



T.C.
NECMETTİN ERBAKAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



DERİN ÖĞRENME İLE ZAMAN
SERİLERİNİN GERÇEK ZAMANLI TAHMİNİ

Ebru Şeyma KARAKOYUN

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Ana Bilim Dalı

Haziran 2018
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Ebru Şeyma Karakoyun tarafından hazırlanan “Derin Öğrenme ile Zaman Serilerinin Gerçek Zamanlı Tahmini” adlı tez çalışması 28/06/2018 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Ana Bilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri**İmza****Başkan**

Prof. Dr. Sabri KOÇER

.....

Danışman

Dr. Öğr. Üyesi Ali Osman ÇIBIKDİKEN

.....

Üye

Dr. Öğr. Üyesi Ahmet Ercan TOPCU

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mehmet KARALI
FBE Müdürü

Bu tez çalışması Bilimsel Araştırma Projeleri Koordinatörlüğü tarafından **181319003** nolu proje ile desteklenmiştir.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Ebru Şeyma Karakoyun

19.06.2018

ÖZET**YÜKSEK LİSANS TEZİ****DERİN ÖĞRENME İLE ZAMAN SERİLERİNİN
GERÇEK ZAMANLI TAHMİNİ****Ebru Şeyma KARAKOYUN****Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Ana Bilim Dalı****Danışman: Dr. Öğr. Üyesi Ali Osman ÇIBIKDİKEN****2018, 69 Sayfa****Jüri****Dr. Öğr. Üyesi Ali Osman ÇIBIKDİKEN****Prof. Dr. Sabri KOÇER****Dr. Öğr. Üyesi Ahmet Ercan TOPCU**

Zaman serisinde, bir olay sırasında alınan ölçümler ardışık zaman dilimi içinde uygun bir sırada düzenlenmiş olarak bulunur. Zaman serileri; iletişim, sağlık, hava tahmini ve finans gibi birçok alanda kullanılmaktadır. Finansal veri içeren zaman serilerinin analizinde; istatistiksel yaklaşım için ARIMA ve makine öğrenmesi yaklaşımı için Uzun-Kısa Süreli Hafıza derin öğrenme mimarisi kullanılmıştır. Tahmin modellemesi için adımlar önerilerek, bu adımlar hisse senedi ve Bitcoin fiyat tahminlemede uygulanmıştır. Tahminleme performans ölçülerine göre sonuçlar değerlendirilerek, Uzun-Kısa Süreli Hafıza derin öğrenme mimarisinin daha başarılı olduğu gözlemlenmiştir. Bu Uzun-Kısa Süreli Hafıza mimarisinin, zaman serilerinin tahmininde geleneksel istatistik yöntemlere göre daha uygun olduğunu göstermektedir. Ayrıca elde edilen derin öğrenme modeline dayanan bir anlık hisse senedi tahminleme web arayüzü geliştirilmiştir.

Anahtar Kelimeler: Zaman serisi, derin öğrenme, makine öğrenmesi, uzun-kısa süreli hafıza

ABSTRACT**MS THESIS****TIME SERIES PREDICTION IN REAL TIME USING DEEP LEARNING****Ebru Şeyma KARAKOYUN****THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
NECMETTIN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN INDUSTRIAL ENGINEERING****Advisor:** Assist. Prof. Dr. Ali Osman ÇIBIKDİKEN**2018, 69 Pages****Jury**

Assist. Prof. Dr. Ali Osman ÇIBIKDİKEN

Prof. Dr. Sabri KOÇER

Assist. Prof. Dr. Ahmet Ercan TOPCU

In the time series, the measurements taken during an event are arranged in a suitable order within the consecutive time slot. Time series are used in communication, health, weather forecasting and finance. ARIMA for the statistical approach and LSTM deep learning architecture for the machine learning approach have been used in the analysis of time series containing financial data. By suggesting steps for forecasting modeling, these steps have been implemented in stock and Bitcoin price prediction. It has been observed that the LSTM deep learning architecture is more successful by evaluating the results according to the estimation performance measures. This shows that the LSTM architecture is more appropriate in predicting time series than traditional statistical methods. In addition, an instant stock forecasting web interface based on the acquired deep learning model has been developed.

Keywords: Time series, prediction, deep learning, machine learning, long-short term memory

ÖNSÖZ

Tez çalışmam sırasında yaptığı katkılar ve desteklerden dolayı danışmanım Necmettin Erbakan Üniversitesi Bilgisayar Mühendisliği öğretim üyesi Dr. Öğr. Üyesi Ali Osman Çıbıkdiken'e ve web platformunun hazırlanmasında verdiği katkılardan dolayı Öğretim Görevlisi Mücahit Büyükyılmaz'a teşekkür ederim. Ayrıca yüksek lisans çalışmam sırasında gösterdiği her türlü sabır ve destek için aileme şükranlarımı sunuyorum.

Ebru Şeyma KARAKOYUN
KONYA-2018

İÇİNDEKİLER

KISALTMALAR	9
1. GİRİŞ	10
1.1. Çalışmanın Amacı	12
1.2. Çalışmanın Önemi	12
1.3. Tezin Yapısı	13
2. KAYNAK ARAŞTIRMASI	14
2.1. Zaman Serisi	14
2.1.1. Zaman Serisinin Bileşenleri	14
2.1.2. Durağan Zaman Serisi	15
2.1.3. Otokovaryans, Otokorelasyon ve Kısmi Otokorelasyon Fonksiyonları	17
2.1.4. Beyaz Gürültü	19
2.1.5. Geri Kaydırma Operatörü	19
2.1.6. Fark Alma	20
2.1.7. Zaman Serisinin İstatiksel Tahminleme Modelleri	21
2.1.7.1. Otoregresif Modeli	21
2.1.7.2. Hareketli Ortalama Modeli	22
2.1.7.3. Otoregresif Hareketli Ortalama Modeli	22
2.1.7.4. Otoregresif Bütünleşik Hareketli Ortalama Modeli	23
2.1.8. Birim Kök Testleri	23
2.1.8.1. Genişletilmiş Dickey-Fuller Testi	24
2.1.8.2. Kwiatkowski-Phillips-Schmidt-Shin Testi	24
2.1.9. Bilgi Kriterleri	24
2.1.10. Zaman Serisi Modelini Tahminleme	25
2.1.11. Tahmin Modelinin Performansını Değerlendirme	26
2.2. Derin Öğrenme	26
2.2.1. Temel Kavramlar	27
2.2.1.1. Nöron ve Yapay Sinir Ağı	27
2.2.1.2. Çok Katmanlı Yapay Sinir Ağı	31
2.2.5. Tekrarlamalı Sinir Ağı	32
2.2.6. Uzun-Kısa Süreli Hafıza	34
2.3. Zaman Serisi Modeli Tahminleme İle İlgili Yapılan Çalışmalar	39
3. MATERYAL VE YÖNTEM	46
3.1. Kullanılan Programlama Dili Kütüphaneleri ve Optimizasyon Algoritmaları	46

3.2. Veri Seti Kaynakları	47
3.4. Tahmin Modeli Geliştirme Adımları	48
3.3. Finansal Zaman Serisi İçin Tahmin Modelleri	49
3.3.1. Finansal Zaman Serisi İçin ARIMA Tahmin Modeli	49
3.3.2. Finansal Zaman Serisi İçin LSTM Tahmin Modeli	50
3.4. ARIMA ve LSTM Modellerinin Çalıştırılmasında Kullanılan Donanım	50
3.5. Hisse Senedi Fiyat Tahmini İçin LSTM Modeline Dayalı Anlık Tahmin Web Arayüz Uygulaması	51
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA	53
4.1. Hisse Senedi Fiyat Tahmin Modelleri	53
4.1.1. Hisse Senedi Fiyat Tahmini İçin ARIMA Modeli	53
4.1.2. Hisse Senedi Fiyat Tahmini İçin LSTM Modeli	57
4.2. Bitcoin Fiyat Tahmin Modelleri	59
4.2.1. Bitcoin Fiyat Tahmini İçin ARIMA Modeli	60
4.2.2. Bitcoin Fiyat Tahmini İçin LSTM Modeli	63
5. SONUÇLAR VE ÖNERİLER	65
5.1. Sonuçlar	65
5.2. Öneriler	65
KAYNAKLAR	67
ÖZGEÇMİŞ	69

KISALTMALAR

Kısaltmalar

- **ARMA** : Autoregressive Moving Average
- **ARIMA** : Autoregressive Integrated Moving Average
- **RNN** : Recurrent Neural Networks
- **LSTM** : Long-Short Term Memory
- **MLP** : Multi Layer Perceptron
- **GRNN** : Generalized Regression Neural Network
- **DL** : Deep Learning
- **RRL** : Recurrent Reinforcement Learning
- **SGD** : Stochastic Gradient Descent
- **ReLU** : Rectified Linear Units
- **GPU** : Graphics Processing Unit
- **CPU** : Central Processing Unit
- **ANN** : Artificial Neural Networks
- **LMS** : Least Mean Square
- **GD** : Gradient Descent
- **SAE** : Stacked Autoencoders
- **WT** : Wavelet Transforms
- **RBF** : Radial basis function neural network
- **ADNN** : Adaptive Neural Network
- **DS** : Directional Symmetry
- **WDS** : Weighted Directional Symmetry
- **BP** : Backpropagation
- **TDNN** : Time Delay Neural Networks
- **SVM** : Support Vector Machine

“Tahmin zordur, özellikle geleceğe aitse...”
Nils Bohr

1. GİRİŞ

Zaman serileri; iletişim, sağlık, hava tahmini ve finans gibi birçok alanda kullanılmaktadır. Bir zaman serisi, ardışık zamanlarda tipik olarak ölçülen bir dizi veri noktasını içerir. Zaman serisinde bir olay sırasında alınan ölçümler uygun bir kronolojik sırada düzenlenmiştir.

Tahmin; iş, sanayi, yönetim, ekonomi, çevre bilimleri, tıp, sosyal bilimler, politika ve finans gibi birçok alanı kapsayan önemli bir sorundur. Zaman serisi analizinin tahmin üzerindeki etkisi önemli şekilde ortaya çıkmaktadır. Örneğin, sürekli olarak günlük borsa fiyatlarına veya aylık işsizlik rakamlarına bakıldığında ekonomi alanında birçok tanıdık zaman serisi görülmektedir. Zaman serileri olarak oluşan veri için ileriye dönük tahmin yapıldığında borsa analizi, döviz analizi, FOREX analizi, GSYH analizi gibi finansal sistemler için yatırıma yönelik ihtiyaç olabilecek sonuçlar elde edilmektedir. Sosyal bilimciler, doğum veya okul kayıtları gibi popülasyon zaman serilerini takip ederler. Bir epidemiyolog, bir süre boyunca gözlenen grip vakalarının sayısı ile ilgilenebilir. Tıpta, zaman içinde kaydedilen kan basıncı ölçümleri, hipertansiyon tedavisinde kullanılan ilaçların değerlendirilmesi için yararlı olabilir. Beyin dalgası zaman serisi modellerinin fonksiyonel manyetik rezonans görüntülemesi, beynin çeşitli deney koşulları altında belirli uyaranlara nasıl tepki verdiğini incelemek için kullanılabilir.

Uygun tahmin yöntemleri büyük ölçüde hangi verilerin mevcut olduğuna bağlıdır. Geçmiş hakkında sayısal bilgiler mevcut ise, geçmiş bilgilere ait **kalıpların** (*pattern*) bazı yönlerinin geleceğe devam edeceğini varsayarsak **niceliksel tahminleme** (*quantitative forecasting*) uygulanabilir. Çoğu niceliksel tahmin problemi, zaman serileri verisini (zaman içinde düzenli aralıklarla toplanan) veya kesitsel veriyi (zaman içinde tek bir noktada toplanan) kullanır.

Niceliksel tahminleme için istatistiki metotlar ve yapay zeka teknikleri sıklıkla kullanılmaktadır:

- **ARIMA** (*autoregressive integrated moving average*) ve **üstel düzeltme** (*exponential smoothing*) modelleri, zaman serileri tahmini için en yaygın kullanılan temel tahminleme yaklaşımlarıdır. Üstel düzeltme modelleri, verilerdeki eğilim ve mevsimsel bir tanımlamaya dayanırken, ARIMA modelleri verilerdeki otokorelasyonları (*autocorrelation*) tanımlamayı amaçlamaktadır.
- Yapay zeka (*artificial intelligence*), insan aklının yeteneklerini simüle etmeye çalışan tekniklerden oluşur. Bu tekniklerin temeli öğrenme yetenekleri ile ilişkili beyin davranışını taklit eden yapay sinir ağlarıdır. **Yapay Sinir Ağları** (YSA), insan beynindeki nöronları temsil eden modeller yaratarak insan öğrenme yeteneklerini taklit etmeye çalışır. Çeşitli problemlerin çözümü için farklı yapay sinir ağı mimarileri kullanılarak **yapay öğrenme** ya da **makine öğrenmesi** (*machine learning*) adı verilen modeller geliştirilmiştir. **Derin öğrenme** (*deep learning*), son dönemin en yaygın yapay öğrenme tekniklerinden bir tanesidir. Çok katmanlı yapay sinir ağları derin öğrenmenin temelini oluşturmaktadır. Derin öğrenme, sadece derin doğrusal olmayan hiyerarşik özellikleri öğrenme ile ilişkili değildir. Aynı zamanda, sıralı bir veri içinde çok uzun doğrusal olmayan zaman bağımlılıklarını tespit ederek öğrenme için de kullanılır. Bu yönüyle zaman serilerinin tahmini için uygun bir yapı içerir.

Tez çalışmasında zaman serisi olarak ekonomik veriler temel alınarak istatistiksel modelleme ile derin öğrenme mimarisi modellerinin karşılaştırılması yapılmıştır.

İstatistiksel ya da yapay zeka teknikleri ile tahminleme için elde edilen modelin yeni bir veri geldiğinde modele eklenerek yeni tahminlerin anlık olarak yapılması karar alıcı için oldukça hızlı işlem yapabilme kabiliyeti getirmektedir. Bu amaçla çalışmada, anlık verilerin toplanması ve tahminleme için web tabanlı bir teknoloji ile bir arayüz

geliştirilmiştir. Böylece toplanan anlık verilerin **derin öğrenme** mimarisi ile oluşturulan modellerin çalıştırılacağı bir platforma aktarılması ve analizi sağlanacaktır.

1.1. Çalışmanın Amacı

Zaman serileri, günümüzde birçok alanda karşımıza çıkmaktadır. Özellikle ekonomideki her türlü veri zamana bağlı oluşmaktadır. Pek çok iktisadi model için kısa dönem tahmin yapılması önemli bir yer tutmaktadır. Zaman serileri ileriye dönük tahmin yapıldığında borsa analizi, Forex analizi, GSYH analizi gibi finansal sistemler için önemli sonuçlar elde edilebilir. Ekonomideki zaman serisi analizinde kullanılan verinin anlık olarak analizde kullanılması elde edilecek tahmin modelinde daha gerçekçi sonuçlar elde edilmesini sağlayabilir. Çalışmada ilk önce, özellikle ekonomik veri üzerinde derin öğrenme mimarisi ve istatistik analiz yöntemleri ile karşılaştırma yapılması amaçlanmıştır. Ayrıca anlık verilerin toplanması için web tabanlı teknolojilerle altyapı kurularak, bu verilerin derin öğrenme algoritması ile oluşturulan modellerin çalıştırılacağı bir platforma aktarılması ve analizinin sağlanması hedeflenmiştir.

1.2. Çalışmanın Önemi

Zaman serisi tahmini önceden bilinen olaylara göre gelecek ile ilgili olayları tahmin etmenin kavramsal bir modelidir. Zaman serisi yöntemleri ekonometrik model değerlendirmesi için faydalı olmaktadır. Bu amaçla, literatürde zaman serisinin ekonomik analizlerinde temel olarak kullanılan ARIMA modelleri ile çeşitli makine öğrenmesi algoritmalarının karşılaştırmaları bulunmaktadır. Derin öğrenme algoritmalarının anlık veriyi alarak analiz sırasında mevcut istatistiksel yaklaşımlara göre nasıl bir performans göstereceği bu çalışmada tespit edilmiştir.

Anlık verinin web teknolojileri ile toplanıp elde edilen modele göre kullanıcıya sonuç üretmesi, yatırımcı açısından karar almada zamanın çok kıymetli olduğu işlemler için önemli bir değerlendirme imkanı sağlayacaktır. Yatırımcı yeniden modelleme ihtiyacı duymadan yeni veriyi modeline ekleyerek tahminleme yaparak, yatırımını doğru şekilde yönetebilecektir.

1.3. Tezin Yapısı

Tez beş bölümden oluşmaktadır.

Birinci bölümde tezin konusunu oluşturan zaman serileri için derin öğrenme ve istatistiksel analiz yaklaşım problemi ele alınmıştır.

İkinci bölümde literatüre dayanarak zaman serisi ve derin öğrenme hakkında gerekli olan detaylı bilgiler verilmiştir. Ayrıca zaman serisi analizinde hem istatistiksel hem de makine öğrenmesi metotlarını kullanan daha önceki çalışmalar incelenmiştir.

Üçüncü bölümde kullanılacak tahminleme modeli adımları, istatistiksel yaklaşım ve derin öğrenme yaklaşımı için ayrı ayrı verilmiştir. Kullanılacak teknolojiler ve veri seti hakkında bilgi verilmiştir. Geliştirilen anlık tahminleme web arayüzü tanıtılmıştır.

Dördüncü bölümde zaman serisi olarak elde edilen finansal veri setleri üzerinde, uzun-kısa süreli hafıza derin öğrenme mimarisi ve ARIMA istatistiksel yaklaşım modelleri uygulanmıştır.

Beşinci bölümde sonuçlar değerlendirilerek gelecek çalışmalar hakkında bilgi verilmiştir.

2. KAYNAK ARAŞTIRMASI

Zaman serisi, yapay sinir ağları, derin öğrenme mimarisi, uzun-kısa süreli hafıza derin öğrenme mimarisi ve tahminleme ile ilgili bilgiler başlıklar halinde verilmiştir. Ayrıca zaman serisi tahmini ile ilgili yapılan başlıca çalışmalar da ayrı bir başlıkta incelenmiştir.

2.1. Zaman Serisi

Zaman serisi (*time series*), her biri belirli bir t zamanında kaydedilmekte olan bir gözlemler kümesidir. Genel olarak zaman serileri **ayrık** ve **sürekli** zaman serileri olarak ele alınır. Ayrık zamanlı seri gözlemlerin yapıldığı zamanlar için T_0 değerinin, ayrık bir dizi olduğu bir seridir. Örneğin, sabit zaman aralıklarında (saatlik, günlük, aylık, yıllık vb.) yapılan gözlemler ayrık zaman serisidir. Sürekli zaman zaman serileri ise gözlemlerin sürekli bir zaman aralığında (örneğin $T_0 = [0,1]$ aralığı) kaydedilerek elde edilir (Brockwell ve Davis, 2002).

Bir zaman serisi, ardışık zamanlarda tipik olarak ölçülen bir dizi veri noktası olarak düşünülebilir. Matematiksel olarak

$$\{y_t\}; t = 0, 1, 2, \dots \quad (1)$$

vektörlerin bir kümesi olarak tanımlanır. Burada t geçen süreyi temsil eder. y_t değişkeni rastgele değişken olarak kabul edilir. Bir zaman serisinde bir olay sırasında alınan ölçümler uygun bir kronolojik sırada düzenlenmiştir (Cochrane, 1997).

2.1.1. Zaman Serisinin Bileşenleri

Zaman serisi model sınıflarını açıklayabilmek için genel zaman serisi bileşenleri aşağıdaki şekilde tanımlanmıştır (Chatfield, 2000):

- **Mevsimsel değişim** (*Seasonal variation*): Genellikle yıllık bir periyottur ve yılın haftalık, aylık veya üç aylık gibi belirli zamanlarında benzer davranış modelleri gözlemlendiğinde ölçülen birçok seri için ortaya çıkar. Yaz aylarında her zaman yüksek olan dondurma satış modeli serisi buna bir örnektir. Bir zaman serisi

sadece yıllık olarak ölçülürse (örneğin yılda bir kez), mevsimsel değişimin mevcut olup olmadığını söylemek mümkün değildir.

- **Eğilim** (*Trend*): Bir seri en azından birkaç ardışık zaman periyodu boyunca sabit bir yukarı doğru büyüme veya aşağı doğru bir düşüş gösterdiğinde ortaya çıkar. Eğilim “ortalama düzeyde uzun vadeli değişim” olarak tanımlanabilir, ancak tam olarak tatmin edici bir matematiksel tanım yoktur. Trend algısı kısmen gözlemlenen serilerin uzunluğuna bağlı olarak ortaya çıkmaktadır.
- **Çevrimsel değişim** (*Cyclic variation*): Bir yıl dışındaki periyotlarda düzenli döngüsel değişimi içerir. Beş yıllık bir süre boyunca oluşan iş çevrimleri ve canlıların biyolojik davranışlarındaki günlük ritmi buna örnek verilebilir.
- **Düzensiz dalgalanmalar** (*Irregular fluctuations*): Eğilim, mevsimsellik ve diğer sistematik etkilerin ortadan kaldırılmasından sonra, "düzensiz dalgalanmalar" ifadesi sıklıkla "geride kalan" değişimleri tanımlamak için kullanılır. Bu nedenle, tamamen rastgele olabilirler ve tahmin edilemezler.

