

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**DERİN ÖĞRENME TEKNİKLERİ KULLANILARAK BORSADAKİ
HİSSE DEĞERLERİNİN TAHMİN EDİLMESİ**

**Hazırlayan
Mehmet OZAN**

**Danışman
Dr. Öğr. Üyesi İlker DALKIRAN**

Yüksek Lisans Tezi

**Temmuz 2021
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**DERİN ÖĞRENME TEKNİKLERİ KULLANILARAK BORSADAKİ
HİSSE DEĞERLERİNİN TAHMİN EDİLMESİ**

**Hazırlayan
Mehmet OZAN**

**Danışman
Dr. Öğr. Üyesi İlker DALKIRAN**

Yüksek Lisans Tezi

**Temmuz 2021
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Mehmet OZAN

İmza

“Derin Öğrenme Teknikleri Kullanılarak Borsadaki Hisse Değerlerinin Tahmin Edilmesi” adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan

Mehmet OZAN

İmza

Danışman

Dr. Öğr. Üyesi İlker DALKIRAN

İmza

Elektrik Elektronik Mühendisliği ABD Başkanı

Prof. Dr. Ömer Galip SARAÇOĞLU

İmza

TEŞEKKÜR

Öncelikle çalışma boyunca bilgi birikimini ve desteğini esirgemeyip her konuda bana destek olan, değerli tavsiyeleri ile bu çalışmayı mümkün kılan danışman hocam sayın Dr. Öğr. Üyesi İlker DALKIRAN 'a teşekkürü bir borç bilirim.

Ayrıca bu günlerimin en büyük mimarları olan değerli annem Berrin OZAN ve babam Osman OZAN 'a, vermiş olduğu manevi destek ile her daim yanımda olan Müge KARACA 'ya ve yapay öğrenme alanında tavsiyeleri ile beni her daim ileriye taşıyan kıymetli Özge ERSÖYLEYEN 'e şükranlarımı sunarım.

Mehmet OZAN

Haziran 2021, KAYSERİ

DERİN ÖĞRENME TEKNİKLERİ KULLANILARAK BORSADAKİ HİSSE DEĞERLERİNİN TAHMİN EDİLMESİ

Mehmet OZAN

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Haziran 2021

Danışman: Dr. Öğr. Üyesi İlker DALKIRAN

ÖZET

Borsa hisse değerlerinin, farklı ve çok miktarda kuvvet tarafından etkilenmesi ve bu kuvvetlerin etkileşimlerinin karmaşık olması sebebiyle tahmin edilmeleri oldukça zordur. Bu kuvvetlerin güçlerindeki ve etkileşimlerindeki ufak değişimler, sonuçlarda büyük değişimler olarak görülmektedirler. Ancak istatistik teorilerinden doğan ve ancak günümüz bilgisayarları ile uygulanması mümkün olan karmaşık derin öğrenme teknikleri ile bu değerlerin kısa dönem tahminleri mümkündür. Çalışmamızda Borsa İstanbul içerisinde işlem görmekte olan ISCTR, VAKBN, GARAN, QNBFB ve AKBNK hisse senetlerinin uzun dönem verilerine bakılarak günlük değer tahmini yapılmıştır.

Bu tez çalışmasında çok değişkenli yapıda uzun kısa-dönem hafıza (Long Short-Term Memory, LSTM) ve kapılı tekrarlayan hücre (Gated Recurrent Unit, GRU) algoritmaları Adam ve RMSProp optimize edicileri ile kullanılmış ve performansları gözlemlenmiştir. Ayrıca ilgili hisselerin verilerine ek olarak, tahmin kabiliyetini artırmak amacı ile USD/TRY, BIST30, BANKX endekslerine ait tarihsel veriler de kullanılmıştır. Yapılan çalışma sonucunda en başarılı tahmin 0.01465 ortalama bağıl hata ile ISCTR için yapılmış olup tüm hisselerle ilişkin tahminlerin ortalama bağıl hatası 0.01633 olarak hesaplanmıştır.

Anahtar Kelimeler: Yapay Zekâ, Derin Öğrenme, Borsa, Regresyon, Tahmin, Yapay Sinir Ağları, Uzun Kısa-Dönem Hafıza, Kapılı Tekrarlayan Hücre, Tekrarlayan Sinir Ağları, Makine Öğrenmesi

STOCK MARKET PREDICTION WITH DEEP LEARNING TECHNIQUES

Mehmet OZAN

Erciyes University, Graduate School of Natural and Applied Sciences

Master Thesis, June 2021

Supervisor: Assistant Professor İlker DALKIRAN

ABSTRACT

Stock market share values are affected by different and many forces and the interactions of these forces are complex. Because of this complexity, they are difficult to predict. Small changes in the strength and interaction of these forces are seen as large changes in results. However, short-term estimations of these values are possible with complex deep learning techniques that arise from statistical theories and can be realized with today's computer technologies. In our study, one-day value estimation of ISCTR, VAKBN, GARAN, QNBFB and AKBNK stocks traded in the Borsa Istanbul has been made.

In the study, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms in multivariate structure were used with Adam and RMSProp optimizers and their performances were observed. In addition to the data of the related stocks, historical data of USD/TRY, BIST30 and BANKX indices were also used in the study in order to increase the estimation ability. As a result of the study, the most successful prediction was made for ISCTR with a mean relative error of 0.01465, and the average relative error of all stocks was calculated as 0.01633.

Keywords: Artificial Intellect, Deep Learning, Stock Market, Regression, Prediction, Artificial Neural Networks, Long Short-Term Memory, Gated Recurrent Unit, Recurrent Neural Networks, Machine Learning

İÇİNDEKİLER

YÖNERGEYE UYGUNLUK.....	iii
KABUL VE ONAY	iv
TEŞEKKÜR	v
ÖZET	vi
ABSTRACT	vii
İÇİNDEKİLER.....	viii
KISALTMALAR.....	xi
TABLolar LİSTESİ.....	xii
ŞEKİLLER LİSTESİ	xiii
GİRİŞ.....	1

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Problem Durumu.....	3
1.2. Araştırmanın Amacı	6
1.3. Araştırmanın Önemi.....	8

2. BÖLÜM

VERİ SETLERİNİN HAZIRLANMASI

2.1. Değeri Tahmin Edilen Borsa Hisselerinin Belirlenmesi	9
2.2. Belirlenen Hisse Senetlerinin Geçmiş Verilerinin Toplanması	9
2.3. Veri Setlerinin Ön İşlenmesi (Pre-processing).....	11
2.3.1 Veri Temizleme (Data Cleansing).....	12
2.3.2. Veri Ölçeklendirme	13
2.3.3. Veri Setlerinin Üç Boyutlu Tensörlere Dönüştürülmesi.....	14

3. BÖLÜM

MODEL SEÇİMİ

3.1 Derin Öğrenme.....	17
3.1.1. Problemin Çözümüne Uygun Modellerin Araştırılması.....	17
3.1.2. Evrimsel Sinir Ağları (CNN).....	18
3.1.3. Tekrarlayan Sinir Ağları	20
3.1.4. Yok Olan Gradyanlar Problemi	21
3.1.5. Uzun Kısa-Dönem Bellek (Long Short-Term Memory, LSTM).....	22
3.1.5. Kapılı Tekrarlayan Hücre (Gated Recurrent Unit, GRU)	26
3.2. Geliştirmede Kullanılan Yazılım Dili ve Kütüphaneler	29
3.2.1. Keras ve TensorFlow Kütüphaneleri	30
3.2.2. Pandas ve Numpy.....	30
3.3. Düğüm Seyreltme (Dropout)	31
3.3.1. Aşırı Öğrenme (Overfitting)	31
3.3.2. Erken Durdurma (Early Stopping)	32
3.4. Gradyan İniş Metodu (Gradient Descent Optimizer).....	32
3.4.1. Öğrenme Oranı (Learning Rate)	34
3.4.2. RMSProp Optimizer	35
3.5. Hata Metrikleri	36
3.5.1. Ortalama Karesel Hata.....	36
3.5.2. Ortalama Mutlak Hata	36
3.5.3. Ortalama Bağıl Hata.....	37

4. BÖLÜM

MODELİN OLUŞTURULMASI

4.1. Model Performanslarının Kıyaslanması	38
4.2. Geliştirme Ortamı	38

4.2.1. Google TPU	40
4.3. Önerilen Model	41
4.4. Önerilen Modele Ait Erken Durdurma (Early Stopping)	43
4.5. Modelin Validasyon ve Test Sonuçları	44

5. BÖLÜM

SONUÇ

5.1 Sonuçların Değerlendirilmesi	68
5.2 Öneriler ve Gelecek Çalışmalar	69
KAYNAKÇA.....	70
ÖZGEÇMİŞ	75

KISALTMALAR

RNN	Tekrarlayan Sinir Ađı (Recurrent Neural Network)
BIST	Borsa İstanbul
API	Uygulama Programlama Arayüzü (Application Programming Interface)
GPU	Grafik İşlemci Birimi (Graphics Processing Unit)
CPU	Merkezi İşlemci Birimi (Central Processing Unit)
TPU	Tensör İşlemci Birimi (Tensor Processing Unit)
MSE	Ortalama Kareysel Hata (Mean Squared Error)
MAE	Ortalama Mutlak Hata (Mean Absolute Error)
MRE	Ortalama Bağlı Hata (Mean Relative Error)
SVM	Destek Vektör Makineleri (Support Vector Machine)
RL	Pekiştirmeli Öğrenme (Reinforcement Learning)
DNN	Derin Sinir Ađı (Deep Neural Network)
LSTM	Uzun Kısa-Dönem Hafıza (Long Short-Term Memory)
GRU	Kapılı Tekrarlayan Hücre (Gated Recurrent Unit)
ELM	Aşırı Öğrenme Makineleri (Extreme Learning Machines)
VAE	Varyasyonlu Otomatik Kodlayıcılar (Variational AutoEncoders)
NaN	Sayı Deđil (Not a Number)
CNN	Evrişimsel Sinir Ađı (Convolutional Neural Network)

TABLULAR LİSTESİ

Tablo 1. 1. İnce'nin yapmış olduğu 0.001 pay için modellerin kesinlik kıyaslaması.....	6
Tablo 1. 2. Site A. tarafından derlenen ve literatürde yer alan çalışmaların listesi	8
Tablo 2. 1. Borsa hisseleri arasındaki korelasyonlar.....	10
Tablo 2. 2. Yahoo Finance kayıtlarına göre hisse verilerinin başlama tarihi.	10
Tablo 4. 1 Farklı hiper-parametre ve algoritmalara karşılık ISCTR hissesi için MSE değerleri.....	38
Tablo 4. 2 Modele ait validasyon ve test sonuçları.....	66

ŞEKİLLER LİSTESİ

Şekil 1. 1. Raşo ve Demirci'nin kullanmış olduğu yapay sinir ağı modeli	4
Şekil 2. 1. ISCTR hissesine ait verilerin tutulduğu DataFrame ait ekran görüntüsü.....	11
Şekil 2. 2. Pandas kütüphanesine ait “merge” işleminin temsili gösterimi.	11
Şekil 2. 3. Vektör, matris ve tensör yapılarının kıyaslaması.....	15
Şekil 2. 4. Model girdimizin şeklinin temsili ifadesi.....	16
Şekil 3. 1. Yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki ilişki.....	18
Şekil 3. 2. CNN mimarisinin modeli.	19
Şekil 3. 4. Basit RNN hücresinin iç yapısı.....	20
Şekil 3. 3. Basit çoktan teke (Many-to-One) RNN yapısı	21
Şekil 3. 5. LSTM Hücresinin iç yapısı	22
Şekil 3. 6. LSTM hücre durumu bilgisinin taşındığı hat	23
Şekil 3. 7. Örnek bir LSTM kapısı.	24
Şekil 3. 8. LSTM modeline ait Unutma kapısı.....	24
Şekil 3. 9. LSTM modeline ait Girdi kapısı.....	25
Şekil 3. 10. LSTM modeline ait Hücre durumunun güncellenmesi	25
Şekil 3. 11. LSTM modeline ait Çıktı kapısı	26
Şekil 3. 12. GRU iç yapısı.....	27
Şekil 3. 13. GRU sıfırlama kapısı.....	28
Şekil 3. 14. GRU güncelleme kapısı Güncelleme Kapısının 1. Aşaması.	28
Şekil 3. 15. GRU Güncelleme Kapısının 2. Aşaması.....	29
Şekil 3. 16. Optimum öğrenme ve Aşırı öğrenme örneğinin grafiksel ifadesi	33
Şekil 3. 17. Örnek bir hata fonksiyonu grafiği.....	34
Şekil 3. 18. Büyük ve küçük öğrenme oranlarının grafik ile izahı.....	35
Şekil 4. 1. TPU tabanlı sistemde eğitim süreci.	40
Şekil 4. 2. GPU tabanlı sistemde eğitim süreci.	40
Şekil 4. 3. Bulut TPU'nun yapısı	41
Şekil 4. 4. Modelimize ait akış şeması.....	42
Şekil 4. 5. Model özetimizin kod çıktısına ait ekran görüntüsü.....	42
Şekil 4. 6. Modelimizin özetinin akış diyagramına ait ekran görüntüsü	43
Şekil 4. 7. Geliştirmiş olduğumuz modele ait erken durdurma kod parçası.	44

Şekil 4. 8. Modelimizin eğitiminde kullanılan AKBNK hissesine ait verilerin grafiği.	45
Şekil 4. 9. Modelimizin eğitiminde kullanılan ISCTR hissesine ait verilerin grafiği. ...	45
Şekil 4. 10. Modelimizin eğitiminde kullanılan GARAN hissesine ait verilerin grafiği.	46
Şekil 4. 11. Modelimizin eğitiminde kullanılan QNBFB hissesine ait verilerin grafiği.	46
Şekil 4. 12. Modelimizin eğitiminde kullanılan VAKBN hissesine ait verilerin grafiği.	47
Şekil 4. 13. Modelimizin eğitiminde kullanılan USD/TRY endeksine ait verilerin grafiği.	47
Şekil 4. 14. Modelimizin eğitiminde kullanılan XBANK hissesine ait verilerin grafiği.	48
Şekil 4. 15. ISCTR hisse senedi için nisan ayına ait fiyat tahminleri.....	49
Şekil 4. 16. ISCTR hisse senedi için nisan ayı test sonuçlarına ait mutlak hata grafiği.	49
Şekil 4. 17. ISCTR hisse senedinin nisan ayı test sonuçlarına ait bağıl hata grafiği.....	50
Şekil 4. 18. ISCTR hisse senedinin mayıs ayına ait test sonuçlarının grafiği.....	50
Şekil 4. 19. ISCTR hisse senedinin mayıs ayı test sonuçlarına ait mutlak hata grafiği.	51
Şekil 4. 20. ISCTR hisse senedinin mayıs ayı test sonuçlarına ait bağıl hata grafiği. ...	51
Şekil 4. 21. AKBNK hisse senedinin nisan ayına ait test sonuçlarının grafiği.	52
Şekil 4. 22. AKBNK hisse senedinin nisan ayı test sonuçlarına ait mutlak hata grafiği.	53
Şekil 4. 23. AKBNK hisse senedinin nisan ayı test sonuçlarına ait bağıl hata grafiği. ..	53
Şekil 4. 24. AKBNK hisse senedinin mayıs ayına ait test sonuçlarının grafiği.	54
Şekil 4. 25. AKBNK hisse senedinin mayıs ayı test sonuçlarına ait mutlak hata grafiği.	54
Şekil 4. 26. AKBNK hisse senedinin mayıs ayı test sonuçlarına ait bağıl hata grafiği. .	55
Şekil 4. 27. GARAN hisse senedinin nisan ayına ait test sonuçlarının grafiği.	56
Şekil 4. 28. GARAN hisse senedinin nisan ayı test sonuçlarına ait mutlak hata grafiği.	56
Şekil 4. 29. GARAN hisse senedinin nisan ayı test sonuçlarına ait bağıl hata grafiği. ..	57
Şekil 4. 30. GARAN hisse senedinin mayıs ayına ait test sonuçlarının grafiği.	57
Şekil 4. 31. GARAN hisse senedinin mayıs ayı test sonuçlarına ait mutlak hata grafiği.	58
Şekil 4. 32. GARAN hisse senedinin mayıs ayı test sonuçlarına ait bağıl hata grafiği. .	58

Şekil 4. 33. QNBFB hisse senedinin nisan ayına ait test sonuçlarının grafiği.....	59
Şekil 4. 34. QNBFB hisse senedinin nisan ayı test sonuçlarına ait bağıl hata grafiği. ...	60
Şekil 4. 35. QNBFB hisse senedinin nisan ayı test sonuçlarına ait mutlak hata grafiği.	60
Şekil 4. 36. QNBFB hisse senedinin mayıs ayına ait test sonuçlarının grafiği.....	61
Şekil 4. 37. QNBFB hisse senedinin mayıs ayı test sonuçlarına ait mutlak hata grafiği.	62
Şekil 4. 38. QNBFB hisse senedinin mayıs ayı test sonuçlarına ait bağıl hata grafiği...	62
Şekil 4. 39. VAKBN hisse senedinin nisan ayına ait test sonuçlarının grafiği.	63
Şekil 4. 40. VAKBN hisse senedinin nisan ayı test sonuçlarına ait mutlak hata grafiği	63
Şekil 4. 41. VAKBN hisse senedinin nisan ayı test sonuçlarına ait bağıl hata grafiği. ...	64
Şekil 4. 42. VAKBN hisse senedinin mayıs ayına ait test sonuçlarının grafiği.	64
Şekil 4. 43. VAKBN hisse senedinin mayıs ayı test sonuçlarına ait mutlak hata grafiği.	65
Şekil 4. 44. VAKBN hisse senedinin mayıs ayı test sonuçlarına ait bağıl hata grafiği. .	65

GİRİŞ

Borsa; menkul kıymet, emtia, döviz, vadeli işlem ve opsiyon sözleşmelerinin halka açık olarak alınıp satıldığı denetlenen bir piyasadır. Sermaye piyasası, üretim faaliyetlerini desteklemek için bir yatırım mekanizması oluşturmuştur ve borsa bu amaca ulaşmak için kullanılan piyasadır. Hisse senedi piyasası, yatırımcı faaliyetleri için güvenli bir ortamın yanı sıra düzenli ticaret mekanizmaları, en son fiyatlar ve gerçek zamanlı alışveriş sağladığı için popülerdir.

Borsa veya sermaye piyasalarının tahmini, büyük kâr imkânı sunar ve bu alandaki çoğu araştırmacı için bu durum temel bir motivasyon kaynağıdır. Pazarı tahmin etmek için çoğu araştırmacı ya teknik ya da temel analizi kullanır. Teknik analiz, gelecekteki fiyatları tahmin etmek için fiyatların yönünü analiz etmeye odaklanırken, temel analiz, finansal haberler ve kazanç raporları gibi yapılandırılmamış metin bilgilerini analiz etmeye dayanır.

Bu çalışmada klasik teknik ve temel analize alternatif olarak borsa hisse değerlerinin derin sinir ağlarının zaman serilerinin tahmininde kullanılmak üzere özelleştirilmiş bir versiyonu olan tekrarlayan sinir ağları (Recurrent Neural Network, RNN) kullanılarak tahmin edilmesi amaçlanmıştır. Gelecek fiyatları tahmin edilecek hisseler, Borsa İstanbul bünyesinde BIST100 ve XBANK endeksleri içerisinde işlem görmekte olan aşağıdaki bankalara ait hisseler olarak seçilmiştir.

- Türkiye İş Bankası A.Ş (ISCTR),
- Türkiye Garanti Bankası A.Ş (GARAN),
- Türkiye Vakıflar Bankası T.A.O (VAKBN),
- QNB Finansbank A.Ş (QNBFB),
- Akbank T.A.Ş (AKBNK)

Bu tez çalışmasında veri seti olarak ilgili hisselerin her bir güne ait açılış değeri, kapanış değeri, gün içi en yüksek değer, gün içi en düşük değer ve işlem hacmi verileri belirlenmiş olup Yahoo Finance internet sitesinden alınmıştır. 31 Aralık 2007 tarihinden 28 Şubat 2021 tarihine kadar olan ilgili veriler eğitim (train) verisi, 01 Mart 2021 tarihinden 31 Mart 2021 tarihine kadar olan veri, doğrulama (validation) verisi ve 01 Nisan 2021 tarihinden 31 Mayıs 2021 tarihine kadar olan veriler ise modelin performansını ölçmek amacıyla test verisi olarak kullanılmıştır.

Bu tez çalışması bünyesinde yer alan yapay zekâ modellerine ait kodlar, Python yazılım dili ile yazılmış olup veri setlerinin hazırlanmasında Pandas ve Numpy kütüphaneleri kullanılmıştır. Yapay zekâ modelinin hazırlanmasında TensorFlow kütüphanesinden türetilmiş Keras kütüphanesi kullanılmıştır. Yapay zekâ modelleri, bir Google bulut hizmeti olan Colab üzerinde eğitilmiş olup, alt yapı olarak Google'ın tahsis etmiş olduğu Tesla T4 model GPU ve Google'ın yapay öğrenme alanlarında kullanılmak üzere tasarlanmış olduğu yeni nesil bir işlemci modeli olan TPU (Tensor Processing Unit) kullanılmıştır.

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Problem Durumu

Yapay zekâ ve derin öğrenme kavramları asıl popülerliğini günümüzde kazanmış olsa da temelleri çok daha eskiye dayanmaktadır. Derin sinir ağlarının temel taşı olarak adlandırabileceğimiz Perceptronların ilk temeli 1957 yılında Frank Rosenblatt tarafından Donald Hebb'in beyin hücresi modeli ve Arthur Samuel'in makine öğrenimi çalışmalarının ışığında atılmıştır. Perceptron, başlangıçta donanımsal olarak tasarlanmış olsa da sonradan ilk Perceptron yazılımı IBM 704 için tasarlanmıştır.

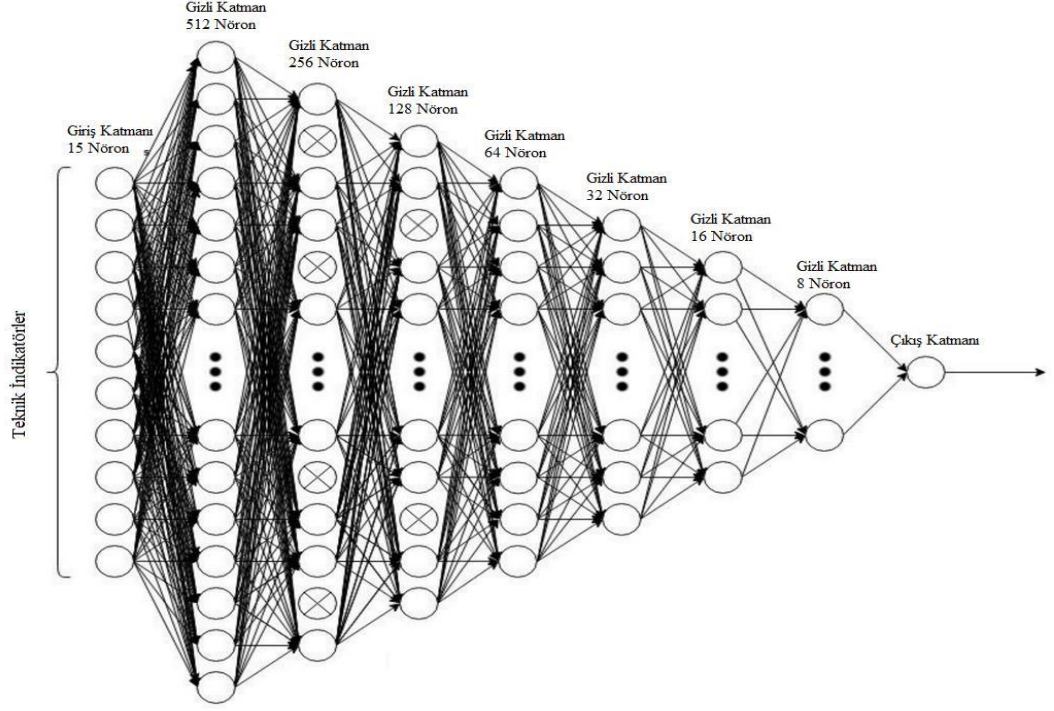
1960'lı yıllarda Perceptronların çok katmanlı kullanılması sinir ağları çalışmalarına yeni bir boyut kazandırmıştır. Perceptronların birden çok katmanda kullanılmasının işlem gücüne önemli ölçüde katkıda bulunduğu keşfedilmesi ile ileri beslemeli sinir ağları (Feedforward Neural Networks) ve geri yayılım (Backpropagation) kavramları önem kazanmıştır.

Makine öğrenmesi, yapısal işlev olarak öğrenebilen ve veriler üzerinden tahmin yapabilen algoritmaların çalışma ve inşalarını araştıran yapay zekanın bir alt dalıdır. Bu başlık altındaki öğrenme algoritmaları, statik program talimatlarını harfiyen takip etmek yerine örnek girişlerden veri tabanlı tahminleri ve kararları gerçekleştirebilmek amacıyla bir model oluşturarak çalışırlar. Makine öğrenimi oldukça fazla kullanım alanına sahiptir. Pazarlama, finans, bankaların geçmiş verilerinin analiz edilerek kredi skorlarının hesaplanması, dolandırıcılık tespiti, sinyal filtreleme, yüz tanıma sistemleri, görüntü işleme ve borsa bu alanlara örnek olarak gösterilebilir. İmalat alanında makine öğrenimi modelleri, optimizasyon, kontrol ve sorun çözme alanlarında kullanılırken tıp alanında teşhis konulmasına yardım edilmesi için kullanılmaktadır. Bir sistemin zeki olması için öğrenebilme kabiliyetine sahip olması beklenmektedir. Eğer bir sistem öğrenebilir ve

değişikliklere uyum sağlayabilirse tasarımcının tüm olası durumları düşünüp her biri için çözüm üretmesine gerek kalmayacaktır [1].

