulama sırasında şartlara göre model ağırlıkları güncellenir. Bu sayede sonuçlar standart öğrenme yöntemlerinden daha iyi hale gelmektedir. Önerilen yöntem, özellikle tatiller için her zaman daha düşük yüzde hata vermektedir. 2002 yılı için ortalama hata %1,60 olarak elde edilmiştir. Söz konusu yöntemin, STLF için klasik yaklaşımların yerine geçmesi öngörülmüştür [20].

Bilgic ve ark. (2010), bölgesel saatlik yük, bölgesel ağırlıklı sıcaklık ve takvim verileri ile ileri beslemeli yapay sinir ağı (YSA) yaklaşımı kullanarak Türkiye'nin dokuz yük dağıtım bölgesi için dokuz STLF modeli geliştirmiştir. YSA modelleri MetrixND yazılımı kullanılarak geliştirilmiş olup, geri yayılım (BP) öğrenme algoritması ile eğitilmiş ve gizli katman için doğrusal ve sigmoid aktivasyon fonksiyonları seçilmiştir. Toplam saatlik elektrik yükünün MAPE değeri, Türkiye veri kümesi için %1.81 olarak elde edilmiştir. Söz konusu STLF problemi için basit YSA modelleri kullanılmasına rağmen, bu çalışmanın MAPE değerleri, karmaşık yaklaşımlar kullananlarınkilerle kıyaslandığında daha iyi bir aralıkta olduğu değerlendirilmektedir [111].

Filik ve ark. (2011), kısa, orta ve uzun vadeli zaman ufkunda, Türkiye'nin saatlik olarak yük tahminini gerçekleştirmek için yeni bir matematiksel yöntem sunmuştur. Tahmin alanındaki önceki çalışmalara benzer bir şekilde, önerilen yöntem gelecekteki değerleri tahmin etmek için zaman serisinin geçmiş değerlerini kullanmaktadır. Önerilen yöntem, modelleme için üç alt bölümün iç içe geçmiş bir kombinasyonundan oluşmaktadır. Modelin ilk kısmı, yıllık ortalama yüklerdeki artıştan oluşmaktadır. İkinci kısım, bir yıl içinde haftalık artış yük değişimlerinin

modellenmesini yönetmektedir. Üçüncü kısım ise iç içe birleştirme yöntemindeki en yüksek çözünürlüğü sağlayan saatlik modelleme kısmıdır. Diğer saatlik tahmin yöntemlerinden farklı olarak, gerçekleştirilen sistem birkaç yıl ilerisi için tahminler yapabilmekte ve bu da onu yük talep tahmini alanında benzersiz kılmaktadır [22].

Çevik ve Çunkaş (2015), bulanık mantık ve uyarlamalı ağ tabanlı bulanık çıkarım sistemi (ANFIS) kullanarak Türkiye için kısa vadeli elektrik yük tahmin modeli geliştirmiştir. Girdi olarak tarihsel elektik yükü, sıcaklık farkı ve mevsimler tercih edilirken çıktı olarak elektrik yükü tahmin edilmiştir. Genel literatürde, bulanık mantık ile saatlik yük tahminleri birkaç gün veya birkaç hafta aralığında test edilmektedir fakat daha önceki çalışmalardan farklı olarak söz konusu çalışma ile saatlik yük tahmini 1 yıl için yapılmaktadır. Bulanık mantık ve ANFIS modellerinin tahmin doğruluğu açısından MAPE değerleri sırasıyla %2.1 ve %1.85 olarak elde edilmiştir. Elde edilen sonuçlar değerlendirildiğinde, bulanık mantık tekniğine dayalı yöntemle kıyasla, ANFIS modelinin tahmin performansı açısından daha iyi MAPE değerine sahip olduğu tespit edilmiştir. Ayrıca, bulanık mantığın 1 yıl gibi çok büyük test veri setlerinde iyi sonuçlar verebileceği tespit edilmiştir. Elde edilen sonuçlar değerlendirildiğinde, önerilen bulanık mantık ve ANFIS modellerinin gerçek verilere çok yakın değerler üretilip verimli bir şekilde yük tahmini yapılabilmesi sebebiyle Türkiye'nin kısa vadeli yük tahmin problemi için alternatif bir yol olduğu görülmektedir [23].

Esener ve ark. (2015), meteorolojik veriler olmadan 24 saat ilerisi için Türkiye'nin kısa vadeli elektriksel yük tahminini incelemiştir. TEİAŞ'tan alınan 1 Aralık 2008 ile 31 Aralık 2010 tarihleri arasındaki elektriksel yük verileri kullanılmıştır. Söz konusu çalışmada, YSA, dalgacık dönüşümü (WT) ve YSA, WT ve radyal tabanlı yapay sinir ağı (RBFNN) ve ampirik mod ayrıştırma (EMD) ve RBFNN yapıları kullanılmıştır. Tahmin hata yüzdeleri, MAPE olarak hesaplanmış ve bu 4 yapı arasında karşılaştırılmıştır. Çalışmada kullanılan tüm yöntemlerden elde edilen sonuçlar değerlendirildiğinde, hava durumu ile ilgili veriler kullanılmadan da tahmin doğrulukları dikkate değerdir. MAPE değerlerine göre, WT ve RBFNN ve EMD ve RBFNN'nin meteorolojik veriler olmadan Türkiye için tahmin yapmak amacıyla kullanılabileceği sonucuna varılmıştır [24].

Yasin ve ark. (2015), Türkiye'nin kısa vadeli elektrik yükünü tahmin edebilen ileri beslemeli bir yapay sinir ağı önermiştir. Veri seti olarak günlük elektrik tüketim ve sıcaklık değerleri kullanılmış olup, Türkiye elektrik piyasası için ertesi günün saatlik yükü tahmin edilmiştir. Ek olarak, yapay sinir ağının mevsimsel tahmin doğruluğu destek vektör makinesi ile karşılaştırılmıştır. Yapay sinir ağı, kış ve bahar verilerinde daha iyi performans gösterirken, SVM tahminlerinin yaz ve sonbahar verilerinde daha doğru olduğu tespit edilmiştir. Ayrıntılı performans değerlendirmesi, tahmin yönteminin saatlik yükü doğru bir şekilde tahmin edebildiğini göstermektedir [26].

Tanıdır ve Tör (2015), bir yapay sinir ağı (YSA) ile Türkiye'nin güç sistemi için bir gün öncesi yük tahmin (DALF) modeli geliştirmiştir. Ulusal ve dini günleri içeren özel tatillerin etkileri ve ark ocağı yükleri nedeniyle Türkiye'nin güç sisteminde gözlemlenen saatlik rastgele yük sapmaları tartışılmıştır. Geliştirilen YSA'nın girdilerini azaltmak ve modeli iyileştirmek için sıcaklık tahminleri açısından en duyarlı şehirler kullanılmıştır. Kullanılan şehirler, yoğun saatlerde sistem yüküne etkileri dikkate alınarak belirlenmiştir. Sonuçlar, iyi eğitilmiş bir YSA'nın, özel olmayan günlerde, yaklaşık %1 gibi tatmin edici günlük MAPE'ler sağlayabilmesine rağmen, özel işlem gerektiren büyük rastgele yük bozulmaları nedeniyle saatlik mutlak yüzde hataların (APE) daha yüksek olduğunu göstermiştir [27].

Yukseltan ve ark. (2017), harmonikleri ve günlük periyodik değişimlerin mevsimsel değişimlerle modülasyonunu hesaba katan doğrusal bir model kullanarak, Türkiye'nin yıllık, haftalık ve günlük tahmin ufkunda saatlik bir talep tahmin yöntemi geliştirmiştir. Model, herhangi bir meteorolojik veya ekonometrik değişken kullanmadan saatlik olarak değişimleri tahminlemektedir. Ayrıca, sadece geçmiş değerlere sahip tarihsel veriler kullanılması sebebiyle, diğer ülkeler için uygulanarak mevsimsel değişimler yakalanabilmektedir. Çalışmada kullanılan model, Türkiye Elektrik piyasasına uygulanarak, herhangi bir iklim bilgisi olmadan tüm yıl boyunca talebi %3 MAPE değeriyle tahmin etmeyi başarmıştır. Söz konusu modelin, başka bir ülkeye uygulanırken iyi bir sonuç elde edebilmesi için ulusal bayramlar, seçimler ve yarışmalar gibi düşük ve yüksek talep dönemlerinin modele dâhil edilmesi gerektiği sonucuna varılmıştır [25].

Tokgöz (2018), yüksek lisans tez çalışmasında tekrarlayan sinir ağlarının alt konusu olan geleneksel tekrarlayan sinir ağları (RNN), uzun kısa süreli bellek ağı (LSTM) ve kapılı tekrarlayan birimler (GRU) yöntemlerini kullanarak Türkiye'nin kısa vadeli elektrik enerjisi tüketim tahmini üzerine deneyler gerçekleştirmiştir. Eğitim seti olarak 2010 yılından 2017 yılına kadar Türkiye'nin elektrik enerjisi tüketim verileri kullanırken, test seti olarak ise Türkiye'nin 2017 yılındaki elektrik enerjisi tüketim verileri kullanılmıştır. Performans ölçütü olarak ise ortalama mutlak yüzde hata (MAPE) değeri dikkate alınmıştır. Yapılan deneyler karşılaştırıldığında, en yüksek performansı GRU ve ardından yakın bir değere sahip LSTM gösterirken, RNN ağının 3 yöntem arasından en kötü performansa sahip olduğu ölçülmüştür. Önerilen toplam sıcaklık değerinin tüketim tahminini gerçekleştirirken önem teşkil ettiği sonucuna varılmıştır. Ayrıca, modellerin tatil günlerindeki elektrik enerjisi tüketimini tahmin etmede başarısız olduğu sonucuna varılarak, söz konusu başarısızlık için vekil veri veya tahmin rötuşlanması gibi önerilerde bulunulmuştur [19].

Permata ve ark. (2021), 1 Ocak 2016 ile 12 Eylül 2020 tarihleri arasındaki Türkiye'nin saatlik elektrik enerjisi tüketim verilerini kullanarak Türkiye'deki elektriksel yük karakteristiğini günlük, haftalık, mevsimsel ve takvimsel değişim etkisine göre belirlemek için bir analiz sunmuştur. Ayrıca, dinamik harmonik regresyon (DHR), hibrit zaman değiştirme rölesi ile DHR (hybrid TSR-DHR) ve çift bütünleşik otoregresif hareketli ortalama (DSARIMAX) modellerini kullanarak Türkiye'nin kısa vadeli elektrik yükü tahminini gerçekleştirmiştir. Takvimsel özelliklerin modellere dâhil edildiğinde doğruluğun yükseldiği görülmüştür. DHR yönteminin uzun mevsimsel kalıpları yakalamada başarısı tespit edilmiştir. Ayrıca hibrit modelin, eğilimlere ek olarak mevsimsel ve takvimsel özellikleri yakalamada yetenekli olduğu sonucuna varılmıştır [28].

Ünlü (2022), Türkiye'nin 5 Ocak 2015 ile 26 Aralık 2021 arasındaki elektrik enerjisi tüketim verilerini kullanarak büyük miktarda doğrusal olmayan veriyi işleme kapasitesine sahip LSTM, GRU ve CNN gibi derin öğrenme algoritmaları ile kısa ve orta vadeli elektrik enerjisi kullanımını tahmin etmiştir. Bilindiği kadarıyla söz konusu çalışma Türkiye'nin kısa vadeli ve orta vadeli algoritmalarını karşılaştıran ilk araştırma çalışmasıdır. Test setindeki tahmin sonuçları, en iyi performansın LSTM tarafından elde edildiğini göstermiştir [38].

2.2. Türkiye'nin Verileri Sunulmuş İle MTLF ve LTMF Çalışmaları

Yumurtacı ve Asmaz (2004), Türkiye'nin 2050 yılına kadar elektrik enerjisi talebinin ve hidrolik enerji üretiminin projeksiyonunu değerlendirmiştir. Öngörülen değerler, geçmiş yıllardan günümüze kadar olan nüfus ve kişi başına düşen enerji tüketim verilerden yola çıkılarak istatistiksel bir teknikle hesaplanmıştır. Hesaplanan bu değerlere göre 2050 yılında Türkiye'nin nüfusunun yaklaşık 115 milyon olacağı ve o yıla ait elektrik enerjisi ihtiyacının 1.173.000 GWh olacağı tahmin edilmiştir. Türkiye'nin enerji ihtiyacının her yıl artacağı ve 2021 yılı için elektrik enerjisi ihtiyacının yaklaşık 240.000 GWh olacağı öngörülmüştür [31]. 2021 yılına ait elektrik enerjisi tüketim miktarı yaklaşık 329.600 GWh'tir [30]. Sonuç olarak söz konusu çalışma, 2021 yılına ait Türkiye'nin elektrik enerjisi tüketim değerini yaklaşık %27 oranında eksik tahmin etmiştir.

Erdoğdu (2006), Türkiye'nin gelecekteki net elektrik enerjisi tüketimini tahmin etmek için bir bütünleşik otoregresif hareketli ortalama (ARIMA) modeli geliştirmiş ve kullanmıştır. Ayrıca, elektrik enerjisi talebindeki büyümeyi tahmin etmek ve sonuçları resmi tahmin ile karşılaştırmayı amaçlamıştır. Değişken olarak, elektrik enerjisi fiyatlarını, kişi başına düşen gayri safi yurtiçi hasılayı ve kişi başına düşen elektrik enerjisi tüketimini kullanmıştır. Çalışma, 2005 yılından 2014 yılına kadar ARIMA modellemesine dayalı olarak, talebin yıllık ortalama %3,3 oranında artmaya devam edeceğini ve 2014 yılında 160.090 GWh'ye (Gigawatt saat) ulaşacağını göstermektedir. Resmi tahmin ise 2014 yılı elektrik enerjisi tüketimini 256.644 GWh olarak tahmin etmiştir [29]. 2014 yılındaki resmi verilere göre Türkiye'nin net elektrik enerjisi tüketimi 257.200 GWh'tir [30]. Resmi tahmin ile söz konusu çalışmanın tahmin değeri kıyaslandığında, resmi tahminin gerçek değere çok daha yakın olduğu görülmektedir.

Hamzaçebi (2007), sanayiye dayalı elektrik enerjisi tüketimini, 2020 senesine kadar Türkiye için tahminini sunmuştur. Aynı anda birden fazla değişkenin gelecekteki değerlerini tahmin edebilmesi ve veri yapısındaki doğrusal olmayan ilişkiyi modelleyebilmesi sebebiyle çalışmada tahmin aracı olarak YSA tercih edilmiştir. Çoğu diğer çalışma gibi YSA tarafından kurulan tahmin sonuçları resmi tahminlerle karşılaştırılmıştır. Söz konusu çalışma 2020 yılı için Türkiye'nin net elektrik enerjisi tüketim tahminini 499.588 GWh olarak elde etmiştir. 2020 yılında

gerçekleşen net elektrik enerjisi tüketimi 290.856 GWh'tir [30]. Çalışmanın tahmin değeri ile 2020 yılı için açıklanan Türkiye'nin elektrik enerjisi tüketim değeri kıyaslandığında arada ciddi fark olduğu görülmektedir.

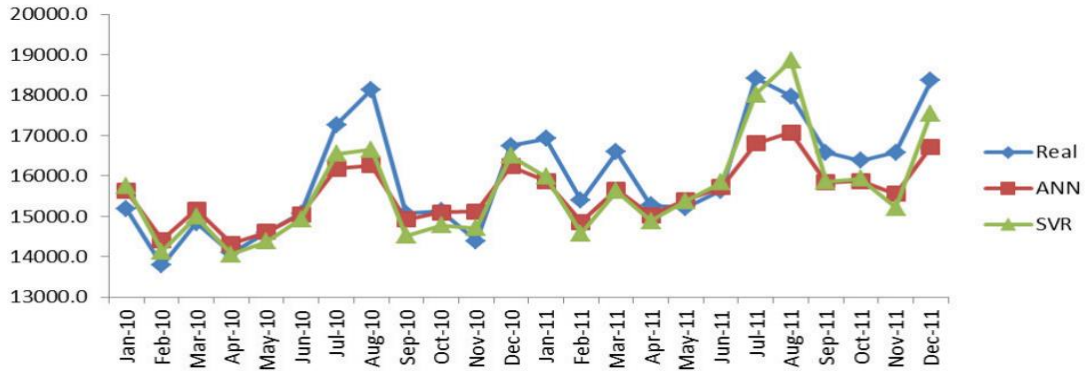
Kavaklıoğlu ve ark. (2009), Türkiye'nin 1975 ile 2006 yılları arasındaki elektrik tüketimi ve diğer ekonomik verilerini kullanarak 2027 yılına kadar elektrik tüketimini modellemek ve tahmin etmek için geri yayılım algoritmali YSA sunmuştur. Ayrıca, tahminler geleceğe doğru hareket ettikçe zaman serisi modellerinin gelecek tahminine olan güvenin azalacağı varsayılarak YSA modeli önerilmiştir. Elektrik enerjisi tüketimi tahmin edilirken nüfus (popülasyon) ve ekonomik etkenler gibi parametreler kullanılmıştır. Farklı modeller arasındaki performans karşılaştırması MAPE ölçütüne göre değerlendirilmiştir. 2021 yılı için elektrik enerjisi ihtiyacının yaklaşık 245.440 GWh olacağı öngörülmüştür [32]. Söz konusu çalışma, 2021 yılına ait Türkiye'nin elektrik enerjisi tüketim değerini yaklaşık %25 oranında eksik tahmin etmiştir. Kavaklıoğlu (2011), gerçekleştirdiği diğer bir çalışmada ise benzer özellikler ile SVR metodolojisi kullanmıştır. 2021 yılı için elektrik enerjisi ihtiyacının yaklaşık 233.200 GWh olacağı öngörülmüştür [33]. Söz konusu çalışma, 2021 yılına ait Türkiye'nin elektrik enerjisi tüketim değerini yaklaşık %29 oranında eksik tahmin etmiştir.

Bilgili ve ark. (2012), girdi olarak kurulu güç, toplam elektrik enerjisi üretimi, abone sayısı ve nüfus gibi değişkenler kullanarak Türkiye'nin konut ve sanayi sektörleri için elektrik enerjisi tüketim tahminini sunmuştur. Ayrıca söz konusu tahmin işlemi gerçekleştirirken, YSA yöntemi ile doğrusal ve doğrusal olmayan regresyon tabanlı modeller uygulanmıştır. Gelecekteki elektrik tüketiminin tahmin etmek için güçlü ve zayıf olmak üzere iki farklı senaryo önerilmiştir. Sanayi sektörü elektrik enerjisi talebi için elde edilen eğitim çıktılarından bazıları ve gerçek değerler Tablo 2.1'de gösterilmiştir. Çalışmanın genel sonuçları değerlendirildiğinde, YSA yönteminin performans değerlerinin, LR ve NLR modellerinin performans değerlerinden daha iyi olduğu görülmüştür [34].

Tablo 2.1. Referanstaki [34] gelecek projeksiyonları ve gerçek değerler [35] (TWh)

Yıl	Güçlü Senaryo			Zayıf Senaryo			Gerçek Değer
	LR	NLR	ANN	LR	NLR	ANN	
2008	83.55	78.33	81.74	83.25	78.07	81.54	74.85
2009	88.05	82.17	85.92	84.46	79.15	83.50	70.47
2010	101.25	93.32	95.91	85.52	80.05	85.55	79.33
2011	119.20	108.60	108.48	93.66	86.95	92.01	87.96
2012	142.55	128.52	123.19	113.04	103.03	106.13	92.30
2013	155.04	139.18	130.29	123.08	111.40	113.41	93.25
2014	168.41	150.62	136.95	131.28	118.28	119.10	97.78
2015	181.76	162.02	142.73	140.16	125.67	124.85	103.54

Oğcu ve ark. (2012), Türkiye'nin 2010 ile 2011 yılları arasındaki elektrik enerjisi tüketimini aylık olarak tahmin etmek için YSA ve mevsimsel bir destek vektör regresyonu (SVR) önermiştir. Gerçek tüketim değerleri, ANN ve SVR kıyaslaması görsel olarak Şekil 2.1'de verilmiştir. Çalışmada, mevsimsel SVR modelinin YSA modelinden daha iyi performans gösterdiği tespit edilmiştir [37].



Şekil 2.1. Referans çalışmadan [37] elde edilmiş tüketim tahmin grafiği

Yetiş ve Jamshidi (2014), YSA'yı tahmin aracı kullanarak Türkiye'nin enerji sektörü bazında 2023 yılına kadarki elektrik enerjisi tüketim tahminini araştırılmıştır. 2023 yılında Türkiye'nin net elektrik tüketim değeri toplamda 373,09 GWh olarak öngörülmüştür. 2021 yılı için ise Türkiye'nin net elektrik tüketim değeri toplamda 340,32 GWh öngörülerek gerçek değerden yaklaşık %3 fazla tahmin edilmiştir [36].

2.3. Farklı Ülke ve Şehirlerin Verileri İle Sunulmuş STLF Çalışmaları

Tian ve ark. (2018), İtalya'nın kuzey bölgesini kapsayan elektrik tüketim veri seti kullanılarak tahmin doğruluğunu iyileştirmek için CNN modelini ve LSTM modelini entegre eden yeni bir derin sinir ağı yaklaşımı önermiştir. Tahmin performansı LSTM modeli ve CNN modeli ile karşılaştırıldığında, önerilen modelin STLF'de daha iyi ve istikrarlı performans elde edebileceği görülmüştür [49].

López ve ark. (2019), İspanya ulusal sistemine ait elektrik tüketim verilerini kullanarak, tatiller veya bayram dönemleri gibi sosyal olayların yük eğrisi üzerindeki etkisini doğrusal regresyon ile modellemiştir. Bu çalışmanın sonuçları kendisinden daha basit olan sürümleriyle karşılaştırıldığında, takvimsel özelliklerin tüketici davranışı üzerindeki çok çeşitli etkilerini doğru bir şekilde modellemek için iyi bir sınıflandırma sisteminin uygulanması gerektiğini göstermektedir. Ayrıca, önerilen yöntemin Fransa, Portekiz ve İtalya gibi benzer sistemlerde faydalı olacağı belirtilmiştir [50].

Zhang ve ark. (2019), güneş paneli kurulumundaki üstel artışın elektrik yükünü doğru bir şekilde tahmin etmeyi zorlaştırması sebebiyle Güney Kaliforniya'daki saatlik elektrik yükünü gün öncesinden tahmin etmek için operasyonel olarak kullanılabilecek bir STLF modeli geliştirmiştir. Bu çalışma, çoklu doğrusal regresyon (MLR), rastgele orman (RF), ve gradyan artırmaya (GB) dayalı üç farklı model tipini karşılaştırarak operasyonel olarak kullanılabilecek bir STLF modeli belirlemeyi amaçlamıştır. Ortalama hata değerleri, rastgele orman için %3.5 olarak elde edilmiştir. Ayrıca söz konusu hata değerleri, çoklu doğrusal regresyon için %3.4 ve gradyan artırma için ise %3.1 olarak elde edilmiştir. Genel olarak, GB modeli en iyi performansı göstermiştir. İlgili çalışmanın sonucu olarak, hava durumuna dayalı en önemli faktörün hava sıcaklığı olduğu ve takvimsel özelliklerin meteorolojik olmayan en önemli değişken olduğu değerlendirilmiştir [40].

Ribeiro ve ark. (2020), düşük hesaplama maliyeti ve güçlü performansa sahip yankı durumu ağı (ESN) kullanarak STLF görevini gerçekleştirmiştir. Ayrıca, ESN'nin uygun hiperparametrelerini bulmak için Gauss sürecine dayalı bayes optimizasyon algortiması (BOA) araştırılmıştır. ESN modelinin etkinliğini test etmek için biri Polonya'dan diğeri Brezilya'dan olmak üzere iki veri seti kullanılmıştır. Deney

sonuçları, kısa vadeli yük tahminin kalitesi açısından doğru hiperparametreler kullanıldığında düşük hata tahminleri elde edilebileceğini göstermiştir [44].

Eskandari ve ark. (2021), İngiltere ve Kuzey Amerika'nın saatlik veri setlerini kullanarak hava durumu, mevsim, takvim gibi birçok dış değişkenler dikkate alınarak saatlik yük tüketimi için yüksek doğrulukta bir tahmin modeli sunmuştur. Yük ve sıcaklık öznitelik çıkarımı için CNN'ler, saatlik elektriksel yük tahmini yapmak için ise çift yönlü GRU ve LSTM yaklaşımları kullanılmıştır. Önerilen modelin geleneksel GRU ve LSTM modellerinden daha iyi performans gösterdiği ve evrişimli katmanlarla öznitelik çıkarımının RNN'ler tarafından tahmin doğruluğunu arttırdığı tespit edilmiştir [46].

Farsi ve ark. (2021), Malezya'nın saatlik yük tüketimini ve Almanya'nın günlük elektrik tüketimini veri seti olarak kullanılarak, paralel LSTM-CNN ağı veya PLCNet olarak adlandırılan yeni bir hibrit derin öğrenme modeli ile STFL gerçekleştirmiştir. Sonuçlar, PLCNet'in Alman verileri için doğruluğu %83,17'den %91,18'e yükselttiği ve Malezya verilerinde %98,23 doğruluk elde ettiğini göstermiştir. Çalışma, söz konusu sonuçları mükemmel olarak tanımlamıştır [47].

2.4. RNN, CNN ve TCN İle Gerçekleştirilmiş Diğer STLF Çalışmaları

Han ve ark. (2018), CNN ve LSTM tabanlı modeller kullanarak Çin'in bir şehri olan Hangzhou ve Kanada'nın en büyük şehri olan Toronto için kısa vadeli yük tahmini gerçekleştirmiştir. TD-CNN yapısı, uzak geçmişten gelen etkiyi yakalamak için kullanılırken, C-LSTM yapısı ise uzun zaman adımlarında LSTM'nin kısıtlamalarının üstesinden gelmek için kullanılmıştır. Deneysel sonuçlar, önerilen yöntemlerin eğitim sürelerinin mevcut yöntemden (LSTM) yaklaşık iki ila beş kat daha kısa olduğunu ve büyük ölçüde azaltılmış hesaplama karmaşıklığı ile mevcut yöntemden daha iyi performansa sahip olduğunu göstermiştir [43].

Muzaffar ve Afshari (2019), uzun süreli bağımlılıkları öğrenme yeteneğine sahip LSTM ağını kullanarak sıcaklık, nem ve rüzgâr hızı gibi dışsal değişkenlere sahip elektrik enerjisi talebini 24 saat, 48 saat, 7 gün ve 30 gün zaman ufuklarında tahmin etmiştir. LSTM ağının performansı ARMA, SARIMA ve ARMAX gibi geleneksel yöntemlerin çıktıları ile karşılaştırılmıştır. Sonuçlar değerlendirildiğinde,

LSTM'nin geleneksel modellere kıyasla, uygun veriler verildiğinde mevsimselliği ve eğilimleri kolay öğrenebildiği için daha iyi performans gösterdiği tespit edilmiştir. Ayrıca, LSTM'nin karmaşık denklemler geliştirmek ve üzerinde çalışmak yerine sağlanan bilgileri kolayca ele alma konusunda basit ve yalnızca kısa vadeli değil, aynı zamanda uzun vadeli tahmin için geleneksel yöntemlerden daha iyi olduğu sonucuna varılmıştır [39].

Wu ve ark. (2020), STLF'nin daha iyi doğruluğu için, kapılı tekrarlayan birim (GRU) ve evrişimli sinir ağlarını (CNN) birleştiren bir GRU-CNN hibrit sinir ağı modeli önermiştir. Önerilen model, güç sistemini etkileyen değişken verilerden öznitelikler çıkararak elektrik yükünü hızlı ve doğru bir şekilde tahmin etmiştir. Gerçekleştirilen modeller değerlendirildiğinde, Şekil 2.2'de gösterildiği üzere en iyi tahmin performansı GRU ve CNN ile uygulanan hibrit modelden elde edilmiştir [42].

Test	BPNN (%)	GRU (%)	CNN (%)	GRU-CNN (%)
Test avg	4.4681	4.6377	3.3890	2.8839

Şekil 2.2. Referans çalışmadaki [42] modellerin ortalama MAPE değerleri

Lara-Benítez ve ark. (2020), tahmin performansını iyileştirmek için uzun vadeli kalıpları çıkarabilen ve hesaplama süresi açısından daha verimli olan TCN tabanlı bir derin öğrenme modeli önermiştir. İspanya'dan gelen ulusal elektrik talebi ve elektrikli araçlar için şarj istasyonlarındaki güç talebi zaman serisi problemi olarak incelenmiştir. TCN modeli ile LSTM tekrarlayan sinir ağının kapsamlı bir karşılaştırması yapıldığında, TCN yaklaşımının tahmin doğruluğu açısından daha iyi performans gösterdiği ve LSTM ağlarına çok güçlü bir alternatif olduğu tespit edilmiştir [45].

Guo ve ark. (2021), SVM, RF ve LSTM gibi üç makine öğrenme yöntemini kullanarak hem kısa vadeli yük tahmini gerçekleştirmiş hem de bu yöntemleri inceleyip karşılaştırmıştır. Ayrıca, tahmin doğruluğunu kapsamlı bir şekilde geliştiren füzyon tahmin yaklaşımı ve bir veri ön işleme tekniği önerilmiştir. SVM, RF, LSTM ve füzyon algoritmasının MAPE değerleri sırasıyla; %23.4, %3.6, %5.3 ve %2.8 olarak elde edilmiştir. Kullanılan yöntem için, füzyon algoritmasının tahmin doğruluğunun herhangi bir geleneksel makine öğrenme yönteminden daha yüksek ve LSTM'nin tahmin doğruluğunun hem SVM'ninkinden hem de RF'ninkinden daha yüksek olduğu sonucuna varılmıştır [41].

2.5. Tezin Amacı Ve Katkıları

Yapılan bu tez çalışması, standart derin öğrenme mimarilerini kullanarak Türkiye'nin kısa vadeli elektrik enerjisi tüketimini tahmin etmeyi odak noktasına almaktadır. Tez;

- Elektrik enerjisi tüketim tahmini için en uygun derin öğrenme teknikleri hakkında kapsamlı bir inceleme,
- Türkiye'yi kapsayan veri kümesi üzerinde birkaç derin öğrenme mimarisinin performansını değerlendiren karşılaştırmalı bir analiz,
- Türkiye'nin elektrik tüketimini etkileyen dinamikler hakkında araştırma,
- Ağırlıklandırılmış toplam hissedilen sıcaklık değişkeninin tahmin performansı üzerindeki etkisini inceleyen deneyler,
- Çeşitli ağ varyasyonlarının tahmin performansı üzerindeki etkisini inceleyen bir çalışma sunar.

Literatürde daha önce gerçekleştiren çalışmalar incelendiğinde bulunan sonuçların klasik modeller ile karşılaştırıldığı görülmüştür. Gerçekleştirilen tez çalışması, standart derin öğrenme mimarilerini kendi aralarında karşılaştırarak araştırmacılar için güvenilir bir kıyaslama kaynağı ve elektrik enerjisi tahmin problemlerine nasıl yaklaşılabileceği konusunda genel kılavuz sağlamaktadır.