Bu dört bileşenin etkileri göz önüne alındığında, genellikle bir zaman serisi için **toplamsal** (*additive*) ve **çarpımsal** (*multiplicative*) olmak üzere iki farklı tipte model kullanılır. t anındaki bileşenler için y_t : gözlem, T_t : eğilim, S_t : mevsimsel, C_t : döngüsel ve I_t : düzensiz varyasyon olmak üzere bu modeller

$$\text{Toplamsal Model: } y_t = T_t + S_t + C_t + I_t \quad (2)$$

$$\text{Çarpımsal Model: } y_t = T_t \times S_t \times C_t \times I_t \quad (3)$$

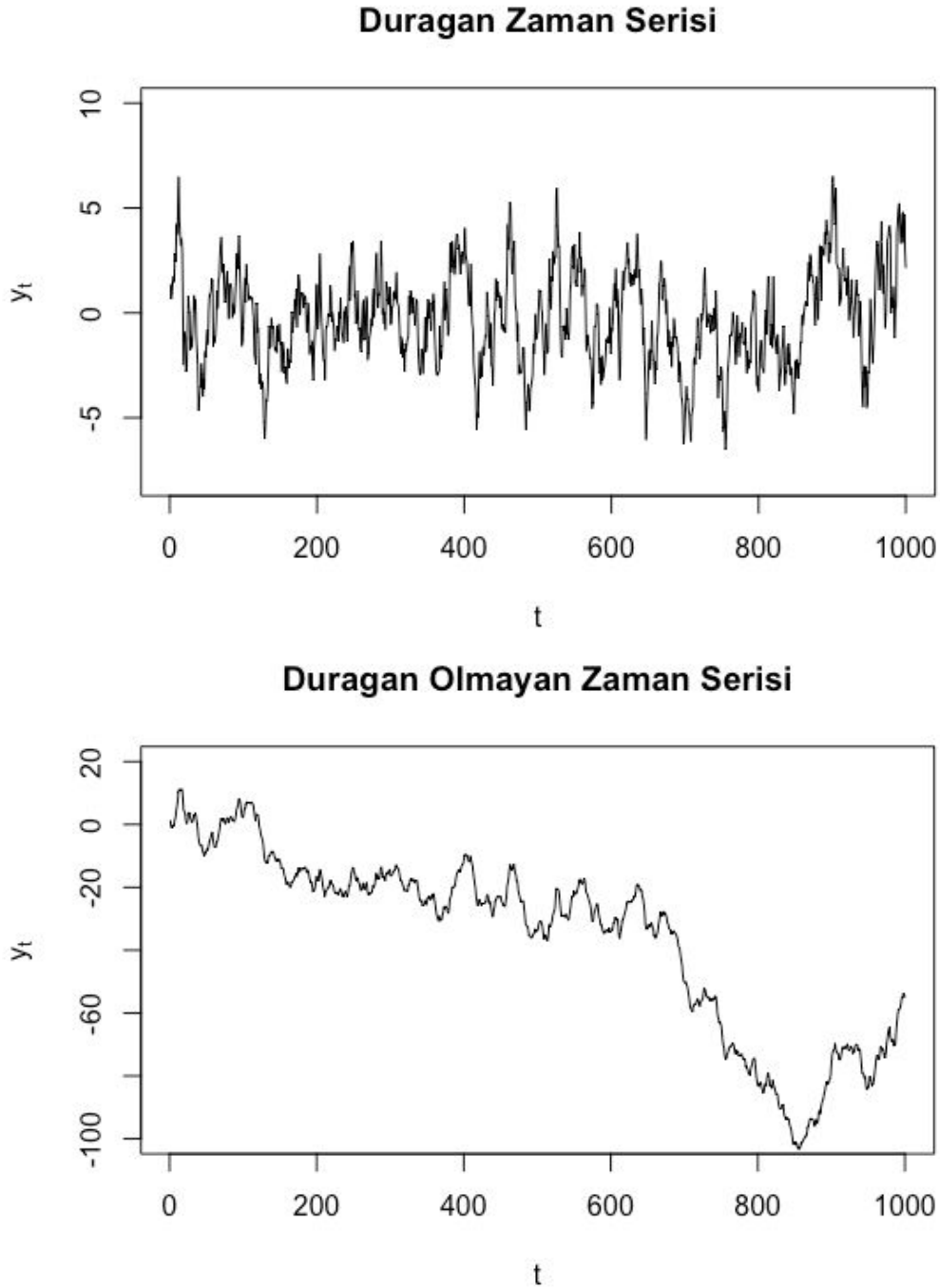
olarak ifade edilir (Folk, 2012; Adhikari ve Agrawal, 2013).

Çarpımsal model, bir zaman serisinin dört bileşeninin mutlaka bağımsız olmadığı ve birbirlerini etkileyebileceği varsayımına dayanır. Toplamsal modelde ise dört bileşen birbirinden bağımsız olabilir.

2.1.2. Durağan Zaman Serisi

Zaman serilerinin çok önemli bir türü **durağan (sabit)** (*stationary*) zaman serisidir. Bir zaman serisinin özellikleri, zaman kaynağındaki bir değişiklikten etkilenmezse, kesinlikle durağan olduğu söylenir. Yani, gözlemlerin $y_t, y_{t+1}, \dots, y_{t+n}$

'nin ortak olasılık dağılımı, gözlemlerin $y_{t+k}, y_{t+k+1}, \dots, y_{t+k+n}$ 'nin ortak olasılık dağılımı ile tamamen aynı ise zaman serisi kesinlikle sabittir. $n = 0$ olduğunda durağanlık varsayımı, y_t 'nin olasılık dağılımının tüm zaman periyotları için aynı olduğu ve $f(y)$ olarak yazılabileceği anlamına gelir (Wei, 2006; Montgomery ve ark., 2008). Şekil 2.1.'de de durağan ve durağan olmayan bir zaman serisi grafiği verilmiştir.



Şekil 2.1. Durağan ve durağan olmayan zaman serisi grafiği.

Durağanlık (sabitlik), verilerde bir tür istatistiksel **denge** (kararlılık) anlamına gelir. Zaman serileri, olağan şekilde

$$\mu_y = E(y) = \int_{-\infty}^{\infty} yf(y)dy \quad (4)$$

olarak tanımlanan **sabit ortalamaya** (*constant mean*) ve

$$\sigma_y^2 = Var(y) = \int_{-\infty}^{\infty} (y - \mu_y)^2 f(y)dy \quad (5)$$

olarak tanımlanan **sabit varyansa** (*constant variance*) sahiptir (Montgomery ve ark., 2008).

2.1.3. Otokovaryans, Otokorelasyon ve Kısmi Otokorelasyon Fonksiyonları

Bir zaman dizisi durağan ise bu, y_t ve y_{t+k} gözleminin ortak olasılık dağılımının, aynı k aralığı ile ayrılan t ve $t + k$ zaman periyodu için aynı olduğu anlamına gelir. Bu zaman serilerinin doğası hakkında yararlı bilgiler, aynı k aralığı ile ayrılmış olan tüm y_t, y_{t+k} veri çiftlerinin bir dağılım diyagramının çizilmesiyle elde edilebilir. k aralığı **gecikme** (*lag*) olarak adlandırılır (Montgomery ve ark., 2008).

y_t ve onun başka bir zaman periyodundaki değeri y_{t+k} arasındaki kovaryansına,

$$\gamma_k = Cov(y_t, y_{t+k}) = E[(y_t - \mu)(y_{t+k} - \mu)] \quad (6)$$

ile tanımlanan k gecikmesindeki **otokovaryans** (*autocovariance*) denir. $k = 0, 1, 2, \dots$ için elde edilen γ_k ise **otokovaryans fonksiyonu** (*autocovariance function*) olarak tanımlanır (Chatfield, 2000; Brockwell ve Davis, 2002; Wei, 2006; Montgomery ve ark., 2008).

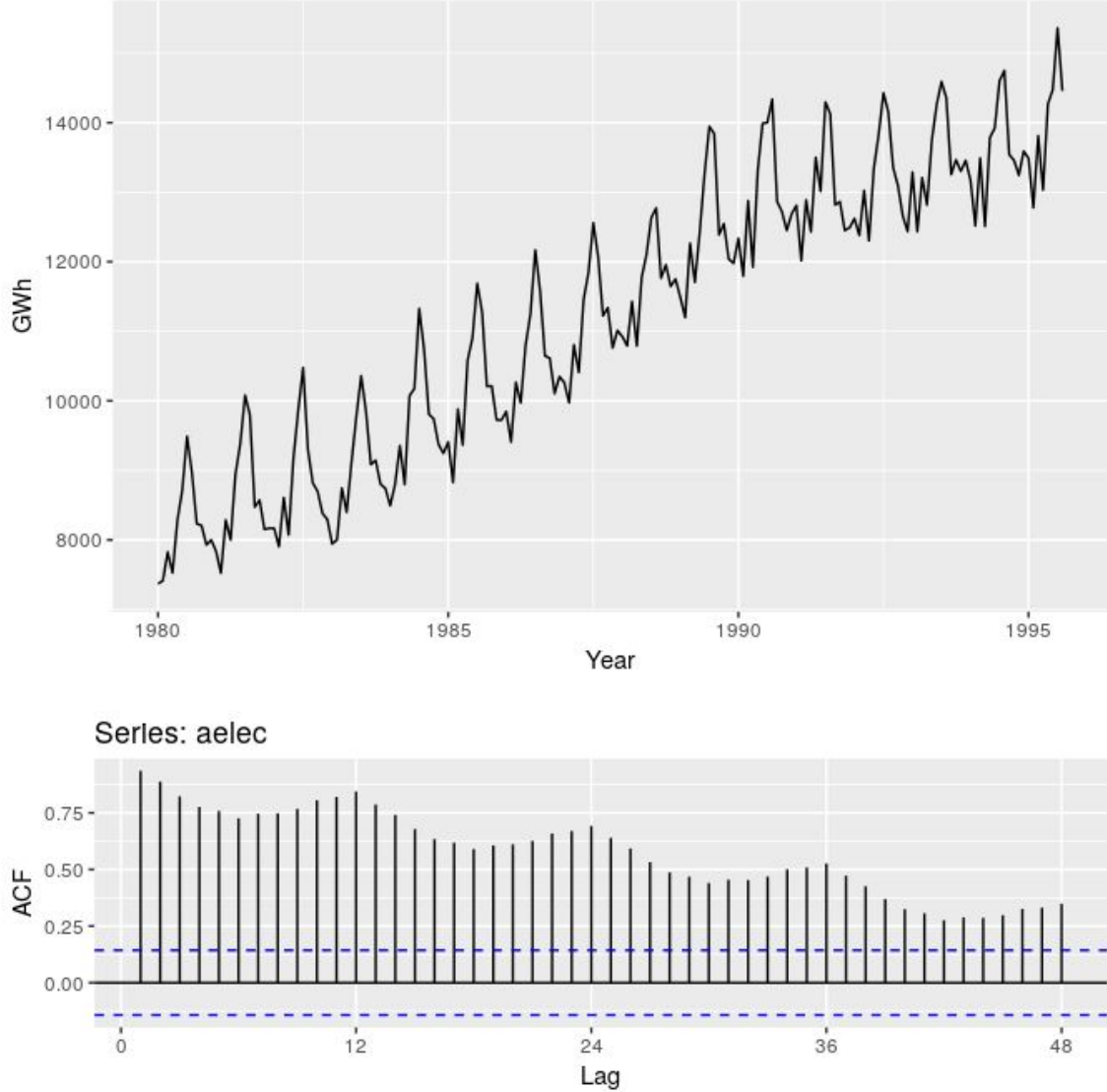
k gecikmesindeki **otokorelasyon katsayısı** (*autocorrelation coefficient*)

$$\rho_k = \frac{E[(y_t - \mu)(y_{t+k} - \mu)]}{\sqrt{E[(y_t - \mu)^2]E[(y_{t+k} - \mu)^2]}} = \frac{Cov(y_t, y_{t+k})}{Var(y_t)} = \frac{\gamma_k}{\gamma_0} \quad (7)$$

şeklinde tanımlanır. $k = 0, 1, 2, \dots$ için elde edilen ρ_k , **otokorelasyon fonksiyonu** (*autocorrelation function - ACF*) olarak adlandırılır (Chatfield, 2000; Brockwell ve Davis, 2002; Wei, 2006; Montgomery ve ark., 2008).

Şekil 2.2.'de Avustralya'nın 1980-1995 yılları arasındaki aylık elektrik ihtiyacını gösteren zaman serisi grafiği ve otokorelasyon fonksiyonu (ACF) grafiği

verilmiştir. Burada veri hem **mevsimsel** hem de **eğilim** bileşenine sahiptir. Otokorelasyon fonksiyonuna göre gecikmelerdeki "yavaş düşüş" eğilime işaret ederken, "taraklı" şekil serinin mevsimselliğe bağlı olduğunu göstermektedir (Hyndman ve Athanasopoulos, 2018).



Şekil 2.2. Avustralya'nın 1980-1995 yılları arasındaki aylık elektrik ihtiyacı

Belirli bir istatistiksel süreç, zamanın farklı noktalarındaki iki rastgele değişken arasındaki bağlantı ile ilgilenir. y_t ve y_{t+r} arasındaki ilişkiyi ölçmenin bir başka yolu da y_{t+r} , y_{t+r-1} arasında bulunan rastgele değişkenlerin doğrusal etkisini filtrelemek ve daha sonra dönüştürülmüş rastgele değişkenleri hesaplamaktır. Buna **kısmi otokorelasyon** (*partial autocorrelation*) denir. k . dereceden kısmi otokorelasyon

$$\eta_{kk} = \text{Corr}(y_t - \Gamma(y_t | y_{t+1}, \dots, y_{t+k-1}), y_{t+k} - \Gamma(y_{t+k} | y_{t+1}, \dots, y_{t+k-1})) \quad (8)$$

şeklinde hesaplanır. Burada $\Gamma(W | Z)$, Z üzerine W 'nin en iyi doğrusal projeksiyonu yani ortalama karesel hatayı en aza indirmeyi sağlayan ifadedir. Birçok k . dereceden hesaplanan η_{kk} lar **kısmi otokorelasyon fonksiyonu** (*partial autocorrelation function* - PACF) olarak tanımlanır (Brockwell ve Davis, 2002; Wei, 2006).

2.1.4. Beyaz Gürültü

Otokorelasyon göstermeyen zaman serilerine **beyaz gürültü** (*white noise*) serisi denir. Beyaz gürültü serileri için, her bir otokorelasyonun sifıra yakın olması beklenir. Ancak bazı rastgele varyasyonlar olduğundan sifıra eşit olmayacaktır. Beyaz gürültü serisi için, T zaman serisinin uzunluğu olmak üzere, otokorelasyon fonksiyonundaki sivri uçların %95'inin $\pm 2/\sqrt{T}$ içinde yer alması beklenir. Sınırların dışında bir veya daha fazla büyük sivri uç varsa veya sivri uçların %5'inden fazlası bu sınırların dışında ise, seri muhtemelen beyaz gürültü değildir.

Şekil 2.3.'de beyaz gürültüye sahip bir zaman serisi örneği verilmiştir. $T = 50$ için $\pm 2/\sqrt{50} = \pm 0.28$ olduğundan otokorelasyon fonksiyonu grafiğinde otokorelasyon katsayılarının tümü bu sınırlar içinde yer alır ve verilerin beyaz gürültü olduğunu doğrular (Hyndman ve Athanasopoulos, 2018).

2.1.5. Geri Kaydırma Operatörü

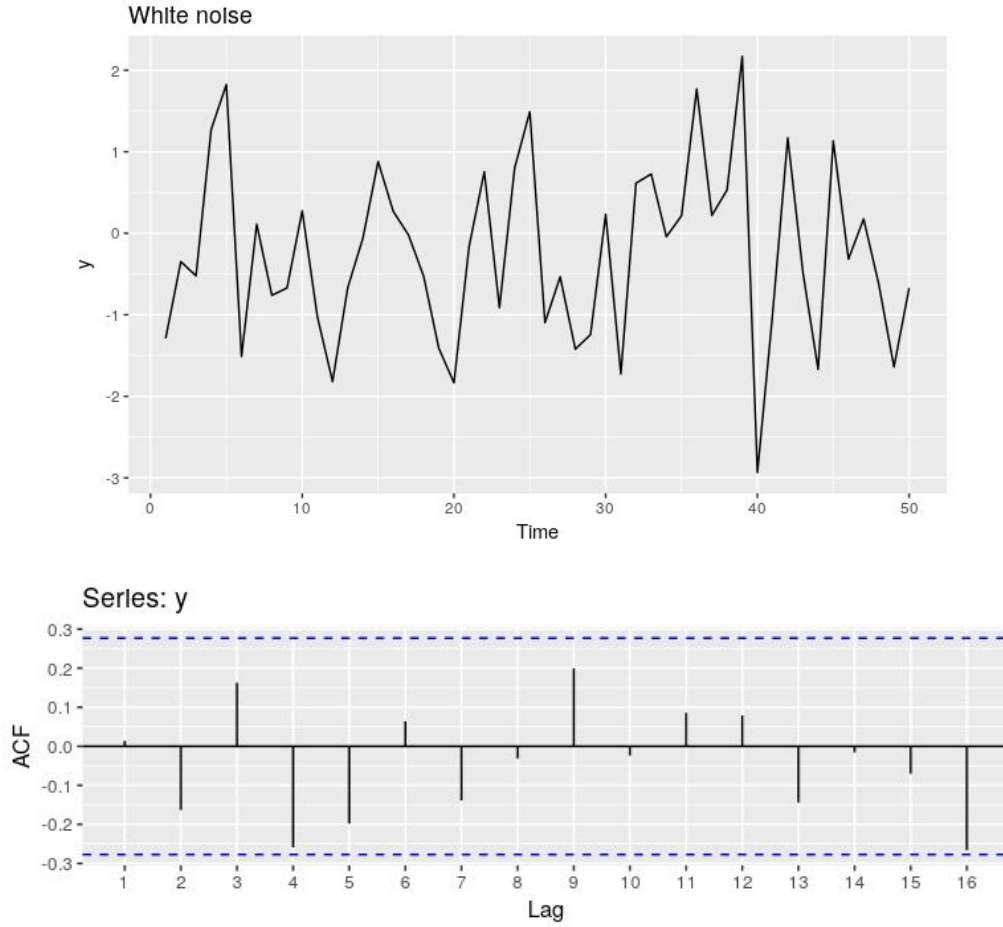
Geri kaydırma (*backshift*) **operatörü**, B ile gösterilmek üzere;

$$B y_t = y_{t-1}, B^2 y_t = B(B y_t) = B y_{t-1} = y_{t-2}, \dots$$

şeklinde devam eden bir notasyondur. Operatörün genel hali

$$B^k y_t = y_{t-k} \quad (9)$$

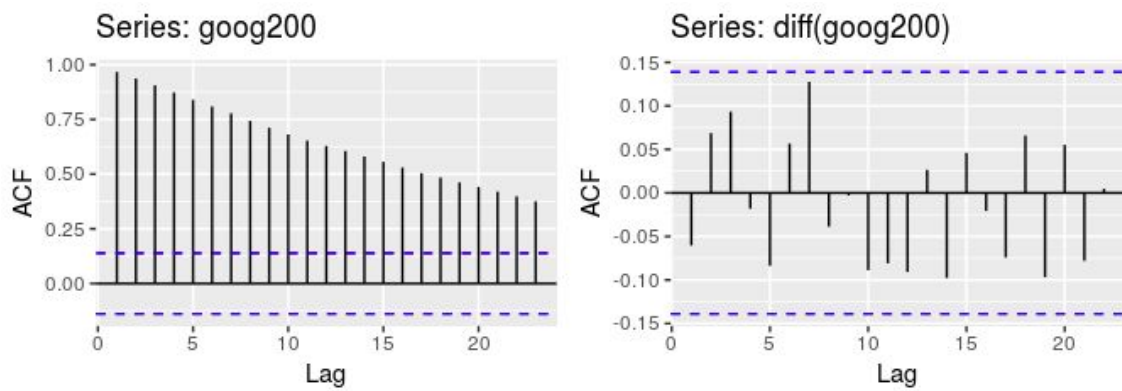
olarak ifade edilir (Chatfield, 2000; Brockwell ve Davis, 2002; Wei, 2006; Montgomery ve ark., 2008).



Şekil 2.3. Beyaz gürültüye sahip zaman serisi grafiği ve otokorelasyon fonksiyonu grafiği

2.1.6. Fark Alma

Bir zaman serisini durağan hale getirmenin bir yolu ardışık gözlemler arasındaki farkları hesaplamaktır. Buna **fark alma** (differencing) denir. Fark alma, bir zaman serisinin seviyesindeki değişiklikleri kaldırır ve bu nedenle trend ve mevsimsellikten arındırarak (veya azaltarak) bir zaman serisinin ortalamasını dengelenmesine yardımcı olabilir. Otokorelasyon fonksiyon grafiği, verilerin zaman çizelgesine bakmanın yanı sıra durağan olmayan zaman serilerini tanımlamak için de yararlıdır. Durağan bir zaman serisi için otokorelasyon fonksiyonu göreceli olarak hızlı bir şekilde sifıra düşecek ve durağan olmayan verilerin otokorelasyon fonksiyonu yavaşça azalacaktır. Şekil 2.4.'de Google hisselerinin 200 günlük fiyatlarından oluşan zaman serisinin otokorelasyon fonksiyon grafiği ve fark alınmış otokorelasyon fonksiyon grafiği gösterilmiştir (Hyndman ve Athanasopoulos, 2018).



Şekil 2.4. Google hisselerinin 200 günlük fiyatlarının otokorelasyon fonksiyon grafiği ve fark otokorelasyon fonksiyon grafikleri.