Moghar, S&P 500 borsasına ait GOOGL ve NKE hisseleri üzerinde yapmış olduğu çalışmada, Uzun Vadeli Borsa (Long Term Stock Exchange, LTSE) tabanlı bir RNN kullanımını önermektedir. 12 iterasyon ile eğitilen bir setin GOOGL için 264sn ve NKE için 132sn de eğitildiğini ve GOOGL için 0.0011 ve NKE için 0.0019 ortalama karesel hata (Mean Squared Error, MSE) cinsinden hata ile gelecek olası hisse değerlerini tahmin etmiştir. Eğitim için gereken iterasyon sayısı 100'e çıkarıldığında algoritmanın GOOGL için 2200sn ve NKE için 1100sn de eğitildiğini MSE cinsinden hataların sırası ile $4,97E-04$ ve $8,74E-04$ 'e kadar gerilediğini belirtmiştir [2]

Raço ve Demirci, yapmış olduğu çalışmada Şekil 1.1.'de blok yapısı verilmiş olan dokuz katmanlı Destek Vektör Makinesi (Support Vector Machine, SVM) tipi bir yapay sinir ağı kullanarak BIST 30 hisse değerleri üzerinde tahminler yapmaya çalışmışlardır. 01.01.2016'dan 11.04.2018'e kadar olan borsa hisselerinin değerlerini veri seti olarak kullanmışlar ve 5 günlük borsa hisse değerlerini tahmin etmeye çalışmışlardır. Kullanmış oldukları yapay sinir ağının giriş değerlerini belirlerken ekonomi yöntemlerini kullanarak hesaplamış oldukları indikatör ve osilatör değerlerinden faydalanmışlardır. Sonuç olarak birinci günden beşinci güne kadar sırası ile 0,0322; 0,109; 0,09; 0,1069; 0,2581 MSE cinsinden hata ile tahminler yapmışlardır. [2].



Şekil 1. 1. Raşo ve Demirci'nin kullanmış olduğu yapay sinir ağı modeli [2].

Konstantinou, yapmış olduğu çalışmada, borsa hisse senedi değerleri üzerinde Japon Mum Grafikleri, Monte-Carlo modelleri ve binomal modeller gibi klasik metotlar yerine makine öğrenme tekniklerini kullanmayı denemiştir ve Pekiştirmeli Öğrenme (Reinforcement Learning, RL) modelini esas alan bir ileri beslemeli yapay sinir ağı kullanmıştır. Veri seti olarak Kasım-2016'dan Haziran-2017 tarihine kadar Stocholm borsasında işlem gören hisseleri tercih etmiştir. Kullanmış olduğu modelde veri setlerindeki aralığı iki dakika, on dakika, otuz dakika ve bir saat olarak kullandığında hep aynı doğrulukla tahminlerin oluştuğunu gözlemlemiştir. Oluşturmuş olduğu model ile ortalama %8 kazanç elde edildiğini raporlamıştır [4].

İnce yapmış olduğu çalışmada, Bitcoin' in olası değerlerini farklı derin öğrenme yöntemleri kullanarak tahmin etmeye çalışmıştır ve açık, kapalı, düşük değer, hacim gibi indikatörlerin kullanılmasının ileri beslemeli Derin Sinir Ağı (Deep Neural Network, DNN), LSTM ve GRU gibi modellerin kesinliğini arttırdığını gözlemlemiştir. Ek olarak bu modeller arasında yaptığı kıyaslamada Tablo 1. 1'de görülen değerlere ulaşmıştır [5].

Tablo 1. 1. İnce'nin yapmış olduğu 0.001 pay için modellerin kesinlik kıyaslaması [5].

Algoritma	Teknik İndikatörler ile Kesinlik Skoru	OHLCV Değerleri ile Kesinlik Skoru
Derin Sinir Ağı (DNN)	%54,01	%50,21
Uzun Kısa – Dönem Hafıza (LSTM)	%56,33	%53,62
Kapılı Tekrarlayan Hücre (GRU)	%54,05	%53,62

Linzie, yaptığı çalışmada, yapay sinir ağlarını kullanarak finansal analiz ve kısa vadeli borsa tahmini yapmaya çalışmıştır. Literatürdeki diğer çalışmaların aksine teknik indikatörleri kullanmamış ve stokastik gradyen inişi (Stochastic Gradient Descent) içeren bir Aşırı Öğrenme Makinesi (Extreme Learning Machine, ELM) kullanmıştır. Elde ettiği sonuçlarda algoritmanın yalnızca işlem hacmi, gün içi yüksek değer, gün içi düşük değer gibi az sayıda girdiye sahip olması sebebiyle diğer metotlardan yaklaşık olarak %18 daha hızlı eğitildiğini ve ürettiği sonuçlarda oluşan hataların ortalama %0,6 kadar daha düşük olduğunu raporlamıştır [6].

Falinoouss, yapmış olduğu çalışmada borsa hisse değerleri hakkında çıkan haberlerin metin madenciliği ismi verilen ve yazılı metinlerin içeriklerinin analiz edilerek sayısal değerlere dönüştürülmesini sağlayan metotlar kullanılarak sayısal olarak anlamlı değerlere dönüştürülebileceğini ve bulunan değerlerin yapay sinir ağlarında giriş değeri olarak kullanılabileceğini keşfetmiştir [7].

Gündüz ve arkadaşları yapmış oldukları bir çalışmada Borsa İstanbul içerisinde işlem gören bankaların hisselerini LSTM ve Light-GBM modellerini kullanarak tahmin etmeye çalışmış ve bu iki modelin performanslarını karşılaştırmışlardır. Hisse öz niteliklerinin sayısını azaltmak için Değişken Otokodlayıcı (Variational AutoEncoders, VAE) kullanmış ve elde ettikleri en iyi başarı değerini LSTM modeli için makro ortalama F-Skoru cinsinden 0.685 olarak belirtmişlerdir [8].

Site A., 2020 yılında yapmış olduğu tez çalışmasında borsa hisselerinin tahmini üzerine literatüre girmiş çalışmaları Tablo 1.2'de görüldüğü şekilde özetlemiştir [9].

1.2. Araştırmanın Amacı

Finans alanında, borsalar ve onların trendleri doğaları gereği fazlasıyla değişkenliğe ve volatiliteye sahiptir. Yatırımcılar ve borsa analistleri, borsa hisse değerlerinin davranışlarını farklı yöntemler kullanarak incelemekte ve bunlara uygun yatırım planları geliştirmektedirler. Fama E. Verimli Pazar Hipotezinin (Efficient Market Hypothesis) yarı güçlü formunda borsa hisse değerlerinin halka açık tüm verileri yansıttığını ve bu nedenle trend hareketlerini tahmin etmenin mümkün olmadığını belirtmiştir [10].

Bu sebeple yapmış olduğumuz tez çalışmada BIST100 ve Bankacılık endeksleri içerisinde işlem görmekte olan ISCTR, GARAN, VAKBN, QNBFB ve AKBAN hisselerinin değerlerinin günlük tahminlerinin yüksek bir kararlılık ile tahmin edilmesi amaçlanmıştır.

1.3. Araştırmanın Önemi

Borsada işlem gören hisselerin fiyat hareketlerinin öngörülebilirliği amaçlanmış ve bu amaca uygun olarak da derin öğrenme yöntemleri uygulanmış, bu alanda literatürde önceden yapılmış çalışmalara kıyasla performans iyileştirmesi ortaya konulmuştur.

Aynı zamanda akademik çalışmalarda sıklıkla kullanılmakta olan ve yenilemeli sinir ağlarının (RNN) özelleştirilmiş bir versiyonu olan LSTM ile yine RNN'lerin özelleştirilmiş bir başka versiyonu olan GRU modeli de tahminlerde kullanılmış ve bu iki modelin tahmin performansları karşılaştırılmıştır.

Tablo 1. 2. Site A. tarafından derlenen ve literatürde yer alan çalışmaların listesi [9].

Yazarlar	Veri Seti	Tarih Aralığı	Öz Nitelik Seti	Hedef	Metot
Kara ve diğerleri., 2011	BIST 100	-	Teknik İndikatörler	Endeks Hareketi	ANN, SVM
Patel ve diğerleri., 2015	4 Hint Hisse Senedi ve 2 Farklı Endeks	2003-2010	Teknik İndikatörler	Hisse Fiyat Hareketi	ANN, SVM, RF, NB
Gündüz ve diğerleri., 2017	BIST 100 en fazla işlem gören 3 hisse	2011-2016	Teknik İndikatörler	Hisse Fiyat Hareketi	LR, Gradyen Güçlendirme Makineleri
Nayak ve diğerleri., 2016	Üç farklı sektörden Hint hisseleri ve Haber kaynakları	2002-2015	Polarite Skorları	Hisse Fiyat Hareketi	Güçlendirilmiş Karar Ağaçları, LR, SVM
Vargas ve diğerleri., 2017	S&P 500 Borsasından farklı hisse ve metinsel kaynaklar	2006-2013	Teknik İndikatörler ve Polarite Skorları	Hisse Fiyat Hareketi	CNN, RNN, ANN, SVM
Hiransha ve diğerleri., 2018	NSE'den 3 ve NYSE'den 2 farklı hisse	2007-2017	-	Hisse Fiyat Tahmini	LSTM, CNN, RNN, MLP, ARIMA
Kim ve diğerleri., 2019	S&P 500 Borsasından farklı hisse ve metinsel kaynaklar	2016-2017	CNN ile çıkarılmış öz nitelikler	Hisse Fiyat Tahmini	CNN, LSTM, nitelik hibritlemeli CNN - LSTM Modeli
Dong ve diğerleri., 2019	Çin Borsasına ait tarihsel veriler	2011-2016	Arama motoru endeks oranları	Hisse Fiyat Tahmini	ARIMA, ANN, SVR ve önermiş oldukları metotlar (LBSI ve TS-GFT)

2. BÖLÜM

VERİ SETLERİNİN HAZIRLANMASI

Bu bölümde, tez çalışması dahilinde eğitimi yapılan derin öğrenme modelleri için gerekli olan borsa hisselerinin belirlenmesinden, belirlenmiş olan hisselerle ilişkin hangi verilerinin nasıl toplandığından bahsedilmiştir. Toplanan verilerin derin öğrenme modellerine giriş olarak uygulanmadan önce ve uygulandıktan sonra yürütülen ön işlemlere (Pre-processing) ve son işlemlere (Post-processing) dair bilgi verilmiştir.

2.1. Değeri Tahmin Edilen Borsa Hisselerinin Belirlenmesi

Bu tez çalışmasında eğitimi yapılan derin öğrenme modellerinin kararlılığını artırmak ve tahmin başarısını artırmak adına volatilitelerinin düşük ve işlem hacimlerinin büyük olması sebebiyle BIST30 ve Borsa İstanbul Bankacılık Endeksi (XBANK) içerisinde yer alan ISCTR, AKBNK, GARAN, VAKBN ve QNBFB hisseleri seçilmiştir.

Günlük fiyat hareketleri tahmin edilen hisselerle ek olarak tahmin başarısını artırmak için veri seti içerisinde XBANK endeksi, BIST30 endeksi, Türk lirası / Amerikan doları endeksi dahil edilmiştir. Irmak ve Çetin'in yapmış olduğu çalışma [11], borsa içerisinde işlem gören hisselerin birbirleri ile olan korelasyonlarını Tablo 2.1'de ortaya koymuştur. Bu sebeple ISCTR ve AKBNK hisselerinin tahminlerini destekleyeceği sebebiyle İş Gayrimenkul Yatırım Ortaklığı A.Ş (ISGYO) ve Aksigorta A.Ş (AKGRT) hisselerine ait veriler de veri setine dahil edilmiştir.

2.2. Belirlenen Hisse Senetlerinin Geçmiş Verilerinin Toplanması

Eğitim ve test aşamasında ihtiyaç duyulan veriler, Yahoo Finance tarafından sağlanmış olan yfinance API'sı aracılığı ile toplanmıştır. Toplanan verilerin yapısına ait ekran görüntüsü Şekil 2.1 de yer almaktadır. İlgili hisse ve endekslerin başlangıç tarihleri Tablo

2.2’de verilmiştir. Tabloda görüleceği üzere veri setinde kullanılan endeks ve hisselerin tamamında ortak olması sebebiyle verinin başlangıç tarihi 31-12-2007 olarak seçilmiştir. Elde edilen veriler, Pandas kütüphanesine ait “DataFrame” veri yapılarında tutulmuştur.

Tablo 2. 1. Borsa hisseleri arasındaki korelasyonlar [11].

	AKBNK	AKGRT	ARCLK	BEKO	DOHOL	FROTO	GARAN	HURGZ	ISCTR	ISGYO	KCHOL	SAHOL
AKGRT	0,676											
ARCLK	0,576	0,558										
BEKO	0,515	0,582	0,659									
DOHOL	0,613	0,600	0,597	0,632								
FROTO	0,508	0,454	0,484	0,445	0,514							
GARAN	0,660	0,645	0,559	0,580	0,659	0,498						
HURGZ	0,617	0,642	0,544	0,591	0,614	0,504	0,647					
ISCTR	0,738	0,685	0,529	0,608	0,659	0,524	0,732	0,678				
ISGYO	0,496	0,633	0,536	0,519	0,519	0,436	0,512	0,563	0,568			
KCHOL	0,600	0,646	0,602	0,592	0,664	0,482	0,622	0,631	0,661	0,552		
SAHOL	0,689	0,661	0,626	0,604	0,732	0,544	0,685	0,620	0,712	0,557	0,713	
TOASO	0,588	0,620	0,567	0,546	0,612	0,455	0,561	0,581	0,562	0,522	0,610	0,651

Tablo 2. 2. Yahoo Finance kayıtlarına göre hisse verilerinin başlama tarihi.

Sembol	Başlangıç Tarihi
ISCTR	31-12-2007
ISGYO	10-05-2000
AKBNK	10-05-2000
AKGRT	10-05-2000
GARAN	10-05-2000
QNBFB	10-05-2000
VAKBN	21-11-2005
XBANK	10-05-2000
BIST30 (XU030)	02-01-1997
TRY/USD	03-01-2005

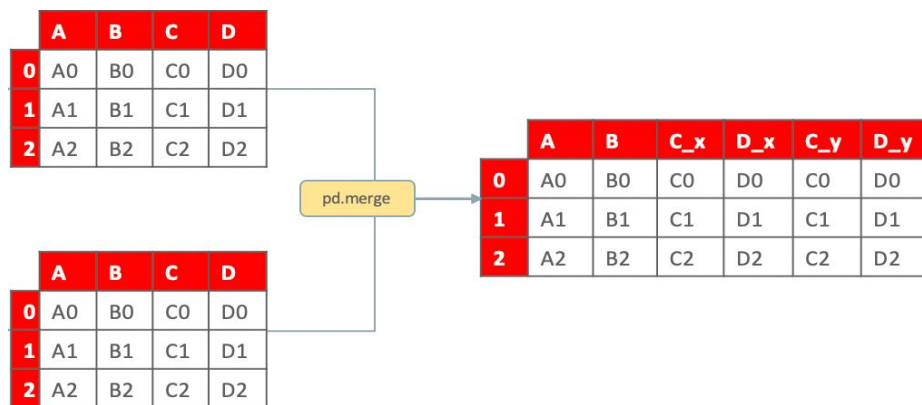
	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2007-12-31	3.505488	3.505488	3.435378	3.435378	8257971	0.0	0.0
2008-01-02	3.412008	3.435378	3.365268	3.388638	7422347	0.0	0.0
2008-01-03	3.341898	3.365268	3.271788	3.318528	18273281	0.0	0.0
2008-01-04	3.341898	3.365268	3.318528	3.341898	8678921	0.0	0.0
2008-01-07	3.271788	3.365268	3.248418	3.318528	23812232	0.0	0.0
...
2021-05-21	5.200000	5.220000	5.130000	5.150000	36348997	0.0	0.0
2021-05-24	5.150000	5.150000	5.070000	5.140000	40902461	0.0	0.0
2021-05-25	5.160000	5.210000	5.020000	5.020000	85072394	0.0	0.0
2021-05-26	5.050000	5.160000	5.020000	5.160000	48849318	0.0	0.0
2021-05-27	5.140000	5.190000	5.100000	5.120000	44799699	0.0	0.0

3464 rows × 7 columns

Şekil 2. 1. ISCTR hissesine ait verilerin tutulduğu DataFrame ait ekran görüntüsü.

2.3. Veri Setlerinin Ön İşlenmesi (Pre-processing)

Öncelikle Yahoo Finance aracılığı ile toplanmış olan veriler, tek bir veri setini oluşturmak amacı ile birleştirilmiştir. Bu işlem yapılırken Pandas kütüphanesinin “birleştirme” (merge) özelliği kullanılmıştır. Bu işlemin çalışma mantığı Şekil 2. 2 ile gösterilmiştir. Bu işlem kullanılarak farklı hisse ve endeks değerlerine ait tablolarda ki veriler, tarih indeksi baz alınarak kolon bazında soldan sağa doğru birleştirilmiştir.



Şekil 2. 2. Pandas kütüphanesine ait “merge” işleminin temsili gösterimi.

Sonrasında sırasıyla eksik veriler temizlenmiş (Data Cleansing), ölçeklendirilmiş (Scaling) ve kayar pencere yapısı kullanılarak derin öğrenme modellerine uygun üç boyutlu bir tensör oluşturulmuştur.

2.3.1 Veri Temizleme (Data Cleansing)

Veri temizleme, bir kayıt kümesinden, tablodan veya veri tabanından bozuk veya yanlış kayıtları tespit etme ve düzeltme (veya silme) işlemidir. Birden çok veri kaynağı birleştirilirken verilerin çoğaltılması veya yanlış etiketlenmesi gibi birçok ihtimal vardır. Veriler yanlışsa, sonuçlar ve algoritmalar doğru görünseler bile güvenilirmezdir. Veri temizleme sürecinde adımları belirlemenin tek bir mutlak yolu yoktur. Çünkü işlemler, veri kümesinden veri kümesine farklılık gösterir. Veri temizliğinin yapılmasında asıl amaç, veri kalitesini artırmaktır [12].

Bir verinin kaliteli olarak tanımlanabilmesi için karşılaması gereken kriterlerden bazıları aşağıda verilmiştir [12].

- **Geçerlilik (Validty):** Verilerin tanımlanmış iş kurallarına veya kısıtlamalarına uygunluk derecesi.
- **Doğruluk (Accuracy):** Verilerin gerçek değerlere yakın olma derecesi.
- **Bütünlük (Completeness):** Gerekli tüm verilerin bilinme derecesi.
- **Tutarlılık (Consistency):** Aynı veri kümesi içinde veya birden çok veri kümesi arasında verilerin tutarlı olma derecesi.
- **Tekdüzelik (Uniformity):** Verilerin aynı ölçü birimi kullanılarak belirtilme derecesi.

Belirtilen kriterlerin sağlanması için veri temizliğinin yapılması gereklidir. Daha önce de bahsedildiği üzere veri temizliği için tercih edilebilecek tek bir doğru yolun olmaması sebebiyle uygulanabilecek metotlar şu şekilde sıralanabilir [13];

- **Tekrarlanan veya alakasız kayıtların silinmesi:** Yinelenen kayıtlar sıklıkla veri toplama sırasında gerçekleşir. Birden çok yerden veri kümeleri birleştirildiğinde yinelenen veriler oluşabilmektedir. Tekilleştirme, bu süreçte dikkate alınması gereken en büyük alanlardan biridir.

- **Yapısal hataların düzeltilmesi:** Verinin yapısında oluşan hatalar olarak ifade edilebilirler. Veri içerisinde yapılan yazım yanlışları veya sayısal ifadelerden oluşan bir veri grubunda farklı tipteki veriler örnek olarak gösterilebilir.
- **İstenmeyen aykırı değerlerin filtrelenmesi:** Genellikle, ilk incelemede göze batan ve veri grubunun geneline uymayan verilerin elenmesidir.
- **Eksik verileri işleme:** Veri grubu içerisindeki giriş-çıkış setlerinin eksik kısımlarını görmezden gelmek mümkün değildir. Çünkü birçok algoritma eksik değerleri kabul etmeyecektir. Eksik verilerle başa çıkmanın birkaç yolu vardır.
 - İlk seçenek olarak, eksik değerlere sahip gözlemler veri grubundan çıkarılabilir. Ancak bunu yapmak bilgi kaybı anlamına gelir ve bu durum bu göz önünde bulundurulmalıdır.
 - İkinci bir seçenek olarak, diğer gözlemlere dayalı olarak eksik değerler türetilir. Bu durumda ise gerçek gözlemlerden değil de varsayımlardan hareket edilir ve veri bütünlüğünün kaybedilme riski vardır.
 - Üçüncü bir seçenek olarak, boş değerlerde etkili bir şekilde gezinmek için verilerin kullanılma şeklini değiştirebilirsiniz.

2.3.2. Veri Ölçeklendirme

Ham verilerin aldığı değerlerin büyük bir aralıkta değişmesi durumunda bazı makine öğrenimi algoritmaları, veriler üzerinde normalleştirme veya ölçeklendirme işlemleri yapılmadan başarılı sonuçlar üretemeyebilir. Veri ölçeklemenin uygulanmasının bir başka nedeni de gradyan inişinin veri ölçekleme ile veri ölçekleme olmaması durumuna kıyasla çok daha hızlı yakınsamasıdır. Gradyan inişin tanımını 3.Bölümde verilmiştir. Veri ölçeklendirmede kullanılan birden fazla metot mevcuttur [14].

- **Yeniden ölçeklendirme (Min-Max normalization):** Min-maks ölçekleme veya min-maks normalleştirme olarak da bilinen en basit yöntemdir. Verilerin [0,1] veya [-1,1] aralığına yeniden ölçeklendirilmesidir. Hedef aralığın seçilmesi, verilerin doğasına bağlıdır. [0, 1] için genel formül şu şekilde verilir:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2.1)$$

- **Ortalama normalleştirme:** Ortalama normalleştirmenin formülü şu şekilde verilir:

$$x' = \frac{x - \text{average}(x)}{\text{max}(x) - \text{min}(x)} \quad (2.2)$$

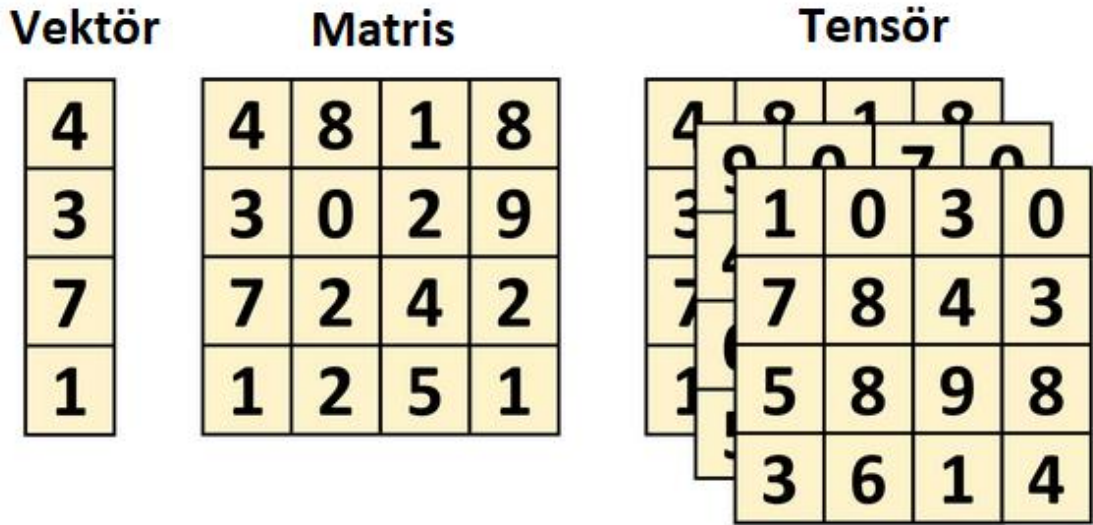
- **Standardizasyon (Z-Skoru normalleştirme):** Makine öğrenimi algoritmalarında çok boyutlu verilerin kullanılması gerektiği durumlarda tercih edilen bir yöntemdir. Veri standardizasyonu, verilerdeki her bir özelliğin değerlerinin sıfır ortalamaya (paydaki ortalama çıkarılırken) ve birim varyansa sahip olmasını sağlar. Yapay sinir ağlarında sıklık ile başvurulan bir yöntemdir. Genel hesaplama yöntemi, her bir özellik için dağılım ortalamasını ve standart sapmayı belirlemektir. Ardından, her bir özellikten ortalama çıkarılır ve her özelliğin değerleri (ortalama zaten çıkarılmıştır) standart sapmasına bölünür. Formülü şu şekilde verilir:

$$x' = \frac{x - \bar{x}}{\sigma} \quad (2.3)$$

$$\bar{x} = \text{average}(x) \quad (2.4)$$

2.3.3. Veri Setlerinin Üç Boyutlu Tensörlere Dönüştürülmesi

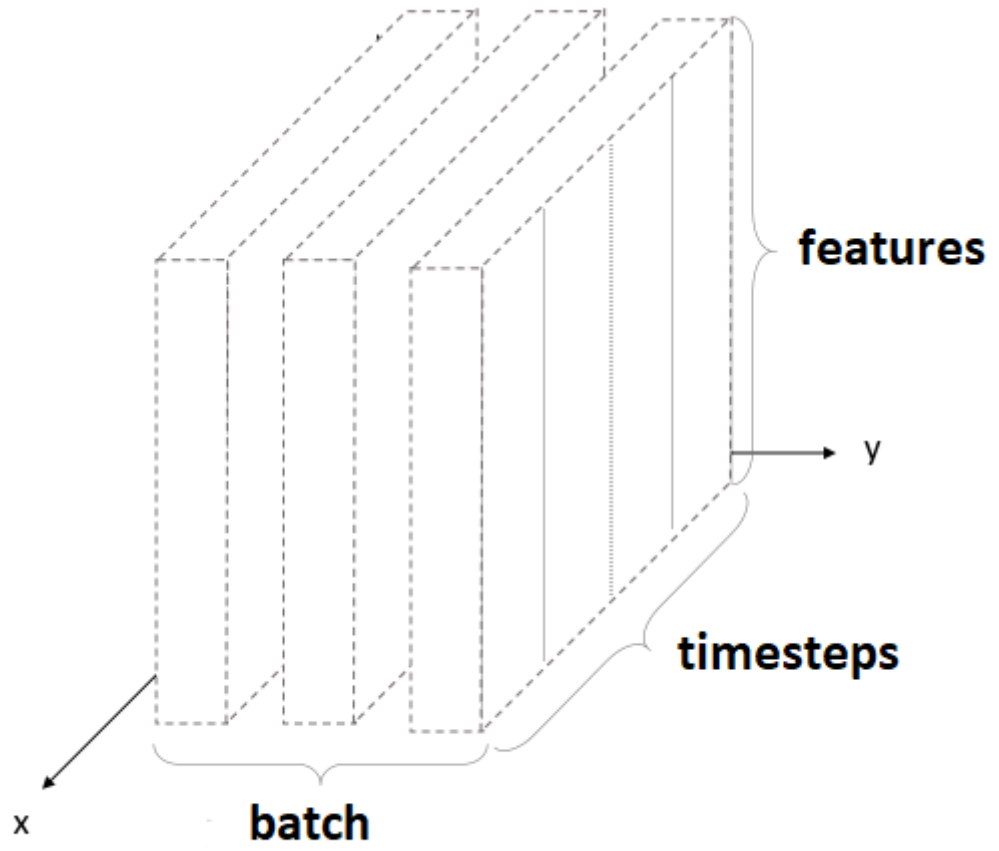
Veri hazırlık aşamasının son adımını ise elde edilen veri setinin derin öğrenme modeline uygun bir girdi haline getirilmesi için üç boyutlu bir tensöre dönüştürülmesidir. Bunun gerekliliğinden bahsetmeden önce anlatımı desteklemek adına vektör, matris ve tensör kıyaslaması görsel olarak Şekil 2.3 ile verilmiştir.