3. ZAMAN SERİSİ TAHMİNİ İÇİN DERİN ÖĞRENME MİMARİLERİ

Bir zaman serisi, sabit zaman aralıklarında kaydedilen ve kronolojik sırada olan bir gözlem dizisidir. Geçmiş değerler göz önünde bulundurarak zaman serisinin gelecekteki değerlerini tahmin etmek amacıyla bir model kullanmak, tahmin problemini ifade etmektedir. $X = \{x_1, x_2, \dots, x_T\}$ bir zaman serisinin geçmişe ait değerleri ve H 'ın zaman serisi probleminde kullanılacak tahmin ufku olarak ele alınsın. Bu durumda istenilen, söz konusu $\{x_{T+1}, \dots, x_{T+H}\}$ serisinin daha sonraki değerlerini tahmin etmektir. $\hat{X} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_T\}$ tahmin edilen değerlerin vektörü olduğundan, amaç denklem (3.1)'deki tahmin hatasını en aza indirmektir [51].

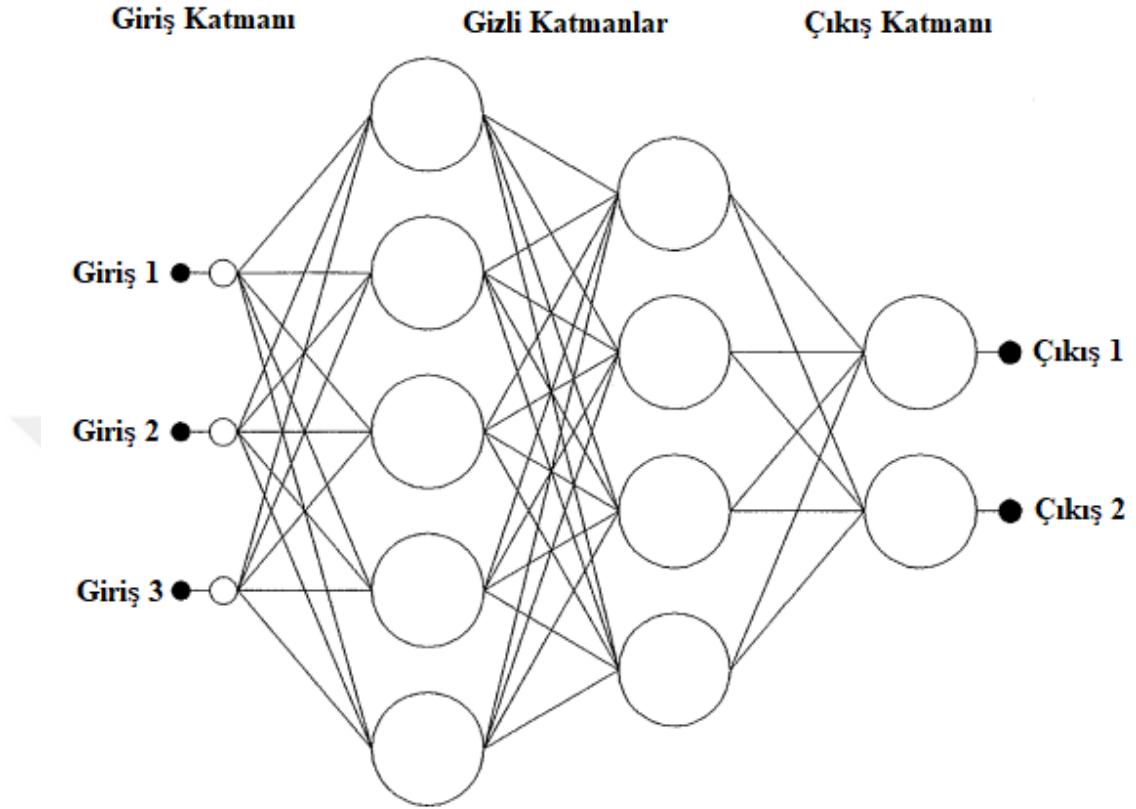
$$\text{Tahmin Hatası} = \sum_{i=1}^{h=H} |x_{T+i} - \hat{x}_{T+i}| \quad (3.1)$$

Söz konusu çalışma prensibine sahip derin öğrenme mimarileri, girdi verilerindeki gürültüye karşı dayanıklıdır ve eksik değerlerin varlığında öğrenmenin yanı sıra tahmini de destekleyebilirler. Daha önce de bahsedildiği gibi doğrusal ve doğrusal olmayan ilişkileri kolayca öğrenebilirler. Ayrıca, isteğe bağlı sayıda girdi veya çıktı değeri sağlayarak çok değişkenli tahminler için kullanılabilirler. Özellikle CNN gibi mimariler, ham girdi verilerinden otomatik olarak öznetelik çıkartımı yapabilirler. LSTM gibi mimariler ise doğal dil işleme gibi problemler için yüksek performans gösterebilirler. Bu bölümde, söz konusu avantajlarından dolayı tez çalışmasında kullanılan mimariler detaylı olarak incelenmiştir.

3.1. Çok Katmanlı Algılayıcı (Multi-Layer Perceptron)

YSA kavramı ilk olarak Warren ve ark. (1943) [52] tarafından beynin işleyişi ilham alınarak ortaya atılmıştır. MLP, ileri beslemeli yapay sinir ağının en temel türüdür ve mimarileri, Şekil 3.1.'de gösterildiği üzere bir giriş katmanı, gizli katmanlar ve bir çıkış katmanı olmak üzere üç çeşit yapıdan oluşur. Şekilde 2 gizli katma sahip bir yapı sunulmuştur. Giriş katmanı hiçbir hesaplama rol oynamaz ve yalnızca giriş vektörünü ağı iletmeye hizmet eder. MLP ağının derinliği, sahip olduğu gizli katman sayısı ile doğru orantılıdır. Sinir ağı giriş ile çıkış arasındaki ilişkiyi çözebilmek için her katmanda bulunan nöron kümeleri arasındaki bağlantıların ağırlıklarını günceller.

Ayrıca, YSA'ların en uygun yapısını ve hiperparametrelerini belirlemek performans üzerinde önemli bir etkiye sahiptir ve zor bir süreçtir [53].



Şekil 3.1. Çok katmanlı bir algılayıcı [53]

YSA'ları modelleme aracı olarak kullanmanın bazı avantajları olduğu kadar dezavantajları da vardır [55]. Avantajları:

- Doğrusal bir teknik kullanılarak çözülemeyen problemler ile sinir ağı çoğu zaman başa çıkabilir.
- Bir sistemi temsil eden modelin geliştirilme ihtiyacını ortadan kaldırır.
- Sinir ağının bir elemanı başarısız olduğunda, ağın paralel yapısı nedeniyle sorunsuz bir şekilde çalışmaya devam edebilir.
- Bir sinir ağı öğrenir ve yeniden programlanması gerekmez.

YSA'ları modelleme aracı olarak kullanmanın dezavantajları:

- Mikroişlemci mimarileri ile kıyaslandığında, bir sinir ağının farklı yapısı sebebiyle uygun bir şekilde taklit işleminin gerçekleştirilmesi gerekmektedir.
- Sinir ağı, iç işleyişlerinden bağımsız olarak sadece girdi ve çıktı olarak görüntülenen bir sistem (black box) olması sebebiyle, nedensel ilişkileri açıkça tespit etme açısından kısıtlı bir yeteneğe sahiptir.
- Derin sinir ağlarını eğitebilmek için yüksek işlem süresine ihtiyaç duyulmaktadır.

MLP ağında bir düğümün çıktısı, ağın bir sonraki katmanındaki düğümlere girdi olmaktadır. Bu, bilgi işleme yönü anlamına gelmektedir ve MLP'ler, ileri beslemeli bir sinir ağı olarak bilinir. MLP gibi ileri beslemeli sinir ağları birçok durumda etkili bir şekilde uygulanmış olsa da, her girdiyi bağımsız olarak ele aldıkları için bir zaman serisinin zamansal sırasını yakalayamazlar. Bu nedenle, zaman serisi problemlerinin çözümünde performansı arttırmak için RNN veya CNN gibi daha özel ağlar kullanılmaktadır [54].

3.1.1. Geri Yayılım (Backpropagation) Algoritması

Geri yayılım, gradyan iniş (gradient descent) kullanılarak yapay sinir ağlarının denetimli öğrenmesini sağlayan bir algoritmadır. İlk olarak 1970'lerde genel bir optimizasyon yöntemi olarak icat edilmiştir. İlerleyen yıllarda, Rumelhart ve ark. (1986) tarafından yayınlanan bir makale ile makine öğrenimi topluluğu tarafından algoritmanın önemi artmıştır [56]. Hata fonksiyonun gradyanı, sinir ağının ağırlıkları temel alınarak söz konusu teknik ile hesaplanmaktadır. Gradyan hesaplamasının ağ üzerinde geriye doğru ilerlemesi sebebiyle geri yayılım olarak tabir edilir. Geri yayılım, parametre güncellemesi görevi için kazandırdığı zaman sebebiyle günümüze kadar birçok uygulamanın çözümünde kullanılmıştır.

Geri yayılım, üç temel bileşene ihtiyaç duymaktadır [56]:

1. Giriş ve çıkış çiftlerinden $(\vec{x}_i, \vec{y}_i)$ oluşan bir veri seti. Burada, $\vec{x}_i$ girdi ve $\vec{y}_i$ ise ağın $\vec{x}_i$ girdisindeki istenilen çıktısıdır. N boyutundaki giriş ve çıkış çiftleri $X = \{(\vec{x}_1, \vec{y}_1), \dots, (\vec{x}_N, \vec{y}_N)\}$ şeklinde ifade edilir.
2. Parametreleri θ (theta) ile gösterilen ileri beslemeli bir sinir ağı. Geri yayılımda, w_{ij}^k (l_k katmanındaki j düğümü ile l_{k-1} katmanındaki i düğümü arasındaki ağırlık değeri) ve b_i^k (l_k katmanındaki i düğümü için eşik yani bias değeri) temel parametrelerdir.
3. İstenilen $\vec{y}_i$ çıktısı ile hesaplanan $\hat{\vec{y}}_i$ arasındaki gerçekleşen hatayı temsil eden ve $E(X, \theta)$ ile gösterilen bir hata fonksiyonu.

Geri yayılım algoritması aşağıdaki beş denkleme bağlıdır:

$$\frac{\partial E_d}{\partial w_{ij}^k} = \delta_j^k o_i^{k-1} \quad (3.2)$$

$$\delta_1^m = g'_o(a_1^m)(\hat{y}_d - y_d) \quad (3.3)$$

$$\delta_j^k = g'(a_j^k) \sum_{l=1}^{r^{k+1}} w_{jl}^{k+1} \delta_l^{k+1} \quad (3.4)$$

$$\frac{\partial E(X, \theta)}{\partial w_{ij}^k} = \frac{1}{N} \sum_{d=1}^N \frac{\partial}{\partial w_{ij}^k} \left(\frac{1}{2} (\hat{y}_d - y_d)^2 \right) = \frac{1}{N} \sum_{d=1}^N \frac{\partial E_d}{\partial w_{ij}^k} \quad (3.5)$$

$$\Delta w_{ij}^k = -\alpha \frac{\partial E(X, \theta)}{\partial w_{ij}^k} \quad (3.6)$$

Kısmi türevler için Denklem (3.2), son katmanın hata terimi için Denklem (3.3), gizli katmanların hata terimleri için Denklem (3.4), her giriş ve çıkış çifti için kısmi türevleri birleştirmek amacıyla Denklem (3.5) ve ağırlıkları güncellemek için Denklem (3.6) kullanılmaktadır. Daha önce açıklananlara ek olarak denklemlerde, E_d (hata fonksiyonu), δ_j^k (k katmanındaki ve j düğümündeki hata terimi), o_i^{k-1} (k-1

katmanındaki ve i düğümündeki çıktı), δ_1^m (son katman yani m 'deki hata terimi), $g'_o(a_1^m)$ ve $g'(a_j^k)$ (aktivasyon fonksiyonu temelli doğrusal olmayan ölçekleme), $\hat{y}_d$ (ağın çıktı değeri), y_d (istenilen çıktı değeri) ve α (öğrenme oranı) terimleri kullanılmaktadır.

Geri yayılım algoritması, uygun bir öğrenme oranı α ve w_{ij}^k parametreleri ile rastgele başlatıldığını varsayarsak aşağıdaki adımlar ile açıklanır:

1. Her giriş ve çıkış çifti için ileri faz hesaplanır ve $\hat{y}_d$, a_j^k (j düğümünün k katmanındaki aktivasyon) ve o_j^k sonuçları saklanır.
2. Her giriş ve çıkış çifti için geri faz hesaplanır ve $\frac{\partial E_d}{\partial w_{ij}^k}$ (bireysel gradyan) sonuçları saklanır.
3. Her giriş ve çıkış çifti için ayrı gradyanlar birleştirilir ve $\frac{\partial E(X, \theta)}{\partial w_{ij}^k}$ (toplam gradyan) elde edilir.
4. Öğrenme oranı ve toplam gradyana göre ağırlıklar güncellenir.

3.2. Tekrarlayan Sinir Ağları (Recurrent Neural Networks)

Tekrarlayan sinir ağı (RNN), zamana bağlı veriler için bir YSA varyantı olarak tanıtılmıştır. RNN'lerin yapısını, doğrusal olmayan dinamiklere sahip ve gizli durum (hidden state) olarak adlandırılan parametreler oluşturmaktadır [58]. Gizli durumlar yapı olarak söz konusu ağın hafızası olarak çalışmaktadır. Gizli katmanın önceki durumuna göre o andaki durumu belirlenmektedir [59]. Bu sayede RNN yapısının, geçmiş ve karmaşık işaretleri, uzun periyotlar boyunca hafızasında tutması ve işlemesi sağlanmaktadır. MLP'ler giriş için kullanılan veri kümesindeki zamansal ilişkileri görmezden gelirken, RNN'ler her bir zaman adımını önceki zaman adımları ile birleştirerek veri kümesindeki zamansal bağımlılıkları modelleyebilmektedirler. [57]. RNN'ler önceki gözlemin tahminle ne kadar alakalı olduğunu öğrenebilirler. Ayrıca, RNN'ler hem girdi ve çıktı arasındaki kalıpları, hem de zaman serisinin gözlemleri arasındaki iç kalıpları da öğrenme yeteneğine sahiptir [63]. Söz konusu özellik sebebiyle RNN'ler, zaman serisi problemlerin çözümünde yaygın olarak kullanılan sinir ağı modelleri olarak değerlendirilmektedir.

RNN'leri modelleme aracı olarak kullanmanın bazı avantajları olduğu kadar dezavantajları da vardır [60]. Avantajları:

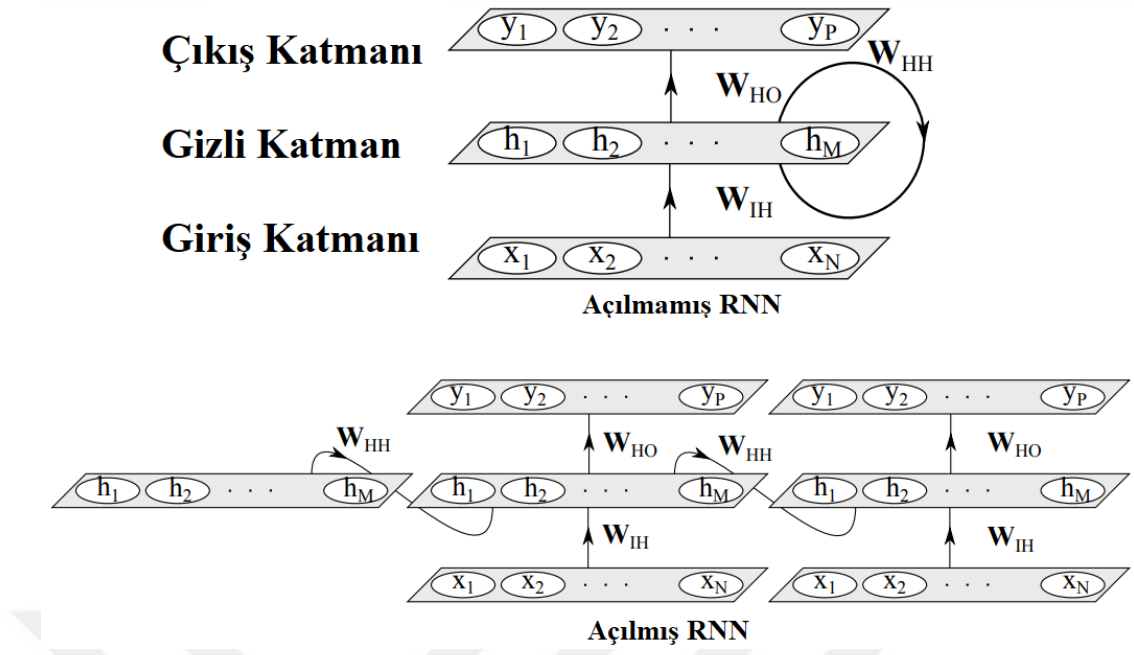
- RNN, dinamik bir sinir ağıdır ve hesaplama açısından güçlüdür bu nedenle birçok zamansal işleme modelinde ve uygulamasında kullanılabilir [61].
- RNN'ler girdi dizilerinden çıktı dizilerine karmaşık eşlemeler gerçekleştirerek doğrusal olmayan sistemlere uygulanabilir [62].
- RNN'in boyutu, ileri beslemeli ağlarla karşılaştırıldığında önemli ölçüde kompakttır [61].
- RNN'ler çağrışımsal bellekler olarak kullanılabilir [61].

RNN'leri modelleme aracı olarak kullanmanın dezavantajları:

- RNN, basit ve güçlü bir model olmasına rağmen pratikte doğru bir şekilde eğitilmesi zordur [64].
- Kaybolan veya patlayan gradyan problemi sebebiyle RNN'ler diğer modellere kıyasla daha hantal (eğitim süresi açısından yavaş) kalmaktadır [65].
- Doğrusal olmayan özelliğe sahip yapısı ve ağırlıkları güncelleme yöntemleri sebebiyle, RNN'lerin ağ kararlılığını tespit etmek zordur [66].

3.2.1. Basit Bir RNN Modelinin Mimarisi

RNN'ler, bir veya daha fazla geri besleme döngüsüne sahip yapay nöronlardan oluşan bir denetimli makine öğrenme modelidir [67]. Geri besleme döngüleri, Şekil 3.2.'de gösterildiği gibi zaman içinde tekrarlayan döngülerdir [68]. Amaç, ağırlıklarını optimize ederek çıktı ve hedef değerleri arasındaki farkı (kayıp değeri) en aza indirmektir. Şekil 3.2.'de gösterildiği üzere basit bir RNN, üç katmandan oluşmaktadır. Bunlar, giriş katmanı, tekrarlayan gizli katman ve çıkış katmanı olarak adlandırılmaktadır. Giriş katmanında N adet giriş birimi vardır. Bu katmanın girdileri, $\{\dots, x_{t-1}, x_t, x_{t+1}\}$ gibi t zamanı boyunca bir vektör dizisidir. Burada kullanılmakta olan, $x_t = (x_1, x_2, \dots, x_N)$ giriş dizisidir. Ayrıca, $h_t = (h_1, h_2, \dots, M)$ ise M adet gizli birimden (hidden unit) oluşmaktadır. RNN'ait giriş birimleri, W_{IH} olarak ifade edilen bir ağırlık matrisi ile gizli katmanda bulunan gizli birimlere bağlanmaktadır.



Şekil 3.2. Basit bir tekrarlayan sinir ağı (RNN) yapısı [68]

Sistemin durum uzayı veya hafızası, gizli katman ile aşağıdaki şekilde tanımlanmaktadır [69]:

$$h_t = f_H(o_t) \quad (3.7)$$

$$o_t = W_{IH}x_t + W_{HH}x_{t-1} + b_h \quad (3.8)$$

Burada; $f_H(.)$ gizli katmanda bulunan aktivasyon fonksiyonudur. Gizli birimlerin eşik vektörü (bias vector) ise b_h ile ifade edilmektedir. W_{OH} ağırlıklı bağlantılar ile gizli birimler çıkış katmanına bağlanmaktadır. Ayrıca, P birime sahip çıktı katmanı $y_t = (y_1, y_2, \dots, y_P)$ şeklinde gösterilir ve şu şekilde hesaplanır:

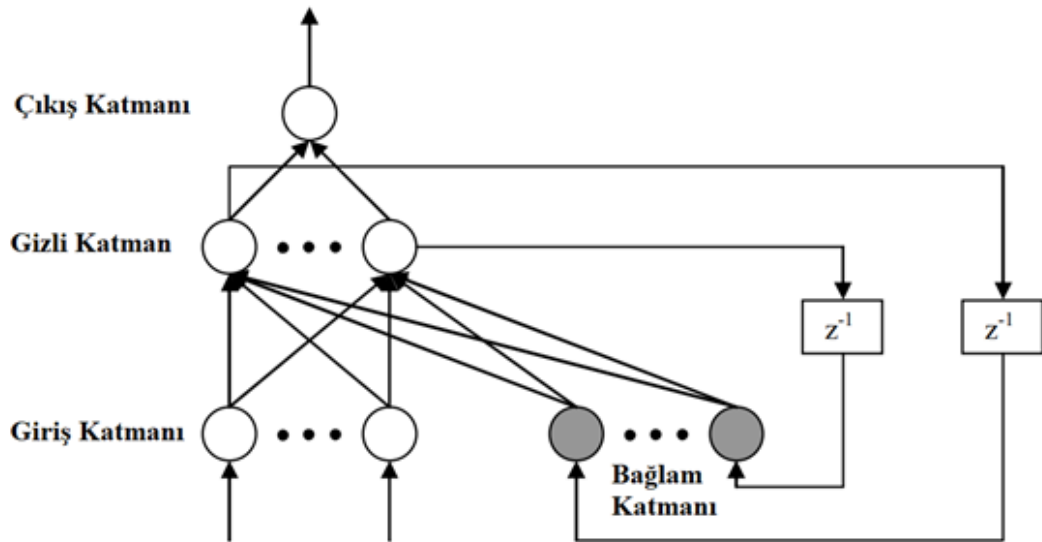
$$y_t = f_o(W_{HO}h_t + b_o) \quad (3.9)$$

Burada; $f_o(.)$ aktivasyon fonksiyonu olarak ifade edilmektedir. Çıktı katmanındaki eşik vektörü ise b_o 'dır. Giriş ve istenilen hedef çiftleri zaman içinde ardışık olarak değerlendirildiğinden, yukarıda gösterilen adımlar $t = (1, \dots, T)$ süresi

boyunca tekrarlanmaktadır. Denklem (3.7) ve Denklem (3.9) bir RNN'nin zaman içinde yinelenen ve doğrusal olmayan durum denklemlerinden oluştuğunu gösterir. Her zaman adımında, gizli durumlar, giriş vektörüne dayalı olarak çıkış katmanında bir tahmin sağlar. RNN yapısında bulunan gizli durum, herhangi bir dış faktör etkisinden bağımsız olarak, ağı geçmişi durumları hakkında ihtiyaç duyulan tüm bilgileri özetleyen bir değerler kümesi olarak değerlendirilmektedir. Bahsedilen bu bilgi ile ağı gelecekteki davranışını belirlenerek çıktı katmanında doğru tahminler üretilebilmektedir [70].

3.2.2. Elman Tekrarlayan Sinir Ağları (Elman Recurrent Neural Networks)

RNN'ler, en az bir geri besleme döngüsü olması bakımından ileri beslemeli ağ mimarisinden farklıdır. Dolayısıyla bu ağlarda, geri besleme bağlantılarına sahip bir katman olabileceği gibi, bir nöronun çıktısının kendisi için geri besleme olarak kullanılan bir bağlantıya sahip nöronlar da olabilir. RNN'lerdeki geri besleme döngüsünün varlığı, ağı öğrenme kapasitesi üzerinde derin bir etkiye sahiptir [72]. Doğrusal olmayan dinamikler, tekrarlayan bir ağı depolama işlevinde önemli bir role sahiptir. Elman (1990), gerçekleştirdiği bir çalışma ile ileri beslemeli gizli katmanın girişini, bir sonraki zaman adımındaki gizli katman çıkışına bağlayan bir RNN modeli geliştirmiştir [57]. Geliştirilen modelin geri besleme döngüleri, doğrusal olmayan dinamik davranışa sahip ve birim gecikme elemanlarından oluşan belirli bağlantıların kullanımına olanak sağlamıştır. Elman tekrarlayan sinir ağındaki (ERNN) bağlam düğümlerinin bağlantıları, ağı giriş verilerinin geçmişine duyarlı hale getirdiğinden, dinamik sistemlerin modellenmesi için yararlı olmaktadır [71]. Söz konusu ERNN ile verilerdeki zaman kalıplarıyla uğraşma sorununun çözülmesi amaçlanmıştır.



Şekil 3.3. Elman tekrarlayan sinir ağının (ERNN) yapısı [73]

ERNN'nin yapısı Şekil 3.3.'de gösterilmektedir. Birim gecikme, z^{-1} ifadesi olarak şekilde sunulmuştur. ERNN yapısı dört katmanda oluşmaktadır. Bunlar, giriş katmanı, gizli katman, bağlam katmanı ve çıkış katmanıdır. Giriş katmanı, o anki zaman adımında bulunan gerçek giriş birimleri ve önceki zaman adımındaki gizli birim aktivasyonlarının bir benzerini bulunduran bağlam (context) birimleri olmak üzere iki bölümden oluşmaktadır. Tahmin doğruluğunun artması için gizli katmanın yeterli sayıda nörona sahip olması gerekmektedir. Ayrıca, ağdaki ileri beslemeli bağlantıların eğitimi için geri yayılım algoritması kullanılabilir. Elman'ın yaklaşımının başarısına rağmen, zaman boyunca kesilmiş geri yayılım (TBPTT) sebebiyle patlayan gradyan problemi ve uzun vadeli kalıpları öğrenmek için aşırı hesaplama süresi gibi iki temel sorun ortaya çıkmaktadır [73].

3.2.3. Uzun Kısa Süreli Bellek Ağları (Long Short-Term Memory Networks)

Uzun kısa süreli bellek (LSTM), Schmidhuber ve Hochreiter (1997) tarafından ERNN'nin sorunlarına bir çözüm olarak tanıtılmıştır [74]. Bu öncü çalışmadan bu yana, LSTM'ler birçok araştırmacı tarafından değiştirilmiş ve popüler hale getirilmiştir. LSTM'ler, kısa zaman periyoduna sahip kalıpları hafızasında tutarak daha geniş ufuklarda da zamansal bağımlılıkları modelleyebilme yeteneğine sahiptirler. LSTM ağları, ERNN ağlarına göre LSTM bellek hücresi olarak adlandırılan

yapılarıyla farklılık göstermektedirler. LSTM hücreleri, çarpımsal bir giriş kapısı kullanarak bellek birimlerini kontrol etmektedirler. Ayrıca bu çarpımsal giriş kapısı, alakasız pertürbasyonlar gibi gürültüler tarafından hafızasının değiştirilmesini önlemektedir. Çarpımsal bir çıktı ise benzer bir şekilde, diğer hücreleri mevcut bellekte depolanan bozulmalardan korumaktadır. Daha sonra literatürde bulunan başka bir çalışma, LSTM bellek hücresine bir unutma kapısı (forget gate) eklemiştir. [75]. Genellikle LSTM hücresi terimi, bir unutma kapısı olan LSTM'yi belirtmektedir. Bahsedilen bu kapı ile ihtiyaç durumunda, LSTM'nin belleği içerisindeki bilginin sıfırlanmasını sağlanmaktadır. Literatürdeki çalışmalar incelendiğinde, LSTM'nin MLP ve ERNN'ye göre avantajlarının olduğu görülmüştür.

LSTM'leri modelleme aracı olarak kullanmanın hem bazı avantajları hem de bazı dezavantajları bulunmaktadır. Avantajları:

- LSTM'ler kaybolan veya patlayan gradyan problemini çözerek uzun vadeli bağımlılıkları yakalamayı mümkün kılmaktadırlar.
- LSTM'ler bilgi akışını daha iyi düzenleyen kapılardan oluşmaktadır.

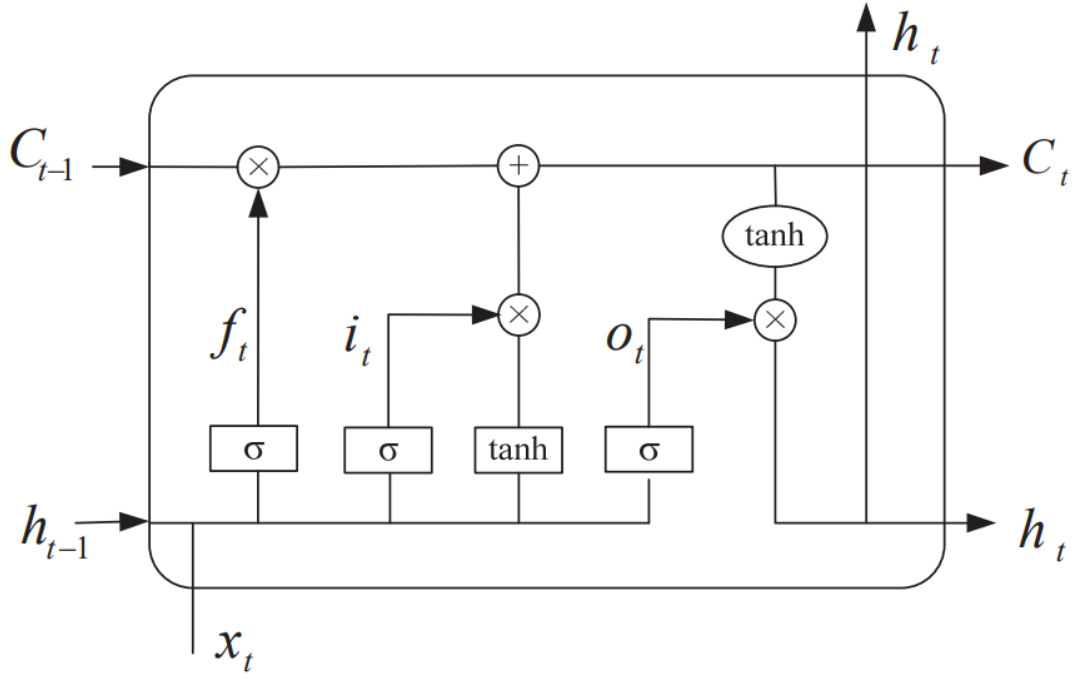
LSTM'leri modelleme aracı olarak kullanmanın dezavantajları:

- Hücreye unutma kapısı gibi ek özelliklerin eklenmesiyle LSTM'ler daha karmaşık hale gelmektedirler.
- Gerçek dünyadaki uygulamaları gerçekleştirebilmek için çok fazla kaynağa ve zamana ihtiyaç duymaktadırlar. Bunun sebebi, her hücrede bulunan doğrusal katmanlar nedeniyle yüksek bellek bant genişliğine ihtiyaç duyulmasındandır. Sonuç olarak, donanım açısından LSTM'ler oldukça verimsizdir.
- Araştırmacılar, geçmişe ait bilgileri LSTM'lerden daha uzun süre boyunca hafızada tutabilme yeteneğine sahip bir model aramaktadırlar. Bu ihtiyaç, veri madenciliği ile ilişkili uygulamaların artmasından kaynaklanmaktadır.
- LSTM'ler farklı rastgele ağırlık başlatmalarından etkilenmektedirler ve bu nedenle ileri beslemeli sinir ağına oldukça benzer davranmaktadırlar.
- LSTM'ler ezberleme veya diğer bir adıyla aşırı öğrenmeye (overfitting) eğilimlidirler. Ezberleme sorununun çözümü için bırakma algoritması (dropout algorithm) kullanılması gerekmektedir ve bu algoritmayı uygulamak zordur.