2.1.7. Zaman Serisinin İstatiksel Tahminleme Modelleri

Genelde zaman serisi verisi modelleri birçok biçime sahip olabilir ve farklı istatistiksel süreçlerle temsil edilebilir. Literatürde **Otoregresif** (*Autoregressive* - AR) ve **Hareketli Ortalama** (*Moving Average* - MA) modelleri olmak üzere yaygın olarak kullanılan iki doğrusal zaman serisi modeli vardır. Bu ikisini birleştiren, **Otoregresif Hareketli Ortalama** (*Autoregressive Moving Average* - ARMA) ve **Otoregresif Bütünleşik Hareketli Ortalama** (*Autoregressive Integrated Moving Average* - ARIMA) modelleri çalışmalarda yaygın olarak kullanılmaktadır. ARIMA modeli ve farklı varyasyonları ünlü Box-Jenkins ilkesine dayanmaktadır ve Box-Jenkins modelleri olarak da bilinir (Box ve Jenkins, 1970; Adhikari ve Agrawal, 2013).

2.1.7.1. Otoregresif Modeli

p dereceli otoregresif modeli kısaca **AR(p)** ile gösterilir ve matematiksel olarak;

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \omega_t; \quad \phi_1, \phi_2, \dots, \phi_p; \quad \phi_p \neq 0 \quad (10)$$

şeklinde ifade edilir (Chatfield, 2000; Brockwell ve Davis, 2002; Shumway ve Stoffer, 2006; Wei, 2006; Montgomery ve ark., 2008; Hyndman ve Athanasopoulos, 2018). Burada y_t durağandır. Aksi belirtilmedikçe, ω_t 'nin, ortalaması sıfır ve varyansı

σ_ω^2 olan bir Gaussian beyaz gürültü serisi olduğu var sayılacaktır. Eğer ortalama $\mu \neq 0$ ise (10) ifadesinde x_t yerine $x_t - \mu$ yazılırsa

$$y_t = \alpha + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \omega_t; \quad \alpha = \mu (1 - \phi_1 - \dots - \phi_p) \quad (11)$$

olur. (11) ile verilen ifade geri kaydırma operatörü yardımıyla

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) y_t = \omega_t; \quad \phi_p(B) y_t = \omega_t \quad (12)$$

olarak yazılabilir.

2.1.7.2. Hareketli Ortalama Modeli

q dereceli hareketli ortalama modeli kısaca **MA(q)** ile gösterilir ve matematiksel olarak;

$$y_t = \omega_t + \theta_1 \omega_{t-1} + \theta_2 \omega_{t-2} + \dots + \theta_q \omega_{t-q}; \quad \theta_1, \theta_2, \dots, \theta_q; \quad \theta_q \neq 0 \quad (13)$$

şeklinde ifade edilir (Chatfield, 2000; Brockwell ve Davis, 2002; Shumway ve Stoffer, 2006; Wei, 2006; Montgomery ve ark., 2008; Hyndman ve Athanasopoulos, 2018). ω_t 'nin bir Gaussian beyaz gürültü olduğu var sayılacaktır. (13) ile verilen ifade geri kaydırma operatörü yardımıyla

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q; \quad y_t = \theta_q(B) \omega_t \quad (14)$$

olarak yazılabilir.

2.1.7.3. Otoregresif Hareketli Ortalama Modeli

p dereceli otoregresif modelinin terimleri ve q dereceli hareketli ortalama modelinin terimleri birleşimi, Otoregresif Hareketli Ortalama Modeli'ni oluşturur ve **ARMA(p, q)** olarak ifade edilir (Chatfield, 2000; Brockwell ve Davis, 2002; Shumway ve Stoffer, 2006; Wei, 2006; Montgomery ve ark., 2008). Bu birleşim (12) ve (14) eşitlikleri ile

$$\phi_p(B) y_t = \theta_q(B) \omega_t \quad (15)$$

olarak yazılabilir.

2.1.7.4. Otoregresif Bütünleşik Hareketli Ortalama Modeli

ARMA modelleri sadece durağan zaman serileri için kullanılabilir. Ancak, uygulamada başta sosyo-ekonomik ve finans olmak üzere ilgili birçok zaman dizisi durağan olmayan davranış göstermektedir. Eğilim ve mevsimsel özellikler içeren zaman serileri de doğada durağan değildir. Bu yüzden ARMA modelleri, uygulamada sıklıkla karşılaşılan durağan olmayan zaman serilerini doğru bir şekilde tanımlamakta yetersiz kalmaktadır. Bu nedenle **Otoregresif Bütünleşik Hareketli Ortalama Modeli** (ARIMA) modeli, durağan olmayan durumların da dahil edildiği bir ARMA modelinin genelleştirilmesi olarak önerilmiştir (Adhikari ve Agrawal, 2013).

ARIMA modellerinde, veri noktalarının sonlu farkları alınarak durağan olmayan bir zaman dizisi durağan hale getirilir ve **ARIMA(p,d,q)** şeklinde ifade edilir (Chatfield, 2000; Brockwell ve Davis, 2002; Shumway ve Stoffer, 2006; Wei, 2006; Montgomery ve ark., 2008). Gecikme polinomları ile birlikte, (12) ve (14) kullanılarak matematiksel model;

$$\varphi_p(B) (1 - B)^d y_t = \theta_q(B) \omega_t \quad (16)$$

şeklinde yazılabilir. Eşitlik daha açık bir ifade ile yazılırsa **ARIMA(p,d,q)** nin

$$\begin{array}{ccc} \varphi_p(B) & (1 - B)^d y_t & \theta_q(B) \omega_t \\ \mathbf{AR}(p) & \mathbf{I}(d) & \mathbf{MA}(q) \end{array}$$

bileşenlerinden oluştuğu görülmektedir. Burada;

- p, d ve q sıfırdan büyük veya sıfıra eşit tamsayılardır. Sırasıyla modelin otoregresif, bütünleşik ve hareketli ortalama bileşenlerini temsil ederler.
- d tamsayısı fark alma derecesini belirler. Genellikle çoğu durumda $d = 1$ yeterlidir. $d = 0$ olduğunda, model ARMA(p,q) modeline indirgenmiş olur.

2.1.8. Birim Kök Testleri

Fark almaya ihtiyacı objektif olarak belirlemenin bir yolu, birim kök testi kullanmaktır. Bunlar, fark almanın gerekli olup olmadığını belirlemek için tasarlanmış durağanlık istatistiksel hipotez testleridir.

Farklı varsayımlara dayanan bir dizi farklı birim kök testleri mevcuttur.

2.1.8.1. Genişletilmiş Dickey-Fuller Testi

En popüler testlerden birisi Genişletilmiş Dickey-Fuller (*Augmented Dickey-Fuller* - ADF) testidir. Bu test için aşağıdaki regresyon modeli değerlendirilmiştir:

$$y'_t = \alpha + \beta_t + \phi y_{t-1} + \gamma_1 y'_{t-1} + \gamma_2 y'_{t-2} + \dots + \gamma_k y'_{t-k} \quad (17)$$

Burada y'_t ilk farkı alınan seriyi belirtir. $y'_t = y_t - y_{t-1}$ ve k regresyona dahil edilecek gecikme sayısıdır ve çoğunlukla 3 olarak belirlenmiştir (Hyndman ve Athanasopoulos, 2018).

2.1.8.2. Kwiatkowski-Phillips-Schmidt-Shin Testi

Bir diğer popüler birim kök testi **Kwiatkowski-Phillips-Schmidt-Shin** (KPSS) testidir. KPSS testi, hipotezi tersine çevirir. Bu nedenle boş hipotez, verilerin durağan olması halidir. Bu durumda, küçük p -değerleri (örneğin, 0.05'in altında), fark almanın gerekli olduğunu düşündürmektedir (Hyndman ve Athanasopoulos, 2018).

2.1.9. Bilgi Kriterleri

Bir ARIMA modelinin mertebesini belirlemek için **Akaike Bilgi Kriteri** (*Akaike's Information Criterion* - AIC) kullanılır ve şu şekilde ifade edilir:

$$AIC = -2 \log(L) + 2(p + q + 1) \quad (18)$$

Burada L , verilerin olasılığıdır (Chatfield, 2000; Brockwell ve Davis, 2002; Wei, 2006).

ARIMA modelleri için **Düzeltilmiş Akaike Bilgi Kriteri** (*Corrected Akaike's Information Criterion* - AICc) ise

$$AICc = AIC + \frac{2(p+q+1)(p+q+2)}{T-p-q-2} \quad (19)$$

olarak yazılır (Chatfield, 2000; Brockwell ve Davis, 2002; Wei, 2006).

Başka bir bilgi kriteri olan **Bayesian Bilgi Kriteri** (*Bayesian Information Criterion* - BIC):

$$BIC = AIC + [\log(T) - 2](p + q + 1) \quad (20)$$

şeklinde ifade edilir (Chatfield, 2000; Brockwell ve Davis, 2002; Wei, 2006).

İyi bir modelleme AIC, AICc veya BIC minimize ederek elde edilir. Tercihen AICc kullanılmaktadır.

Bu bilgi kriterlerinin, bir modelin farkının alma mertebesini seçmek için iyi kriterler olmamasına rağmen p ve q değerlerini seçmek için idealdir. Bunun nedeni, farkın, olasılığın hesaplandığı veriyi değiştirmesi, fark alma mertebesi farklı olan modeller arasındaki AIC değerlerinin kıyaslanamaz hale getirilmesidir. Dolayısıyla, d seçmek için başka bir yaklaşım kullanıp ve sonra p ve q seçmek için AICc kullanılabilir.

2.1.10. Zaman Serisi Modelini Tahminleme

Zaman serisi modellerinin çeşitlerine göre, verilerdeki tarihsel kalıpların tanımına en uygun modelin nasıl belirleneceği ve buna dayalı olarak doğru tahmin üretebilecek uygun bir modelin seçilmesi önemlidir. İstatistikçiler George Box ve Gwilym Jenkins, belirli bir zaman serisine en uygun ARIMA modelini oluşturmak için pratik bir yaklaşım geliştirmişlerdir. Ortaya koydukları kavramlar zaman serileri analizi ve tahmini alanında temel bir öneme sahiptir (Box ve Jenkins, 1970).

Bir ARIMA modelini bir dizi (mevsimsel olmayan) zaman serisi verisine uyarlarken, aşağıdaki adımlar yararlı bir genel yaklaşım sağlar (Hyndman ve Athanasopoulos, 2018) :

- Verileri çizin ve olağandışı gözlemleri tanımlayın.
- Varyansı dengelemek için gerekirse verileri bir Box-Cox dönüşümünü kullanarak dönüştürün.
- Veriler durağan değilse, veriler durağan olana kadar verilerin ilk farklarını alın.
- ACF / PACF'yi inceleyin: Bir ARIMA (p, d, 0) veya ARIMA (0, d, q) modeli uygun mu tespit etmeye çalışın.
- Seçtiğiniz model yada modelleri deneyin ve daha iyi bir model aramak için AICc'yi kullanın.
- Artıkların ACF'sini çizerek seçtiğiniz modeldeki kalıntıları kontrol edin. Beyaz gürültü gibi görünmüyorsa, değiştirilmiş bir model deneyin.
- Artıklar beyaz gürültü gibi görününce, tahminleri hesaplayın.

2.1.11. Tahmin Modelinin Performansını Değerlendirme

Adhikari ve Agrawal (2013) tarafından, tahmin modelinin performansını belirlemek için bazı ölçekler önerilmiştir. \hat{y}_t, y_t nin tahmini ve e_t hata olmak üzere;

$$e_t = y_t - \hat{y}_t \quad (21)$$

için ölçekler Çizelge 2.1.'de tanımlanmıştır.

Çizelge 2.1. Tahmin modeli performans değerlendirme ölçekleri

Ölçek	Formülü
Ortalama Mutlak Hata (<i>Mean Absolute Error - MAE</i>)	$MAE = \frac{1}{n} \sum_{t=1}^n e_t $
Ortalama Mutlak Yüzde Hatası (<i>Mean Absolute Percentage Error - MAPE</i>)	$MAPE = \frac{1}{n} \sum_{t=1}^n \left \frac{e_t}{y_t} \right \times 100$
Ortalama Yüzde Hatası (<i>Mean Percentage Error - MPE</i>)	$MPE = \frac{1}{n} \sum_{t=1}^n \left(\frac{e_t}{y_t} \right) \times 100$
Karekök Ortalama Hata (<i>Root Mean Squared Error - RMSE</i>)	$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2}$

2.2. Derin Öğrenme

Derin öğrenme (*deep learning - DL*) öğrenme sürecini yürütmek için yapay sinir ağlarının hiyerarşik seviyelerini kullanan bir yapay zeka yaklaşımıdır. Makine öğrenimi (*machine learning - ML*) metotları içerisinde yer alan derin öğrenme, yapılandırılmamış (*unstructured*) veya etiketlenmemiş (*unlabeled*) verilerden denetimsiz öğrenebilen (*unsupervised*) ağlara sahip makine öğrenme mimarilerini kullanır (Bengio, 2009).

Büyük miktarlarda etiketlenmemiş eğitim verilerinden özellik saptama yapabilen sistemler oluşturmak için çok katmanlı “derin” nöral ağların kullanan derin öğrenme, önemli derecede yatırım ve araştırmanın yapıldığı bir alandır. Endüstri ve akademik çevrelerdeki veri bilimciler görüntü sınıflandırma, video analizi, konuşma tanıma ve

doğal dil öğrenme süreci dahil olmak üzere çeşitli uygulamalarda çığır aşan gelişmeler elde etmek üzere makineyle öğrenmede GPU'ları (Graphics Processor Unit) kullanmaktadır. Derin öğrenme algoritmalarındaki hızlı gelişmenin ana sebeplerinden birisi GPU hesaplama gücünün kullanılmasıdır. Binlerce hesaplama çekirdeği ve tek başına çalıştırılan CPU'lar ile karşılaştırıldığında 10 ile 100 kat uygulama performansı sunan GPU'lar, veri bilimcilerin büyük verilerin işlenmesinde tercih ettikleri işlemci olmuştur.

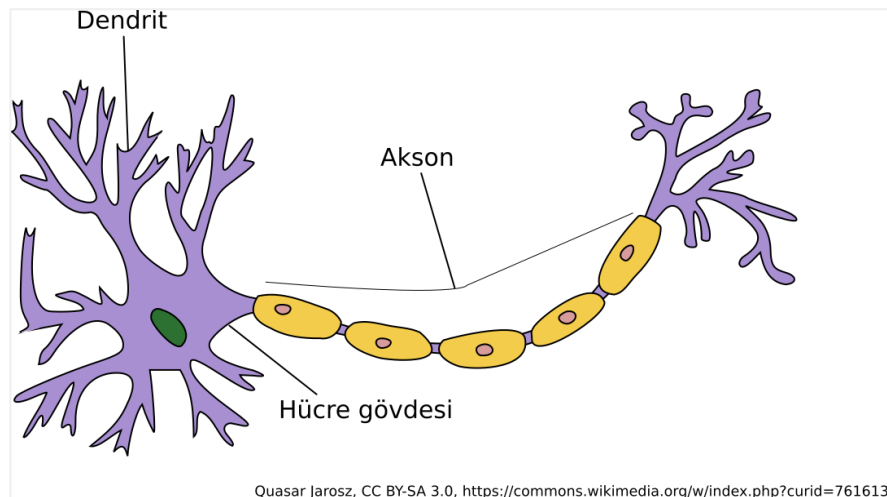
2.2.1. Temel Kavramlar

2.2.1.1. Nöron ve Yapay Sinir Ağı

Nöron (*neuron*), sinir hücresi olarak da bilinen, elektrik ve kimyasal sinyaller yoluyla bilgiyi alan, işleyen ve ileten elektriksel olarak uyarılabilen bir yapıdır. Beyin hücrelerindeki ağların temelini oluşturur (Seymour, 1997).

Nöron faaliyetlerini aşağıdaki bileşenler ile gerçekleştirir (Şekil 2.5.):

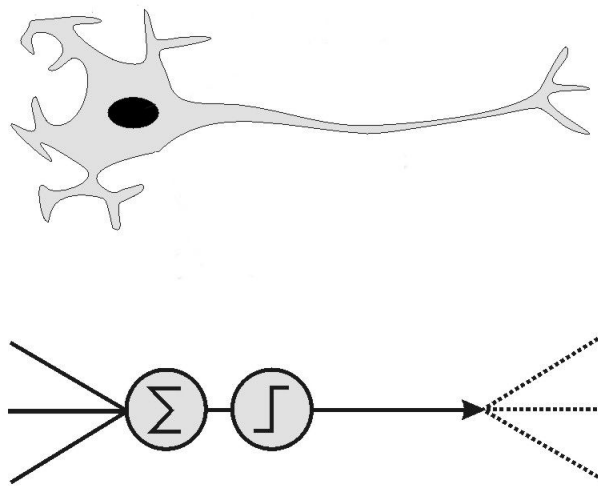
- **Dendrit** (*Dendrite*): Sinir hücrelerinden gelen sinyalin sinir hücre çekirdeğine iletilmesinden sorumludur.
- **Hücre Gövdesi** (*Cell body*): Dendrit üzerinden gelen sinyalleri toplayan merkezdir.
- **Akson** (*Axon*): Çekirdekte alınan bilginin sonraki sinir hücresine iletilmesini sağlar.



Şekil 2.5. Nöronu oluşturan bileşenler

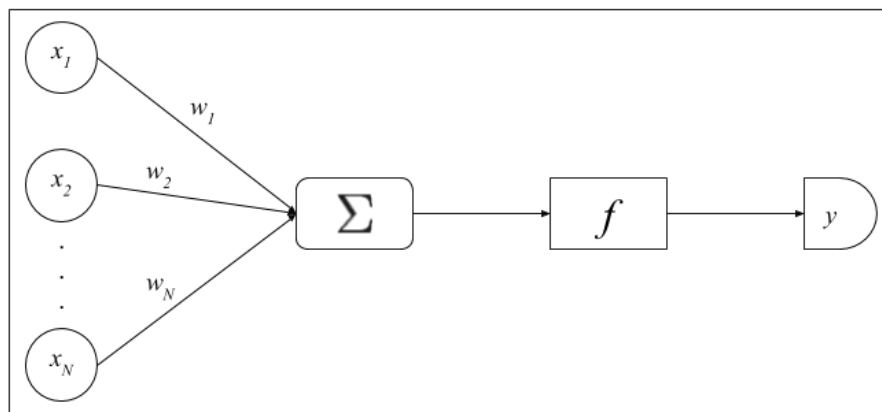
Nöronlar arasındaki sinyaller, **sinaps** (*synapses*) olarak adlandırılan özel bağlantılar aracılığıyla gerçekleştirilir.

McCulloch ve Pitts (1943) tarafından beynin öğrenme aktivitesindeki nöronun rolü benzetilerek matematiksel bir model tanıtılmıştır. Oluşturulan modelde yapay sinir (*artificial neuron*) ile biyolojik sinir arasında bir ilişki kurulmuştur. Şekil 2.6.'da bu benzetim şematik olarak gösterilmiştir (Winston, 1991). Böylece nöronun çalışma prensibi ile **öğrenmenin** (*learning*) yapay bir şekilde elde edilmesi sağlanmıştır.



Şekil 2.6. Biyolojik sinir ile yapay sinir arasındaki benzetim

Yapay sinir ağı (*artificial neural network*); birden fazla **giriş değeri** (*input*) ve bu giriş değerlerinin ağırlıklı toplamlarının belli eşik değeri ile **çıkış değeri** (*output*) oluşturması prensibi ile çalışan yapay sinirlerden meydana gelir (Şekil 2.7.).



Şekil 2.7. Yapay sinir ağı matematiksel modeli

Yapay sinir ağı yaklaşımı matematiksel olarak gösterildiğinde;

- N : Giriş değeri sayısı
- x : Giriş değerleri (*input*)
- w : Ağırlık (*weight*)
- b : Ön giriş (*bias*)
- f : Doğrusal olmayan aktivasyon fonksiyonu (*activation function*)
- y : Çıkış değeri (*output*)

olmak üzere,

$$y = f\left(\sum_{i=1}^N w_i x_i + b\right) \quad (22)$$

şeklinde ifade edilir (Haykin, 1999; Rosenblatt, 1962; ; Demuth ve ark. 2014). Bu aynı zamanda **tek katmanlı** (*single layer*) bir yapay sinir ağı olarak kabul edilebilir.