Şekil 2. 3. Vektör, matris ve tensör yapılarının kıyaslaması.

Tensör kavramını daha iyi ifade etmek gerekirse; tensör, matematikte çok boyutlu verinin simgelenebildiği geometrik bir nesnedir. Skaler denilen yönsüz nicel büyüklükler, vektör denilen yönlü büyüklükler ve matris denilen iki boyutlu nesnelere birer tensördür. Tensör, tüm bu nesnelere genelleştirilmiş halidir ve çok boyutlu veri kümeleri için kullanılır [15].

Bu tez çalışmasında derin öğrenme modeli olarak Keras kütüphanesini temel alan LSTM ve GRU modelleri -detayları bir sonraki bölümde verilmiştir- kullanılmıştır. Kütüphane sayfası incelendiği zaman önerilen modelin LSTM için kabul edeceği girdi şekli “[batch, timesteps, feature]” olarak ifade edilmiştir. Bu ifadeler, sırası ile veri setinde kullanılması tercih edilen kayar pencerelerin uzunluğunu, veri setinin sahip olduğu toplam zaman adımını bir başka ifade ile toplam kayıt sayısını ve önerilen model içerisinde kullanmaya karar verilen hisse ve endeks verilerinin öz niteliklerinin toplam sayısını ifade etmektedir. GRU modeli için de girdi şekli tam olarak aynıdır. Önerilen modele girdi olarak verilen veri setinin son şekli bu bilgiler doğrultusunda Şekil 2.4 ile temsil edilmiştir.



Şekil 2. 4. Model girdimizin şeklinin temsili ifadesi.

3. BÖLÜM

MODEL SEÇİMİ

3.1 Derin Öğrenme

Makine öğrenmesinin (Machine Learning) bir alt dalı olan derin öğrenme (Deep Learning), insan beynindeki sinir hücrelerinin davranış ve öğrenme şeklinden esinlenerek tasarlanmış algoritmalar topluluğu olarak tanımlanabilir. Derin öğrenme, aslında bir makine öğrenmesi yöntemidir. Derin öğrenme modellerini kullanarak herhangi bir veri setinin çıktıları üzerinde tahminler yapılabilir. Derin öğrenme modelleri, derin öğrenme algoritmaları kullanılarak eğitilirler ve derin öğrenme algoritmaları her çalıştırıldığında model parametreleri güncellenerek modelin tahmin performansı geliştirilmeye çalışılır. Bu durum, insan beyнинin bir durumu veya olayı her incelediğinde farklı fikirler üretmesi gibi düşünülebilir. Ayrıca derin öğrenme modelleri, öğrenmenin sağlanması için çok fazla miktarda veri setine gereksinim duyarlar [16].

3.1.1. Problemin Çözümüne Uygun Modellerin Araştırılması

Yapılan literatür taraması neticesinde zaman serilerinin tahmini konusunda farklı makine öğrenimi ve derin öğrenme modellerinin kullanıldığı görülmüştür. Bu tez çalışmasının ilk aşamasında, kullanılacak yöntem olarak seçilen derin öğrenme başlığı altında incelenebilecek modeller ise RNN, LSTM, GRU ve CNN (Convolutional Neural Network, Evrişimsel Sinir Ağı) olarak görülmüştür.

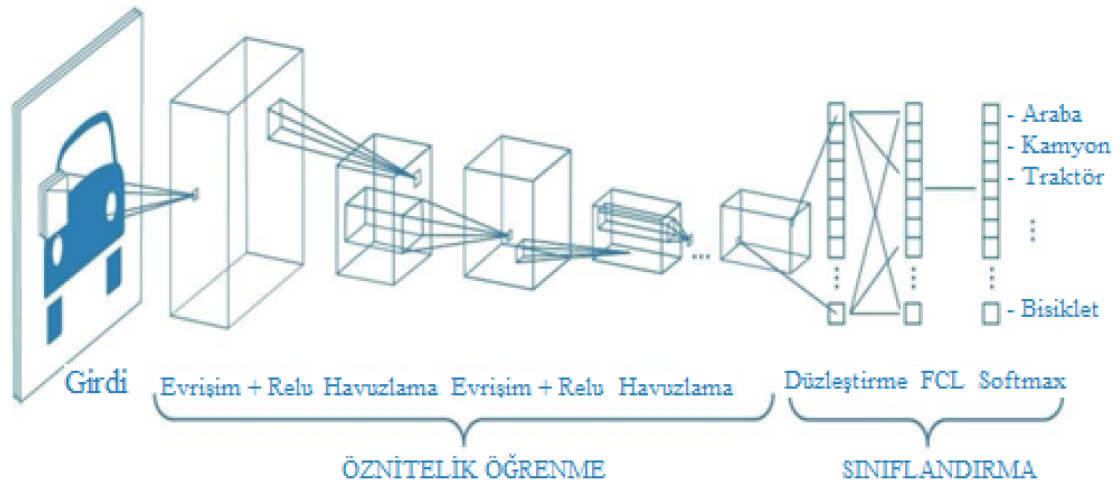


Şekil 3. 1. Yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki ilişki [17].

3.1.2. Evrişimsel Sinir Ağları (CNN)

Derin öğrenmede, evrişimsel sinir ağları (CNN veya ConvNet), en yaygın olarak görsel görüntüleri analiz etmek için uygulanan bir derin sinir ağı sınıfıdır [18]. Görüntü ve video tanıma, tavsiye sistemleri [19], görüntü sınıflandırma, görüntü bölütleme, tıbbi görüntü analizi, doğal dil işleme, [20] beyin-bilgisayar arayüzleri [21] ve finansal zaman serilerinin tahmini [22] alanlarında uygulamaları vardır.

Evrişimsel sinir ağlarının temel mimari yapısı Şekil 3.2 ile gösterilmiştir. Şekilde görüleceği üzere birden fazla katmandan oluşan CNN'lerin katmanlarının temel işlevleri şu şekilde özetlenebilir.



Şekil 3. 2. CNN mimarisinin modeli [23].

- Convolution Layer – The Kernel (Evrışim Katmanı – Çekirdek):** Evrişim katmanının amacı, giriş görüntüsünden kenarlar gibi yüksek öneme sahip özellikleri çıkarmaktır. ConvNet'lerin yalnızca bir evrişim katmanı ile sınırlı olması gerekmez. Geleneksel olarak, ilk ConvLayer, kenarlar, renk, gradyan oryantasyonu vb. düşük öncelikli özellikleri yakalamaktan sorumludur. Eklenen katmanlarla, mimari, üst düzey özelliklere de uyum sağlar ve veri kümesindeki görüntüleri sağlıklı bir şekilde anlayan bir ağ sunar [23].
- Pooling Layer (Havuzlama Katmanı):** Evrişim katmanına benzer şekilde, Havuzlama katmanı, evrişimli özelliğin uzamsal boyutunu küçültmekten sorumludur. Bu, boyutsallık azaltma yoluyla verileri işlemek için gereken hesaplama gücünü azaltmaktır. Ayrıca, rotasyonel ve konumsal olarak değişmeyen baskın özelliklerin çıkarılması, böylece modelin etkili bir şekilde eğitilmesi sürecini sürdürmek için kullanışlıdır [23].
- Classification – Full Connected Layer (Sınıflandırma – FC Katmanı)** Tamamen bağlı bir katman eklemek, evrişimli katmanın çıktısı ile temsil edilen yüksek seviyeli özelliklerin doğrusal olmayan kombinasyonlarını öğrenmenin (genellikle) ucuz bir yoldur. Tam bağlı katman, bu alanda muhtemelen doğrusal olmayan bir işlevi öğreniyor [23].

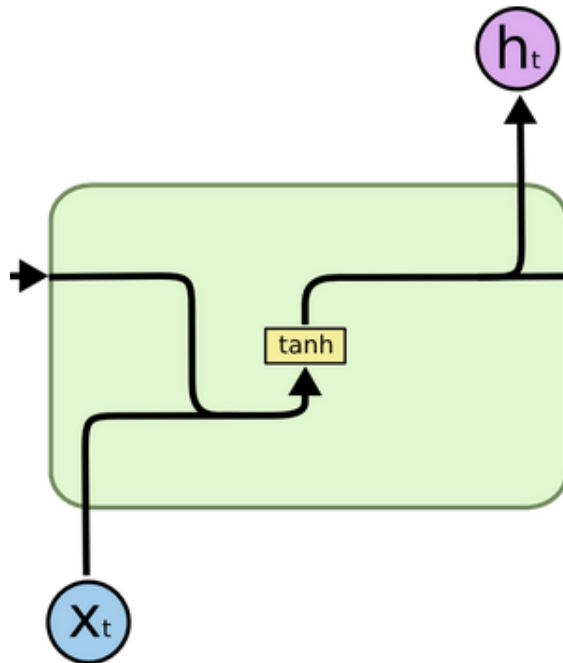
3.1.3. Tekrarlayan Sinir Ağları

Tekrarlayan sinir ağları (RNN), düğümler arasındaki bağlantıların zamansal bir dizi boyunca yönlendirilmiş bir grafik oluşturduğu bir yapay sinir ağı sınıfıdır. Basit RNN hücresinin iç yapısı Şekil 3.3 ile gösterilmiştir. Bu, zamansal dinamik davranış sergilemesine izin verir. İleri beslemeli sinir ağlarından türetilen RNN'ler, değişken uzunluktaki girdi dizilerini işlemek için dahili durumlarını (belleklerini) kullanabilirler [24-26]. Bu, onları bölümlere ayrılmamış, bağlantılı el yazısı tanıma [27], konuşma tanıma [28, 29] veya zamana bağlı serilerin analizi gibi görevlere uygulanabilir kılar.

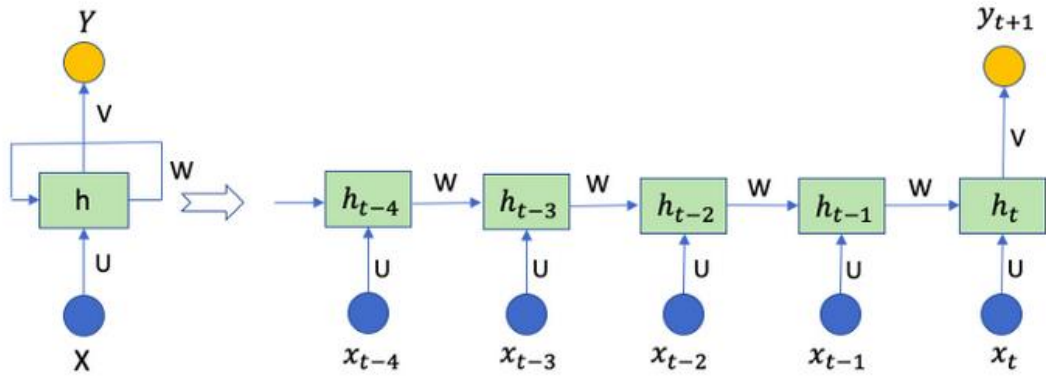
Şekil 3.4 ile gösterilmiş olan yapının çıkış fonksiyonu şu şekilde ifade edilmektedir.

$$h_t = f(U * x_t + W * h_{t-1} + b_t) \quad (3.1)$$

Formülde h_t t zamanına ait gizli katman durumunu, x_t t zamanına ait girdilerin durumunu, b_t gürültü bileşenini, U ve W ise parametre matrislerini ifade etmektedir [29].



Şekil 3. 3. Basit RNN hücresinin iç yapısı [29].



Şekil 3. 4. Basit çoktan teke (Many-to-One) RNN yapısı. [29]

3.1.4. Yok Olan Gradyanlar Problemi

Sinir ağlarına belirli aktivasyon fonksiyonlarını kullanan daha fazla katman eklendikçe, kayıp fonksiyonunun gradyanları sifira yaklaşır ve ağı eğitilmesi zorlaşır. Bu durum, yapay zekâ alanında “Yok Olan Gradyanlar Problemi (Vanishing Gradient Problem)” olarak adlandırılmaktadır ve çok uzun seriler ile eğitilmesi gereken modellerin eğitilemez duruma gelmelerine sebep olmaktadır. Gradyanların kaybolma süreci, RNN'yi geçmiş veri durumlarını saklamak için iyi bir seçenek yapmaz. Bilginin hızla kaybolmadığı özyinelemeli bir yapıya duyulan ihtiyaçtan dolayı LSTM ve GRU modelleri ortaya çıkmıştır. Kaybolan gradyan probleminin matematiksel olarak izahı şu şekildedir [30].

- Problemin daha kolay anlatılabilmesi için b_t gürültü bileşeninin olmadığı kabul edilecektir. Bu durumda sigmoid aktivasyon fonksiyonu için gizli katman durumu aşağıdaki gibi ifade edilebilir [30].

$$h_t = \sigma(wh_{t-1}) \quad (3.2)$$

- Bu formülün birinci dereceden türevinin alınması durumunda aşağıdaki denklem ortaya çıkacaktır. Bu denkleme ait w^{t_2-t} bileşeni gradyanların eksponansiyel olarak artmasına ve patlamasına ya da sifira yakınsayarak çok hızlı bir şekilde yok olmalarına sebep olacaktır [30].

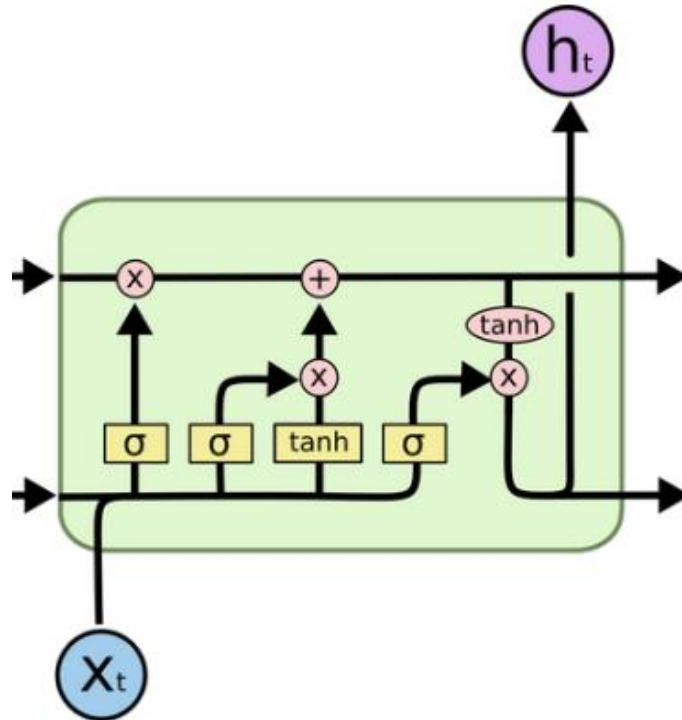
$$\frac{\partial h_{t_2}}{\partial h_t} = w^{t_2-t} \prod_{k=1}^{t_2-t} w\sigma'(wh_{t_2-k}) \quad (3.3)$$

- Aynı formülün LSTM ve GRU için hesaplanması durumunda aşağıdaki denklem ortaya çıkacaktır ve görüldüğü üzere w^{t_2-t} bileşenin olmaması gradyanların daha yavaş bir şekilde sifıra yakınsamalarını sağlayacaktır. Fakat unutulmamalıdır ki; bu modeller, gradyanların yok olmasına tamamen dirençli değildir. Yalnızca sifıra daha yavaş yakınsamaları sebebiyle daha uzun verileri destekleyebilmektedirler [30].

$$\frac{\partial s_{t_2}}{\partial s_t} = \prod_{k=1}^{t_2-t} \sigma(h_{t_2-k}) \quad (3.4)$$

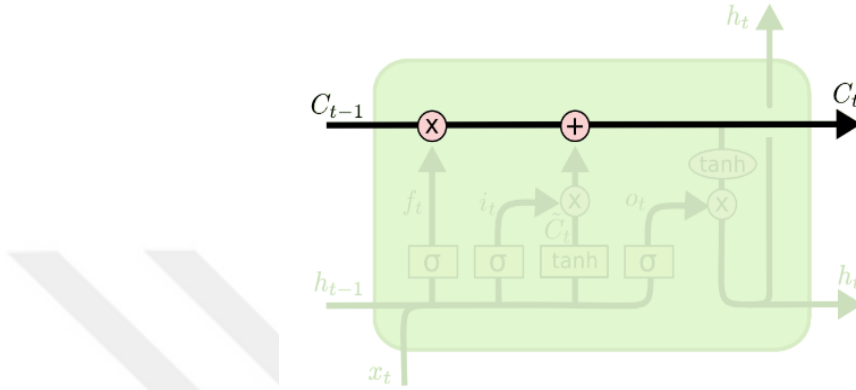
3.1.5. Uzun Kısa-Dönem Bellek (Long Short-Term Memory, LSTM)

Uzun Kısa-Dönem Bellek Ağları – genellikle “LSTM” olarak adlandırılır – uzun süreli bağımlılıkları öğrenebilen özel bir RNN türüdür. LSTM ağları ilk olarak Hochreiter & Schmidhuber (1997) tarafından tanıtılmış ve Felix Gers, Fred Cummins, Santiago Fernandez, Justin Bayer, Daan Wierstra, Julian Togelius, Faustino Gomez, Matteo Gagliolo ve Alex Graves’in yapmış olduğu çalışmalar ile ün kazanmıştır. Çok çeşitli problemlerin çözümü için tercih edilmiş ve tatmin edici sonuçlar vermiştir [31]. LSTM hücresinin iç yapısı Şekil 3.5. ile gösterilmiştir.



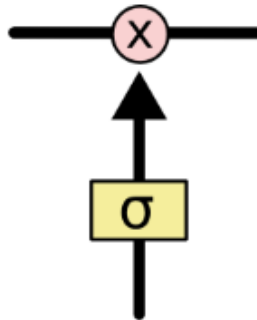
Şekil 3. 5. LSTM Hücresinin iç yapısı [31].

LSTM'lerin farklılaştığı nokta, hücre durumudur. Şekil 3.6. da görüldüğü gibi diyagramın üstünden geçen yatay çizgidir. Hücre durumu bir tür taşıma bandı gibidir. Sadece bazı küçük doğrusal etkileşimlerle, tüm zincir boyunca dosdoğru çalışır. Bilginin değişmeden akması çok kolaydır [31].



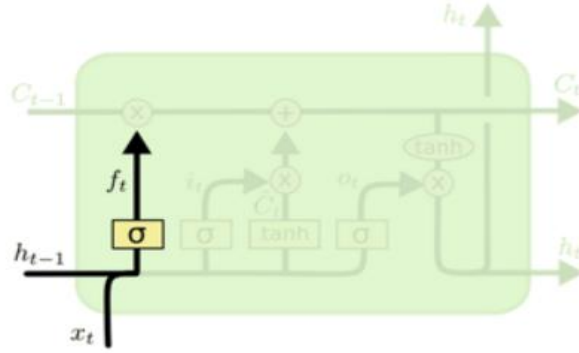
Şekil 3. 6. LSTM hücre durumu bilgisinin taşındığı hat [31].

Şekil 3.7. de görüldüğü gibi LSTM, kapılar adı verilen yapılar tarafından dikkatlice düzenlenen hücre durumuna bilgi ekleme veya çıkarma yeteneğine sahiptir. Kapılar, isteğe bağlı olarak bilgi aktarmanın bir yoludur. Kapılar, sigmoid sinir ağı katmanından ve noktasal çarpma işleminden oluşurlar [31].



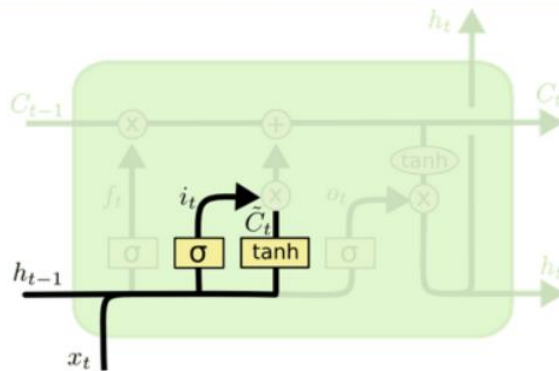
Şekil 3. 7. Örnek bir LSTM kapısı [31].

LSTM'deki ilk adım, hücre durumundan hangi bilgilerin atılacağına karar vermektir. Bu işlemin yapılmasını sağlayan kapı Şekil 3.8 ile gösterilmiştir. Bu karar, “unutma kapısı (forget gate)” olarak adlandırılan sigmoid katman tarafından verilir [31].



Şekil 3. 8. LSTM modeline ait Unutma kapısı [31].

Sonraki adım, hücre durumunda hangi yeni bilgilerin depolanacağına karar vermektir. Bu süreç iki aşamadan oluşur. İlk olarak, “girdi kapısı” (input gate) adı verilen bir sigmoid katman, hangi değerlerin güncelleneceğine karar verir. Girdi kapısı Şekil 3.9 ile gösterilmiştir. İkinci aşamada ise eski hücre durumu C_{t-1} 'i, yeni hücre durumu C_t 'ye güncellenir. Güncelleme kapısı Şekil 3.10 ile gösterilmiştir [31].

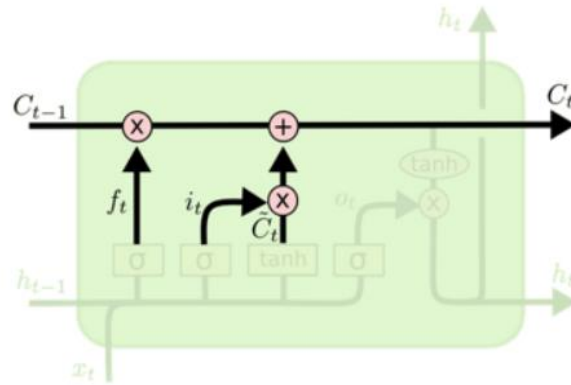


Şekil 3. 9. LSTM modeline ait Girdi kapısı [27].

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3.5)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3.6)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3.7)$$



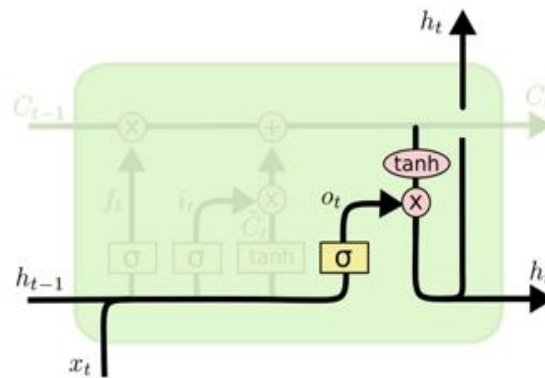
Şekil 3. 10. LSTM modeline ait Hücre durumunun güncellenmesi [31].

Son olarak, çıktının hangi değeri alacağına karar verilir. Bu çıktı, hücre durumuna dayalı olacaktır, ancak filtrelenmiş bir sürüm olacaktır. Çıktı kapısı Şekil 3.11 ile gösterilmiştir [31].

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (3.8)$$

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.9)$$

$$h_t = O_t * \tanh(C_t) \quad (3.10)$$



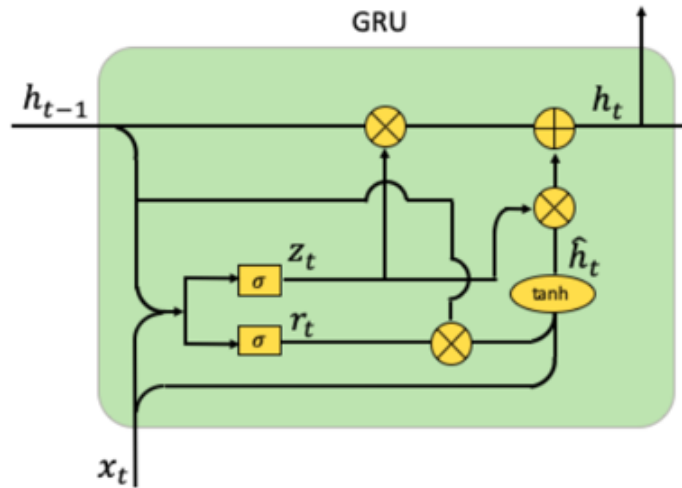
Şekil 3. 11. LSTM modeline ait Çıktı kapısı [31].

3.1.5. Kapılı Tekrarlayan Hücre (Gated Recurrent Unit, GRU)

Standart bir RNN'nin kaybolan gradyan problemini çözmek için GRU bünyesinde güncelleme kapısı ve sıfırlama kapısı adı verilen yapılar kullanır. Temel olarak bunlar,

çıkı olarak hangi bilgilerin iletilmesi gerektiğine karar veren iki vektördür. Bu kapılarla ilgili özel olan durum, uzun zaman öncesinden gelen bilgileri kaybetmemesi veya tahminle alakasız bilgileri tutmaması için eğitilebilmeleridir [31].

GRU, Cho ve arkadaşları tarafından literatüre sunulmuştur [32]. GRU, LSTM'deki gibi hücre durumuna ve çıkış kapısına sahip değildir. Bu nedenle LSTM'den daha az parametreye sahiptir. GRU, bilgileri aktarmak için gizli katmanları kullanır. GRU'nun tarihsel olarak az sayıda veri içeren görevlerde ve daha küçük veri kümelerinde daha iyi performans sergilediği gözlemlenmiştir, ancak LSTM genel olarak GRU'dan daha iyi performans gösterme eğilimindedir [32]. GRU modelinin iç yapısı Şekil 3.12 ile gösterilmiştir.



