

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

**DERİN ÖĞRENME VE İSTATİSTİKSEL MODELLEME
YÖNTEMLERİ İLE GÜNLÜK DOĞALGAZ TÜKETİMİNİN
TAHMİNİ VE KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

Ömer EVCİM

**Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM
SİSTEMLERİ MÜHENDİSLİĞİ**
Tez Danışmanı : Doç. Dr. İhsan Hakan SELVİ

Haziran 2019

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME VE İSTATİSTİKSEL MODELLEME
YÖNTEMLERİ İLE GÜNLÜK DOĞALGAZ TÜKETİMİNİN
TAHMİNİ VE KARŞILAŞTIRILMASI

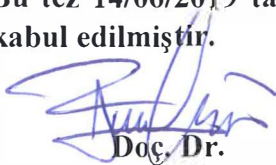
YÜKSEK LİSANS TEZİ

Ömer EVCİM

Enstitü Anabilim Dalı

BİLGİSAYAR VE BİLİŞİM
SİSTEMLERİ MÜHENDİSLİĞİ

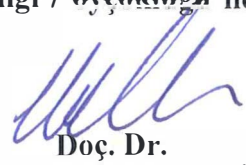
Bu tez 14/06/2019 tarihinde aşağıdaki jüri tarafından oybirliği / oylukla ile kabul edilmiştir.



Doç. Dr.
Fatih TAŞPINAR
Jüri Başkanı



Doç. Dr.
Numan ÇELEBİ
Üye



Doç. Dr.
İhsan Hakan SELVİ
Üye

BEYAN

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

Ömer EVCİM

14.06.2019

TEŐEKKÜR

Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Doç. Dr. İhsan Hakan SELVİ'ye, çalışmam boyunca gösterdikleri sabır ve anlayış için sevgili eşim Firdevs EVCİM'e, bu günlere gelmemde çok büyük emekleri ve fedakarlıkları olan annem ve babama teşekkürlerimi sunarım.

Ayrıca bu çalışmamda kullandığım verileri sağlayan Adapazarı Gaz Dağıtım Anonim Şirketine ve Meteoroloji Genel Müdürlüğü İstanbul Veri Merkezine teşekkür ederim.

İÇİNDEKİLER

TEŞEKKÜR.....	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ	v
ŞEKİLLER LİSTESİ	vi
TABLOLAR LİSTESİ.....	viii
ÖZET.....	ix
SUMMARY	x
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
KAYNAK ARAŞTIRMASI	4
BÖLÜM 3.	
TAHMİN YÖNTEMLERİ.....	11
3.1. Zaman Serisi Analizine Dayalı Modeller	11
3.1.1. Zaman serileri	11
3.1.2. Zaman serilerinin modellenmesi	12
3.1.2.1 Öz bağlantım (AR) modeli.....	12
3.1.2.2 Hareketli ortalamalar (MA) modeli	12
3.1.2.3 Öz bağlantımlı bütünleşik hareketli ortalama (ARIMA)	13
3.1.2.4 Mevsimsel öz bağlantımlı bütünleşik hareketli ortalama...	13
3.1.2.5 Model bileşenlerinin derecelerinin belirlenmesi	14
3.2. Yapay Sinir Ağları	14
3.2.1. Basit yapay sinir ağları	15

3.2.2. Tam bağılı derin yapay sinir ağları	15
3.2.3. Basit öz yineleyen yapay sinir ağları	16
3.2.4. Uzun-kısa dönem hafıza öz yineleyen yapay sinir ağları	16
3.2.5. Kapılı öz yineleyen yapay sinir düğümü ağları	17
3.2.6. Yapay sinir ağlarının temel sorunları	17
3.2.6.1 Ezberleme	17
3.2.6.2 Türevin çok azalması (Vanishing gradient)	18
3.2.6.3 Türevin çok artması (Exploiding gradient)	18
3.2.7. Aktivasyon fonksiyonları	19

BÖLÜM 4.

MATERYAL VE YÖNTEM	21
4.1. Materyal ve Araçlar	21
4.2. Verilerin Hazırlanması	22
4.2.1. Özellik Seçme Yöntemleri	23
4.2.1.1 RFE	23
4.2.1.2 LASSO	24
4.2.1.3 Uzman Görüşü	24
4.3. Dönemsel Özbağlanımlı Bütünleşik Hareketli Ortalama Modeli (SARIMA) ile Tahmin	25
4.4. Basit Yapay Sinir Ağı ile Tahmin	25
4.5. Tam Bağılı Derin Yapay Sinir Ağı ile Tahmin	27
4.6. Öz Yineleyen Yapay Sinir Ağları ile Tahmin	28

BÖLÜM 5.

ARAŞTIRMA BULGULARI	30
5.1. Öz Yineleyen Sinir Ağlarında Kullanılacak Veri Uzunluğunun Belirlenmesi	30
5.2. Elde Edilen Sonuçlar	31
5.2.1. SARIMA tahmin sonuçları	31
5.2.2. LASSO veri kümesi ile tahmin sonuçları	32
5.2.3. RFE veri kümesi ile tahmin sonuçları	36

5.2.4. Uzman görüşüne dayalı verilerle tahmin sonuçları	41
5.3. Eğitim Sayılarının Sonuca Etkisi	45
BÖLÜM 6.	
TARTIŞMA VE SONUÇ	46
KAYNAKLAR	48
EKLER	53
ÖZGEÇMİŞ	75

SİMGELER VE KISALTMALAR LİSTESİ

AIC	: Akaike bilgi kriteri
ARIMA	: Öz bağımlı bütünleşik hareketli ortalama
ARMA	: Öz bağımlı hareketli ortalama
ANFIS	: Adaptif sinirsel bulanık çıkarım sistemi
BIC	: Bayesyen bilgi kriteri
DNN	: Tam bağılı derin sinir ağı
ELU	: Ekspansiyel doğrusal birim
GRU	: Kapılı öz yineleyen öge
LASSO	: En Az Mutlak Büzülme Seçici Operatörü
LSTM	: Uzun-kısa dönem hafıza sinir ağı
MAPE	: Yüzde ortalama mutlak hata
MLP	: Çok katmanlı perceptron ağı
MLR	: Çoklu doğrusal bağlanım
YSA	: Yapay sinir ağı
SVR	: Destek vektör bağlanım
SVM	: Destek vektör makinesi
RNN	: Öz yineleyen sinir ağı
ReLU	: Düzeltilmiş doğrusal birim
RFE	: Tekrarlayan özellik eleme
SARIMA	: Mevsimsel öz bağımlı bütünleşik hareketli ortalama
SELU	: Ölçeklenmiş ekspansiyel doğrusal birim
TEİAŞ	: Türkiye elektrik iletim anonim şirketi

ŞEKİLLER LİSTESİ

Şekil 3.1. Basit YSA Modeli (Kırbaş, 2018)	15
Şekil 3.2. LSTM ve GRU düğümlerinin iç yapısı (Chung ve ark, 2014)	17
Şekil 3.3. Sık kullanılan aktivasyon fonksiyonları	20
Şekil 4.1. Çalışmada izlenen yol	21
Şekil 4.2. SARIMA ile yapılan tahmin adımları.....	25
Şekil 4.3. Doğalgaz tüketimindeki haftalık dalgalanma	26
Şekil 4.4. Basit YSA ve Tam Bağlı Derin YSA ile tahmin adımları	27
Şekil 4.5. Öz Yineleyen Sinir Ağları için hazırlanan veri yapısı	28
Şekil 4.6. Öz Yineleyen Sinir Ağları ile tahmin adımları	29
Şekil 5.1. SARIMA ile tahmin sonucu.....	31
Şekil 5.2. SARIMA ile tahmin performansı.....	31
Şekil 5.3. LASSO parametreleri ile YSA tahmin sonucu	32
Şekil 5.4. LASSO ile YSA tahmin performansı	32
Şekil 5.5. LASSO ile DNN tahmin sonucu.....	33
Şekil 5.6. LASSO ile DNN tahmin performansı.....	33
Şekil 5.7. LASSO ile SRNN tahmin sonucu.....	33
Şekil 5.8. LASSO ile SRNN tahmin performansı.....	34
Şekil 5.9. LASSO ile LSTM tahmin sonucu.....	34
Şekil 5.10. LASSO ile LSTM tahmin performansı.....	34
Şekil 5.11. LASSO ile GRU tahmin sonucu	35
Şekil 5.12. LASSO ile GRU tahmin performansı	35
Şekil 5.13. RFE ile YSA tahmin sonucu.....	36
Şekil 5.14. RFE ile YSA tahmin performansı.....	36
Şekil 5.15. RFE ile DNN tahmin sonucu	37
Şekil 5.16. RFE ile DNN tahmin performansı	37
Şekil 5.17. RFE ile SRNN tahmin sonucu	38

Şekil 5.18. RFE ile SRNN tahmin performansı	38
Şekil 5.19. RFE ile LSTM tahmin sonucu	39
Şekil 5.20. RFE ile LSTM tahmin performansı	39
Şekil 5.21. RFE ile GRU tahmin sonucu	40
Şekil 5.22. RFE ile GRU tahmin performansı	40
Şekil 5.23. Uzman görüşü ile YSA tahmin sonucu.....	41
Şekil 5.24. Uzman görüşü ile YSA tahmin performansı.....	41
Şekil 5.25. Uzman görüşü ile DNN tahmin sonucu	42
Şekil 5.26. Uzman görüşü ile DNN tahmin performansı	42
Şekil 5.27. Uzman görüşü ile SRNN tahmin sonucu	42
Şekil 5.28. Uzman görüşü ile SRNN tahmin performansı	43
Şekil 5.29. Uzman görüşü ile LSTM tahmin sonucu	43
Şekil 5.30. Uzman görüşü ile LSTM tahmin performansı	43
Şekil 5.31. Uzman görüşü ile GRU tahmin sonucu	44
Şekil 5.32. Uzman görüşü ile GRU tahmin performansı	44

TABLolar LİSTESİ

Tablo 4.1. Veri setindeki parametreler.....	22
Tablo 5.1. Veri zinciri uzunluğunun tahmin performansına etkisi	30
Tablo 5.2. Tutarlılık ve Hata Sonuçları (SARIMA)	31
Tablo 5.3. Tutarlılık ve Hata Sonuçları (LASSO YSA)	32
Tablo 5.4. Tutarlılık ve Hata Sonuçları (LASSO DNN).....	33
Tablo 5.5. Tutarlılık ve Hata Sonuçları (LASSO SRNN).....	34
Tablo 5.6. Tutarlılık ve Hata Sonuçları (LASSO LSTM).....	35
Tablo 5.7. Tutarlılık ve Hata Sonuçları (LASSO GRU).....	35
Tablo 5.8. Tutarlılık ve Hata Sonuçları (RFE YSA).....	36
Tablo 5.9. Tutarlılık ve Hata Sonuçları (RFE DNN).....	37
Tablo 5.10. Tutarlılık ve Hata Sonuçları (RFE SRNN).....	38
Tablo 5.11. Tutarlılık ve Hata Sonuçları (RFE LSTM).....	39
Tablo 5.12. Tutarlılık ve Hata Sonuçları (RFE GRU)	40
Tablo 5.13. Tutarlılık ve Hata Sonuçları (Uzman YSA)	41
Tablo 5.14. Tutarlılık ve Hata Sonuçları (Uzman DNN).....	42
Tablo 5.15. Tutarlılık ve Hata Sonuçları (Uzman SRNN).....	43
Tablo 5.16. Tutarlılık ve Hata Sonuçları (Uzman LSTM).....	44
Tablo 5.17. Tutarlılık ve Hata Sonuçları (Uzman GRU).....	44
Tablo 5.18. Eğitim sayılarının ve veri karıştırmanın sonuca etkisi.....	45

ÖZET

Anahtar kelimeler: İstatistiksel tahmin yöntemleri, yapay sinir ağları, derin sinir ağları, enerji tüketim tahmini

Bu çalışmada, Ocak 2014 – Haziran 2018 arası 54 aylık döneme ait Sakarya ilinde gerçekleşen doğalgaz tüketim verileri kullanılarak istatistiksel tahmin yöntemlerinden zaman serisinin dönemselsel otoregresif bütünleşik hareketli ortalama modeli ile birer günlük tahmini yapılmış, bu modelin başarısını basit yapay sinir ağları ve derin sinir ağlarının meteorolojik veriler de kullanılarak yapılan birer günlük tahminler ile başarısı karşılaştırılmıştır.

Mevsimsel Otoregresif Bütünleşik Hareketli Ortalama modelinin (SARIMA) 365 günlük tahmin başarısı ortalama mutlak hata oranı cinsinden (MAPE) yüzde 10,33 olmuştur. Yapay sinir ağlarının üç farklı parametre seçme metodu ile oluşturulan veri kümeleri kullanılarak eğitilmesi sonucu basit Yapay Sinir Ağı ile 6,68, Derin Yapay Sinir Ağı ile 7,23 MAPE değerine ulaşılmıştır. Söz konusu parametrelerin yer aldığı yedi gün geçmişe kadar veriler kullanılarak Basit Öz Yineleyen Yapay Sinir Ağı (SRNN) ile 7,61, Uzun-Kısa Dönem Hafıza Öz Yineleyen Sinir Ağları (LSTM) ile 6,45 ve Kapılı Öz Yineleyen Öge Sinir Ağları (GRU) ile ise 6,38 MAPE oranına ulaşılmıştır.

Araştırmada elde edilen bulgulara göre, uzman görüşü ile seçilen özellikler ile eğitilen kapılı öz yineleyen birim sinir ağı ile yapılan tahminlerin diğer yöntemlerden daha iyi sonuçlar verdiği görülmüştür.

ESTIMATION AND COMPARISON OF DAILY NATURAL GAS CONSUMPTION WITH DEEP LEARNING AND STATISTICAL MODELING METHODS

SUMMARY

Keywords: Forecasting, Energy Consumption, SARIMA, deep learning, RNN

In this study, forecasts made for the next day's consumption of natural gas by using natural gas consumption data in Sakarya province between January 2014 and June 2018. Forecasts were made by the periodical autoregressive integrated moving average model with using only daily natural gas consumption values. However for the forecasts were made by simple neural networks and deep neural networks, has been used both natural gas consumption values and the meteorological values of Sakarya province. And success rates of those forecasts were compared each other.

The 365-day forecast success of the Seasonal Autoregressive Integrated Moving Average model was 10.33 percent in terms of Mean Absolute Error Rate(MAPE) achieved. As a result of training of artificial neural networks by using data sets created with three different parameter selection methods, MAPE values of 6.68 with simple Artificial Neural Network and 7.23 with Deep Artificial Neural Network were reached. Using the data up to seven days in which these parameters are included, 7,61 with the Simple Recurrent Neural Network, 6,45 with the Long-Short-Term Memory Neural Networks and 6,38 with the Gated Recurrent Unit Neural Networks, Mean Absolute Percentage Error (MAPE) ratios has been reached.

According to the findings obtained in this research, it was seen that the predictions made with the gated self-repeating unit neural network trained with the characteristics chosen by the expert opinion gave better results than other methods.

BÖLÜM 1. GİRİŞ

İnsanođlu hayatını kolaylařtırmak ve devam ettirebilmek için enerjiyi her alanda kullanmıřtır. Bu nedenle enerji insanlık tarihi boyunca ve günümüzde en temel ihtiyaç olarak karřımıza çıkmaktadır. Dünya'nın güçlü ülkeleri arasına girebilme ve süper güç olabilme ile enerji kaynakları arasında doğrudan bir ilişki vardır. Ülkelerin sürdürülebilir kalkınma ve toplumsal refahı gerçekleřtirmeleri için artan enerji talebini karřılamaları ile mümkündür.

Çağımızın en büyük enerji kaynaklarından birisi olan ve gelişmiş ülkelerde çok daha önceleri kullanılmaya başlanan doğalgaz 1988 yılından itibaren ülkemizde de kullanılmaktadır. Tüm dünyada ve ülkemizde doğalgaz tüketimi gün geçtikçe artmakta bu nedenle ihtiyaç duyulan doğalgaz miktarının doğru olarak tahmin edilmesi, doğalgaz kullanıcılarının ihtiyaçlarının kesintisiz olarak sağlanması için önem arz etmektedir.

Tüketim tahminlerinin doğru bir şekilde yapılması, gelecekte yapılması gereken faaliyetlerin belirlenmesi ve planlanmasında önemli bir girdi olarak kullanılabilir. Kısa, orta ve uzun dönemli tahmin değerleri bilindiği takdirde üretim ve satınalma kararları doğru olarak verilebilecek böylece kıt olan kaynaklar daha ekonomik olarak kullanılabilir.

Tüm üretim süreçlerinin en önemli maliyet kalemlerinden birisi de enerji maliyetleridir. Enerji ihtiyacının tahmin edilebilmesi tedarik ve depolama maliyetlerini düşüreceği için hem enerji dağıtım şirketleri hem de tüketiciler açısından önem taşımaktadır (Yavuz, 2015). Ayrıca doğalgaz dağıtım şirketleriyle ana sağlayıcı şirketler arasındaki anlaşma hükümlerine göre doğalgaz firmaları hatalı tahminler

sebebiyle cezai müeyyidelerle karşı karşıya kalabilmektedir (Akpınar ve Yumuşak, 2017).

Tahmin metodları arasında, zaman serilerinde gelecek verilerin tahmininde kullanılan istatistiksel tahmin yöntemleri ve son zamanlarda popülerliği gittikçe artan yapay sinir ağları ile tahmin yöntemleri ön plana çıkmaktadır (Taşkın, 2018). Bu tahmin yöntemlerinin geliştirilmesi ve optimizasyon işlemleri tahmin sonuçlarının daha isabetli olması açısından önem arz etmektedir.

Bazı örneklerde tahmin işlemlerinde kullanılmak üzere eğitilen yapay sinir ağı gizli katman sayısının artırılmasının (derin sinir ağları) doğruluk oranını arttıracakı eskiden beri düşünülmektedir (Ahmad ve Chen, 2019). Daha önceleri bu daha çok düğüm ve katman sayısı içeren ağların eğitilmesi kabul edilebilir süreler içerisinde mümkün değilken, bilgisayar sistemlerinin gelişmesi ile artık bu mümkün hale gelmiştir. Ayrıca ekran kartı üreticilerinin geliştirdiği kütüphaneler ve sundukları donanım destekleri sayesinde hem eğitim süreleri önemli düzeylerde düşmüş hem de artan veri boyutlarıyla yapay sinir ağlarını eğitmek ve daha yetenekli hale getirmek mümkün olmuştur.

Bu tez çalışması kapsamında Adapazarı Gaz Dağıtım A.Ş.'den aldığımız Sakarya ili günlük doğalgaz tüketim verilerini, Meteoroloji Müdürlüğünden elde edilen sıcaklık, basınç, rüzgar vb. meteorolojik verilerle birleştirdikten sonra derin öğrenme temelli teknikler ve temel istatistiksel yöntemlerle karşılaştırılması yapılmıştır.

İkinci bölümde literatürde yer alan elektrik, doğalgaz vb. enerji kaynaklarının tüketim miktarlarının tahminine yönelik çalışmalar incelenmiştir.

Üçüncü bölümde bu tez çalışmasında kullanılmak üzere yöntemlerin yapıları açıklanmıştır.

Dördüncü bölümde elde edilen veriler üzerinde gerçekleştirilen işlemler detaylı olarak anlatılmıştır.

Beşinci bölümde ise dördüncü bölümde yapılan çalışmalar sonucunda elde edilen bulgular irdelenmiştir.

Altıncı bölümde ise farklı modellerin kullanılması ile elde edilen sonuçlar karşılaştırılarak yorumlanmıştır.

BÖLÜM 2. KAYNAK ARAŞTIRMASI

Bu bölümde enerji tüketim tahminini konu alan akademik çalışmalar kronolojik olarak verilmiştir.

Ankara ili elektrik yük tahminini konu alan çalışmada mevsimsel zaman serisi ayrıştırılmaya dayalı gri tahmin metodları uygulanarak aylık yük tahminleri yapılmış ve elde edilen sonuçlara dayalı olarak diğer metodlarla kıyaslanmıştır (Özkara, 2009). Türkiye Elektrik İletim Anonim Şirketi verilerine dayalı olarak gerçekleştirilen doktora çalışmasında ise öz bağlantım(regresyon) analizi ve yapay sinir ağı kullanılarak elektrik enerjisi talep tahmini yapılmış ve yapılan tahminler gerçek değerlerle ve birbirleri ile mukayese edilmiştir (Pınarbaşı, 2009).

Çin'in 1995 ile 2008 yılları arasındaki doğalgaz tüketim verilerinin kullanıldığı ve 2009-2015 yıllarında gerçekleşebilecek doğalgaz tüketim miktarlarının Polinomal Eğri ve Hareketli Ortalama Projeksiyonu modeli kullanılarak tahmin edildiği çalışma sonucunda 2015 yılında Çin'in doğalgaz tüketiminin 171600 milyon metre küpe ulaşacağı ön görülmüştür (Xu ve Wang., 2010).

Yamaçlı'nın yaptığı çalışmada ise zaman serisi analizi, çoklu doğrusal öz bağlantım analizi ve yapay sinir ağları kullanılarak Türkiye'nin uzun dönem elektrik talebi tahmin edilip bu değerler birbirleriyle mukayese edilmiştir. Elde edilen sonuçlara göre YSA'nın geçmiş veriler için yaptığı tahminler konusunda en iyi başarıyı yakaladığı görülmüştür (2010). Orta vadeli elektrik talebi tahmini konulu bir çalışmada mevcut kullanılan modellerin eksikliklerine değinildikten sonra adaptif sinirsel bulanık çıkarım sistemi (ANFIS) modeli kurulmuş ve bu modelle yapılan tahminler diğer öz bağlantım analizi metodları ile karşılaştırılarak çok başarılı sonuçlar verdiği görülmüştür (Ok, 2010). 1970 ve 2008 yılları arasındaki yıllık elektrik tüketim verileri

kullanılarak bir tahmin modeli oluşturmayı amaçlayan bir çalışmada elektrik tüketimini etkileyebileceği düşünülen parametreler (nelerdir) belirlenmiş ve bu parametreler kullanılarak çoklu regresyon analizi metodu ile bir model kurulmuştur (Genç, 2010).

Türk Telekom Anonim Şirketi'nin elektrik tüketiminin tahminini konu alan bir çalışmada ise şirketin İstanbul Anadolu yakasına ait enerji tüketim verileri ele alınmış ve MLP tipi bir yapay sinir ağı yardımıyla 120 aylık tahmin yapılmıştır. Çalışmada sıcaklık değerlerinin ve internet abone sayılarının enerji tüketimi üzerindeki etkisi vurgulanmıştır (Kaya, 2010).