Bir yapay sinir ağında, çıktı değerini geçiren aktivasyon fonksiyonu için genellikle **tanh** ya da **sigmoid** fonksiyonları seçilmektedir. Bu fonksiyonlar ve grafikleri

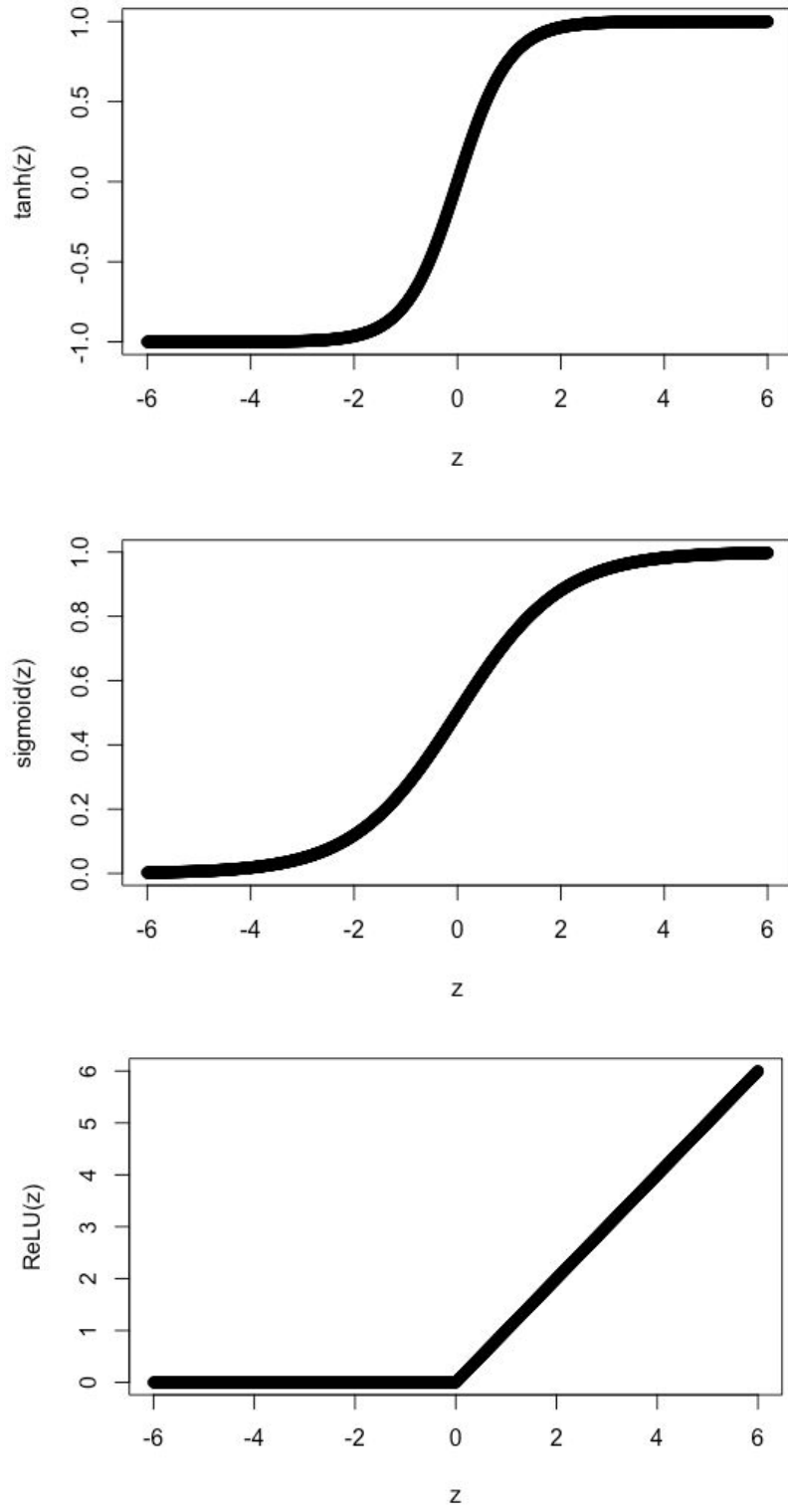
$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (23)$$

$$\text{sigmoid} = \sigma(z) = \frac{1}{1 + e^{-z}} \quad (24)$$

olarak ifade edilir (Haykin, 1999; ; Demuth ve ark. 2014). Ancak her iki doğrusal olmayan fonksiyon da yavaş çalışmaktadır. Bu nedenle genellikle **Düzeltilmiş Doğrusal Birim** (*Rectified Linear Unit - ReLU*) fonksiyonu tercih edilmektedir. ReLU fonksiyonu;

$$\text{ReLU}(z) = \max(0, z) \quad (25)$$

olarak gösterilir (Hinton ve ark., 2010). tanh, sigmoid ve ReLU fonksiyonlarının grafikleri Şekil 2.8.'de gösterilmiştir.



Şekil 2.8. tanh, sigmoid ve ReLU fonksiyon grafikleri

2.2.1.2. Çok Katmanlı Yapay Sinir Ağı

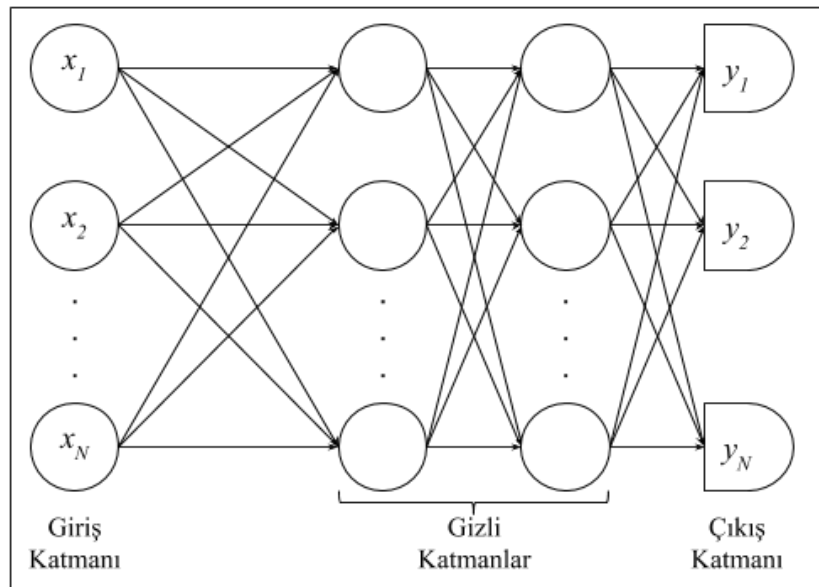
Yapay sinir ağında çıkış değeri olarak elde edilen sonuç yeniden giriş değeri bir sonraki katmana aktarılarak işleme alındığında kullanılan katman sayısı artar. İşleme alınarak yeniden ağa girmesini sağlayan aradaki katmanlara **gizli katman** (*hidden layer*) adı verilir. **Çok katmanlı yapay sinir ağı** (*MLP-Multi Layer Perceptron*) gizli katmanlar içeren ağlardan oluşur (Rumelhart ve ark., 1986).

Yeniden işleme alma öğrenmenin iyileştirilmesi amacıyla yapılır. Burada

- **İleri doğru işlem** (*feed-forward*)
- **Geriye yayımlı işlem** (*backpropagation*)

olmak üzere iki türlü iyileştirme işlemi vardır.

İleri doğru işlemde, bir önceki katmanda elde edilen çıktı değeri yeni ağırlıklarla birlikte bir sonraki katmana aktarılır ve tüm yapay sinir ağı boyunca işlemler yeniden yapılır (Şekil 2.9.).



Şekil 2.9. İleri doğru işlemlili çok katmanlı yapay sinir ağı modeli.

Geriye yayımlı işlemde, yapay sinir ağında hesaplanan çıktı sonucu olan \hat{y} , eldeki çıktı verisi olan y ile karşılaştırılır. Arada oluşan fark belirlenen bir hata fonksiyonu ile değerlendirilir. Burada; y ile \hat{y} arasındaki hatanın minimize edilerek

işleme alınması hedeflenmektedir. Hatanın minimize edilmesinde genellikle en küçük ortalama kareler (*least mean squares* - LMS) için **eğim iniş metodu** (*gradient descent*-GD) kullanılır. Metod başlangıçtaki w ağırlık ve b ön giriş değerlerinin seçimi problemini çözmek için uygundur. En küçük ortalama kareler yöntemi ile w ağırlıklar, y hedeflenen değer ve \hat{y} hesaplanan değer olmak üzere arasındaki hata;

$$J(w) = \frac{1}{2}(y - \hat{y})^2 \quad (26)$$

ile hesaplanır. Başlangıçta ağırlıklar rastgele değerlerle başlatılır ve η **öğrenme oranı** (*learning rate*) olmak üzere hata

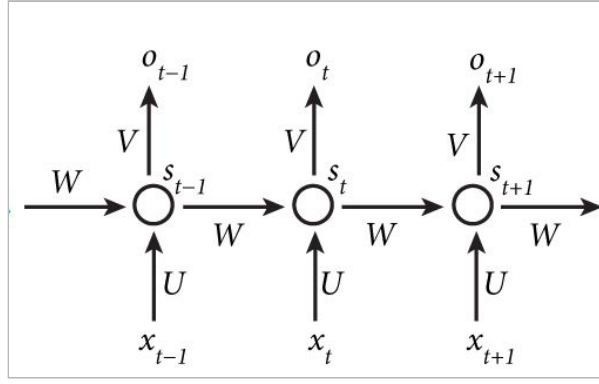
$$\Delta w = -\eta \frac{\partial J}{\partial w} \quad (27)$$

şeklinde azaltılarak ağda tekrar kullanılır. Böylece ağırlık değerlerinin hesaplanarak yeni bir ağırlık değeri oluşturulması ile hata azaltarak öğrenmeyi iyileştirmektedir. Bu şekilde yapılan işlem ile yeni ağırlık değerleri tekrar yapay sinir ağında çalıştırılır (Rumelhart ve ark., 1986; Duda ve ark., 2012; Demuth ve ark., 2014,).

2.2.5. Tekrarlamalı Sinir Ağı

Geleneksel sinir ağları, bütün girdi ve çıktıları birbirinden bağımsız şekilde varsayarak çalışır. Fakat bu yaklaşım bazı problemler için uygun değildir. **Tekrarlamalı Sinir Ağı** (*Recurrent Neural Networks* - RNN), giriş verisini birbirleri ile bağımlı olan bir giriş verisini **dizi halinde** kullanma yaklaşımı sonucu ortaya çıkmıştır. Örneğin, bir cümlede birkaç kelime sonra bir kelimeyi tahmin etmek için, önceki bütün kelime ilişkileri bilgisine ihtiyaç vardır. Bu yaklaşımda, bir dizi olarak giriş verisi alınıp ve aynı hedefi yerine getirmek üzere çıktıların bir önceki hesaplamaya bağlı olarak tekrarlı çalıştırılması esastır. Tekrarlamalı Sinir Ağı, sinir ağı boyunca adeta bilgiyi tutan bir hafıza gibi çalışır. Teoride dizi boyunca bu belleğin çalışması öngörülmüş olsa da pratik uygulamalarda çok fazla adımda hafıza özelliğini yitirmektedir (Haykin, 1999; Hu ve Balasubramaniam, 2008).

Tekrarlamalı Sinir Ağı, çalışma prensibi Şekil 2.10.'da verilmiştir.



Şekil 2.10. Tekrarlamalı Sinir Ağı çalışma şeması.

Burada;

- x_t : t anındaki girdi (*input*)
- s_t : t anındaki saklı durum (*hidden state*). Ağın hafıza kısmını oluşturur.
 s_t bir önceki hesaplanan saklı durum ve mevcut girdi değeri ile aşağıdaki gibi bulunur:

$$s_t = f(Ux_t + Ws_{t-1}) \quad (28)$$

Burada f lineer olmayan bir **tanh** ya da **ReLU** fonksiyonu olabilir.

- o_t : t anındaki çıktı (*output*). Bu çıktı:

$$o_t = \text{softmax}(Vs_t) \quad (29)$$

ile hesaplanabilir.

Bir sonraki katmana taşımak için, s_t bir önceki adımlardaki bilgiyi tutar. Adımların sonunda o_t , t anındaki bütün bilgiyi kullanarak elde edilen çıktıyı oluşturur. Diğer sinir ağlarının aksine, Tekrarlamalı Sinir Ağı (U , V , W) parametrelerini ağ boyunca kullanır. Burada;

- U : gizli katmanın ağırlık değerlerini
- V : çıktı katmanının ağırlık değerlerini
- W : gizli durum geçişinin ağırlık değerlerini

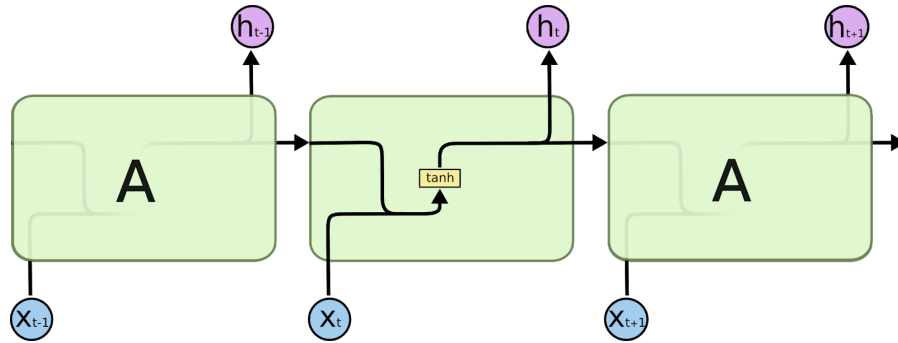
ifade etmektedir.

Bu yönüyle bir dizi halinde girdiler kullanıldığında, aynı hedefi ulaşmak için her bir adım birlikte çalışmış olur. Böylece öğrenme için gerekli çok sayıdaki parametre indirgenmiş olur.

2.2.6. Uzun-Kısa Süreli Hafıza

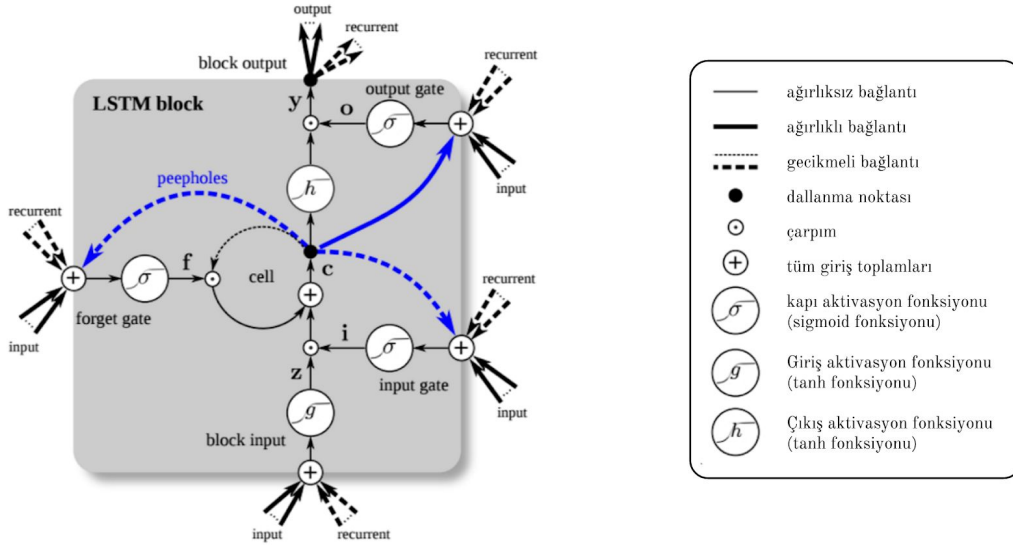
Uzun - Kısa Süreli Hafıza (*Long Short Term Memory* - LSTM) uzun süreli öğrenme kapasitesine sahip bir Tekrarlamalı Sinir Ağı mimarisidir. Hochreiter ve Schmidhuber (1997) tarafından ortaya atılmıştır. Uzun bir zaman periyodu içerisinde bilginin hatırlanması pratikte doğal bir davranıştır, öğrenme için gerekli olan bir uğraşı değildir.

Uzun - Kısa Süreli Hafıza mimarisi, Tekrarlamalı Sinir Ağı'nın uzun süreli bellek problemini çözmek için geliştirilmiştir. Tekrarlamalı Sinir Ağı, üzerinde bilgiyi tekrar eden modüllerden oluşan bir zincire sahiptir. Standart Tekrarlamalı Sinir Ağı'nda tekrar eden bu modüller tek bir *tanh* fonksiyonu kullanan katmandan oluşan basit bir yapıya sahiptir (Şekil 2.11.) (Olah, 2015).



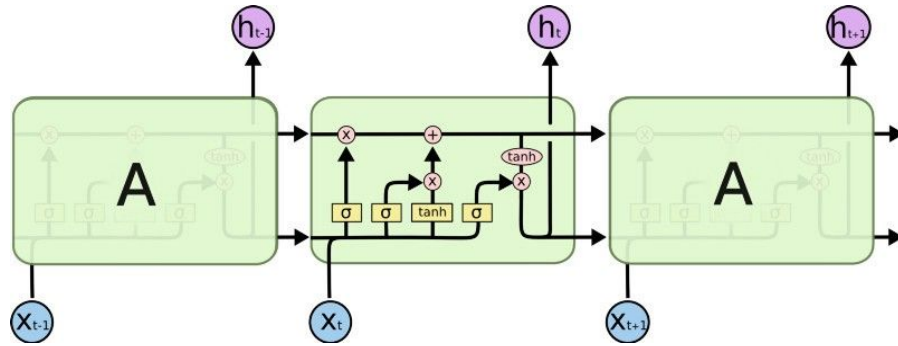
Şekil 2.11. Tekrarlamalı Sinir Ağı içerisinde tek katmandan oluşan ve tanh fonksiyonu kullanan bir blok

Uzun - Kısa Süreli Hafıza mimarisi ise farklı bir yapıya sahip tekrar eden bloklardan oluşan zincire benzeyen bir ağ şeklindedir (Şekil 2.12.) (Greff ve ark., 2015).



Şekil 2.12. Uzun - Kısa Süreli Hafıza blok çalışma şeması

Uzun - Kısa Süreli Hafıza bloklarında tek bir katman yerine, birbirleriyle özel bir şekilde iletişime sahip 4 tane katman bulunur (Şekil 2.13.) (Olah, 2015).



Şekil 2.13. Uzun - Kısa Süreli Hafıza bloğunda bulunan 4 katman (Katmanlar sarı ile belirtilen bölgelerle temsil edilmektedir)

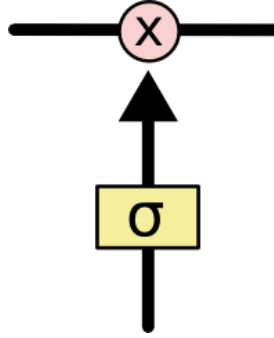
Blok içerisindeki 4 katman ve çalışma prensibi adım adım verilmiştir (Graves, 2014; Olah, 2015). Burada;

- \otimes : Hadamard çarpımını (vektörün eleman eleman çarpımı)
- \oplus : Elemanlar toplamını (vektörün eleman eleman toplamı)

göstermektedir.

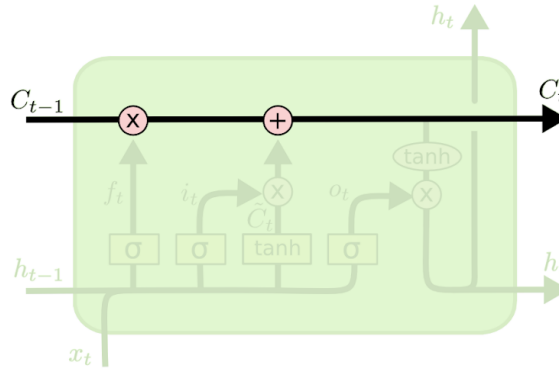
Uzun - Kısa Süreli Hafızada, **kapı** (*gate*) olarak adlandırılan, isteğe bağlı olarak bilgiyi almaya uygun yapıdır. Kapı, **sigmoid katmanı** (σ) ve \otimes işleminden oluşmaktadır (Şekil 2.14.). Kapı, **hücre durumuna** (*cell state*) bilgi çıkarma veya ekleme yeteneğine sahiptir.

Sigmoid katmanı (σ), her bir bileşen için izin durumunu belirleyen 0 ve 1 arasında bir değer üretir. Eğer değer 0 ise “hiçbir şeye izin verme”, 1 ise “herşeye izin ver” anlamındadır.



Şekil 2.14. Uzun - Kısa Süreli Hafıza kapı yapısı (sigmoid katmanı (σ) ile birlikte)

Yatay bir doğru boyunca çalışan ve bloktaki durumunu bir sonraki bloğa aktaran yapı adeta taşıyıcı bir hat gibidir. Bilgiyi değiştirmeden kolay bir şekilde doğrusal olarak zincir boyunca aktarır (Şekil 2.15.). C_{t-1} bir önceki bloktan taşınan bilgiyi, C_t bir sonraki bloğa taşınacak bilgiyi tutmaktadır.

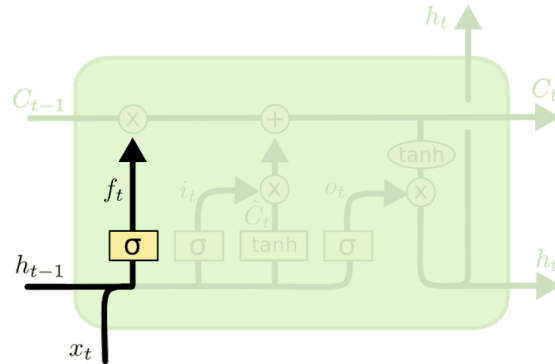


Şekil 2.15. Uzun - Kısa Süreli Hafıza hücre durumu.

Uzun - Kısa Süreli Hafıza’da ilk adım, hangi bilginin hücre durumundan çıkarılacağını belirlenmesidir. Bu karar **unutma kapısı katmanı** (*forget gate layer*) adı verilen sigmoid katmanı tarafından gerçekleştirilir (Şekil 2.16.). h_{t-1} bir önceki blok çıktı değeri, x_t giriş değeri ve b_f ön bilgi değeri olmak üzere; f_t unutma kapısı değeri

$$f_t = \sigma (W_{f,x} x_t + W_{f,h} h_{t-1} + b_f) \quad (30)$$

ile hesaplanır.