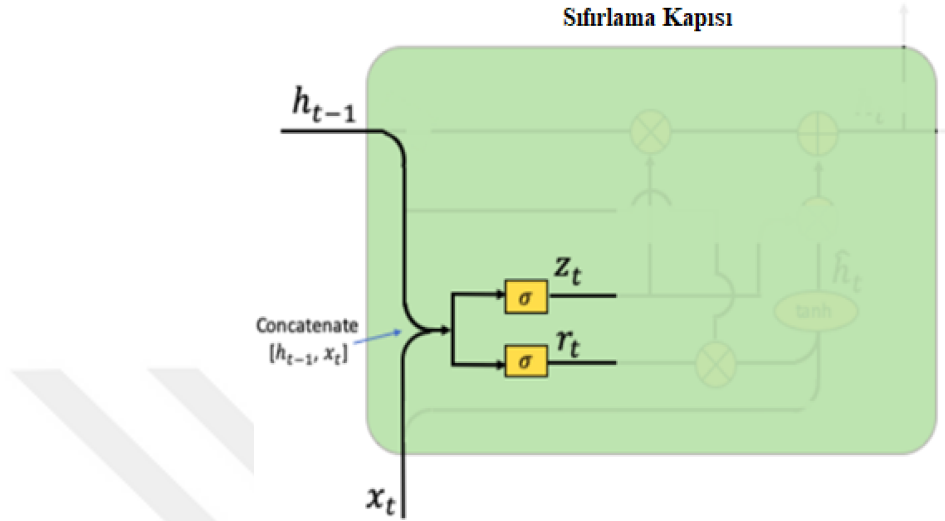
Şekil 3. 12. GRU iç yapısı [32].

GRU parametreleri arasında W_r , W_z ve W bulunur. Sıfırlama sinyali r_t , önceki gizli durumun göz ardı edilip edilmeyeceğini belirlerken, güncelleme sinyali z_t , gizli durum h_t 'nin yeni gizli durum h_t^{\wedge} ile güncellenmesi gerekip gerekmediğini belirler [32].

Sıfırlama Kapısı: LSTM bünyesinde yer alan girdi ve unutma kapılarıyla aynı amaca yönelik kullanımı söz konusudur. Bu kapının grafiksel ifadesi Şekil 3.13 ile yapılmıştır. Kapı r_t , önceki gizli durumun göz ardı edilip edilmeyeceğini belirler. h_t^{\wedge} ile güncelleme kapısı için z_t kapısı oluşturulur. W_z ve W_r eğitilecek ağırlık parametreleri olup b_z ve b_r gürültü vektörleridir [32].

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (3.11)$$

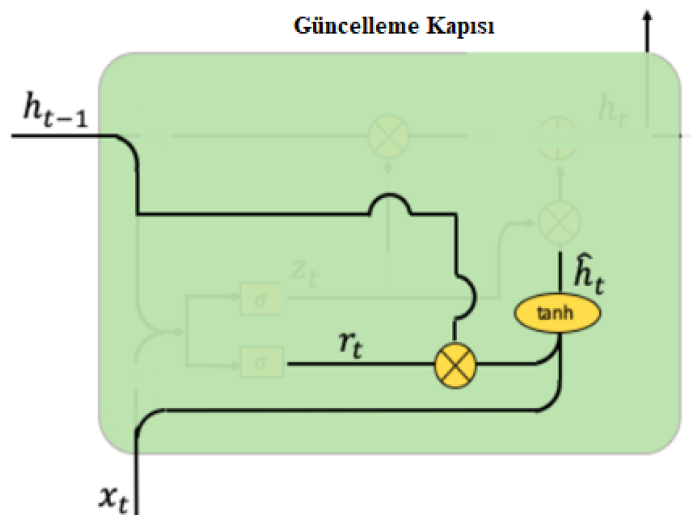
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (3.12)$$



Şekil 3. 13. GRU sıfırlama kapısı [32].

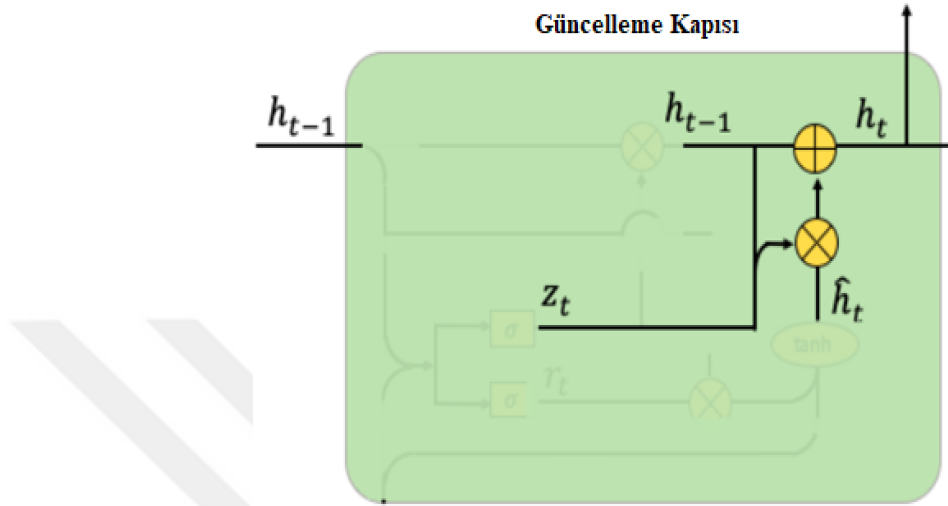
Güncelleme Kapısı (1. Bölüm): Şekil 3.14 ile izah edilen bu kısım r_t veya r_t ve h_{t-1} çarpılır. Çarpma, h_{t-1} 'in ne kadarının tutulacağı veya yok sayılacağı anlamına gelir. Bu, h_t 'nin güncellenmesi için kullanılacak geçici bir \hat{h}_t oluşturur. W_h ve b_h ağırlık parametreleri ve gürültü vektörleridir [32].

$$\hat{h}_t = \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h) \quad (3.13)$$



Şekil 3. 14. GRU güncelleme kapısı Güncelleme Kapısının 1. Aşaması [32].

Güncelleme Kapısı (2.Bölüm): Şekil 3.15 ile izah edilen bu kısım, z_t ağırlığına göre h_{t-1} ile h_t arasındaki ağırlıklı ortalamayı hesaplar. z_t sıfıra yakınsa, geçmiş bilgiler çıktı değerine çok az katkıda bulunur yeni bilgiler ise daha fazla katkıda bulunur [32].



Şekil 3. 15. GRU Güncelleme Kapısının 2. Aşaması [32].

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \hat{h}_t \quad (3.14)$$

3.2. Geliştirmede Kullanılan Yazılım Dili ve Kütüphaneler

Bu tez çalışmasında, günümüzde yapay zekâ çalışmalarında çok popüler olması nedeniyle ve birçok açık kaynak kodu olması sebebiyle Python yazılım dili kullanılmıştır. Python yazılım dilinin yapay zekâ alanındaki popülerliğinin nedeni, Google tarafından geliştirilen TensorFlow kütüphanesine ve bu kütüphane üzerine inşa edilmiş derin öğrenme uygulamalarında sıklıkla kullanılan Keras kütüphanesine sahip olmasından kaynaklanmaktadır. Bu araçlara destekleyici olarak Python, veri işlemlerinde kullanılan Pandas, Numpy, Scikit-Learn gibi kütüphanelere ve veri görselleştirmede kullanılan Matplotlib kütüphanesine sahiptir.

3.2.1. Keras ve TensorFlow Kütüphaneleri

TensorFlow, uçtan uca, açık kaynaklı bir makine öğrenimi platformudur. Diferansiyellenebilir programlama (Differentiable programming) için bir altyapı katmanı olarak düşünülebilir. TensorFlow, dört temel yeteneği birleştirir [33]:

- CPU, GPU veya TPU üzerinde düşük seviyeli tensör işlemlerini verimli bir şekilde yürütme.
- Rastgele türevlenebilir ifadelerin gradyanını hesaplama.
- Hesaplamayı birçok cihaza ölçeklendirme (ör. 27.000 GPU'yu kapsayan Oak Ridge National Lab'deki Summit süper bilgisayarı).
- Programları ("grafikleri") sunucular, tarayıcılar, mobil ve gömülü cihazlar gibi harici çalışma zamanlarına aktarma.

Keras, Python ile yazılmış, makine öğrenimi platformu TensorFlow üzerinde çalışan bir derin öğrenme API'sidir. Hızlı denemeyi mümkün kılmaya odaklanılarak geliştirilmiştir. Bu tez çalışması kapsamında Keras kütüphanesinin 2.4.3 ve TensorFlow kütüphanesinin 2.5 versiyonu kullanılmıştır.

3.2.2. Pandas ve Numpy

NumPy, çok boyutlu bir dizi nesnesi, türetilmiş çeşitli nesnelere (maskelenmiş diziler ve matrisler gibi) ve diziler üzerinde matematiksel, mantıksal, şekil işleme, sıralama, seçme dahil olmak üzere hızlı işlemler için çeşitli rutinler sağlayan bir Python kütüphanesidir. Ayırık Fourier dönüşümleri, temel lineer cebirsel işlemler, temel istatistiksel işlemler, rastgele simülasyonlar için de kullanılmaktadır. NumPy paketinin merkezinde ndarray nesnesi bulunur [34]

Pandas, veri bilimi/veri analizi ve makine öğrenimi görevleri için en yaygın şekilde kullanılan, açık kaynak kodlu bir Python kütüphanesidir. Çok boyutlu diziler için destek sağlayan Numpy adlı kütüphane üzerine inşa edilmiştir [35]. En popüler veri işleme paketlerinden biri olan Pandas, Python ekosistemindeki diğer birçok veri bilimi modülüyle uyumlu çalışır ve genellikle her Python versiyonuna dahil edilir. Pandas,

aşağıda listelenen işlemler de dahil olmak üzere, veri setleri üzerindeki zaman alan ve tekrarlayan görevlerin çoğunun gerçekleştirilmesinde kolaylık sağlar: [35]

- Veri temizleme
- Veri doldurma
- Veri normelleştirme
- Veri setlerini birleştirme
- Veri görselleştirme
- İstatistiksel analiz
- Veri denetimi
- Veri saklama ve yükleme

3.3. Düşüm Seyreltme (Dropout)

Çok sayıda parametreye sahip olan derin sinir ağları, çok güçlü makine öğrenme sistemleridir. Ancak, bu tür ağlarda aşırı öğrenme ciddi bir sorundur. Düşüm seyrletme, bu sorunu çözmek için bir tekniktir. Düşüm seyrletme tekniği genelde tam bağlı katmanlarda sonra kullanılır. Düşüm seyrletme kullanılarak tam bağlı katmanlardaki bağlar koparılır. Böylece düğümler birbiri hakkında daha az bilgiye sahip olur ve bunun doğal sonucu olarak düğümler birbirlerinin ağırlık değişimlerinden daha az etkilenirler. Bu nedenle düşüm seyrletme yöntemi ile daha tutarlı modeller oluşturulabilmektedir. Aynı zamanda her bir katmanda farklı gizli katman kombinasyonları birbiriyle çalıştığı için daha iyi bir öğrenme gerçekleşecektir. Bu, aşırı öğrenmeyi önemli ölçüde azaltır ve diğer düzenleme yöntemlerine göre büyük iyileştirmeler sağlar. Düşüm seyrletmenin sinir ağlarının görme, konuşma tanıma, belge sınıflandırma ve hesaplamalı biyolojideki denetimli öğrenme görevlerinde modelin performansını geliştirdiğini Srivastaya ve arkadaşları yaptıkları çalışmada göstermiştir. [36].

3.3.1. Aşırı Öğrenme (Overfitting)

Bir model, eğitim verilerindeki ayrıntıları ve gürültüyü, modelin yeni veriler üzerindeki performansını olumsuz yönde etkileyecek ölçüde öğrendiğinde, aşırı öğrenme gerçekleşir. Bu, eğitim verilerindeki gürültü veya rastgele dalgalanmaların model

tarafından kavram olarak alındığı ve öğrenildiği anlamına gelir. Sorun, bu kavramların yeni verilere uygulanması ve modellerin genelleme yeteneğini olumsuz etkilemesidir [37].

Bir hedef işlevi öğrenirken daha fazla esnekliğe sahip olan, parametrik olmayan ve doğrusal olmayan modellerde aşırı öğrenme daha olasıdır. Bu nedenle, birçok parametrik olmayan makine öğrenimi algoritması, modelin ne kadar ayrıntı öğreneceğini sınırlamak için parametreler veya teknikler de içerir [37].

Veri dağılımına karşılık modelin temsili istenen öğrenme ve aşırı öğrenme örneği Şekil 3.16 ile izah edilmiştir.

3.3.2. Erken Durdurma (Early Stopping)

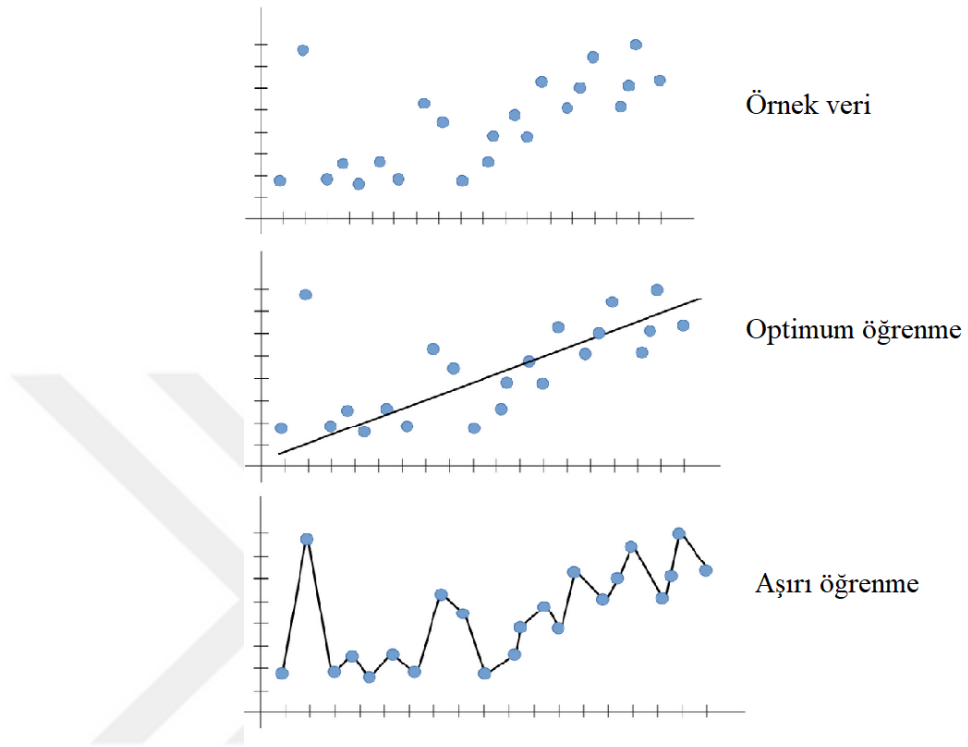
Eğitim sırasında iterasyon sayısının çok büyük seçilmesi, veri setinin model tarafından aşırı öğrenilmesine yol açabilirken, çok düşük seçilmesi ise eksik öğrenmeye (underfitting) neden olabilir. Erken durdurma, isteğe bağlı olarak çok sayıda eğitim dönemi belirlenmesine ve eğitilen modelin hata performansının daha fazla iyileşmediği anda eğitimin durdurulmasına olanak tanıyan bir yöntemdir [38].

3.4. Gradyan İniş Metodu (Gradient Descent Optimizer)

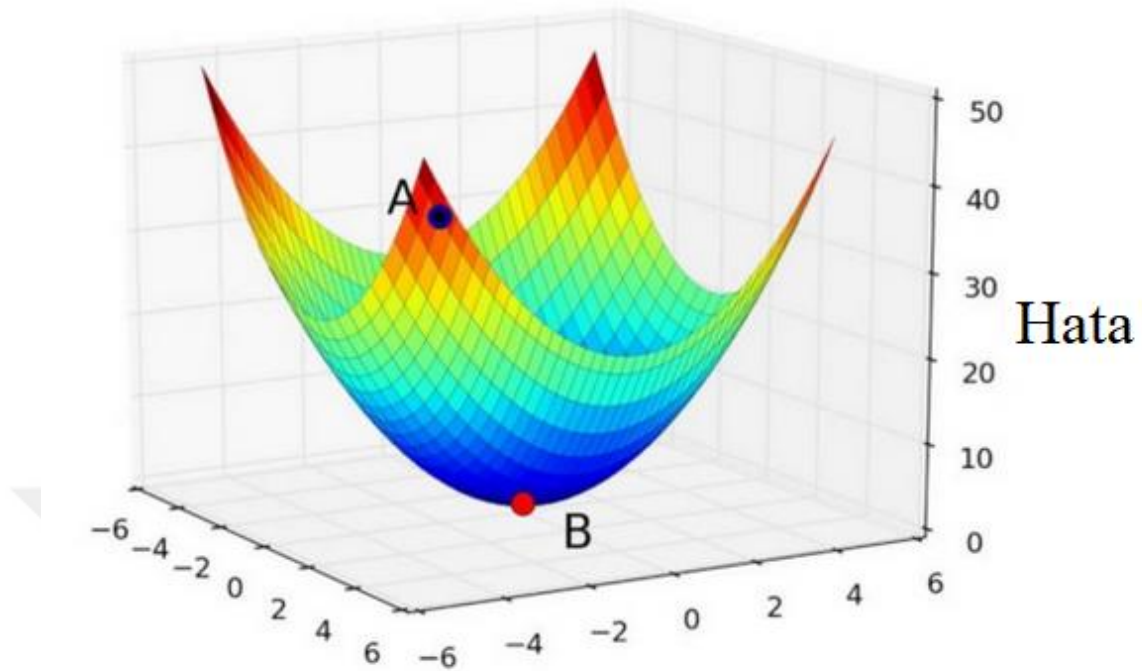
Gradyan iniş, muhtemelen tüm optimizasyonlar arasında en popüler ve yaygın olarak kullanılanıdır. Sinir ağı için en düşük hata ile çıkışı tahmin edebilecek optimum parametre değerlerini bulmak temel hedefdir. Bu amaca yönelik olarak tüm optimizasyonlar, sinir ağı parametrelerinin güncellenmesine odaklanırlar. Ancak hata fonksiyonu belirlenirken dikkat edilmesi gereken nokta, hatanın lokal minimumları yerine global minimum değerine ulaşılmalıdır. Hata fonksiyonunun üç boyutlu bir görseli, Şekil 3.17 de verilmiştir [39].

Gradyan, her defasında yeniden hesaplandığında ve ağırlık değerleri güncellendiğinde, optimum değere yaklaşırlar. Sinir ağını eğitmeye başlamadan önce, Şekil 3.17’de gösterilen A noktası ile temsil edilen hata değeri yüksek olacaktır. Sinir ağının eğitiminin her yinelenmesinde (hata fonksiyonuna ait gradyanları hesaplama ve ağırlıkları

güncelleme), hata değeri azalır ve Şekil 3.17’de B noktası ile temsil edilen global minimum değere yaklaşılır [39].



Şekil 3. 16. Optimum öğrenme ve Aşırı öğrenme örneğinin grafiksel ifadesi.

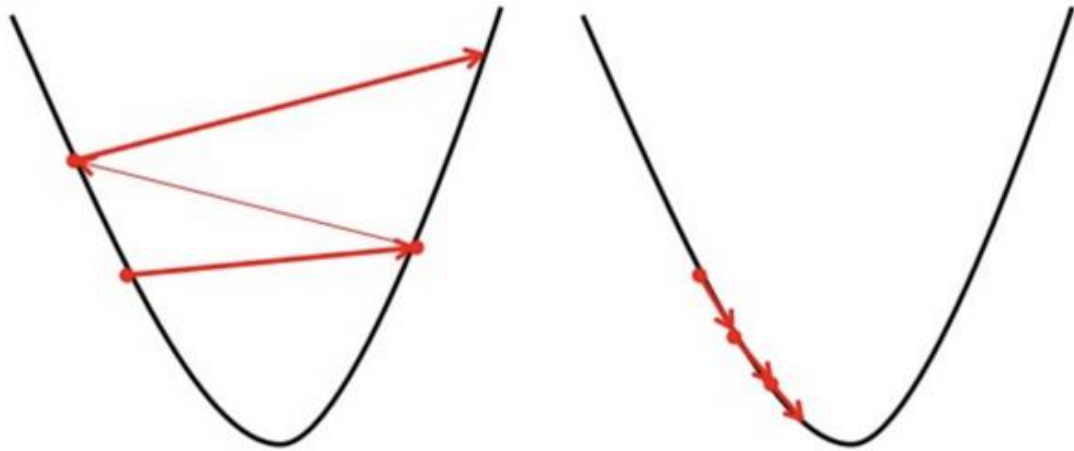


Şekil 3. 17. Örnek bir hata fonksiyonu grafiği [40].

3.4.1. Öğrenme Oranı (Learning Rate)

Öğrenme oranı, gradyan inişinin ve ayrıca diğer optimizasyon algoritmalarının en önemli özelliğidir. Öğrenme oranı hata fonksiyonunun minimum noktasına ulaşmak için modele ait ağırlık parametrelerinin hangi miktarda değişeceğini belirtmektedir. Öğrenme oranı olarak büyük bir değer seçilirse, ağırlık ve kutup (bias) değerlerinde büyük değişiklikler oluşur, yani minimum noktasına ulaşmak için büyük değişimler yapıyor olacaktır. Ayrıca, global minimumun aşılması ve ulaşılmak istenilen asıl hata değeri yerine istenmeyen bir durum olan fonksiyonun farklı lokal minimum bölgelerinin bulunması ihtimali de söz konusu olur. Öğrenme oranı olarak küçük bir değer seçilirse, global minimumların aşılma riskini ortadan kaldırmış olur ancak algoritmanın yakınsaması daha uzun sürer [40].

Büyük öğrenme oranı Küçük öğrenme oranı



Şekil 3. 18. Büyük ve küçük öğrenme oranlarının grafik ile izahı [40].

3.4.2. RMSProp Optimizer

RMSprop optimizasyonu, momentumlu gradyan iniş algoritmasına benzerdir. RMSprop optimize edici, hata fonksiyonuna ait salınımları dikey yönde kısıtlar. Böylece öğrenme hızı artırılabilir ve öğrenme algoritması yatay yönde daha hızlı yakınsayarak daha büyük adımlar atabilir. RMSprop ve gradyan inişi arasındaki fark, gradyanların nasıl hesaplandığı ile ilgilidir. Denklem 3.15 – 3.18 RMSprop ve momentumlu gradyan inişi için gradyanların nasıl hesaplandığını göstermektedir. Momentumun değeri beta ile gösterilir ve genellikle 0.9'a ayarlanır [40].

$$v_{dw} = \beta \cdot v_{dw} + (1 - \beta) \cdot dw^2 \quad (3.15)$$

$$v_{db} = \beta \cdot v_{db} + (1 - \beta) \cdot db^2 \quad (3.16)$$

$$W = W - \alpha \cdot \frac{d_w}{\epsilon + \sqrt{v_{dw}}} \quad (3.17)$$

$$b = b - \alpha \cdot \frac{d_b}{\epsilon + \sqrt{v_{db}}} \quad (3.18)$$

Bazen v_{dw} değeri gerçekten 0'a yakın olabilir. O zaman W ağırlıklarının değeri yapay zekâ çıktılarını anlamsızlaştıracak biçimde artabilir. Gradyanların patlamasını önlemek için Denklem 3.17 ve 3.18'de görüleceği üzere paydaya küçük bir değere sahip bir parametre olan epsilon eklenir [40].

3.5. Hata Metrikleri

Hata metrikleri yapay zekâ alanında oluşturulan modellerin başarılarını karşılaştırmak amacıyla kullanılan parametreleri ifade etmektedir. Tasarlanan derin öğrenme modeli, temelde bir regresyon modeli olması sebebiyle modellerin performanslarının değerlendirmesi aşamasında ortalama karesel hata (Mean squared error, MSE), ortalama bağıl hata (Mean relative error, MRE) ve ortalama mutlak hata (Mean absolute error, MAE) kullanılmıştır.

3.5.1. Ortalama Karesel Hata

Basitçe, ortalama karesel hata, bir regresyon eğrisinin bir dizi noktaya ne kadar yakın olduğunu söyler. MSE, bir makine öğrenmesi modelinin, tahmin performansını ölçer. MSE değeri, her zaman pozitif değerlidir ve bu değer in sıfıra yakın olması modellerin daha iyi bir tahmin yaptığı anlamına gelir. Ortalama karesel hatanın formülü Denklem 3.19 ile ifade edilmiştir [41].

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2 \quad (3.19)$$

3.5.2. Ortalama Mutlak Hata

Ortalama mutlak hata, iki sürekli değişken arasındaki farkın ölçüsüdür. MAE, her gerçek değer ile veriye en iyi uyan çizgi arasındaki ortalama dikey mesafedir. MAE, aynı zamanda her veri noktası ile en iyi uyan çizgi arasındaki ortalama yatay mesafedir. MAE değeri kolay yorumlanabilir olduğu için regresyon ve zaman serisi problemlerinde sıkça kullanılmaktadır. MAE, yönlerini dikkate almadan bir dizi tahminindeki hataların ortalama büyüklüğünü ölçen, tüm tekil hataların ortalamada eşit olarak ağırlıklandırıldığı doğrusal bir skordur. MAE değeri 0'dan $+\infty$ 'a kadar değişebilir. Daha düşük MAE değerine sahip modeller daha iyi performans gösterir. Ortalama mutlak hatanın formülü Denklem 3.20 ile gösterilmiştir [41].

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \tilde{y}_i| \quad (3.20)$$

3.5.3. Ortalama Bağıl Hata

Bağıl hata, mutlak hatanın gerçek değere bölümü ile hesaplanan bir hata metriğidir. Regresyon problemlerinde çok geniş bir aralıkta tahmin yapılması durumunda MRE' nin kullanılması MSE veya MAE'ye kıyasla daha mantıklıdır. Ortalama bağıl hata, elde edilen bağıl hataların aritmetik ortalamalarının hesaplanması ile elde edilir. Bağıl hata ve ortalama bağıl hatanın formülleri Denklem 3.21 ve Denklem 3.22 ile verilmiştir [41].

$$RE = \frac{|y_i - \tilde{y}_i|}{|y_i|} \quad (3.21)$$

$$MRE = \frac{1}{n} \sum_{i=1}^n RE_i \quad (3.22)$$

4. BÖLÜM

MODELİN OLUŞTURULMASI

Bu bölümde tez çalışmasında kullanılan derin öğrenme modelinin detaylarından, modeli oluşturmak için kullanılan araçlardan, modelin eğitildiği ortamdan bahsedilecektir.