Kocaeli ilinin elektrik yük tahminini konu alan bir çalışmada sıcaklık ve nüfus gibi bilgiler de dahil edilerek YSA modeli oluşturulmuş ve bu model ile yıllık elektrik yük tahminleri yapılmıştır (Özdemir, 2011). Elektrik enerjisi tahmini konulu bir diğer çalışmada ise tahmin için bir model oluşturulmuş ve oluşturulan modelle yapılan tahminler önceki çalışmalar ve TEİAŞ tarafından kullanılan yazılımın yaptığı tahminlerle kıyaslanmıştır (Oğurlu, 2011).

Çin'in Doğal gaz tüketim miktarındaki artışı tahmin etmeyi konu alan bir çalışmada sistem dinamikleri yaklaşımı kullanılmıştır (Li ve ark., 2011).

İzmir'de bulunan 148 binanın mimari projesi ve ısı raporları kullanılarak yapılan çalışmada 4 farklı sinir ağı ve bir bulanık mantık modeli oluşturulmuş, sonuçlar karşılaştırılmıştır. Çalışma sonucunda yapay sinir ağlarının tahmin başarıları vurgulanmıştır (Turhan, 2012).

2007-2011 yılları arasında Sakarya ilinde gerçekleşen doğalgaz tüketimi ile ilgili veriler kullanılan bir çalışmada Mevsimsel Öz Bağımlı Bütünleşik Hareketli Ortalamalar metodu, Radyal Temelli Fonksiyon Yapay Sinir Ağları ve Çok Katlı Perceptron Yapay Sinir Ağlarının Tahmin performansları mukayese edilmiştir (Taşpınar ve ark., 2013).

Kısa dönem elektrik talebini ve fiyat tahmini yapmayı hedefleyen bir çalışmada elektrik talebinin haftalık periyotlar şeklinde dalgalandığı, saatlik yükün sıcaklıkla bağlantısının tek bir bağıntıyla ifade edilemeyeceği, YSA'nın Destek Vektör Makinesinden(SVM) genellikle daha iyi sonuçlar verdiği sonuçlarına ulaşılmıştır (Yasin, 2013).

Yapılan bir yüksek lisans çalışmasında çoklu öz bağlantım ve öz bağlantımlı bütünleşik hareketli ortalamalar metodları yardımıyla evsel kullanıcılara ait doğalgaz tüketimi tahmini yapılmış ve kullanılan modellerin tahmin başarıları mukayese edilmiştir (Akpınar, 2014).

Çin'in enerji tüketimi ve üretimini tahmin etmeyi amaçlayan bir çalışmada yeni bilgi önceliği prensibine göre optimize edilmiş gri model ve başka 5 gri model oluşturulup bunlarla yapılan tahminlerin başarısı karşılaştırılmıştır (Xiong ve ark., 2014). Diğer bir kısa dönem elektrik tüketim talebi tahmini konulu çalışmada ise zaman serisi analizine (trend analizi) dayalı tahmin yöntemi, bulanık model ile tahmin yöntemi ve doğrusal öz bağlantım analizi metodlarının tahmin başarıları karşılaştırılmış ve zaman serisi analizi metodunun bu problem için kısa dönem tahmininde diğer metodlardan daha başarılı olduğu öne sürülmüştür (Yavuzdemir, 2014).

Türkiyenin aylık elektrik tahminini konu alan bir diğer çalışmada ise 2009-2013 yıllarına ait tüketim verileri kullanılarak zaman serileri analizi, öz bağlantım ve yapay sinir ağlar metodlarıyla yapılan tahminler sonucunda Çarpımsal üstel düzleştirme metodunun başarılı sonuçlar verdiği ifade edilmiştir (Aygördü, 2015). Singapurdaki endüstriyel binaların soğutmada kullandıkları enerji gözlemlendiği bir çalışmada Yapay sinir ağı ve Adaptif Neuro Bulanık Arayüz(ANFIS) kullanılarak tahmin yapılmıştır (Deb ve ark., 2015).

Ankara ve ilçelerindeki ev kullanıcılarının doğalgaz tüketimlerinin günlük tahminini konu alan bir çalışmada Çok Değişkenli Uyarlanabilir Öz Bağlantım Eğrileri (MARS) ve Konik Çok Değişkenli Uyarlanabilir Öz Bağlantım Eğrileri (CMARS) yöntemleri

kullanılmış ve bu alanda diğer yöntemlerle rekabet edebileceğine işaret edilmiştir (Yılmaz, 2015).

Yapılan bir çalışmada yapısal kalibre edilmiş destek vektörüne dayalı öz bağlantım (regresyon) metodu ile günlük doğalgaz tüketimi tahmini yapılmıştır (Bai ve Li, 2016). 12. Uluslararası Stratejik Yönetim Konferansında duyurulan bir çalışmada Öz Bağlanımlı Hareketli Ortalama Modeline Genetik Algoritmalar uygulanarak İstanbul ilinin doğalgaz tüketim tahminin klasik Öz Bağlanımlı Hareketli Ortalama Modeli ile yapılan tahminlerden daha başarılı olduğu sonucuna ulaşılmıştır (Ervural ve ark., 2016).

Başka bir çalışmada İran'ın enerji tüketimi tahmin için ARIMA-ANFIS hybrid modelinin 3 ayrı örüntüsü kullanılmış ve bu modellerin performansı karşılaştırılmıştır (Barak ve Sadegh, 2016). Bir çalışmada Hindistan demir üretimi sektöründe kullanılan enerji ve sera gazı emisyonu ile ilgili Öz Bağlanımlı Bütünleşik Hareketli Ortalamalar (ARIMA) metodu ile tahminlerde bulunulmuştur (Sen ve ark., 2016).

Türkiye'de gerçekleşen 2012-2014 yıllarına ait yük talep ve fiyat verileri kullanılarak saatlik yük talep ve fiyat tahminleri yapılmak üzere yapay sinir ağı modelleri geliştirilmiş ve elde edilen sonuçlar literatürde daha önce daha farklı yöntemlerle yapılmış olan çalışma sonuçları ile kıyaslanmıştır (Filiz, 2016).

Başka bir kısa vadeli elektrik enerjisi talep tahmini konulu çalışmada ise parametrelerin tahmin sonuçlarına etkisi incelenmiş, ve çalışma sonucunda en iyi sonuç veren parametreler tespit edilmiştir (Tutu, 2017). 2016 yılında yapılan bir çalışmada ise Adaptif Diferansiyel Gelişim algoritması uygulanmış bir geri yayılım yapay sinir ağı geliştirilmiş ve tahmin performansı değerlendirilmiştir (Zeng ve ark., 2017). Çinde elektrik tüketim değerlerinin tahmini üzerine yapılan çalışmada Çinin çeşitli bölgelerindeki 2016-2017 yıllarına ait aylık elektrik tüketimlerinin tahmini konusunda ekonomik durum ağı diferansiyel evrim (ESN-DE) algoritmasını geliştirmek üzerine yoğunlaşmış ve diğer tahmin yöntemleri ile geliştirdikleri algoritma mukayese edilmiştir (Wang ve ark., 2017). Londra'da bulunan bir idari

binanın elektrik tüketimi tahmini konulu bir çalışmada Çoklu Öz Bağlanım (Multiple Regression), Genetik Programlama, Yapay Sinir Ağı ve Derin Sinir Ağı gibi akıllı yöntemlerle tahmin edilmiş ve bu metodların tahmin başarıları karşılaştırılmıştır (Amber ve ark., 2017).

Çin'in doğalgaz talebinin tahminini konu alan bir çalışmada gri tahmin metodları kullanılmış ve doğrusal olmayan gri tahmin modeli optimize edilmeye çalışılmıştır. Bu çalışmada Çin'in 2020 yılında doğalgaz talebinin 315 milyon metre küpe ulaşacağı tahmin edilmiştir (Shaikh ve ark., 2017). Sakarya ilindeki doğalgaz tüketimini konu alan bir doktora çalışmasında ARIMA, Çok Katmanlı Perceptron(MLP) ve Yapay Arı Kolonisi(YSA-YAK) ağları oluşturulmuş ve günlük tahminde en iyi sonucu veren yapay arı kolonisi ile %14.9 MAPE değerine ulaşıldığı belirtilmiştir (Akpınar, 2017).

Çin'in 1978-2014 yılları arasındaki enerji tüketim verilerinin kullanıldığı bir çalışmada enerji tahmini için hibrit bir model geliştirilmiştir (Xiao ve ark., 2018). Gelişmekte olan ülkelerin elektrik tüketim miktarları ele alınan bir çalışmada Öz Bağlanımlı Bütünleşik Hareketli Ortalama Modeli ve Exponansiyel Düzleştirme Metodlarının tahmin performansları karşılaştırılmıştır (Meira ve Cyrino, 2018).

Yapılan başka bir çalışmada zaman serisi tahmin metodları kıyaslanmış ve gerçek hayatta nasıl kullanılması gerektiği ile ilgili önerilerde bulunulmuştur (Chou ve Tran, 2018). Enerji ihtiyacının Çinde sebep olduğu karbondioksit salınımının gelecekteki miktarının tahminini konu alan çalışmada Öz Bağlanımlı Dağıtılmış Gecikme sınır testi ve Granger nedensellik testine dayalı Vektör Hata Düzeltme Modeli kullanılarak 2020 ve 2030 yılları için Gayri Safi Yurt İçi Hasıladaki artışa da bağlı olarak Çin'in kullanacağı enerji kaynakları tahmin edilmiştir (Zhao ve Luo, 2018).

Çinde kentsel alanlardaki doğalgaz tüketim miktarlarının tahminini konu alan çalışmada Çin'in 30 bölgesinden alınan 2006-2015 yılları arasındaki veriler üzerinde genelleştirilmiş en küçük kareler metodu kullanılarak çalışılmış ve tüketim verileri incelenerek evsel doğalgaz tüketiminin yapısı ve gelecekteki durumu ile ilgili değerlendirmeler yapılmıştır (Liu ve ark., 2018). Ankara ilinin doğalgaz tüketiminin

yapay sinir ağı ile tahmin edilmesi konulu bir çalışmada ise sıcaklık, hissedilen sıcaklık ve nem değerleri kullanılarak tahminler yapılmış ve geliştirilen ağın performansı ölçülmüştür (Taşkiner, 2018). 1995 ile 2011 yılları arasına ait Çin'in doğalgaz tüketimi verisinin kullanıldığı çalışmada Boltzman modeli ve 3. derece polinomal eğri modeli kurulmuş, yapılan çalışmalar her iki modelin birleşimi ile oluşturulan modelin eğitimde başarılı uyum gösterdiği görülmüş ve bu modele göre 2012-2018 yılları arasında tüketimin yıllık ortalama %16,27 artacağı tahmin edilmiştir (Zhang ve Zhou, 2018).

Türkiye'nin saatlik enerji tüketimlerinin kullanıldığı bir çalışmada parametre olarak ülke geneli nüfus ağırlıklı sıcaklık verisi kullanılarak Öz yinelemeli sinir ağı kullanılarak (RNN, LSTM, GRU) başarılı tahminler yaptığı gözlemlenmiştir (Tokgöz, 2018). Kısa dönemli yük tahmini konulu bir çalışmada ise yapay sinir ağı ile oluşturulan modellerin kısa dönem elektrik yüklerinin tahmini uygulaması yapılmış ve literatürde yer alan diğer çalışmalardaki sürelerden daha kısa süreler için de yapay sinir ağlarının başarılı sonuçlar üretebildiği sonucuna ulaşılmıştır (Khan, 2018).

Türkiye'nin elektrik enerjisi ihtiyacını yapay sinir ağı ile tahmin etmek konulu çalışmada 2018-2023 yılları için elektrik enerjisi ihtiyacı tahmin edilmiştir (Durğun, 2018). Enerji tüketim talebi konulu diğer bir çalışmada ise destek vektör makinesi ve yapay sinir ağı modelleri kullanılmış ve elektrik tüketiminin meteorolojik veriler de kullanılarak % 97 doğrulukta tahmin edildiği ifade edilmiştir (Balsever, 2018).

Yapılan bir çalışmada doğalgaz tüketiminin tahmininde Lineer Regresyon, Kernel Makinesi ve Öz Yineleyen Sinir Ağlarını kullanılarak kentsel bir alanda (Slovenya'nın Ljubljana şehri) 60 saate kadar tahmin edilmiş ve sonuçlar karşılaştırılmıştır (Hribar ve ark., 2019). İstanbul ilinin doğal gaz tüketiminin tahminini konu alan bir çalışmada çoklu lineer öz bağlanım (MLR), yapay sinir ağı (YSA), ve destek vektör öz bağlanımı (SVR) gibi makine öğrenmesi metodları kullanılmış ve elde edilen sonuçlara göre SVR'nin doğalgaz tüketim tahmininde YSA'dan daha iyi sonuçlar verdiğini tespit etmişlerdir (Beyca ve ark., 2019).

Akıllı evlerin ürettikleri ve tükettikleri elektrik enerjisi miktarlarının tahminini konu alan bir çalışmada, Markov metodu ile tahmin yapılmış ve bu tahmin edilen değerlerle ilgili mukayeseler yapılmıştır (Gellert ve ark., 2019). Amerika Birleşik Devletlerinin New England bölgesindeki elektrik tüketim miktarlarını konu alan çalışmada elektrik tüketim miktarlarındaki saatlik değişimleri bir haftaya kadar tahmin eden yapay sinir ağları oluşturulmuştur (Ahmad ve Chen, 2019).

Çin'deki yenilenebilir enerji tüketim miktarlarının kullanıldığı çalışmada Kesir Birikimli Doğrusal Olmayan Bernoulli modeli geliştirilerek, diğer gri tahmin yöntemlerinin tahmin performansları ile mukayese edilmiştir (Wu ve ark., 2019).

Başka bir çalışmada ise Atina şehrinde bulunan Enerji düzenleme kurumu binasının enerji tüketiminin tahmini için en yakın meteoroloji istasyonundan veriler alınmıştır. Kurulan 3 farklı yapay sinir ağı modeli ile binanın enerji tüketimi tahmin edilmeye çalışılmıştır (Katsatos ve ark., 2019).

Diğer bir çalışmada ise Katar elektrik tüketiminin gayri safi milli hasıla ve nüfus artışı ile bağlantılı olduğu bir model kurulmuş, kurulan bu öz bağıntılı dağıtık gecikme modeliyle uzun vadede Katar elektrik tüketiminin farklı senaryolarda nasıl olacağı tahmin edilmiştir. Bu tahminler ışığında elektrik verimliliğinin geliştirilmesinin bu tüketim artışını kontrol etmek için ne kadar önemli olduğu gösterilmiştir (Khalifa ve ark., 2019).

Söz konusu çalışmalar incelendiğinde çalışmaların genellikle elektrik yükü tahmini konusunda yoğunlaştığı görülmektedir. Doğalgaz tüketim talebi ile ilgili çalışmaların kısıtlı olduğu, yapılan çalışmaların ise genellikle hazır programlarla yapıldığı ve son zamanlarda duyurulan kütüphanelerle (Tensorflow, Theno, CNTK v.b.) yapılan çalışmaların eksikliği görülmüştür.

BÖLÜM 3. TAHMİN YÖNTEMLERİ

Bu çalışmada Mevsimsel Öz Bağımlı Bütünleşik Hareketli Ortalamalar (SARIMA) ve yapay sinir ağı türlerinden Basit Yapay Sinir Ağı (3 katmanlı), Derin Tam Bağlı Yapay Sinir Ağı ve Öz Yineleyen Yapay Sinir Ağı çeşitlerinden olan Basit Öz Yineleyen Yapay Sinir Ağı, Uzun-Kısa Dönem Hafıza Öz Yineleyen Yapay Sinir Ağı ve Kapılı Öz Yineleyen Öge Yapay Sinir Ağı kullanılmıştır. Takip eden bölümde bu yapılarla ilgili bilgiler verilecektir.

3.1. Zaman Serisi Analizine Dayalı Modeller

Bu çalışmada istatistiksel tahmin metodlarından zaman serisi analizine dayalı tahmin metodlarından birisi olan tek değişkenli Mevsimsel Öz bağımlı Bütünleşik Hareketli Ortalamalar modeli (SARIMA) kullanılmıştır. Takip eden bölümde zaman serileri, zaman serisi analizi modelleri hakkında bilgiler verilecektir.

3.1.1. Zaman serileri

Bir olayın yada durumun zaman içerisindeki değişimi veya gözlem değerlerinin oluşturduğu dizi olarak tanımlanabilen zaman serileri temel olarak 4 bileşene ayrılabilir:

- Trend:seri içerisinde gözlemlenen değerlerde zamana bağlı olarak meydana gelen genel bir artışı ifade eden bileşendir.
- Mevsimsel Dalgalanma: Seri içerisinde belirli dönemlerde meydana gelen periyodik değişimleri ifade eden bileşendir.
- Çevrimsel Bileşen: Seri içerisinde genel eğilim etrafında meydana gelen ve belirli bir periyod içermeyen dalgalanmaları ifade eder.

- Düzensiz Bileşen: Zaman serisi içerisinde meydana gelen ve belirli bir modelle tam olarak açıklanamayan değişimleri ifade eder (Duru, 2007).

3.1.2. Zaman serilerinin modellenmesi

Zaman serilerinin modellenmesi ve bu modellerle tahmin yapılabilmesi için geliştirilen yöntemlerde seri durağan hale getirilerek bu geliştirilen modelle tahmin yapılması amaçlanır. Eğer incelenen seri Öz Bağımlı (AR), Hareketli Ortalama (MA) veya Öz Bağımlı Hareketli Ortalama (ARMA) modellerinden birine uygunsa durağan doğrusal stokastik model ile modellenebilir bir seridir. Aksi durumda ise durağan olmayan doğrusal stokastik modellerle (ARIMA veya SARIMA) modellenmesi gerekir (Bekin, 2015).

3.1.2.1 Öz bağımlı (AR) modeli

Verilerin daha önceki belirli sayıda verilerle doğrusal ilişkisinin hata terimi ile birlikte gösterildiği modeldir.

$$Y_t = a_1 * Y_{t-1} + a_2 * Y_{t-2} + \dots + a_n * Y_t + e_t \quad (3.1)$$

Bu modelin derecesi de verinin kendisinden önceki kaç veri ile ilişkili olarak ifade edildiğini göstermektedir (Duru, 2007). AR(n) modelinin genel gösterimi (Denklem 3.1) formülü ile ifade edilebilir.

3.1.2.2 Hareketli ortalamalar (MA) modeli

Zaman serisinin herhangi bir adımında meydana gelen hatanın belirtilen derece kadar önceki hataların bir fonksiyonu olarak ifade edilebileceği durumlarda zaman serisinin hareketli ortalamalar bileşeni bulunduğu anlamına gelir.

$$Y_t = \epsilon_t + \epsilon_{t-1} * \sigma_1 + \epsilon_{t-2} * \sigma_2 + \dots + \epsilon_{t-q} * \sigma_q \quad (3.2)$$

Bir t zamanına ait MA(q) bileşeni (Denklem 3.2) formülünde ifade edildiği gibi gösterilebilir (Duru, 2007).

3.1.2.3 Öz bağımlı bütünleşik hareketli ortalama (ARIMA)

Eğer bir zaman serisi AR modeli, MA modeli veya bu ikisinin birleşimi (ARMA) modeli ile durağan hale getirilemiyorsa bu zaman serisinin durağan hale getirilmesi için;

$$W_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) - \dots - (Y_{t-(n-1)} - Y_{t-n}) \quad (3.3)$$

(Denklem 3.3)'de $d=n$ için genel formülü verilen fark alma işlemi yapılır. Bu işlemde d genellikle 2'den büyük seçilmez. n sayısı fark alma işleminin derecesini gösterir (Nou; Duru, 2007).

Zaman serilerinin AR, MA ve fark alma bileşenlerinin 3'ünün birden kullanılarak modellenmesine Öz Bağımlı Bütünleşik Hareketli Ortalama Modeli (ARIMA) denir. Bu model ARIMA(p,d,q) şeklinde gösterilir.

- p, mevsimsel olmayan AR bileşeninin derecesini
- d, mevsimsel olmayan fark bileşeninin derecesini
- q, mevsimsel olmayan MA bileşeninin derecesini

ifade eder. Ayrıca bu gösterimde p,d,q değerleri sıfır yazılarak AR, MA, ARMA ve rastsal yürüyüş modelleri de ifade edilebilir.

3.1.2.4 Mevsimsel öz bağımlı bütünleşik hareketli ortalama

Zaman serisinde mevsimsel dalgalanmalar mevcutsa o zaman seri ARIMA modeli ile modellenemez. Bu tip serilerin modellenmesi için ARIMA modeline mevsimsel bileşenlerin eklenmesi ile Mevsimsel Öz Bağımlı Bütünleşik Hareketli Ortalama

(SARIMA) modeli oluşturulur. Bu model $ARIMA(p,d,q)(P,D,Q,m)$ şeklinde ifade edilmektedir. Bu gösterimde ;

p, serinin AR bileşeninin derecesini

d, serinin fark bileşenini derecesini

q, serinin hareketli ortalama bileşenini derecesini

P, serinin mevsimsel AR bileşeninin derecesini

D, serinin mevsimsel fark bileşeninin derecesini

Q, serinin mevsimsel hareketli ortalama bileşeninin derecesini

m, serinin mevsimsel periyot bileşeninin derecesini

ifade etmektedir.

3.1.2.5 Model bileşenlerinin derecelerinin belirlenmesi

ARIMA modelinde yer alan bileşenlerin derecelerinin belirlenmesi için genellikle öz bağlanım fonksiyonu (ACF) ve kısmi öz bağlanım fonksiyonu(PACF) testleri uygulanmaktadır. Ancak bu testler çoğu zaman kesin sonuç vermediği için Akaike Bilgi Kriteri (AIC) ve Bayesian Bilgi Kriterine (BIC) başvurulması gerekmektedir (Bek, 2008).

Bu uygulamada kullandığımız autoarima kütüphanesi de bu katsayıları belirlemek için AIC ve BIC kriterlerinden yararlanmaktadır.