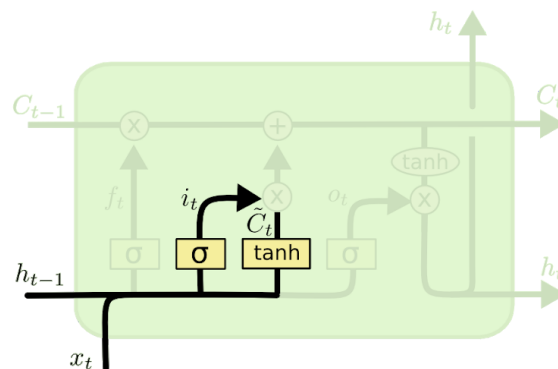
Şekil 2.16. Uzun - Kısa Süreli Hafıza unutma kapısı katmanı (forget gate layer)

Sonraki adım hangi yeni bilginin hücre durumu içinde saklanacağına karar vermektir. Bu iki aşamada gerçekleşir. Önce **giriş kapısı katmanı** (*input gate layer*) olarak adlandırılan sigmoid katmanı hangi değerlerin yenileneceğine karar verir. Daha sonra **tanh** katmanı yeni \hat{C}_t değerini oluşturur. Üretilen bu iki değer birleştirilerek durum yenilenmiş olur (Şekil 2.17.). i_t giriş kapısı değeri, \hat{C}_t ara değeri, b_i giriş ön bilgi ve b_c hücre ön bilgi değeri olmak üzere;

$$i_t = \sigma (W_{i,x} x_t + W_{i,h} h_{t-1} + b_i) \quad (31)$$

$$\hat{C}_t = \tanh(W_{c,x} x_t + W_{c,h} h_{t-1} + b_c) \quad (32)$$

ile hesaplanır.

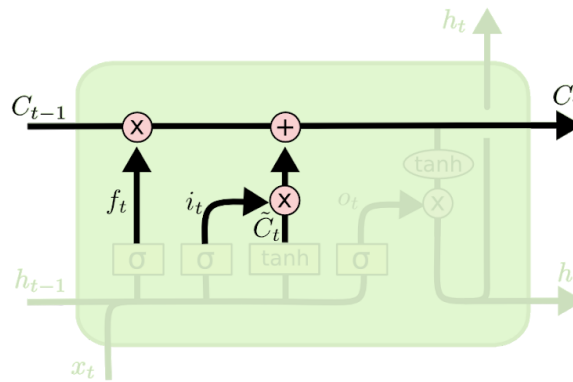


Şekil 2.17. Uzun - Kısa Süreli Hafıza'da yeni bilginin hücre durumunda saklanması

Bu aşamada; C_{t-1} eski hücre durumu ile \hat{C}_t yeni hücre ara durumu değerleri yenilenmiş olur. C_{t-1} eski durum değeri ile f_t unutmaya kapısı değeri \otimes işleminden geçirilir. Böylece önceki adımda unutulmasına karar verdiğimiz bilgi unutulur. i_t giriş kapısı değeri ile \hat{C}_t yeni hücre ara durum değeri \otimes işleminden geçirilir. Sonra bu iki değer toplanır. Yeni elde edilen bu değer, her bir durum değerinin ne kadar yenilenmesine karar verilen ve sonraki bloğa aktarılmak üzere hafızaya alınan değer olur (Şekil 2.18.). Bu değer;

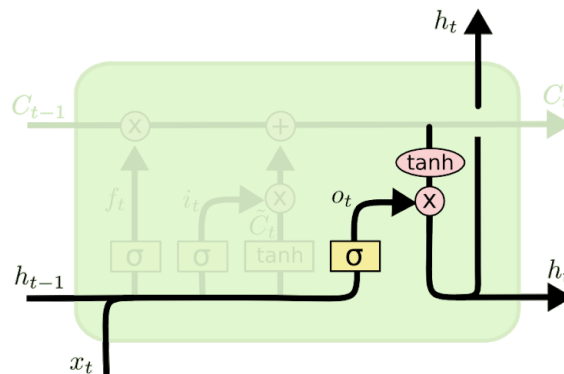
$$C_t = f_t \otimes C_{t-1} + i_t \otimes \hat{C}_t \quad (33)$$

ile bulunur.



Şekil 2.18. Uzun - Kısa Süreli Hafıza'da bilginin unutulması ve yeni bilginin aktarılması.

Son aşamada çıktının ne olacağına karar verilmesi gerekir. Çıktı kapı durumuna bağlı olarak belirlenir, ancak bu değer filtrelenmiş bir değerdir. Çıktı için hücre durumundaki değere sigmoid katmanında **tanh** fonksiyonu uygulanır ve sigmoid kapı çıktı değeri ile \otimes işlemi uygulanır (değerin -1 ila 1 arasında bir değer alması sağlanır). Böylece çıktının karar verilen kısmı için h_t değeri elde edilmiş olur (Şekil 2.19).



Şekil 2.19. Uzun - Kısa Süreli Hafıza'da çıktı değerinin oluşturulması.

o_t ve h_t çıktı değeri, b_o ön bilgi çıktı değeri olmak üzere;

$$o_t = \sigma (W_{o,x} x_t + W_{o,h} h_{t-1} + b_o) \quad (34)$$

$$h_t = o_t \otimes \tanh(C_t) \quad (35)$$

ile hesaplanır.

Tüm hesaplamalar boyunca kullanılan, $W_{f,x}$, $W_{f,h}$, $W_{c,x}$, $W_{c,h}$, $W_{i,x}$, $W_{i,h}$, $W_{o,x}$, $W_{o,h}$ değerleri ilgili kapıların ağırlık değerlerine karşılık gelmektedir.

2.3. Zaman Serisi Modeli Tahminleme İle İlgili Yapılan Çalışmalar

Kreesuradej (1993) tez çalışmasında geçmişten gelen bilgilerini kullanarak zaman serilerinin gelecekteki değerlerini tahmin etmek için Zaman Gecikmeli Sinir Ağlarını (TDNN) kullanmışlardır. Sınırlı kayıt uzunluğu verileri için, uyum sorunu ciddi bir şekilde ağ performansını düşürmesi nedeniyle karşılıklı çapraz doğrulama yöntemi kullanılarak sorun giderilmiştir.

Björklund ve Uhlin (2017) finansal zaman serilerinin mevcut özelliklerinin gözden geçirmiş ve istatistiksel özelliklerini yapay sinir ağında kullanarak, potansiyel anormalliklerin vurgulandığı etkin piyasa hipotezine yönelik bir çalışmaya yapmışlardır. Teorik incelemeye dayanarak, makine öğrenimi ve finansman arasındaki disiplinlerarası bir yaklaşım, çeyreklik bazda beklenen zaman serisini tahmin eden bir modele dönüşmüştür. İlgili bir bağlamda gelecekteki getiri tahminlerinin kullanılmasını değerlendirmek için stok optimizasyon modelinden yararlanılarak hisse senedi ve emtia gibi farklı varlık sınıflarından döviz kurları ve endeksleri kullanmışlardır.

Tan (2009) yapay ağ, durağan dalgacık dönüşümü ve istatistiksel zaman serileri analiz tekniklerine dayanan geliştirilmiş bir döviz kuru tahmin modeli önermiştir. Yeni modelin performansını, yapay sinir ağı tahmin modeli, dalgacık/dalgacık-paket-dengi temelli tahmin modelleri ve önceki çalışmalarda kullanılan yöntemlerle karşılaştırılmıştır. Deneysel sonuçlar, önerilen yöntemin mevcut yaklaşımları önemli ölçüde geride bıraktığını göstermektedir.

Mitchell (1995) bilgisayar bilimlerindeki en yeni ve en hızlı büyüyen araştırma alanlarından birisi olan "Bilgi keşfi (*knowledge discovery*)" üzerinde çalışmıştır. Büyük veri tabanlarından anlamlı bilgi bulmak ve çıkarmak için, makine öğrenimi ve

veritabanı teknolojisi teknikleri birleştirilmiştir. En güncel bilgi keşif sistemleri, benzerlik temelli makine öğrenme yöntemlerini — genel olarak bu tür veriye çok uygun olmayan “örneklerden öğrenme” yi kullanır. Zaman serisi analiz teknikleri, sinyal işleme ve konuşma tanıma gibi dizi tanımlama uygulamalarında yaygın olarak kullanılmaktadır. Ancak çoğunlukla bilgi keşif görevleri için dikkate alınmamıştır. Gerçek dünyadaki zaman serisi verisinde bilgiyi keşfetmek için yeni yöntemler sunulmuştur. Önerilen yöntemler geniş bir tarımsal veri tabanına uygulanmış ve elde edilen deneysel sonuçlar sunulmuş ve analiz edilmiştir.

Grachev (2017) çalışmasında hisse senedi piyasası için ARIMA, GARCH ve ARMA-GARCH gibi zaman serisi modellerinin etkinlik ve sınırlamalarını incelemiştir. Çalışma ilk olarak finansal verilerin benzersiz özelliklerini, özellikle de geri dönüş dağılımındaki dalgalanma kümelenme ve aşırılıkları değerlendirmekte ve finansal ekonomide otoregresif bütünleşik hareketli ortalama (ARIMA) modellerinin kullanılmasının sınırlamalarını ele almaktadır. İkinci olarak, ARMA-GARCH modellerinin hem koşullu araçların hem de geri dönüşlerin koşullu varyansının tahmin edilmesine yönelik uygulamasını incelenmiştir. Son olarak, AIC, BIC, SIC ve HQIC gibi standart model seçim kriterleri kullanılarak çeşitli aday ARMA-GARCH modellerinin tahmin performansı ele alınmıştır. MSCI Dünya Endeksinin fazla getiri ve Fama-French 3 faktörlü fazla getiri modelleri için, bir ARMA (1,0) + GARCH (1,1)'in her iki grupta da aynı dönemde en iyi sonuçların elde edildiği görülmüştür.

Sewell (2017) makine öğrenimi uygulanması yoluyla finansal zaman serileri analizinde teknolojinin son durumunu değerlendirdiği bir çalışma yapmıştır. Çalışmasını 3 ana başlık altında değerlendirerek; finansal zaman serilerinin karakterizasyonuna, modellemesine ve tahminine ilişkin çalışmaların sonuçları bir araya getirmiştir. Lineer ve non-lineer ilişkileri tespit eden testleri kullanarak DJIA ve döviz logu geri dönüşlerinin analizinde, önceden belgelenmemiş birkaç anomali tespit etmiştir. Günlük USD/DEM, USD/JPY, GBP/USD, USD/CHF ve GBP/CHF döviz bilgileri ile uygulamasını yaparak, her biri için şaşırtıcı bir şekilde azalan getiri dizileri sergilediğini göstermiştir.

Tay ve Cao (2001) finansal zaman dizisi tahmininde destek vektör makinesi (SVM) sinir ağı tekniği uygulaması ele almışlardır. SVM'nin finansal zaman serileri tahmininde çok katmanlı geri yayımlı (BP) sinir ağı ile karşılaştırılarak fizibilitesini

incelemektir. Veri seti olarak Chicago Ticaret Merkezi'nden derlenen beş adet gerçek vadeli işlem sözleşmesi kullanılmıştır. Çalışma, SVM'nin normalize edilmiş ortalama kare hatası (NMSE), ortalama mutlak hata (MAE), yönlü simetri (DS) ve ağırlıklı yönlü simetri (WDS) kriterlerine dayanarak BP sinir ağından daha iyi performans gösterdiğini göstermektedir. SVM'lerin serbest parametrelerini seçmek için yapılandırılmış bir yol olmadığı için, bu çalışmada serbest parametrelere göre performanstaki değişkenlik araştırılmıştır. Çalışma sonuçlarının analizi, finansal zaman serilerini tahmin etmek için SVM'lerin uygulanmasının avantajlı olduğunu kanıtlamıştır.

Jansze (2013) Barselona metro istasyonunun enerji tüketimini en aza indirmek için gerçekleştirilen SEAM4US projesinin bir parçası olarak bir çalışma yapmıştır. Enerjinin en aza indirilmesi, bir zaman dizisi tahminine dayalı enerji kullanan sistemlerin yönetimine dayandırılmıştır. Popüler Fourier dönüşümünü trend belirleme ile birlikte kullanılmış, farklı veri setlerini test ederek ve iyi bilinen tekniklerle karşılaştırarak yöntemini doğrulamıştır. Kullanılan tekniğin SEAM4US projesi için ve muhtemelen çok sayıda zaman serisi tahmini için çok kullanışlı olduğu sonucuna ulaşmıştır.

Collantes-Duarte ve Rivas-Echeverria (2001) çalışmada ARIMA, Sinir Ağları ve Neo Fuzzy Neurons kullanarak zaman serisi tahminlemesi yapmışlardır. Bir uygulama vakası kullanılarak üç zaman serisi tahmin teknikleri karşılaştırılmıştır. Elde edilen performansın farklı yöntemlerle karşılaştırılması için, istatistik ve tahmin aşamalarında, istatistiksel değerlendirmeyi öngörmede yaygın olarak kullanılan karşılaştırma kriterlerini kullanmışlardır. Çalışma ile Neo Fuzzy Neurons metodunun tahmin etme yeteneklerinin ARIMA modelinden bile daha iyi olduğunu tespit etmişlerdir.

Moody ve Saffell (1999) takviyeli öğrenme yoluyla finansal amaç fonksiyonlarını optimize ederek ticaret sistemlerini eğitmeyi öneren bir çalışma yapmışlardır. Kar ve Sharpe oranını dikkate alarak öğrenme için yeni bir Sharpe oranı önermişlerdir. Q-Learning'i Tekrarlayan Takviyeli Öğrenme (RRL) algoritmasına eklemek önceki çalışmalarını temel almışlardır. Aylık S&P 500 Hisse Senedi Endeksi'nde, 1970'den 1994'e kadar olan 25 yıllık dönem için öngörülebilirliğin varlığını gösteren bir simülasyon sonuçları ve yatırımcının yapısına ekonomik bakış açısı sağlayan bir duyarlılık analizi sunmuşlardır.

Heaton ve ark. (2016) çalışmalarında finansal tahmin ve sınıflandırma problemleri için derin öğrenme hiyerarşik modellerinin kullanımını araştırmışlardır. Finansal tahmin sorunları (menkul kıymetleri geliştirme ve fiyatlandırma, portföy oluşturma ve risk yönetimi gibi uygulamalar vb.) çoğu zaman, tam bir ekonomik modelde belirtilmesi zor veya imkansız olan karmaşık veri etkileşimlerine sahip büyük veri kümelerini içerir. Bu problemlere derin öğrenme yöntemlerini uygulayarak finasta standart yöntemlerden daha faydalı sonuçlar üretebildiğini göstermişlerdir. Özellikle derin öğrenmenin, en azından halihazırda mevcut herhangi bir finansal ekonomi teorisinde görünmez olan verilerdeki etkileşimleri tespit etmede faydalı olduğunu ortaya koymuşlardır.

Navon ve Keller (2017) finansal zaman serilerine uygulanan, zaman serileri tahmini için, veri odaklı, uçtan uca Derin Öğrenme yaklaşımı sunan bir çalışma gerçekleştirmişlerdir. Çalışmalarında NYSE veya NASDAQ'daki stokların ve ETF'lerin zamansal eğilimlerini tahmin etmek için bir Derin Öğrenme şeması oluşturmuşlardır. Ham finansal veri girdilerine uygulanan bir sinir ağına (NN) dayanan ve stokların ve ETF'lerin zamansal eğilimlerini öngörmek için eğitildiği bir yaklaşım ortaya koymuşlardır. Komisyona dayalı ticareti ele almak için, NN'nin olasılıksal çıktılarını kullanan ve ortalama getiriyi optimize eden bir yatırım stratejisi elde ederek, finansal piyasa eğilimlerinin istatistiksel olarak anlamlı ve doğru tahminleri sağladığını göstermişlerdir.

Moreno ve ark. (2011) yaptıkları çalışmada zaman serileri tahmininde yararlı olduğu kanıtlanmış Yapay Sinir Ağları (YSA) ana modellerinin bir tanımını, karşılaştırmasını ve ayrıca bu tip bir görevde YSA'nın pratik uygulaması için standart bir adımlar listesini sunmuşlardır. Çok katmanlı algılayıcı (MLP), radyal tabanlı fonksiyon (RBF), genelleştirilmiş regresyon sinir ağı (GRNN) ve tekrarlayan sinir ağı (RNN) modelleri analiz edilmiştir. Bu amaçla, 244 zaman noktasından oluşan bir zaman dizisi kullanmışlardır. Karşılaştırmalı olarak yaptıkları çalışmada, analiz edilen dört nöral ağ modeli tarafından yapılan hatanın% 10'dan daha az olduğunu göstermişlerdir. En iyi performansa sahip model RBF, ardından RNN ve MLP olarak görülmektedir. GRNN modeli en kötü performansa sahip olan modeldir.

Thomas ve Christopher (2017) Uzun-kısa süreli bellek (Long short-Term Memory - LSTM) ağlarının sıralı öğrenme için en yeni teknik olduğuna vurgu yaparak,

finansal zaman serileri tahminlerinin doğal olarak uygun olmasına rağmen daha az uygulanmakta olduğunu belirtmişlerdir. 1992'den 2015'e kadar olan S&P500'ün kurucu hisse senetleri için örnek dışı yönlü hareketleri tahmin etmek için LSTM ağlarını kullanmışlardır. İşlem maliyetlerinden günlük olarak yüzde 0.46 geri dönüş ve Sharpe Oranı ile 5.8 dolarlık bir performans sunduğunu göstermişlerdir. Özellikle, alım satım için seçilen hisse senetleri arasında ortak bir model geliştirmişlerdir.

Wong ve ark. (2010) zaman serisi tahmini için uyarlamalı girdi ölçütlerine sahip yeni bir Uyarlamalı Sinir Ağı (*Adaptive Neural Network* - ADNN) ve çıktılarının katkısı için yeni bir yaklaşım önermişlerdir. Adaptif giriş ölçüleri, genlik değişimi ve trend belirleme problemlerini çözebilir ve ağların aşırı uyumunu önleyebilir. Çıktıların karışımı için önerilen mekanizma, tahmin sonuçlarını göreceli hata ile ayarlayabilir ve daha doğru bir hale getirebilir şekilde düzenlenmiştir. Önerilen ADNN yöntemi, periyodik zaman serilerini karmaşık bir yapıyla tahmin edebilmiştir. Deneysel sonuçlar, önerilen modelin oto-regresyon (AR), yapay sinir ağı (YSA) ve adaptif K-en yakın komşu (AKN) modellerinden daha üstün olduğunu göstermektedir. Çalışmada ADNN modelinin, özellikle hem kaotik hem de gerçek zaman serisi tahminleri için yüksek sağlamlığa sahip yeni yapısıyla YSA'nın ve AKN'nin yararlarından faydalandığı kanıtlanmıştır.

Bernal ve ark. (2012) çalışmalarında S&P500'ün hisse senedi fiyatlarını tahmin etmek için, tekrarlayan sinir ağlarının bir alt sınıfı olan eko devlet ağlarını kullanmışlardır. Hisse fiyatındaki daha yüksek frekans dalgalanmalarını tahmin eden bir Kalman filtresinin daha iyi performans gösterdiğini göstermişlerdir.

Bircan ve Karagöz (2003) geleceği tahmin metotlarından Box-Jenkins metoduyla döviz kurları üzerinde bir uygulama yapmışlardır. Zaman serileri hakkında genel bilgiler verilerek, Box-Jenkins modelleri incelenmiş ve Box-Jenkins metodunun uygulama safhaları açıklanmıştır. Çalışmada, Ocak 1991 ve Aralık 2002 dönemini kapsayan 132 aylık döviz kuru serisi için en uygun tahmin modeli tespit edilmiştir.

Fakhr ve Baasher (2011) günlük eğilimi tahmin etmenin zorlu bir problem olduğu oldukça değişken bir karmaşık zaman serisi olan döviz piyasası piyasası (FOREX) üzerine çalışma yapmışlardır. Yüksek döviz kuru günlük eğiliminin, yükseliş eğilimi ve düşüş eğilimi sonuçları ile ikili bir sınıflandırma sorunu olarak alınmıştır. Teknik analiz özellikleri dahil olmak üzere zaman serisi verisinden kaynaklanan çok

sayıda temel özellik, çoklu geçmiş zaman pencereleri kullanılarak oluşturulmuş ve sınıflandırma problemi için en iyi alt kümeleri bulmak için çeşitli özellik seçimi ile özellik çıkarma teknikleri kullanılmıştır. Radyal tabanlı fonksiyon sinir ağı (RBF), çok katmanlı algılayıcı sinir ağı (MLP) ve destek vektör makinesi (SVM) makine öğrenme metotları, her bir özellik alt kümesi için test ve analiz edilmiştir. Dört önemli FOREX döviz çifti incelenerek, sonuçların günlük tahminlerde ve beklenen kârda tutarlı bir başarı gösterdiği tespit edilmiştir.

Zhang (2003) zaman serileri için ARIMA modellerinin ve YSA'ların, genellikle tahmin sonuçlarındaki üstünlük açısından karışık sonuçlar ile karşılaştırıldığına vurgu yapılan çalışmasında, ARIMA ve ANN modellerini birleştiren hibrid bir metodoloji geliştirmiştir. Doğrusal ve doğrusal olmayan modellemede, ARIMA ve ANN modellerinin benzersiz gücünden yararlanması önerilmektedir. Gerçek veri setleri ile yapılan deneysel sonuçların, birleştirilmiş modelin, ayrı olarak kullanılan modellerden herhangi biri ile elde edilen tahmin doğruluğunu iyileştirmenin etkili bir yolu olabileceğini göstermiştir.

Bao ve ark. (2017) çalışmada, dalgacık dönüşümü (*wavelet transforms* - WT), yığılmış otomatik kodlayıcılar (*stacked autoencoders* - SAE) ve uzun-kısa süreli hafızanın (LSTM) stok fiyat tahmini için birleştirildiği yeni bir derin öğrenme çerçevesi sunmaktadır. Hiyerarşik olarak ayıklanan özellikler için SAE ilk kez hisse senedi fiyat tahminine dahil edilmiştir. Derin öğrenme çerçevesi üç aşamadan oluşmaktadır. İlk olarak, hisse senedi fiyat zaman serileri gürültüyü ortadan kaldırmak için WT tarafından ayrıştırılır. İkinci olarak SAE hisse senedi fiyatını tahmin edecek derin yüksek seviyeli özellikler üretmek için uygulanır. Son aşamada, yüksek seviyeli gürültüden arındırılmış özellikler, ertesi günün kapanış fiyatını tahmin etmek için LSTM'ye aktarılır. Önerilen modelin performansını incelemek için altı piyasa endeksi ve bunlara karşılık gelen endeks vadeli işlemler seçilmiştir. Sonuçlar, önerilen modelin hem benzer tahminde hem de karlılık performansında diğer benzer modellerden daha iyi performans gösterdiğini göstermektedir.