3.2.3.1 LSTM Mimarisi

Şekil 3.4.'de gösterilen LSTM hücresi incelendiğinde, genel bir LSTM hücresi, bilgi akışını düzenleyen ve kapılar (gates) adı verilen iç mekanizmalara sahiptirler. Bu kapılar sayesinde LSTM'ler, zaman serisindeki hangi verinin hafızada tutulması ve hangi verinin hafızadan atılması gerektiğini öğrenebilmektedir. Yani, LSTM'lerin temel konsepti hücre durumu (cell state) ve çeşitli kapılardır. Hücre durumu, bilgiyi dizi zincirinin sonuna kadar aktaran bir taşıma yolu görevi görmektedir. Bu ağın hafızası olarak düşünülebilir. Hücre durumu, teoride, dizi boyunca ilgili bilgileri taşıyabilmektedir. Bu nedenle, daha önceki zaman adımlarından gelen bilgiler bile daha sonraki zaman adımlarına geçiş yapabilir. Hücre durumu yolculuğuna devam ederken, kapılar aracılığıyla hücre durumuna bilgi eklenir veya çıkarılır.

Kapılar sayesinde hücre durumunda hangi bilgilerin tutulacağına karar verilebilmektedir. Unutma kapısı (forget gate), hangi bilgilerin atılacağına veya saklanacağına karar verir. Önceki gizli durumdan ve şuandaki girişten gelen bilgiler bir sigmoid fonksiyondan geçirilir. Sigmoid fonksiyon çıktı olarak 0 ile 1 arasında değerler elde edilmektedir. Söz konusu değer 0'a yaklaştıkça unutma, 1'e yaklaştıkça tutma işlevi gerçekleşir. Giriş kapısı (input gate), hücre durumunu güncellemek için kullanılır. Önceki gizli durum ve mevcut giriş bir sigmoid fonksiyonundan geçirilerek 0 ile 1 arasında değerler elde edilir. Bu, 0 ile 1 arasında olan değerler dönüştürülerek hangi değerlerin güncelleneceğine karar verilir. Ayrıca, gizli durum ve giriş bilgileri tanh fonksiyonundan geçirilerek, LSTM ağının düzenlenmesi için katkı sağlanmaktadır. Daha sonra tanh fonksiyonunun çıkış değeri ile sigmoid fonksiyonunun çıkış değeri çarpılarak, hangi bilgilerin hafızada tutulmasının daha önemli olduğuna karar verilmektedir. Bir sonraki adıma ait gizli durumun ne olacağına ise çıkış kapısı (output gate) karar vermektedir [76].



Şekil 3.4. Uzun kısa süreli belleğin (LSTM) yapısı [76]

LSTM nöral ağının matematiksel modeli aşağıdaki denklemlere göre tanımlanır [77]:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (3.10)$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (3.11)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (3.12)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (3.13)$$

$$h_t = o_t \tanh(c_t) \quad (3.14)$$

Burada; i , f , o ve c sırasıyla, giriş kapısı, unutma kapısı, çıkış kapısı ve hücre durumunu temsil etmektedir. W ağırlık katsayısı matrisini, σ ve $\tanh$ terimleri ise sırasıyla, sigmoid ve hiperbolik tanjant aktivasyon fonksiyonunu temsil etmektedir.

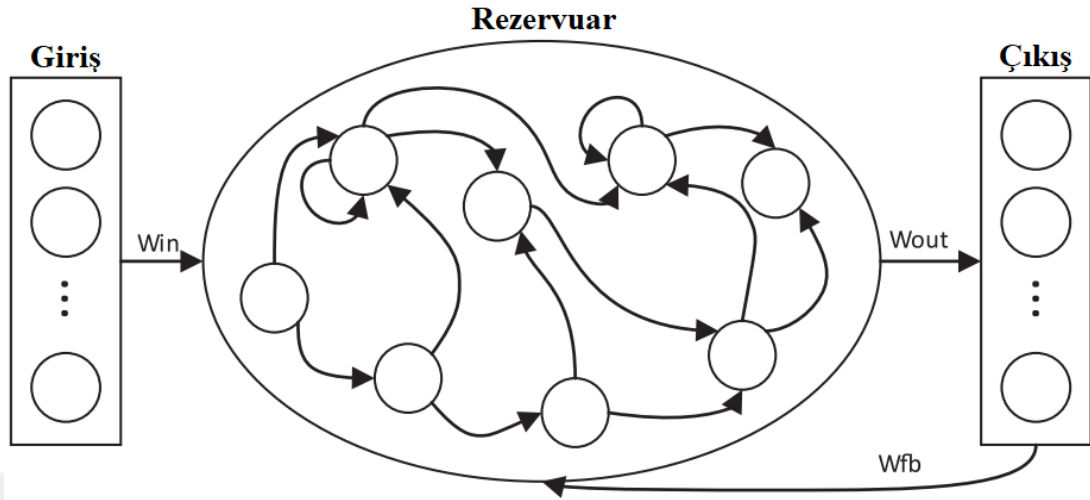
3.2.4. Yankı Durumu Ağları (Echo State Networks)

Jaeger (2001), geleneksel RNN'lerin eğitim yöntemini daha basitleştirerek rezervuar hesaplamaya (reservoir computing) dayalı bir ağ mimarisi sunmuştur [78]. Sunulan bu ağ, yankı durumu ağları (ESN'ler) olarak adlandırılmaktadır ve sadece çıktı nöronlarından gelen ağırlık değerlerini güncelleyerek zaman serisi problemini basit bir doğrusal regresyon problemi haline getirmektedir. Daha önce bahsedilen ERNN ve LSTM ağları bu ağın aksine, tüm nöronlar için en iyi ağırlık değerlerini hesaplamaktadır [79]. ESN'lerde gizli katman yerine rezervuar olarak adlandırılan bir yapı bulunmaktadır. Bu yapı, rastgele bir RNN'e sahip sinir ağından oluşmaktadır. Rezervuar genellikle yüksek sayıda nörona ve gizli nöronlar arasında da seyrek ara bağlantıya (yaklaşık %1) sahiptir. Bu özellik, rezervuarı, belirli zamansal kalıpları yeniden üretebilen bir dizi alt sistem yapar [80]. MLP'ler ve daha önce bahsedilen istatistiksel yöntemler ile kıyaslandığında, ESN'ler kaotik yapıya sahip zaman serisi verilerini modellerken daha iyi performans göstermektedir. Ayrıca, eğitilebilir olmayan rezervuar nöronlarının bu ağı diğer RNN'lere kıyasla, eğitim süresi açısından çok daha verimli bir model haline getirmektedir.

3.2.4.1. ESN Mimarisi

Genel bir ESN, Şekil 3.5.'de gösterildiği üzere bir girdi katmanı, bir rezervuar katmanı (reservoir layer) ve bir çıktı katmanı içermektedir. Ayrıca ilgili şekilde W_{in} , W_{out} ve W_{fb} matrisleri giriş, çıkış ve geri besleme parametreleridir. W_{state} matrisi ise rezervuara gömülüdür. Ağın kalbi olarak adlandırılan rezervuar katmanı, zengin tekrarlayan bağlantılara sahiptir. Rezervuar, seyrek ve rastgele bağlı nöronlardan oluşmaktadır. Girdi vektörlerine ve rezervuar katmanında bulunan ağırlıklara göre rezervuar katmanının durumları hesaplanmaktadır. Biyolojik olarak ilham alınmış bu model, mimarisi gereği kaybolan gradyan problemini (vanishing gradient problem) ortadan kaldırmaktadır [81]. ESN'nin çıkış katmanı, rezervuar katmanı sinyalleri ile istenen çıkış sinyalini doğrusal olarak birleştirir. Burada ana fikir, basit doğrusal regresyon algoritmaları kullanılarak yalnızca çıktı katmanı ağırlıklarının eğitilmesidir.

Bu sayede ESN, rezervuar katmanı için yüksek performanslı bir matematiksel çerçeve sağlar [82].



Şekil 3.5. Yankı durumu ağının (ESN) yapısı [84]

ESN'nin matematiksel modeli aşağıdaki denklemlere göre tanımlanır [84]:

$$x(t) = f \left(W_{in}u(t) + W_{state}x(t-1) + W_{fb}y(t-1) \right) \quad (3.15)$$

$$y(t) = W_{out}x(t) \quad (3.16)$$

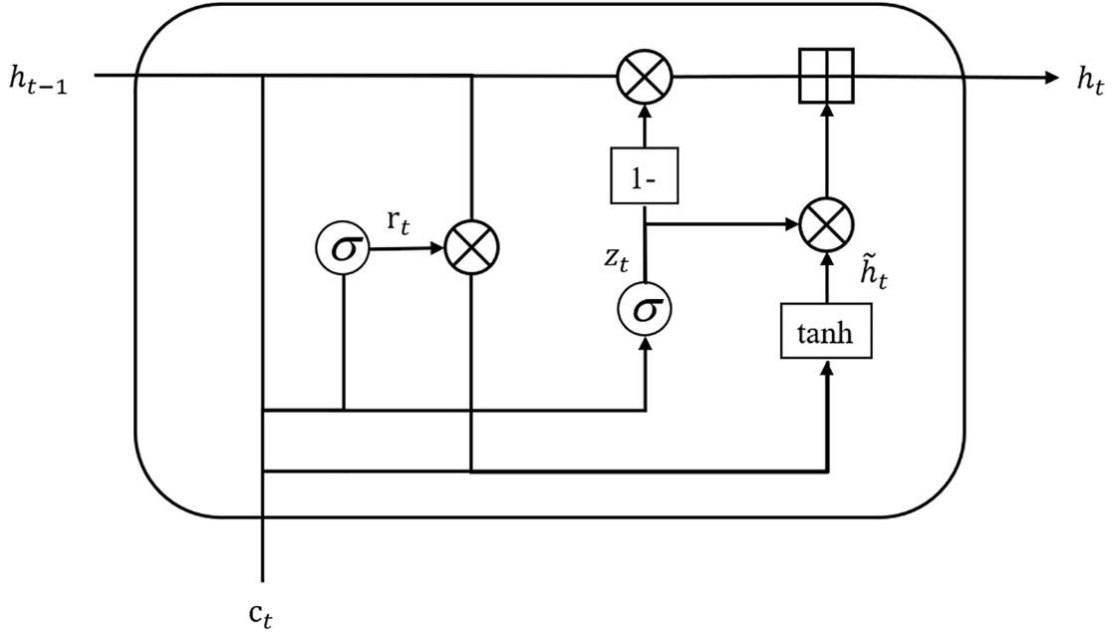
Denklem (3.15) ve Denklem (3.16) incelendiğinde; $u(t)$ her t anı için girdiği, $y(t)$ çıktıyı, $x(t)$ ise rezervuardaki nöronların aktivasyonunu temsil etmektedir. Ayrıca, W_{in} , W_{out} , W_{fb} ve W_{state} sırasıyla; giriş katmanı, çıkış katmanı, geri besleme (feedback) ve rezervuar parametrelerini temsil etmektedir. f ise rezervuardan gelen her bir nöronun aktivasyon fonksiyonudur. Sonuç olarak, ESN'nin çıkışı, rezervuar durumuna doğrusal olarak bağlıdır.

3.2.5. Kapılı Tekrarlayan Birimler (Gated Recurrent Units)

Cho ve ark. (2014), ERNN'lerin patlayan gradyan veya kaybolan gradyan sorunun çözümü için LSTM birimlerinin basitleştirilmiş bir hali olan kapılı tekrarlayan birimleri (GRU) sunmuştur [85]. GRU'da, unutma ve giriş kapısı tek bir güncelleme kapısında (update gate) birleştirilir. Bu değişiklik, eğitilmesi gereken parametre sayısını azaltarak benzer performans için daha kısa hesaplama süresi sağlar [86]. Geçmiş verilere ait bilgilerin ne kadarının geleceğe aktarılması için gerekli olduğuna karar vermekten sorumlu olan kapı güncelleme kapısıdır. Geçmiş verilere ait bilgilerin ne kadarının unutulacağına karar vermekten sorumlu kapı ise sıfırlama kapısıdır (reset gate). LSTM ile karşılaştırıldığında, GRU daha kolay eğitilir ve bu da eğitim verimliliğini büyük ölçüde artırır.

3.2.5.1. GRU Mimarisi

GRU'nun giriş ve çıkış yapısı, sıradan RNN'ninkiyle aynıdır. Geçerli bir x_t girişi ve önceki düğüm hakkında bilgi içeren ve önceki düğüm tarafından aktarılan gizli bir h_{t-1} durumu vardır. h_{t-1} ile birleştirilmiş x_t , h_t çıktısını oluşturmaktadır. LSTM ile karşılaştırıldığında, GRU, Şekil 3.6.'da gösterildiği gibi, sıfırlama kapısı ve güncelleme kapısı olan yalnızca iki kapılı birime sahiptir.



Şekil 3.6. GRU'nun mimari grafiği [87]

GRU'nun matematiksel modeli aşağıdaki denklemlere göre tanımlanır [87]:

$$r_t = \sigma(w_r[h_{t-1}, x_t]) \quad (3.17)$$

$$z_t = \sigma(w_z[h_{t-1}, x_t]) \quad (3.18)$$

$$\tilde{h}_t = \tanh(w_{\tilde{h}}[h'_{t-1}, x_t]) \quad (3.19)$$

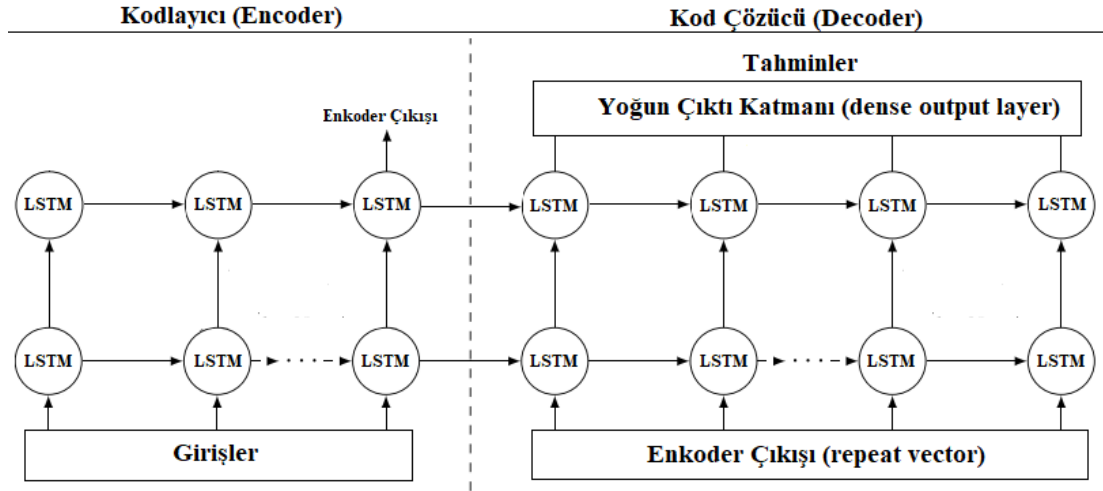
$$h'_{t-1} = h_{t-1}r_t \quad (3.20)$$

$$h_t = z_t\tilde{h}_t + (1 - z_t)h_{t-1} \quad (3.21)$$

Denklem (3.17) ve Denklem (3.18) ile gösterildiği üzere, ilk adımda, son iletilen durum h_{t-1} ve mevcut düğümün girişi x_t aracılığıyla r_t ve z_t elde edilir. r_t sıfırlamanın kapılı kontrolü, z_t güncellemenin kapılı kontrolü, w_r sıfırlama kapısının ağırlığı, w_z güncelleme kapısının ağırlığıdır. σ ile gösterilen terim ise sigmoid fonksiyonudur ve değerler bu fonksiyondan geçerek sıfır ile bir aralığında değerlere dönüştürülmesi sağlanır. Denklem (3.20) ile gösterilen h'_{t-1} sıfırlama verisini (reset data) elde etmek için sıfırlama kapısı olan r_t kullanılır. Veriler, Denklem (3.18) ile tanımlanan tanh aktivasyon fonksiyonu ile -1 ile 1 arasında ölçeklenir. $\tilde{h}_t$ esas olarak x_t girişinin verilerini içerir. Denklem (3.21), bellek güncelleme aşaması olarak adlandırılır ve GRU'daki en kritik adımdır. Daha önceki güncelleme fonksiyonu kullanılmaktadır. Güncellenen geçit sinyali (gated signal) aralığı 0 ile 1 arasındadır. Geçit sinyali 1'e yaklaştıkça daha çok veri hatırlanır ve 0'a yaklaştıkça daha çok veri unutulur.

3.2.6. Kodlayıcı-Kod Çözücü İstiflenmiş Uzun Kısa Süreli Bellek (Encoder-Decoder Stacked LSTM)

Encoder-Decoder LSTM modeli, Chrisman (1991) tarafından bir dizi geçmiş veri ile gelecekteki çıktıları tahmin etmek amacıyla sunulmuştur [88]. Kodlayıcı ve kod çözücü olmak üzere iki kısımdan oluşan model, Şekil 3.7.'de sunulmuştur. Bahsedilen ağın, hem kodlayıcı hem de kod çözücü bölümleri bir LSTM (tekrarlayan sinir ağı) katmanına sahiptir. Bu katmanlar derinleştirilerek istiflenmiş (stacked) hale getirilebilir. Kodlayıcı kısmı, zaman serisi girişini sabit uzunlukta bir vektör ile ifade edilen temsilini öğrenmektedir. Kod çözücü kısmı ise hem mevcut gizli durumu hem de geçmiş (bir önceki) zaman adımında tahmin edilen bilgiyi kullanarak zaman serisini yeniden oluşturmaktadır [89]. Kodlayıcıdan alınan bilgileri, kod çözücüye girdi olarak iletirmek için bağlam vektörü (repeat vector) kullanılır. Kod çözücü kısmında yoğun çıktı katmanı (dense output layer) olarak isimlendirilen ek bir katman bulunmaktadır. Bu katman, tek bir yoğun katmandan oluşabileceği gibi birden çok yoğun katmandan da oluşabilmektedir. Ayrıca bahsedilen bu katman, çıkışın tahmin edilmesi görevinden sorumludur [90].



Şekil 3.7. Encoder-Decoder Stacked LSTM'nin yapısı

3.3. Evriřimli Sinir Aęları (Convolutional Neural Networks)

Hubel ve Wiesel (1968), yaptıkları araştırma ile hayvanlarda ışıęı algılama görevinden, görsel kortekslerinde bulunan hücrelerin sorumlu olduğunu keřfetmiştir [94]. Bu keřiftten ilham alan Fukushima ve Miyake (1982), CNN'nin atası olarak kabul edilebilecek bir yöntem önermiştir [95]. Evriřimli sinir aęları (CNN'ler), canlıların doğal görsel algılama mekanizmasından esinlenerek geliştirilmiş ve evriřimli yapılarıyla verilerden öznitelik çıkarabilen bir tür ileri beslemeli sinir aęlarıdır [91]. Geleneksel öznitelik çıkarma yöntemlerinden farklı olarak, CNN'nin öznitelikleri el ile çıkarmasına gerek yoktur. CNN'ler, yüksek boyuta sahip ve işlem görmemiş yani ham verilerden oluşan özniteliklerin, otomatik olarak çıkartma işlemini yapabilmektedirler. CNN'ler, derin yapılardan oluşan nöral mimarilerdir. İlk başlarda, bilgisayarlar göre uygulamalarını gerçekleřtirmek için tasarlanmışlardır. CNN'ler dil işleme, nesneleri tanıma ve konuşma tanıma gibi birçok sınıflandırma görevi için gelişmiş çalışmalar olarak kabul edilmektedirler [92]. Model, verilerin farklı bölgelerinde tekrarlanan desenleri yakalamayı amaçlayan kayan bir filtre olan evriřimli işlemi kullanarak ham verilerden anlamlı özellikler çıkarır. Bu sayede, CNN'ler tek boyutlu verilere sahip problemler (zaman serileri gibi) için uygun bir çözüm teknięi haline gelmektedir. Zaman serisi verileri, evriřimli fonksiyon ile özniteliklerinin çıkarım işleminin gerçekleştirilebileceęi tek boyutlu bir görüntü olarak düşünülebilmektedir.

CNN'leri modelleme aracı olarak kullanmanın bazı avantajları olduęu kadar dezavantajları da vardır [83]. Avantajları:

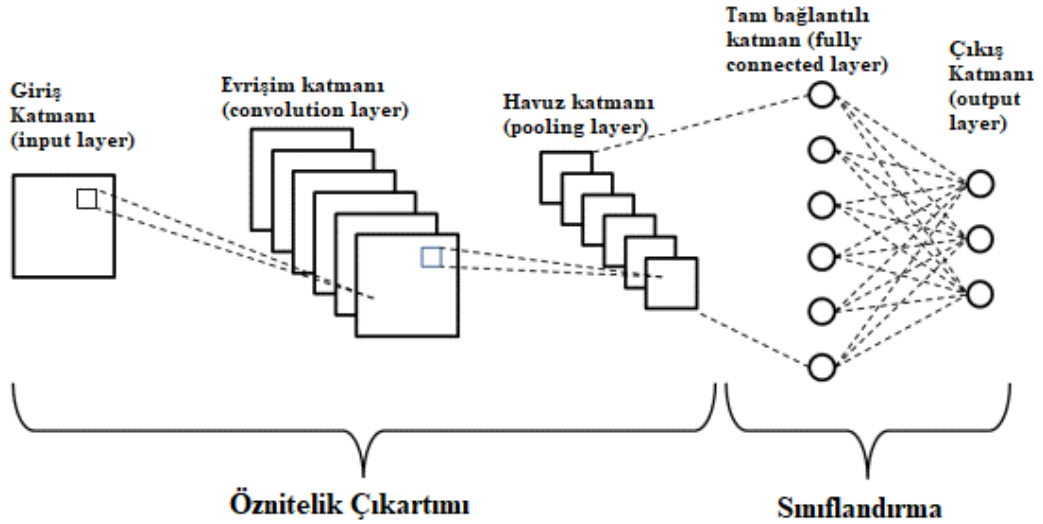
- CNN'ler tahmin veya tanımlama gibi görevleri verimli bir şekilde gerçekleřtirirler.
- GPU düşük veya eksik olduęunda aęrılık paylaşımı (weight sharing) özellięi ile hesaplama maliyetini en aza indirir.
- Azaltılmış parametre sayısı ile bellek tasarrufuna yardımcı olur.
- Eřdeęerlik, CNN'lerin özellięidir ve çıktıdaki herhangi bir büyük deęişiklięin belirlenmesine ve modelin güvenilirlięinin korunmasına yardımcı olur.
- CNN'ler ölçekleme, döndürme gibi geometrik dönüşümlerden baęımsızdır.

CNN'leri modelleme aracı olarak kullanmanın dezavantajları:

- Bir CNN'i işlemek ve eğitmek için büyük bir veri kümesi gerekir.
- CNN, maxpool gibi bir işlemler nedeniyle önemli ölçüde yavaştır. Ayrıca, ağ derinleştikçe, her eğitim adımı çok daha uzun sürmektedir.
- CNN'nin birkaç katmanı varsa ve bilgisayar iyi bir GPU'dan oluşmuyorsa eğitim süreci zaman alır.

3.3.1. CNN Mimarisi

Bir CNN mimarisi örnek olarak Şekil 3.8.'de gösterilmiştir. CNN'ler genellikle giriş katmanından (input layer), evrişim katmanlarından (convolution layers), grup katmanlarından (pooling layers), tam bağlantılı katmanlardan (fully connected layers) ve çıkış katmanından (output layer) oluşur. CNN'lerde, her düğüm yalnızca giriş alanı (receptive field) olarak adlandırılan giriş bölgesine bağlanmaktadır. Bu özellik CNN'leri standart MLP ağından ayırmaktadır. Ayrıca, aynı katmanlardaki nöronlar, aynı ağırlık matrisine sahip olmaktadır. Bu özellikler ise, CNN'lerin RNN'lere kıyasla çok daha az sayıda eğitilebilir parametreye sahip olmasına olanak sağlar ve bu sayede daha verimli öğrenme sürelerine sahip olurlar [93]. Ayrıca CNN'ler, farklı evrişimli katmanları istifleme imkânı ile derin öğrenmeyi sağlarlar.



Şekil 3.8. Evrişimli sinir ağının (CNN) yapısı [93]

Evriřim katmanı, CNN'lerin nasıl alıřtıęı konusunda hayati bir rol oynar. Katman parametreleri, renilebilir ekirdeklerin (kernels) kullanımına odaklanır. Evriřimsel katmanlar, derinlik (depth), adım (stride) ve sıfır dolgu (zero-padding) ayarı olmak zere  hiperparametre aracılıęıyla optimizasyon iřlemi gerekleřtirerek modelin karmařıklıęını nemli lde azaltır. Grup katmanları, temsilin boyutunu kademeli olarak azaltarak parametre sayısını ve modelin hesaplama karmařıklıęını dřrmeyi amalar. Tam baęlantılı katman ise iki bitiřik katmandaki nronlara doęrudan baęlı nronları ierir ve sınıflandırma grevi iin kullanılır [96].

3.4. Zamansal Evriřimli Aę (Temporal Convolutional Network)

Son alıřmalar, herhangi bir uzunluktaki diziyi iřleyebilen, sz konusu diziyi aynı uzunluktaki bir ıktıya eřleyebilen ve zamansal evriřimli aę (TCN) olarak bilinen daha zel bir CNN mimarisi nermiřtir [97]. TCN'ler, zaman serilerini iřlemek iin uygun hale getiren belirli bir tasarıma sahip bir tr evriřimli sinir aęıdır. TCN'ler iki ana ilkeyi karřılar: aęın ıkıřı, giriř dizisiyle aynı uzunluęa sahiptir ve nedensel kıvrımlar kullanarak gelecekte gemiře bilgi sızmasını nlerler. TCN mimarisi, geniřletilmiř olarak tanımlanan nedensel evriřimlerden faydalanmaktadır. Bu sayede aę, zaman serilerinde mevcut olan uzun vadeli baęımlılıklarını ğrenebilmektedir. Bu evriřim, grupta iřlemine (pooling operation) gerek olmaması sebebiyle aęın alıcı alanını znrlk kaybı yařatmadan arttırır [98].

TCN'leri modelleme aracı olarak kullanmanın bazı avantajları olduęu kadar dezavantajları da vardır. Avantajları:

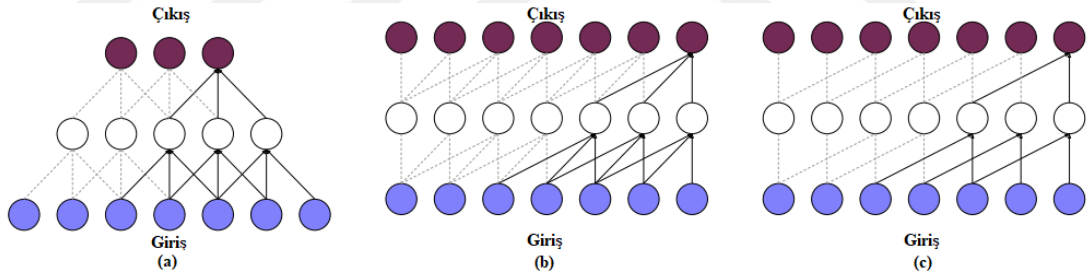
- Byk lekli paralel hesaplama ile patlayan gradyan (exploding gradient) problemini azaltırlar ve model yapısında daha fazla esneklik saęlarlar [99].
- Paylařılan evriřimli filtreler sayesinde eęitim iin dřk kısa sreli bellek gereksinimine ihtiya duyarlar.
- Uzun giriř dizilerine sahip zamansal veriler iin uygundur.
- eřitli grev ve veri kmelerinde LSTM'ler gibi tekrarlayan sinir aęlarından daha iyi performans gsterirler.

TCN'leri modelleme aracı olarak kullanmanın dezavantajları:

- Eğitim için etkin geçmiş uzunluğuna kadar ham diziyi almaları gerekmektedir, bu nedenle değerlendirme sırasında daha fazla depolama alanına ihtiyaç duyarlar.
- Farklı uygulama alanlarında (domains), modelin tahmin etmek için ihtiyaç duyduğu geçmiş veri miktarı da farklı olur. Çok az belleğe ihtiyaç duyulan bir uygulama alanından (küçük dilation factor ve filter size) çok daha fazla belleğin gerekli olduğu bir alana (küçük dilation factor ve filter size) aktarılırken, yeterince geniş bir alıcı alana sahip olmadığı için kötü performans gösterebilirler.

3.4.1. TCN Mimarisi

Nedensel evrişim, t zamanında çıktı elde etmek için gerçekleştirilen evrişim işleminin girdi olarak gelecekteki değerleri almaması gerçeğiyle standart evrişimden farklıdır. Nedensel evrişime örnek olarak, çekirdek boyutunun (kernel size) k olduğu durumda, O_t çıkışını elde etmek için $X_{t-(k-1)}, X_{t-(k-2)}, \dots, X_{t-1}, X_t$ değerlerinin kullanıldığı Şekil 3.9.'da görülmektedir. Ayrıca, giriş dizisiyle aynı uzunluğu korumak için her katmanda $k - 1$ uzunluğunda sıfır dolgu (zero-padding) kullanılır [45].



Şekil 3.9. Zamansal evrişimli ağ (TCN) mimarisi [45]

Şekil 3.9'da gösterilen TCN mimarisinde, ilk kısım (a ile gösterilen) çekirdek boyutu 3 olan iki katmanlı standart evrişimli blok, ikinci kısım (b ile gösterilen) çekirdek boyutu 3 olan iki katmanlı nedensel evrişimli bloğu temsil etmektedir. Üçüncü kısım (c ile gösterilen) ise çekirdek boyutu 2, genişleme oranı 2 olan iki katmana sahip genişletilmiş nedensel evrişimli bloğu temsil etmektedir.

Daha uzun vadeli kalıpları yakalamak amacıyla, TCN'ler tek boyutlu genişletilmiş evrişim kullanır. Bu evrişim, graplama işlemleri kullanmadan ağırlı alıcı alanını artırır, dolayısıyla çözünürlük kaybı olmaz [98]. Genişletme (dilation), evrişimli işlemin girişleri arasında d değeri kadar atlanmasından oluşur. Genişletilmiş nedensel evrişim işlemi aşağıdaki denkleme göre tanımlanır [100]:

$$x_l^t = g \left(\sum_{k=0}^{K-1} w_l^k x_{(l-1)}^{(t-(kxd))} + b_l \right) \quad (3.22)$$

Denklem (3.22) incelendiğinde; x_l^t değişkeni l numaralı katmandaki (t) konumundaki nöronun çıktısını, K evrişimli çekirdeğin (convolutional kernel) genişliğini, w_l^k değişkeni (k) konumunun ağırlığını, d evrişimin genişleme faktörünü (dilation factor) ve b_l ise bias terimini temsil etmektedir.

4. MATERYAL VE YÖNTEM

Tez çalışmasının bu bölümünde, deneysel çalışma için kullanılan veri seti, incelenen mimarilerin hiperparametre konfigürasyonları, çalışılan ortamın özellikleri ve hesaplama prosedürleri ayrıntılı olarak sunulmuştur. Ayrıca kullanılan yöntemin kaynak kodları, ilgili referansta paylaşılmaktadır. Söz konusu kodlar, tez çalışmasını okuyan araştırmacılar için referans niteliğindedir [113].