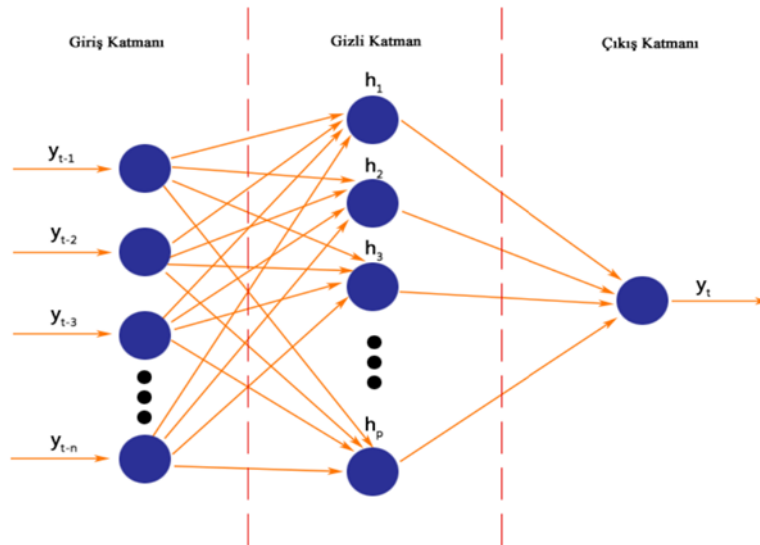
3.2. Yapay Sinir Ağları

Biyolojik sinir hücrelerinin çalışma prensibinin matematiksel olarak modellenmesi ve bu modelin bilgisayar sistemlerine uygulanması diye tanımlanabilen yapay sinir ağlarının temel özelliği insan beyninin öğrenme yeteneğini taklit etmesidir (Khan, 2018).

Devam eden bölümde bu çalışmada kullanılan yapay sinir ağları hakkında bilgiler verilecek, daha sonra ise yapay sinir ağları ile ilgili genel kavramlara değinilecektir.

3.2.1. Basit yapay sinir ağıları

Basit Yapay Sinir Ağları, çözülecek problem çok karmaşık değil ve uzun dönemli bağlantılar içermiyorsa işlem gücü açısından büyük avantaj sağlar. Literatürde Multi Layer Perceptron (MLP) olarak tanımlanan sinir ağıdır.



Şekil 3.1. Basit YSA Modeli (Kırbaş, 2018)

Şekil 3.1.'de görüldüğü gibi bir giriş katmanı bir gizli katman ve bir çıkış katmanından oluşan yapay sinir ağılarıdır.

3.2.2. Tam bağlı derin yapay sinir ağıları

Basit yapay sinir ağlarından farklı olarak çözülmesi gereken problem çok karmaşık ise gizli katman sayısının artırılması ağın kabiliyetini arttıracak düşünülebilir. Ancak bu ağ yapısında katman sayısının artmasıyla birlikte eğitim sürecinin uzaması gibi bir maliyete katlanmak ve türevin kaybolup gitmesi (vanishing gradient) gibi sorunlara çözümler geliştirmek gerekmektedir. Bu sorun ve çözüm yolları ilerleyen bölümlerde açıklanacaktır.

3.2.3. Basit öz yineleyen yapay sinir ağıları

Basit Öz Yineleyen Sinir Hücreleri önceki katmandan gelen girişin yanı sıra, kendi gizli durumu ve -katmanın ilk düğümü hariç- aynı katmanda bulunan bir önceki düğümün çıkışını girdi olarak kullanmaktadır.

$$h_t = g(W * x_t + U * h_{t-1}) \quad (3.4)$$

Bu gizli durum (Denklem 3.4)'deki formüle göre her bir $t > 0$ zamanında formülü ile ifade edebilecek bir şekilde güncellenir (Chung ve ark., 2014).

$g()$, aktivasyon fonksiyonu

W , ağırlıklar

U , gizli durumun ağırlığı

x_t , t zamanına ait veri

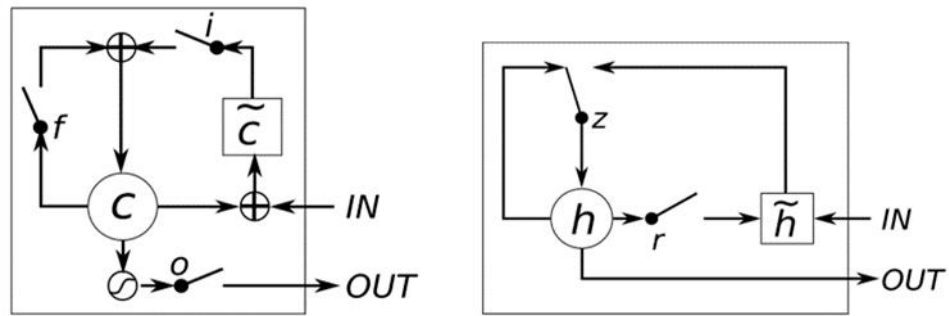
Bu ağ tipi mevcut veri ile önceki veri ile arasında kısa dönem bağlantı olduğu durumlarda kısa dönem hafıza yeteneği sayesinde başarılı sonuçlar verebilmektedir (Bengio, 1994).

3.2.4. Uzun-kısa dönem hafıza öz yineleyen yapay sinir ağıları

Bu öz yineleyen yapay sinir ağı çeşidi, basit sinir ağlarının algılayamadığı yada saklayamadığı bağımlılıkları algılama/saklama problemini çözmek için geliştirilmiştir. Örneğin bir cümleyi tamamlaması istenen basit öz yineleyen sinir ağı için pek çok seçenek vardır, uzun-kısa dönem hafıza öz yineleyen sinir ağıları böyle bir problemde önceki cümlede geçen bir durumun saklanıp, bu bağlantı kullanılarak mevcut seçeneklerin azaltılması ve daha isabetli tahminler yapılabilmesi gibi durumlarda daha başarılı sonuçlar verebilir diyebiliriz (Graves, 2013).

3.2.5. Kapılı öz yineleyen yapay sinir düğümü ağları

2014 yılında uzun-kısa dönem hafıza sinir ağlarının daha basit bir versiyonu olarak ortaya sürülmüştür (Cho ve ark., 2014). Bir çok problemde uzun-kısa dönem hafıza ağlarına benzer sonuçlar vermesi ve daha hızlı olması gibi sebeplerle tercih edilir olmuştur.



Şekil 3.2. LSTM ve GRU düğümlerinin iç yapısı (Chung ve ark, 2014)

Şekil 3.2.'de uzun-kısa dönem hafıza ağları ile kapılı öz yineleyen ağların iç yapısı görülmektedir.

3.2.6. Yapay sinir ağlarının temel sorunları

Bu bölümde yapay sinir ağlarının eğitimi ve tasarlanması sırasında karşılaşılan/üstesinden gelinmesi gereken bir takım problemler ve geliştirilen çözümlere değinilecektir.

3.2.6.1 Ezberleme

Yapay sinir ağlarının eğitim verisi ile olması gerekenden fazla eğitilmesi durumunda ağın daha önce karşılaşmadığı verilerle kötü sonuçlar vermesi durumudur (Bekin, 2015). Bu durum her problem ve veri setinde farklı eğitim sayılarında ortaya çıkmaktadır.

Bu sorunun üstesinden gelmek için veri setinin bir kısmının doğrulama için ayrılarak eğitim durumunun kontrol edilmesi ve ağı ezberleme safhasına geçtiği eğitim adımı sayısının tespit edilerek, eğitim sayısının bu sayının altında tutulması yada eğitimin farklı sayılarda adımda gerçekleştirilerek bu eğitilen ağların başarılarının karşılaştırılması gibi şekillerde eğitim sayısının optimum değerinin bulunması yoluyla çözülebilir (Srivastava ve ark., 2014). Ancak bu çözüm çok sayıda eğitimin tekrar tekrar yapılması ve çoğu zaman en optimum durumun yakalanamaması sorunları ile karşı karşıyadır.

Ezberleme sorununun diğer bir çözüm yolu ise ağa dropout katmanları eklenerek ve bu surette eğitim aşamasının her bir adımında belirli oranda sinir düğümü pasif yapılarak ağı ezberleme aşamasına geçmesinin engellenmesi yöntemidir. (Srivastava ve ark., 2014). Bu çözüm yönteminde ise dropout oranlarının belirlenmesi problemi ve eğitim sayılarının çok artması gibi sorunlar ile karşılaşmaktadır.

3.2.6.2 Türevin çok azalması (Vanishing gradient)

Bu problem ağ eğitilirken hatanın her katman için türev alınarak ağırlıklar güncellenmesi işleminde ilk katmanlara doğru ilerlendikçe türevin çok küçülmesi ve buna bağlı olarak ilk katmanların çok zor eğitilmesi ve eğitim sürecinin çok uzaması gibi sonuçlarla karşılaşılması durumudur.

Öz yineleyen ağlar bu problemden çok fazla etkilenmektedir. Bu sorunun çözümü için türevi her yerde aynı olan aktivasyon fonksiyonları veya LSTM ve GRU tipi öz yineleyen sinir ağları kullanılması tercih edilebilir (Tokgöz, 2018).

3.2.6.3 Türevin çok artması (Exploiding gradient)

Hataya bağlı olarak ağı güncellenmesi sırasında türevin aşırı artması ve buna bağlı olarak ağı eğitim esnasında kararsız bir hal sergilemesi durumudur. Bu durumda ağ eğitilemez ve doğru sonuçlar veremez.

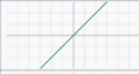
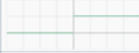







Bu problemi çözmek için türevin sınırlandırılması veya uzun-kısa dönem hafıza ağlarının kullanılması yöntemleri kullanılabilir.

3.2.7. Aktivasyon fonksiyonları

Sinir düğümüne gelen verilerin ağırlıklarla çarpımlarına karşılık sinir düğümünün hangi sonucu vereceğini belirleyen fonksiyona aktivasyon fonksiyonu denir (Özdemir, 2011). Yapay sinir ağlarında kullanılan aktivasyon fonksiyonları ağın öğrenme hızını ve başarısını etkileyen önemli bir faktördür. Her aktivasyon fonksiyonunun diğerlerine göre üstünlükleri varken, zayıflıkları/çözülmesi gereken problemleri de vardır. Bu sebeple aktivasyon fonksiyonlarının özelliklerini bilmek, yapay sinir ağlarından istenilen sonuçların alınabilmesi açısından önemlidir (Pedamonti, 2018).

Burada kullanılabilecek aktivasyon fonksiyonlarının özelliklerine ve zaafiyetlerine değinilecektir.

- Lineer (Birim) fonksiyon: doğrusal fonksiyonların bileşkesi yine bir doğrusal fonksiyon olacağı için bir katmandan fazla kullanılmasının bir anlamı yoktur.
- Sigmoid/Lojistik fonksiyon: derin sinir ağlarında kullanıldığında veya sinir düğümüne gelen verilerin çok büyük olması durumunda eğitim hızının çok düşmesi sorunu vardır.
- Tanjant hiperbolik fonksiyon: sigmoid fonksiyonundan farkı sigmoid fonksiyonundan daha büyük bir türeve sahip olması ve negatif sonuçlar verebilmesidir. Buna rağmen sigmoid fonksiyonu ile aynı problemlerle karşı karşıyadır.
- Rectified Linear Unit (ReLU): derin sinir ağlarında ve basit öz yineleyen sinir ağlarında türevin erimesi problemine çözüm olmasına rağmen negatif giriş değerlerinde öğrenme olmaması sebebiyle sinir hücrelerinin ölmesi problemine yol açabilmektedir. Bu problemi çözen ve daha iyi performans sergileyen Leaky ReLU, Parametrik ReLU, Exponansiyel Linear Unit (ELU) ve Ölçeklenmiş ELU (SELU) fonksiyonları geliştirilmiştir (Pedamonti, 2018).

Aktivasyon Fonksiyonu	Çizim	Formülü	Türevi
Birim (Lineer)		$f(x) = x$	$f'(x) = 1$
İkili Basamak		$f(x) = \begin{cases} 0 & x < 0 \text{ için} \\ 1 & x \geq 0 \text{ için} \end{cases}$	$f'(x) = \begin{cases} 0 & x \neq 0 \text{ için} \\ ? & x = 0 \text{ için} \end{cases}$
Sigmoid / Lojistik		$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'(x) = 1 - f(x)^2$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & x < 0 \text{ için} \\ x & x \geq 0 \text{ için} \end{cases}$	$f'(x) = \begin{cases} 0 & x < 0 \text{ için} \\ 1 & x \geq 0 \text{ için} \end{cases}$
Leaky ReLU		$f(x) = \begin{cases} 0.01x & x < 0 \text{ için} \\ x & x \geq 0 \text{ için} \end{cases}$	$f'(x) = \begin{cases} 0.01 & x < 0 \text{ için} \\ 1 & x \geq 0 \text{ için} \end{cases}$
Parametrik ReLU (PReLU)		$f(\alpha, x) = \begin{cases} \alpha x & x < 0 \text{ için} \\ x & x \geq 0 \text{ için} \end{cases}$	$f'(\alpha, x) = \begin{cases} \alpha & x < 0 \text{ için} \\ 1 & x \geq 0 \text{ için} \end{cases}$
Exponansiyel Lineer Unit (ELU)		$f(\alpha, x) = \begin{cases} \alpha(e^x - 1) & x \leq 0 \text{ için} \\ x & x > 0 \text{ için} \end{cases}$	$f'(\alpha, x) = \begin{cases} f(\alpha, x) + \alpha & x \leq 0 \text{ için} \\ 1 & x > 0 \text{ için} \end{cases}$
Scaled Exponansiyel Lineer Unit (SELU)		$f(\lambda, \alpha, x) = \lambda \begin{cases} \alpha(e^x - 1) & x \leq 0 \text{ için} \\ x & x > 0 \text{ için} \end{cases}$	$f'(\lambda, \alpha, x) = \begin{cases} f(\alpha, x) + \alpha \lambda & x \leq 0 \text{ için} \\ \lambda & x > 0 \text{ için} \end{cases}$

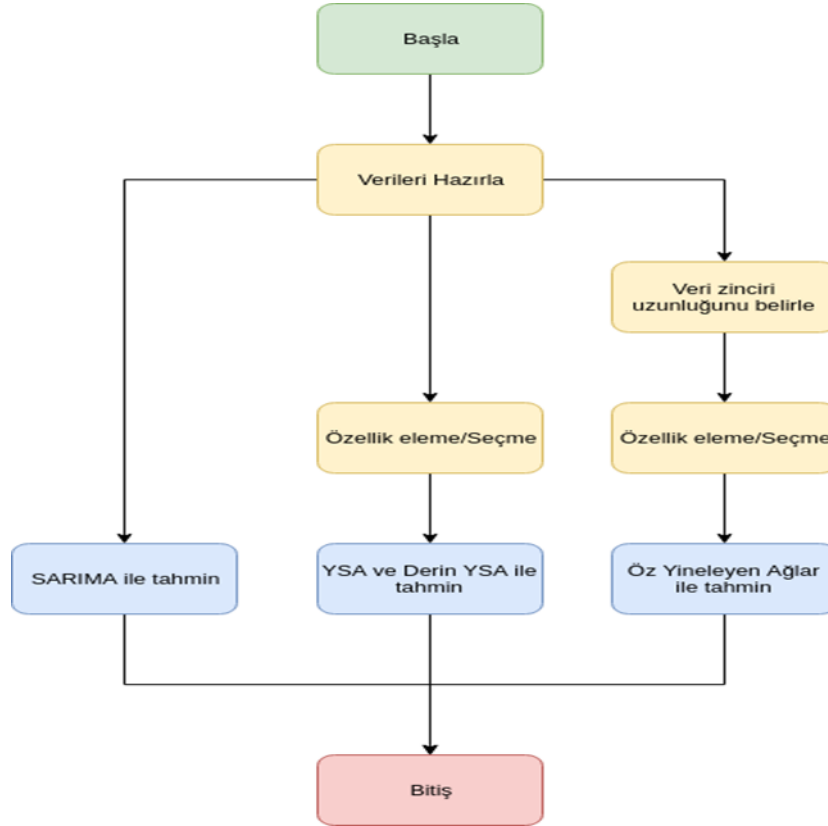
Şekil 3.3. Sık kullanılan aktivasyon fonksiyonları

Yapay sinir ağlarında sıkça kullanılan aktivasyon fonksiyonlarının tanımları, grafikleri ve türevleri Şekil 3.3.'de yer almaktadır.

BÖLÜM 4. MATERYAL VE YÖNTEM

4.1. Materyal ve Araçlar

Bu araştırmada kullanılmak üzere, 1 Ocak 2014 ve 30 Haziran 2018 tarihleri arasında Sakarya ilinde bulunan 1, 2 ve 3 tipi abonelere ait gaz çekiş verileri Adapazarı Gaz Dağıtım Anonim Şirketinden temin edilmiş ayrıca aynı döneme ait Sakarya ilinde bulunan meteoroloji istasyonları tarafından ölçülen saatlik sıcaklık, saatlik basınç, saatlik nisbi nem ve günlük ortalama rüzgar hızı ve yönü Meteoroloji Genel Müdürlüğünden temin edilmiştir.



Şekil 4.1. Çalışmada izlenen yol

Çalışmada uygulamalar Python dilinde yazılmış olup, Dönemsel Özbağlanımlı Bütünleşik Hareketli Ortalama Modeli (SARIMA) ile tahmin işleminde pmdarima kütüphanesi ve statmodels kütüphaneleri kullanılmış, yapay sinir ağları ile tahmin işlemi için ise Keras kütüphanesi ve Tensorflow kütüphaneleri kullanılmıştır. Verilerin hazırlanması için veriler SQLite veri tabanına aktarılmıştır. Ayrıca sonuçlar ve işlenen veriler hesap tablosuna yazılarak kullanılmıştır.

Çalışmanın genel yapısı ve izlenen yol Şekil 4.1.'de görülmekte olup bu adımlarda hangi işlemlerin yapıldığı ilerleyen bölümlerde açıklanacaktır.

4.2. Verilerin Hazırlanması

Çalışmada kullanılmak üzere veriler öncelikle SQLite veri tabanına atılmış, kullanılacak veriler Sakarya ili genelinde bulunan tüm istasyonlar tarafından ölçülmüş verilerin ortalamaları -ölçüm yapılmayan istasyonlar hesaba katılmadan- alınarak, eksik veri bırakılmaması sağlanmıştır. Ayrıca 2016 yılında şubat ayının 29'uncu gününe ait veriler silinmiştir. Yapay sinir ağları eğitilirken yılın günleri ve haftanın günleri de numaralandırılarak ağa giriş olarak verilmiştir. Verilerin işlendiği python kodu tezin ekler kısmında yer almaktadır.

Tablo 4.1. Veri setindeki parametreler

Parametre No	Parametre Adı
1	Yılın kaçınıcı günü
2	Haftanın kaçınıcı günü
3	Hava basıncı
4	Ortalama sıcaklık
5	Gün başlangıcı sıcaklık
6	En düşük sıcaklık
7	Bağıl Nem
8	Bulutluluk
9	Ortalama Rüzgar kuvveti
10	Rüzgar kuzey bileşen ortalama kuvveti
11	Rüzgar güney bileşen ortalama kuvveti
12	Rüzgar batı bileşen ortalama kuvveti
13	Rüzgar doğu bileşen ortalama kuvveti
14	Ay başlangıcı abone sayısı
15	Ay sonu abone sayısı
16	Gün Öncesi doğalgaz tüketim değeri

Bu araştırma sırasında Tablo 4.1.'de yer alan 16 özellikten oluşan giriş parametreleri elde edilmiştir. Bu özelliklerin azaltılması amacıyla veri setine parametre azaltma(özellik seçme) işlemi yapılmıştır.

4.2.1. Özellik Seçme Yöntemleri

Gerçek dünya problemlerinde ölçüm/gözlem hatası, ölçüm aletlerinin hassas olmaması, gibi nedenlerle veriler gürültülü olabilmektedir. Bu durum ölçülen değişkenlerle sonuçlar arasında iyi bir korelasyon bulunmasını engellemektedir. Bu sebeple aralarında sadece güçlü bir korelasyon bulunan verilerin seçilmesi için özellik seçme metodları geliştirilmiştir. Bu metodların kullanılması;

- Eğitim süresinin kısaltılması,
- Karmaşıklığın azaltılması,
- Doğruluk ve performans artışıyla birlikte “ezberleme” probleminin önüne geçilmesi

gibi avantajlar sağlamaktadır.

Bu çalışmada parametre seçme Öz Yineleyen Özellik Seçme (RFE) ve En Az Mutlak Büzülme Seçici Operatörü (LASSO) metodlarının Çapraz Doğrulama ile uygulanmış şekli olan RFECV ve LASSOCV metodaları kullanılmıştır.

4.2.1.1 RFE

Kendisine dışarıdan parametre olarak verilen tahmin metodunu kullanarak istenilen değere ulaşılan kadar tekrarlayan bir şekilde veri kümesinin alt kümelerinin seçilmesi şeklinde çalışan bir yöntemdir. İlk adımda verilen veri seti kullanılarak tahmin yöntemi eğitilip özelliklerin önem yada etkileşim dereceleri hesaplanır. Elde edilen sonuca göre en düşük dereceye sahip olan özellik veri setinden çıkartılır. Bu işlem tekrarlayan bir şekilde devam ettirilir.

Bu yöntem veri setinde bulunan haftanın günü, ortalama sıcaklık, ay sonu abone sayısı ve gün öncesi tüketim özelliklerinden oluşan 4 özellikten oluşan veri setinin kullanılmasını önermektedir.

4.2.1.2 LASSO

Lasso regresyonu matematiksel olarak;

(Denklem 4.1) denkleminde $\lambda=1$ olacak şekilde amaç fonksiyonunu minimum yapan çözüm olarak tanımlanabilir. Lasso regresyonunun çözümü aynı zamanda özellik seçimi probleminin de çözümüdür.

Bu yöntem yılın kaçınıcı günü, haftanın kaçınıcı günü, basınç, ortalama sıcaklık, en düşük sıcaklık, rüzgar kuvveti, rüzgar kuzey bileşen kuvveti, rüzgar güney bileşen kuvveti, rüzgar doğu bileşen kuvveti, ay sonu abone sayısı ve gün öncesi tüketim verilerinden oluşan 11 özelliğin kullanılması gerektiği sonucunu vermiştir.