4.1. Model Performanslarının Kıyaslanması

Kullanılan algoritmaya ve hiper-parametrelere karar verebilmek için öncelikle ISCTR hissesinin günlük fiyatlarını tahmin etmeye yönelik olarak farklı eğitim ortamı ve parametreler ile Tablo 4.1 ile gösterilen çalışma yapılmıştır. Yapılan çalışmada hesaplanan hata değerleri, ortalama karesel hata (Mean Squared Error, MSE) cinsinden hesaplanmışlardır. Tabloda görülen hata değerleri, ilgili parametreler kullanılarak rastgele ağırlıklar ile eğitilen 25'er farklı modele ait elde edilmiş olan en düşük hata değerleridir.

Tablo 4.1'den de görüleceği üzere en düşük hata değerleri 32 Batch boyutu, 64 kayar pencere uzunluklu, 128 gizli katman ve RMSProp optimizere kullanılarak TPU ortamında eğitilen LSTM algoritması için elde edilmiştir. Bu sebeple çalışmada kullanılmak üzere LSTM algoritması seçilmiştir.

4.2. Geliştirme Ortamı

Colaboratory veya kısaca "Colab", Google Research'ün bir ürünüdür. Colab, herkesin tarayıcı aracılığıyla rastgele python kodu yazmasına ve yürütmesine izin veren ve özellikle makine öğrenimi, veri analizi ve derin öğrenme için çok uygun bir platformdur. Daha teknik olarak, Colab, GPU'lar dahil bilgi işlem kaynaklarına ücretsiz erişim

sağlarken, kullanmak için kurulum gerektirmeyen, Jupyter teknolojisini temel alan bir bulut hizmettir.

Tablo 4. 1 Farklı hiper-parametre ve algoritmalara karşılık ISCTR hissesi için MSE değerleri.

Batch Boyutu	Kayar Pencere Uzunluğu	Gizli Katman Sayısı	Optimizer	TPU ortamında eğitilmiş GRU	TPU ortamında eğitilmiş LSTM	GPU ortamında eğitilmiş GRU	GPU ortamında eğitilmiş LSTM
32	32	64	Adam	0.01667	0.02812	0.01942	0.02061
32	32	128	Adam	0.01449	0.06096	0.01654	0.01599
32	32	256	Adam	0.01651	0.01886	0.01395	0.01407
32	32	512	Adam	0.02875	0.02275	0.03562	0.05146
32	64	64	Adam	0.01647	0.01671	0.01561	0.01734
32	64	128	Adam	0.01635	0.01487	0.01512	0.02403
32	64	256	Adam	0.01234	0.02447	0.02380	0.01619
32	64	512	Adam	0.01809	0.01993	0.01693	0.02938
32	128	64	Adam	0.02855	0.01899	0.01729	0.02604
32	128	128	Adam	0.01937	0.01540	0.02986	0.01492
32	128	256	Adam	0.01326	0.02449	0.01658	0.03910
32	128	512	Adam	0.03492	0.01813	0.01546	0.03769
32	32	64	RMSProp	0.02482	0.02326	0.01438	0.01374
32	32	128	RMSProp	0.02149	0.01511	0.03262	0.03464
32	32	256	RMSProp	0.01698	0.04402	0.04644	0.12471
32	32	512	RMSProp	0.02302	0.02166	0.07166	0.04352
32	64	64	RMSProp	0.02467	0.01541	0.01423	0.02718
32	64	128	RMSProp	0.04828	0.02853	0.06263	0.07134
32	64	256	RMSProp	0.02636	0.01184	0.02403	0.01305
32	64	512	RMSProp	0.03901	0.06555	0.02058	0.02111
32	128	64	RMSProp	0.02270	0.01446	0.01989	0.01805
32	128	128	RMSProp	0.02339	0.02267	0.03242	0.01223
32	128	256	RMSProp	0.10977	0.01536	0.04259	0.02569
32	128	512	RMSProp	0.02601	0.03352	0.01526	0.04066

Tablo-4.1’ de de görüleceği üzere hem GPU hem de TPU ortamlarında eğitimler yapılmış olup eğitim süresi ile model sonuçlarında da bir farklılık olup olmadığı gözlemlenmiştir.

Colab ortamında kullanıcıların hizmetine sunulmuş olan GPU'lar, NVIDIA markasına ait ve yapay zekâ çalışmalarında kullanılmak üzere geliştirilmiş olan Tesla T4 modelleridir. Şekil 4.1 ve Şekil 4.2 de bu tez çalışması kapsamında önerilen derin öğrenme modelinin eğitim sürecinin her bir iterasyonuna ait süre bilgileri verilmiştir. Her bir iterasyonun üç saniyeden daha kısa sürmesi sebebiyle önerilen model için TPU tabanlı sistem kullanılmıştır.

```
Epoch 2/1000
105/105 [=====] - 6s 61ms/step
Epoch 3/1000
105/105 [=====] - 6s 59ms/step
Epoch 4/1000
105/105 [=====] - 6s 60ms/step
```

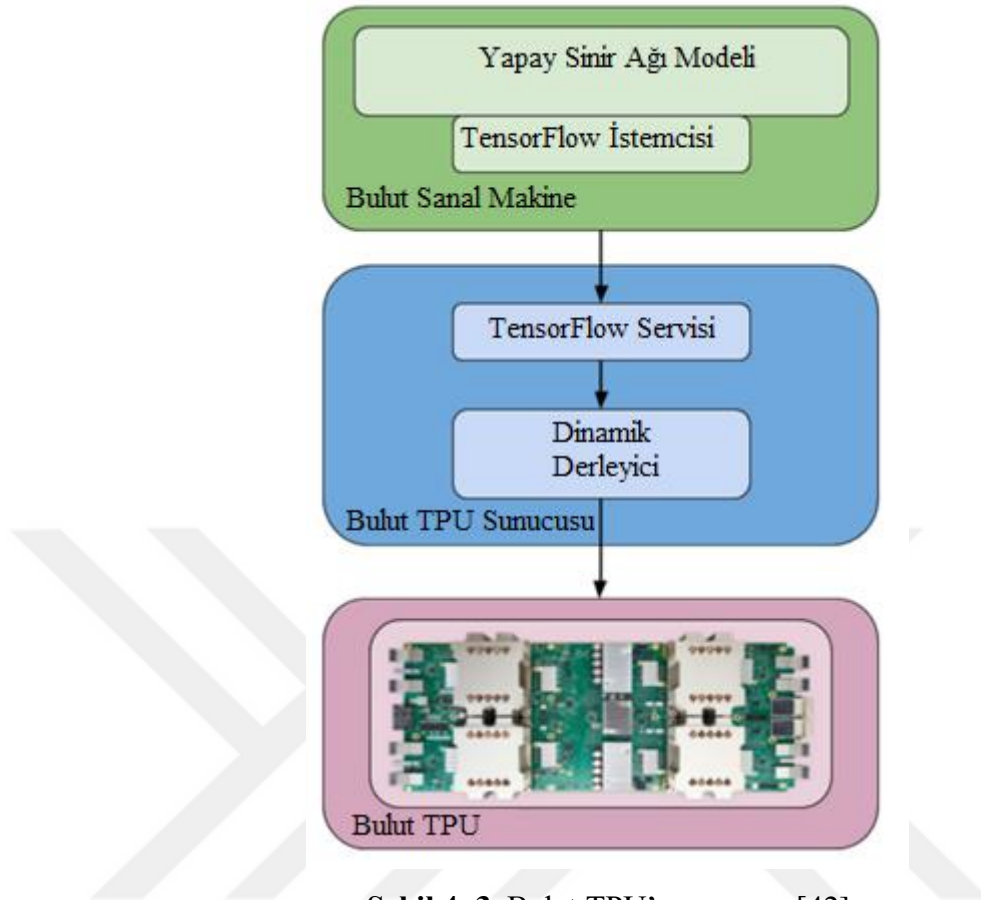
Şekil 4. 1. TPU tabanlı sistemde eğitim süreci.

```
Epoch 2/1000
105/105 [=====] - 9s 83ms/step
Epoch 3/1000
105/105 [=====] - 9s 84ms/step
Epoch 4/1000
105/105 [=====] - 9s 85ms/step
```

Şekil 4. 2. GPU tabanlı sistemde eğitim süreci.

4.2.1. Google TPU

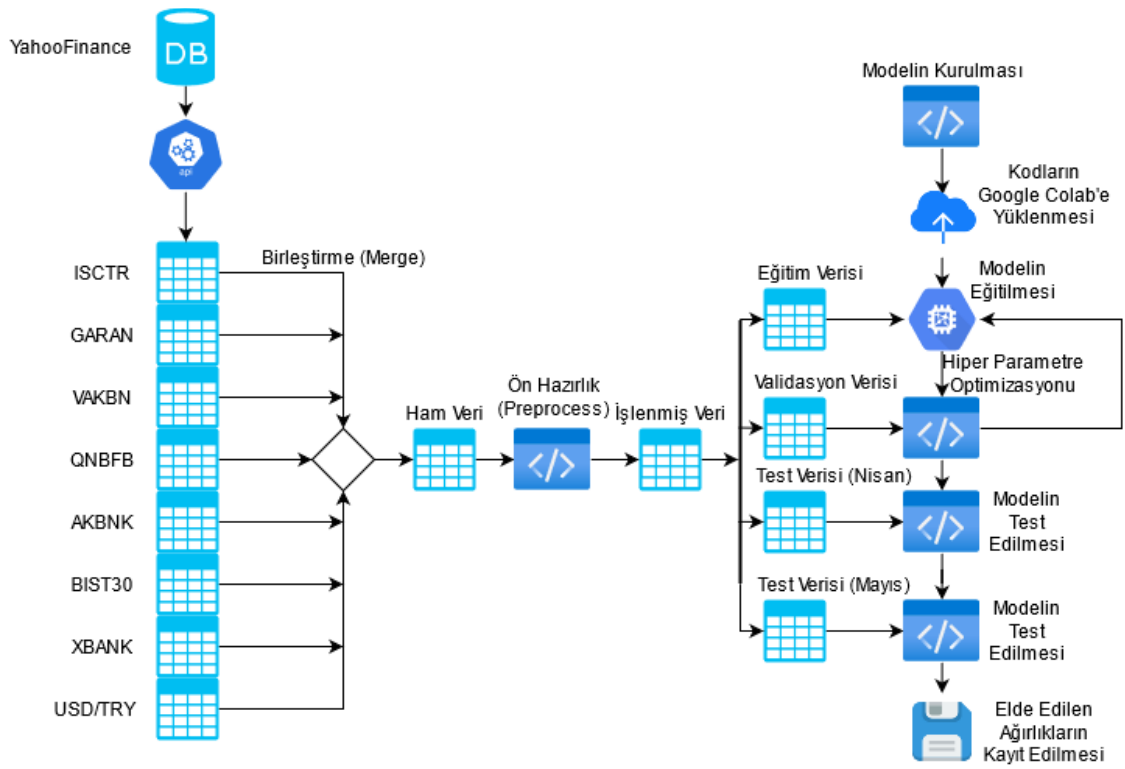
Tensör İşleme Birimleri (TPU'lar), makine öğreniminde iş yüklerini hızlandırmak için kullanılan, Google tarafından özel olarak geliştirilmiş uygulamaya özel entegre devreleridir. TPU'lar, Google'ın makine öğrenimindeki derin deneyiminden yararlanılarak yeni baştan tasarlanmıştır. Bu yapılar, dört bağımsız çipten oluşur ve her bir çip, skaler, vektör ve matris birimlerini (MXU'lar) içeren Tensör Çekirdekleri adı verilen iki hesaplama çekirdeğinden oluşur [42]. Google TPU'nun yapısı Şekil 4.3. ile gösterilmiştir.



Şekil 4. 3. Bulut TPU'nun yapısı [42]

4.3. Önerilen Model

Bu tez çalışması sürecinde yapılan eğitimler neticesinde önerilen model, ISCTR, ISGYO, AKBNK, AKGRT, GARAN, QNBFB, VAKBN hisse senetlerinin ve XBANK, BIST30, TRY/USD endekslerinin önceki 64 güne ait verilerini kullanarak ilgili hisse senetlerinin bir sonraki gün için değerini tahmin etmeye çalışmaktadır. Önerilen modele ait akış şeması Şekil 4.4. ile ifade edilmiştir. Şekil 4.5'te görüleceği üzere önerilen modelde 1.852.673 adet eğitilebilir parametre, yani başka bir deyişle ağırlık (W) bulunmaktadır. Bu modelin Keras kütüphanesi kullanılarak görselleştirilmiş hali Şekil 4.6'da verilmiştir. Şekilden görüldüğü üzere LSTM hücrelerinin her birinde 256 adet gizli katman bulunmaktadır. Düşüm seyreltme (Dropout) katmanlarının seyreltme oranları 0,2 olarak seçilmiştir.



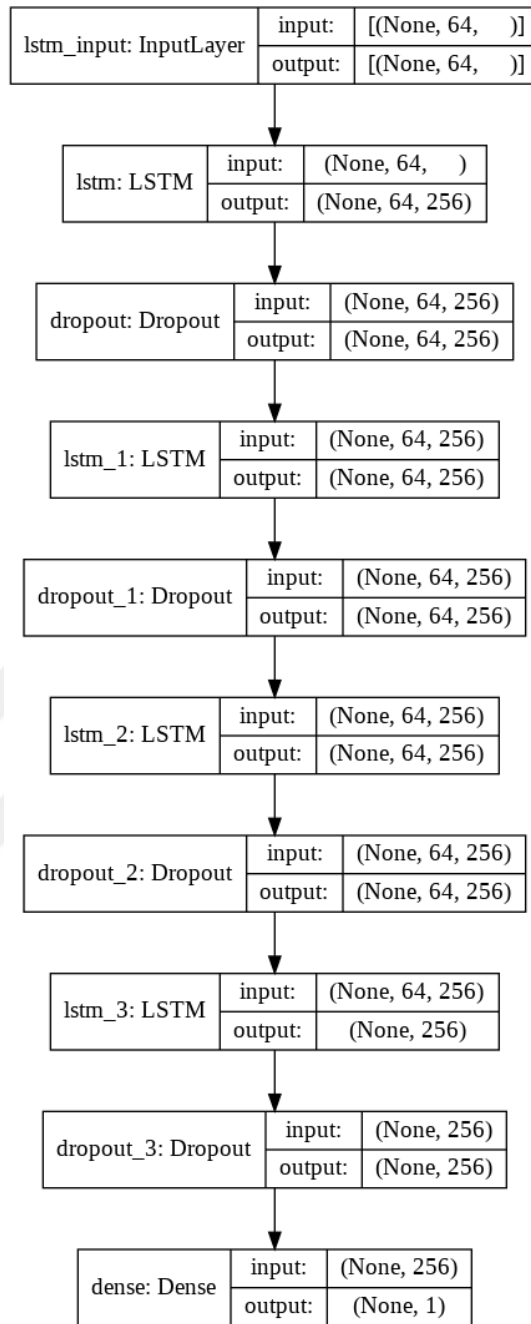
Şekil 4. 4. Modelimize ait akış şeması.

```

=====
Total params: 1,852,673
Trainable params: 1,852,673
Non-trainable params: 0
=====

```

Şekil 4. 5. Model özetimizin kod çıktısına ait ekran görüntüsü.



Şekil 4. 6. Modelimizin özeti için akış diyagramına ait ekran görüntüsü

4.4. Önerilen Modele Ait Erken Durdurma (Early Stopping)

Şekil 4.7 ile önerilen modelin eğitimine ait erken durdurma kod parçası paylaşılmıştır. Kısaca açıklamak gerekirse “monitor” ibaresi ile erken durdurma kriteri olarak takip edilecek parametre belirtilmektedir. Bu aşamada erken durdurma kararının, hata fonksiyonuna bağlı olarak verileceği belirtilmiştir. Bir sonraki argüman olan “min_delta”

ile hata değerinde meydana gelmesi beklenen minimum değişim ifade edilmiştir. Bu değer, önerilen model için iterasyon başına 0.0001 olarak belirlenmiştir. “patience” parametresinin değeri 20 olarak belirlenmiştir. 20 iterasyon boyunca hata değerinde 0.0001 veya daha fazla bir değişim olmaması halinde “restore_best_weights” argümanı “True” seçilmiş ve bu sayede önceki iterasyonlarda elde edilmiş en iyi hata değerine karşılık gelen ağırlık değerlerinin modelde kullanılmak istendiği ifade edilmiştir. Sabit bir iterasyon sayısı belirlemek yerine, kaç iterasyon kullanılması gerektiği bu şekilde hesaplanarak eksik veya aşırı öğrenme durumlarının önüne geçilmiş ve önerilen modelin başarısı artırılmıştır.

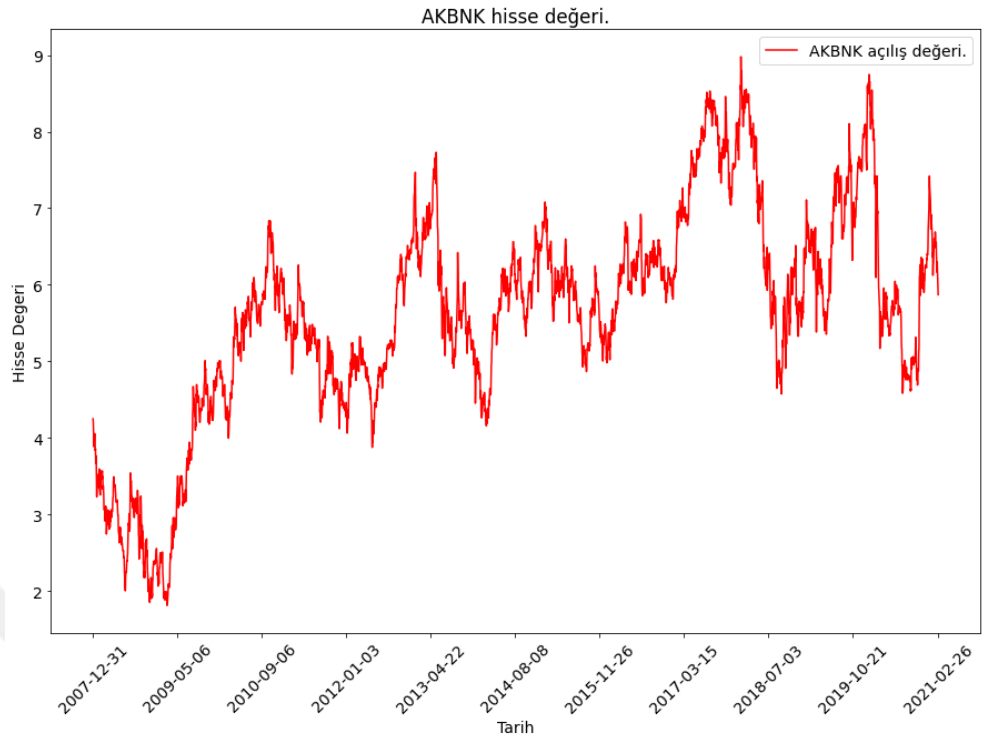
```
callback = EarlyStopping(
    monitor = "loss",
    min_delta = config["min_delta"], # 0.0001
    patience = config["patience"], # 20
    mode = "auto",
    restore_best_weights = True
)
```

Şekil 4. 7. Geliştirmiş olduğumuz modele ait erken durdurma kod parçası.

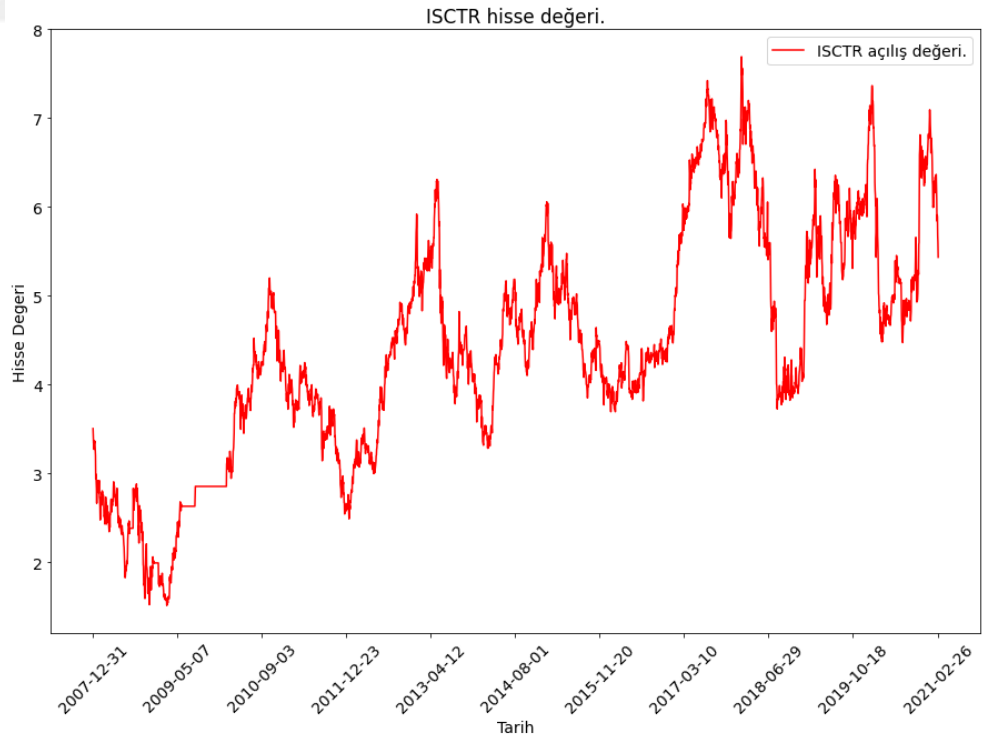
4.5. Modelin Validasyon ve Test Sonuçları

Bu tez çalışması kapsamında önerilen derin öğrenme modeli, BANKX indeksi içerisinde işlem gören AKBNK, GARAN, ISCTR, QNBFB ve VAKBN hisselerinin bir sonraki güne ait değerlerinin tahmin edilmesi için kullanılmıştır. Modelin eğitiminde kullanılan veri setlerinin grafiksel gösterimleri Şekil 4.8 – Şekil 4.14 arasında verilmiştir.

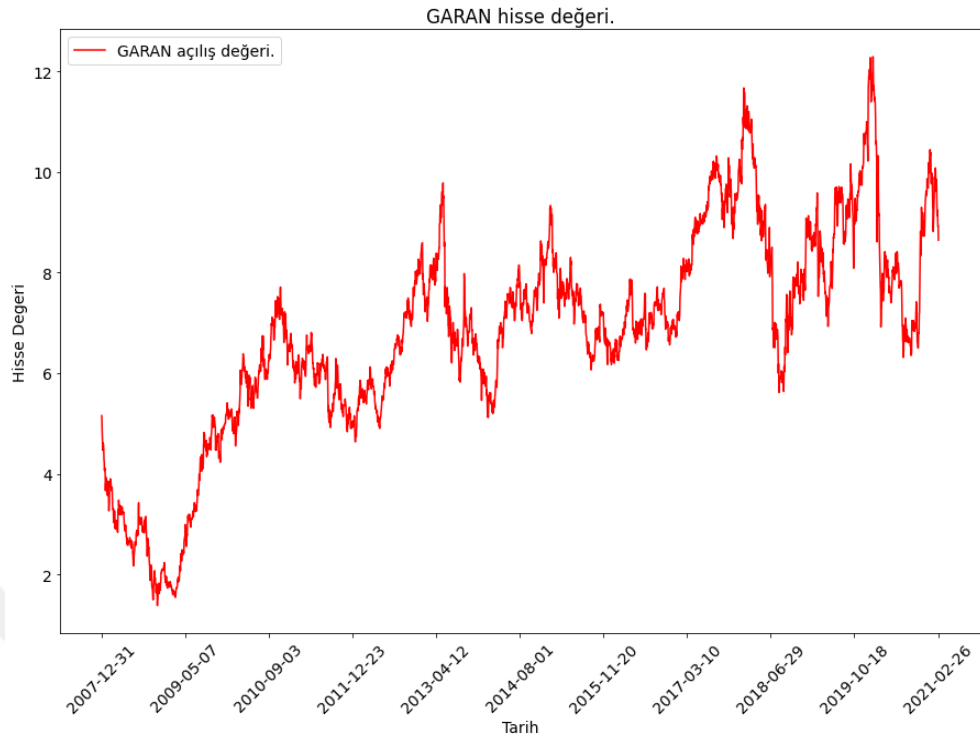
Önerilen modelin genel performansını ölçebilmek adına, eğitim veri setinin içerisinde olmayan zaman dilimlerine ait 2 aylık test verileri ayrıca hazırlanmıştır. Zaman adımlarına ait bağıl hata, mutlak hata ve karesel hata değerleri hesaplanıp grafikler ile ifade edilmiştir.



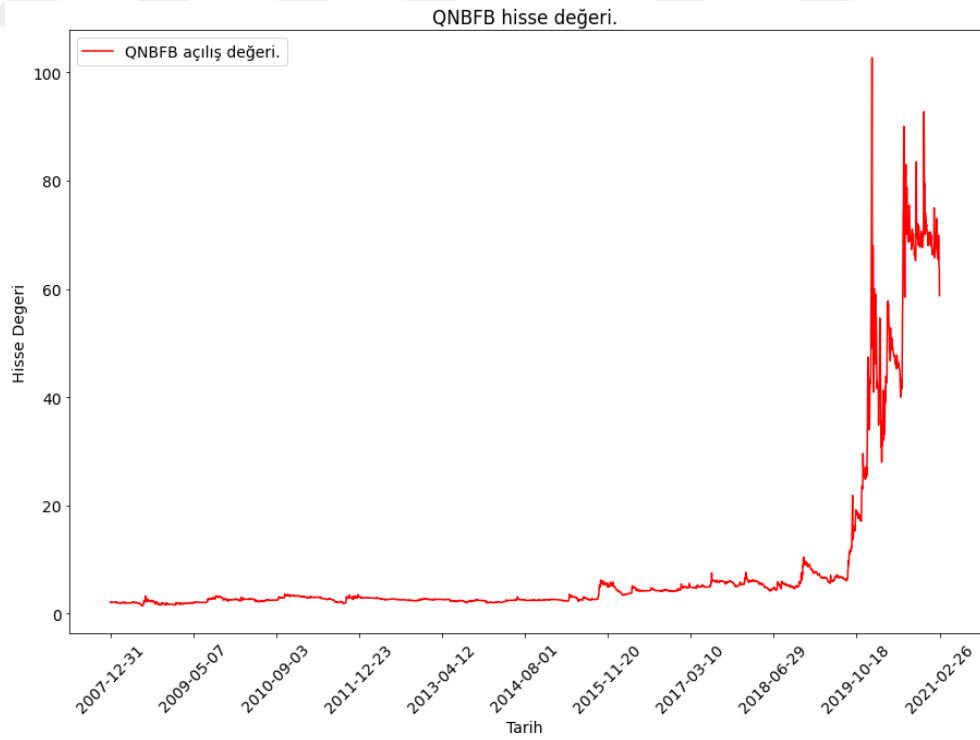
Şekil 4. 8. Modelimizin eğitiminde kullanılan AKBNK hissesine ait verilerin grafiği.