4.1. Veri Setine Genel Bir Bakış

Gerçekleştirilen tez çalışmasında, 01.01.2010 tarihi saat 00:00'dan 31.12.2020 tarihi saat 23:00'a kadar saatlik olarak örneklenmiş ve 96432 satırdan oluşan 11 sütunlu bir veri seti kullanılmıştır. Veri setinde kullanılan sütunlar ve ilgili sütunlarda kullanılan değişkenlerin özellikleri Tablo 4.1'de sunulmuştur. Tarih sütunu, ölçülen değerlerin hangi tarih ve zamanda ölçüldüğünü, tüketim sütunu ilgili tarih ve zaman aralığında ölçülen Türkiye'nin saatlik elektrik enerjisi tüketim değerini (MWh), Ağırlıklandırılmış toplam hissedilen sıcaklık sütunu ilgili tarih ve zaman aralığında ölçülen Türkiye'nin ağırlıklandırılmış toplam hissedilen sıcaklık değerini göstermektedir. Ayrıca, tatil sütunu ilgili tarihin tatil olup olmadığını, nüfus sütunu ilgili tarihteki Türkiye'nin nüfus değerini ve GSYH ise ilgili tarihteki milyon dolar cinsinden Türkiye'nin gayri safi yurt içi hasılasını (GSYH) göstermektedir. Ek olarak, elektrik enerjisi tüketim verileri Türkiye Elektrik İletim A.Ş. (TEİAŞ) [30], sıcaklık verileri The Weather Channel [101], ekonomik ve nüfus verileri ise Türkiye İstatistik Kurumu (TÜİK) [8,9] vasıtasıyla elde edilmiştir.

Veri setinde kullanılan geçen gün sayısı sütunu, sekans oluşturarak modelin mevsimler arasındaki elektrik tüketimindeki sürekli değişimleri yakalaması için dâhil edilmiştir. Önceki gün ve önceki hafta değişkenlerini içeren sütunlar ise ağırlıklandırılmış toplam hissedilen sıcaklık ve elektrik enerjisi tüketim değerleri için güncel trendi yakalamak amacıyla kullanılmıştır. Örnek olarak günlerin giderek soğuması veya sıcaklaması verilebilir. Bu yüzden, belirli tarih ve saatteki söz konusu değerlerin 24 ve 168 saat öncesindeki değerleri de eğitime dâhil edilmektedir.

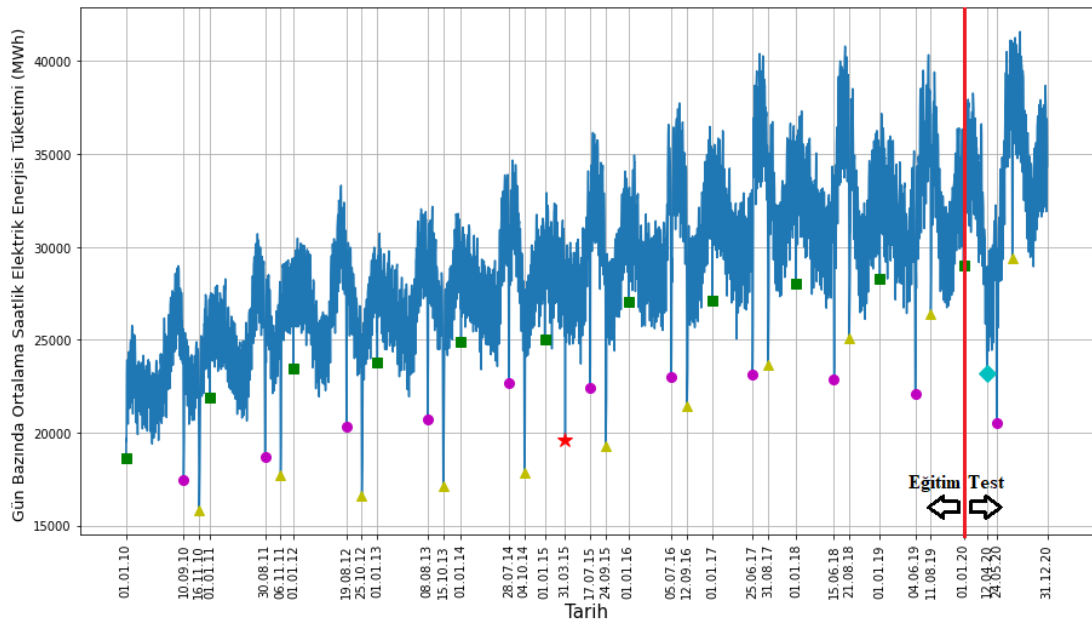
Tablo 4.1. Tez çalışmasında kullanılan veri setindeki sütunlar ve özellikleri

Sütun Nu.	Sütun İsmi	Değişken Tipi
1	Tarih	Tarihsel (Gün.Ay.Yıl Saat:Dakika)
2	Tüketim (Mwh)	Nümerik
3	Geçen Gün Sayısı	Nümerik (0 – 4017,96 arasında ardışık olarak artan değerler)
4	Onceki Gun Tüketim (Mwh)	Nümerik (1566.24 ile 49341.89 arasındaki değerler)
5	Onceki Hafta Tüketim (Mwh)	Nümerik (1566.24 ile 49341.89 arasındaki değerler)
6	Ağrlk. Toplam Hissedilen C	Nümerik (-11.83 ile 36.86 arasındaki değerler)
7	Önc. Gün Ağrlk. Top. His. C	Nümerik (-11.83 ile 36.86 arasındaki değerler)
8	Önc. Haf. Ağrlk. Top. His. C	Nümerik (-11.83 ile 36.86 arasındaki değerler)
9	Tatil	Kategorik (0 ya da 1'den oluşan değerler)
10	Türkiye Nufusu (Kisi)	Nümerik (72561312 ile başlayan ve 83614362 ile sonlanan artan değerler)
11	GSYH (Milyon Dolar)	Nümerik (152268 ile 235838 arasındaki değerler)

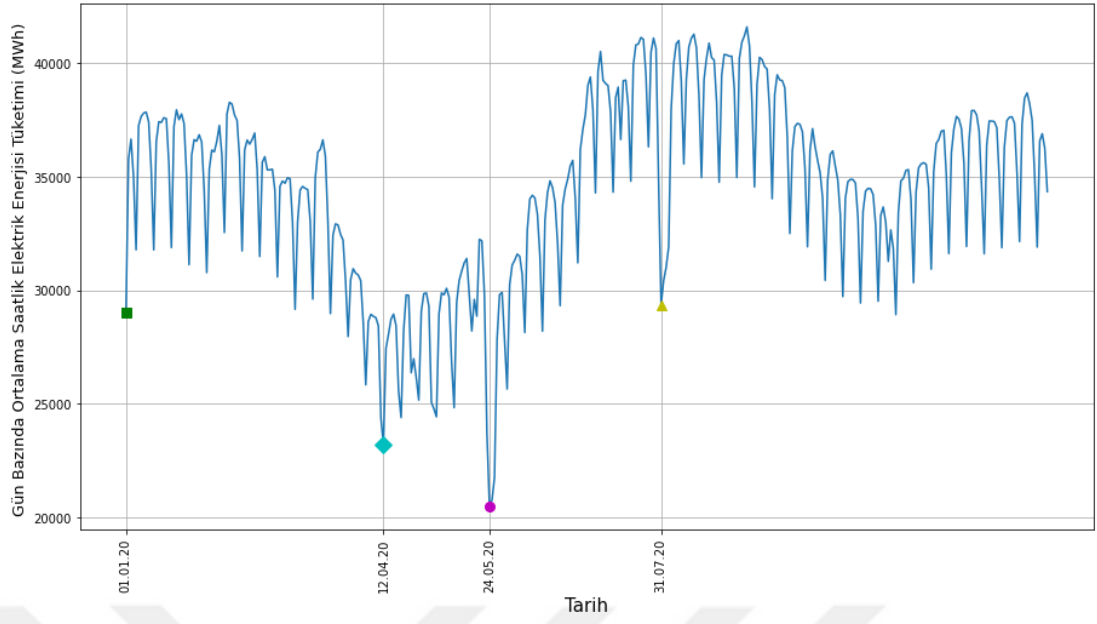
Şekil 4.1. ve Şekil 4.2’de gösterilen yeşil kareler yılbaşlarında, pembe yuvarlaklar ramazan bayramlarında ve sarı üçgenler kurban bayramlarında elektrik enerjisi tüketim değerini temsil etmektedir. Ayrıca, kırmızı yıldız 31 Mart 2015 tarihinde gerçekleşen ülke geneli elektrik kesintisinde ve mavi elmas şekli ise COVID-19 pandemisi sebebiyle sokağa çıkma yasağının olduğu gündeki elektrik enerjisi tüketim değerini temsil etmektedir. Söz konusu şekillerde gösterilen veriler incelendiğinde, yılbaşının, ramazan bayramının, kurban bayramının ve sokağa çıkma yasağının olduğu günlerde elektrik tüketiminde ani olarak düşüş gerçekleştiği görülmektedir. 31 Mart 2015 tarihinde elektrik iletim sisteminde gerçekleşen bir arıza sebebiyle arz ve talep dengesizliği yaşanmıştır ve bu sebeple Türkiye geneli elektrik

kesintisi meydana gelmiştir. Söz konusu tarihte elektrik enerjisi tüketiminde, kesinti kaynaklı ani bir düşüş gerçekleşmiştir.

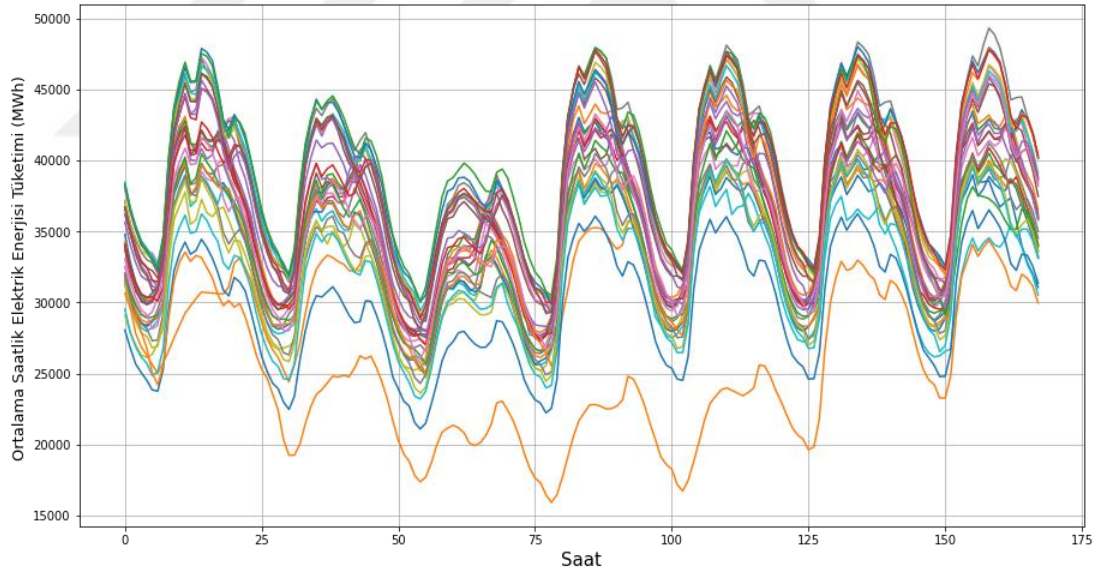
Şekil 4.2’de gösterilen grafikte Türkiye’nin yıl bazında günlük ortalama saatlik enerji tüketimi incelendiğinde, mevsimsellik kaynaklı artış ve azalışlar ve periyodiklik görülmektedir. Belirgin sıcak veya soğuk olan dönemlerde, ısıtma ve soğutma cihazları kullanımdan dolayı yüksek elektrik enerjisi tüketimi görülmektedir. Şekil 4.3’de gösterilen ve 2020 yılının 32 haftası için Türkiye’nin haftalık bazda saatlik ortalama elektrik enerjisi tüketimi incelendiğinde, özellikle pazar günlerinde haftanın en düşük elektrik enerjisi tüketim değerinin olduğu ve haftalar arasında benzer karakteristik özellikler olduğu tespit edilmiştir. Düşük tüketim değerleri, sanayi sektöründeki çoğu firma için pazar günü mesai olmadığından kaynaklanmaktadır. Şekil 4.4’de gösterilen 2020 yılının 32 günü için Türkiye’nin günlük bazda saatlik ortalama elektrik enerjisi tüketimi incelendiğinde, günlük elektrik tüketim karakteristiğinin Türkiye için benzer olduğu görülmektedir. Ayrıca, saat 08:00 ile 17:00 arasında mesai kaynaklı yüksek elektrik enerjisi tüketimi, saat 11:00’da en yüksek tüketim değeri, saat 04:00’da en düşük tüketim değeri görülmektedir. Sonuç olarak, elektrik enerjisi tüketimi üzerinde farklı zaman aralıklarında farklı dinamikler mevcut olduğu değerlendirilmektedir.



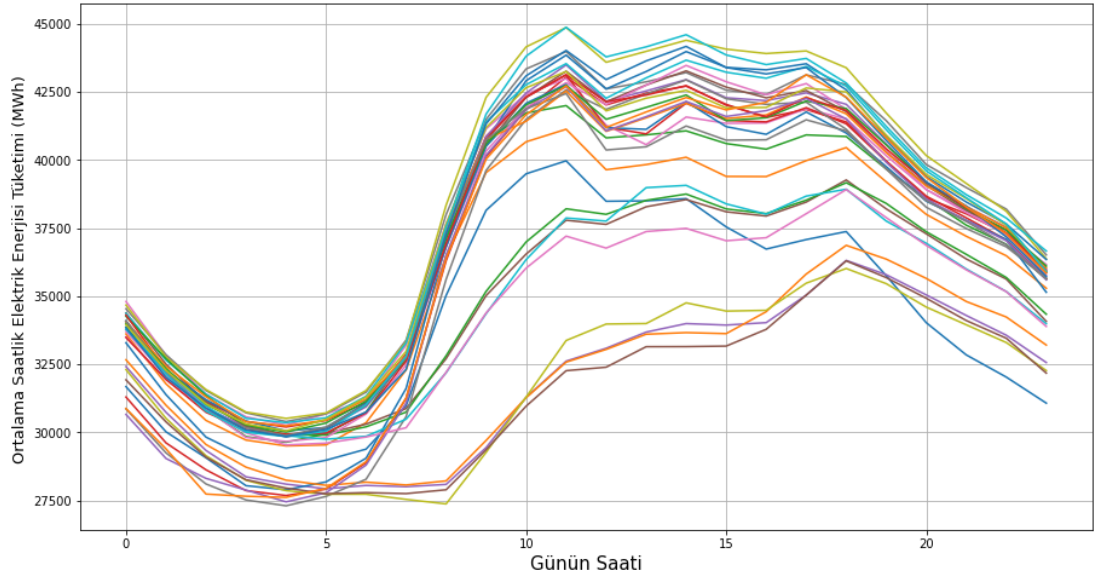
Şekil 4.1. Türkiye’nin gün bazında ortalama saatlik elektrik enerjisi tüketimi (MWh)



Şekil 4.2. Türkiye'nin gün bazında ortalama saatlik elektrik enerjisi tüketimi (MWh)



Şekil 4.3. 32 hafta için Türkiye'nin ortalama saatlik elektrik enerjisi tüketimi (MWh)



Şekil 4.4. 32 gün için Türkiye'nin ortalama saatlik elektrik enerjisi tüketimi (MWh)

Şekil 4.3'te 2020 yılı içerisinde rastgele olarak seçilmiş 32 haftada ve Şekil 4.4'te ise 2020 yılı içerisinde rastgele olarak seçilmiş 32 günde, Türkiye'nin ortalama saatlik elektrik enerjisi tüketim değerleri verilmiştir. Söz konusu şekillerin sunulma amacı, elektrik enerjisi tüketim karakteristiğinin farklı tarihlerde bile benzerlik gösterdiğini belirtmektir.

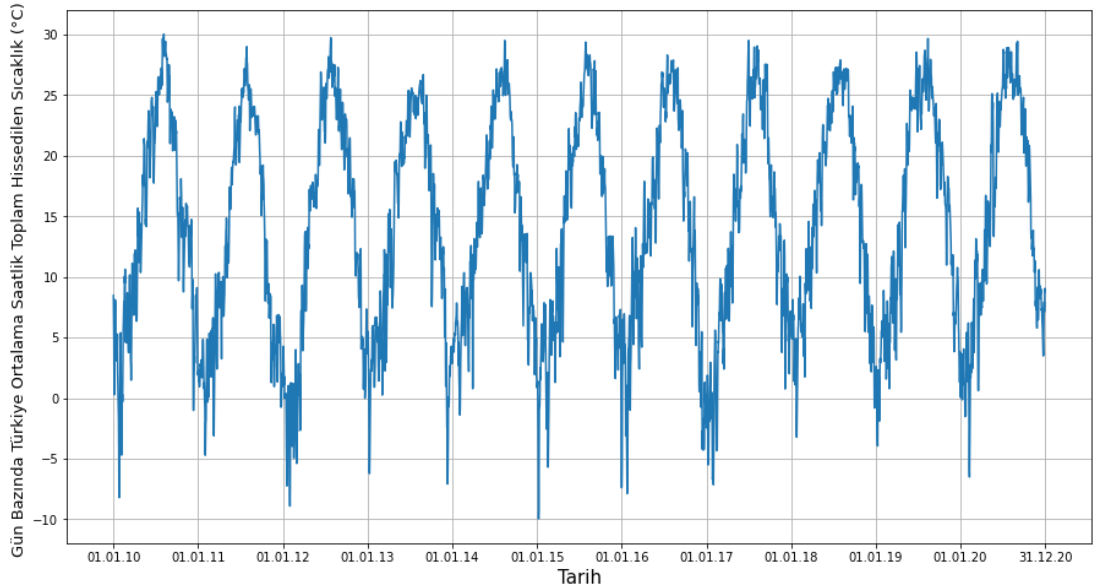
Literatür incelendiğinde, kısa vadeli elektrik enerjisi tüketimi üzerindeki en önemli faktörlerden birinin sıcaklık olduğu tespit edilmiştir. Bu tez çalışmasında, şehirlerin nüfus değerlerine göre oranlanmış, Türkiye'nin tüm şehirlerini içine alan ve 01.01.2010 tarihi saat 00:00'dan 31.12.2020 tarihi saat 23:00'a kadar olan dönemi kapsayan saatlik olarak örneklenmiş ağırlıklandırılmış toplam hissedilen sıcaklık değişkeni önerilmiştir. İlk önce Türkiye'nin 81 ili için ayrı ayrı ve saatlik örneklenmiş olarak hissedilen sıcaklık değerleri kaydedilmiştir. Daha sonra Denklem (4.1)'de gösterildiği üzere her şehrin kendi nüfusunu Türkiye'nin toplam nüfusuna bölerek ağırlık değerleri elde edilmiştir. Söz konusu ağırlık değerleri her şehir için saatlik olarak örneklenmiş hissedilen sıcaklık değeri ile çarpılıp kaydedilmiştir. Daha sonra Denklem (4.2)'de gösterildiği üzere bu değerler toplanarak saatlik olarak örneklenmiş Türkiye geneli ağırlıklandırılmış toplam hissedilen sıcaklık verisi elde edilmiştir. Ağırlıklandırılmış toplam hissedilen sıcaklık değerinin formülasyonu aşağıda gösterilmektedir:

$$Ağırlık_{şehir} = \frac{Nüfus_{şehir}}{\sum_{S \in Şehirler} Nüfus_S} \quad (4.1)$$

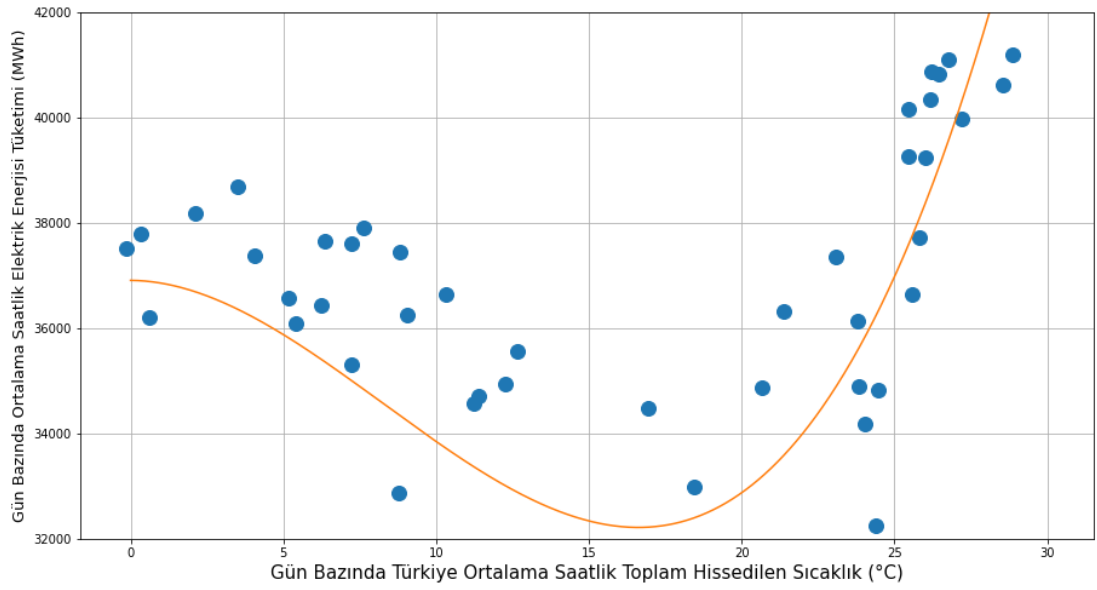
Ağırlıklandırılmış Toplam Hissedilen Sıcaklık

$$= \sum_{S \in Şehirler} Hissedilen Sıcaklık_S * Ağırlık_S \quad (4.2)$$

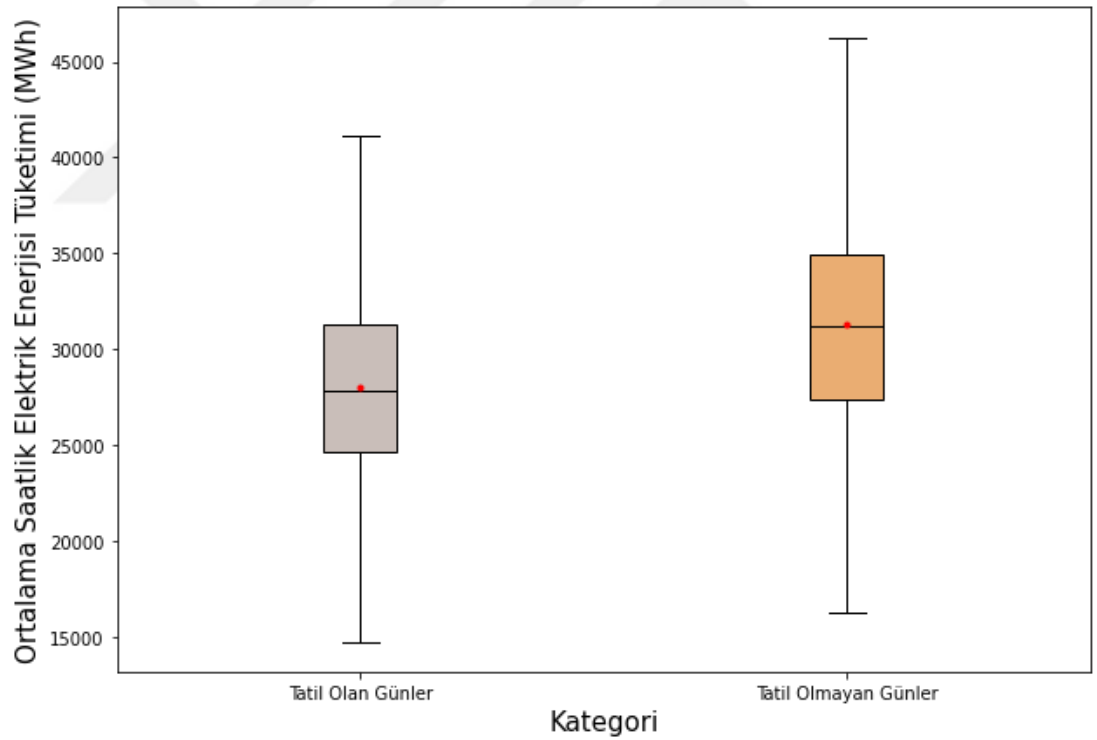
Hissedilen sıcaklık, termometrenin ölçtüğü değerden farklı olarak insan vücudunun hissettiği sıcaklıktır. Denklem (4.1) ve Denklem (4.2)'den elde edilen değerler ile ağırlıklandırılmış toplam hissedilen sıcaklık veri seti oluşturulmuştur. Veri setini elde ederken Türkiye'nin her şehri için ihtiyaç duyulan hissedilen sıcaklık değeri veri dağıtıcısından elde edilmiştir. Söz konusu veri dağıtıcısı hissedilen sıcaklık değerini, ilgili şehrin sıcaklık, nem ve rüzgâr ölçümlerinden elde ederek halka sunmaktadır. Oluşturulan veri seti Şekil 4.5'te gösterilmektedir. Şekil incelendiğinde ağırlıklandırılmış toplam hissedilen sıcaklık değerinde mevsim kaynaklı periyodiklik olduğu görülmektedir. Şekil 4.6'da ise elektrik enerjisi tüketiminin mevsimsel dinamiklerden kaynaklı sıcaklıklardan etkilendiğine dair ölçümler gösterilmektedir. Ayrıca, 32000 MWh değerinden yüksek ölçümler grafiğe dâhil edilmiştir.



Şekil 4.5. Türkiye'nin gün bazında ortalama hissedilen sıcaklık değeri (°C)



Şekil 4.6. Hissedilen sıcaklığa (°C) göre elektrik enerjisi tüketimi (MWh)



Şekil 4.7. Takvimsel özelliklere göre ortalama saatlik elektrik tüketimi (MWh)

Literatür incelendiğinde, STLF problemi için takvimsel özelliklerin büyük bir etkisinin olduğu tespit edilmiştir. Bu yüzden, öznitelik sütunu olarak ilgili tarihin tatil günü olup olmadığını işaret eden tatil sütunu eklenmiştir. Tatil sütunundaki değer 0

ise o gün herhangi bir tatil söz konusu değilken, tatil sütunundaki değer 1 ise o gün tatil olduğu (yılbaşı, ramazan bayramı, kurban bayramı, hafta sonu, resmi tatiller ve özel günler) işaret edilmektedir. Şekil 4.7’de 2020 yılı için tatil olan ve olmayan günlerdeki saatlik ortalama elektrik enerjisi tüketimini sınıflandıran bir grafik gösterilmiştir. Söz konusu kutu grafiği incelendiğinde, tatil olmayan günlerdeki saatlik ortalama elektrik enerjisi tüketim değerlerinin tatil olan günlere kıyasla daha yüksek olduğu görülmektedir. Ayrıca en ilgili tarihteki en yüksek tüketim değeri tatil olmayan günde gerçekleşirken, en düşük tüketim değeri ise tatil olan günde gerçekleştiği grafik üzerinde görülmektedir.

Nüfus (popülasyon) ve gayrisafi yurtiçi hasıla (GSYİH) gibi ekonomik verilerin, uzun vadeli dönem için elektrik enerjisi tüketimi üzerinde etkisinin olduğunu gösteren çalışmalar literatürde mevcuttur. Gerçekleştirilen bu tez çalışmasında, nüfus ve GSYİH değerlerinin STFL üzerindeki etkisi de araştırılmıştır. Bu yüzden veri seti içine bu bilgileri içeren 2 sütun eklenmiştir. Ülkemizde, nüfus ve GSYİH verileri belirli aralıklarla açıklanmaktadır. Örnek olarak, GSYİH değerleri yılın her çeyreğinde TÜİK tarafından açıklanmaktadır. Söz konusu 2 değişkenin frekansı ile tezde kullanılan veri setinin frekansı farklı olması sebebiyle interpolasyon işlemine ihtiyaç duyulmuştur. Veri setindeki boş satırları doldurmak için söz konusu 2 değişkenin mevcut sayısal değerleri kullanılarak, boş noktalardaki değerler tahmin edilmiştir. Bu sayede, nüfus ve GSYİH bilgileri saatlik olarak örneklenmiştir.

4.2. Derin Öğrenme Mimarilerinin Deneysel Kurulumu

Bu bölümde, zaman serisi tahmini için tezde kullanılmış olan 8 derin öğrenme mimarisi için hiperparametre konfigürasyonlarının yanı sıra değerlendirme prosedürü de anlatılmıştır. Tez çalışması içinde gerçekleştirilen deneysel çalışmada, farklı kombinasyona sahip veri kümeleri üzerinde birçok farklı model mimari konfigürasyonu uygulanmıştır. Bu uygulamalar ile elde edilen deney sonuçlarının istatistik tabanlı bir analizi gerçekleştirilmiştir. Bu deneysel çalışmada, MLP, ERNN, LSTM, Encoder-Decoder SLSTM, GRU, ESN, CNN ve TCN gibi 8 farklı derin öğrenme modeli karşılaştırılmıştır. Tüm bu modeller için yapılandırılması gereken hiperparametrelerin sayısı, geleneksel makine öğrenimi tekniklerine kıyasla daha yüksektir. Bu sebeple, söz konusu parametrelerin uygun bir şekilde ayarlanması, önemli ölçüde hem uzmanlık gerektiren hem de çoğu zaman sezgiye dayalı karmaşık

bir görevdir. Bu çalışmada, en uygun hiperparametreleri bulmak amacıyla her bir mimari tipinin konfigürasyonu için kapsamlı bir grid araması (grid search) gerçekleştirilmiştir.

Tablo 4.2. Derin öğrenme mimarileri için kullanılan parametreler

Modeller	Parametreler	Değerler
MLP	Hidden Layers	[8], [8, 16], [16, 8], [8, 16, 32], [32, 16, 8], [8, 16, 32, 16, 8], [32], [32, 64], [64, 32], [32, 64, 128], [128, 64, 32], [32, 64, 128, 64, 32]
ERNN	Layers	1, 2, 4
	Units	32, 64, 128
	Return sequence	True, False
LSTM	Layers	1, 2, 4
	Units	32, 64, 128
	Return sequence	True, False
SLSTM	Layers	[1, 2]
	Return sequence	True, False
GRU	Layers	1, 2, 4
	Units	32, 64, 128
	Return sequence	True, False
ESN	Layers	1, 2, 4
	Units	32, 64, 128
	Return sequence	True, False
CNN	Layers	1, 2, 4
	Filters	16, 32, 64
	Pool size	0, 2
TCN	Layers	1, 3
	Filters	32, 64
	Dilations	[1, 2, 4, 8], [1, 2, 4, 8, 16]
	Kernel size	3, 6
	Return sequence	True, False

Katman (layer), birim sayısı (units) gibi derin öğrenme modellerinin tezde kullanılan ana parametreleri Tablo 4.2’de gösterilmektedir. Nöronların, birimlerin veya filtrelerin sayısı için 2 ve 2’nin kuvvetleri seçilmiştir. Bu değerler literatürde kullanılan genel değerlerdir. Sonuç olarak, olasılıklar tablosu oluşturulurken daha önce

literatürde gerçekleştirilmiş çalışmalarda yaygın olarak ele alınan değerler kullanılmıştır. İlk olarak her katmandaki birim sayısı farklılık gösteren 12 farklı MLP modelinin uygulaması gerçekleştirilmiştir. Her katmandaki nöronların sayısı 8 ila 128 arasında değişmektedir ve bunlar, uzunluğu ve zaman ufku değişen dizilere uygulanmıştır. İlk katmanlarda daha fazla ve sonunda daha az nöron veya tam tersi şeklinde uygulanarak, hem kodlayıcı hem de kod çözücü benzeri yapılara benzetilmesi amaçlanmıştır.