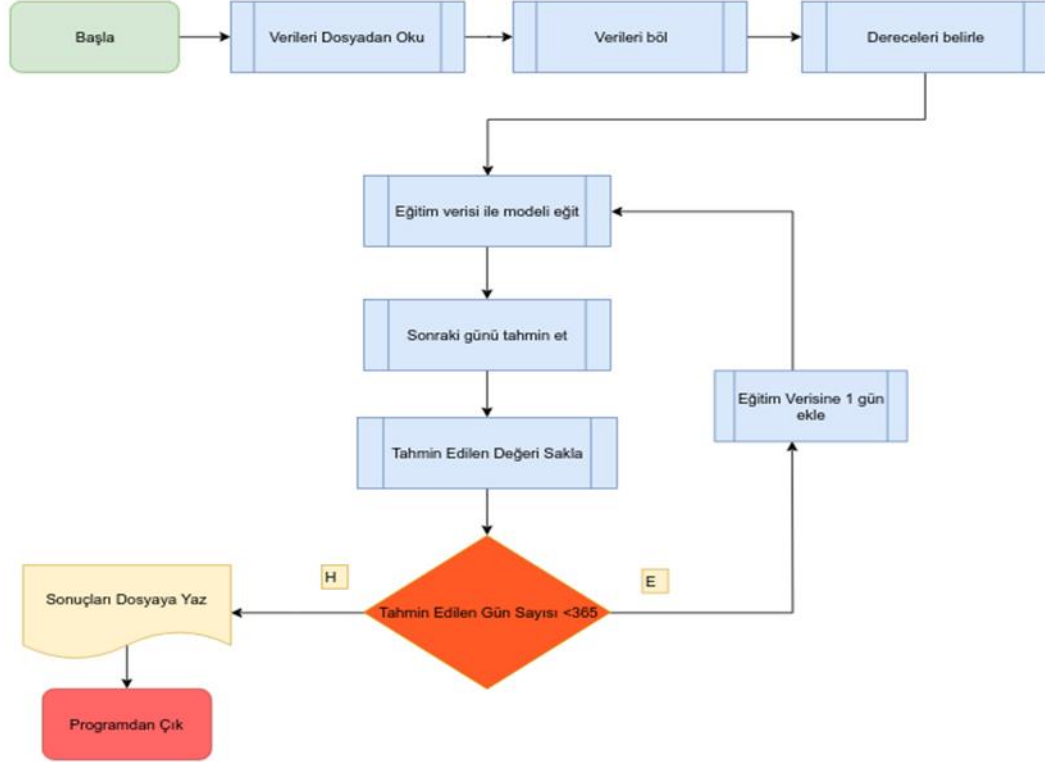
4.2.1.3 Uzman Görüşü

Bu özellik seçimi yönteminde ise literatürde yer alan çalışmalardan yararlanılarak doğalgaz tüketimi üzerinde etkisinin yüksek olduğu düşünülen özellikler seçilmiştir.

Bu yöntemle yılın kaçınıcı günü, haftanın kaçınıcı günü, günlük ortalama sıcaklık, bağıl nem ve gün öncesi dağalgaz tüketimi özelliklerinin kullanılması önerilmiştir.

Dönemsel Özbağlanımlı Bütünleşik Hareketli Ortalama Modeli (SARIMA) ile Tahmin

4.3. Dönemsel Özbağlanımlı Bütünleşik Hareketli Ortalama Modeli (SARIMA) ile Tahmin

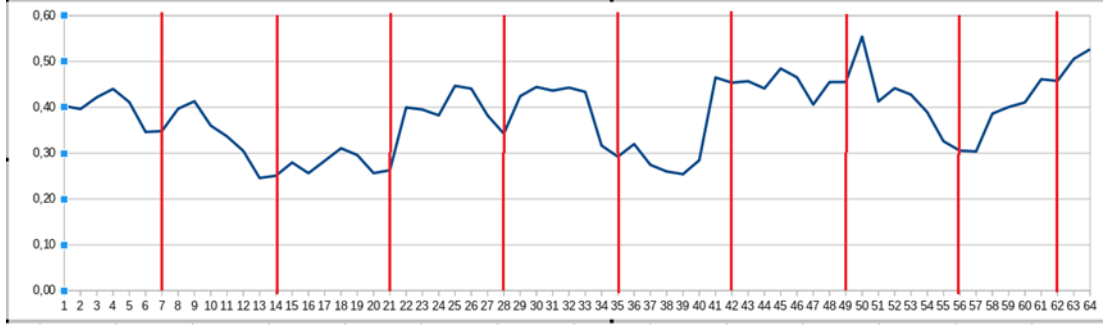


Şekil 4.2. SARIMA ile yapılan tahmin adımları

Doğalgaz tüketim verilerinin son 365 tanesi test için ayrıldıktan sonra pmdarima kütüphanesinin autoarima metoduna verilerek (pdq)(PDQm) (1, 1, 1)(0, 0, 1, 12) değerleri elde edilmiş ve bu değerlerle statmodels kütüphanesine ait SARIMAX metodunda iterasyon yapılarak her seferinde önceki günler verilerek suretiyle test için ayrılan 365 gün için tahmin işlemi yapılmıştır. Şekil 4.1.'de de görülen SARIMA ile tahmin adımlarına ait kodlar tezin ek bölümünde yer almaktadır.

4.4. Basit Yapay Sinir Ağı ile Tahmin

Yapay sinir ağı ile tahmin yapılırken ise tahmin yapılacak güne ait ölçülen meteorolojik değerler ve bir önceki güne ait doğalgaz tüketim verisi, o günün yılın kaçınıcı günü olduğu verisi kullanılmıştır.

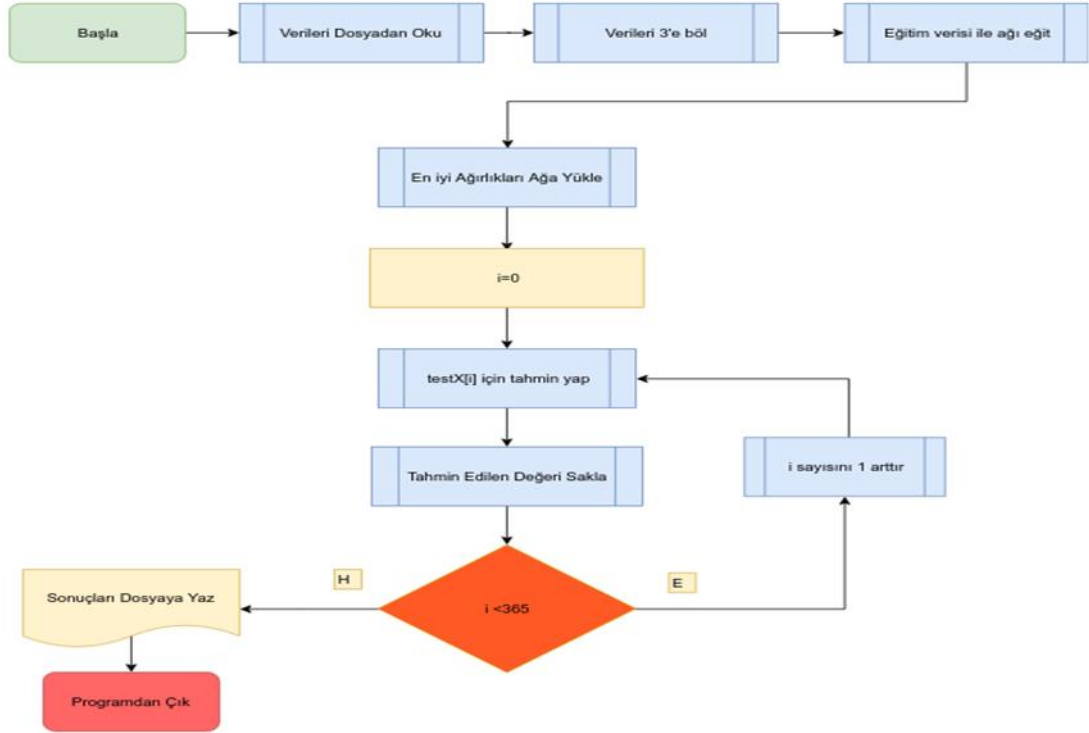


Şekil 4.3. Doğalgaz tüketimindeki haftalık dalgalanma

Ayrıca Şekil 4.2.'de de görüldüğü üzere haftalık olarak tüketim değerlerinde benzer dalgalanmalar olduğu düşünülerek haftanın kaçınıcı günü olduğuna dair bilgi Pazartesi günü 6 pazar günü 0 olacak şekilde sisteme verilmiştir.

Yapay sinir ağı eğitilirken verinin ilk 1100 günlük kısmı eğitim verisi olarak kullanılmış, takip eden 176 günlük veri doğrulama için kullanılırken kalan 365 günlük veri ise test verisi olarak ayrılmıştır. Üç katmanlı kurduğumuz modelde ilk katman girilen parametre sayısı kadar düğümden oluşurken, gizli -ara- katman da 30 düğümden oluşturulmuş ve çıkış katmanı ise tek sinir düğümünden ibarettir.

Eğitim işlemi sırasında Checkpoint kullanılarak o adıma kadar doğrulama verisi üzerinde en iyi başarıyı elde eden durumun ağırlıkları kaydedilmiştir. Daha sonraki eğitim adımlarında ağıın ezberleme (overfitting) durumuna bağlı olarak daha kötü sonuçlar alınmaya başlanması durumunda kaydedilen bu ağırlıklar ağı tahmin adımında yüklenerek kötü sonuçlardan sakınılmaya çalışılmıştır. Bu yapı sayesinde ağıın herhangi bir eğitim adımında o ana kadar elde edilen en iyi ağırlıklar ile tahmin yapılabilmektedir. Bu yöntem ayrıca bir optimizasyon gerektirmediği için tercih edilmiştir.



Şekil 4.4. Basit YSA ve Tam Bağlı Derin YSA ile tahmin adımları

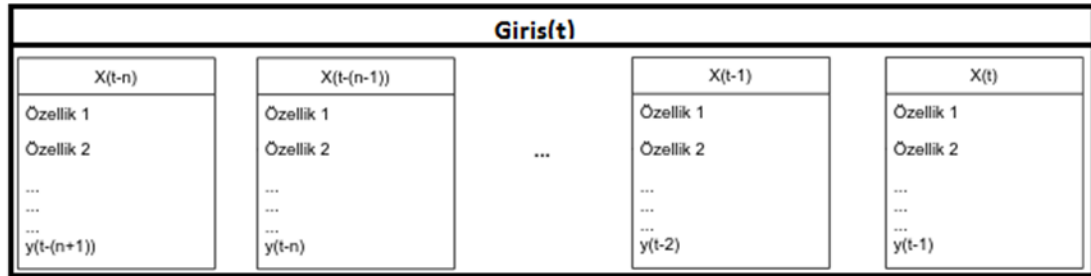
4.5. Tam Bağlı Derin Yapay Sinir Ağı ile Tahmin

Tam bağlı yapay sinir ağı da basit yapay sinir ağında olduğu gibi parametre sayısı kadar düğüm içeren giriş katmanı ve her katmanında 30 düğüm içeren 4 gizli katman ve 1 düğümünden oluşan bir çıkış katmanından oluşturulmuştur. Bu ağa veriler basit yapay sinir ağında olduğu gibi verilmiş ve benzer şekilde Checkpoint yapısı kullanılarak ezberleme durumu bertaraf edilmeye çalışılmıştır.

Şekil 4.3. ile Basit YSA ve Tam Bağlı Derin YSA'nın tahmin işlemlerine ait akış şeması gösterilmiştir. Basit sinir ağı ile Tam Bağlı Derin YSA'nın bir birlerine benzerlikler göstermesi sebebiyle ekte yer alan python kodu her iki ağ için de -ilgili satırlar yorum satırı haline getirilmek suretiyle kullanılmış, ekte de kullanıldığı şekilde yer verilmiştir.

4.6. Öz Yineleyen Yapay Sinir Ağları ile Tahmin

Öz yineleyen yapay sinir ağları verilen parametreleri işledikten sonra sonucu hem yanındaki düğüme hem de bir sonraki katmandaki düğümlere girdi olarak gönderdiği için ağları eğitirken verilerin karıştırılarak verilmesi özelliğinden yararlanmak için basit yapay sinir ağı ve tam bağlı yapay sinir ağından farklı olarak istenilen gün kadar öncesinin giriş parametrelerinden bir zincir oluşturulması ve bu veri zinciri o güne ait tek bir veriymiş gibi ağa verilerek, ağın bu verilerle eğitilmesi, doğrulanması ve test edilmesi gerekmektedir.

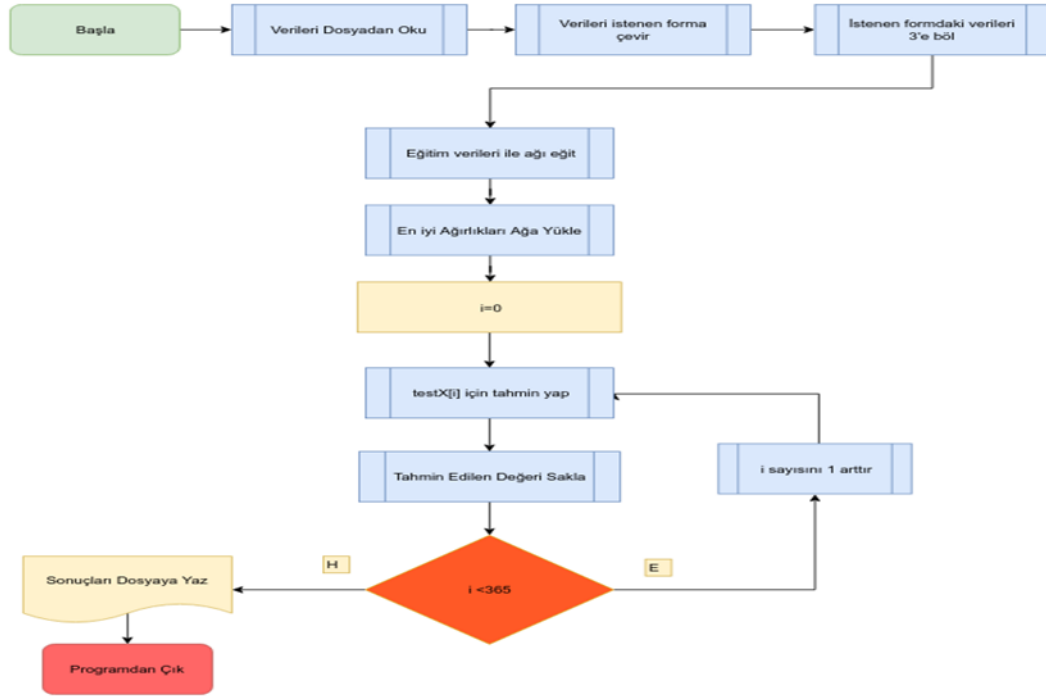


Şekil 4.5. Öz Yineleyen Sinir Ağları için hazırlanan veri yapısı

n kullanılacak gün sayısı, $X(t)$ istenen güne ait özelliklerin oluşturduğu vektör olmak üzere, $Giris(t)$ tahmin edilecek t günü için öz yineleyen yapay sinir ağlarına verilmek üzere oluşturulan t zamanına ait giriş verisinin genel yapısı Şekil 4.4.'de görüldüğü gibi oluşturulmaktadır.

Basit Öz Yineleyen Sinir Ağı, Uzun-Kısa Dönem Hafıza Öz Yineleyen Sinir Ağı ve Kapılı Öz Yineleyen Öge Sinir Ağları ise giriş katmanında parametre sayısı ve veri zinciri uzunluğunun çarpımı –özellik sayısı kadar- öz yineleyen düğüm ve herbirinde 30'ar öz yineleyen düğümden oluşan 3 gizli katman ve çıkış katmanında ise tek bir sinir düğümünün bulunduğu katmanlardan oluşturulmuştur.

Öz yineleyen ağlar girdi olarak aynı veri zincirini aldıkları için tek program içerisinde ard arda çalıştırılacak şekilde kodlanmıştır.



Şekil 4.6. Öz Yineleyen Sinir Ağları ile tahmin adımları

Öz yineleyen sinir ağlarıyla sonuçların tahmin işlemine ait adımlar Şekil 4.5.'de görüldüğü gibi yapılmıştır. Program içerisinde üç çeşit öz yineleyen sinir ağı verilerin bölünmesi adımından sonraki adımları döngü içerisinde birbirleri ardınca çalışmaktadır.

BÖLÜM 5. ARAŞTIRMA BULGULARI

5.1. Öz Yineleyen Sinir Ağlarında Kullanılacak Veri Uzunluğunun Belirlenmesi

Öz yineleyen ağlara tahmin edilecek günün verilerine ek olarak kaç güne kadar öncesinin verileceğinin belirlenmesi amacıyla bu çalışmada denemeler yapılmış ve en iyi veri zinciri uzunluğunun 3-7 günlük parametrelerden oluşturulan veri zincirlerinden elde edildiği görülmüştür.

Tablo 5.1. Veri zinciri uzunluğunun tahmin performansına etkisi

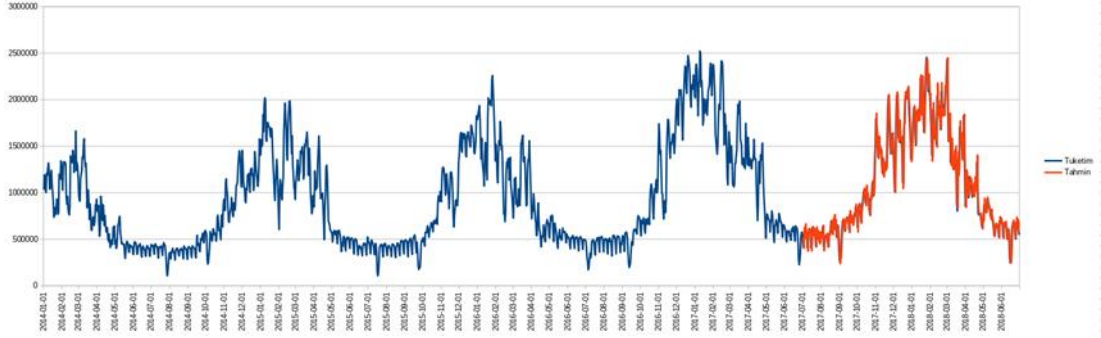
Veri Zinciri Uzunluğu (Gün)	Elde Edilen En İyi Sonuç (MAPE)
75	10,04
50	8,74
25	8,14
20	8,02
18	7,01
13	7,00
10	6,98
9	6,78
8	6,52
7	6,27
6	6,34
5	6,34
4	6,51
3	6,33
2	6,58

lr(öğrenme katsayısı)=0,0001 alınarak sonuçlar alınmıştır.

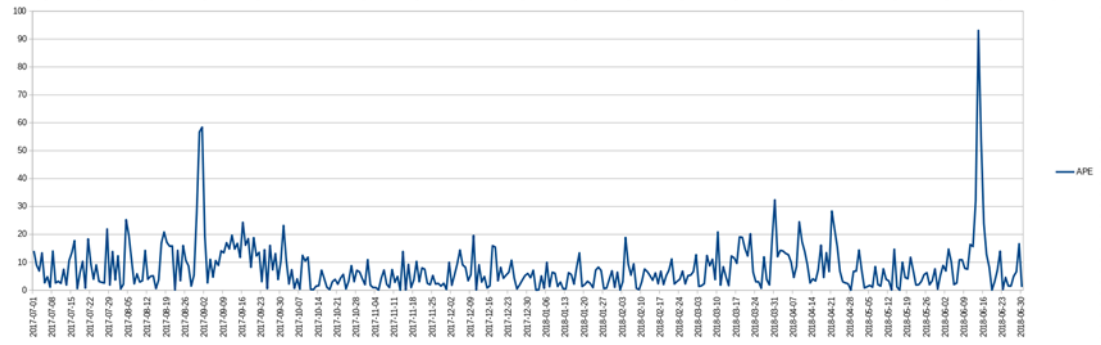
Tablo 5.1.'de alınan sonuçların yer aldığı tablodan da görüleceği gibi bu veri seti için doğal gaz tüketiminin 7 günden daha eski günlerdeki tüketimlerden etkilenmediği söylenebilir. Bu sebeple öz yineleyen sinir ağlarında 7 gün uzunluğundaki veri yapıları kullanılarak ağ eğitilerek en iyi sonuçlar elde edilmeye çalışılmıştır.

5.2. Elde Edilen Sonuçlar

5.2.1. SARIMA tahmin sonuçları



Şekil 5.1. SARIMA ile tahmin sonucu



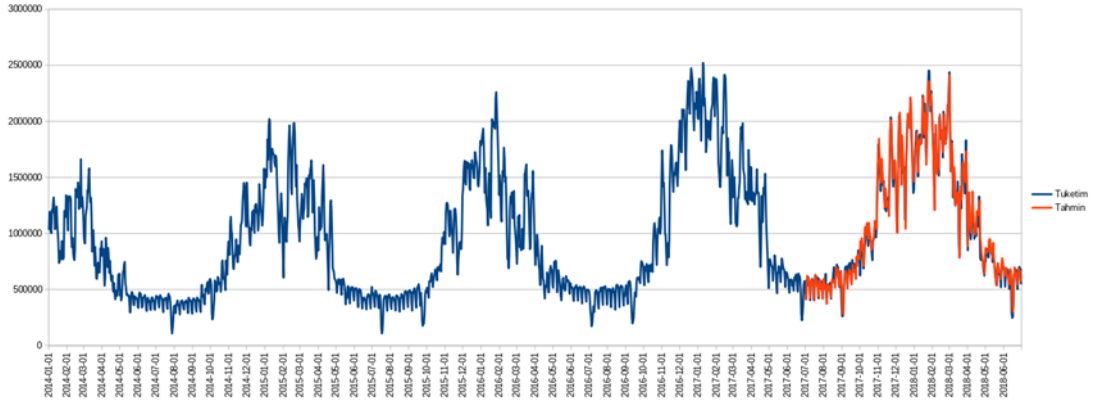
Şekil 5.2. SARIMA ile tahmin performansı

SARIMA ile yapılan tahminler Şekil 5.1.'de görüldüğü gibi gerçekleşirken bu tahminlere ait hata grafiği Şekil 5.2.'de ve bu tahminlere ait tutarlılık testleri de Tablo 5.2.'de görüldüğü gibi oluşmuştur.