Morariu ve ark. (2009) tahmin için "desen tanıma (*pattern recognition*)" ve "çok katmanlı algılayıcı (*multilayer perceptron*)" yaklaşımlarını kullanmışlardır. Bir zaman serisi göstergeleri için tahminlemede bulunmak için model tanıma teknikleri ve yapay sinir ağlarının kullanımına ilişkin metotlar sunulmuştur. Sınıflandırma ve tanıma

teknikleriyle elde edilen sınıflar üzerinden, negatif gelişimden pozitif gelişime doğru temsil edilmesine izin veren bir ilişki tanımlanmıştır. Ayrıca yazarlar tarafından yazılan REFORME yazılımı ve bu yazılımla elde edilen deney sonuçları, 2003-2010 yılları arasında makroekonomik tanı ve tahmin için kullanılmıştır.

Kuna (2015) tez çalışmasında yapay sinir ağını kullanarak zaman serilerini tahmin etmeye çalışmıştır. Tahmin etme süreçlerini RNN, Real-Time Recurrent Learning ve Backpropagation Through Time ile gerçekleştirmiştir. Elman'ın Standart RNN mimarisine ek olarak Clockwork-RNN incelemiştir. Mimariler, gerçek zamanlı uygulamalarda kritik olan tahmin doğruluğu ve hesaplama süresi açısından karşılaştırılmıştır. Çalışmada, robotik ve ağ trafiği izlemesinde test edilmiş modeller ve çalışma uygulamaları deneysel olarak kullanılmıştır.

Kaynar ve Taştan (2009), makalelerinde Box-Jenkins Modeli ile yapay sinir ağı karşılaştırması yapmışlardır. Veri seti olarak aylık ve günlük TL/Dolar kuru verilerini kullanmışlardır.

3. MATERYAL VE YÖNTEM

Zaman serisi modellemesi, özellikle son yıllarda araştırmacıların dikkatini çeken dinamik bir çalışma alanı içinde yer almaktadır. Zaman serileri modellemesinin temel hedefi, matematiksel veya istatistiksel yaklaşımla serinin veri seti yapısına uygun bir model geliştirmek amacıyla geçmişteki gözlemleri dikkatlice toplamak ve incelemektir. Elde edilen model gelecek değerleri tahmin etmekte kullanılır. Bu çalışmada tahminleme için yaygın şekilde kullanılan istatistiki bir metod olan **ARIMA** ile oldukça popüler hale gelen **LSTM** derin öğrenme mimarisi kullanılmıştır. İki metod karşılaştırılarak sonuçlar çeşitli model tahmin ölçekleri ile değerlendirilmiştir.

Eğitilmiş model üzerinde anlık olarak yeni veri geldiğinde tahminleme yapılabilecek bir web arayüzü geliştirilmiştir.

Tez çalışması sırasında zaman serisi verisini analiz etmek için açık kaynak kodlu yazılımlar, kütüphaneler ve çeşitli online veri kaynaklarından anlık veri kullanılmıştır.

LSTM derin öğrenme mimarisi için Python programlama dili ile modelleme yapılırken, ARIMA istatistiki metodu için model R uygulama platformu kullanılarak gerçekleştirilmiştir.

3.1. Kullanılan Programlama Dili Kütüphaneleri ve Optimizasyon Algoritmaları

Theano; matematiksel formülleri tanımlamak, optimize etmek ve bu formüllerin çok boyutlu diziler üzerinde kolayca çalıştırılabilmesine olanak sağlayan açık kaynak kodlu bir Python kütüphanesidir. Derin Öğrenme ve diğer makine öğrenmesi algoritmalarının bilgisayar üzerinde matematiksel olarak tasarlanması ve çalıştırılması için oldukça uygun arayüzler sunar (Bergstra ve ark., 2010).

Keras; Theano kütüphanesini kullanarak derin öğrenme algoritmalarını kolayca geliştirmek ve uygulamak için hazırlanmış açık kaynak kodlu derin öğrenme Python programlama dili kütüphanesidir. Keras ilk olarak ONEIROS araştırma projesinin (Açık uçlu Neuro-Elektronik Akıllı Robot İşletim Sistemi) bir parçası olarak geliştirilmiştir (Chollet, 2016).

R: İstatistiksel hesaplama ve grafik çizimi için kullanılan bir programlama dilidir. John Chambers ve meslektaşları tarafından Bell Laboratuvarlarında geliştirilen S

diline ve çevreye benzer bir GNU projesi olarak ortaya çıkmıştır. R çok çeşitli istatistiksel (doğrusal ve doğrusal olmayan modelleme, klasik istatistiksel testler, zaman serileri analizi, sınıflandırma, kümeleme, vb) hesaplamalar yapabilir ve grafik görseller oluşturabilir. R'nin güçlü yönlerinden biri, ihtiyaç duyulan matematiksel semboller ve formüller de dahil olmak üzere iyi tasarlanmış yayın kalitesinde grafiklerin üretilebilme kolaylığıdır (R Development Core Team, 2008).

R platformunda veri seti yükleme ve zaman serisi analizleri için **forecast**, **ggplot2**, **tseries** ve **readr** kütüphaneleri kullanılmaktadır.

NumPY: Python ile bilimsel hesaplama için kullanılan temel paketlerden biridir. Temel özellikleri; güçlü bir N-boyutlu dizi nesnesi, sofistike (yayın) fonksiyonları, C/C++ ve Fortran kodunu entegre etmek için araçlar, kullanışlı doğrusal cebir, Fourier dönüşümü kullanabilmesidir. NumPy'nin çok çeşitli veri tabanları ile sorunsuz ve hızlı bir şekilde entegre olabilir. NumPy, BSD lisansı kapsamında lisanslıdır ve birkaç kısıtlama ile yeniden kullanılabilir (Oliphant, 2006).

Ön Yüz İçin Kullanılan Javascript kütüphaneler: Ön yüz tasarımında kullanıcı ile etkileşim ve grafik arayüzleri için **VueJS**, **Bootstrap4** ve **chartJS** kütüphaneleri kullanılmıştır.

ADAM optimizasyon algoritması: İstatistiksel objektif fonksiyonların birinci dereceden gradyan tabanlı optimizasyonunu sağlayan bir algoritmadır. Yöntemin uygulanması basittir ve sayısal olarak etkilidir. Bellek gereksinimi azdır. Gradyanların çapraz yeniden ölçeklenmesinde değişmezdir. Ayrıca veri ve parametreler açısından büyük problemler için çok uygundur. Yöntem aynı zamanda durağan olmayan hedefler ve çok gürültülü ve/veya seyrek gradyanlarla ilgili problemler için de uygundur. Hiper parametreler sezgisel yorumlara sahiptir ve genellikle çok az ayar gerektirir (Kingma ve Adam, 2014).

3.2. Veri Seti Kaynakları

Ekonomi zaman serisi verilerine uygun belirlenecek temel veri setleri için Forex, S&P500, IMKB100 gibi kaynaklar kullanılabilir. Bu çalışmada hisse senedi verileri için ücretsiz kullanılabilmesi nedeniyle Alpha Vantage kullanılmıştır. Araştırmacılar, mühendisler ve iş profesyonellerinin sıkı sıkıya bağlı bir topluluktan oluşan Alpha

Vantage Inc.; hisse senetleri, fiziki para birimleri ve dijital/kripto para birimleriyle ilgili gerçek zamanlı ve geçmişe dönük veriler için lider bir ücretsiz API'ler sağlayıcısıdır (<https://www.alphavantage.co>).

Bitcoin ile ilgili veriler için Quandl veri kaynağı kullanılmıştır (www.quandl.com)

Veri seti kaynaklarından elde edilen ve çalışmada kullanılan veri setlerine *projects.aocdiken.com* adresinden erişilebilir.

3.4. Tahmin Modeli Geliştirme Adımları

Bir zaman serisinin tahmini genellikle aşağıdaki aşamalardan oluşur (Brockwell ve Davis, 2002):

- **Problem tanımı:**

Çoğu zaman bu, tahmin etmenin en zor kısmıdır. Sorunun dikkatle tanımlanması, tahminlerin nasıl kullanılacağını, tahminleri kimin gerektirdiğini ve tahmin işlevinin kuruluşun öngörülere nasıl uyduğunu anlamayı gerektirir. Bir tahminci, gelecekteki planlamalar için tahminlerde kullanmak üzere veri toplamaya, veri tabanlarını korumaya ve kullanmaya zaman harcamalıdır.

- **Bilgi toplama:**

En az iki tür bilgi gereklidir: (a) istatistiksel veriler ve (b) verileri toplayan ve tahminleri kullanan kişilerin birikmiş uzmanlıkları. Çoğu zaman, iyi bir istatistiksel modele uymak için yeterli tarihsel veri elde etmek zor olacaktır. Bazen, eski veriler, tahmin edilen sistemdeki yapısal değişiklikler nedeniyle daha az yararlı olacaktır; o zaman sadece en yeni verileri kullanmayı seçebiliriz. Ancak, iyi istatistiksel modellerin sistemdeki evrimsel değişimleri ele alacağını unutulmamalı; gereksiz veri gereksiz yere atılmamalıdır.

- **Ön (keşif) analizi:**

Her zaman verilerin grafikleri ile başlanmalıdır. Tutarlı modeller var mı? Önemli bir eğilim var mı? Mevsimsellik önemli mi? İş döngüsünün varlığının kanıtı var mı? Verilerde uzman bilgisi olan kişiler tarafından açıklanması

gereken herhangi bir aykırılık var mı? Analiz için değişkenler arasındaki ilişkiler ne kadar güçlüdür? Bu analize yardımcı olmak için çeşitli araçlar geliştirilmiştir.

- **Modellerin seçilmesi ve yerleştirilmesi:**

Kullanılacak en iyi model, tarihsel verilerin kullanılabilirliğine, tahmin değişkeniyle açıklayıcı değişkenler arasındaki ilişkilerin gücüne ve tahminlerin nasıl kullanılacağına bağlıdır. İki veya üç potansiyel modeli karşılaştırmak yaygındır. Her modelin kendisi bir varsayımlar setine (açık ve kapalı) dayanan ve genellikle bilinen tarihsel verileri kullanarak tahmin edilmesi gereken bir veya daha fazla parametreyi içeren yapay bir yapıdır. Kullanılabilecek başlıca modeller; regresyon modelleri, üstel düzeltme yöntemleri, Box-Jenkins ARIMA modelleri, dinamik regresyon modelleri, hiyerarşik tahminleme ve yapay sinir ağları gibi modeller olabilir.

- **Bir tahmin modelinin kullanılması ve değerlendirilmesi:**

Bir model seçildikten ve parametreleri tahmin edildikten sonra, model tahmin yapmak için kullanılır. Modelin performansı, yalnızca öngörü dönemi verileri hazırlandıktan sonra uygun şekilde değerlendirilebilir. Tahminlerin doğruluğunu değerlendirmede yardımcı olacak birkaç yöntem geliştirilmiştir. Pratikte bir tahmin modeli kullanıldığında, eksik değerler ve aykırı değerlerin nasıl ele alınacağı veya kısa zaman serileri ile nasıl baş edileceği gibi birçok pratik sorun ortaya çıkmaktadır.

3.3. Finansal Zaman Serisi İçin Tahmin Modelleri

Finansal veriler zaman serilerine en uygun örnekler olarak kullanılmaktadır. Bu çalışmada hisse senedi ve Bitcoin kripto para finansal verileri için tahmin modelleri gerçekleştirilmiş ve sonuçları karşılaştırılmıştır.

3.3.1. Finansal Zaman Serisi İçin ARIMA Tahmin Modeli

ARIMA tahmin modeli için önerilen adımlar Çizelge 3.1.'de verilmiştir:

Çizelge 3.1. ARIMA tahmin modeli adımları

- **Adım 0.** Veri seti ve gerekli kütüphaneler yüklenir.
- **Adım 1.** Zaman serisi verisi oluşturulur. Zaman serisi grafiği çizilir.
- **Adım 2.** Durağanlık kontrol edilir.
- **Adım 3.** Mevsimsellik kontrol edilir.
- **Adım 4.** ARIMA modeli elde edilir.
- **Adım 5.** Tahmin modeli oluşturulur.
- **Adım 6.** Eğitim verileri oluşturulur ve veriler test edilir.
- **Adım 7.** Test verilerinin doğruluğu için performans ölçümü yapılır.

3.3.2. Finansal Zaman Serisi İçin LSTM Tahmin Modeli

LSTM derin öğrenme mimarisi kullanılarak tahmin modeli için önerilen adımlar Çizelge 3.2.'de verilmiştir:

Çizelge 3.2. LSTM tahmin modeli adımları

- **Adım 0.** Zaman serisi veri seti yüklenir.
- **Adım 1.** Derin öğrenme algoritması için ağ modeli oluşturulur.
- **Adım 2.** Veri seti 30 günlük zaman serisi şeklinde düzenlenir.
- **Adım 3.** Veri setinin %20 si test için ayrılır, geri kalanı eğitim verisi olarak belirlenir.
- **Adım 4.** Eğitim verileri kullanılarak model eğitilir.
- **Adım 5.** Eğitilen model üzerinde test verileri tahmin edilir.
- **Adım 6.** Test sonucu hata oranları ekrana yazdırılır.

3.4. ARIMA ve LSTM Modellerinin Çalıştırılmasında Kullanılan Donanım

Derin öğrenme algoritmaları yüksek hesaplama gücü gerektirmektedir. Tezde yapılan çalışmalar için eğitim, doğrulama ve test adımları Necmettin Erbakan Üniversitesi İleri Hesaplama ve Veri Analizi Laboratuvarı'nda yapılmıştır. Bu merkezde

1 adet GPU düğümü, 22 adet CPU düğümü arařtırmacıların kullanımına sunulmaktadır. Tezde gerekleřtirilen alıřmalar GPU düğümü üzerinde yapılmıřtır. 1 GPU makinesine ait donanım bilgileri izelge 3.3.'de gsterilmiřtir.

izelge 3.3. GPU düğümü donanım zellikleri

İřlemci	4C Core i7-7700K 4.2G 8M 8GT/s DMI
Bellek	16 GB DDR4
Disk	240GB SSD Drive
GPU	GTX 1080 TI 11G 352Bit
Ağ Baęlantısı	1 x Juniper QFX 5100

3.5. Hisse Senedi Fiyat Tahmini İin LSTM Modeline Dayalı Anlık Tahmin Web Arayüz Uygulaması

LSTM modelinin oluřturulması Keras kütüphanesi ile giriř deęerleri, ıktı ve ara katmanlar tasarlanarak hazırlanmıřtır. Keras kütüphanesi ile her bir hisse senedi iin eęitilen modeller **Hdf5** dosyası olarak kaydedilmiřtir. Kaydedilen modellerin test edilmesi iin bir web arayüzü geliřtirilmiřtir. Arayüze ve kaynak kodlarına projects.aocdiken.com web adresinden eriřilenebilir.

Geliřtirilen web arayüzü eęitim ařamasında kullanılan gemiř verileri ve anlık verilere gre yapılan tahminleri farklı renklerle gstermektedir.

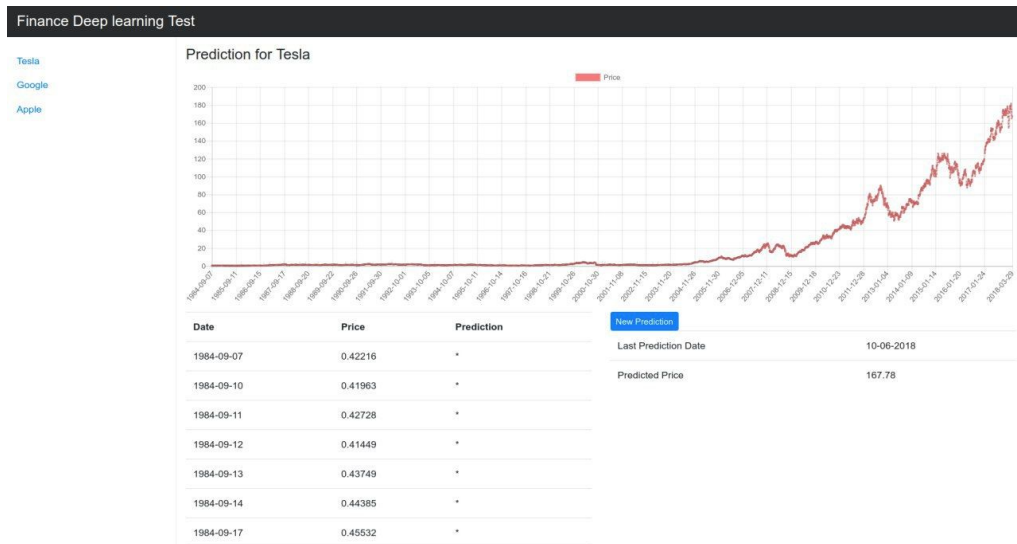
Web arayüzü **sunucu uygulaması** (*backend*) ve **nyüz** (*frontend*) olmak üzere iki ayrı proje řeklinde geliřtirilmiřtir. nyüzde **VueJS erevesi** (*framework*) kullanılarak Javascript ile geliřtirilmiřtir. Arayüz tasarımı **Bootstrap4 kütüphanesi** kullanılarak yapılmıřtır. Gemiř veriler ve tahminlerin grafik olarak gsterimini saęlamak iin **chartJS kütüphanesi** kullanılmıřtır. nyüz ve sunucu uygulaması arasındaki baęlantılar Rest API üzerinden saęlanmaktadır.

Sunucu uygulama projesi Python programlama dilinde varsayılan olarak gelen **SimpleHTTP Server** fonksiyonu kullanılarak geliřtirilmiřtir. Sunucu uygulama projesi 3 fonksiyon iin Rest API sunmaktadır:

- Birinci fonksiyon nyüz tarafından gnderilen tarihten nceki 30 gnlük verileri anlık olarak Alpha Vantage üzerinden sorgulamakta ve arayüze aktarmaktadır.

- İkinci fonksiyon kullanıcıdan gelen 30 günlük veriyi kullanarak bir sonraki günü tahmin etmekte ve sonucunu göstermektedir. Tahmin edilecek hisse senedine ait en başarılı Hdf5 dosyasındaki model Keras kullanılarak yüklenmekte ve model ile ağırlıkları oluşturulmaktadır. Tahmin fonksiyonu gelen verileri eğitim aşamasındaki verileri kullanarak 0-1 arasında ölçeklemektedir. Ölçeklenen 30 günlük veri iki boyutlu dizi olarak modele girdi verilerek tahmin işlemi gerçekleştirilmektedir. Model çalıştırılarak üretilen tahmin 0-1 arasından gerçek değerine ölçekleyerek arayüzden kullanıcıya döndürülmektedir.
- Üçüncü fonksiyon eğitim aşamasında kullanılmayan yeni verilerin tekrar eğitime alınarak modelin geliştirilmesi için kullanılmaktadır. Fonksiyon verilen tarih için 30 günlük geçmiş veriyi ve gerçekleşen değeri almaktadır. 30 günlük veri 2 boyutlu dizi olarak modele eğitim verisi olarak verilmekte ve 1 geri besleme şeklinde çalıştırılmaktadır. Geliştirilen model ilk eğitim aşamasında kullanılan verilerden rastgele olarak seçilen %20 lik veri kullanılarak test edilmektedir. Test sonucu hata oranı bir önceki hata oranından daha düşük ise yeni model Hdf5 dosyası olarak kaydedilmektedir.

Oluşturulan web arayüzü örneği Şekil 3.1.'de gösterilmektedir.



Şekil 3.1. Apple, Google ve Tesla hisse senedi fiyatları için anlık tahmin web arayüzü

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Geliştirilen algoritma adımları, hisse senedi ve Bitcoin fiyatı veri setleri için ayrı ayrı uygulanmış ve sonuçları elde edilmiştir. Çalışmalar yapılırken veri setindeki günlük fiyat verileri ile bir zaman serisi oluşturulmuştur. Oluşturulan zaman serileri üzerinde istatistiki yaklaşım ile ARIMA modeli, makine öğrenmesi yaklaşımı ile LSTM derin öğrenme modelleri elde edilmiştir. Algoritmalar ayrı ayrı çalıştırılmış ve sonuçlar bu bölümde paylaşılmıştır.

4.1. Hisse Senedi Fiyat Tahmin Modelleri

NASDAQ dünyanın ikinci büyük hisse senetleri borsasıdır. NASDAQ borsasında işlem gören Apple firmasına ait hisse senedi tahminlemesi ARIMA ve LSTM modelleri kullanılarak yapılmıştır.

Veri seti olarak; Apple firmasına ait, 7 Eylül 1984 ile 14 Şubat 2018 tarihleri arasındaki 8428 adet günlük hisse kapanış fiyatı kullanılmıştır.

4.1.1. Hisse Senedi Fiyat Tahmini İçin ARIMA Modeli

Çizelge 3.1. kullanılarak ARIMA tahmin modeli adımları R platformunda gerçekleştirilmiş ve çıktıları verilmiştir.

- **Adım 0.** Veri seti ve gerekli kütüphaneler yüklenir.

```
> library(forecast)
> library(ggplot2)
> library(tseries)
> library(readr)
> apple_gunluk <- read_csv("apple_gunluk.csv")
```

- **Adım 1.** Zaman serisi verisi oluşturulur.

```
> Apple <- ts(apple_gunluk, frequency=365)
```

- **Adım 2.** Durağanlık kontrol edilir. Zaman serisi grafiği çizilir.