Tekrarlayan sinir ağları (RNN) ile ilgili olarak, bu çalışmada ERNN, LSTM, Encoder-Decoder SLSTM, GRU, ve ESN olmak üzere 5 farklı mimari kullanılmıştır. Her mimari için 18 farklı modelle sonuçlanan ve 3 ana parametre üzerinde bir grid araması gerçekleştirilmiştir. Birinci ana parametre, yığılmış (stacked) tekrarlayan katmanların sayısıdır (layers). İkinci ana parametre ise her katmandaki tekrarlayan birimlerin sayısıdır (units). Grid araması gerçekleştirilen üçüncü parametre ise son katmanın durum dizisini (state sequence) mi yoksa yalnızca son durumu (final state) mu döndürdüğüyle ilgili olan parametredir. Dönüş dizisi (return sequence), false (varsayılan) ise ağ yalnızca son zaman adımıdaki gizli durumu çıktı olarak vermektedir. Aksi takdirde, ağ, gizli durumların tam sırasını çıktı olarak vermektedir. Söz konusu parametrenin, ağ katmanının iç denklemlerinin değişmesine sebep olmamaktadır. Birim sayısının 32 ile 128 arasında olması, paralel olarak daha fazla veya daha az öğrenme hücresine sahip olmanın model performansı üzerindeki etkisini keşfetmeyi amaçlamaktadır.

CNN ile ilgili olarak, bu çalışmada 18 farklı modelle sonuçlanan ve 3 ana parametre üzerinde bir grid araması gerçekleştirilmiştir. Tek katmanlı modellerin çekirdek boyutu 3, iki katmanlı modellerin çekirdek boyutları 5 ve 3, dört katmanlı modellerde çekirdek boyutu ise 7-5-3 ve 3'tür. TCN ile ilgili olarak ise bu çalışmada 32 farklı modelle sonuçlanan ve 5 ana parametre üzerinde bir grid araması gerçekleştirilmiştir. Grid araması, katmanlarının sayısı (layers), evrişimli filtrelerin sayısı (filters), genişleme faktörleri (dilations), evrişimli çekirdek boyutu (kernel size) ve son çıktının mı yoksa tam dizinin mi döndürüleceğini belirleyen dönüş dizisi (return sequence) parametreleri üzerinden gerçekleştirilmiştir.

4.3. Değerlendirme Prosedürü

Tez çalışmasının bu bölümü, elde edilen veri seti ile kullanılan modellerin eğitim sürecini kapsamaktadır. Söz konusunu çalışma, Jupyter Notebook App tarafından sağlanan ve “Python” programlama dili ile çalışmaya imkân tanıyan not defterleri (notebooks) üzerinde gerçekleştirilmiştir. Jupyter Notebook uygulaması, Chrome gibi bir web tarayıcısı (web browser) aracılığıyla istenilen belgelerin düzenlenmesine ve çalıştırılmasına izin veren bir uygulamadır. İstenildiği takdirde söz konusu uygulama ile, internet erişimine ihtiyaç duymaya yerel (local) bir masaüstünde çalıştırılabilmekte veya uzak bir sunucuya kurularak internet aracılığıyla belgelerin düzenlenmesine olanak sağlamaktadır.

Jupyter notebook için erişim Google firmasının sağladığı “Colab” uygulaması ile gerçekleştirilmiştir. Colab, tamamen bulutta çalışan ücretsiz bir Jupyter notebook ortamıdır. Birçok popüler makine öğrenimi kütüphanesini desteklemesi, bir kurulum gerektirmemesi, kolayca “Python” programa dilinde kod yazılıp yürütülmeye imkân tanınması, matematiksel denklemleri desteklemesi, Google Drive ile koordineli çalışması ve pratik olarak dosya paylaşmaya olanak sağlaması sebebiyle tercih edilmiştir. Tez çalışmasında gerçekleştirilen modelin eğitimleri uzun bir süre boyunca devam etmesi, Google Drive tarafından problem oluşturmaya sebep olmuştur. Bu nedenle, depolama ve dosya aktarım aracı olarak Google Drive kullanılması yerine, tez çalışmasına özel dosya transfer protokolü (ftp server) kullanılmıştır. FTP Sunucusu en basit tanımıyla dosyaların bir bilgisayardan diğerine aktarılmasını sağlayan bir yazılım uygulamasıdır. Dünyadaki internete bağlı herhangi bir bilgisayara dosya aktarmanın bir yoludur. Söz konusu protokolün kullanılması sayesinde, model eğitimleri kesintisiz ve sürekli olarak devam etmiştir.

Tez çalışmasında derin sinir ağları ile hızlı deneyler gerçekleştirmek amacıyla Google tarafından geliştirilen “TensorFlow” adında açık kaynaklı sinir ağı kütüphanesi kullanılmıştır. Ayrıca Tensorflow mimarisi, veri ön işleme, model oluşturma ve modeli eğitme amacıyla kullanılmaktadır. Bu kütüphane, açık kaynaklı olması, veriyi görselleştirebilmesi, kullanıcı dostu olması, ölçeklenebilir olması sebebiyle tercih edilmiştir. Tensorflow kütüphanesinin avantajları olduğu kadar dezavantajları da vardır. Bunlar; kullanıcı için ek yük oluşturan sık güncellemeler, bir

Colab ekranında aynı anda sadece tek bir modelin eğitime izin vermesi, rakiplerine göre daha düşük hıza sahip olmasıdır.

4.3.1. Veri Setinin Bölünmesi

Veri setinin 1 Ocak 2010 tarihinden 31 Aralık 2019 tarihine kadar olan kısmı eğitim için kullanılırken, 1 Ocak 2020 tarihinden 31 Aralık 2020 tarihine kadar olan kısmı ise test işlemi için kullanılmıştır. Tez çalışmasında, Şekil 4.8.'de gösterildiği üzere eğitim, test ve doğrulama setleri kullanılmıştır. Eğitim seti (training data set), yeni ve modelin daha önce hiç görmediği veriler için iyi bir genelleme veya tahmin yapabilen bir model eğitebilmek amacıyla kullanılan veri setidir. Söz konusu veri seti, öğrenme süreci sırasında kullanılmaktadır. Başka bir deyişle, eğitim seti, ilgili modelin parametrelerini en istenilen çıktıyı vermesi için ayarlamaya yaramaktadır.



Şekil 4.8. Tez çalışmasında kullanılan eğitim, test ve doğrulama setlerinin ayrılması

İlgili modelin eğitimi sırasında performansını değerlendirmek amacıyla eğitim setinin içinden doğrulama seti (validation data set) kullanılmıştır. Doğrulama veri seti, test için kullanılan eğitim verileridir, fakat son testin bir parçası olarak kullanılmamıştır. Test veri seti (test data set) ise eğitim veri setinden bağımsız olan ancak eğitim veri seti ile aynı olasılık dağılımını izleyen bir veri setidir. Diğer bir deyişle test seti, tahmin edilen çıktıların doğruluğunu test etmek için kullanılmaktadır. Test setinde kullanılan verilerin model tarafından öğrenilmesinin önüne geçmek amacıyla, tez çalışmasında kullanılan modellerin eğitimi gerçekleştirilirken doğrulama ve test veri kümelerindeki örnekler kullanılmamıştır.

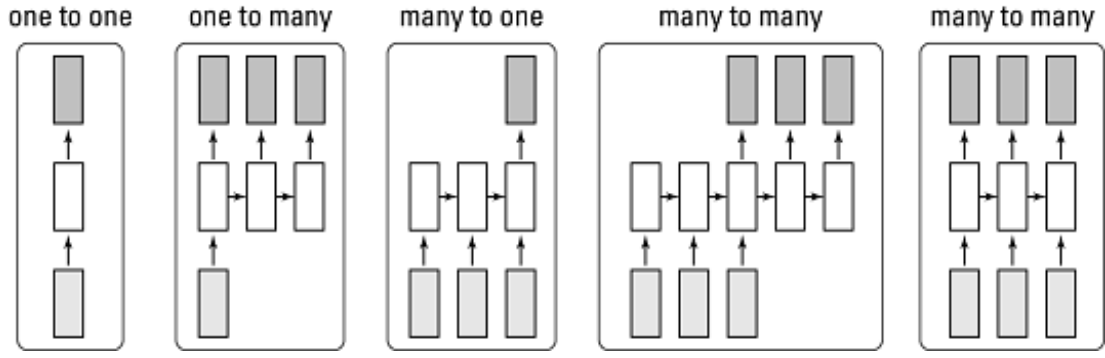
4.3.2. Ön İşleme (Pre-Processing) ve Eğitimin (Training) Gerçekleştirilmesi

Tez çalışmasındaki zaman serisini eğitim ve test olarak ayırdıktan sonra, bir dizi ön işleme adımı gerçekleştirilmiştir. İlk olarak, veri seti 0 ile 1 arasında ölçeklenecek şekilde normalleştirme (normalization) işlemi uygulanmıştır. Normalleştirme, modelin eğitim süresini büyük ölçüde azaltabileceğinden derin öğrenmede her zaman aktif bir araştırma alanı olmuştur. Tez çalışması için kullanılan veri setinde, bazı öznitelikler diğerlerine göre daha yüksek sayısal değerlere sahiptir. Bu nedenle, her öznenin eğitime olan katkısı konusunda ağın tarafsız olmasını sağlamak amacıyla normalleştirme işlemi gerçekleştirilmiştir. Eğitim sırasında ağ parametreleri değişime uğramaktadır. Bu nedenle ağ aktivasyonlarının dağılımındaki farklılığı azaltmak için normalleştirme adımı uygulanmıştır. Ayrıca normalleştirme gradyanların büyüklüklerini sınırlamakta, ağırlıkların her yerde patlamasına izin vermediği ve onları belirli bir aralıkla sınırladığı için optimizasyonu daha hızlı hale getirmektedir. Literatürde en yaygın olarak kullanılan 2 normalizasyon fonksiyonu aşağıda gösterilmektedir [107]:

$$\min - \max(x) = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4.3)$$

$$z - score(x) = \frac{x - ortalama(x)}{\max(x) - \min(x)} \quad (4.4)$$

Denklem (4.3)'de "min-max" normalizasyonu, Denklem (4.4)'te ise "z-score" normalizasyonu gösterilmektedir. Min-max normalizasyonu ile değişkenler 0 ile 1 arasında ölçeklenmektedir. Söz konusu normalizasyon yöntemi verideki maksimum ve minimum yani aykırı (outliers) değerlerden büyük ölçüde etkilenmektedir. Yöntemin bu özelliği bias oluşmasına sebep olmaktadır. Aykırı değerler ile ya el ile kaldırarak ya da verileri nicelik aralığına göre ölçekleyen "RobustScaler" kullanarak başa çıkılabilir. Z-score normalizasyonu ise bir veri kümesindeki her değeri, tüm değerlerin ortalaması 0 ve standart sapması 1 olacak şekilde normalleştirme sürecini ifade etmektedir. Bu yöntem sayesinde, aykırı değerlerin model uyumu üzerinde etkisi azaltılmış olmaktadır. Anlatılan normalleştirme yöntemleri tez çalışmasında da kullanılmıştır.



Şekil 4.9. RNN’ler için farklı giriş-çıkış konfigürasyonları

Çıktının nasıl kullanılacağına karar vermek, RNN’ler tarafından desteklenen bir derin öğrenme mimarisinin neler başarabileceğini belirlemektedir. Bu amaç için kullanılabilecek teknikler Şekil 4.9’da gösterilmektedir. “One to one” yapısı, bir girdi ve sonuç olarak bir çıktı olduğunda, “one to many” yapısı bir girdi ve sonuç olarak bir dizi çıktı olduğunda (otomatik altyazı oluşturma gibi), “many to one” yapısı metin gibi bir dizi girdi ve tek bir sonuç olarak çıktı olduğunda kullanılmaktadır. “many to many” yapısı ise girdi olarak bir dizi ve çıktı olarak bir sonuç dizisi sağlamaktadır [110]. Bu, en etkileyici derin öğrenme destekli yapay zekâ uygulamalarının çoğu için temel mimaridir. Bu yaklaşım, makine çevirisi, sohbet robotları veya görüntü sınıflandırıcı gibi uygulamalarda kullanılmaktadır. Tez çalışmasında, çoklu giriş çoklu çıkış (mimo) strateji seçilmiştir. Bunun sebebi, tahminler üzerinde hatayı biriktirmediğinden diğer yaklaşımların aksine daha iyi performans göstermesidir. Ayrıca MIMO yaklaşımı, her tahmin için tek bir model kullandığından hesaplama süresi açısından doğrudan stratejiye göre daha verimlidir [102].

Ön işleme adımları gerçekleştirildikten sonra, modeller eğitime başlanmıştır. Ayrıca, modellerin eğitim işlemleri gerçekleştirilirken optimal eğitim konfigürasyonu üzerinde bir grid araması da uygulanmıştır. Grid aramasında kullanılan parametre değerleri Tablo 4.3’te verilmiştir. Parti boyutu (batch size), ağırlık değeri gibi model parametrelerini en uygun değere ayarlamadan (güncellemeden) önce üzerinde çalışılacak örnek sayısını ifade eden bir hiperparametredir. Diğer bir deyişle, ağırlık verilen alt örneklerin sayısıdır. “Mini-Batch” kullanıldığı zaman, hata değerinin arttığı fakat zamandan kazanıldığı tespit edilmiştir.

“Batch size” değerini verilebilecek en küçük değer olan 1 olarak seçildiğinde sadece doğrudan veri seti üzerinden eğitim işlemi gerçekleştirilir. Ek olarak, “Batch size” kullanımı ilgili modelin gürültüyü öğrenmesine sebep olabilmektedir. Bir Epoch, tüm veri kümesinin sinir ağı üzerinden yalnızca bir kez geçirilmesidir. Gerçekleştirilen her epoch için söz konusu epoch’ta batch sayısı, iterasyon sayısına eşit olmaktadır. Bir sinir ağı modelini eğitmek için kullanılan epoch sayısı gereğinden fazla olduğunda, eğitim modeli büyük ölçüde örnek verilere özgü kalıpları öğrenmektedir. Gereğinden düşük epoch sayısı ise ağı öğrenme performansını düşürmektedir.

Minimum kayıp değerine doğru ilerlerken, her iterasyonda modelin attığı adım boyutunu ifade eden ve optimizasyon algoritmasında bulunan ayar parametresi, öğrenme oranı (learning rate) olarak adlandırılmaktadır. Öğrenme oranı düştükçe, eğitim daha kararlı ve güvenilir hale gelmektedir. Bu koşulda kayıp fonksiyonunun minimuma doğru attığı adımları küçük olması sebebiyle optimizasyon yada eğitim süreci çok daha fazla zaman almaktadır. Öğrenme oranı yüksek olduğu durumda ise, optimize edici (optimizer) minimumu aşmakta ve söz konusu kaybı daha kötüleştirmektedir. Derin öğrenme modelini eğitirken, her epoch için ağırlıkların değiştirilmesi ve kayıp fonksiyonunun en aza indirilmesi gerekmektedir. Optimize edici olarak tanımlanan parametre ise sinir ağının ağırlıklar ve öğrenme hızı gibi konularda değerlerini değiştiren bir fonksiyon veya algoritmadır. Böylece, genel kaybın azaltılmasına ve doğruluğun artırılmasına yardımcı olmaktadır [109]. Tez çalışmasındaki deneylerin çoğunda, çeşitli makine öğrenimi için çok uygun olduğu kanıtlanmış, uyarlanabilir bir stokastik optimizasyon yöntemini uygulayan “Adam” optimize edici seçilmiştir. Ayrıca, tez çalışmasında çoğunluk “batch size” için 32 ve 64, epoch için 10, normalizasyon yöntemi için “z-score” ve öğrenme oranı için ise 0,001 ve 0,01 yaygın kullanım değerleri kullanılmıştır. Ayrıca tez çalışmasında eğitim işlemi devam ederken, doğrulama seti üzerinden kosinüs benzerliği (cosine similarity) alternatif bir performans ölçütü olarak sürekli ölçülmektedir.

Tablo 4.3. Tez çalışmasına dâhil edilen eğitim parametreleri

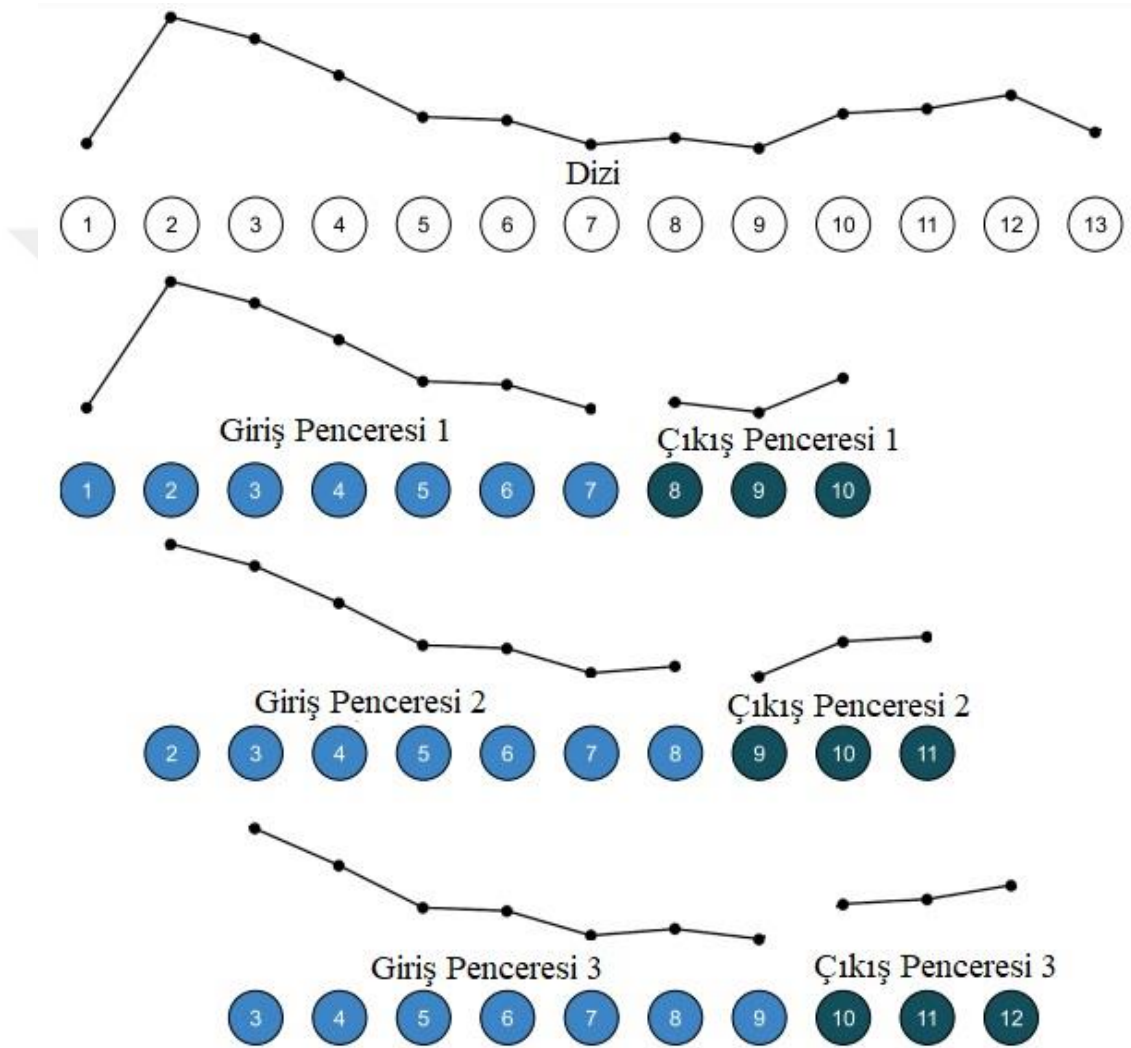
Parametreler	Değerler
Tahmin Ufku	1,6 ve 24 Saat
Geçmiş Değer	72,168 ve 336 Saat
Epoch Sayısı	1,5,10,20,30,40
Batch Size	2,4,8,16,32,64
Optimize Edici	Adam, SGD, RMSprop, Adadelta, Adagrad, Adamax
Öğrenme Oranı	0.000001, 0.00001, 0.0001, 0.001, 0.01
Normalizasyon Yöntemi	min-max ve z-score

Hem veri setinin büyük olması, hem de kapsamlı eğitim parametreleri sebebiyle, tez çalışmasında gerçekleştirilen çoğu eğitimin süresi oldukça fazladır. Harcanan eğitim zamanının boşa gitmesini önlemek amacıyla, söz konusu problem için çözüm ihtiyacı ortaya çıkmıştır. Çözüm olarak ise her epoch işleminden sonra bir kontrol noktası (check-point) oluşturulması önerilmiştir. Bu sayede her epoch işleminden sonra modelin özellikle ağırlık değerleri kaydedilerek bir sonraki epoch işlemine geçilmektedir. Eğer bir sonraki epoch işlemi için performans ölçütü, bir öncekine kıyasla daha yüksek çıkarsa, check-point için oluşturulan değişkenin içi yeni epoch işleminde bulunan ağırlık değerleri ile güncellenmektedir. Bu sayede, bütün epoch işlemlerinin sonunda, en yüksek performansın elde edildiği epoch ve ağırlık değerleri elde tutulmaktadır [108]. Ayrıca tez çalışmasında, check-point işlemini denetleyen performans ölçütü olarak doğrulama setinin ortalama mutlak yüzde hatası (mape) tercih edilmiştir. Bu sayede model performansı, doğrulama seti üzerinden objektif olarak değerlendirilmektedir.

4.3.3. Hareketli Pencere Prosedürü (Moving Window Procedure)

Tez çalışmasında kullanılan ve 01.01.2020 tarihinden sonraki zaman dilimini kapsayan test verisini, modelleri besleyebilecek eğitim örneklerine dönüştürebilmek için hareketli bir pencere şeması kullanılmıştır. Bu işlem, sabit boyutta bir pencereyi kaydırarak her konumda bir girdi-çıkı örneği oluşturur. Bu sayede, 2020 yılı içindeki herhangi bir tarih aralığı için belirlenen zaman ufku tahmin yapılabilir. Bahsedilen tekniğin temelinde, veri güncelleme sistemi bulunmaktadır. Çalışma

prensibi, hareketli pencere bir sonraki adıma geçtiğinde, yeni örnekler pencereye eklenmekte ve en eski örnekler pencereden çıkarılmaktadır. Örneklerin bu yöntem ile güncellenmesi, modele dinamik bir özellik kazandırmaktadır [103]. Girdinin uzunluğu, tahmin işlemi yapılırken ne kadar geçmiş değere bakılacağı ile ilgilidir. Şekil 4.10.'da geçmiş tarih olarak 7 gözlem ve 3 değerlik bir tahmin ufku ile bu tekniğin bir örneğini gösterilmiştir.



Şekil 4.10. Tez çalışmasında kullanılan hareketli pencere yönteminin gösterimi [51]

4.3.4. Değerlendirme Metrikleri

Tez çalışmasında önerilen modellerin performansı, doğruluk ve verimlilik açısından değerlendirilmiştir. Her mimari çeşidiyle bulunan sonuçlar analiz edilmiştir. Ayrıca, farklı hiperparametre yapılandırmalarıyla bulunan sonuçların dağılımı da değerlendirilmiştir. Ortalama mutlak yüzde hata (MAPE) ölçümü kullanılarak eğitilen modeller için tahmin performansı elde edilmiştir. Bu metrik, yaygın kullanılan tahmin doğruluğu ölçümlerinden biridir [104]. Bir veri kümesindeki her bir girişin mutlak yüzde hatalarının ortalamasını temsil eder ve ortalama olarak, tahmin edilen miktarların gerçek miktarlarla karşılaştırıldığında ne kadar doğru olduğunu gösterir. MAPE genellikle daha büyük veri kümelerini analiz etmek için kullanılmaktadır. MAPE ölçümü, yorumlanması kolay ve tahminlerin güvenilirliği açısından doğru bir değerlendirme sağlayabildiği için tez çalışmasında tercih edilmiştir. Ayrıca, ileriki çalışmalar için katkı sağlaması amacıyla her bir modelin eğitim işlemi bittikten sonra aşağıdaki denklemlerde gösterilmiş olan ölçüm metrikleri de hesaplanıp kaydedilmiştir [106].

$$e_j = A_j - P_j \quad (4.5)$$

$$|e_j| = |A_j - P_j| \quad (4.6)$$

$$e_j^2 = (A_j - P_j)^2 \quad (4.7)$$

$$MAE = \frac{1}{n} \sum_{j=1}^n |e_j| \quad (4.8)$$

$$MSE = \frac{1}{n} \sum_{j=1}^n e_j^2 \quad (4.9)$$

$$RMSE = \sqrt{\frac{\sum_{j=1}^n e_j^2}{n}} = \sqrt{MSE} \quad (4.10)$$

$$MAPE = \frac{100}{n} \sum_{j=1}^n \frac{|e_j|}{|A_j|} \quad (4.11)$$

$$WAPE = \frac{\sum_{j=1}^n |e_j|}{\sum_{j=1}^n A_j} \quad (4.12)$$

$$sMAPE = \frac{100}{n} \sum_{j=1}^n \frac{2 * |e_j|}{|A_j + P_j|} \quad (4.13)$$

$$MASE = \frac{\sum_{j=1}^n |e_j|}{\frac{n}{n-1} \sum_{j=2}^n |A_j - A_{j-1}|} \quad (4.14)$$

$$MARE = \frac{1}{n} \sum_{j=1}^n \frac{|e_j|}{|A_j|} \quad (4.15)$$

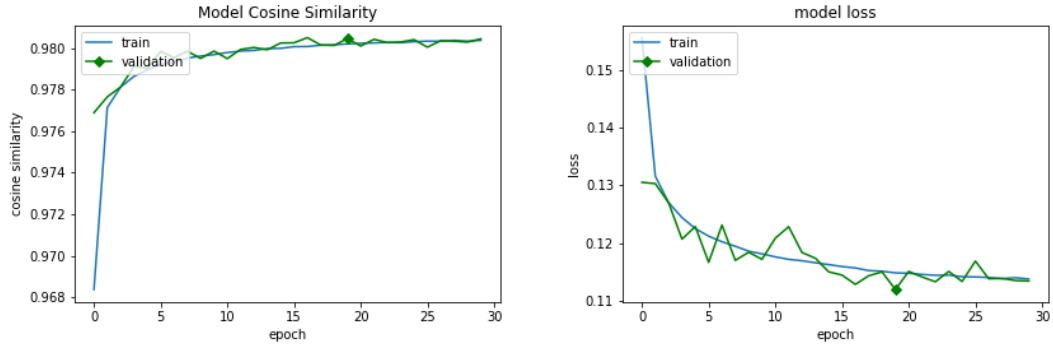
$$RAE = \sum_{j=1}^n \frac{|e_j|}{|A_j - \bar{A}|} \quad (4.16)$$

$$MRAE = \frac{1}{n} \sum_{j=1}^n \frac{|e_j|}{|A_j - \bar{A}|} \quad (4.17)$$

Denklem (4.5)'teki A_j gerçek değer, P_j tahmin edilen değer ve e_j ise hatadır (error). Denklem (4.8)'te ortalama mutlak hatanın, Denklem (4.9)'da ortalama karesel hatanın, Denklem (4.10)'da ortalama hata kareleri toplamı kökünün, Denklem (4.11)'de tezde kullanılan ortalama mutlak yüzde hatanın, Denklem (4.12)'de ağırlıklı mutlak yüzde hatanın hesaplanması verilmektedir. Ayrıca, Denklem (4.13)'te simetrik ortalama mutlak yüzde hatanın, Denklem (4.14)'te ortalama mutlak ölçekli hatanın, Denklem (4.15)'te ortalama mutlak bağıl hatanın, Denklem (4.16)'da göreceli Mutlak hatanın ve Denklem (4.16)'de ise ortalama göreceli mutlak hatanın hesaplanması verilmektedir. Denklemlerde kullanılan $\bar{A}$ değeri gerçek değerlerin ortalamasını, n değeri ise veri setinin boyutunu temsil etmektedir.

$$\text{benzerlik} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{P}}{\|\mathbf{A}\| \|\mathbf{P}\|} = \frac{\sum_{j=1}^n A_j P_j}{\sqrt{\sum_{j=1}^n A_j^2} \sqrt{\sum_{j=1}^n P_j^2}} \quad (4.18)$$

Ölçümlere ek olarak, eğitim sırasındaki her epoch işleminin sonunda elde edilen değerlere göre ilgili modelin doğrulama seti üzerinden Denklem (4.18)'de gösterilen kosinüs benzerliği (cosine similarity) ölçülmüştür. Her eğitimin sonunda ise bu değerler çizdirilerek kaydedilmiştir. Bir iç çarpım uzayının iki vektörü arasındaki ($\mathbf{A}$ ile $\mathbf{P}$) yakınlık bulunarak kosinüs benzerliği ölçülmektedir. Kosinüs benzerliği bahsedilen iki vektör arasındaki açının kosinüsü ile ölçülmektedir. Ayrıca bu parametre, bahsedilen iki vektörün aynı yönü gösterip göstermediği hakkında bilgi vermektedir [105]. Şekil 4.11'de, MLP ağı için kaydedilen rasgele bir eğitimin kosinüs benzerliği ve hata grafikleri verilmiştir. Çizdirilen grafiklerin verileri, eğitim sırasında toplanmaktadır ve yeşil kutu ile gösterilen değer, doğrulama seti üzerindeki en iyi performansının görüldüğü epoch değerini temsil etmektedir. Tahminler için, bu epoch değerindeki ağırlık parametreleri kullanılmaktadır.



Şekil 4.11. Her eğitimin sonunda kaydedilen örnek grafikler

4.3.5. İstatistiksel Analiz

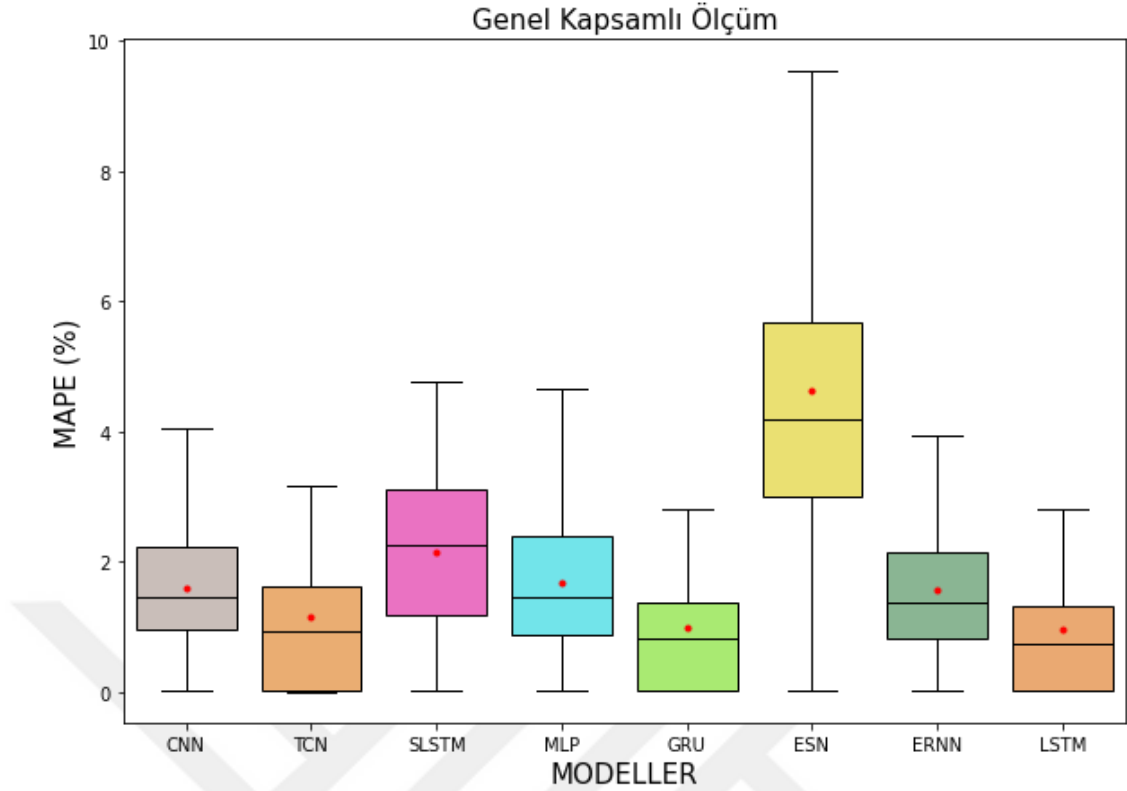
Tez çalışmasında, 8 tür derin öğrenme modelini tüm açılardan değerlendirmek için farklı performans metrikleri üzerinden istatistiksel analiz gerçekleştirilmiştir. Modeller, tahmin doğruluğu (MAPE) ve verimlilik (eğitim ve çıkarım süreleri) açısından en iyi sonuçlara göre sıralanmaktadır. Ayrıca, en kötü performanslar ve her bir mimari türü ile elde edilen sonuçların ortalaması da değerlendirilmiştir. Son olarak, hangi modellerin genel olarak daha iyi olduğunu görmek için bir sıralama yapılmıştır

5. ARAŞTIRMA BULGULARI VE TARTIŞMA

Tez çalışmasının bu bölümünde, Türkiye'nin kısa vadeli elektrik tüketim tahmini için sunulan yöntem ile gerçekleştirilmiş deney bulgularının gösterilmesi amaçlanmıştır. Google Colab ortamında gerçekleştirilen deneyler yaklaşık bir yıl sürmüştür. Ayrıca, yaklaşık olarak 2950 saat boyunca model eğitimi işlemi gerçekleştirilmiştir.