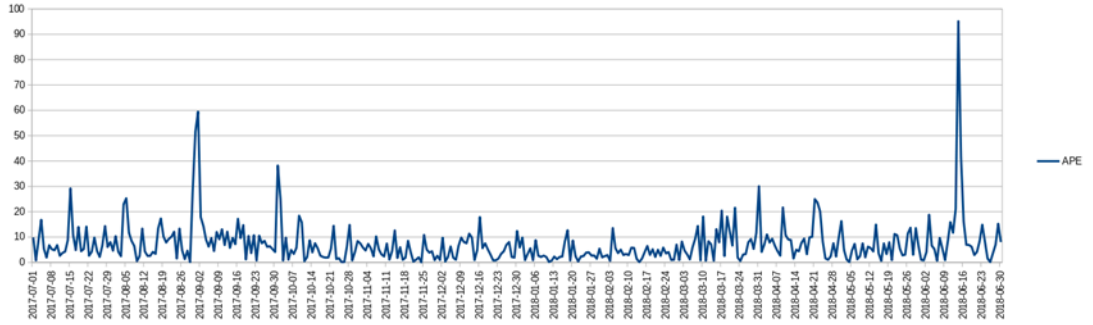
Tablo 5.2. Tutarlılık ve Hata Sonuçları (SARIMA)

Test	RMSE	NMSE	R ²
	0,00013	5,92	0,93

5.2.2. LASSO veri kümesi ile tahmin sonuçları



Şekil 5.3. LASSO parametreleri ile YSA tahmin sonucu

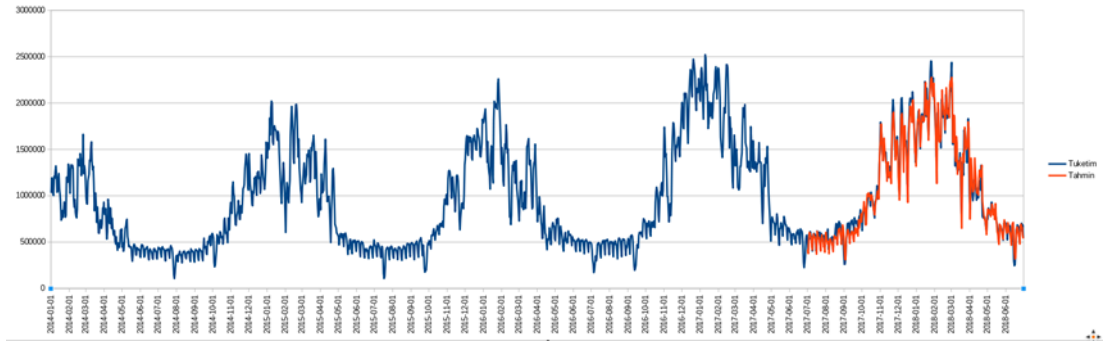


Şekil 5.4. LASSO ile YSA tahmin performansı

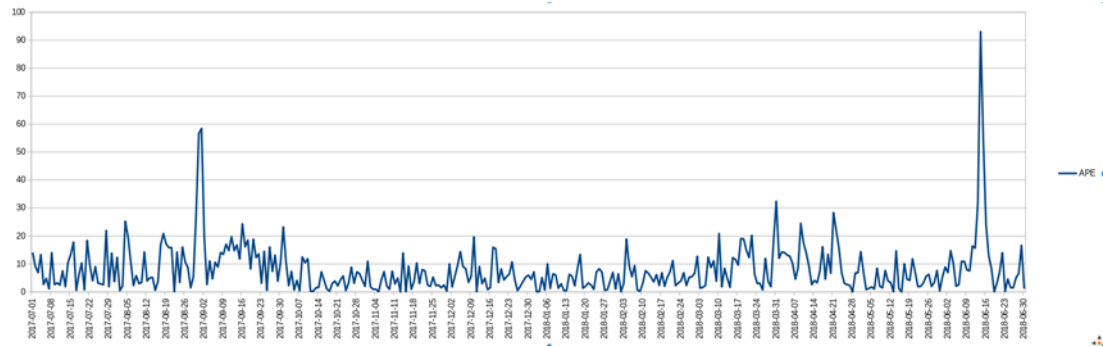
Tablo 5.3. Tutarlılık ve Hata Sonuçları (LASSO YSA)

	RMSE	NMSE	R ²
Eğitim	0,000019	6,063	0,984
Test	0,0000753	1,986	0,979

LASSO özellik seçme yöntemi ile oluşturulan veri seti kullanılarak eğitilen YSA ile yapılan tahmin sonucu Şekil 5.3.'de görüldüğü şekilde gerçekleşirken, elde edilen tahminlerin hata oranları Şekil 5.4.'de, bu tahminlere ait tutarlılık ve hata sonuçları ise Tablo 5.3.'te gösterilmiştir.



Şekil 5.5. LASSO ile DNN tahmin sonucu

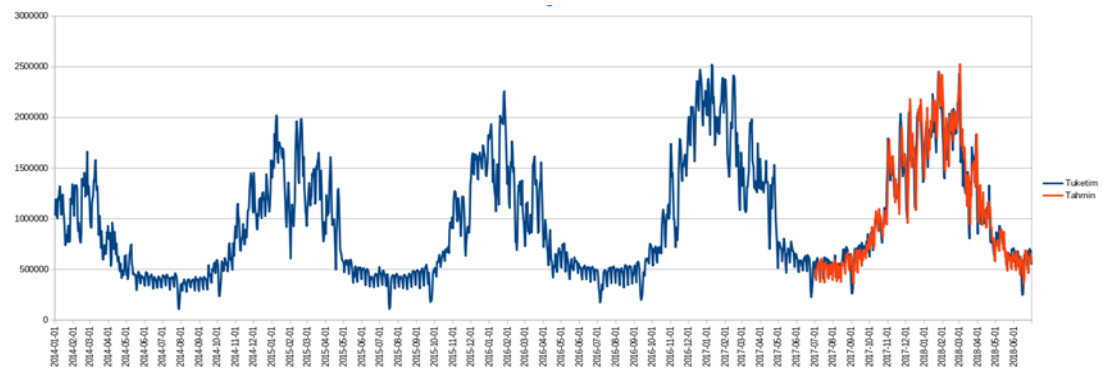


Şekil 5.6. LASSO ile DNN tahmin performansı

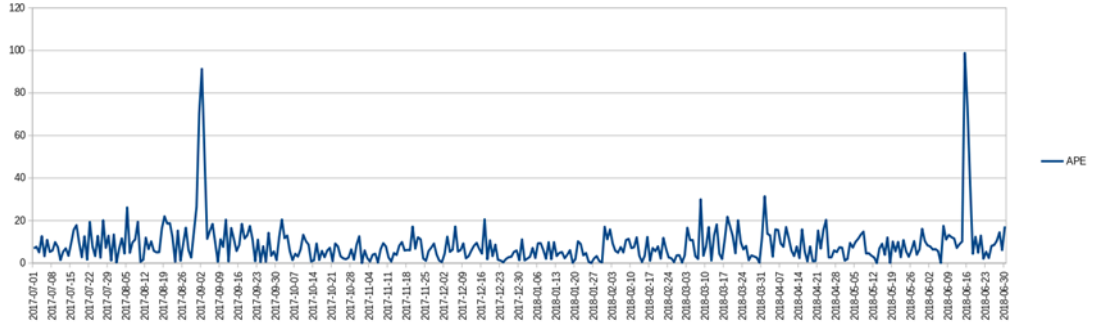
Tablo 5.4. Tutarlılık ve Hata Sonuçları (LASSO DNN)

	RMSE	NMSE	R ²
Eğitim	0,000017	4,880	0,986
Test	0,000084	2,497	0,972

LASSO yöntemi ile oluşturulan veri seti kullanılarak Tam Bağlı Derin Yapay sinir ağıyla yapılan tahminler Şekil 5.5.'te, bu tahminlere ait hata oranları Şekil 5.6.'da ve bu ağa ait tutarlılık ve hata sonuçları Tablo 5.4.'te görüldüğü gibi gerçekleşmiştir.



Şekil 5.7. LASSO ile SRNN tahmin sonucu

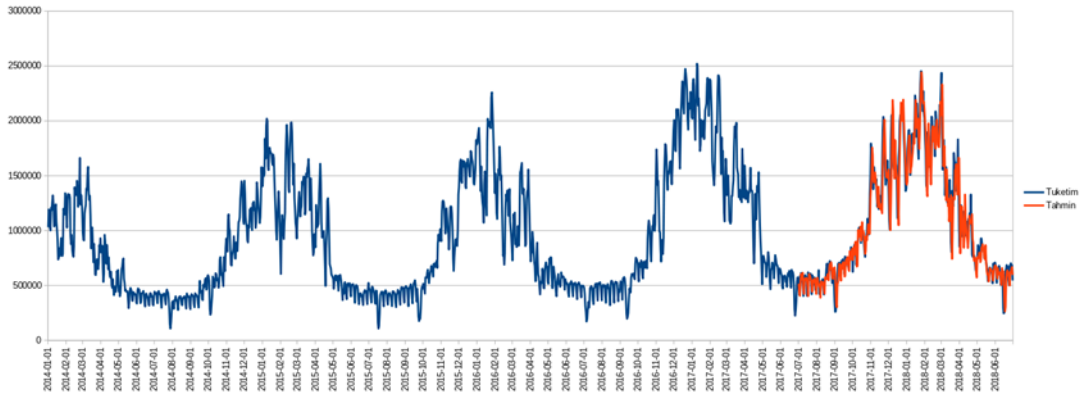


Şekil 5.8. LASSO ile SRNN tahmin performansı

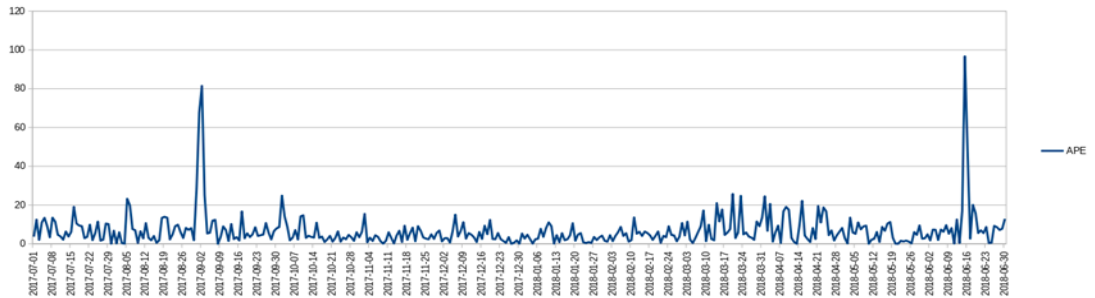
Tablo 5.5. Tutarlılık ve Hata Sonuçları (LASSO SRNN)

	RMSE	NMSE	R ²
Eğitim	0,000020	5,425	0,983
Test	0,000077	4,011	0,968

Şekil 5.7.'te gösterilen basit öz yineleyen sinir ağı ile yapılan tahminlere ait hata grafiği Şekil 5.8.'de, bu modele ait tutarlılık ve hata sonuçları ise Tablo 5.5.'te görülmektedir.



Şekil 5.9. LASSO ile LSTM tahmin sonucu

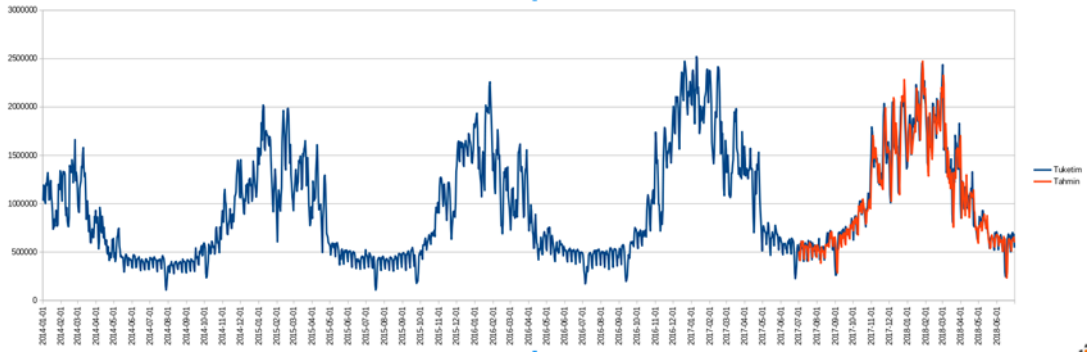


Şekil 5.10. LASSO ile LSTM tahmin performansı

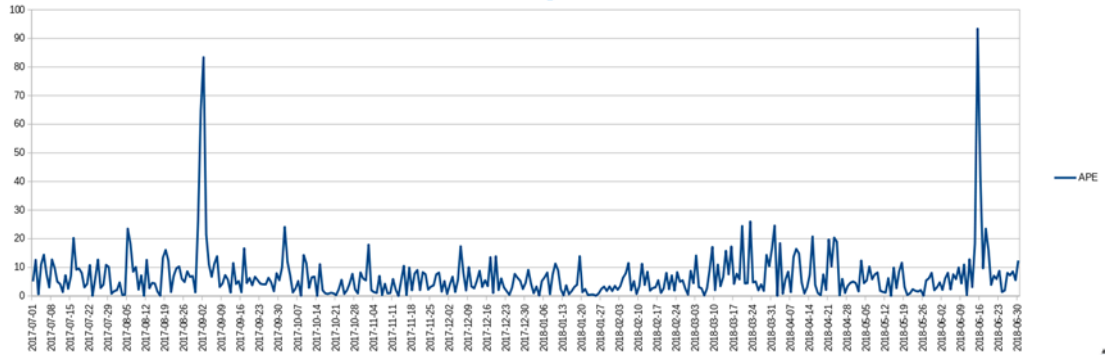
Tablo 5.6. Tutarlılık ve Hata Sonuçları (LASSO LSTM)

	RMSE	NMSE	R ²
Eğitim	0,000025	8,470	0,973
Test	0,000067	3,084	0,973

LSTM ağı ile yapılan tahminler Şekil 5.9.'de ve bu tahminlere ait mutlak yüzde hatalar da Şekil 5.10.'de görüldüğü gibi gerçekleşmiştir. Tablo 5.6.'daki verilere bakıldığında LSTM'in SRNN'den daha iyi sonuç verdiği görülmektedir.



Şekil 5.11. LASSO ile GRU tahmin sonucu



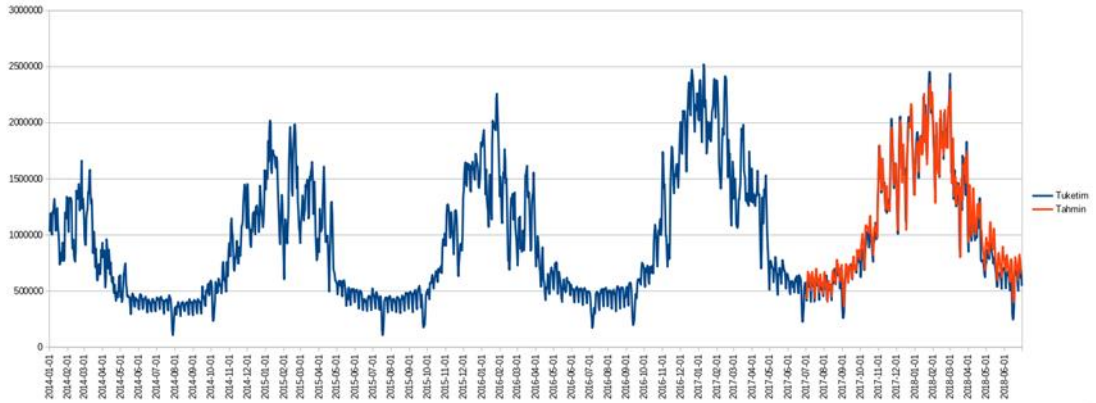
Şekil 5.12. LASSO ile GRU tahmin performansı

Tablo 5.7. Tutarlılık ve Hata Sonuçları (LASSO GRU)

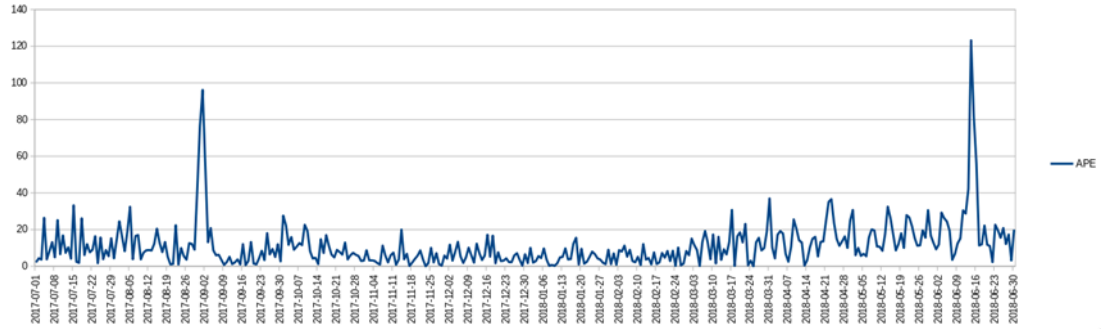
	RMSE	NMSE	R ²
Eğitim	0,000025	8,541	0,972
Test	0,000069	3,259	0,971

Kapılı öz yineleyen düğüm ağlarıyla yapılan tahminler Şekil 5.9.'da ve bu tahminlere ait hatalar Şekil 5.10.'da görülmektedir. Tablo 5.7.'de ağı tahmin performansı LSTM'e yakın sonuçlar verdiği görülmektedir.

5.2.3. RFE veri kümesi ile tahmin sonuçları



Şekil 5.13. RFE ile YSA tahmin sonucu

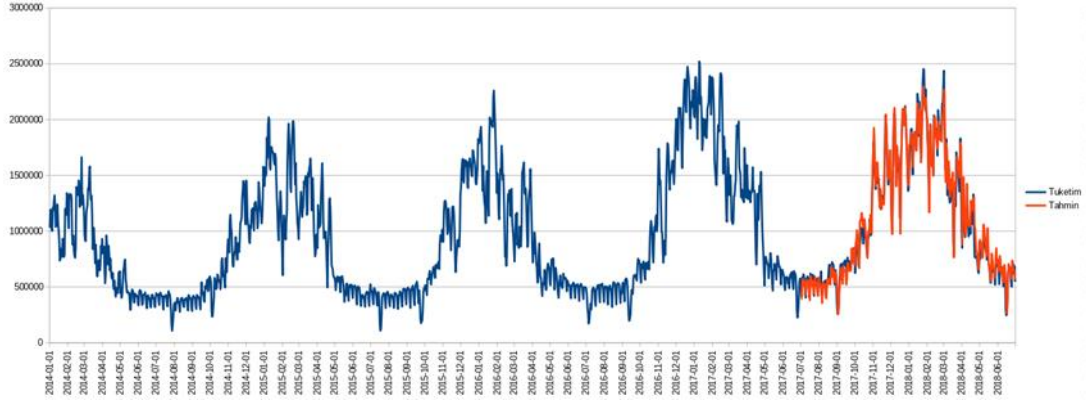


Şekil 5.14. RFE ile YSA tahmin performansı

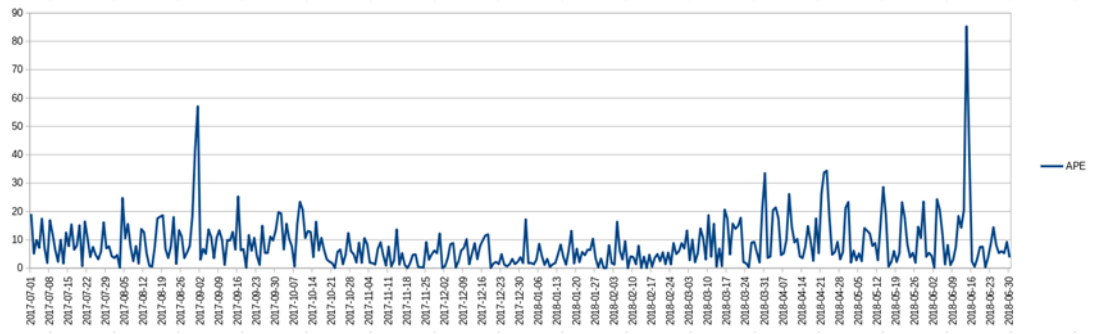
Tablo 5.8. Tutarlılık ve Hata Sonuçları (RFE YSA)

	RMSE	NMSE	R ²
Eğitim	0,000026	10,912	0,969
Test	0,000087	2,670	0,969

RFE yöntemiyle oluşturulan 4 parametrelili veri seti kullanılarak yapılan tahmin Şekil 5.13.'te ve bu tahminlere ait hatalar Şekil 5.14.' te görülmektedir. Modelin eğitim veri seti ve test veri seti için tutarlılık ve hata sonuçları Tablo 5.8.'de görüldüğü gibi olmaktadır.



Şekil 5.15. RFE ile DNN tahmin sonucu

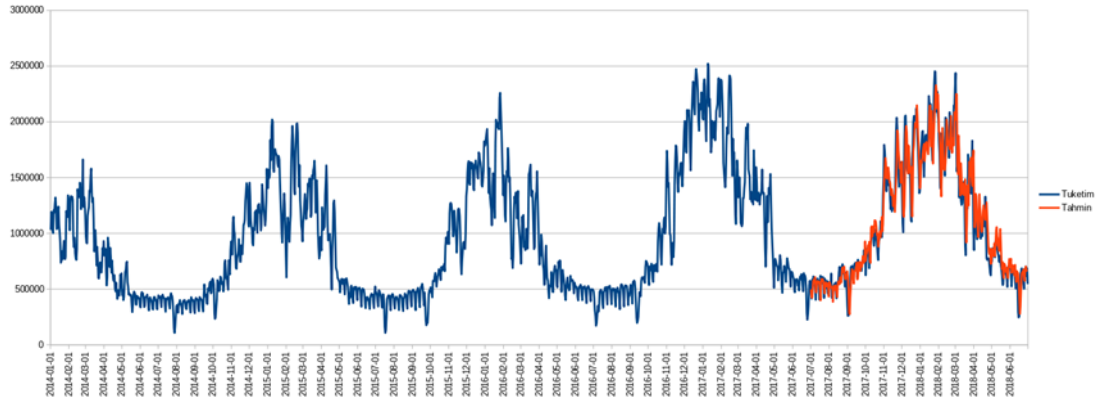


Şekil 5.16. RFE ile DNN tahmin performansı

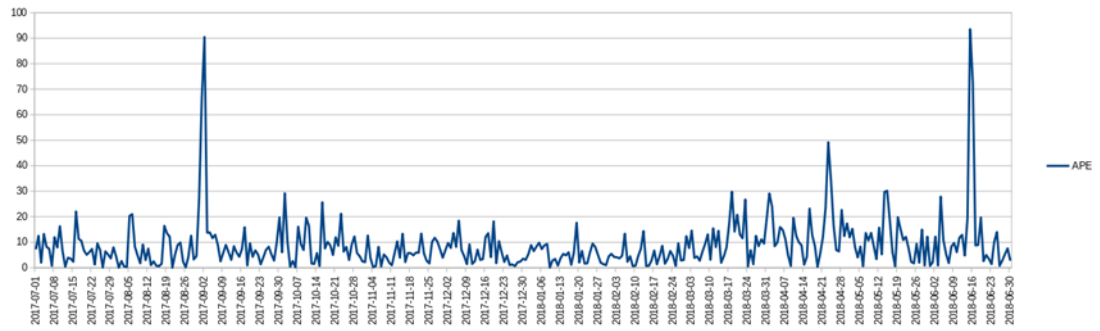
Tablo 5.9. Tutarlılık ve Hata Sonuçları (RFE DNN)

	RMSE	NMSE	R ²
Eğitim	0,000023	8,557	0,977
Test	0,000094	3,057	0,964

RFE yöntemiyle oluşturulan veri kümesinin 365 günlük tahmini Şekil 5.15., bu 365 günlük tahminin mutlak hata yüzdesi Şekil 5.16. görüldüğü gibi iken, modelin performans ve tutarlılık verileri Tablo 5.9.'da yer almaktadır.