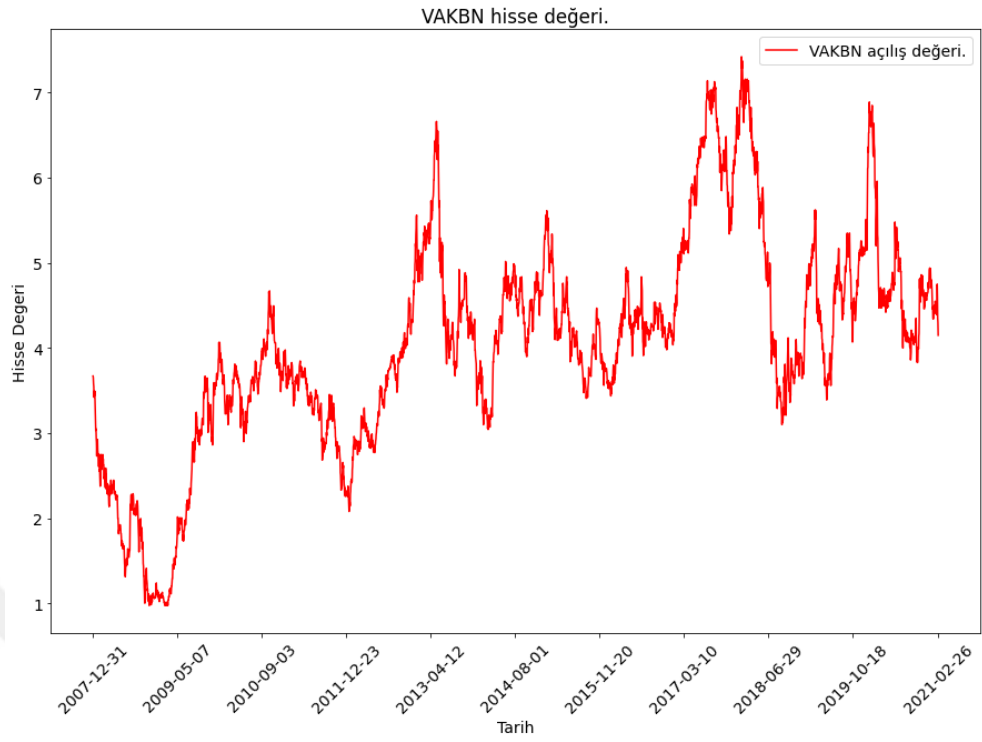
Şekil 4. 9. Modelimizin eğitiminde kullanılan ISCTR hissesine ait verilerin grafiği.



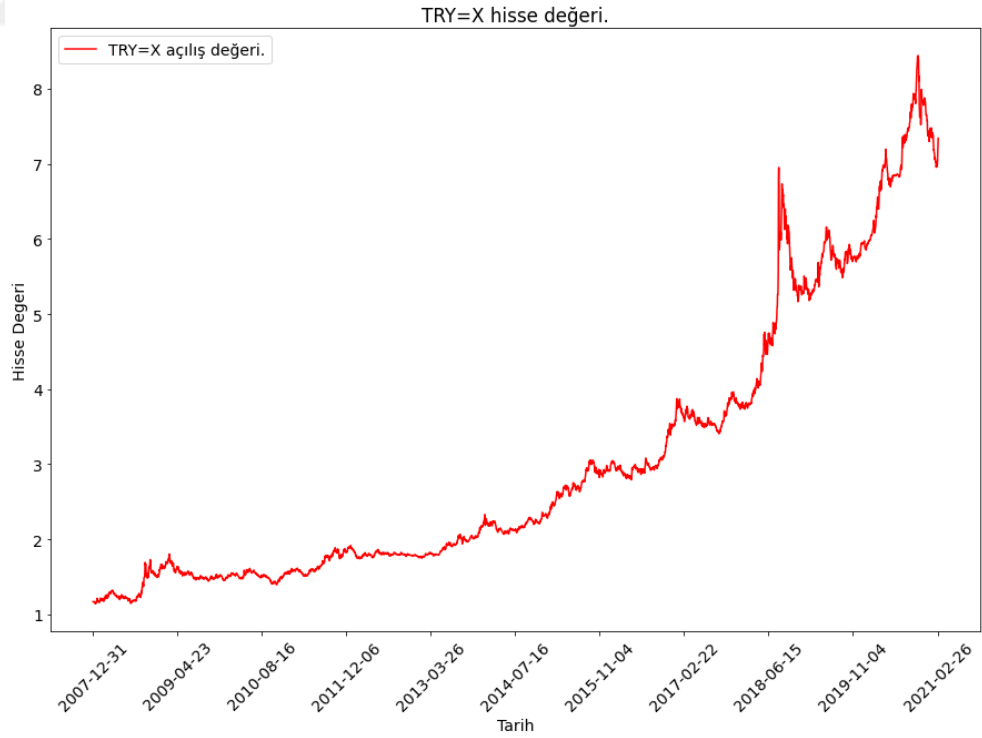
Şekil 4. 10. Modelimizin eğitiminde kullanılan GARAN hissesine ait verilerin grafiği.



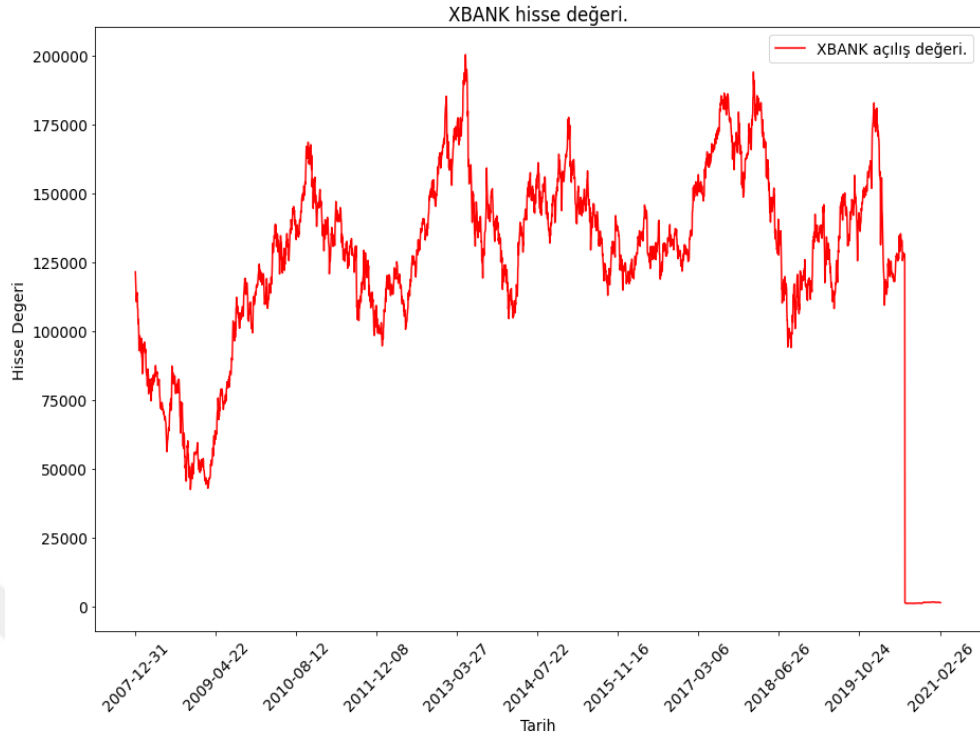
Şekil 4. 11. Modelimizin eğitiminde kullanılan QNBFB hissesine ait verilerin grafiği.



Şekil 4. 12. Modelimizin eğitiminde kullanılan VAKBN hissesine ait verilerin grafiği.

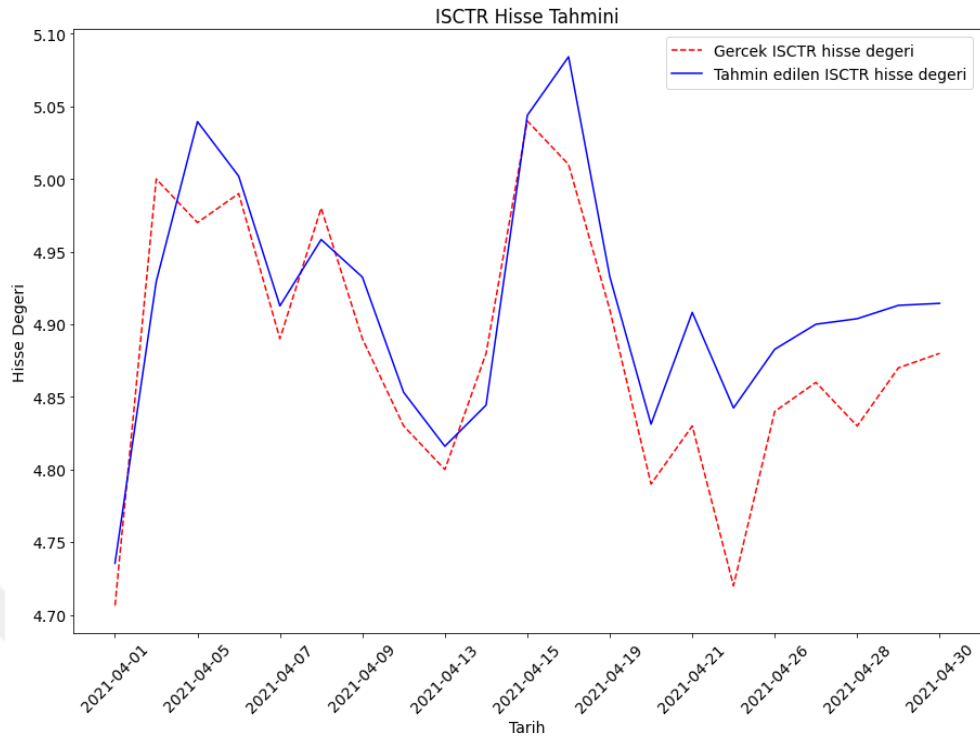


Şekil 4. 13. Modelimizin eğitiminde kullanılan USD/TRY endeksine ait verilerin grafiği.

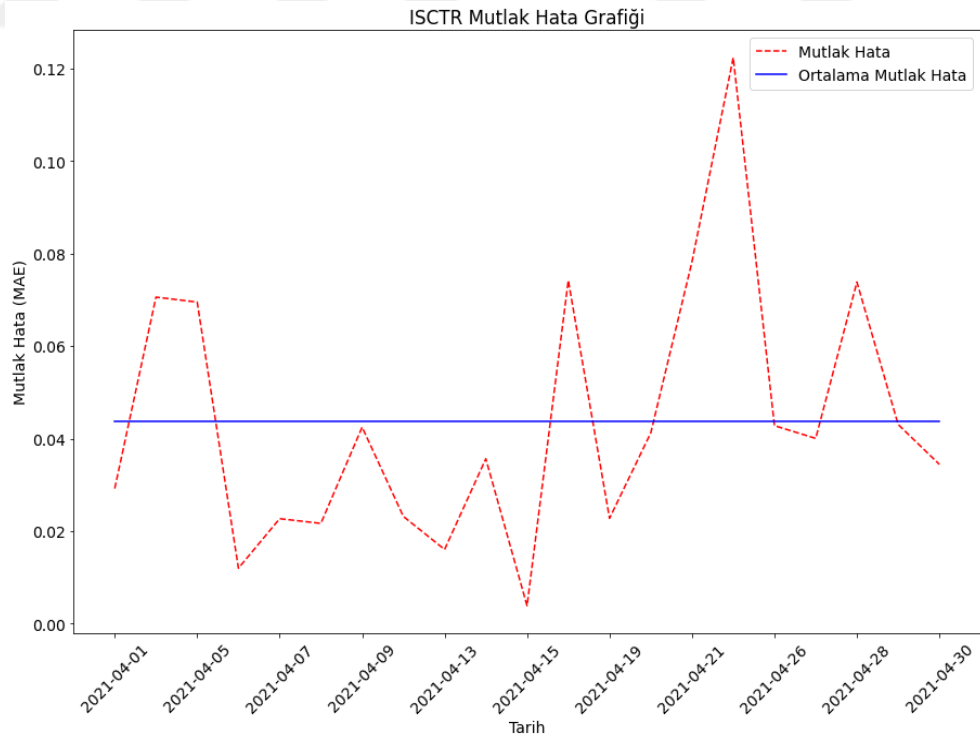


Şekil 4. 14. Modelimizin eğitiminde kullanılan XBANK hissesine ait verilerin grafiği.

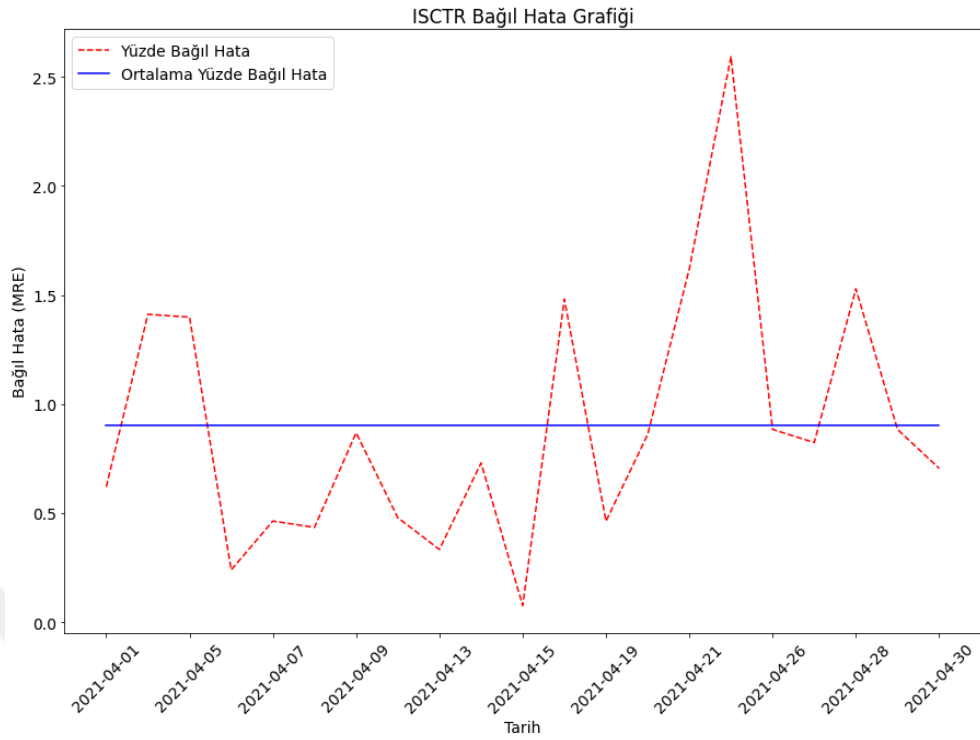
ISCTR hissesinin Nisan ayına ait fiyat tahminlerinin gerçek değerlere karşılık çizilmesi ile elde edilen grafik Şekil 4.15. ile gösterilmiştir. Nisan ayı içerisinde ISCTR hisse değerleri tahminleri sonucu hesaplanmış günlük mutlak hatalar ve nisan ayının ortalama mutlak hatası Şekil 4.16’da bağıl hata ise Şekil 4.17 ile gösterilmiştir. Hata değerlerinin gün bazlı olarak grafik haline getirilmesinde, elde edilmiş olan minimum ve maksimum hata değerlerinin tespit edilmesi amaçlanmıştır. Nisan ayı için Şekil 4.15 – Şekil 4.17 ile ifade edilmiş olan değerler mayıs ayı için Şekil 4.18 – Şekil 4.20 ile gösterilmiştir.



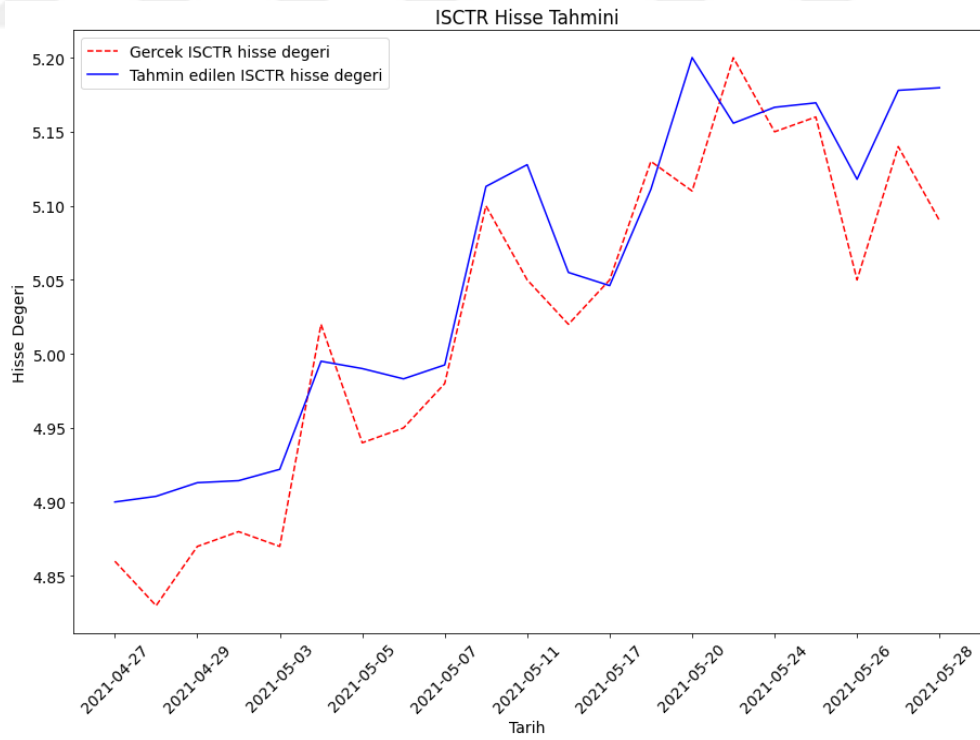
Şekil 4. 15. ISCTR hisse senedi için nisan ayına ait fiyat tahminleri.



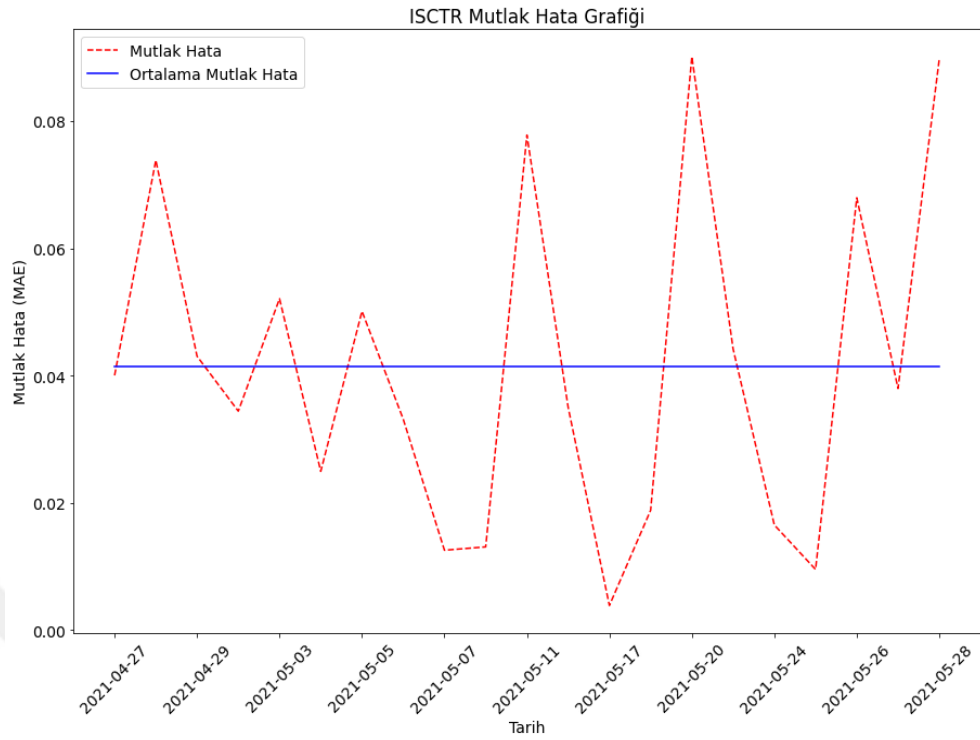
Şekil 4. 16. ISCTR hisse senedi için nisan ayı test sonuçlarına ait mutlak hata grafiği.



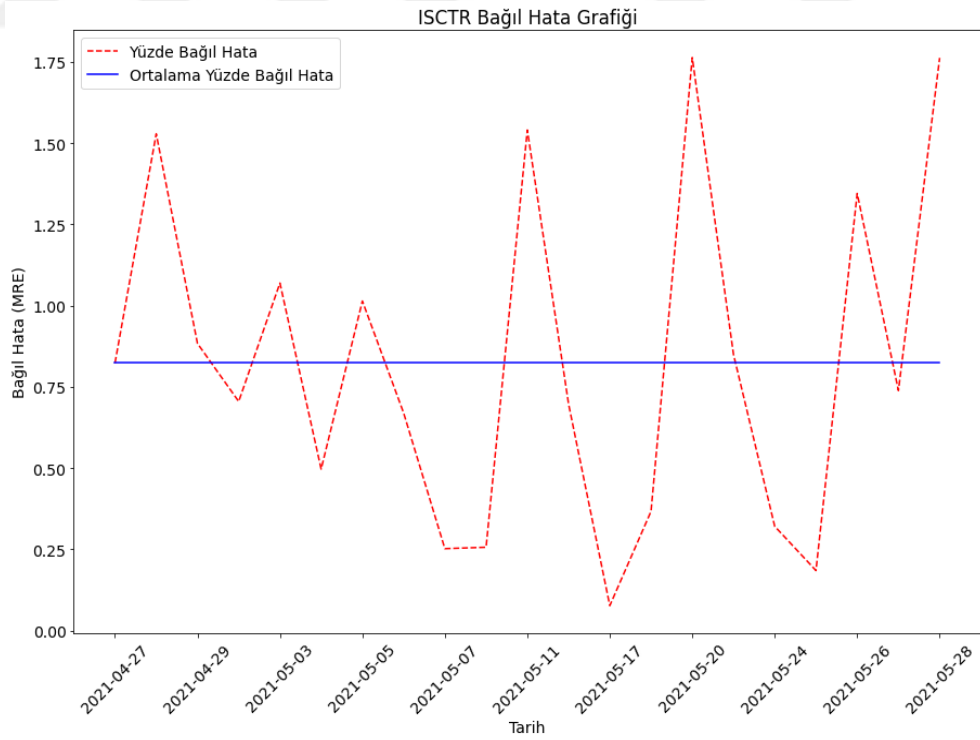
Şekil 4. 17. ISCTR hisse senedinin nisan ayı test sonuçlarına ait bağıl hata grafiği.



Şekil 4. 18. ISCTR hisse senedinin mayıs ayına ait test sonuçlarının grafiği.

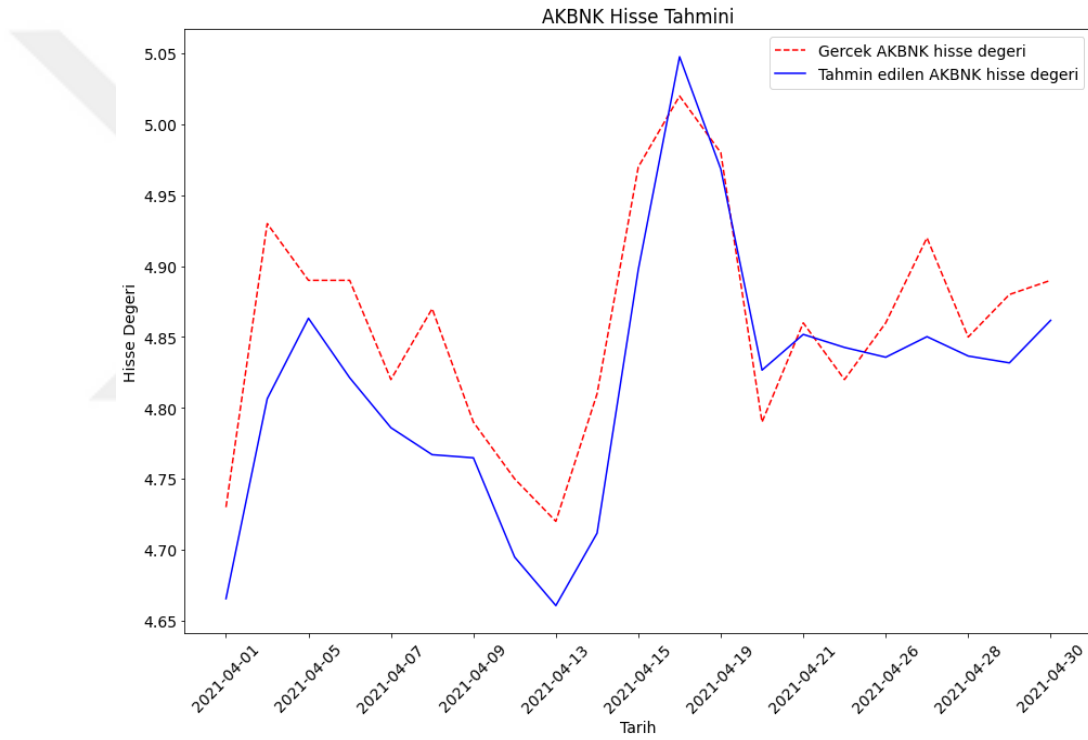


Şekil 4. 19. ISCTR hisse senedinin mayıs ayı test sonuçlarına ait mutlak hata grafiđi.