5.1. Modellerin Tahmin Doğruluğuna Etkisi

Analizin ilk kısmı, MAPE metriğini kullanarak her model için elde edilen tahmin doğruluğuna odaklanmaktadır. Tez çalışmasında kullanılan tüm farklı model mimarileri eğitildikten sonra, test verisi ile elde edilen sonuçlar Şekil 5.1.'de sunulmuştur. Kırmızı nokta ortalama MAPE'yi, kutu ise sonuçların çeyreklerini göstermektedir. Bazı grafiklerde, modellerin daha iyi görselleştirilmesine izin vermek için aykırı değerler çıkarılmıştır. İlk bakışta, ESN gibi bazı mimarilerin, diğerlerine kıyasla daha geniş bir MAPE dağılımı sunduğu görülmektedir. Bu nedenle bu tür mimarilerin parametrelendirmeye daha duyarlı oldukları tespit edilmiştir. Genel olarak, zaman serisi problemiyle başa çıkmak için özel olarak tasarlanmayan MLP ve ESN dışında, mimarilerin geri kalanı hemen hemen her durumda en iyi modele benzer tahmin doğruluğu elde edebileceği değerlendirilmektedir. Bu çıkarım, sonuç olarak birbirine yakın olan modellerin en düşük MAPE değerleri incelenerek yapılmaktadır. Daha geniş dağılımlar, optimal hiperparametre konfigürasyonunu bulmanın daha yüksek bir zorluğunu göstermektedir. Bu anlamda, CNN, TCN, GRU ve LSTM, iyi performansa sahip modellerin hızlı tasarımı için en uygun alternatiflerdir.



Şekil 5.1. Modeller için MAPE sonuçlarının dağılımını gösteren kutu grafikleri

Tez çalışmasındaki deneylerde kullanılan her bir veri kümesi için eğiten her bir mimarinin sonuçları, ayrıntılı olarak Tablo 5.1.'de sunulmuştur. 1'den 7'ye kadar gösterilen veri kümelerinin hangi öznelilikleri kapsadığı ileride bahsedilmektedir. GRU ve LSTM modellerin en iyi performansı göstermesi sebebiyle çok tutarlı bir yöntem olduğu görülmektedir. MLP ağlarının diğer mimarilerle faydalı bir karşılaştırma temeli olarak hizmet edebilecek basit ama etkili modeller olduğu değerlendirilmektedir fakat Tablo 5.2. incelendiğinde MLP ağının ESN'den sonra en kötü performansa sahip olduğu görülmektedir. Bunun sebebi, MLP'nin genel olarak, zaman serileriyle başa çıkmak için özel olarak tasarlanmamış olmasındandır. TCN modeli 4 farklı veri setinde en iyi sonucu elde etmiştir. ESN ve SLSTM modelleri, veri kümesine bağlı olarak çok farklı sonuçlar elde ettiği için daha kararsızdırlar. CNN ile TCN davranışı kıyaslandığında, CNN için biraz daha kötü olduğu çıkarımı yapılmıştır. Son olarak, ESN'in elde ettiği sonuçlar değerlendirildiğinde, çoğu veri seti için en kötü model olduğu tespit edilmiştir. Ortalama MAPE açısından değerlendirildiğinde, en iyi performansı LSTM'nin elde ettiği görülmektedir. Bu nedenle, LSTM'nin tekrarlayan sinir ağları arasında en iyi alternatif olduğu söylenebilmektedir. Ayrıca, evrişimli

modellerin düşük standart sapmaları nedeniyle, yüksek performanslı bir parametre seti bulmak çoğu modelden daha kolay olmaktadır.

Tablo 5.1. Modeller ve öznitelikler için elde edilen en iyi MAPE ölçümleri

Model	Farklı Öznitelikler İle Oluşturulan Veri Setleri							Ort. MAPE
	1	2	3	4	5	6	7	
MLP	0.0084	0.0090	0.0089	0.0081	0.0089	0.0082	0.0089	0.0086
ERNN	0.0138	0.0331	0.0349	0.0188	0.0174	0.0173	0.0176	0.0218
LSTM	0.0115	0.0070	0.0068	0.0067	0.0063	0.0067	0.0070	0.0074
SLSTM	0.0108	1.0228	1.0839	1.1599	1.0071	1.0388	0.0080	0.7616
GRU	0.0081	0.0077	0.0065	0.0066	0.0069	0.0075	0.0073	0.0072
ESN	0.0538	0.0429	0.0472	0.0481	0.0395	0.0467	0.0099	0.0411
CNN	0.0140	0.0131	0.0150	0.0141	0.0126	0.0150	0.0135	0.0139
TCN	0.6425	0.0062	0.0072	0.0063	0.0063	0.0063	0.0064	0.0973
Ort. MAPE	0.0954	0.1427	0.1513	0.1586	0.1381	0.1433	0.0098	

Tablo 5.2. Modellerin ortalama, en düşük ve en yüksek MAPE değerleri

Model	Ortalama MAPE	En Düşük MAPE	En Yüksek MAPE
MLP	1.6709	0.0081	5.2078
ERNN	1.5713	0.0138	27.2490
LSTM	0.9497	0.0063	2.8001
SLSTM	2.1541	0.0080	4.7791
GRU	0.9927	0.0065	2.7980
ESN	4.6159	0.0099	16.5364
CNN	1.5857	0.0126	4.2724
TCN	1.1435	0.0062	3.1611

5.2. Özniteliklerin Tahmin Doğruluğuna Etkisi

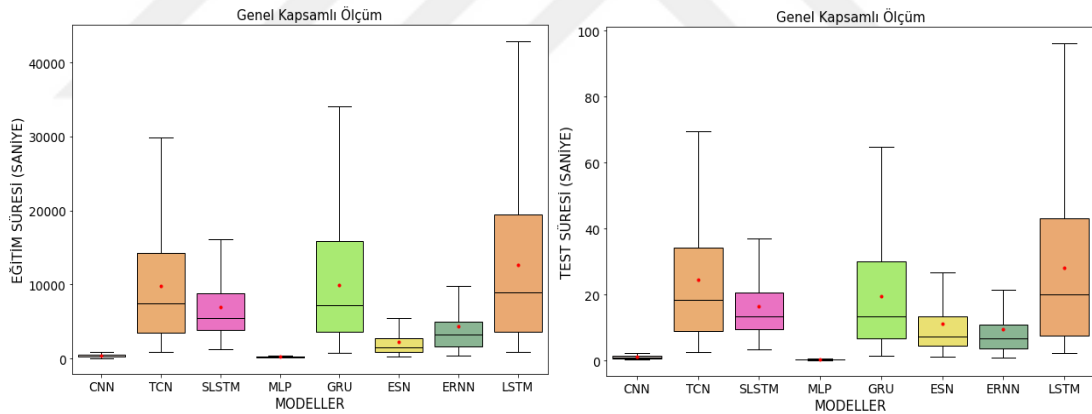
Model eğitime dâhil edilen öznitelik sütunları ve tüm modelleri kapsayan ortalama MAPE ve ortalama eğitim süreleri Tablo 5.3'te verilmiştir. Sonuçlar incelendiğinde, GSYH değişkenin STLF problemi üzerinde bir etkisinin olmadığı gibi model eğitimi sırasında gürültüye yol açarak model performansını düşürdüğü değerlendirilmiştir. GSYH değerinin, daha uzun zaman ufkuna sahip tahmin problemleri için etkili olduğu fikri ile literatürdeki çalışmalara paralel bir sonuç elde edilmiştir. Ölçüm yapılan zamanın tatil günü sınırları içerisinde olup olmadığını gösteren öznitelik sütununun ise model performansı üzerinde büyük bir etkisinin olduğu tespit edilmiştir. Modele bu bilginin sağlanmasının tahmin doğruluğunu arttırdığı görülmüştür. Bu yüzden STLF problemi için takvimsel özelliklerin doğrudan etkili olduğu değerlendirilmektedir. Literatürdeki çalışmalarda, popülasyon değerinin uzun vadeli tahminler üzerinde etkili olduğu sunulmasına rağmen STLF problemi için de olumlu bir etkisi olduğu görülmüştür. Tez çalışmasında önerilen ağırlıklandırılmış toplam hissedilen sıcaklık değerinin, STLF problemi için tahmin performansını bir miktar arttırdığı gözlemlenmiştir. Önceki günün ve önceki haftanın elektrik enerjisi tüketim bilgilerini sağlayan sütunların ise eğitilen modelin trendi yakalaması açısından etkili olduğu incelenmiştir. Ayrıca elde edilen bulgular sonucunda, model kurulumlarını gerçekleştirmeden önce öznitelik seçiminin (feature selection) çok önemli bir etken olduğu sonucuna varılmıştır. Gerçekleştirilen tez çalışmasında öznitelik seçimi rasgele olarak gerçekleştirilip hiçbir yöntem kullanılmamıştır.

Tablo 5.3. Özniteliklerin ortalama MAPE ve süre ölçümleri

Nu.	Öznitelik Matrisi	Öznitelik Matrisi Açıklaması	Ort. MAPE	Ort. Süre (Sn)
1	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	Bütün öznitelikler bulunuyor.	1.9293	4009
2	[0, 1, 2, 3, 4, 5, 6, 7, 8]	Sadece GSYH bilgisi bulunmuyor.	1.9153	2804
3	[0, 1, 2, 3, 4, 5, 6, 8, 9]	Sadece tatil bilgisi bulunmuyor.	2.3161	2410
4	[0, 1, 2, 3, 4, 5, 6, 7, 9]	Sadece nüfus bilgisi bulunmuyor.	2.0881	2450
5	[0, 1, 2, 3, 7, 8, 9]	Sadece top. his. sıcaklık. bulunmuyor.	2.0502	2465
6	[0, 1, 4, 5, 6, 7, 8, 9]	Sadece önceki tük. bulunmuyor.	2.0536	2329
7	[0]	Sadece anlık tüketim bulunuyor.	1.4373	2411

5.3. Modellerin Hesaplama Zamanına Etkisi

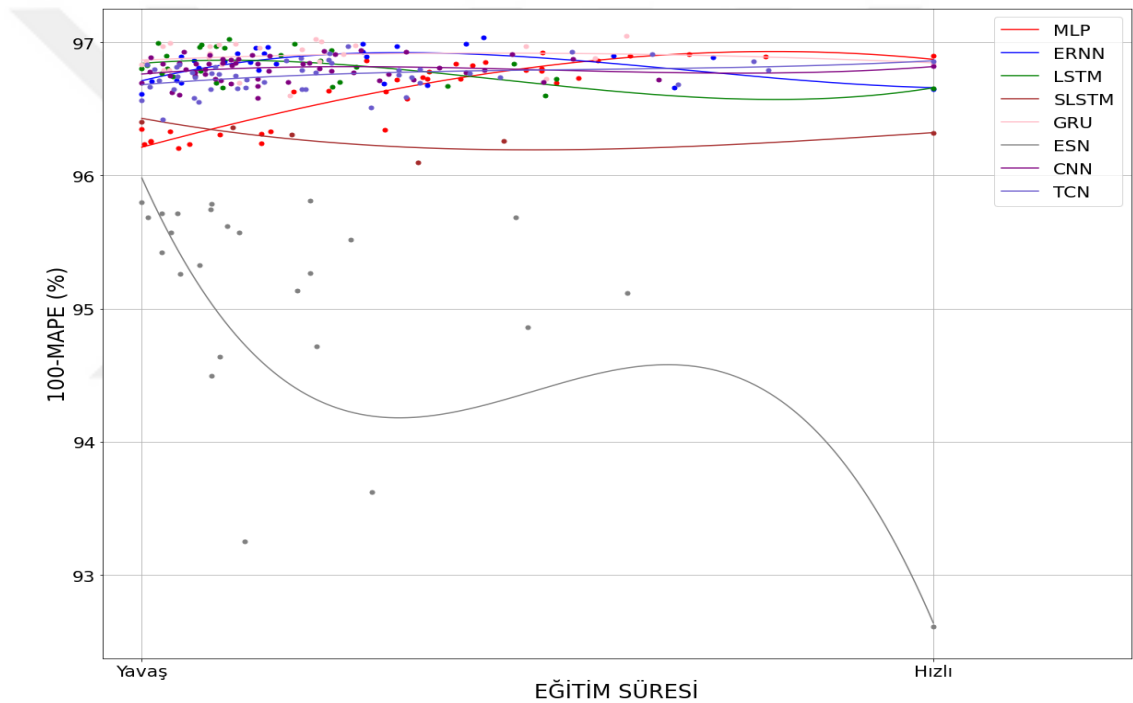
Tahmin doğruluğundan farklı olarak, derin öğrenme mimarileri hesaplama verimliliği açısından da değerlendirilmektedir. Eğitim ve test süresi açısından her bir mimarinin dağılımı Şekil 5.2’de sunulmuştur. Elde edilen sonuçlardan çıkan şekil incelendiğinde, MLP'nin en hızlı model olduğu görülmektedir. Ayrıca onu, arkasından CNN takip etmektedir. CNN ile diğer modeller kıyaslandığında, eğitim ve test süresi açısından ciddi fark söz konusu olmaktadır. Tekrarlayan sinir ağlarında, LSTM ve GRU hem benzer şekilde performans göstermekte hem de en yavaş mimari sınıfına girmektedirler. Genel olarak, tüm mimarilerin analizi, eğitim ve test sürelerini karşılaştırırken benzer olduğu tespit edilmiştir. Ancak, ağırlıklarının çoğu eğitilemez olduğundan ESN modelleri bu standardı karşılayamaz, bu da verimli eğitim ve yavaş çıkarım süresi ile sonuçlanmaktadır. TCN mimarisi ise GRU ve LSTM modelleri ile ortalama eğitim ve test süresi açısından karşılaştırılabilir sonuçlar sunmaktadır. Sonuç olarak, Bu nedenle, çok derin tekrarlayan modeller tasarlamak kısıtlamaları sebebiyle uygun olmamaktadır.



Şekil 5.2. Eğitim ve test süresi açısından tüm modellerin dağılımı

Şekil 5.2’in ilk kısmında, tez çalışmasında kullanılan modellerin eğitim süreleri ve Şekil 5.2’in ikinci kısmında ise tez çalışmasında kullanılan modellerin test süreleri, kutu grafiği olarak verilmektedir.

Eğitim süreleri 0 ile 1 arasına ölçeklenmiş modellerin eğitim sürelerine göre performans değerleri Şekil 5.3.'de sunulmuştur. Grafik, her derin öğrenme modeli için hesaplama süresine karşı doğruluğunu göstermektedir. Tahmin performansı ve hesaplama süresi açısından incelendiğinde, en iyi ağ modelinin LSTM olduğu görülmektedir. Sunulan grafik incelendiğinde, çoğu model için eğitim süresi arttıkça performansının arttığını fakat belirli bir zaman sonra doyuma ulaşp eski verimini koruyamadığı değerlendirilmektedir. Ayrıca, ESN modeli hariç olmak üzere modellerin belirli bir tahmin doğruluğuna ulaşması için asgari düzeyde zamana ihtiyaç duyduğu ve söz konusu modellere bu zamandan fazlasının sunulmasının tahmin doğruluğunu çok fazla değiştirmedığı sonucuna varılmıştır.



Şekil 5.3. Modellerin normalize edilmiş eğitim sürelerine göre performansları

5.4. Modellerin Mimarisi Yapısının Tahmin Doğruluğuna Etkisi

İncelenen derin öğrenme modelleri için en uygun mimariyi tasarlamak, önemli ölçüde uzmanlık gerektiren çok karmaşık bir iştir. Bu nedenle, her mimari için farklı parametrelerle elde edilen sonuçlar açısından analiz edilmiştir. Tez çalışmasında gerçekleştirilen bu analizde, model mimarileri için sıralama gerçekleştirilirken sekiz ağ türü için ayrı ayrı olarak değerlendirilmiştir. Farklı eğitim hiperparametreleriyle, her model konfigürasyonu için eğitim işlemi gerçekleştirilmiştir.

Tablo 5.4. Farklı MLP konfigürasyonları için ölçüm değerleri

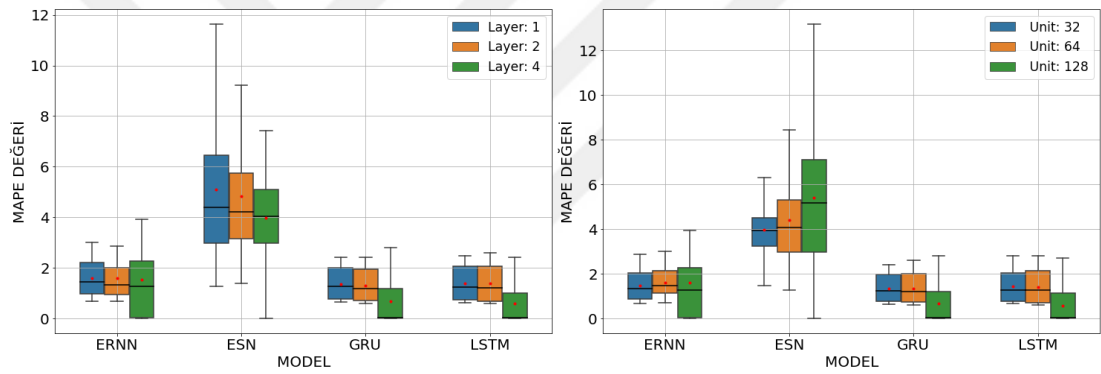
Nu.	Gizli Katman Yapısı	Ortalama MAPE	Ortalama Eğitim
1	[32, 64, 128, 64, 32]	0.8952	221
2	[64, 32]	1.7410	279
3	[32, 64]	1.7611	248
4	[32, 64, 128]	1.7641	269
5	[32]	1.7651	241
6	[128, 64, 32]	1.7903	334
7	[32, 16, 8]	1.8362	255
8	[8, 16, 32]	1.8403	242
9	[8, 16]	1.8504	231
10	[8, 16, 32, 16, 8]	1.8543	254
11	[8]	1.8596	220
12	[16, 8]	1.8694	244

MLP ağıının farklı katman yapılarına göre ortalama eğitim sonuçları Tablo 5.4'te sunulmuştur. En üst sıradaki MLP modeli, toplam 320 nöron ile 5 gizli katmandan oluşmaktadır. MLP tasarımının iyi çalışılmış bir alan olduğu göz önüne alındığında, tablodaki sonuçlar literatürü doğrulamaktadır. Sunulan tablo incelendiğinde, gizli nöron sayısının doğrudan model performansı üzerinde bir etkisinin olmadığı görülmektedir. Ayrıca, karmaşık yapıya sahip bazı ağların daha az karmaşık yapıya sahip diğer ağlardan daha az performans sergilediği tespit edilmesine rağmen en karmaşık ağ en yüksek performansı göstermiştir.

Tablo 5.5. Farklı LSTM konfigürasyonları için ölçüm değerleri

Nu.	Model Parametreleri		Ortalama MAPE	Ortalama Eğitim Süresi (Sn)
	Katman	Birim		
1	4	128	0.2580	22821
2	2	32	1.3671	4094
3	1	128	1.3805	4818
4	1	64	1.3865	2567
5	2	64	1.3948	5255
6	1	32	1.3970	1975
7	2	128	1.4407	9503
8	4	64	1.4718	10450
9	4	32	1.5490	7690

Tekrarlayan sinir ağı mimarileri arasında en iyi performans LSTM modelleri ile elde edilmiştir. Deneyler sonucunda, farklı LSTM model konfigürasyonları için bulunan MAPE ölçümleri, Tablo 5.5’te gösterilmektedir. En iyi sonuçlar, 4 katmana sahip ve 128 birimden oluşan mimariden elde edilmiştir. Ayrıca, en yüksek performansa sahip ağ konfigürasyonu için ölçülen eğitim süresi diğer konfigürasyonlar ile kıyaslandığında en yüksek değere sahiptir. Farklı RNN mimarilerinin sonuçları, 2 ana parametreye bağlı olarak Şekil 5.4.’te sunulmuştur. LSTM, ESN ve GRU için yapılan inceleme ile katman sayısı parametresi artırıldığında daha iyi ortalama sonuçlar elde edildiği tespit edilmiştir. ERNN modelinin özellikle 4 katmanlı yapısında, performans düştüğü gözlemlenmiştir. Ayrıca, birim sayısı parametresini analiz ederken, LSTM, ERNN ve GRU’nun benzer davranışlara sahip olduğu ve özellikle 128 birimlik yapıda daha iyi ortalama sonuçlar elde ettikleri gözlemlenmiştir.



Şekil 5.4. Farklı konfigürasyonlara sahip farklı RNN mimarileri için MAPE değerleri

Şekil 5.4’te tez çalışmasında kullanılan farklı konfigürasyonlara sahip çeşitli RNN mimarileri için katman ve birim sayılarına göre ortalama MAPE değerleri kutu grafikleri ile gösterilmiştir. İlk grafik, 1 (mavi), 2 (turuncu) ve 4 (yeşil) katmana sahip çeşitli RNN modellerinin ayrı ayrı ortalama MAPE değerlerini, ikinci grafik ise 32 (mavi), 64 (turuncu) ve 128 (yeşil) birime sahip çeşitli RNN modellerinin ayrı ayrı ortalama MAPE değerlerini kutu grafikleri ile göstermektedir.

CNN model konfigürasyonlarının ortalama MAPE değerine göre sıralamaları Tablo 5.6.'da sunulmuştur. Tahmin doğruluğunun genellikle evrişim katmanlarının sayısı ile orantılı olduğu fark edilmiştir. Filtre sayısı açısından ise ilk bakışta önemli bir fark olmadığı tespit edilmiştir. Ayrıca en iyi tahminlerin maksimum gruplama (max-pooling) içermeyen modellerden elde edilmiştir ve bu da bu popüler görüntü işleme işleminin zaman serisi tahmini için uygun olmadığını düşündürmektedir.

Tablo 5.6. Farklı CNN konfigürasyonları için ölçüm değerleri

Nu.	Evrişimli Blokların Yapısı	Ortalama MAPE	Ortalama Eğitim
1	[[64, 7, 0], [64, 5, 0], [64, 2, 0]]	1.1206	1004.0131
2	[[16, 7, 0], [16, 5, 0], [16, 2, 0]]	1.4789	380.4661
3	[[32, 5, 0], [32, 3, 0]]	1.4968	352.2998
4	[[16, 5, 0], [16, 3, 0]]	1.4902	279.2388
5	[[64, 5, 0], [64, 3, 0]]	1.5100	577.2208
6	[[32, 7, 0], [32, 5, 0], [32, 2, 0]]	1.5313	535.8508
7	[[64, 3, 0]]	1.5743	252.4033
8	[[16, 3, 0]]	1.5927	169.8535
9	[[32, 3, 0]]	1.5966	199.5620
10	[[64, 3, 2]]	1.6133	292.6250
11	[[32, 5, 2], [32, 3, 2]]	1.6402	362.6795
12	[[64, 5, 2], [64, 3, 2]]	1.6467	521.1283
13	[[32, 3, 2]]	1.6791	232.7414
14	[[16, 3, 2]]	1.6860	196.6678
15	[[16, 5, 2], [16, 3, 2]]	1.7000	290.5257
16	[[64, 7, 2], [64, 5, 2], [64, 2, 2]]	1.7429	708.7024
17	[[32, 7, 2], [32, 5, 2], [32, 2, 2]]	1.7641	469.8698
18	[[16, 7, 2], [16, 5, 2], [16, 2, 2]]	1.8601	360.8586

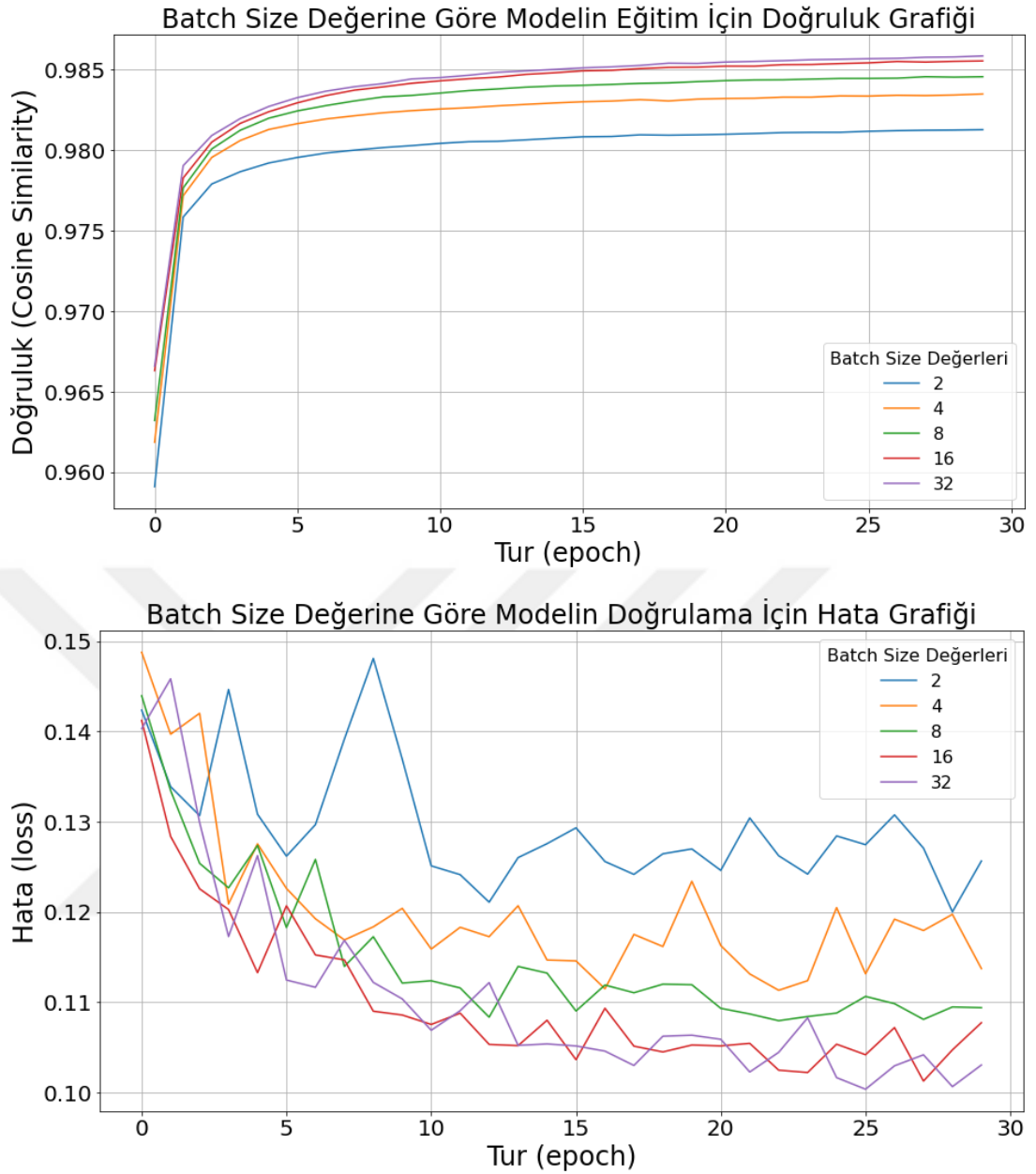
TCN modellerinin farklı konfigürasyonları ile elde edilen sonuçları Tablo 5.7'de sunulmuştur. Genel olarak en iyi ağ, 4 genişletilmiş evrişim katmanı (dilations), 3 uzunluğunda bir çekirdek (kernel Size) ve 32 filtre (nb filters) içeren tek bir TCN katmanı ile (stacks) tasarlanmıştır. Birkaç genişletilmiş katmana sahip tek katmanlı modellerin daha karmaşık modellerden daha iyi performans gösterdiği değerlendirilmektedir. Ayrıca, düşük sayıda filtre içeren modellerin daha yüksek performans gösterdiği tespit edilmiştir.