Şekil 5.17. RFE ile SRNN tahmin sonucu

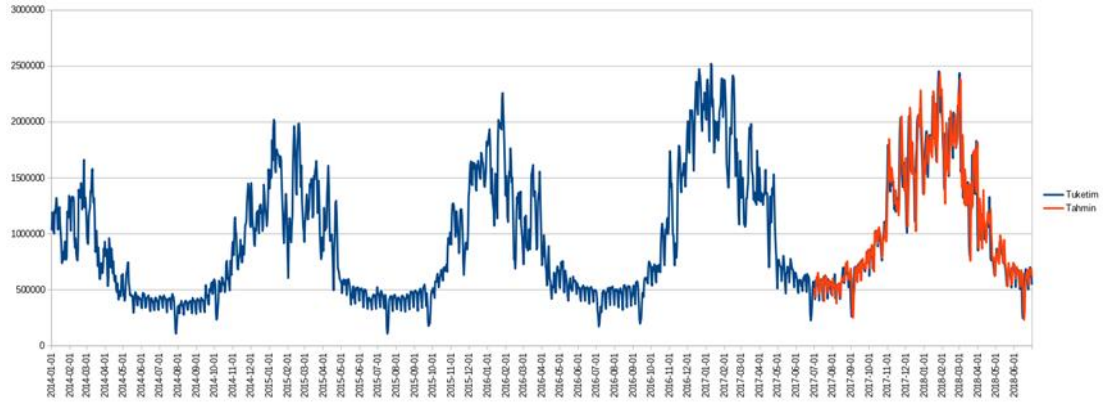


Şekil 5.18. RFE ile SRNN tahmin performansı

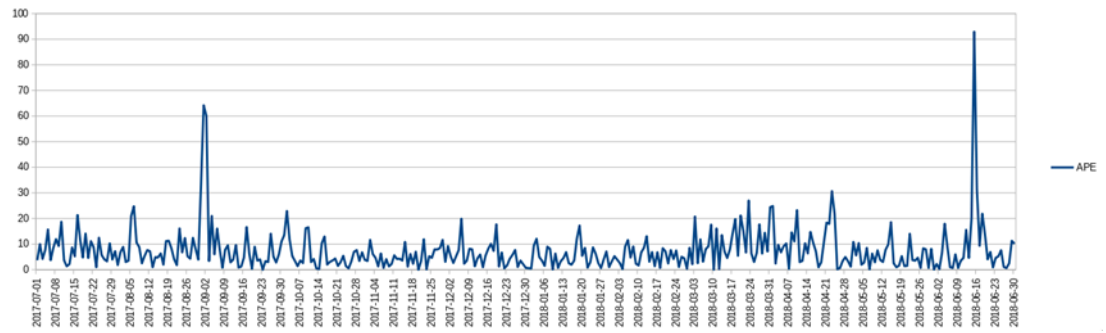
Tablo 5.10. Tutarlılık ve Hata Sonuçları (RFE SRNN)

	RMSE	NMSE	R ²
Eğitim	0,000028	10,907	0,967
Test	0,000083	4,586	0,956

RFE yöntemi ile seçilmiş veriler kullanılarak basit öz yineleyen sinir ağı ile yapılan 365 günlük tahmin Şekil 5.17.'de, 365 günlük mutlak yüzdelik hata Şekil 5.18.'de ve ağı tutarlılık ve hata sonuçları Tablo 5.10.'da görüldüğü gibidir.



Şekil 5.19. RFE ile LSTM tahmin sonucu

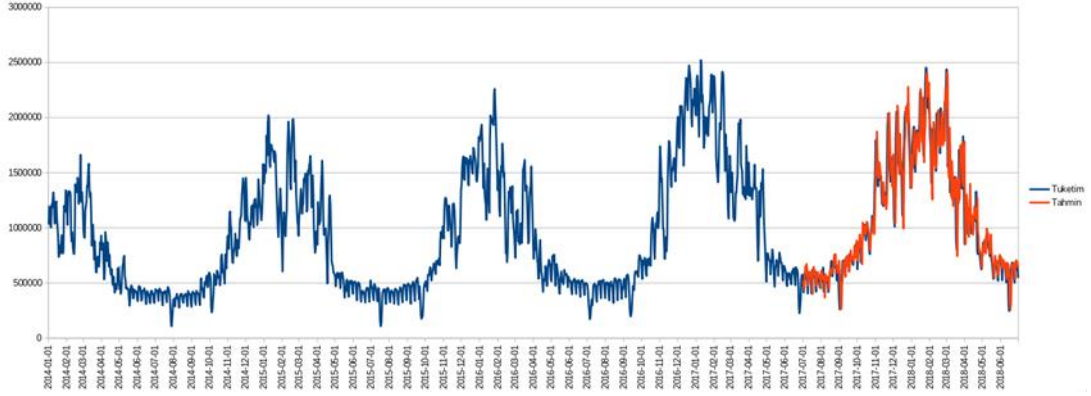


Şekil 5.20. RFE ile LSTM tahmin performansı

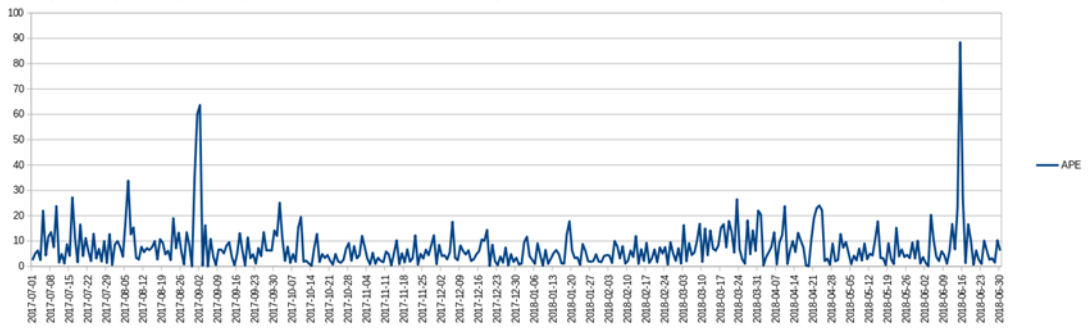
Tablo 5.11. Tutarlılık ve Hata Sonuçları (RFE LSTM)

	RMSE	NMSE	R ²
Eğitim	0,000028	11,122	0,963
Test	0,000076	3,820	0,966

LSTM ağının RFE metodu kullanılarak oluşturulmuş verilerle yaptığı tahmin ve performansı Şekil 5.19., Şekil 5.20. ve Tablo 5.11.'de görülmektedir.



Şekil 5.21. RFE ile GRU tahmin sonucu



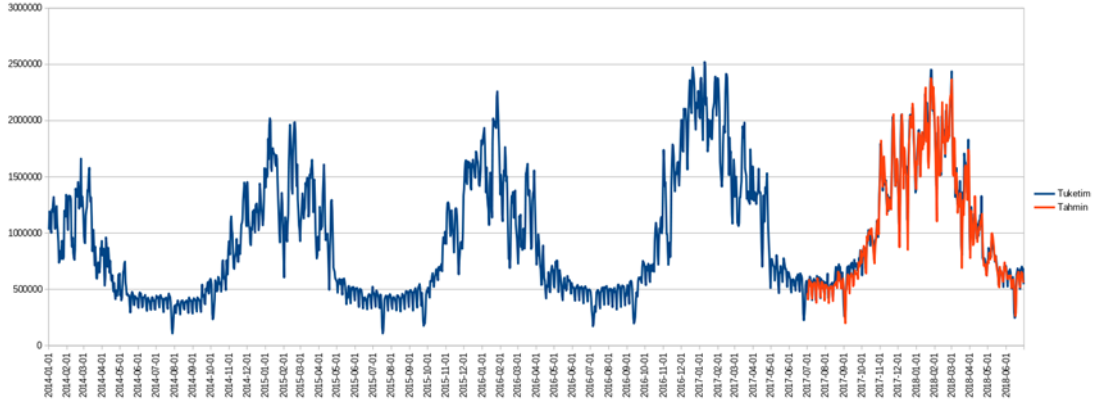
Şekil 5.22. RFE ile GRU tahmin performansı

Tablo 5.12. Tutarlılık ve Hata Sonuçları (RFE GRU)

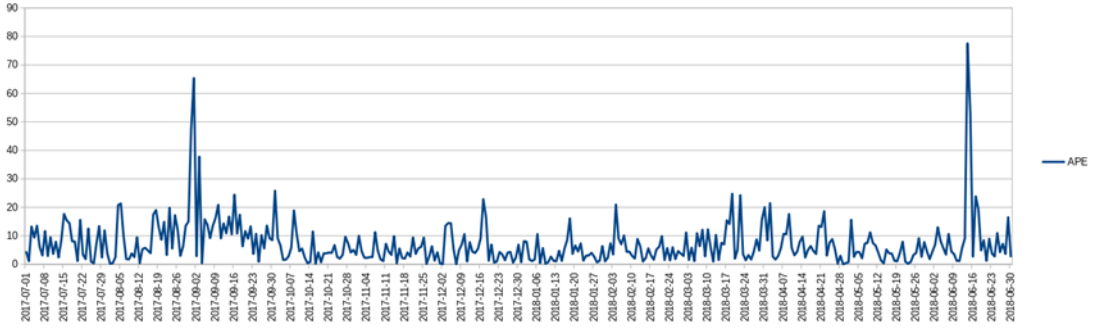
	RMSE	NMSE	R ²
Eğitim	0,000028	10,485	0,965
Test	0,000073	3,527	0,968

RFE ile oluşturulan veri kümesi ile GRU ağının 365 günlük tahmini Şekil 5.21.'de ve bu tahminlerin mutlak hata oranları Şekil 5.22.'de görülmektedir. Tablo 5.12.'de bulunan veriler incelendiğinde LSTM ve GRU'nun yakın sonuçlar verdiği söylenebilir.

5.2.4. Uzman görüşüne dayalı verilerle tahmin sonuçları



Şekil 5.23. Uzman görüşü ile YSA tahmin sonucu

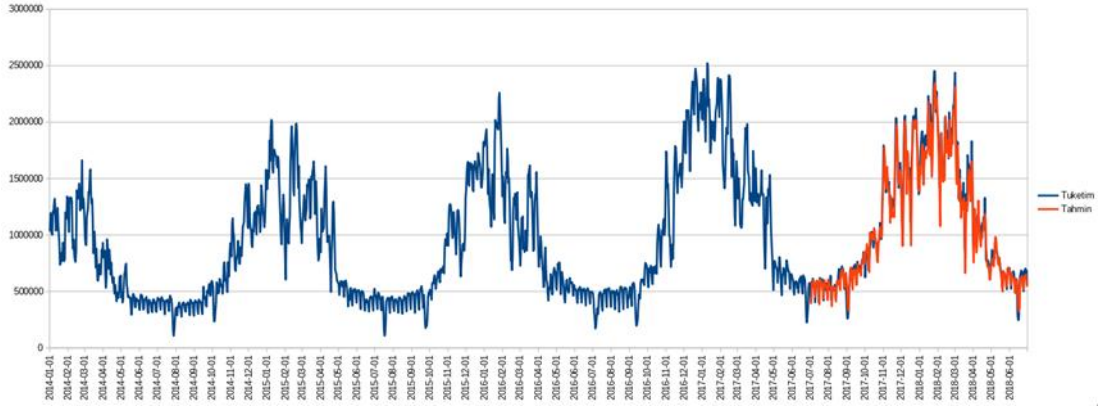


Şekil 5.24. Uzman görüşü ile YSA tahmin performansı

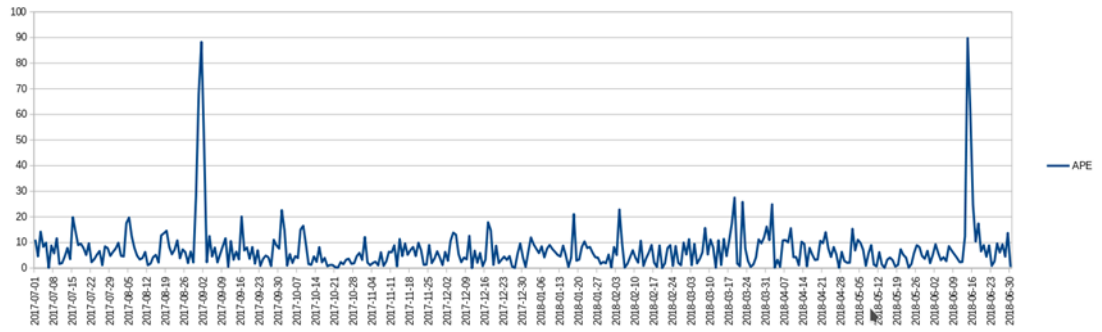
Tablo 5.13. Tutarlılık ve Hata Sonuçları (Uzman YSA)

	RMSE	NMSE	R ²
Eğitim	0,000026	10,309	0,973
Test	0,000086	2,646	0,969

Uzman görüşü doğrultusunda seçilen veriler kullanılarak basit YSA ile yapılan tahminler Şekil 5.23.'te bu tahminlere ait mutlak hata oranları Şekil 5.24.'te ağırlıklı tutarlılık ve performans kriterleri Tablo 5.13.'te görüldüğü gibi oluşmuştur.



Şekil 5.25. Uzman görüşü ile DNN tahmin sonucu

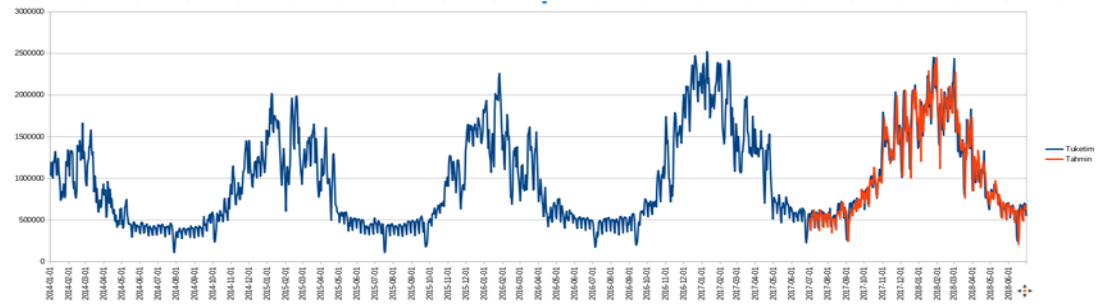


Şekil 5.26. Uzman görüşü ile DNN tahmin performansı

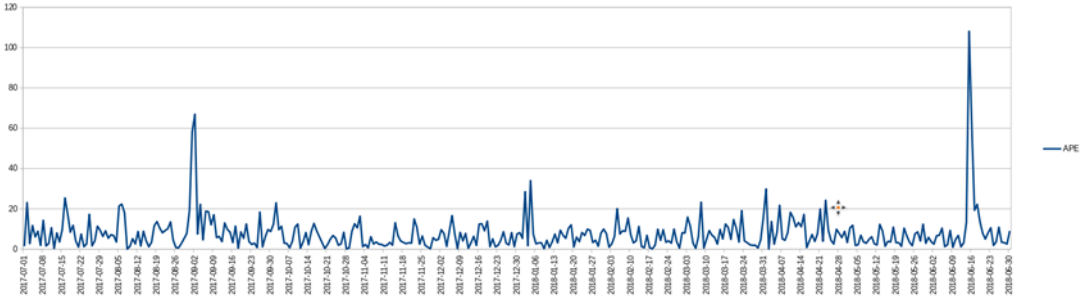
Tablo 5.14. Tutarlılık ve Hata Sonuçları (Uzman DNN)

	RMSE	NMSE	R ²
Eğitim	0,000028	11,775	0,970
Test	0,000087	2,711	0,971

Tam bağlı derin YSA modelinin Uzman görüşü ile oluşturulan veri kümesi ile yaptığı tahminler Şekil 5.25.'te, tahminlerin mutlak hata oranları Şekil 5.26.'da ve performans sonuçları ise Tablo 5.14.'te yer almaktadır.



Şekil 5.27. Uzman görüşü ile SRNN tahmin sonucu

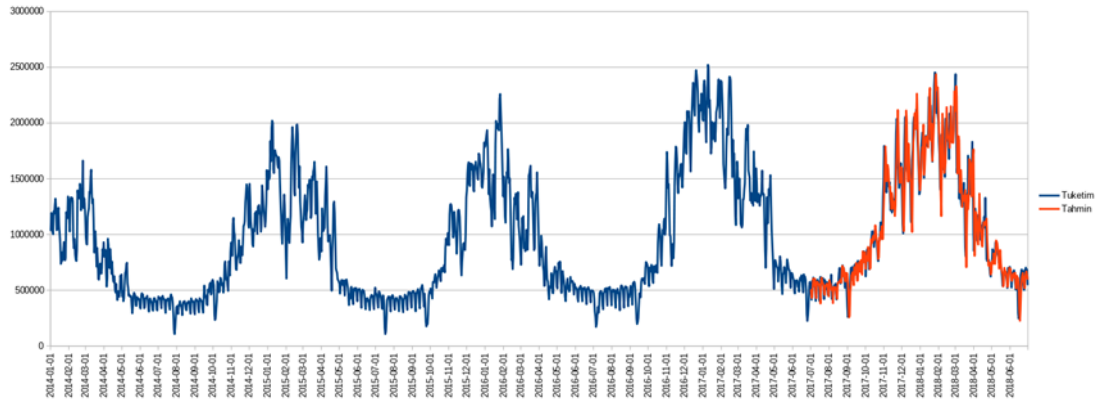


Şekil 5.28. Uzman görüşü ile SRNN tahmin performansı

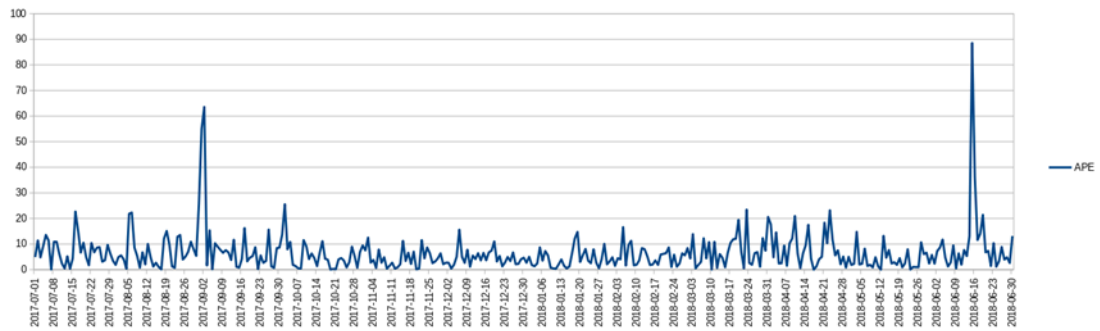
Tablo 5.15. Tutarlılık ve Hata Sonuçları (Uzman SRNN)

	RMSE	NMSE	R ²
Eğitim	0,000027	10,132	0,968
Test	0,000078	4,149	0,964

Basit RNN modelinin Uzman görüşü ile oluşturulan veri kümesi ile yaptığı tahminler Şekil 5.27.'de, tahminlerin mutlak hata oranları Şekil 5.28.'de ve performans sonuçları ise Tablo 5.15.'te yer almaktadır.



Şekil 5.29. Uzman görüşü ile LSTM tahmin sonucu

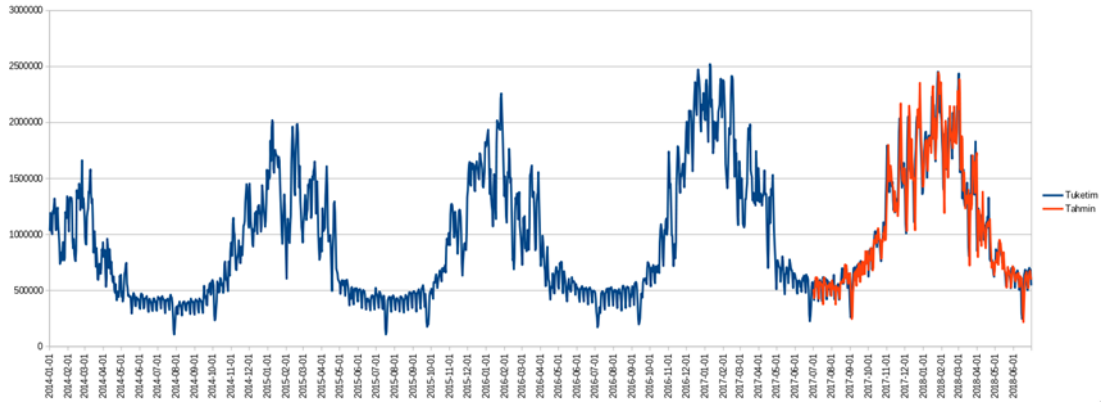


Şekil 5.30. Uzman görüşü ile LSTM tahmin performansı

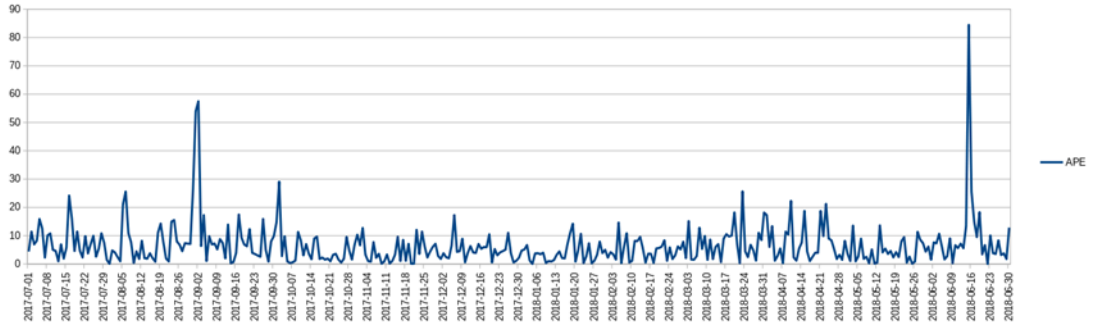
Tablo 5.16. Tutarlılık ve Hata Sonuçları (Uzman LSTM)

	RMSE	NMSE	R ²
Eğitim	0,000026	9,147	0,970
Test	0,000065	2,831	0,976

LSTM ağının Uzman görüşü ile oluşturulan veri kümesi ile yaptığı tahminler Şekil 5.29.'da, tahminlerin mutlak hata oranları Şekil 5.30.'da ve performans sonuçları ise Tablo 5.16.'da yer almaktadır.