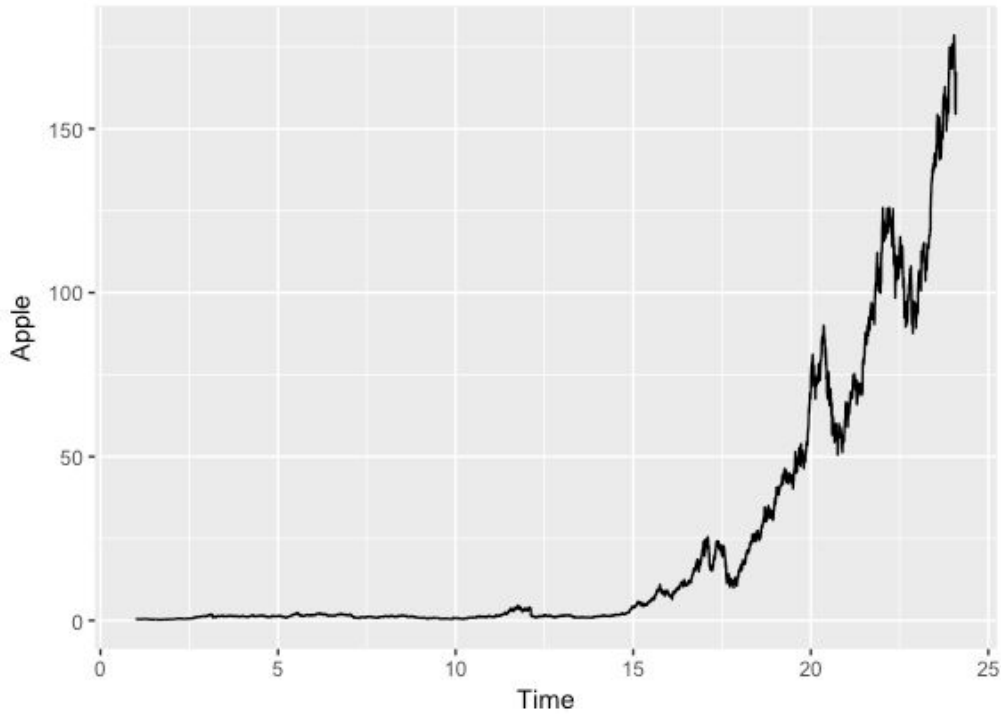
```
> kpss.test(Apple)
```

```
KPSS Test for Level Stationarity
```

```
data: Apple
```

```
KPSS Level = 23.495, Truncation lag parameter=21, p-value=0.01
```

```
> autoplot(Apple)
```



Şekil 4.1. Apple hisse senedi kapanış fiyatlarının zaman serisi grafiği

Burada KPSS testine göre $p\text{-value}=0.01$ olduğundan zaman serisi durağan değildir.

- **Adım 3.** Mevsimsellik kontrol edilir.

```
> Test_Mevsimsellik <- tbats(Apple)
```

```
> Mevsimsellik <- !is.null(Test_Mevsimsellik$seasonal)
```

```
> Mevsimsellik
```

```
[1] FALSE
```

Burada Mevsimsellik değeri FALSE olduğundan zaman serisi mevsimsel değildir.

- **Adım 4.** ARIMA modeli elde edilir.

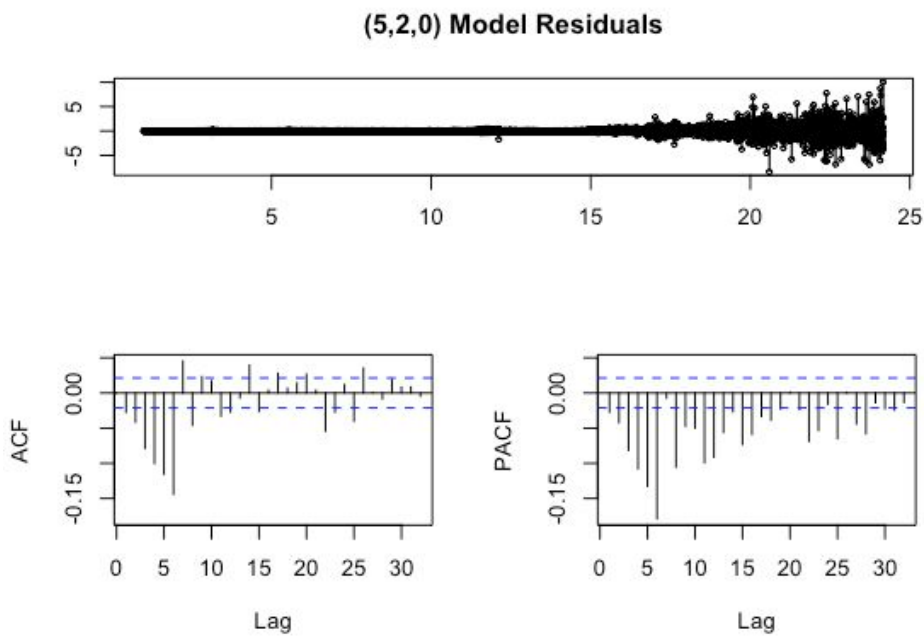
```
> Model<-auto.arima(Apple, seasonal=FALSE,  
stepwise=FALSE, approximation=FALSE, trace=TRUE)
```

```

> Model
Series: Apple
ARIMA(5,2,0)
Coefficients:
          ar1          ar2          ar3          ar4          ar5
      -0.8051  -0.6824  -0.5167  -0.3052  -0.1494
s.e.    0.0108   0.0135   0.0144   0.0137   0.0109
sigma^2 estimated as 0.5371:  log likelihood=-9335.18
AIC=18682.36  AICc=18682.37  BIC=18724.6
> autoplot(Model)

```

Burada ARIMA(5,2,0) olacak şekilde model elde edilmiştir. Elde edilen model için ACF ve PACF grafikleri verilmiştir (Şekil 4.2.).



Şekil 4.2. ARIMA(5,2,0) modeli için ACF ve PACF grafikleri

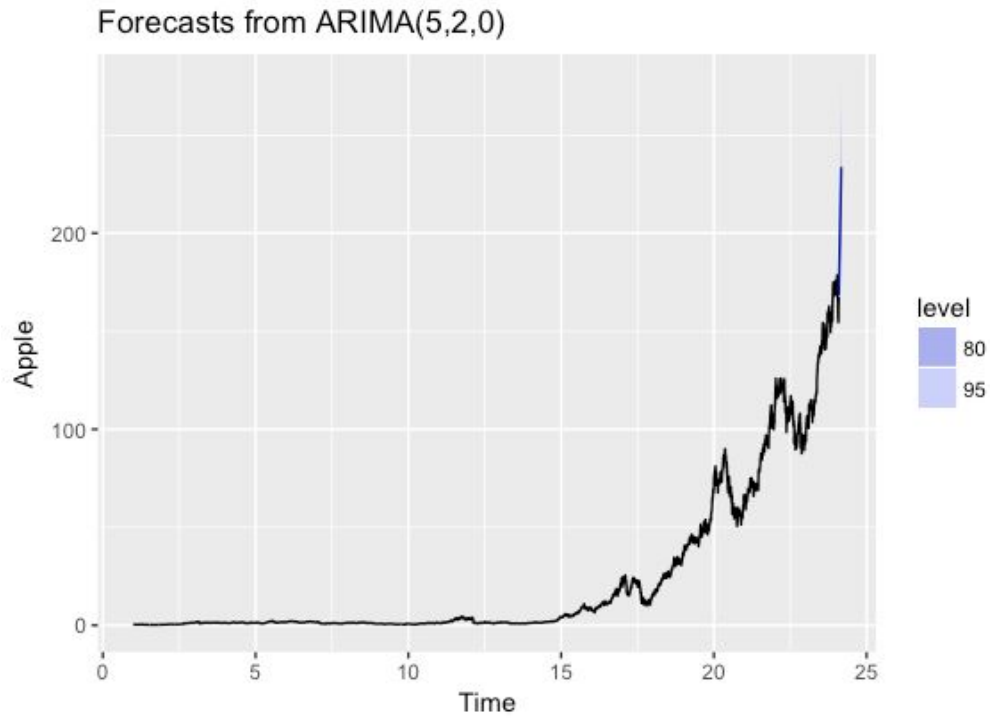
- **Adım 5.** Tahmin modeli oluşturulur.

```

> Tahmin_30gun <- forecast(Model,h=30)
> autoplot(Tahmin_30gun)

```

Burada sonraki 30 gün (15 Şubat 2018 - 29 Mart 2018) için tahminleme yapılmıştır (Şekil 4.3.).



Şekil 4.3. ARIMA(5,2,0) modeli için 30 günlük fiyat tahmin grafiği

- **Adım 6.** Eğitim verileri oluşturulur ve veriler test edilir.

```
> Eğitimverisi <- data.frame(Tahmin_30gun)
> Eğitimverisi <- Eğitimverisi$Point.Forecast
> Testverisi <- c(172.99, 172.43, 171.85, 171.07, 172.5, 175.5,
178.97, 178.39, 178.12, 175, 176.21, 176.82, 176.67, 175.03,
176.94, 179.98, 181.72, 179.97, 178.44, 178.65, 178.02, 175.3,
175.24, 171.27, 168.85, 164.94, 172.77, 168.34, 166.48, 167.78)
```

- **Adım 7.** Test verilerinin doğruluğu için performans ölçümünü yapılır.

```
> accuracy(Eğitimverisi,Testverisi)
              ME      RMSE      MAE      MPE      MAPE
Test set -26.55865 34.04762 27.01132 -15.38245 15.6444
```

ARIMA(5,2,0) tahmin modeli için performans değerleri Çizelge 4.1.'de verilmiştir.

Çizelge 4.1. Hisse senedi fiyatı ARIMA(5,2,0) tahmin modeli için performans değerleri.

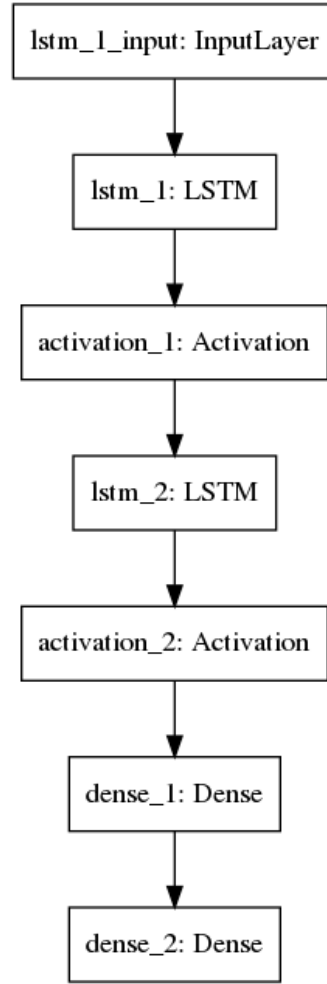
Hisse senedi fiyatı	RMSE	MAE	MPE	MAPE
ARIMA(5,2,0) Modeli	34.04762	27.01132	-15.38245	15.6444

4.1.2. Hisse Senedi Fiyat Tahmini İçin LSTM Modeli

Keras kütüphanesi kullanılarak geliştirilen modelde 30 günlük hisse senedi fiyat zaman serisi verisi girdi olarak kullanılmıştır. 30 günlük girdi değerleri **NumPy** ile tek boyutlu diziye dönüştürülmüştür. Tüm veriler 0-1 arasında ölçeklendirilmiştir. Keras kütüphanesi kullanılarak oluşturulan modelde katmanlara göre yapılan işlemler şunlardır:

- **Giriş katmanı:** 30 eleman ile bire bir bağlı 64 LSTM nöronu bulunmaktadır. İkinci katman ile arasında aktivasyon fonksiyonu olarak ReLU kullanılmıştır.
- **İkinci katman:** 128 nöron bulunmaktadır. Giriş katmanına benzer şekilde ikinci katmanda da LSTM kullanılmıştır. Üçüncü katman ile arasında aktivasyon fonksiyonu olarak ReLU kullanılmıştır.
- **Üçüncü katman:** 32 MLP nöron bulunmaktadır. Bu katman çıkış katmanı öncesi LSTM katmanlarından gelen verinin birleştirilmesi için kullanılmıştır.
- **Çıkış katmanı:** 1 MLP nöron bulunmaktadır. Herhangi bir aktivasyon fonksiyonu kullanılmamıştır. Tüm veri seti 0-1 arasında ölçeklendirildiği için çıkış katmanının değeri de tersine çevrilmelidir.

Optimizasyon algoritması olarak ADAM kullanılmıştır. Algoritma üzerinde öğrenme oranı olarak 0.001 tercih edilmiştir. Öğrenme esnasında **geri besleme işlemi** (*epoch*) 32 kez tekrarlanmıştır. Oluşturulan modelin Keras kütüphanesi yardımıyla elde edilen mimarisi Şekil 4.4.' de gösterilmiştir.



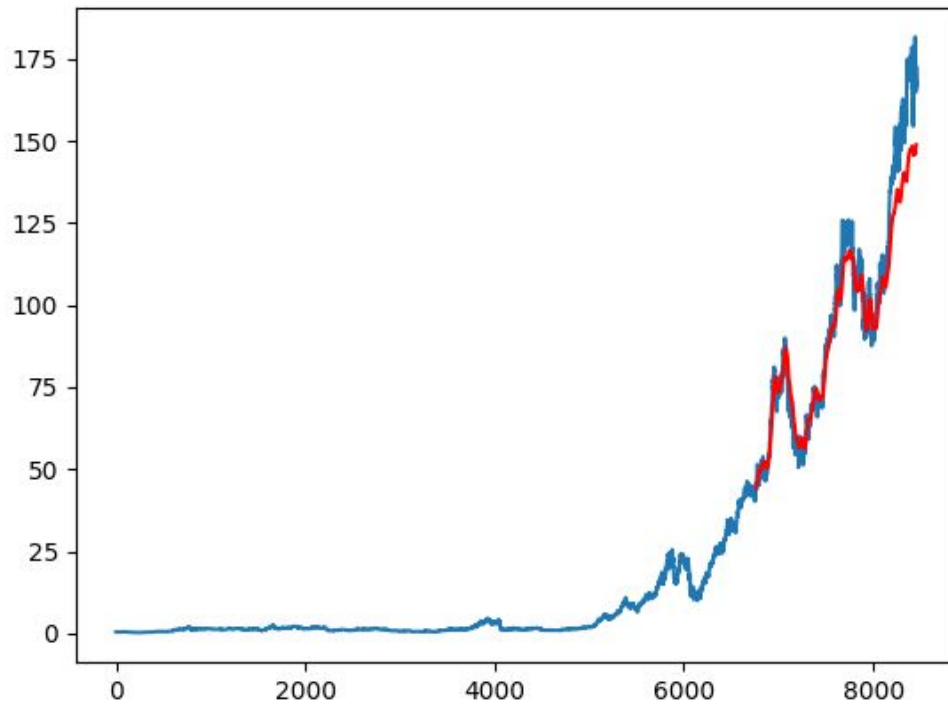
Şekil 4.4. Hisse senedi tahmini için oluşturulan LSTM model mimarisi

Elde edilen model sonucunda performans değerleri Çizelge 4.2.' de verilmiştir.

Çizelge 4.2. Hisse senedi fiyatı LSTM tahmin modeli için performans değerleri.

Hisse senedi fiyatı	RMSE	MAE	MPE	MAPE
LSTM Modeli	9.61	6.48	2.75	5.78

Hisse senedi fiyatı LSTM modeli tahmin grafiği Şekil 4.5.'de verilmiştir.



Şekil 4.5. Hisse senedi fiyatı LSTM modeli tahmin grafiği (Kırmızı çizgi tahmin değerlerini göstermektedir)

4.2. Bitcoin Fiyat Tahmin Modelleri

Bitcoin, 2008 yılında Satoshi Nakamoto adını taşıyan gizemli bir yazılım geliştiricisi tarafından matematiksel kanıt ve kriptografik yöntemle dayanan bir elektronik ödeme sistemi olarak önerilmiştir (Nakamoto, 2008). Sistem herhangi bir merkezi otoriteden bağımsız olarak çalışabilir. Para elektronik olarak güvenli, kanıtlanabilir ve değişmez olarak aktarılabilir. Bitcoin, Satoshi'nin makalesinden sonra anlık ödemeleri teşvik etmek için merkezi olmayan ağı kullanan ilk dijital para birimlerinden biri oldu. Bitcoin bir yatırım olarak tasarlanmamış olmasına rağmen, birçok kişi yatırım için borsalarda Bitcoin ve kripto para birimleri satın almaktadır. Bugün, Bitcoin dünya çapında 40'ın üzerinde borsada işlem görmektedir. Pazar büyüklüğü 2018 Nisan ayında yaklaşık 11 milyar dolara ulaşmıştır (<https://coinmarketcap.com/charts/>). Fiyat verisini zaman serisi olarak elde etmek için herhangi bir dönemdeki (ay, hafta, gün, bir saat vb.) kapanış fiyatları kullanılır.

Veri seti, 28 Nisan 2013 ve 29 Ekim 2017 arasında günlük olarak alınan Bitcoin fiyatından oluşmaktadır (www.quandl.com). Eğitim için 1646 günlük Bitcoin fiyatları kullanılmıştır. Elde edilen modeller sonraki 30 gün için Bitcoin fiyatlarını öngörmektedir.

4.2.1. Bitcoin Fiyat Tahmini İçin ARIMA Modeli

Çizelge 3.1. kullanılarak ARIMA tahmin modeli adımları R platformunda gerçekleştirilmiş ve çıktıları verilmiştir.

- **Adım 0.** Veri seti ve gerekli kütüphaneler yüklenir.

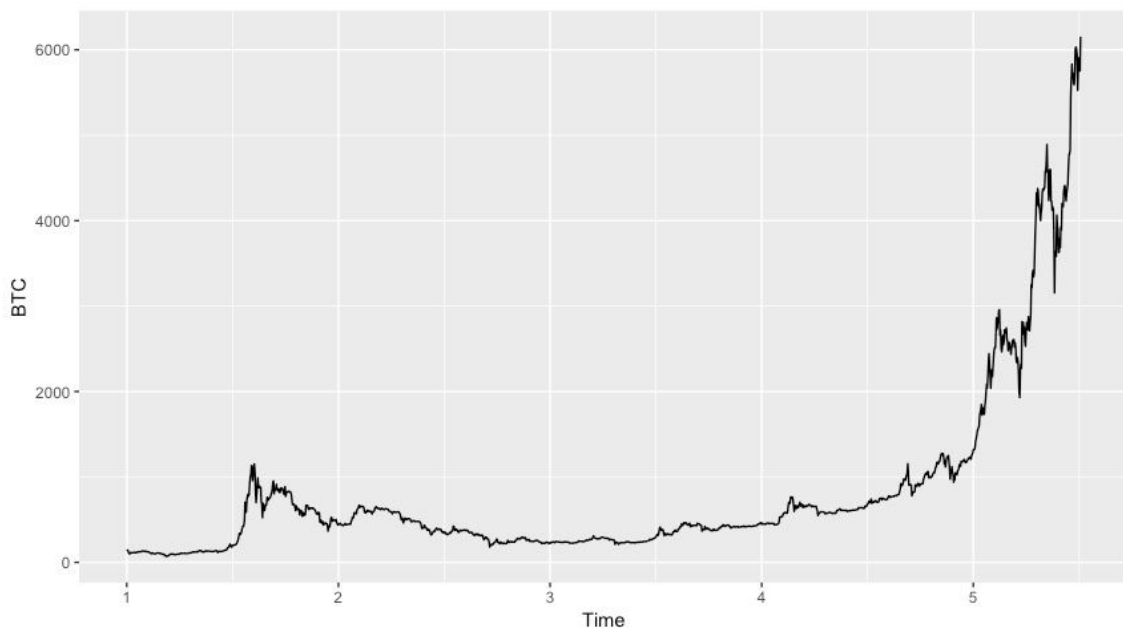
```
> library(forecast)
> library(ggplot2)
> library(tseries)
> library(readr)
> btc_usd_gunluk <- read_csv("btc_usd_gunluk.csv")
```

- **Adım 1.** Zaman serisi verisi oluşturulur. Zaman serisi grafiği çizilir.

```
> BTC <- ts(btc_usd_gunluk, frequency=365)
```

- **Adım 2.** Durağanlık kontrol edilir.

```
> kpss.test(BTC)
> autoplot(BTC)
```



Şekil 4.6. BTC fiyat grafiği.

Burada KPSS testine göre $p\text{-value}=0.01$ olduğundan zaman serisi durağan değildir.

- **Adım 3.** Mevsimsellik kontrol edilir.