Tablo 5.7. Farklı TCN konfigürasyonları için ölçüm değerleri

Nu.	Model Parametreleri				Ortalama MAPE	Ortalama Eğitim
	Stacks	Nb Filters	Dilations	Kernel		
1	1	32	[1, 2, 4, 8]	3	0.2240	17375
2	1	32	[1, 2, 4, 8]	6	1.4494	8751
3	1	32	[1, 2, 4, 8, 16]	3	1.4812	7057
4	1	32	[1, 2, 4, 8, 16]	6	1.4857	5375
5	1	64	[1, 2, 4, 8]	3	1.5122	12684
6	1	64	[1, 2, 4, 8]	6	1.5134	13068
7	1	64	[1, 2, 4, 8, 16]	3	1.5205	2318
8	1	64	[1, 2, 4, 8, 16]	6	1.5443	3144
9	3	32	[1, 2, 4, 8]	3	1.5773	6155
10	3	32	[1, 2, 4, 8]	6	1.5827	4880
11	3	32	[1, 2, 4, 8, 16]	3	1.5875	6868
12	3	32	[1, 2, 4, 8, 16]	6	1.6131	2399
13	3	64	[1, 2, 4, 8]	3	1.6311	10698
14	3	64	[1, 2, 4, 8]	6	1.6433	4890
15	3	64	[1, 2, 4, 8, 16]	3	1.7047	3613
16	3	64	[1, 2, 4, 8, 16]	6	1.7425	1790

5.5. Model Hiperparametrelerinin Eğitim Dinamikleri Üzerindeki Etkisi

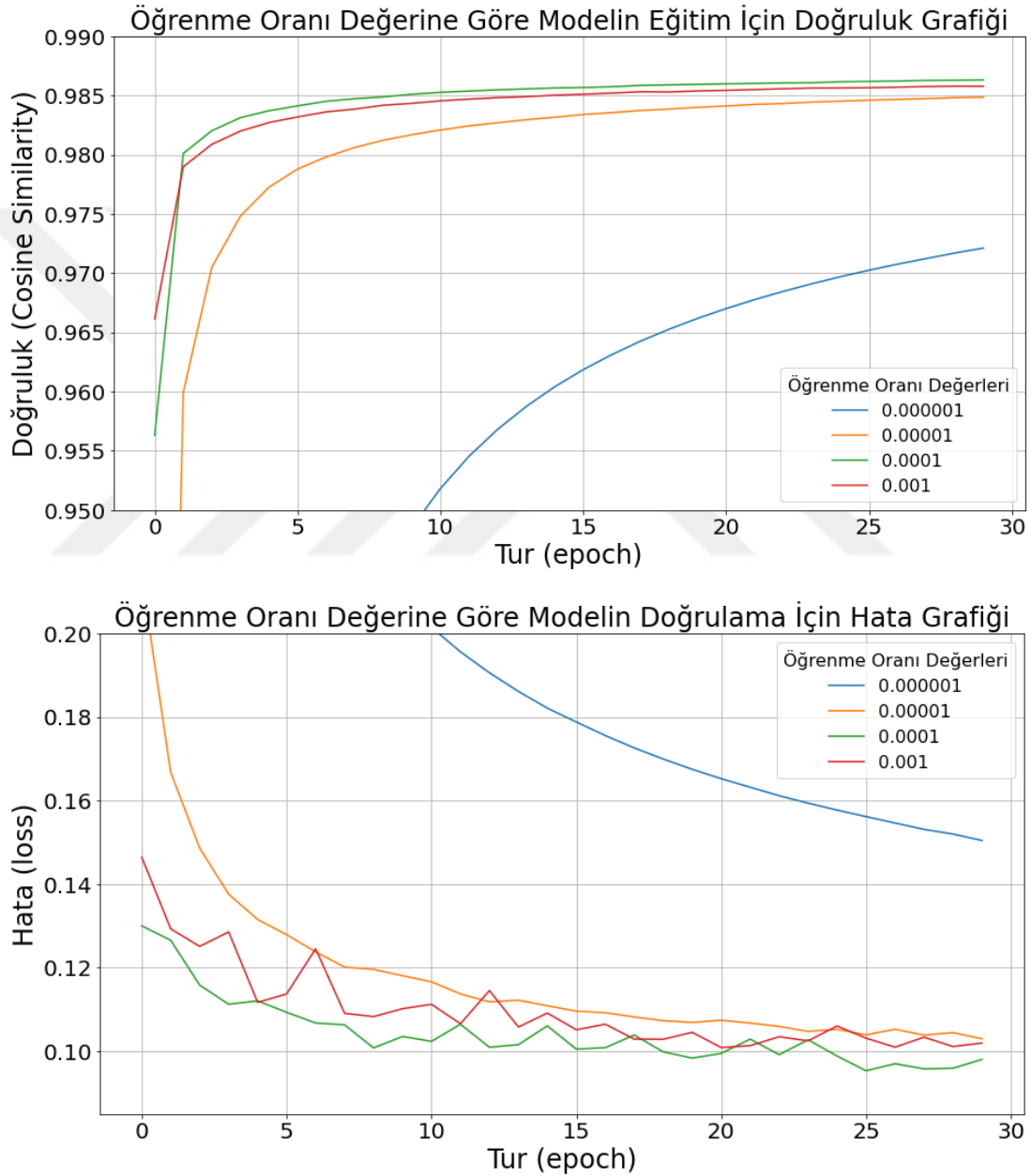
Daha önceki bölümlerde bahsedilen (learning rate), parti boyutu (batch size) ve optimize edici (optimizer) hiperparametrelerinin, model eğitim sürecini nasıl etkilediği araştırılmıştır. Her hiperparametrenin farklı değerleri ile MLP modeli eğitilip sonuçlar kaydedilmiştir. Söz konusu deneyler gerçekleştirilirken MLP ağının tercih edilmesinin sebebi, hızlı eğitebiliyor olmasındandır. Bu sayede kısa süre içerisinde çeşitli hiperparametrenin derin öğrenme modellerini nasıl etkilediği konusunda fikir sahibi olunması amaçlanmıştır. İlk önce, batch size hiperparametresinin eğitim dinamikleri üzerindeki etkisi araştırılmıştır. Yapılan deneyler sonucunda eğitim ve doğrulama seti için ölçülen doğruluk değerlerinin, batch size ile doğrusal bir ilişkisinin olduğu tespit edilmiştir. Ayrıca, doğrulama seti için hata değerlerinin ise batch size ile ters bir ilişkisinin olduğu değerlendirilmiştir. Daha küçük batch size değeri, modelin genelleme kabiliyetinde önemli bir bozulmaya yol açmaktadır. Bunun sebebi, küçük batch size kullanılan yöntemlerin, hatayı minimize edememesinden kaynaklanmaktadır. Ek olarak, öğrenme süresini ve modelin kararlılığını batch size parametresinin etkilediği sonucuna varılmıştır. Söz konusu ölçüm grafikleri Şekil 5.5.'da gösterilmiştir.



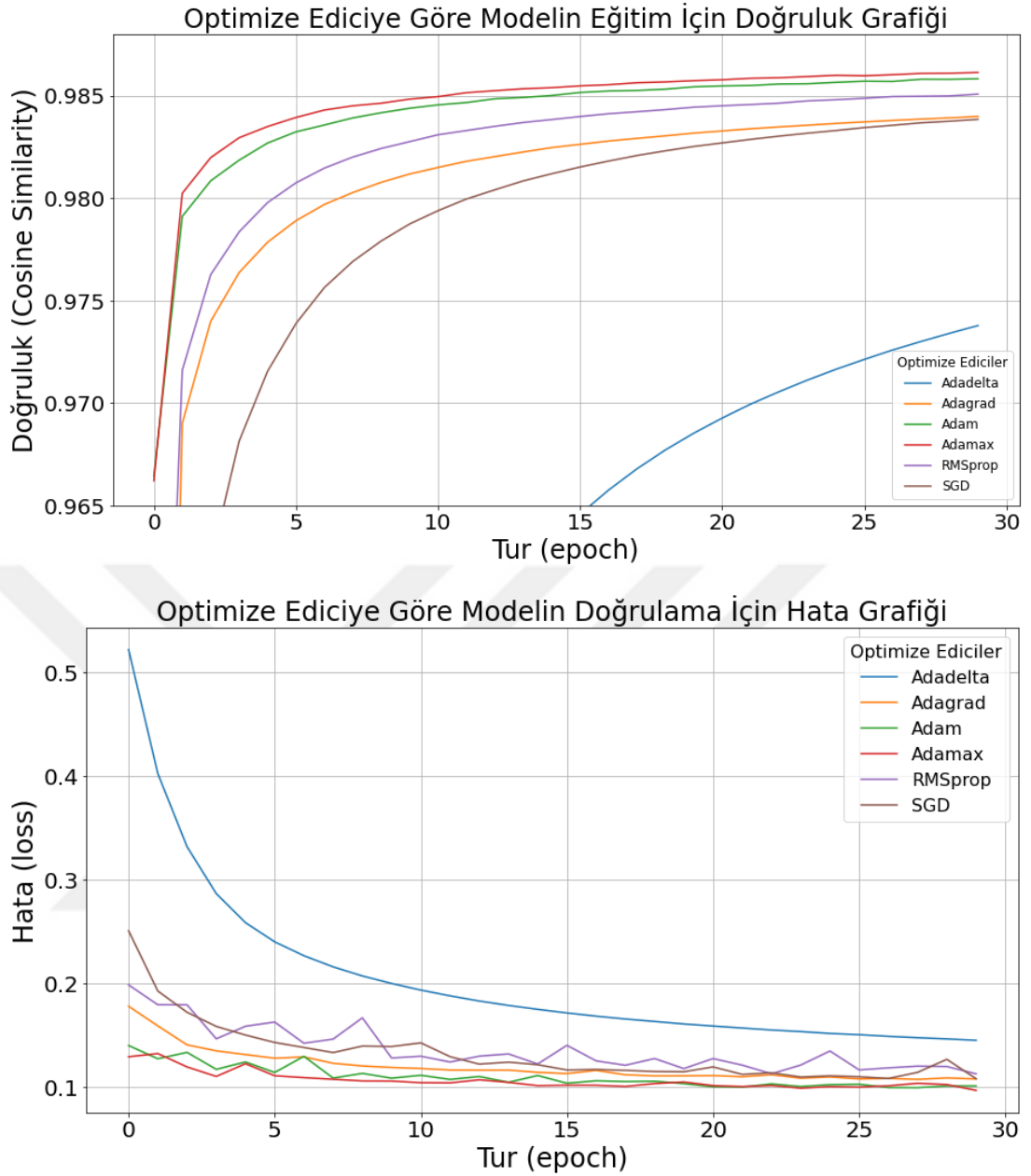
Şekil 5.5. MLP modeli için kullanılan batch size değerlerine göre ölçümler

İkinci olarak, öğrenme oranı (learning rate) hiperparametresinin eğitim dinamikleri üzerindeki etkisi araştırılmıştır. Modelin ilgili probleme adapte olma süresi, öğrenme oranı tarafından kontrol edilmektedir. Deney aşamasında, öğrenme oranı için 0 ile 1 arasında küçük değerler kullanılmıştır. Her güncellemede ağırlıklarda yapılan daha küçük değişiklikler göz önüne alındığında, daha küçük öğrenme oranları daha fazla epoch gerektirirken, daha büyük öğrenme oranları hızlı değişikliklerle sonuçlanması sebebiyle ve daha az epoch gerektirmektedir. Çok büyük bir öğrenme oranının, modelin çok hızlı bir şekilde optimal olmayan bir çözüme yakınsamasına

neden olabileceği ve çok küçük bir öğrenme oranının ise eğitim sürecinde tıkanmalara neden olabileceği tespit edilmiştir. Yapılan deneyler sonucunda eğitim ve doğrulama seti için ölçülen doğruluk değerlerinin, öğrenme oranı ile genellikle doğrusal bir ilişkisinin olduğu değerlendirilmiştir. Ayrıca, doğrulama seti için hata değerlerinin ise öğrenme oranı ile genellikle ters bir ilişkisinin olduğu sonucuna varılmıştır. Söz konusu ölçüm grafikleri Şekil 5.6.'da gösterilmiştir.



Şekil 5.6. MLP için öğrenme oranı (learning rate) değerlerine göre ölçümler



Şekil 5.7. MLP için optimize edicilere (optimizer) göre eğitim ölçümleri

Son olarak, optimize edicinin (optimizer) eğitim dinamikleri üzerindeki etkisi araştırılmıştır. Model eğitimi ve tahmin edilen çıktı üzerinde etkisi olan ağ parametrelerini güncellemek ve hesaplamak için optimize edici kullanılmaktadır. Bu sayede kayıp fonksiyonu en aza indirilerek optimal bir değere yaklaşması sağlanır. Derin sinir ağı eğitiminin özü budur. Bu nedenle, derin öğrenme modellerini eğitmek için uygun bir optimize edici seçmek önemlidir. Gerçekleştirilen deneylerde, Adadelta, Adagrad, Adam, Adamax, RMSprop ve SGD gibi birkaç yaygın optimize

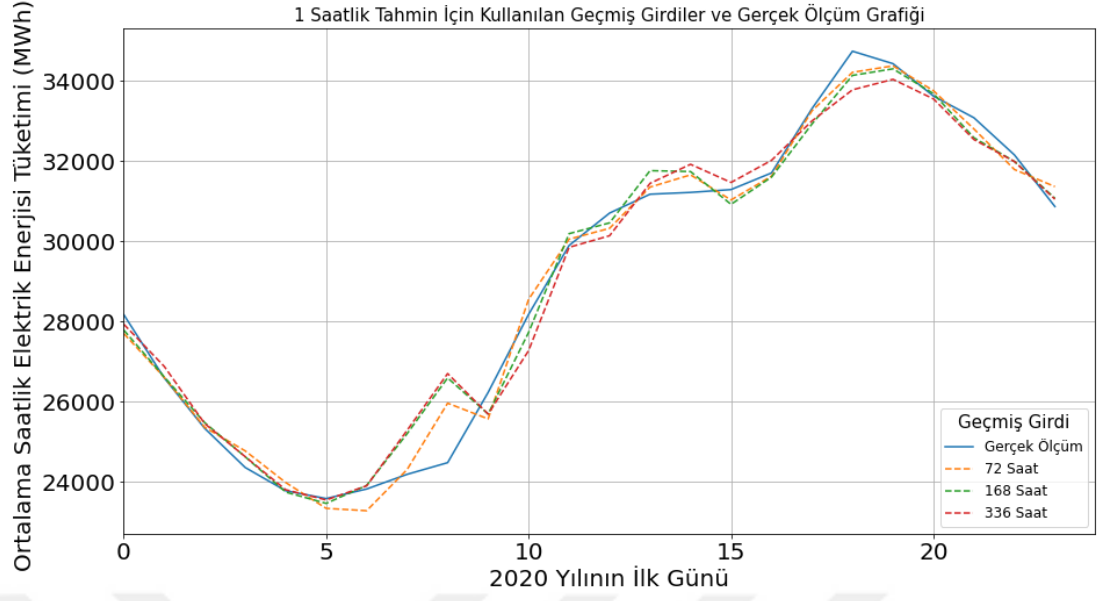
edici, Şekil 5.7.'de gösterildiği gibi kullanılmıştır. Doğruluk açısından en yüksek performansa sahip optimize edici Adam ve Adamax olduğu tespit edilmiştir. Adadelta optimize edicinin ise kötü doğruluk performansına rağmen hata grafiğinde kararlı bir inişe sahip olduğu değerlendirilmektedir. Ayrıca diğer optimize edicilerin doğruluk performansı açısından birbirlerine yakın olduğu sonucuna varılmıştır.

5.6. Geçmiş Değerlerin ve Tahmin Ufkunun Tahmin Doğruluğuna Etkisi

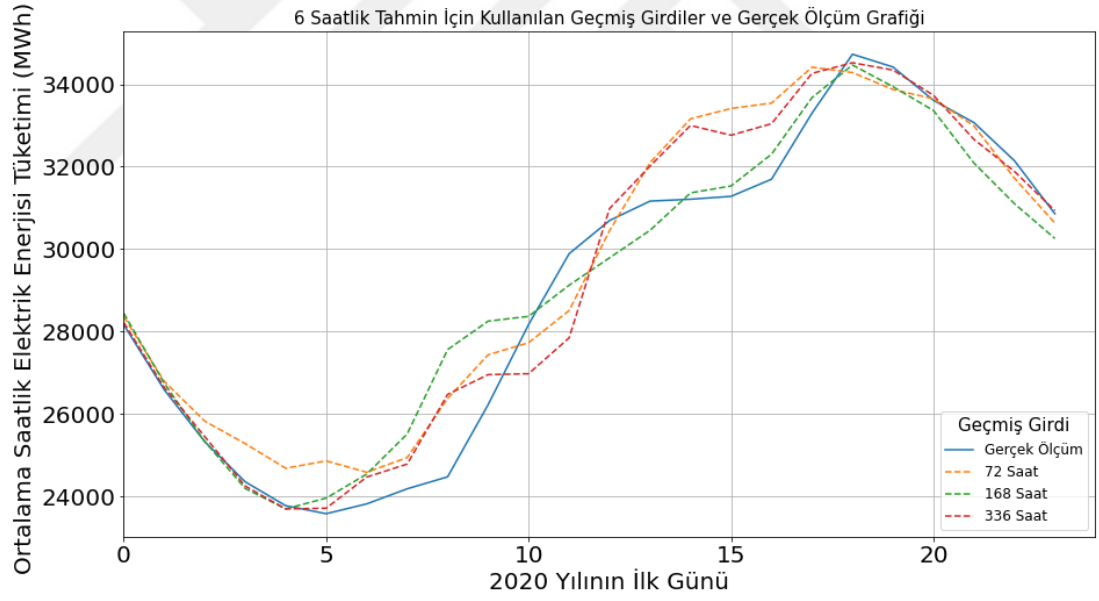
Gerçekleştirilen tez çalışmasında, geçmiş tarih ve tahmin ufku ile ilgili çeşitli değerler kullanılarak modeller eğitilmiştir. Kullanılan geçmiş tarih değeri ve tahmin ufkuna göre söz konusu modellerin performanslarında farklılık görülmüştür. Tablo 5.8., Şekil 5.8., Şekil 5.9. ve Şekil 5.10.'da sunulan deney sonuçları incelendiğinde, tahmin ufku arttıkça ortalama MAPE değerinde yükselme (tahmin performansında düşüş) gözlemlenmiştir. Ayrıca, 1 ve 6 saatlik tahmin için en ideal geçmiş değerin 72 saat, 24 saatlik tahmin için ise en ideal geçmiş değerin 336 saat olduğu tespit edilmiştir. 24 saatlik tahmin ufku için 72 saatlik geçmiş değerin kullanılması en kötü MAPE performansına yol açmıştır. Bunun sebebi, modelin tahmin işlemi gerçekleştirmek için ihtiyaç duyduğu bilginin yetersiz kalmasıdır.

Tablo 5.8. Eğitimde kullanılan geçmiş değer ve tahmin ufkuna göre ortalama MAPE

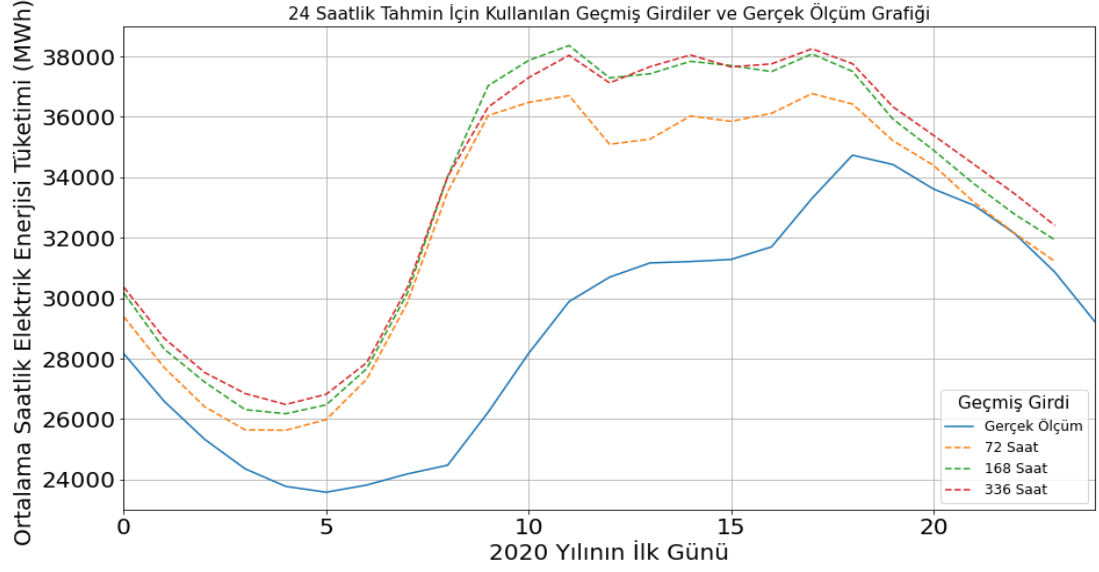
Tahmin Ufku	Kullanılan Geçmiş Girdi		
	72 Saat	168 Saat	336 Saat
1 Saat	1.0031	1.2629	1.5096
6 Saat	1.7272	1.7310	2.0875
24 Saat	3.0120	2.5524	2.9876



Şekil 5.8. 72, 168 ve 336 saatlik geçmiş değere göre 1 saatlik tahmin grafikleri



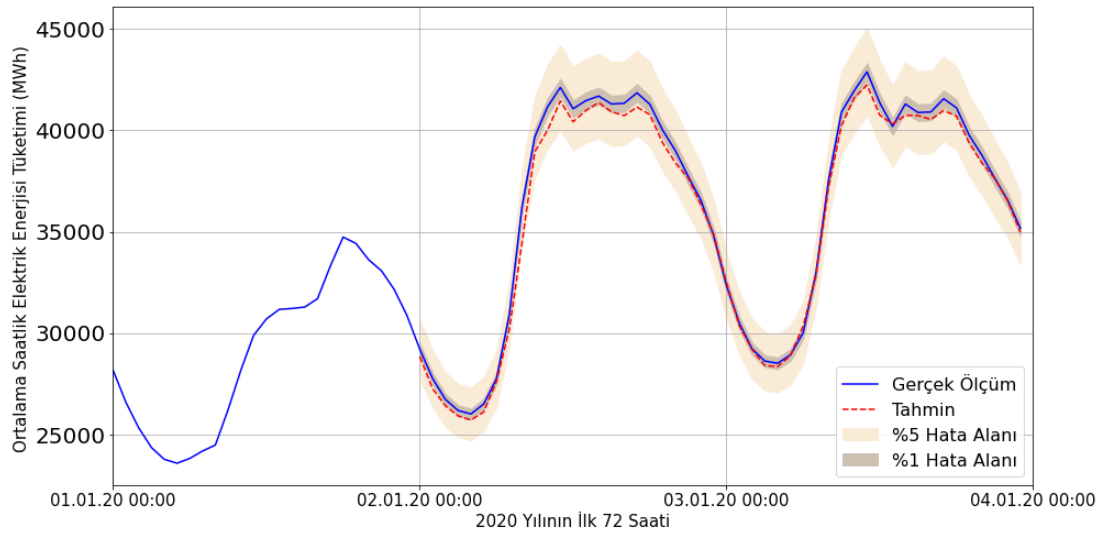
Şekil 5.9. 72, 168 ve 336 saatlik geçmiş değere göre 6 saatlik tahmin grafikleri



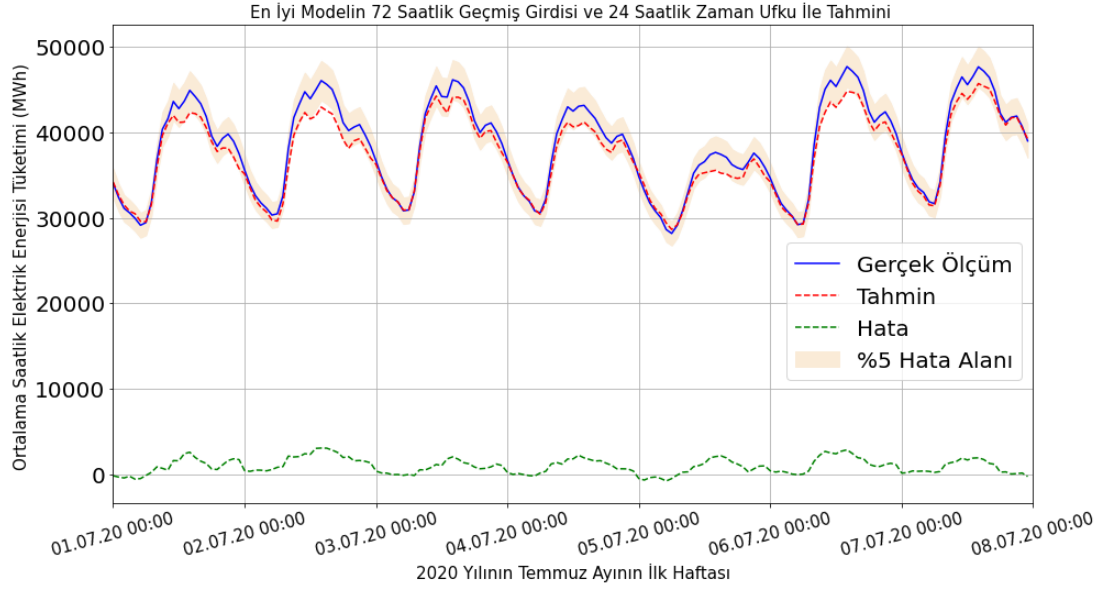
Şekil 5.10. 72, 168 ve 336 saatlik geçmiş değere göre 24 saatlik tahmin grafikleri

5.7. En İyi Model İle Gerçekleştirilen Tahminler

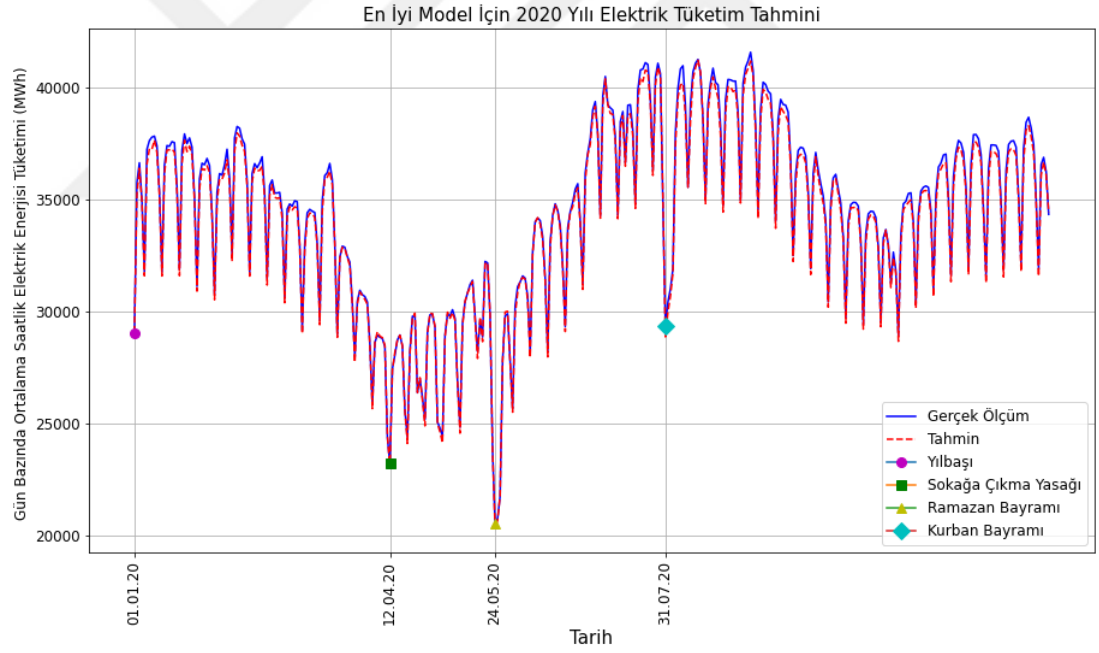
Elde edilen tüm model çıktıları incelendiğinde, en iyi tahmin performansı LSTM mimarisine sahip, 4 katmanlı ve 128 birimli yapıda elde edilmiştir. Şekil 5.11.'de, 2021 yılının ilk 24 ile 72 saati arasındaki elektrik enerjisi tüketim tahmini gösterilmiştir. Şekil 5.12.'de, temmuz ayının ilk haftasının tahmin ve hata grafikleri gösterilmiştir. Şekil 5.13.'de ise en iyi modelin 1 yıllık tahmini gösterilmiştir.



Şekil 5.11. En iyi model ile gerçekleştirilen 48 saatlik tahmin (1 saatlik zaman ufku)



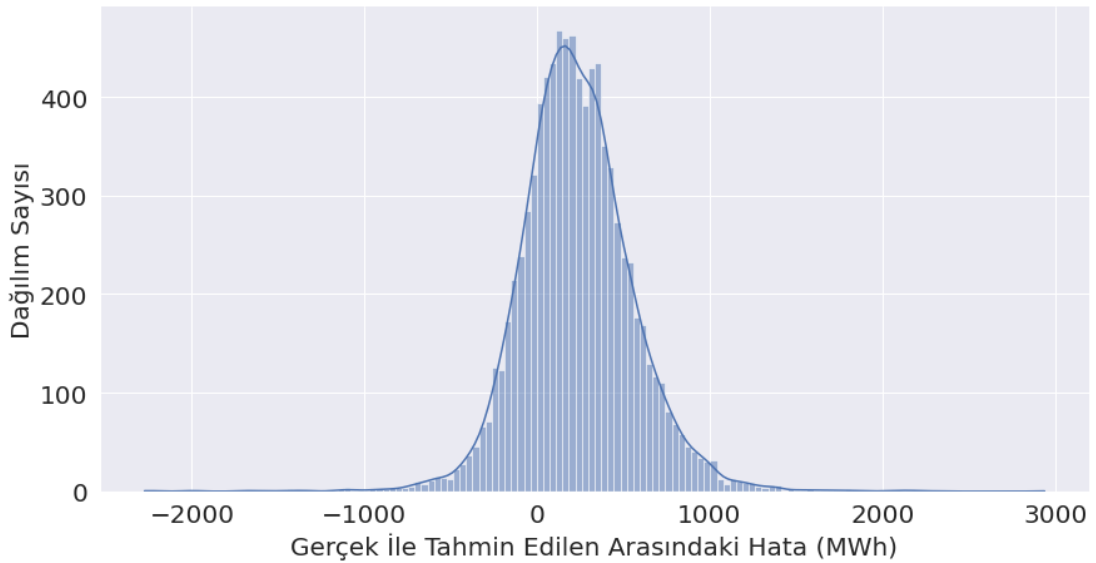
Şekil 5.12. En iyi model ile gerçekleştirilen 1 haftalık tahmin (24 saatlik zaman ufku)



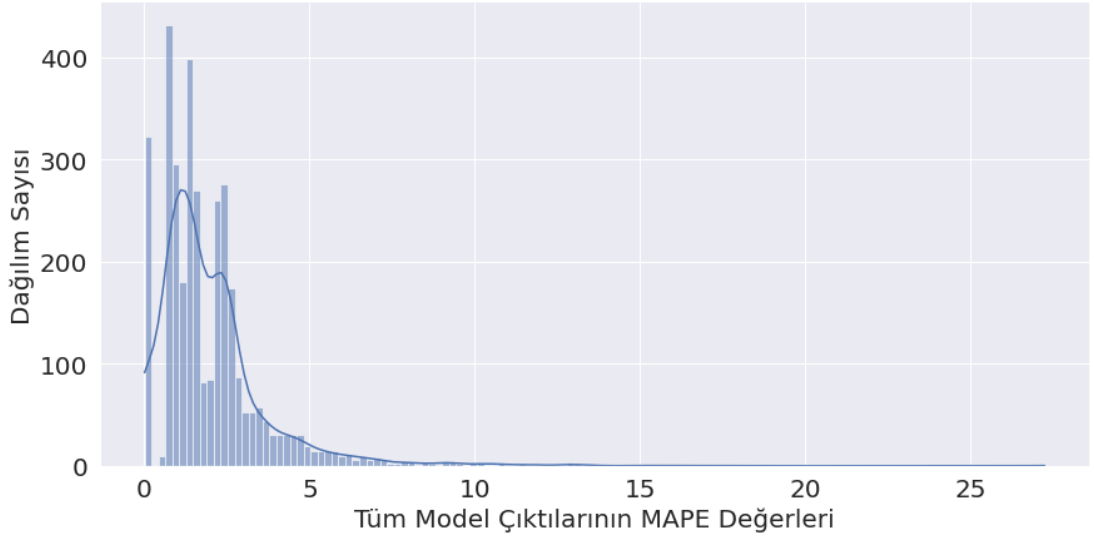
Şekil 5.13. En iyi model ile gerçekleştirilen 1 yıllık tahmin (1 saatlik zaman ufku)

5.8. Hata ve MAPE Dağılımı (Loss and MAPE Distribution)

Normal dağılım (Normal distribution) veya diğer bir adıyla Gauss tipi dağılım, ortalamaya yakın verilerin, ortalamadan uzaktaki verilere göre daha sık meydana geldiğini gösteren bir olasılık dağılımıdır. Herhangi bir olasılık dağılımında olduğu gibi, normal dağılım bir değişkenin değerlerinin nasıl dağıldığını tanımlamaktadır. Şekil 5.14'te en iyi model için hatanın dağılımı sunulmuştur. Hata, gerçekleşen elektrik tüketiminden tüketim tahmini çıkarılarak hesaplanmıştır. Bu nedenle, negatif hata, modelin gerçekleşen elektrik tüketimini olduğundan az tahmin ettiği anlamına gelirken, pozitif hata, fazla tahmin anlamına gelmektedir. Hata dağılım grafiği incelendiğinde, en iyi modelin genellikle gerçekten değerden daha yüksek tahmin değeri elde ettiği görülmektedir. Şekil 5.14'te ise gerçekleştirilen tüm eğitimlerin sonunda elde edilen MAPE değerlerinin dağılımı sunulmuştur. MAPE dağılım grafiği incelendiğinde, literatürde gerçekleştirilen çalışmaların ortalamasında daha iyi tahmin performansı elde edildiği tespit edilmiştir.