Şekil 5.31. Uzman görüşü ile GRU tahmin sonucu



Şekil 5.32. Uzman görüşü ile GRU tahmin performansı

Tablo 5.17. Tutarlılık ve Hata Sonuçları (Uzman GRU)

	RMSE	NMSE	R ²
Eğitim	0,000025	8,666	0,971
Test	0,000065	2,850	0,976

GRU ağının Uzman görüşü ile oluşturulan veri kümesi ile yaptığı tahminler Şekil 5.31.'de, tahminlerin mutlak hata oranları Şekil 5.32.'de ve performans sonuçları ise Tablo 5.17.'de yer almaktadır.

5.3. Eğitim Sayılarının Sonuca Etkisi

Tablo 5.18. Eğitim sayılarının ve veri karıştırmanın sonuca etkisi

	Veri Karıştırma	Kapalı							Açık						
		Epoch	1000	1500	2000	2500	3000	4000	5000	1000	1500	2000	2500	3000	4000
	SARIMA	10,33													
UZMAN	GRU	7,77	7,43	6,98	6,78	6,66	6,51	6,38	7,32	7,04	6,86	6,74	6,65	6,48	6,46
	LSTM	8,39	7,68	7,27	7,15	6,93	6,70	6,45	7,66	7,17	6,93	6,78	6,69	6,62	6,56
	SRNN	9,19	8,84	8,64	8,57	8,65	8,10	7,97	8,81	8,78	8,49	8,08	8,25	7,81	7,61
	DNN	8,00	8,00	7,93	7,93	7,93	7,58	7,23	7,23	7,23	7,23	7,23	7,58	7,30	7,30
	YSA	7,75	7,73	7,56	7,25	7,25	7,49	7,25	7,04	6,88	6,73	6,73	6,73	6,68	6,84
RFE	GRU	8,97	7,98	8,44	8,17	7,68	7,38	7,46	7,97	7,49	7,52	7,54	7,29	7,20	7,13
	LSTM	10,39	8,81	8,42	7,82	7,75	7,45	7,30	7,92	7,53	7,38	7,30	7,25	7,21	7,26
	SRNN	9,56	10,32	9,22	8,90	8,44	8,27	7,99	11,14	11,14	10,07	10,04	8,57	8,39	8,32
	DNN	9,28	9,45	11,22	8,44	8,76	9,89	8,28	8,48	7,78	7,70	7,53	7,53	8,09	8,09
	YSA	12,18	12,18	11,16	11,16	11,16	11,16	11,16	7,65	7,69	7,60	7,50	7,41	7,26	7,30
LASSO	GRU	8,15	7,66	7,13	7,18	7,04	7,07	7,07	7,23	6,94	7,01	6,62	6,72	6,56	6,50
	LSTM	8,75	8,41	7,63	7,59	7,42	7,03	7,01	7,68	7,41	7,29	7,26	7,12	6,88	6,95
	SRNN	13,71	9,56	9,21	9,21	11,35	7,90	7,90	11,13	11,13	8,63	8,63	8,68	9,16	9,16
	DNN	7,95	9,16	8,59	8,57	8,57	7,56	7,56	7,46	7,46	7,46	7,46	7,46	7,46	7,46
	YSA	7,92	7,70	7,70	7,53	7,30	7,35	7,26	7,24	6,99	7,03	7,03	7,33	6,99	6,99

* $\alpha=0.0001$ alınmıştır.

365 günlük tahminler yapılarak elde edilen MAPE oranları Tablo 5.18.'de verilmiştir. Bu verilere göre Uzman görüşü ile indirgenen veri setinde GRU ağı en iyi sonucu vermiştir.

BÖLÜM 6. TARTIŞMA VE SONUÇ

Talep tahmini, gerek kıtlık oluşması problemini, gerekse ihtiyaçtan fazla malzemenin tedarik edilmesi ve depolanması problemlerinin önüne geçebilmek için gerçeğe olabildiğince yakın bir şekilde ihtiyacı belirleme problemidir. Günümüzde piyasada oluşan aşırı rekabetçi ortam sebebiyle, maliyetlerin azaltılması, israf ve kayıpların en aza indirilmesini zorunlu hale gelmiştir. Bu durum talep tahminine daha fazla önem verilmesini ve daha başarılı tahmin yöntemlerinin geliştirilmesini gerekli kılmaktadır.

Talep tahminleri uzun dönem (bir yıl ve üzeri), orta dönem (1 kaç günden 1 yıla kadar) ve kısa dönem (1 kaç saat ile 1 kaç güne kadar) olmak üzere talep süresine göre sınıflandırılmaktadır. Çalışmanın kapsamı gün öncesi talep tahmini -kısa dönemli doğalgaz talep tahmini- olarak belirlenmiştir.

Bu çalışmada 2014 Ocak-2018 Haziran arasında Sakarya ilinde gerçekleşen doğalgaz tüketim verileri kullanılarak Sezonsal Öz Bağımlı Bütünleşik Hareketli Ortalama metodu ile, basit YSA ve derin yapay sinir ağı modellerinin (GRU, LSTM, Simple RNN) doğalgaz tüketim tahmini performansları incelenmiştir.

Bu çalışmada SARIMA, basit ve derin yapay sinir ağlarının (Tam Bağlı Derin YSA, Simple RNN, LSTM ve GRU) tahmin performansları incelenmiştir. Ayrıca verilerin hazırlanması aşamasından sonra RFE, LASSO ve Uzman görüşü alınarak özellik seçimi işlemi yapılmıştır. Böylece hangi özellik azaltma yöntemlerinin daha başarılı olduğu da incelenmiştir. Bu yöntemler sayesinde sonuca etkisi az olan veya gürültülü veriler elenerek eğitim maliyetlerinin azaltılması sağlanmıştır.

Çalışma kapsamında yapılan tahminler için python dilinde (Python 3) uygulamalar geliştirilmiştir. SARIMA için pmdarima kütüphanesi, yapay sinir ağları için Keras ve

Tensorflow kütüphaneleri kullanılırken, RFE ve LASSO için ise sklearn kütüphanesi kullanılmıştır.

Özellik indirgeme işlemleri sonucu LASSO metodu 11 parametrelidir, RFE metodu 4 parametrelidir ve uzman görüşüne dayalı olarak 5 parametrelidir veri kümeleri oluşturulmuştur. Bu veri setleri kullanılarak oluşturulmuş olan 5 yapay sinir ağı modeli eğitilmiş ve bu veri kümelerinden eğitim adımından önce ayrılmış olan test verileri kullanılarak tahminler yapılmıştır. Tüm bu eğitimler sırasında öğrenme katsayısı olarak 0.0001 kullanılmış ve elde edilen sonuca göre en başarılı tahmini GRU ağı Uzman görüşü ile indirgenen veri setinde, veri karıştırma kapalı olarak eğitildiğinde elde edildiği görülmüştür.

Sonuç olarak çalışmamızdan elde edilen önemli bulgular şu şekilde özetlenebilir:

- Uygulamada kullandığımız veri seti için yılın günü, haftanın günü gibi ilave parametreler eklenerek tahmin başarısı arttırılabilmektedir.
- Çalışmamız sonucunda veri azaltma indirgeme yöntemlerinden en başarılısı Uzman görüşü alınarak özelliklerin seçildiği yöntem olduğu görülmüştür.
- Uzman görüşü metodu ile özellik azaltma yöntemi kullanıldığında bağıl nem ve günlük ortalama sıcaklık verisi dışındaki meteorolojik verilerin hatalı ölçümünden/tahmininden kaynaklanabilecek sakıncalardan sistem kurtarılabilmektedir.
- Veri karıştırma parametresinin eğitim esnasında açık olması öz yineleyen yapay sinir ağlarının performanslarını çok fazla etkilemektedir.

Yapılan bu tez çalışmasında öz yinelemeli yapay sinir ağlarının zaman serilerinde tahmin uygulamasında diğer tekniklerle karşılaştırılması yapılmıştır ve daha başarılı sonuçlar elde edildiği görülmüştür. Bu konuda yapılabilecek olan daha sonraki çalışmalarda ise modele dini ve milli bayramlar ile tatil günlerini de ilave ederek modelin başarısına etkisi incelenebilir.

KAYNAKLAR

- Ahmad T., Chen H., 2019. Deep learning for multi-scale smart energy forecasting. *Energy* 175, 98–112.
- Akpınar M. 2014. Eysel doğal gaz kullanımında konjonktür etkisinden arındırılmış tüketimin ARIMA ve Çoklu regresyon yöntemleri ile tahmini., Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar ve Bilişim Mühendisliği, Yüksek Lisans Tezi.
- Akpınar M. 2017. [İstatistik ve yapay zeka teknikleri ile enerji tüketiminin tahmini : Sakarya doğal gaz tüketiminin uygulaması.](#), Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar ve Bilişim Mühendisliği, Doktora Tezi.
- Akpınar M., Adak M. F., Yumusak N. 2017. Day-ahead natural gas demand forecasting using optimized abc-based neural network with sliding window technique: the case study of regional basis in Turkey., *Energies*, 10, 781, doi:10.3390/en10060781.
- Aygördü S. 2015. Kantitatif tahmin yöntemleri ile Türkiye aylık elektrik talebinin tahmini. Kara Harp Okulu, Savunma Bilimleri Enstitüsü, Harekât Araştırması, Yüksek Lisans Tezi.
- Bai Y., Li C., 2016. Daily natural gas consumption forecasting based on a structure-calibrated support vector regression approach. *Energy Build.* 127, 571–579.
- Balsever A. E. 2018. Enerji tüketim talebinin yapay zeka yöntemleri ile tahmini., İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği, Yüksek Lisans Tezi.
- Barak S., Sadegh S.S., 2016. Electrical Power and Energy Systems Forecasting energy consumption using ensemble ARIMA – ANFIS hybrid algorithm. *Int. J. Electr. Power Energy Syst.* 82, 92–104.
- Bek İ. M. 2008. Zaman serisi analizi ve tarımsal uygulaması. Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, Zootekni, Yüksek Lisans Tezi.
- Bekin A. 2015. Türkiye’de bazı temel gıda fiyatları için yapay sinir ağları ve zaman serisi tahmin modellerinin karşılaştırmalı analizi., Pamukkale Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme, Yüksek Lisans Tezi.

- Bengio Y., Simard P., Frasconi P. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.
- Beyca O.F., Ervural B.C., Tatoglu, E., Ozuyar P. G., Zaim S., 2019. Using machine learning tools for forecasting natural gas consumption in the province of Istanbul. *Energy Econ.* 80, 937–949.
- Cho K., van Merriënboer B., Bahdanau D., Y. Bengio Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches. <http://arxiv.org/abs/1409.1259>.
- Chou J.S., Tran D.S., 2018. Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential householders. *Energy* 165, 709–726.
- Chung J., Gulcehre C., Cho K., Bengio Y. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. <http://arxiv.org/abs/1412.3555>.
- Deb C., Eang L.S., Yang, J., Santamouris, M., 2015. Forecasting Energy Consumption of Institutional Buildings in Singapore. *Procedia Eng.* 121, 1734–1740.
- Durğun S. 2018. Türkiye'nin enerji talebinin yapay zekâ teknikleriyle uzun dönem tahmini., Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü, Enerji Sistemleri Mühendisliği, Yüksek Lisans Tezi.
- Duru Ö. 2007. Zaman serileri analizinde arıma modelleri ve bir uygulama., İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, Ekonometri, Yüksek Lisans Tezi.
- Ervural B. C., Beyca O. F., Zaim S., 2016. Model Estimation of ARMA Using Genetic Algorithms: A Case Study of Forecasting Natural Gas Consumption. *Procedia - Soc. Behav. Sci.* 235, 537–545.
- Filiz F. 2016. Elektrik piyasalarında elektrik yük talebi ve gün öncesi elektrik fiyat tahmini: Türkiye uygulaması., Ankara Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme, Yüksek Lisans Tezi.
- Gellert A., Florea A., Fiore U., Palmieri F., Zanetti P., 2019. A study on forecasting electricity production and consumption in smart cities and factories. *International Journal of Information Management* 1–11.
- Genç E. H. 2010. Construction of forecasting model for electric consumption in Turkey. Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Yüksek Lisans Tezi.
- Graves A. 2013. Generating sequences with recurrent neural networks. <https://arxiv.org/pdf/1308.0850.pdf>.

- Hribar R., Potocnik P., Silc J., Papa G., 2019. A comparison of models for forecasting the heating load of an urban area. *Energy* 167.
- Katsatos A.L., Moustiris K.P., Andri I., Pina A., Ferrão P., Fournier J., Lacarrière B., Corre O. Le, 2019. Application of Artificial Neuron Networks as energy consumption Application of Artificial Neuron Networks energy consumption forecasting tool in the building of Regulatory Authority of Energy, *Energy Procedia* 157, 851–861.
- Kaya Y. 2010. Türk Telekomünikasyon A.Ş.'de elektrik enerjisi tüketiminin yapay sinir ağları ile yerel tahmini., Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği, Yüksek Lisans Tezi.
- Khalifa, A., Caporin, M., Di Fonzo, T., 2019. Scenario-based forecast for the electricity demand in Qatar and the role of energy efficiency improvements. *Energy Policy* 127, 155–164.
- Khan U. N. 2018. Short-term load forecasting by using artificial neural networks. Bahçeşehir Üniversitesi, Doğa Bilimleri ve Uygulamalı Bilimler Enstitüsü, Elektrik ve Elektronik Mühendisliği, Yüksek Lisans Tezi.
- Kırbaş İ. 2018. İstatistiksel metotlar ve yapay sinir ağları kullanarak kısa dönem çok adımlı rüzgâr hızı tahmini. *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22(1), 24-38.
- Li, J., Dong, X., Shangguan, J., Hook, M., 2011. Forecasting the growth of China's natural gas consumption. *Energy* 36, 1380–1385.
- Liu, G., Dong, X., Jiang, Q., Dong, C., Li, J., 2018. Natural gas consumption of urban households in China and corresponding influencing factors. *Energy Policy* 122, 17–26.
- Meira, E.M., Cyrino, F.L., 2018. Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods. *Energy* 144, 776–788.
- Nou R. <https://people.duke.edu/~rnau/411arim.htm>, Erişim Tarihi 20.04.2019.
- Oğurlu H. 2011. Matematiksel modelleme kullanarak Türkiye'nin uzun dönem elektrik yük tahmini. Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik - Elektronik Mühendisliği, Yüksek Lisans Tezi.
- Ok Y. 2010. Adaptif Sinirsel Bulanık Çıkarım Sistemi (ANFIS) ile Türkiye'de orta dönemli elektrik enerjisi talep tahmini., Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği, Yüksek Lisans Tezi.
- Özdemir D. 2011. Yapay sinir ağları ile Kocaeli ilinin puant yük tahmini., Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Mühendisliği, Yüksek Lisans Tezi.

- Özkara Y. 2009. Mevsimsel ayrıştırma temelli gri tahmin yöntemi ile aylık elektrik yük tahmini., Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Bölümü, Yüksek Lisans Tezi.
- Pedamonti, D., 2018. Comparison of non-linear activation functions for deep neural networks on MNIST classification task., <https://arxiv.org/pdf/1804.02763.pdf>.
- Pınarbaşı M. 2009. Elektrik enerji sistemlerinde talep tahmin yöntemleri ve yapay sinir ağları uygulaması., İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği, Yüksek Lisans Tezi.
- Sen, P., Roy, M., Pal, P., 2016. Application of ARIMA for forecasting energy consumption and GHG emission: A case study of an Indian pig iron manufacturing organization. *Energy* 116, 1031–1038.
- Shaikh, F., Ji, Q., Shaikh, P.H., Mirjat, N.H., Uqaili, M.A., 2017. Forecasting China's natural gas demand based on optimised nonlinear grey models. *Energy* 140, 941–951.
- Srivastava N., Hinton G., Krizhevsky A. Sutskever I., Salakhutdinov R. 2014. Dropout: A simple way to prevent neural networks from overfitting., *Journal of Machine Learning Research*, 15, 1929-1958.
- Taşkıner B. 2018. Ankara ili doğal gaz tüketiminin yapay sinir ağları ile öngörüsü., İstanbul Teknik Üniversitesi, Enerji Enstitüsü, Enerji Bilim ve Teknoloji, Yüksek Lisans Tezi.
- Taşpınar, F., Çelebi, N., Tutkun, N., 2013. Forecasting of daily natural gas consumption on regional basis in Turkey using various computational methods. *Energy Build.* 56, 23–31.
- Tokgöz A. 2018. Tekrarlayan sinir ağı tabanlı elektrik tüketim tahmini, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği, Yüksek Lisans Tezi.
- Turhan C. 2012. Prediction of energy consumption of residential buildings by artificial neural networks and fuzzy logic., İzmir Yüksek Teknoloji Enstitüsü, Mühendislik ve Fen Bilimleri Enstitüsü, Enerji Mühendisliği, Yüksek Lisans Tezi.
- Tutu B. E. 2017. Türkiye için kısa vadeli elektrik enerjisi talep tahmini., TOBB Ekonomi ve Teknoloji Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme, Yüksek Lisans Tezi.
- Wu, W., Ma, X., Zeng, B., Wang, Y., Cai, W., 2019. Forecasting short-term renewable energy consumption of China using a novel fractional nonlinear grey Bernoulli model. *Renew. Energy* 140, 70–87.

- Xiao, J., Li, Y., Xie, L., Liu, D., Huang, J., 2018. A hybrid model based on selective ensemble for energy consumption forecasting in China. *Energy* 159, 534–546.
- Xiong, P., Dang, Y., Yao, T., Wang, Z., 2014. Optimal modeling and forecasting of the energy consumption and production in China. *Energy* 77, 623-634.
- Xu G., Wang W., 2010. Forecasting China's natural gas consumption based on a combination model. *J. Nat. Gas Chem.* 19, 493–496.
- Yamaçlı M. 2010. Türkiye'nin uzun dönem elektrik yük tahmini., Dumlupınar Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Elektronik Mühendisliği, Yüksek Lisans Tezi.
- Yasin G. M. 2013. Turkey's short term hourly electricity demand forecasting using artificial neural network. Başakşehir Üniversitesi, Doğa Bilimleri ve Uygulamalı Bilimler Fakültesi, Bilgisayar Mühendisliği, Yüksek Lisans Tezi.
- Yavuz Y. 2015. Ankara' daki evsel kullanıcıların MARS ve CMARS modelleri ile günlük doğal gaz tüketim tahmini., Orta Doğu Teknik Üniversitesi, Uygulamalı Matematik Enstitüsü, Bilimsel Hesaplama, Yüksek Lisans Tezi.
- Yavuzdemir M. 2014. Türkiye'nin kısa dönem yıllık brüt elektrik enerjisi talep tahmini., Ankara Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme, Yüksek Lisans Tezi.
- Zhang X., Zhou W. 2018. Forecast of China's natural gas consumption using mathematical models, *Energy Sources, Part B: Economics, Planning, and Policy*, 13:4, 246-250, DOI: 10.1080/15567249.2013.863410.