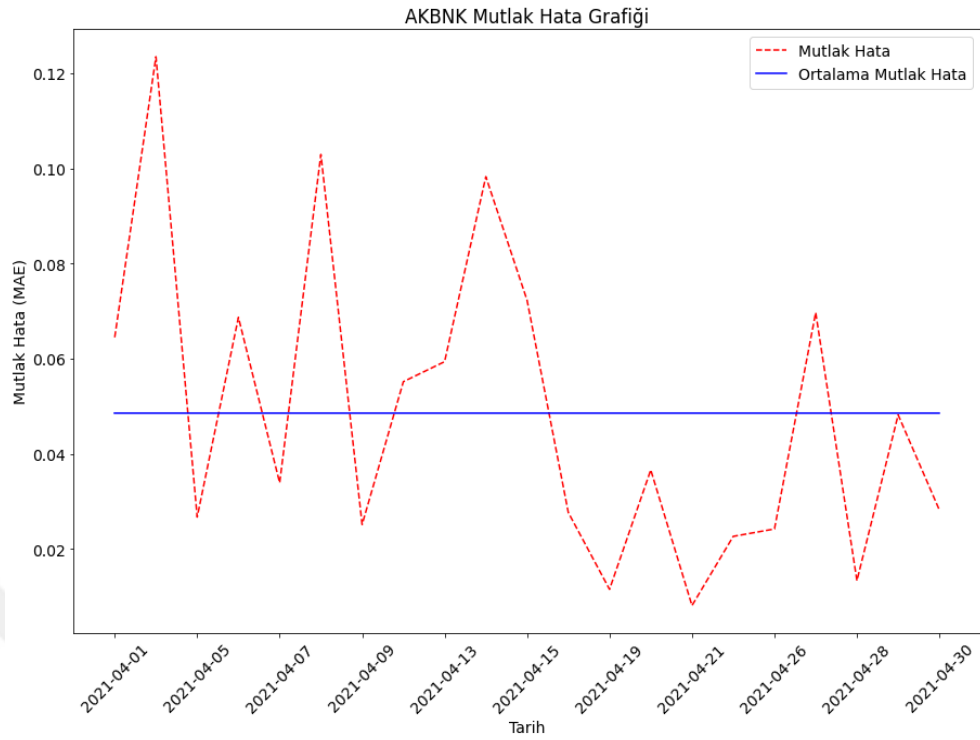


Şekil 4. 20. ISCTR hisse senedinin mayıs ayı test sonuçlarına ait bađıl hata grafiđi.

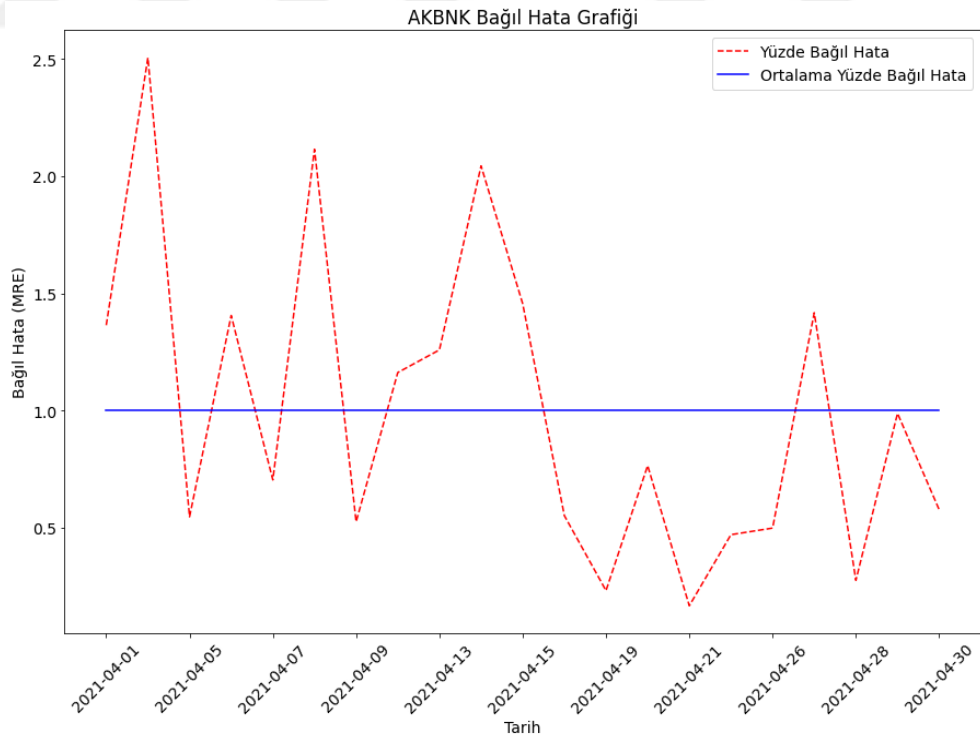
AKBNK hissesinin Nisan ayına ait fiyat tahminlerinin gerçek değerlere karşılık çizilmesi ile elde edilen grafik Şekil 4.21. ile gösterilmiştir. Nisan ayı içerisinde AKBNK hisse değerleri tahminleri sonucu hesaplanmış günlük mutlak hatalar ve nisan ayının ortalama mutlak hatası Şekil 4.22’da bağıl hata ise Şekil 4.23 ile gösterilmiştir. Hata değerlerinin gün bazlı olarak grafik haline getirilmesinde, elde edilmiş olan minimum ve maksimum hata değerlerinin tespit edilmesi amaçlanmıştır. Nisan ayı için Şekil 4.21 – Şekil 4.23 ile ifade edilmiş olan değerler mayıs ayı için Şekil 4.24 – Şekil 4.26 ile gösterilmiştir.



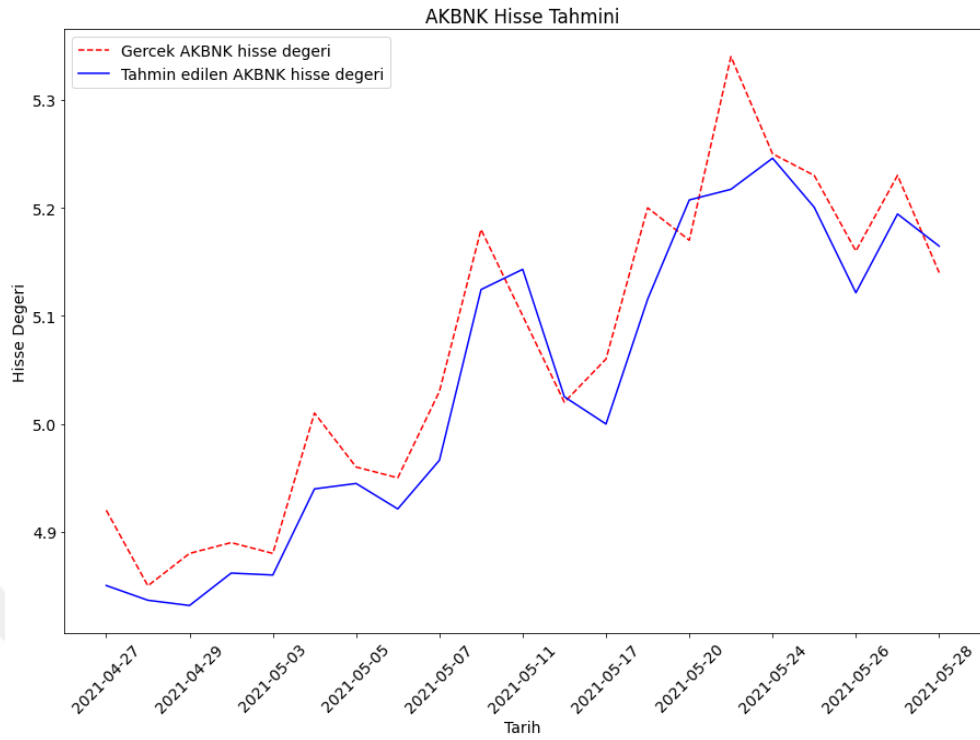
Şekil 4. 21. AKBNK hisse senedinin nisan ayına ait test sonuçlarının grafiği.



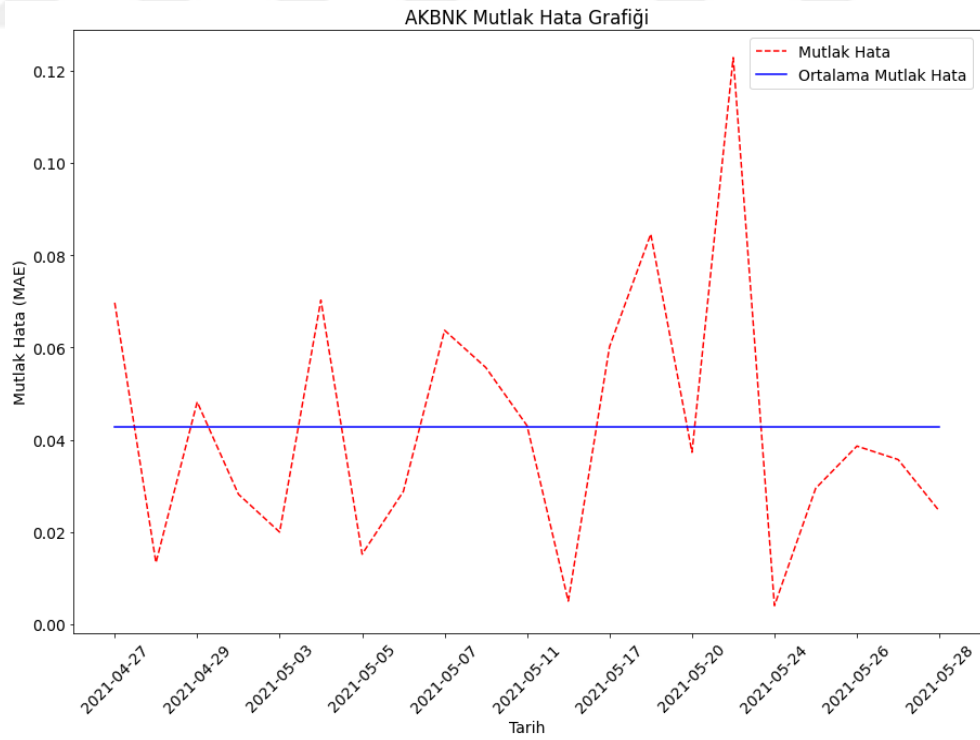
Şekil 4. 22. AKBNK hisse senedinin nisan ayı test sonuçlarına ait mutlak hata grafiđi.



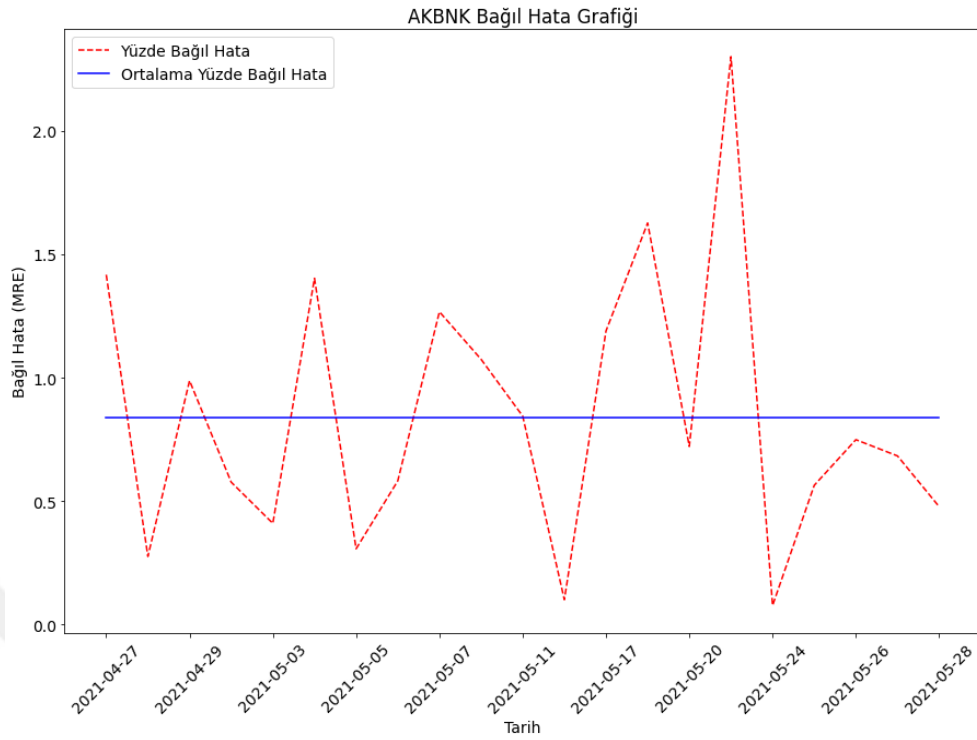
Şekil 4. 23. AKBNK hisse senedinin nisan ayı test sonuçlarına ait bađıl hata grafiđi.



Şekil 4. 24. AKBNK hisse senedinin mayıs ayına ait test sonuçlarının grafiği.

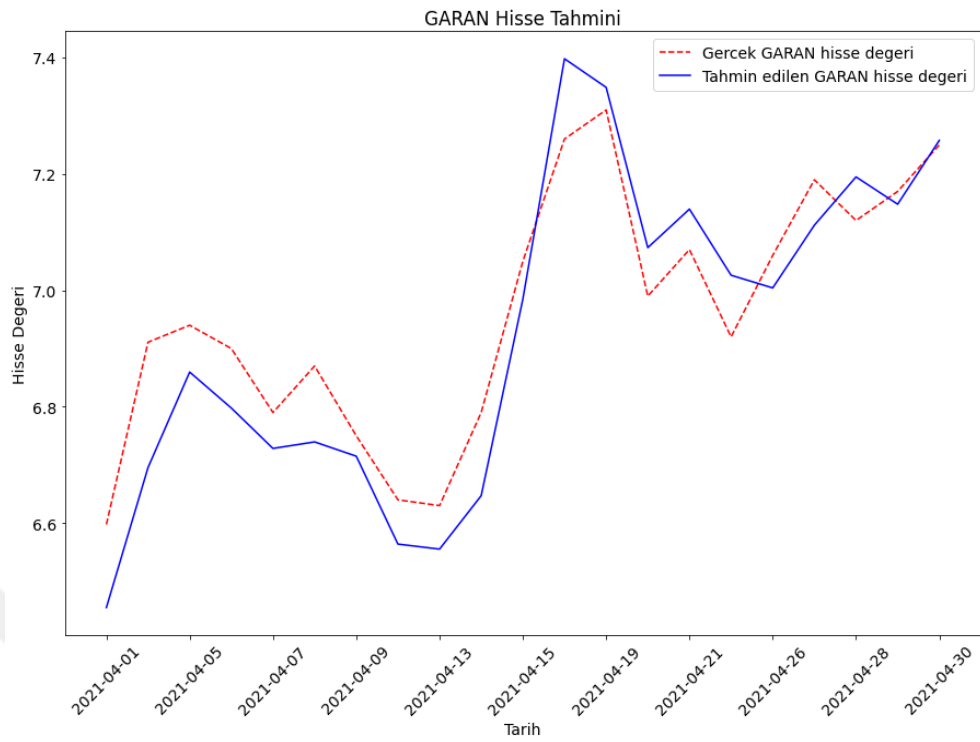


Şekil 4. 25. AKBNK hisse senedinin mayıs ayı test sonuçlarına ait mutlak hata grafiği.

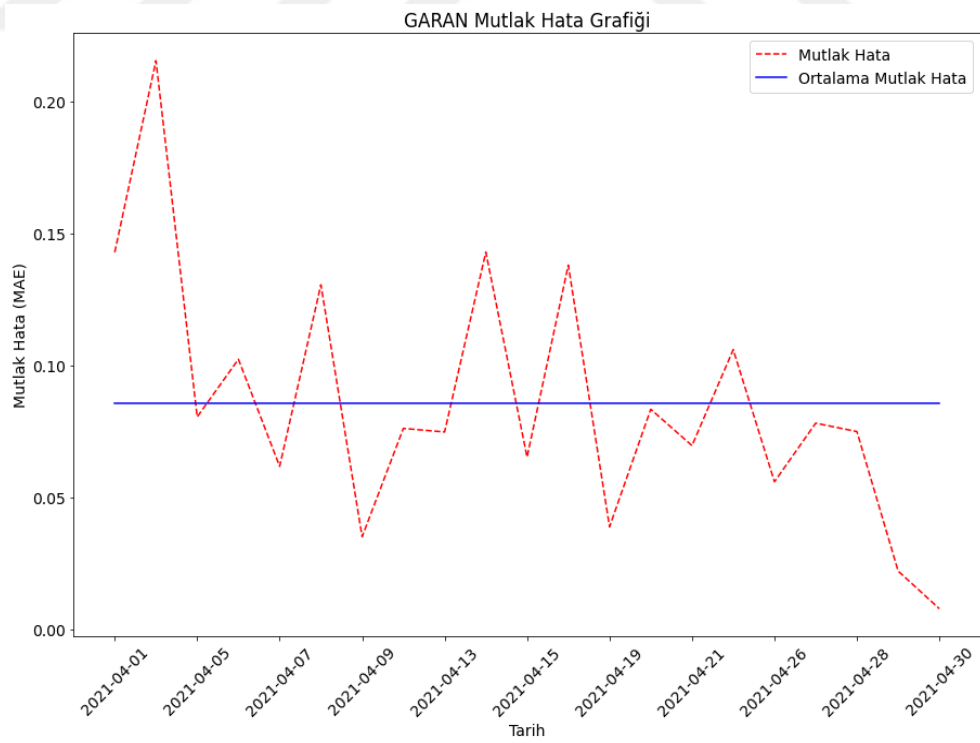


Şekil 4. 26. AKBNK hisse senedinin mayıs ayı test sonuçlarına ait bağıl hata grafiği.

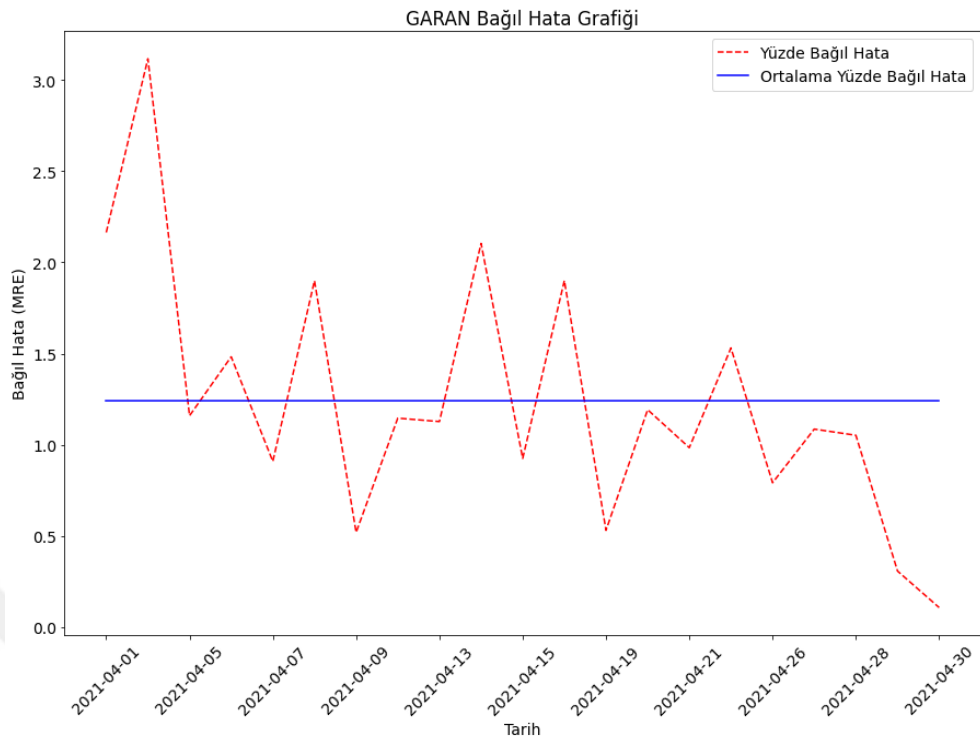
GARAN hissesinin Nisan ayına ait fiyat tahminlerinin gerçek değerlere karşılık çizilmesi ile elde edilen grafik Şekil 4.27. ile gösterilmiştir. Nisan ayı içerisinde GARAN hisse değerleri tahminleri sonucu hesaplanmış günlük mutlak hatalar ve nisan ayının ortalama mutlak hatası Şekil 4.28’de bağıl hata ise Şekil 4.29 ile gösterilmiştir. Hata değerlerinin gün bazlı olarak grafik haline getirilmesinde, elde edilmiş olan minimum ve maksimum hata değerlerinin tespit edilmesi amaçlanmıştır. Nisan ayı için Şekil 4.27 – Şekil 4.29 ile ifade edilmiş olan değerler mayıs ayı için Şekil 4.30 – Şekil 4.32 ile gösterilmiştir.



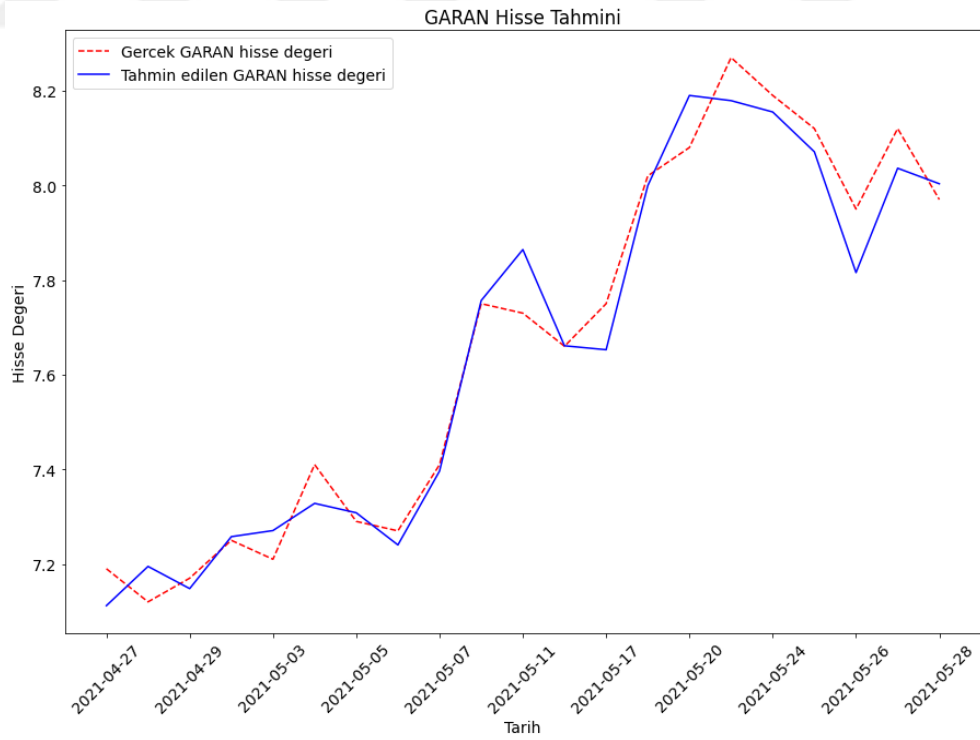
Şekil 4. 27. GARAN hisse senedinin nisan ayına ait test sonuçlarının grafiği.



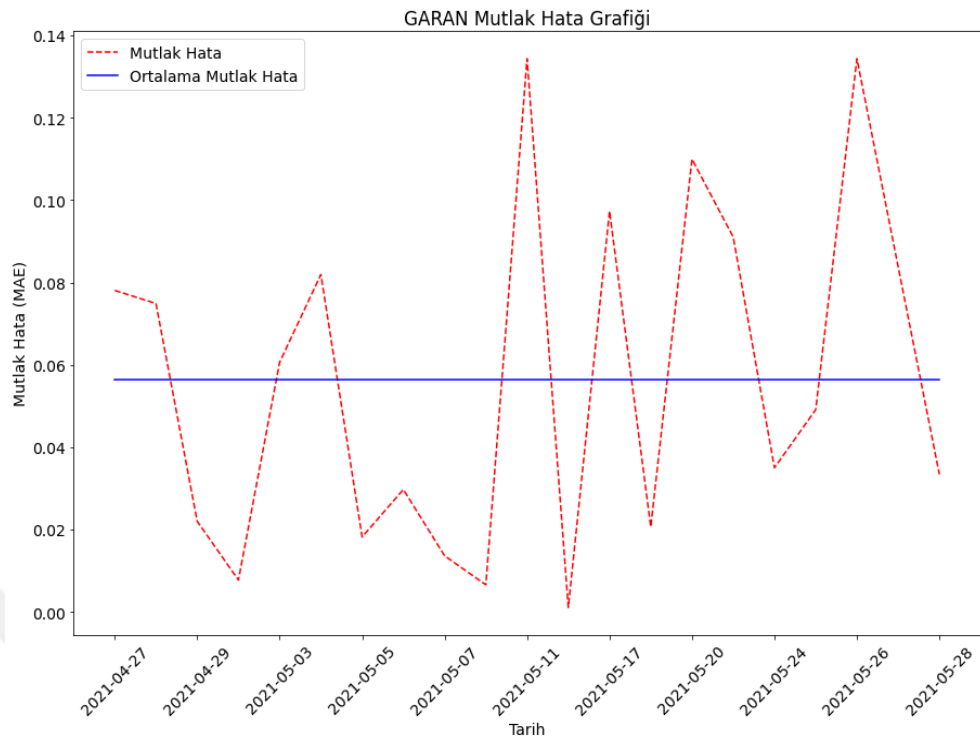
Şekil 4. 28. GARAN hisse senedinin nisan ayı test sonuçlarına ait mutlak hata grafiği.



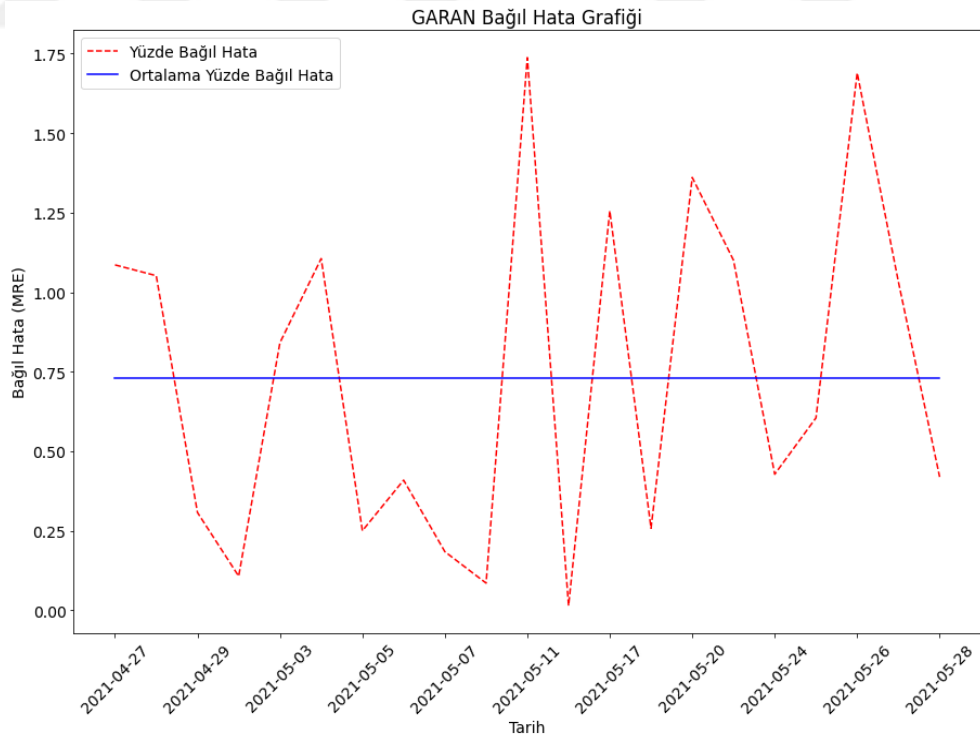
Şekil 4. 29. GARAN hisse senedinin nisan ayı test sonuçlarına ait bağıl hata grafiği.