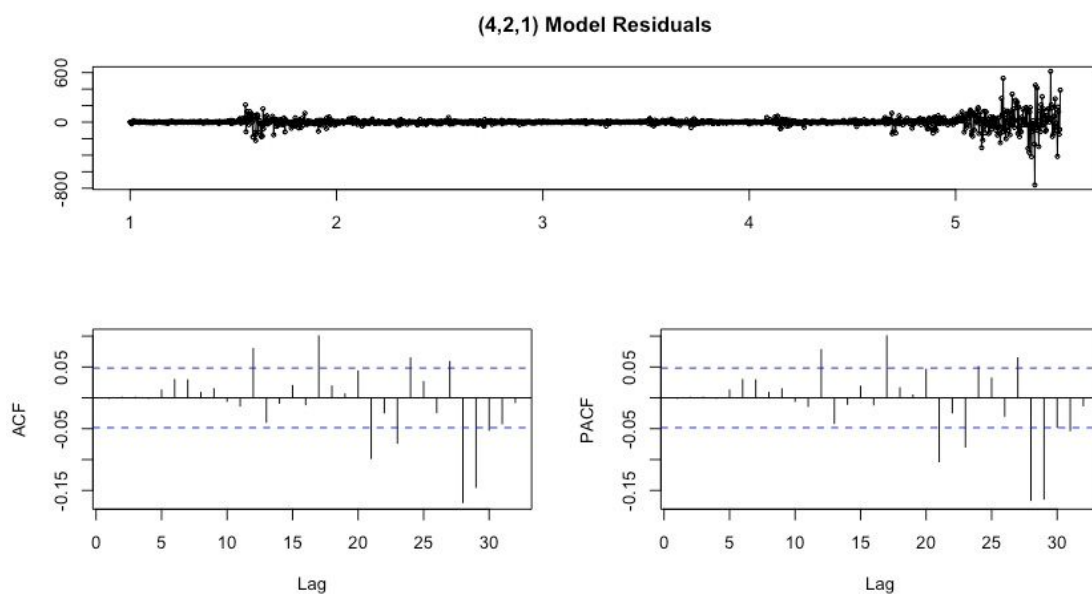
```
> Test_Mevsimsellik <- tbats(BTC)
> Mevsimsellik <- !is.null(Test_Mevsimsellik$seasonal)
> Mevsimsellik
[1] FALSE
```

Burada Mevsimsellik değeri FALSE olduğundan zaman serisi mevsimsel değildir.

- **Adım 4.** ARIMA modelini elde edilir.

```
> Model<-auto.arima(BTC, seasonal=FALSE,
stepwise=FALSE, approximation=FALSE, trace=TRUE)
> Model
Series: BTC
ARIMA(4,2,1)
Coefficients:
          ar1          ar2          ar3          ar4          ma1
      -0.0391  -0.0138   0.0034  -0.0745  -0.9899
s.e.      0.0251   0.0251   0.0252   0.0251   0.0036
sigma^2 estimated as 3465: log likelihood=-9032
AIC=18076.01  AICc=18076.06  BIC=18108.44
> tsdisplay(residuals(Model), lag.max=32, main="(4,2,1) Model
Residuals")
```

Burada ARIMA(4,2,1) olacak şekilde model elde edilmiştir. Elde edilen model için ACF ve PACF grafikleri verilmiştir (Şekil 4.7.).

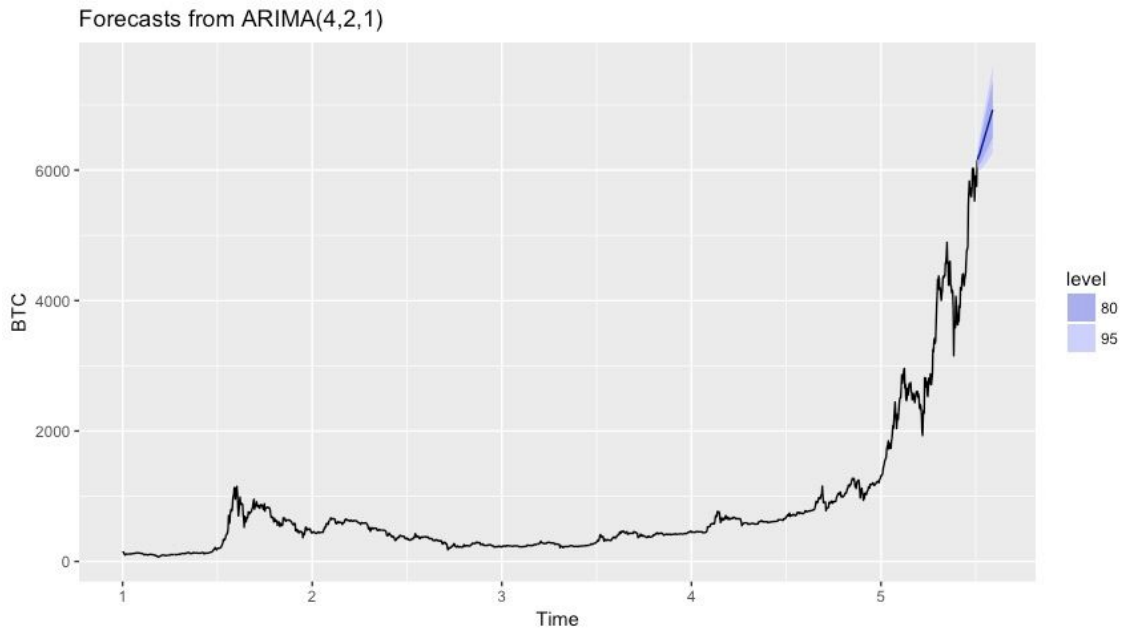


Şekil 4.7. ARIMA(4,2,1) modeli için ACF ve PACF grafikleri

- **Adım 5.** Tahmin modeli oluşturulur.

```
> Tahmin_30gun <- forecast(Model, h=30)
```

Burada sonraki 30 gün (30 Ekim 2017 - 30 Kasım 2018) için tahminleme yapılmıştır (Şekil 4.8.).



Şekil 4.8. ARIMA(4,2,1) modeli için 30 günlük fiyat tahmin grafiği

- **Adım 6.** Eğitim verileri oluşturulur ve veriler test edilir.

```
> Eğitimverisi <- data.frame(Tahmin_30gun)
> Eğitimverisi <- Eğitimverisi$Point.Forecast
> Testverisi <- c(6158.76, 6115.15, 6411.84, 6694.38, 7069.03,
7199.62, 7369.08, 7400.39, 7025.14, 7167.19, 7447.27, 7154.71,
6622.42, 6368.32, 5852.81, 6527.20, 6714.17, 7280.20, 7814.49,
7738.42, 7833.45, 8012.64, 8225.41, 8132.54, 8260.71, 8054.45,
8239.31, 8706.60, 9207.99, 9713.31)
```

- **Adım 7.** Test verilerinin doğruluğu için performans ölçümü yapılır.

```
> accuracy(Eğitimverisi, Testverisi)
              ME      RMSE      MAE      MPE      MAPE
Test set 880.0015 1146.067 939.5819 10.86483 11.86484
```

ARIMA(4,2,1) tahmin modeli için performans değerleri Çizelge 4.3.'de verilmiştir.

Çizelge 4.3. Bitcoin fiyatı ARIMA(4,2,1) tahmin modeli için performans değerleri.

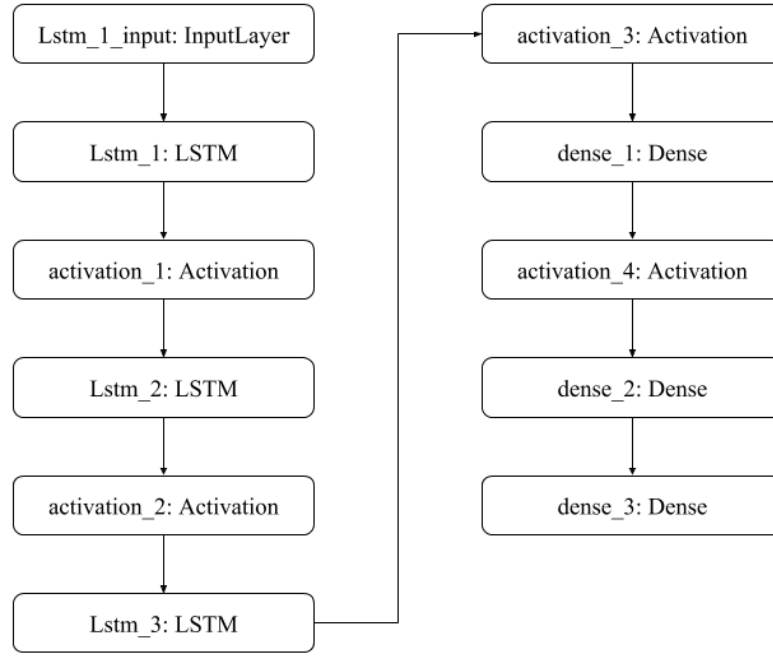
Bitcoin fiyatı	RMSE	MAE	MPE	MAPE
ARIMA(4,2,1) Modeli	1146.067	939.5819	10.86483	11.86484

4.2.2. Bitcoin Fiyat Tahmini İçin LSTM Modeli

Keras kütüphanesi kullanılarak geliştirilen modelde 30 günlük Bitcoin fiyat zaman serisi verisi girdi olarak kullanılmıştır. 30 günlük girdi değerleri **NumPy** ile tek boyutlu diziye dönüştürülmüştür. Tüm veriler 0-1 arasında ölçeklendirilmiştir. Model, giriş ve çıkış katmanları arasında 3 LSTM ve 2 gizli katmana sahiptir. Oluşturulan modelde katmanlara göre yapılan işlemler şunlardır:

- **Giriş katmanı:** İlk LSTM katmanı, giriş katmanları ile tam olarak bağlanan 64 nörona sahiptir.
- **İkinci katman:** 128 nörona sahip LSTM katmanıdır. ReLU aktivasyon fonksiyonu uygulanmıştır.
- **Üçüncü katman:** 128 nörona sahip LSTM katmanıdır. ReLU aktivasyon fonksiyonu uygulanmıştır.
- **Dördüncü katman:** 64 MLP nöronu vardır. ReLU aktivasyon fonksiyonu uygulanmıştır.
- **Beşinci katman:** 64 MLP nöronu vardır. ReLU aktivasyon fonksiyonu uygulanmıştır.
- **Çıkış katmanı:** İkinci MLP ve çıkış katmanları arasında hiçbir aktivasyon fonksiyonu yoktur. Tüm veri seti 0-1 arasında ölçeklendirildiği için çıkış katmanının değeri de tersine çevrilmelidir.

Optimizasyon algoritması olarak ADAM kullanılmıştır. Algoritma üzerinde öğrenme oranı olarak 0.001 tercih edilmiştir. Öğrenme esnasında **geri besleme işlemi** (*epoch*) 1000 kez tekrarlanmıştır. Oluşturulan modelin Keras kütüphanesi yardımıyla elde edilen mimarisi Şekil 4.9.' de gösterilmiştir.



Şekil 4.9. Bitcoin fiyat tahmini için oluşturulan LSTM model mimarisi

Elde edilen model sonucunda performans değerleri Çizelge 3.5.' de verilmiştir.

Çizelge 4.4. Bitcoin fiyatı LSTM tahmin modeli için performans değerleri.

Bitcoin fiyatı	RMSE	MAE	MPE	MAPE
LSTM Modeli	93.27	81.56	1.40	1.40

5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

Apple hisse senedi fiyatı ve Bitcoin fiyatına ait finansal veri setleri zaman serilerine dönüştürülerek, Çizelge 3.1 ve Çizege 3.2.'deki algoritmalar yardımıya tahmin modelleri geliştirilmiştir. Geliştirilen tahmin modelleri için Çizelge 2.1.'deki performans ölçüleri kullanılmıştır. Elde edilen modellerden fiyat tahminleri Çizelge 5.1.'de özetlenmiştir.

Çizelge 5.1. Apple hisse senedi fiyatı ve Bitcoin fiyatına ait ARIMA ve LSTM tahmin modelleri için performans değerleri.

	Modeller	RMSE	MAE	MPE	MAPE
Apple hisse fiyatı	ARIMA(5,2,0) Modeli	34.04762	27.01132	-15.38245	15.6444
	LSTM Modeli	9.61	6.48	2.75	5.78
Bitcoin fiyatı	ARIMA(4,2,1) Modeli	1146.067	939.5819	10.86483	11.86484
	LSTM Modeli	93.27	81.56	1.40	1.40

Çizelge 5.1.'de görüldüğü gibi LSTM, Apple hisse senedi fiyatı tahminlemede %5.78 ve Bitcoin fiyatına ait tahmin modellemede %1.40 oranında hata yaparak, ARIMA modellerine karşı daha iyi sonuç vermiştir. Diğer performans ölçülerinde de LSTM daha başarılı gözükmektedir. Bu LSTM mimarisinin, zaman serilerinin tahmininde geleneksel istatistik yöntemlere göre daha uygun olduğunu göstermektedir. Bu sonuçlara dayanarak anlık tahminleme için model olarak LSTM mimarisi seçilerek bir web arayüzü hazırlanmıştır.

5.2. Öneriler

Zaman serileri tek değişkenli ve çok değişkenli olarak karşımıza çıkmaktadır. Bu çalışmada tek değişkenli zaman serileri için istatistiki yaklaşım ve makine öğrenmesi

yaklaşımı ile tahmin modelleri kullanılmıştır. Sonraki aşamada çok değişkenli zaman serileri için benzer karşılaştırmaların yapılacağı çalışmalar gerçekleştirilmesi planlanmaktadır.

Geliştirilen arayüz günlük verilerin tahminlemesi için kullanılabilir. Ancak finansal verilerin gün içerisinde etkin kullanımı önemli olduğu için 1, 5, 15, 30 ve 60 dakikalık anlık veri akışına uygun olarak web ara yüzünün geliştirilmesine devam edilecektir.

KAYNAKLAR

- Adhikari, R. ve Agrawal, R. K., 2013, An Introductory Study on Time Series Modeling and Forecasting, Lambert Academic Publishing.
- Bao, W., Yue, J., Rao, Y., 2017, A deep learning framework for financial time series using stacked autoencoders and long-short term memory, PLoS ONE 12(7): e0180944.
- Bengio, Y., 2009, Learning deep architectures for AI, Foundations and trends in Machine Learning, 2(1):1-127.
- Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y., 2010, Theano: a CPU and GPU math expression compiler, In Proceedings of the Python for Scientific Computing Conference (SciPy)-June 2010.
- Bernal, A., Fok, S., Pidaparathi, R., 2012, Financial Market Time Series Prediction with Recurrent Neural Networks, pp. 1–5.
- Bircan, H., Karagöz, Y., 2003, Box-Jenkins Modelleri ile Aylık Döviz Kuru Tahmini Üzerine Bir Uygulama, Kocaeli Üniversitesi Sosyal Bilimler Enstitüsü Dergisi (6) 2003 / 2, 49-62.
- Björklund, S., Uhlin, T., 2017, Artificial Neural Networks for Financial Time Series Prediction and Portfolio Optimization, Master of Science Thesis in Industrial Engineering and Management Department of Management and Engineering, Linköping University.
- Box, G.E.P. ve Jenkins, G.M., 1970, Time series analysis: Forecasting and control, San Francisco: Holden-Day.
- Brockwell, P. J. ve Davis, R. A., 2002, Introduction to Time Series and Forecasting, Springer-Verlag New York, Inc.
- Chatfield, C., 2000, Time-Series Forecasting, Chapman & Hall/CRC.
- Chollet, F., 2016, Keras, <https://github.com/fchollet/keras>, [Ziyaret Tarihi: 1 Nisan 2018].
- Chong Tan, C., 2009, Financial Time Series Forecasting Using Improved Wavelet Neural Network, Aarhus Universitet, Datalogisk Institut.
- Cochrane, J. H., 1997, Time Series for Macroeconomics and Finance, Graduate School of Business, University of Chicago.
- Collantes-Duarte, J., Rivas-Echeverria, F., 2001, Forecasting using ARIMA, Neural Networks and Neo Fuzzy Neurons.

- Fakhr, W., Baasher, A., 2011, FOREX Daily Trend Prediction using machine learning techniques, ACS'11 Proceedings of the 11th WSEAS International Conference on Applied Computer Science, 41-47.
- Folk, M., 2012, A First Course on Time Series Analysis — Examples with SAS, Chair of Statistics, University of Würzburg.
- Grachev, O., 2017, Application of Time Series Models (ARIMA, GARCH, and ARMA-GARCH) for Stock Market Forecasting, Master Thesis in University Honors Program Baccalaureate Degree Department of Economics.
- Graves, A., 2014, Generating Sequences With Recurrent Neural Networks, <https://arxiv.org/abs/1308.0850>.
- Greff, K., Srivastava, R. K., Koutnik J., Steunebrink B.R., Schmidhuber J., 2015, LSTM: A Search Space Odyssey, IEEE Transactions on Neural Networks and Learning Systems.
- Hagan, M. T., Demuth, H. B., Beale, M. H., De Jesús, O., 2014, Neural Network Design, 2nd Edition.
- Haykin, S., 1999, Neural Networks: A Comprehensive Foundation, Prentice Hall.
- Heaton, J. B., Polson, N. G., Witte, J. H., 2016, Deep Learning in Finance, <https://arxiv.org/abs/1602.06561>.
- Hinton, G. E., Nair, V., 2010, Proceeding ICML'10 Proceedings of the 27th International Conference on International Conference on Machine Learning, Pages 807-814.
- Hochreiter, S., Schmidhuber, J., 1997, Long Short-Term Memory, Neural Computation, 9(8):1735-1780.
- Hu, X. ve Balasubramaniam, P., 2008, Recurrent Neural Networks, In-Teh Publishing.
- Hyndman, R. J. ve Athanasopoulos, G., 2018, Forecasting: Principles and Practice, Otext.
- Jansze, J., 2013, Time Series Machine Learning Technique with Application to Barcelona Metro Station Energy Minimization, Master Thesis in Intelligent systems group Information and Computing Sciences, University Utrecht.
- Kaynar, O. ve Taştan, S., 2009, Zaman Serisi Analizinde MLP Yapay Sinir Ağları ve ARIMA Modelinin Karşılaştırılması, Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Kayseri, (13), 161-172.
- Kingma D. P., Ba J., 2014, Adam: A Method for Stochastic Optimization, <https://arxiv.org/abs/1412.6980>.

- Kreesuradej W.,1993, Time Series Prediction Using Neural Networks, Master Of Science Thesis in Electrical Engineering, Texas Tech University.
- Kuna, K., 2015, Time Series Prediction Using Neural Networks, Bachelor Thesis in Faculty of Informatics, Masaryk University.
- McCulloch, W.S. ve Pitts, W., 1943, A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 5: 115.
- Mitchell, S., 1995, The Application of Machine Learning Techniques to Time-Series Data, Master Thesis in Computing and Mathematical Sciences at the University of Waikato.
- Montgomery, D.C., Jennings, C.L., Kulahci, M., 2008, *Introduction to Time Series Analysis and Forecasting*, John Wiley & Sons Inc.
- Moody J., Saffell, M., 1999, Reinforcement learning for trading. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, David A. Cohn (Ed.). MIT Press, Cambridge, MA, USA, 917-923.
- Morariu, N., Iancu, E., Vlad, S., 2009, A Neural Network Model For Time-Series Forecasting, *Romanian Journal of Economic Forecasting* (4/2009), 213-223.
- Moreno, J. J. M., Pol, A. P., Gracia, P. M., 2011, Artificial neural networks applied to forecasting time series, *Psicothema* 2011, Vol. 23, no 2, pp. 322-329.
- Nakamoto, S., 2008. *Bitcoin: A Peer-to-Peer Electronic Cash System*.
- Navon, A., Keller, Y., 2017, Financial Time Series Prediction using Deep Learning, <https://arxiv.org/abs/1711.04174>.
- Olah, C., 2015, Understanding LSTM Networks, <http://colah.github.io>, [Ziyaret Tarihi: 1 Nisan 2018].
- Oliphant, T. E., 2006, *A guide to NumPy*, USA: Trelgol Publishing.
- R Development Core Team (2008). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>, [Ziyaret Tarihi: 1 Nisan 2018].
- Rosenblatt, F. 1962. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, New York: Spartan.
- Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986, Learning Internal Representations by Error Propagation, *Parallel distributed processing: Explorations in the microstructure of cognition*, Vol: 1, MIT Press.

- Sewell, M. V., 2017, Application of Machine Learning to Financial Time Series Analysis, Doctor Thesis in Philosophy of UCL Department of Computer Science University College London University of London.
- Seymour, S., 1997, The Brain: Our Nervous System, New York: William Morrow.
- Shumway, R. H. ve Stoffer, D. S., 2006, Time Series Analysis and Its Applications With R Examples, Springer Science+Business Media.
- Tay, E. H. F., Cao, L., 2001, Application of support vector machines in Financial time series forecasting, The International Journal of Management Science, Omega 29, 309-317.
- Thomas, F., Christopher, K., 2017, Deep learning with long short-term memory networks for financial market predictions, FAU Discussion Papers in Economics, No. 11/2017.
- Wei, W. W. S., 2006, Time Series Analysis Univariate and Multivariate Methods, Pearson Education.
- Winston, P.H., 1991. Artificial Intelligence. Addison-Wesley Publishing Co., Reading, Mass.
- Wong, W.K., Min, X., Chu, W.C., 2010, Adaptive neural network model for time-series forecasting, European Journal of Operational Research 207, 807–816.
- Zhang, G. P., Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing 50 (2003), 159–175.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Ebru Şeyma Karakoyun
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Karaman, 10.01.1992
Telefon :
Faks :
e-mail : ebruseymakarakoyun@gmail.com

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Karaman Anadolu Lisesi, Merkez, Karaman	2010
Üniversite	: Muğla Sıtkı Koçman Üniversitesi, Merkez, Muğla	2015
Yüksek Lisans	: Necmettin Erbakan Üniversitesi, Meram, Konya	2018
Doktora	:	

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
-----	-------	--------

UZMANLIK ALANI

- **Programlama Dilleri:** Python, R, C
- **İşletim Sistemleri:** MacOS, Windows
- **Uygulama:** Jupyter Notebook
- **Veri tabanı:** MySQL

YABANCI DİLLER

- **İngilizce** (Okuma: İyi, Konuşma: İyi, Yazma: İyi)

BELİRTMEK İSTEDİĞİNİZ DİĞER ÖZELLİKLER

YAYINLAR

- Karakoyun, Ş. ve Çıbıkdiken A. O., 2018, Comparison of ARIMA Time Series Model and LSTM Deep Learning Algorithm for Bitcoin Price Forecasting, *Proceedings of MAC 2018 in Prague*, 171-179.