Şekil 5.14. En iyi model için hata dağılım grafiği



Şekil 5.15. Gerçekleştirilen tüm eğitimler için MAPE dağılım grafiği

6. SONUÇ VE ÖNERİLER

Gerçekleştirilen bu tez çalışmasında, 01.01.2010 tarihi saat 00:00'dan 31.12.2020 tarihi saat 23:00'a kadar saatlik olarak örneklenmiş ve Türkiye'nin verilerinden oluşan veri seti ile kısa vadeli zaman serisi tahmini gerçekleştirilmiştir. Ayrıca, zaman serisi problemini çözmek amacıyla kullanılan derin öğrenme modelleri üzerine, deneysel bir araştırma gerçekleştirilmiştir. Söz konusu problem için literatürdeki en yaygın yaklaşımın, tekrarlayan sinir ağları olduğu gözlemlenmiştir. Bununla birlikte, zamansal evrişimli ağların gelişmesiyle, evrişimli ağların kullanımı da giderek arttığı tespit edilmiştir.

Tez çalışmasında, sekiz popüler derin öğrenme mimarisi kullanarak kapsamlı bir deneysel çalışma yürütülmüştür. Bunlar; çok katmanlı algılayıcı (MLP), elman tekrarlayan sinir ağı (ERNN), uzun kısa süreli bellek (LSTM), yankı durumu ağı (ESN), kapılı tekrarlayan birimler (GRU), kodlayıcı-kod Çözücü istiflenmiş uzun kısa süreli bellek (Encoder-Decoder Stacked LSTM), evrişimli sinir ağı (CNN) ve zamansal evrişimli sinir ağı'dır (TCN).

Kullanılan modellerin performansı, 7 farklı öznitelik kombinasyonu ve çeşitli konfigürasyon ve eğitim hiperparametrelerinden oluşan birçok model eğitilerek ölçülmüştür. Modellerin performansı ise tahmin doğruluğu ve verimlilik açısından değerlendirilip, çeşitli ölçüm metrikleri ile kapsamlı bir istatistiksel analiz gerçekleştirilmiştir. Ayrıca, Tablo 5.3.'te gösterildiği üzere, tez çalışmasında önerilen ağırlıklandırılmış toplam hissedilen sıcaklık özniteliğinin, tahmin performansı üzerinde artırıcı yönde etkisinin olduğu gözlemlenmiştir. Bu deneysel çalışmadan elde edilen sonuçlar aşağıda özetlenmiştir:

- Tüm modeller doğru parametrelendiğinde doğru tahminler elde etmektedir. Ancak, modellerin sonuçlarındaki farklılıklar, uygun bir mimari konfigürasyon bulmanın model ve tahmin performansı açısından önemini göstermektedir.
- Gizli katmanların derinliğinden bağımsız olarak çoğu modelde, MLP ağları zaman serisi verilerini zamansal sırasını modelleyemediği için düşük tahmin performansı göstermiştir.
- En iyi performansı LSTM ve onu yakından takip eden GRU elde etmiştir.

- CNN'ler hıza göre en iyi doğruluk değerini elde etmiştir. Bu da onları gerçek zamanlı uygulamalar için tekrarlayan yaklaşımlardan daha uygun hale getirmektedir.
- LSTM ağları, GRU mimarisinin aksine daha az sayıda yığılmış katmanla daha iyi sonuçlar elde etmektedir.
- CNN'ler için en iyi tahminlerin maksimum gruplama (max-pooling) içermeyen modellerden elde edildiği ve TCN'lerde ise tek katmanlı modellerden elde edildiği tespit edilmiştir.
- Batch size, öğrenme oranı ve optimize edici gibi hiperparametrelerin tahmin doğruluğu ve eğitim süresi üzerinde etkisinin olduğu görülmüştür.
- Model eğitilirken kullanılan geçmiş girdi ve tahmin ufku parametrelerinin, model performansı üzerinde etkisinin olduğu tespit edilmiştir.
- Hem modellemenin hesaplama maliyetini azaltmak hem de bazı durumlarda modelin performansını iyileştirmek için öznitelik seçiminin önemli bir etken olduğu değerlendirilmiştir.
- Takvimsel ve ekonomik öznitelikler, tahmin doğruluğunu pozitif yönde etkilemektedir.
- Ülke bazında elektrik enerjisi tüketim tahmini için varyasyon daha düşük olması sebebiyle lokala göre modellenmesinin daha kolay olduğu değerlendirilmektedir.
- Hareketli pencere prosedürü ile test işlemi daha verimli hale gelmektedir.

Farklı kombinasyonlarla gerçekleştirilen deneyler, hiperparametre seçimi konusunda aydınlatıcı olmasına rağmen, daha detaylı deneyler gerçekleştirebilmek için yüksek güçlü bilgisayarlara ve zamana ihtiyaç duyulmaktadır. Tez çalışmasında gerçekleştirilen deneylerin bazıları, bu eksiklikler yüzünden tam verimlilikte gerçekleştirilememiştir. Bu çalışmada ve literatürde gerçekleştirilmeyen parametre kombinasyonları, gelecek çalışmalar için bir odak noktası olabilmektedir. Ayrıca gelecekteki çalışmalar için, farklı veri setleri, ön işleme teknikleri gibi yaklaşımları kullanarak tahmin doğruluğunu arttırmaya imkân tanınmaktadır. Yöntem olarak yapay sinir ağı teknikleri kullanılmadan önce ARIMA ile modelleme işleminin gerçekleştirilmesi önerilmektedir.

KAYNAKLAR

- [1] Türkiye İstatistik Kurumu (TÜİK), Nüfus Projeksiyonları, 2018-2080. (Erişim Tarihi: May 10, 2021). 2018, <https://data.tuik.gov.tr/Bulten/Index?p=Nufus-Projeksiyonlari-2018-2080-30567>
- [2] Ogulata, R. T. Energy planning for Turkey. *Energy sources*. 2004, 26(3), 323-335. <https://doi.org/10.1080/00908310490256635>
- [3] T.C. Enerji Piyasası Düzenleme Kurumu (EPDK). 2020 Yılı Elektrik Piyasası Gelişim Raporu. (Erişim Tarihi: Haz 18, 2021). 2020, <http://epdk.gov.tr/Detay/DownloadDocument?id=7CLIQ2xB5V0=>
- [4] Altınay, G., Karagöl, E. Electricity consumption and economic growth: Evidence from Turkey. *Energy economics*. 2005, 27(6), 849-856. <https://doi.org/10.1016/j.eneco.2005.07.002>
- [5] Tutu, B. Türkiye için kısa vadeli elektrik enerjisi talep tahmini. Tobb Ekonomi Ve Teknoloji Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Ankara, 2017, 109 s. (Yüksek Lisans Tezi).
- [6] Dilaver, Z., Hunt, L. C. Industrial electricity demand for Turkey: a structural time series analysis. *Energy Economics*. 2011, 33(3), 426-436. <https://doi.org/10.1016/j.eneco.2010.10.001>
- [7] Enerji Piyasaları İşletme A.Ş. (EPİAŞ). Gerçek Zamanlı Tüketim. (Erişim Tarihi: Tem 19, 2021). 2020, <https://seffaflik.epias.com.tr/transparency/tuketim/gerceklesen-tuketim/gercek-zamanli-tuketim.xhtml>
- [8] Türkiye İstatistik Kurumu (TÜİK). İstatistik Veri Portalı. (Erişim Tarihi: Tem 22, 2021). 2020, <https://data.tuik.gov.tr/Search/Search?text=ekonomi>
- [9] Türkiye İstatistik Kurumu (TÜİK). İstatistik Veri Portalı. (Erişim Tarihi: Tem 22, 2021). 2020, <https://data.tuik.gov.tr/Search/Search?text=n%C3%BCfus>
- [10] Akgöl, S. Yapay Sinir Ağları İle Türkiye Elektrik Enerjisi Talep Tahmini Ve Diğer Yöntemlerle Karşılaştırmalı Analizi. Atatürk Üniversitesi, Sosyal Bilimler Enstitüsü, İktisat Anabilim Dalı, Erzurum, 2013, 218 s. (Doktora Tezi).

- [11] Lotufo, A. D. P., Minussi, C. R. Electric power systems load forecasting: A survey. PowerTech Budapest 99. Abstract Records. 1999, (Cat. No.99EX376), 36. DOI: 10.1109/PTC.1999.826467
- [12] Tokgöz, A., Ünal, G. A RNN based time series approach for forecasting turkish electricity load. 2018 26th Signal Processing and Communications Applications Conference (SIU). 2018, 1-4. DOI: 10.1109/SIU.2018.8404313
- [13] Hernandez, L., Baladron, C., Aguiar, J. M., Carro, B., Sanchez-Esguevillas, A. J., Lloret, J., Massana, J. A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings. IEEE Communications Surveys & Tutorials. 2014, 16(3), 1460-1495. DOI:10.1109/SURV.2014.032014.00094
- [14] Zhang, G. P. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing. 2003, 50, 159-175. [https://doi.org/10.1016/S0925-2312\(01\)00702-0](https://doi.org/10.1016/S0925-2312(01)00702-0)
- [15] Schmidhuber, J. Deep learning in neural networks: An overview. Neural networks. 2015, 61, 85-117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- [16] Gamboa, J. C. B. Deep learning for time-series analysis. arXiv preprint arXiv:1701.01887. 2017. <https://doi.org/10.48550/arXiv.1701.01887>
- [17] Deb, C., Zhang, F., Yang, J., Lee, S. E., Shah, K. W. A review on time series forecasting techniques for building energy consumption. Renewable and Sustainable Energy Reviews. 2017, 74, 902-924. <https://doi.org/10.1016/j.rser.2017.02.085>
- [18] Enerji Piyasaları İşletme A.Ş. (EPİAŞ). Şeffaflık Platformu. (Erişim Tarihi: Tem 23, 2021). 2020, <https://seffaflik.epias.com.tr/transparency/>
- [19] Tokgöz, A. Recurrent neural network based approaches for electricity consumption forecasting. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, İstanbul, 2018, 71 s. (Yüksek Lisans Tezi).
- [20] Topalli, A. K., Erkmen, I., Topalli, I. Intelligent short-term load forecasting in Turkey. International Journal of Electrical Power & Energy Systems. 2006, 28(7), 437-447. <https://doi.org/10.1016/j.ijepes.2006.02.004>

- [21] Erkmén, I., Özdoğan, A. Short term load forecasting using genetically optimized neural network cascaded with a modified Kohonen clustering process. In Proceedings of 12th IEEE International Symposium on Intelligent Control. IEEE. 1997, 107-112. DOI: 10.1109/ISIC.1997.626422
- [22] Filik, Ü. B., Gerek, Ö. N., Kurban, M. A novel modeling approach for hourly forecasting of long-term electric energy demand. *Energy Conversion and Management*. 2011, 52(1), 199-211.
<https://doi.org/10.1016/j.enconman.2010.06.059>
- [23] Çevik, H. H., Çunkaş, M. Short-term load forecasting using fuzzy logic and ANFIS. *Neural Computing and Applications*. 2015, 26(6), 1355-1367.
<https://doi.org/10.1007/s00521-014-1809-4>
- [24] Esener, İ. I., Yüksel, T., Kurban, M. Short-term load forecasting without meteorological data using AI-based structures. *Turkish Journal of Electrical Engineering and Computer Sciences*. 2015, 23(2), 370-380.
<https://doi.org/10.3906/elk-1209-28>
- [25] Yükseltan, E., Yucekaya, A., Bilge, A. H. Forecasting electricity demand for Turkey: Modeling periodic variations and demand segregation. *Applied Energy*. 2017, 193, 287-296.
<https://doi.org/10.1016/j.apenergy.2017.02.054>
- [26] Ishik, M. Y., Göze, T., Özcan, İ., Güngör, V. Ç., Aydın, Z. Short term electricity load forecasting: A case study of electric utility market in Turkey. In 2015 3rd International Istanbul Smart Grid Congress and Fair (ICSG). IEEE. 2015, 1-5. DOI: 10.1109/SGCF.2015.7354928
- [27] Tanidir, Ö., Tör, O. B. Accuracy of ANN based day-ahead load forecasting in Turkish power system: Degrading and improving factors. *Neural Network World*. 2015, 25(4), 443. DOI: 10.14311/NNW.2015.25.023
- [28] Permata, R. P., Prastyo, D. D. Hybrid dynamic harmonic regression with calendar variation for Turkey short-term electricity load forecasting. *Procedia Computer Science*. 2022, 197, 25-33.
<https://doi.org/10.1016/j.procs.2021.12.114>
- [29] Erdoğan, E. Electricity demand analysis using cointegration and ARIMA modelling: A case study of Turkey. *Energy policy*. 2007, 35(2), 1129-1146.
<https://doi.org/10.1016/j.enpol.2006.02.013>

- [30] Türkiye Elektrik İletim A.Ş. (TEİAŞ). Elektrik Üretim-Tüketim Raporları. (Erişim Tarihi: Ağu 8, 2021). 2020, <https://www.teias.gov.tr/turkiye-elektrik-uretim-iletim-istatistikleri>
- [31] Yumurtaci, Z., Asmaz, E. Electric energy demand of Turkey for the year 2050. *Energy Sources*. 2004, 26(12), 1157-1164.
<https://doi.org/10.1080/00908310490441520>
- [32] Kavaklioglu, K., Ceylan, H., Ozturk, H. K., Canyurt, O. E. Modeling and prediction of Turkey's electricity consumption using artificial neural networks. *Energy Conversion and Management*. 2009, 50(11), 2719-2727.
<https://doi.org/10.1016/j.enconman.2009.06.016>
- [33] Kavaklioglu, K. Modeling and prediction of Turkey's electricity consumption using Support Vector Regression. *Applied Energy*. 2011, 88(1), 368-375.
<https://doi.org/10.1016/j.apenergy.2010.07.021>
- [34] Bilgili, M., Sahin, B., Yasar, A., Simsek, E. Electric energy demands of Turkey in residential and industrial sectors. *Renewable and Sustainable Energy Reviews*. 2012, 16(1), 404-414.
<https://doi.org/10.1016/j.rser.2011.08.005>
- [35] Türkiye Elektrik Dağıtım A.Ş. (TEDAŞ). Sektör Raporları. (Erişim Tarihi: Ağu 20, 2021). 2020, https://www.tedas.gov.tr/#!/Sektor_rapor
- [36] Yetis, Y., Jamshidi, M. Forecasting of Turkey's electricity consumption using Artificial Neural Network. In 2014 World Automation Congress (WAC). 2014, 723-728.
DOI: 10.1109/WAC.2014.6936121
- [37] Oğcu, G., Demirel, O. F., Zaim, S. Forecasting electricity consumption with neural networks and support vector regression. *Procedia-Social and Behavioral Sciences*. 2012, 58, 1576-1585.
<https://doi.org/10.1016/j.sbspro.2012.09.1144>
- [38] Ünlü, K. D. A Data-Driven Model to Forecast Multi-Step Ahead Time Series of Turkish Daily Electricity Load. *Electronics*. 2022, 11(10), 1524.
<https://doi.org/10.3390/electronics11101524>
- [39] Muzaffar, S., Afshari, A. Short-term load forecasts using LSTM networks. *Energy Procedia*. 2019, 158, 2922-2927.
<https://doi.org/10.1016/j.egypro.2019.01.952>

- [40] Zhang, N., Li, Z., Zou, X., Quiring, S. M. Comparison of three short-term load forecast models in Southern California. *Energy*. 2019, 189, 116358.
<https://doi.org/10.1016/j.energy.2019.116358>
- [41] Guo, W., Che, L., Shahidehpour, M., Wan, X. Machine-Learning based methods in short-term load forecasting. *The Electricity Journal*. 2021, 34(1), 106884.
<https://doi.org/10.1016/j.tej.2020.106884>
- [42] Wu, L., Kong, C., Hao, X., Chen, W. A short-term load forecasting method based on GRU-CNN hybrid neural network model. *Mathematical Problems in Engineering*. 2020.
<https://doi.org/10.1155/2020/1428104>
- [43] Han, L., Peng, Y., Li, Y., Yong, B., Zhou, Q., Shu, L. Enhanced deep networks for short-term and medium-term load forecasting. *Ieee Access*, 7. 2018, 4045-4055.
 DOI: 10.1109/ACCESS.2018.2888978
- [44] Trierweiler Ribeiro, G., Guilherme Sauer, J., Fraccanabbia, N., Cocco Mariani, V., dos Santos Coelho, L. Bayesian optimized echo state network applied to short-term load forecasting. *Energies*. 2020, 13(9), 2390.
<https://doi.org/10.3390/en13092390>
- [45] Lara-Benítez, P., Carranza-García, M., Luna-Romera, J. M., Riquelme, J. C. Temporal convolutional networks applied to energy-related time series forecasting. *applied sciences*. 2020, 10(7), 2322.
<https://doi.org/10.3390/app10072322>
- [46] Eskandari, H., Imani, M., Moghaddam, M. P. Convolutional and recurrent neural network based model for short-term load forecasting. *Electric Power Systems Research*, 195. 2021, 107173.
<https://doi.org/10.1016/j.epsr.2021.107173>
- [47] Farsi, B., Amayri, M., Bouguila, N., Eicker, U. On short-term load forecasting using machine learning techniques and a novel parallel deep LSTM-CNN approach. *IEEE Access*, 9. 2021, 31191-31212.
 DOI: 10.1109/ACCESS.2021.3060290
- [48] Sadaei, H. J., e Silva, P. C. D. L., Guimarães, F. G., Lee, M. H. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy*. 2019, 175, 365-377.
<https://doi.org/10.1016/j.energy.2019.03.081>

- [49] Tian, C., Ma, J., Zhang, C., Zhan, P. A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network. *Energies*. 2018, 11(12), 3493.
<https://doi.org/10.3390/en11123493>
- [50] López, M., Sans, C., Valero, S., Senabre, C. Classification of special days in short-term load forecasting: the Spanish case study. *Energies*. 2019, 12(7), 1253.
<https://doi.org/10.3390/en12071253>
- [51] Lara-Benítez, P., Carranza-García, M., Riquelme, J. C. An experimental review on deep learning architectures for time series forecasting. *International Journal of Neural Systems*. 2021, 31(03), 2130001.
<https://doi.org/10.1142/S0129065721300011>
- [52] McCulloch, W. S., Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*. 1943, 5(4), 115-133.
<https://doi.org/10.1007/BF02478259>
- [53] Gardner, M. W., Dorling, S. R. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*. 1998, 32(14-15), 2627-2636.
[https://doi.org/10.1016/S1352-2310\(97\)00447-0](https://doi.org/10.1016/S1352-2310(97)00447-0)
- [54] Soh, H., Demiris, Y. Spatio-temporal learning with the online finite and infinite echo-state Gaussian processes. *IEEE transactions on neural networks and learning systems*. 2014, 26(3), 522-536.
 DOI: 10.1109/TNNLS.2014.2316291
- [55] Puig-Arnabat, M., Bruno, J. C. Artificial neural networks for thermochemical conversion of biomass. In *Recent advances in thermo-chemical conversion of biomass*. Elsevier, 2015, 133-156.
<https://doi.org/10.1016/B978-0-444-63289-0.00005-3>
- [56] Rumelhart, D. E., Hinton, G. E., Williams, R. J. Learning representations by back-propagating errors. *Nature*. 1986, 323(6088), 533-536.
<https://doi.org/10.1038/323533a0>
- [57] Elman, J. L. Finding structure in time. *Cognitive science*. 1990, 14(2), 179-211.
https://doi.org/10.1207/s15516709cog1402_1
- [58] Sutskever, I., Martens, J., Hinton, G. E. (2011, January). Generating text with recurrent neural networks. In *ICML*. 2011.

- [59] Mikolov, T., Joulin, A., Chopra, S., Mathieu, M., Ranzato, M. A. Learning longer memory in recurrent neural networks. 2014, arXiv preprint arXiv:1412.7753.
<https://doi.org/10.48550/arXiv.1412.7753>
- [60] Sharkawy, A. N. Principle of neural network and its main types. Journal of Advances in Applied & Computational Mathematics. 2020, 7, 8-19.
<https://doi.org/10.15377/2409-5761.2020.07.2>
- [61] Du, K. L., Swamy, M. N. Neural networks and statistical learning. Springer Science & Business Media. 2013, 824 s.
- [62] Maass, W., Joshi, P., Sontag, E. D. Computational aspects of feedback in neural circuits. PLoS computational biology. 2007, 3(1), e165.
<https://doi.org/10.1371/journal.pcbi.0020165>
- [63] Zhao, X., Chumkamon, S., Duan, S., Rojas, J., Pan, J. Collaborative human-robot motion generation using LSTM-RNN. In 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids). IEEE. 2018, 1-9.
DOI: 10.1109/HUMANOIDS.2018.8625068
- [64] Pascanu, R., Mikolov, T., Bengio, Y. On the difficulty of training recurrent neural networks. In International conference on machine learning. 2013, 1310-1318.
<https://doi.org/10.48550/arXiv.1211.5063>
- [65] Bengio, Y., Simard, P., Frasconi, P. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks. 1994, 5(2), 157-166.
DOI: 10.1109/72.279181
- [66] Chiang, Y. M., Chang, L. C., Chang, F. J. Comparison of static-feedforward and dynamic-feedback neural networks for rainfall-runoff modeling. Journal of hydrology. 2004, 290(3-4), 297-311.
<https://doi.org/10.1016/j.jhydrol.2003.12.033>
- [67] Lippmann, R. Book Review:" Neural Networks, A Comprehensive Foundation", by Simon Haykin. International Journal of Neural Systems. 1994, 5(04), 363-364.
<https://doi.org/10.1142/S0129065794000372>
- [68] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S. Recurrent neural network based language model. In Interspeech. 2010, 2(3), 1045-1048.
- [69] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaee, S. Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078. 2017.
<https://doi.org/10.48550/arXiv.1801.01078>

- [70] Sutskever, I., Martens, J., Hinton, G. E. Generating text with recurrent neural networks. In ICML. 2011.
- [71] Ab Aziz, M. F., Mostafa, S. A., Foozy, C. F. M, Mohammed, M. A., Elhoseny, M., Abualkishik, A. Z. Integrating Elman recurrent neural network with particle swarm optimization algorithms for an improved hybrid training of multidisciplinary datasets. *Expert Systems with Applications*. 2021, 183, 115441.
<https://doi.org/10.1016/j.eswa.2021.115441>
- [72] Linkens, D. A., Nyongesa, H. O. Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic algorithm control applications. *IEE Proceedings- Control Theory and Applications*. 1996, 143(4), 367-386.
 DOI:10.1049/IP-CTA:19960392
- [73] Toha, S. F., Tokhi, M. O. MLP and Elman recurrent neural network modelling for the TRMS. In 2008 7th IEEE international conference on cybernetic intelligent systems. IEEE. 2008, 1-6.
 DOI: 10.1109/UKRICIS.2008.4798969
- [74] Hochreiter, S., Schmidhuber, J. Long short-term memory. *Neural computation*, 1997, 9(8), 1735-1780.
 DOI: 10.1162/neco.1997.9.8.1735
- [75] Gers, F. A., Schmidhuber, J., Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural computation*. 2000, 12(10), 2451-2471.
 DOI: 10.1162/089976600300015015
- [76] Wang, Y., Zhu, S., Li, C. Research on multistep time series prediction based on LSTM. In 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE). IEEE. 2019, 1155-1159.
 DOI: 10.1109/EITCE47263.2019.9095044
- [77] Akbari Asanjan, A., Yang, T., Hsu, K., Sorooshian, S., Lin, J., Peng, Q. Short-term precipitation forecast based on the PERSIANN system and LSTM recurrent neural networks. *Journal of Geophysical Research: Atmospheres*. 2018, 123(22), 12-543.
<https://doi.org/10.1029/2018JD028375>
- [78] Jaeger, H. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report. 2001, 148(34), 13.

- [79] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P. A. Deep learning for time series classification: a review. *Data mining and knowledge discovery*. 2019, 33(4), 917-963.
<https://doi.org/10.1007/s10618-019-00619-1>
- [80] Jaeger, H., Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*. 2004, 304(5667), 78-80.
 DOI:10.1126/science.1091277
- [81] Li, G., Li, B. J., Yu, X. G., Cheng, C. T. Echo state network with Bayesian regularization for forecasting short-term power production of small hydropower plants. *Energies*. 2015, 8(10), 12228-12241.
<https://doi.org/10.3390/en81012228>
- [82] Kudithipudi, D., Saleh, Q., Merkel, C., Thesing, J., Wysocki, B. Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing. *Frontiers in neuroscience*. 2016, 9, 502.
<https://doi.org/10.3389/fnins.2015.00502>
- [83] Yang, R., Singh, S. K., Tavakkoli, M., Amiri, N., Yang, Y., Karami, M. A., Rai, R. CNN-LSTM deep learning architecture for computer vision-based modal frequency detection. *Mechanical Systems and signal processing*. 2020, 144, 106885.
<https://doi.org/10.1016/j.ymssp.2020.106885>
- [84] Cabrera, D., Sancho, F., Cerrada, M., Sanchez, R. V., Tobar, F. Echo state network and variational autoencoder for efficient one-class learning on dynamical systems. *Journal of Intelligent & Fuzzy Systems*. 2018, 34(6), 3799-3809.
 DOI: 10.3233/JIFS-169552
- [85] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. 2014
<https://doi.org/10.48550/arXiv.1406.1078>
- [86] Ravanelli, M., Brakel, P., Omologo, M., Bengio, Y. Light gated recurrent units for speech recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence*. 2018, 2(2), 92-102.
 DOI: 10.1109/TETCI.2017.2762739

- [87] Lu, Q., Zhu, Z., Xu, F., Zhang, D., Wu, W., Guo, Q. Bi-gru sentiment classification for chinese based on grammar rules and bert. *International Journal of Computational Intelligence Systems*. 2020, 13(1), 538-548.
<https://doi.org/10.2991/ijcis.d.200423.001>
- [88] Chrisman, L. Learning recursive distributed representations for holistic computation. *Connection Science*. 1991, 3(4), 345-366.
<https://doi.org/10.1080/09540099108946592>
- [89] Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*. 2016.
<https://doi.org/10.48550/arXiv.1607.00148>
- [90] Ellis, M. J., Chinde, V. An encoder–decoder LSTM-based EMPC framework applied to a building HVAC system. *Chemical Engineering Research and Design*. 2020, 160, 508-520.
<https://doi.org/10.1016/j.cherd.2020.06.008>
- [91] Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*. 2021.
 DOI: 10.1109/TNNLS.2021.3084827
- [92] Chen, Y. Convolutional neural network for sentence classification. University of Waterloo. Kanada, 2015, 51 s. (Yüksek Lisans Tezi)
- [93] Borovykh, A., Bohte, S., Oosterlee, C. W. Dilated convolutional neural networks for time series forecasting. *Journal of Computational Finance*, Forthcoming. 2018.
- [94] Hubel, D. H., Wiesel, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*. 1968, 195(1), 215-243.
<https://doi.org/10.1113/jphysiol.1968.sp008455>
- [95] Fukushima, K., Miyake, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*. Springer. Berlin, 1982, 267-285
<https://doi.org/10.1007/BF00344251>
- [96] O'Shea, K., Nash, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*. 2015.
<https://doi.org/10.48550/arXiv.1511.08458>

- [97] Bai, S., Kolter, J. Z., Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271. 2018.
<https://doi.org/10.48550/arXiv.1803.01271>
- [98] Yu, F., Koltun, V. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122. 2015.
<https://doi.org/10.48550/arXiv.1511.07122>
- [99] Wan, R., Mei, S., Wang, J., Liu, M., Yang, F. Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting. Electronics. 2019, 8(8), 876.
<https://doi.org/10.3390/electronics8080876>
- [100] Lara-Benítez, P., Carranza-García, M., García-Gutiérrez, J., Riquelme, J. C. Asynchronous dual-pipeline deep learning framework for online data stream classification. Integrated Computer-Aided Engineering. 2020, 27(2), 101-119.
DOI: 10.3233/ICA-200617
- [101] The Weather Channel (An IBM Business). Saatlik Hava Durumu. (Erişim Tarihi: Eki 11, 2021).
<https://weather.com/>
- [102] Taieb, S. B., Bontempi, G., Atiya, A. F., Sorjamaa, A. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert systems with applications. 2012, 39(8), 7067-7083.
<https://doi.org/10.1016/j.eswa.2012.01.039>
- [103] Shi, J., Cheng, L. Financial crisis dynamic prediction based on sliding window technology and mahalanobis-taguchi system. In 2011 International Conference of Information Technology, Computer Engineering and Management Sciences. IEEE. 2011, 4, 65-68
DOI: 10.1109/ICM.2011.307
- [104] Kim, S., Kim, H. A new metric of absolute percentage error for intermittent demand forecasts. International Journal of Forecasting. 2016, 32(3), 669-679.
<https://doi.org/10.1016/j.ijforecast.2015.12.003>
- [105] Luo, C., Zhan, J., Xue, X., Wang, L., Ren, R., Yang, Q. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In International Conference on Artificial Neural Networks. Springer, Cham. 2018, 382-391.
https://doi.org/10.1007/978-3-030-01418-6_38

- [106] Flores, B. E. A pragmatic view of accuracy measurement in forecasting. *Omega*. 1986, 14(2), 93-98.
[https://doi.org/10.1016/0305-0483\(86\)90013-7](https://doi.org/10.1016/0305-0483(86)90013-7)
- [107] Patro, S., Sahu, K. K. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*. 2015.
<https://doi.org/10.48550/arXiv.1503.06462>
- [108] Wang, Y. A Generalized Framework for Machine Re-learning of Complex Process and Kinetic Models. University of Washington. Washington, 2019, 24 s. (Doktora Tezi)
- [109] Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*. 2018.
<https://doi.org/10.48550/arXiv.1803.09820>
- [110] Zhou, Y. A. Sentiment classification with deep neural networks. 2019. (Yüksek Lisans Tezi)
- [111] Bilgic, M., Girep, C. P., Aslanoglu, S. Y., Aydinalp-Koksal, M. Forecasting Turkey's short term hourly load with artificial neural networks. In *IEEE PES T&D 2010*. IEEE. 2010, 1-7.
DOI: 10.1109/TDC.2010.5484442
- [112] Ediger, V. Ş., Tatlıdil, H. Forecasting the primary energy demand in Turkey and analysis of cyclic patterns. *Energy Conversion and Management*. 2002, 43(4), 473-487.
[https://doi.org/10.1016/S0196-8904\(01\)00033-4](https://doi.org/10.1016/S0196-8904(01)00033-4)
- [113] Google Drive. Derin Yapay Sinir Ağları İle Elektrik Tüketim Tahmini Tez Çalışmasının Kaynak Kodları. (Erişim Tarihi: Ağu 23, 2021).
https://drive.google.com/file/d/1aGWYox9MGyjjwMrDpmUlD9ZivQpxbp7_/view?usp=sharing