EKLER

EK 1: Veri Hazırlama Kodu

```
import catch as catch
import pandas as pd
import numpy as np
import xlwt
import openpyxl
import sqlite3 as db
sicaklik=[];
path = "/home/omer/Projeler"
con=db.connect("/home/omer/Projelerim/mydb.db")
cur = con.cursor()

x_train = []
y_train = []
i = 2013
cols = ['Yil', 'Ay', 'Gun','YGunu','Basinc', 'Sicaklik', 'G_SicaklikFarki',
        'G_BaslSicaklik', 'Nem', 'Bulutluluk', 'Ruzgar', 'Kuzey', 'Guney', 'Bati','Dogu',
        'AybasiAbone', 'AysonuAbone', 'Tuketim']
ydf = pd.DataFrame(columns=cols)
ydf2=pd.DataFrame(columns=['YGunu','EnDusukOrtalamaSicaklik'])
ay = 1
gun=1
ygun=0
gunBasiSicaklik = 0
abonesayisi1=0
```

```

abonesayisi=0
gunortalama=[[0]*2 for i in range(365)]
while i < 2019:
    # ay = 1
    while ay < 13:
        # gun = 1
        sicakliksayaci = 0
        sicakliktoplami = 0
        nemsayaci = 0
        nemtoplami = 0
        basincsayaci = 0
        basinctoplami = 0
        sayac4 = 0
        toplam4 = 0
        sayac5 = 0
        toplam5 = 0
        index = 0
        strGun = "
        strAy = "
        tuketim = 0
        abonesayisi1 = abonesayisi
        if i>2013:
            resp = cur.execute("SELECT * FROM Aboneler WHERE Yil==" + str(i) + "
                AND Ay==" + str(ay))
        else:
            if ay <12:
                resp = cur.execute("SELECT * FROM Aboneler WHERE Yil==" + str(i) +
                    " AND Gun==1 AND Ay==" + str(ay+1))
            else:
                resp = cur.execute("SELECT * FROM Aboneler WHERE Yil==" + str(i) +
                    " AND Gun==31 AND Ay==" + str(ay))
        res = resp.fetchall()

```

```

abonesayisi = 0
if res != []:
    for a in res:
        if i<2012:
            abonesayisi = a[3]
        else:
            abonesayisi=a[3]*1000
while gun < 32:
    if ay==2 and gun==29:
        ygun=ygun
    else:
        ygun=ygun+1
        if i==2007:
            gunortalama.append(([0], [0]))
if gun < 10:
    strGun = '0' + str(gun)
else:
    strGun = str(gun)
if ay < 10:
    strAy = '0' + str(ay)
else:
    strAy = str(ay)
resp = cur.execute("SELECT * FROM sicaklik WHERE Yil==" + str(i) + "
                    AND Gun==" + str(gun))
sicakliklar = resp.fetchall()
sicakliksayaci = 0
sicakliktoplami=0
for s in sicakliklar:
    if s[ay+3]!="":
        sicakliksayaci += 1
        sicakliktoplami += float(s[ay+3])
endusuksicaklik=50

```

```

resp2 = cur.execute('SELECT * FROM saatlikSicaklik WHERE Yil==' +
                    str(i) + ' AND Ay==' + str(ay) + ' AND Gun==' + str(gun))
ssicakliklar = resp2.fetchall()
for ss in ssicakliklar:
    if ss[6]<endusuksicaklik:
        endusuksicaklik=ss[6]
if endusuksicaklik<50:
    gunortalama[ygun-1][0]+=endusuksicaklik
    gunortalama[ygun-1][1]+=1
resp2 = cur.execute("SELECT * FROM nem WHERE Yil==" + str(i) + "
                    AND Gun==" + str(gun))
nemler = resp.fetchall()
nemsayaci = 0
nemtoplami=0
for n in nemler:
    if n[ay+3]!="":
        nemsayaci +=1
        nemtoplami += float(n[ay+3])
resp = cur.execute("SELECT * FROM basinc WHERE Yil==" + str(i) + "
                    AND Gun==" + str(gun))
basinclar = resp.fetchall()
basincsayaci = 0
basinctoplami=0
for b in basinclar:
    if b[ay+3]!="":
        basincsayaci += 1
        basinctoplami += float(b[ay+3])
resp2=cur.execute('SELECT * FROM ruzgar WHERE Yil==' + str(i) + ' AND
                    Gun==' + str(gun))
ruzgarlar=resp2.fetchall()
ruzgarsayaci=0
toplamruzgar=0.0

```

```
kuzey=0
```

```
guney=0
```

```
dogu=0
```

```
bati=0
```

```
for r in ruzgarlar:
```

```
    bosluk=0
```

```
    kuvvet=""
```

```
    index=0
```

```
    if len(r[ay+3])>0:
```

```
        ruzgarsayaci +=1
```

```
        while index<len(r[ay+3]):
```

```
            if bosluk==0:
```

```
                if r[ay+3][index]=='W':
```

```
                    bati+=1
```

```
                if r[ay+3][index]=='E':
```

```
                    dogu+=1
```

```
                if r[ay+3][index]=='N':
```

```
                    kuzey+=1
```

```
                if r[ay+3][index]=='S':
```

```
                    guney+=1
```

```
                if r[ay+3][index] == ' ':
```

```
                    bosluk=1
```

```
            else:
```

```
                kuvvet=kuvvet+r[ay+3][index]
```

```
            index+=1
```

```
        if len(kuvvet)>0:
```

```
            toplamruzgar+=float(kuvvet)
```

```
resp = cur.execute("SELECT * FROM tuketim WHERE Yil==" + str(i) + " "
                    "AND Ay==" + str(ay) + " AND Gun==" + str(gun))
```

```
tuketimler = resp.fetchall()
```

```

ttuketim=0.0
if tuketimler!=[]:
    for t in tuketimler:
        say=0
        dec=""
        while say<len(t[4]):
            if t[4][say]!=" " and t[4][say]!="," and t[4][say]!=".":
                dec+=t[4][say]
            elif t[4][say]==" ,":
                dec+="."
            say+=1
        ttuketim=float(dec)
resp = cur.execute("SELECT * FROM bulutluluk WHERE Yil==" + str(i) +
    " AND Gun==" + str(gun))
bulut = resp.fetchall()
bulutToplam = 0
bulutsayaci=0
if bulut!=[]:
    for b in bulut:
        if b[ay+3] != 0:
            try:
                bulutToplam += np.float(b[ay+3])
                bulutsayaci += 1
            except:
                continue
ortalama1 = 0
ortalama2 = 0
ortalama3 = 0
ortalama4 = 0
ortalama5 = 0
if sicakliksayaci > 0:
    ortalama3 = sicakliktoplami / sicakliksayaci

```

```

if basincsayaci > 0:
    ortalama1 = basinctoplami / basincsayaci
if nemsayaci > 0:
    ortalama4 = nemtoplami / nemsayaci
if ruzgarsayaci>0:
    ortalama2=toplamruzgar/ruzgarsayaci
    if bati>0:
        bati=(bati/ruzgarsayaci)
    if dogu>0:
        dogu=(dogu/ruzgarsayaci)
    if kuzey>0:
        kuzey=(kuzey/ruzgarsayaci)
    if guney>0:
        guney=(guney/ruzgarsayaci)
if bulutsayaci > 0:
    ortalama5 = bulutToplam/basincsayaci
ydf = ydf.append({'Yil': i, 'Ay': ay, 'Gun': gun,'YGunu':ygun, 'Basinc':
    ortalama1, 'Sicaklik': ortalama3, 'G_SicaklikFarki': endusuksicaklik,
    'G_BaslSicaklik': gunBasiSicaklik, 'Nem': ortalama4, 'Bulutluluk':
    ortalama5, 'Ruzgar':ortalama2, 'Kuzey':kuzey, 'Guney':guney,
    'Bati':bati, 'Dogu':dogu, 'AybasiAbone': abonesayisi1,
'AysonuAbone': abonesayisi, 'Tuketim': ttuketim}, ignore_index=True)
gunBasiSicaklik = ortalama3
print(str(gun) + "/" + str(ay) + "/" + str(i))
gun = gun + 1
if ay == 2 and gun == 29 and i % 4 != 0:
    gun = 32
    continue
if ay == 2 and gun == 30 and i % 4 == 0:
    gun = 32
    continue
elif ay == 4 or ay == 6 or ay == 9 or ay == 11:

```



```
        if gun == 31:
            gun = 32
            continue
    ay = ay + 1
    gun = 1
    i = i + 1
    ay = 1
    ygun=0

writer = pd.ExcelWriter("/home/omer/Projelerim/2014_2018_Dolu_Veriler.xlsx")
ydf.to_excel(writer, sheet_name="Sayfa 1")
writer.save()
indis=0
while indis<365:
    ydf2=ydf2.append({'YGunu': indis, 'EnDusukOrtalamaSicaklik':
                    (gunortalama[indis][0] / gunortalama[indis][1])}, ignore_index=True)
    indis+=1
writer2 =
pd.ExcelWriter("/home/omer/Projelerim/2014_2018_Ortalama_Sicaklik.xlsx")
ydf2.to_excel(writer2, sheet_name="Sayfa 1")
writer2.save()
print("\n")
```

EK 2: SARIMA Tahmin Kodu

```
import warnings
import pandas as pd
from pandas.plotting import autocorrelation_plot
from pmdarima import auto_arima
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.arima_model import ARIMA
import numpy as np
import matplotlib
matplotlib.use('Qt5Agg')
import itertools
import statsmodels.api as sm

path = "/home/omer/"
df = pd.read_excel(path+"ScaledDataY.xlsx", sheet_name="Sayfa 1")
dr=pd.date_range(start='01/01/2014', end='30/06/2018',freq='D')
df2 =df
df2.set_index(dr, inplace=True)

print(df.index)
a=1642-365
time_series2=df.head(a)
print("train :")
print(time_series2)
time_series=df.head(a).resample('M').mean()
test=df['Tuketim'].tail(365)
print('test :',test)

ord = (1, 1, 1)
seasonal_ord = (0, 0, 1, 12)
```

```

ydf=pd.DataFrame(columns=['tuketim', 'tahmin','otuketim'])
warnings.filterwarnings("ignore") # specify to ignore warning messages
#model2=auto_arima(df.head(a), start_p=0, start_q=0, start_P=0, start_Q=0,      #
                    max_d=2, max_p=3, max_q=3, max_P=3, max_Q=3, max_D=3, trace =
#                                True, m=12, seasonal=True)
#print(model2.order)
#print(model2.seasonal_order)
otuketim = 0
ind=0
t = 0
while t < 365:
    ind=a+t
    model2 = SARIMAX(df.head(a+t), order=ord, seasonal_order=seasonal_ord,
                    enforce_stationarity= False, enforce_invertibility=False)
    fitted = model2.fit(dispatch=False)
    tahmin = fitted.predict(start=test.index[t], end=test.index[t])
    otuketim = df["Tuketim"].head(a+t)[a+t-1]*3000000
    print('Onceki Tuketim = '+str(otuketim))
    print(tahmin[0]*3000000)
    ydf = ydf.append({'tuketim': test[t]*3000000, 'tahmin': tahmin[0]*3000000,
                    'otuketim' : otuketim}, ignore_index=True)
    print(t, "/ 365")
    t += 1

writer = pd.ExcelWriter("/home/omer/Sarima.xlsx")
ydf.to_excel(writer, sheet_name="Sayfa 1")
writer.save()

```

EK 3: Basit YSA ve Tam Bağlı Derin Sinir Ağı ile tahmin kodu

```

import pandas as pd
import numpy as np
from keras.layers import Dense, Dropout
from keras.models import Sequential, load_model
from keras.callbacks import ModelCheckpoint

path = "/home/omer/Belgeler/Dogalgaz/Toplu/"
df1 = pd.read_excel(path + "ScaledDataX05_4.xlsx", sheet_name="Sayfa 1")
df2 = pd.read_excel(path+"ScaledDataY.xlsx", sheet_name="Sayfa 1")

datas = []
x_train = []
y_train = []
xvalid=[]
yvalid=[]
x_test = []
y_test = []

i = 1
datasetsize = 1642
train_data_size = datasetsize - 365
gunList = []
xvalues = df1.values[:, 1:] # Tarih kısmı alma
yvalues = df2.values[:, 1:]

def prepareDatas2():
    i=0
    while i < 1641:
        if i < 1100:
            x_train.append([xvalues[i]])
            y_train.append([yvalues[i]])

```

```

elif i < 1276:
    xvalid.append([xvalues[i]])
    yvalid.append([yvalues[i]])
else:
    x_test.append([xvalues[i]])
    y_test.append([yvalues[i]])
i = i + 1
prepareDatas2()
giris = np.array(x_train).reshape(1100, 6)
cikis = np.array(y_train).reshape(1100, 1)
validx=np.array(xvalid).reshape(176, 6)
validy=np.array(yvalid).reshape(176, 1)
x_test_array = np.array(x_test).reshape(365, 6)
y_test_array = np.array(y_test).reshape(365, 1)

def buildDNNModel(input_dim):
    model1 = Sequential()
    model1.add(Dense(input_dim,input_dim=input_dim, activation="selu"))
    model1.add(Dense(30, activation="selu"))
    model1.add(Dense(30, activation="selu"))
    model1.add(Dense(30, activation="selu"))
    model1.add(Dense(1, activation="linear"))
    return model1

def buildYSAModel(input_dim):
    model1 = Sequential()
    model1.add(Dense(input_dim, input_dim=input_dim, activation="selu"))
    model1.add(Dense(30, activation="selu"))
    model1.add(Dense(1, activation="linear"))
    return model1

epoch=100

```

```

shf=False
fileName="model_YSA"
checkpointer = ModelCheckpoint(filepath=fileName+".hdf5", verbose=1,
                               save_best_only=True, monitor='val_loss', save_weights_only=False')
t=0
i=0
m=0
model=buildYSAModel(5)
#model= buildDNNModel(5) //YSA çalıştırılmak için yorum yapılmıştır.
while i<2:
    t=0
    if i == 0:
        shf = False
    else:
        shf = True
    while t<7:
        if t==0: epoch=1000
        elif t==1: epoch=1500
        elif t==2:epoch=2000
        elif t==3:epoch=2500
        elif t==4:epoch=3000
        elif t==5:epoch=4000
        else: epoch=5000
        model.compile(loss='mae', optimizer='rmsprop')
        model.summary()
        if(t>1):
            model.load_weights('last+'.hdf5')
        history = model.fit(x=giris, y=cikis, epochs=epoch-m, batch_size=64, verbose=0,
                           callbacks=[checkpointer], validation_data= (validx, validy), shuffle=shf)
        model.save(fileName+'.h5')
        model.save_weights('last+'.hdf5)

```

```

model.load_weights(fileName+'.hdf5')
model.compile(loss='mae', optimizer='rmsprop')
y_tahmin = model.predict(x_test_array)
y_tahmin_array = np.array(y_tahmin)
cols = ['Yil', 'Ay', 'Gun', 'Tahmin1', 'Tuketim1','APE']
ydf = pd.DataFrame(columns=cols)
k = 0
gtahmin=0
gtuketim=0
while k < 365:
    gtahmin=y_tahmin_array[k][0] * 3000000
    gtuketim=y_test_array[k][0] * 3000000
    ydf = ydf.append({'Tahmin1': gtahmin, 'Tuketim1': gtuketim,
                    'APE': (abs(gtahmin-gtuketim) / gtuketim)*100}, ignore_index=True)
    k = k + 1
    writer = pd.ExcelWriter( "/home/omer/Belgeler/Dogalgaz/Toplu/tahminler/"
                            + fileName + "_06prm_shf" + str(shf) + "_epoch" + str(epoch)+".xlsx")
    ydf.to_excel(writer, sheet_name="Sayfa 1")
    m=epoch
    t+=1
writer.save()
print("\n")
i+=1

```

EK 4: Öz Yineleyen Ağlarla tahmin kodu:

```

import pandas as pd
import numpy as np
import tensorflow as tf
import keras
from keras.layers import Dense, LSTM, GRU, SimpleRNN, Dropout
from keras.models import Sequential, load_model
from keras.callbacks import ModelCheckpoint

path = "/home/omer/Belgeler/Dogalgaz/Toplu/"
df1 = pd.read_excel("/media/omer/DEPO1/scaled/ScaledDataX05_4.xlsx",
sheet_name="Sayfa 1")
datasX = df1.as_matrix()
df2 = pd.read_excel(path+"ScaledDataY.xlsx", sheet_name="Sayfa 1")
datasY = df2.as_matrix()

x_array = []
y_array = []
x_train = []
y_train = []
x_valid = []
y_valid = []
x_test = []
y_test = []
preparedX=[]
preparedY=[]
input_dim = np.array(datasX).shape[1]-1
sonucList=[]
input_len = 50
batch_size = 50

def yazdirr():

```



```

a = 0
while a < 5:
    print(x_train[batch_size+a+1][0][9]*3000000)
    print(y_train[a][0]*3000000)
    a += 1

```

```

def buildGRUModel2(input_dim1, input_dim2):
    model1 = Sequential()
    model1.add(GRU(30, input_shape=(1, input_dim2), return_sequences=True))
    model1.add(GRU(30, return_sequences=True))
    model1.add(GRU(30, return_sequences=True))
    model1.add(GRU(30))
    model1.add(Dense(1, activation="linear"))
    return model1

```

```

def buildLSTMMModel2(input_dim1, input_dim2):
    model1 = Sequential()
    model1.add(LSTM(30, input_shape=(1, input_dim2), return_sequences=True))
    model1.add(LSTM(30, return_sequences=True))
    model1.add(LSTM(30, return_sequences=True))
    model1.add(LSTM(30))
    model1.add(Dense(1, activation="linear"))
    return model1

```

```

def buildSimpleRNNModel2(input_dim1, input_dim2):
    model1 = Sequential()
    model1.add(SimpleRNN(30, input_shape=(1, input_dim2),
activation="selu", return_sequences = True))
    model1.add(SimpleRNN(30, activation="selu", return_sequences=True))
    model1.add(SimpleRNN(30, activation="selu", return_sequences=True))
    model1.add(SimpleRNN(30, activation="selu"))
    model1.add(Dense(1, activation="linear"))

```

```
return model1
```

```
def PrepareDatas(dataLength):
    i = dataLength
    length = len(datasX)
    values = df1.values[:,1:] #Tarih kısmını alma
    yvalues = df2.values
    while i < length:
        k = i-dataLength
        tmpdata2 = []
        tmpdata1 = []
        tmpdata = []
        tmpdata1 = values[:i, :] #i.elemanına kadar al
        tmpdata2 = tmpdata1[k:, :]
        tmpdata = np.array(tmpdata2).reshape(1, -1)

        if k < 1000:
            x_train.append(tmpdata[0])
            y_train.append(yvalues[i-1][1])
        elif k < 1200:
            x_valid.append(tmpdata[0])
            y_valid.append(yvalues[i-1][1])
        else:
            x_test.append(tmpdata[0])
            y_test.append(yvalues[i-1][1])
        i += 1
    return
```

```
def sonucYaz (sonucarray2):
    cols2=['Dosya', 'Sonuc']
    ydf5=pd.DataFrame(columns=cols2)
    sayi=len(sonucarray2)
```

```

p=0
while p < sayi:
    ydf5 = ydf5.append({'Dosya': sonucarray2[p][0], 'Sonuc':
sonucarray2[p][1]},ignore_index=True)
    p += 1
writer2 =
pd.ExcelWriter("/media/omer/DEPO1/Sonuclar/Sonuc0"+str(sayi)+".xlsx")
ydf5.to_excel(writer2)
writer2.save()
return

```

```

n = 0
t = 0
t1 = 0
m = 0
sfl = False
SonucArray = []

```

```

sfl = True
epoch = 2000
while n < 1:
    if n == 0:
        input_len = 7
    elif n == 1:
        input_len = 5
    elif n == 2:
        input_len = 4
    elif n == 3:
        input_len = 3
    else:
        input_len = 2
    n += 1

```

```

x_train = []
y_train = []
x_valid = []
y_valid = []
x_test = []
y_test = []
PrepareDatas(input_len)
giris = np.array(x_train)
valid_X = np.array(x_valid)
test_X = np.array(x_test)
giris = giris.reshape(giris.shape[0], 1, giris.shape[1])
cikis = np.array(y_train).reshape(-1, 1)
valid_X = valid_X.reshape(valid_X.shape[0], 1, valid_X.shape[1])
valid_y = np.array(y_valid).reshape(-1, 1)
test_X = test_X.reshape(test_X.shape[0], 1, test_X.shape[1])
test_y = np.array(y_test).reshape(-1, 1)
print("Input Length="+str(input_len))
print(giris.shape, cikis.shape, valid_X.shape, valid_y.shape, test_X.shape,
test_y.shape)
print("Input Length="+str(input_len))

t = 0
t1 = 0
m = 0
sfl = False
while m < 3:
    if m == 0:
        dosyaAd = "SRNN_model0"+str(input_len)+"_sfl_"
    elif m == 1:
        dosyaAd = "LSTM_model0"+str(input_len)+"_sfl_"
    else:
        dosyaAd = "GRU_model0"+str(input_len)+"_sfl_"

```

```

if m == 0:
    model = buildSimpleRNNModel2(giris.shape[1], giris.shape[2])
elif m == 1:
    model = buildLSTMMModel2(giris.shape[1], giris.shape[2])
else:
    model = buildGRUModel2(giris.shape[1], giris.shape[2])
model.save_weights(dosyaAd+"_initial")
Opt = keras.optimizers.rmsprop(lr=.0005)
l = 0
while l < 2:

    if l == 0:
        sfl = False
    else:
        sfl = True

    writer = pd.ExcelWriter("/media/omer/DEPO1/Sonuclar/" + dosyaAd + "_" +
str(input_dim)
                                + "prm_1out_shf" + str(sfl) + ".xlsx")

    t = 0
    t1 = 0

    checkpointer = ModelCheckpoint(filepath=dosyaAd + ".hdf5", verbose=1,
save_best_only =True,monitor='val_loss',
save_weights_only='False')

    while t < 7:
        if t == 0:
            epoch = 1000
            model.load_weights(dosyaAd+"_initial")
        elif t == 1:
            epoch = 1500
        elif t == 2:
            epoch = 2000

```

```

elif t == 3:
    epoch = 2500
elif t == 4:
    epoch = 3000
elif t == 5:
    epoch = 4000
else:
    epoch = 5000

if t != 0:
    model.load_weights("sonAgirliklar.hdf5")
    model.compile(loss='mae', optimizer=Opt)
    model.summary()
    history = model.fit(x = giris, y = cikis, epochs = (epoch-t1), verbose =0,
callbacks =
[checkpointer],validation_data=(valid_X,valid_y),batch_size =
    input_len * 25, shuffle = sfl)
    model.save_weights("sonAgirliklar.hdf5")
    model.load_weights(dosyaAd + '.hdf5')
    model.compile(loss='mae', optimizer='rmsprop')
    y_tahmin = model.predict(test_X)
    y_tahmin_array = np.array(y_tahmin*3000000)
    cols = ['Tahmin', 'Tuketim', 'APE']
    ydf = pd.DataFrame(columns=cols)
    sum = 0.0
    k = 0
    while k < 365:
        tuketim = test_y[k][0]*3000000
        ape = ((abs(y_tahmin_array[k][0]-tuketim)/tuketim)*100)
        tahmin = y_tahmin_array[k][0]
        ydf = ydf.append({'Tahmin': tahmin, 'Tuketim': tuketim, 'APE': ape},
ignore_index=True)

```

```

        sum += ape
        k = k + 1
        ydf = ydf.append({'Tahmin': "", 'Tuketim': "", 'APE': sum/365},
ignore_index=True)

```

```

SonucArray.append([dosyaAd+"_epoch"+str(epoch)+'_sfl_'+str(sfl),sum/365])

```

```

        ydf.to_excel(writer, sheet_name="epoch" + str(epoch))
        t += 1
        t1 = epoch
        print("t1= "+str(t1))
        writer.save()
        print(dosyaAd)
        print("*****\n")
        t = 0
        t1 = 0
        l += 1
        m += 1
        l = 0
        sonucYaz(SonucArray)
        m = 0
        l = 0

```

ÖZGEÇMİŞ

Ömer Evcim, 01.08.1984'de Muğla'da doğdu. İlkokulu Muğla'da, ortaokulu Manisa'da ve lise eğitimini Ankarada'da tamamladı. 2006 yılında Sakarya Üniversitesinde memur olarak göreve başladı. 2008 yılında Anadolu Üniversitesi İşletme bölümünden, 2009 yılında Sakarya Üniversitesi Adapazarı Meslek Yüksekokulu Bilgisayar Teknolojileri ve Programlama bölümünden mezun oldu. 2010 yılında askerlik görevini kısa dönem erbaş olarak yaptı. 2013 yılında Sakarya Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünden mezun olan Ömer Evcim, evli ve bir çocuk babasıdır.