Şekil 4. 30. GARAN hisse senedinin mayıs ayına ait test sonuçlarının grafiği.

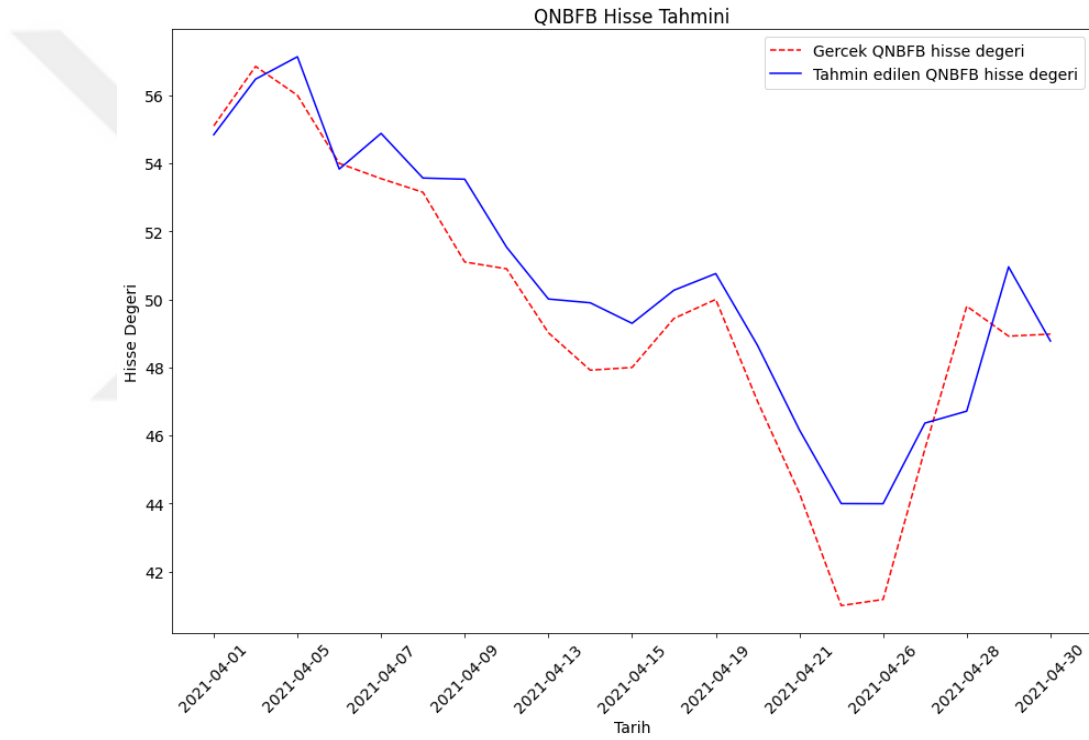


Şekil 4. 31. GARAN hisse senedinin mayıs ayı test sonuçlarına ait mutlak hata grafiği.

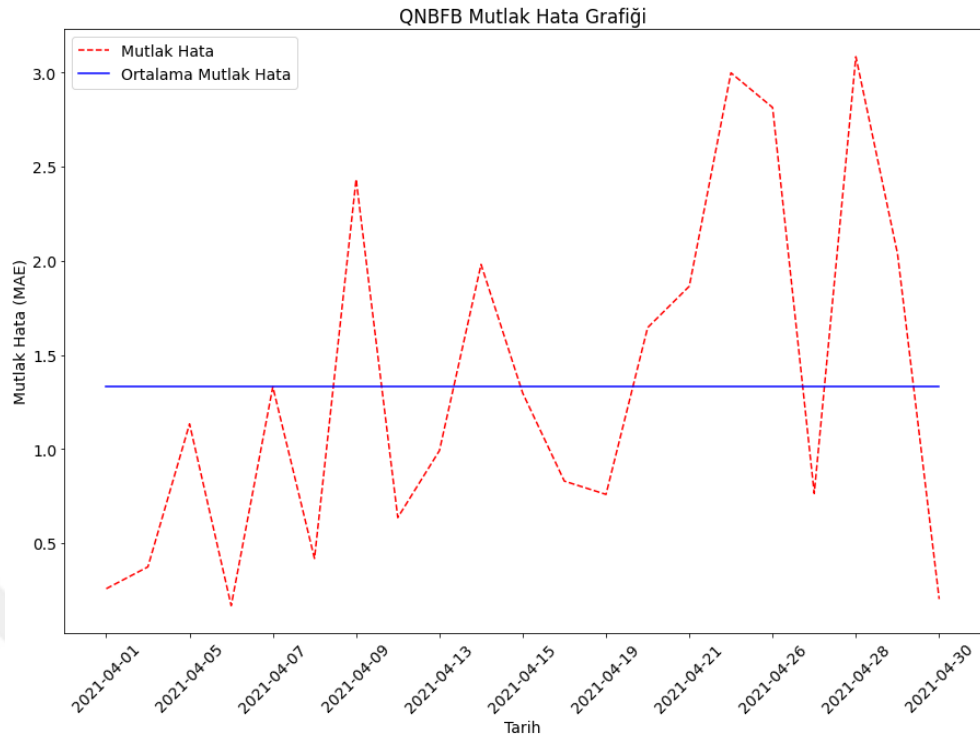


Şekil 4. 32. GARAN hisse senedinin mayıs ayı test sonuçlarına ait bağıl hata grafiği.

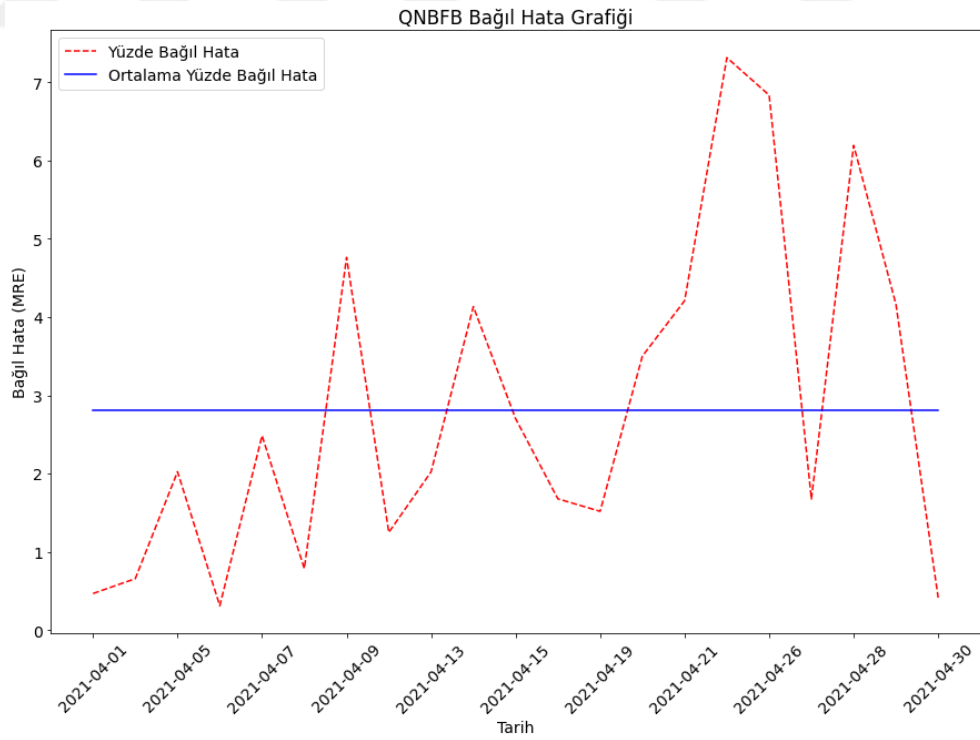
QNBFB hissesinin Nisan ayına ait fiyat tahminlerinin gerçek değerlere karşılık çizilmesi ile elde edilen grafik Şekil 4.33. ile gösterilmiştir. Nisan ayı içerisinde QNBFB hisse değerleri tahminleri sonucu hesaplanmış günlük mutlak hatalar ve nisan ayının ortalama mutlak hatası Şekil 4.34'te bağıl hata ise Şekil 4.35 ile gösterilmiştir. Hata değerlerinin gün bazlı olarak grafik haline getirilmesinde, elde edilmiş olan minimum ve maksimum hata değerlerinin tespit edilmesi amaçlanmıştır. Nisan ayı için Şekil 4.33 – Şekil 4.35 ile ifade edilmiş olan değerler mayıs ayı için Şekil 4.36 – Şekil 4.38 ile gösterilmiştir.



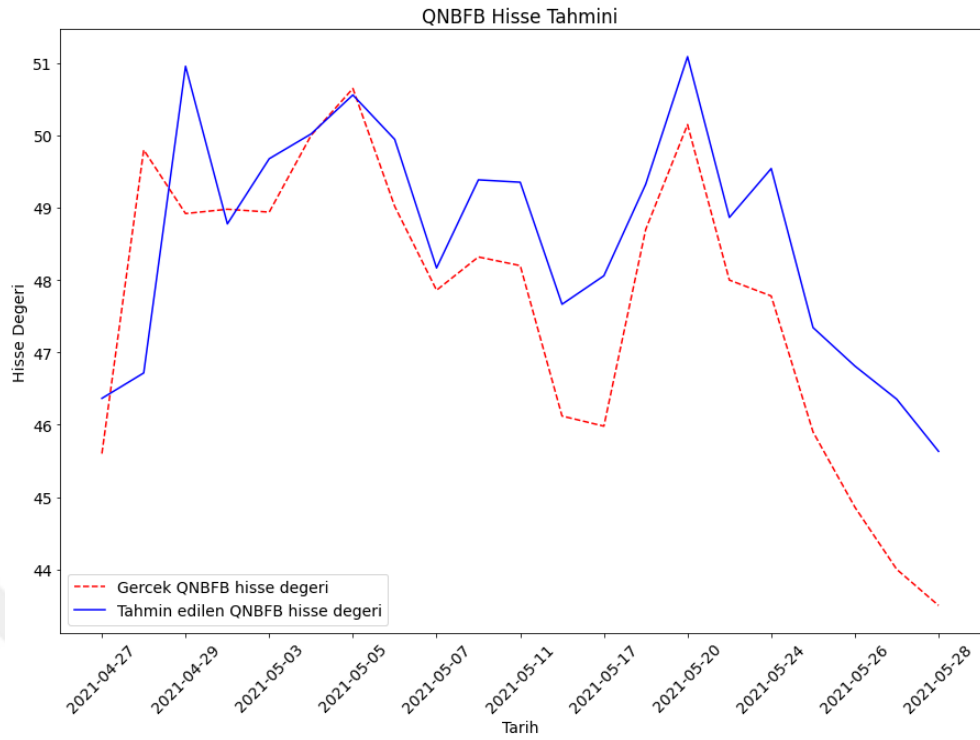
Şekil 4. 33. QNBFB hisse senedinin nisan ayına ait test sonuçlarının grafiği.



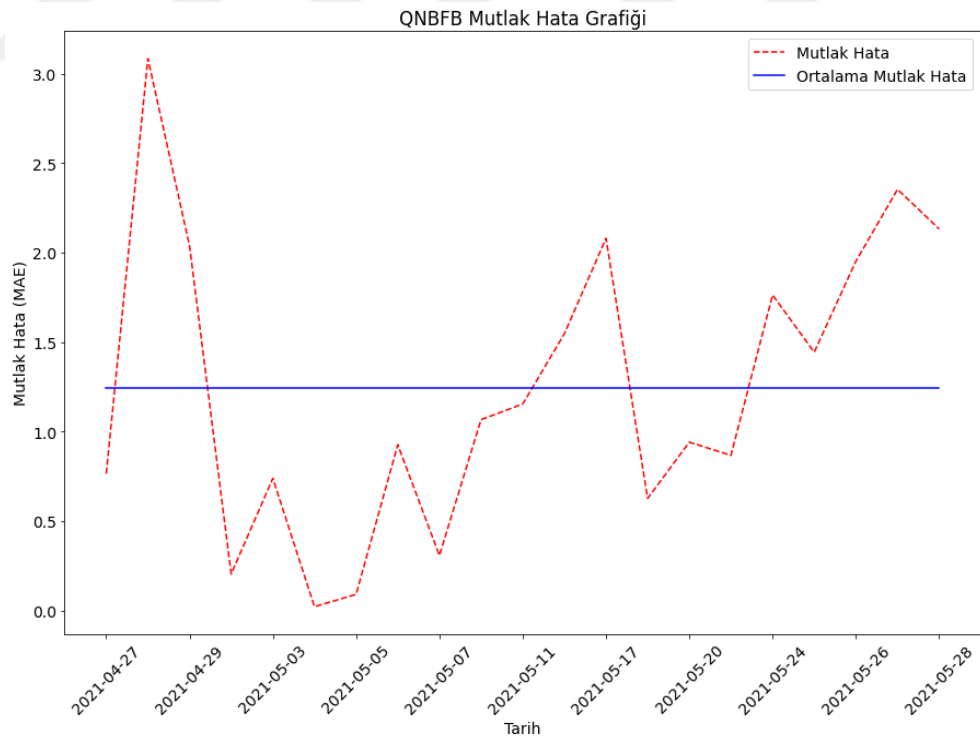
Şekil 4. 34. QNBFB hisse senedinin nisan ayı test sonuçlarına ait bađıl hata grafiđi.



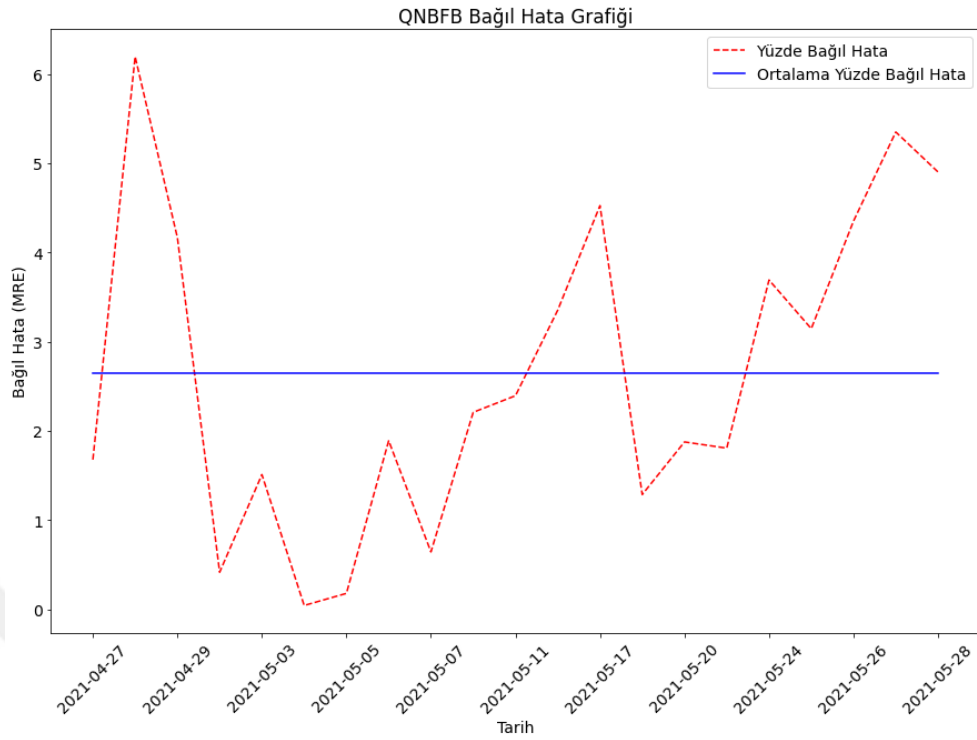
Şekil 4. 35. QNBFB hisse senedinin nisan ayı test sonuçlarına ait mutlak hata grafiđi.



Şekil 4. 36. QNBFB hisse senedinin mayıs ayına ait test sonuçlarının grafiği.

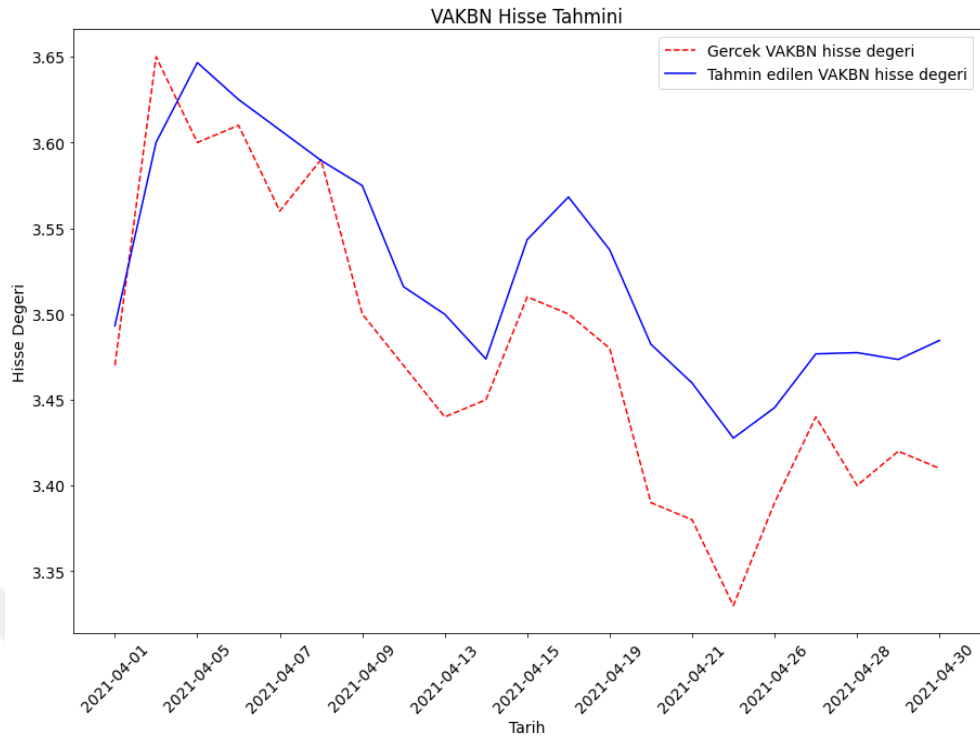


Şekil 4. 37. QNBFB hisse senedinin mayıs ayı test sonuçlarına ait mutlak hata grafiği.

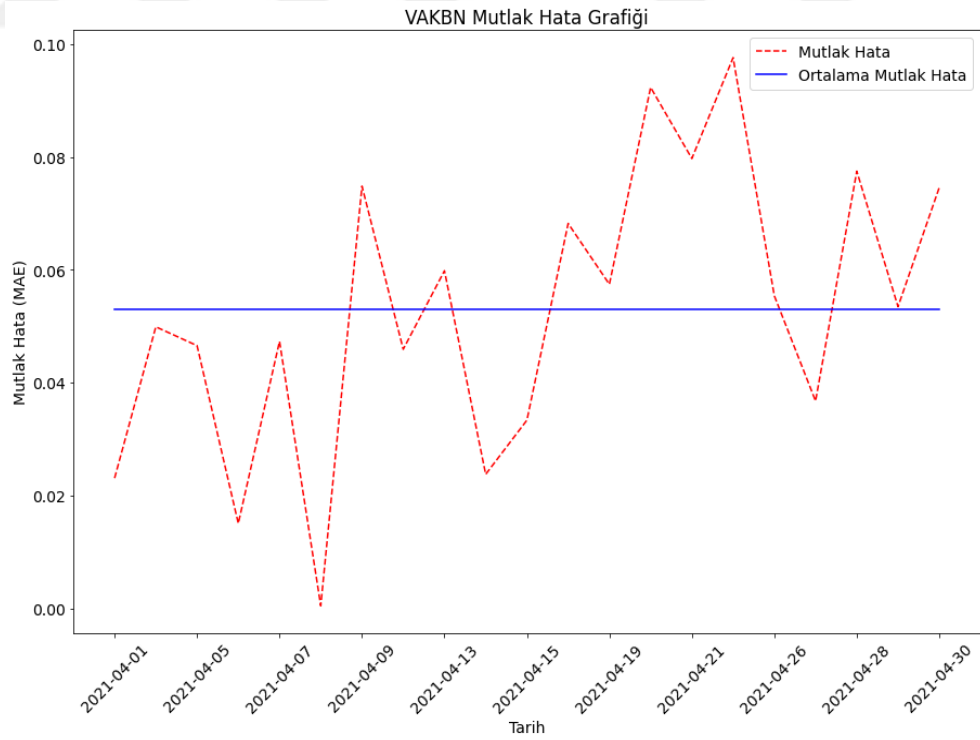


Şekil 4. 38. QNBFB hisse senedinin mayıs ayı test sonuçlarına ait bağıl hata grafiği.

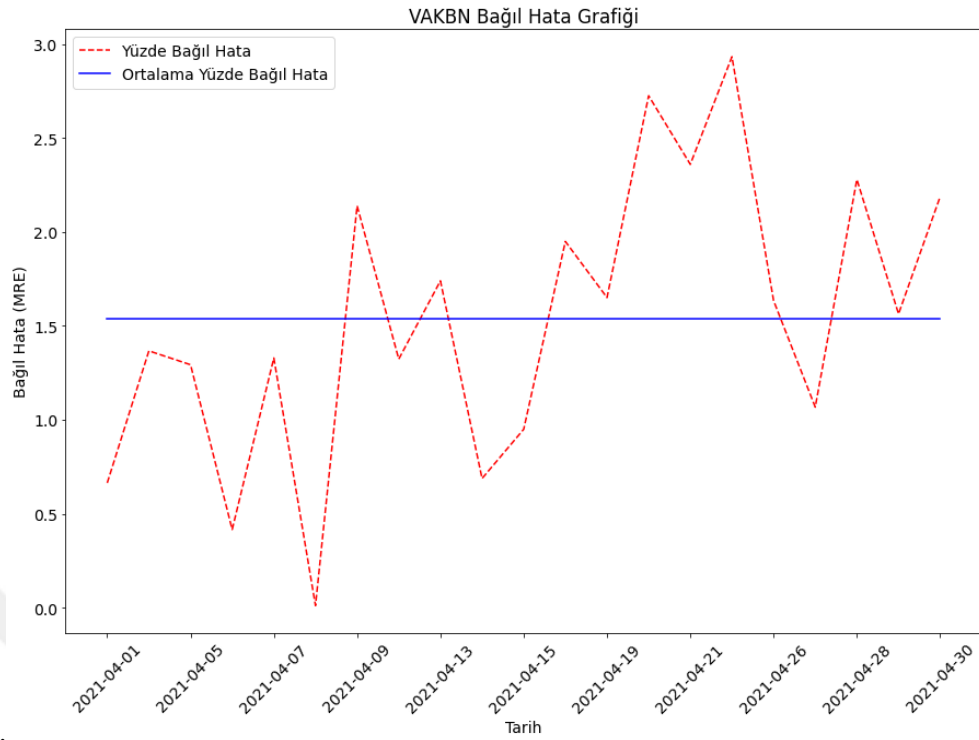
VAKBN hissesinin Nisan ayına ait fiyat tahminlerinin gerçek değerlere karşılık çizilmesi ile elde edilen grafik Şekil 4.39 ile gösterilmiştir. Nisan ayı içerisinde VAKBN hisse değerleri tahminleri sonucu hesaplanmış günlük mutlak hatalar ve nisan ayının ortalama mutlak hatası Şekil 4.40'ta bağıl hata ise Şekil 4.41 ile gösterilmiştir. Hata değerlerinin gün bazlı olarak grafik haline getirilmesinde, elde edilmiş olan minimum ve maksimum hata değerlerinin tespit edilmesi amaçlanmıştır. Nisan ayı için Şekil 4.39 – Şekil 4.41 ile ifade edilmiş olan değerler mayıs ayı için Şekil 4.42 – Şekil 4.44 ile gösterilmiştir.



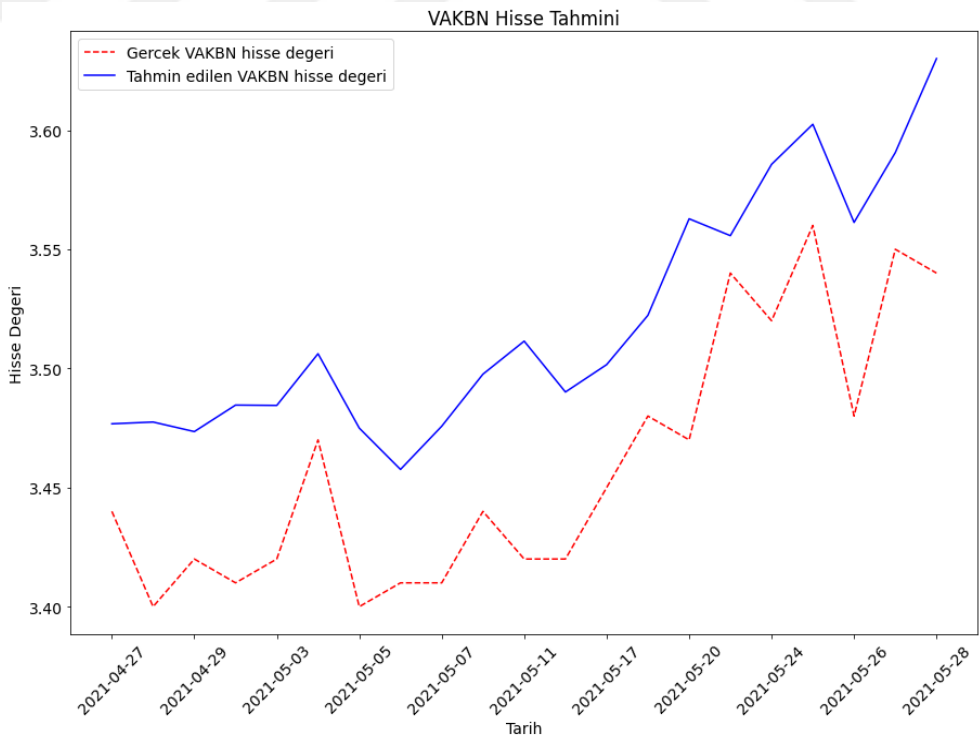
Şekil 4. 39. VAKBN hisse senedinin nisan ayına ait test sonuçlarının grafiği.



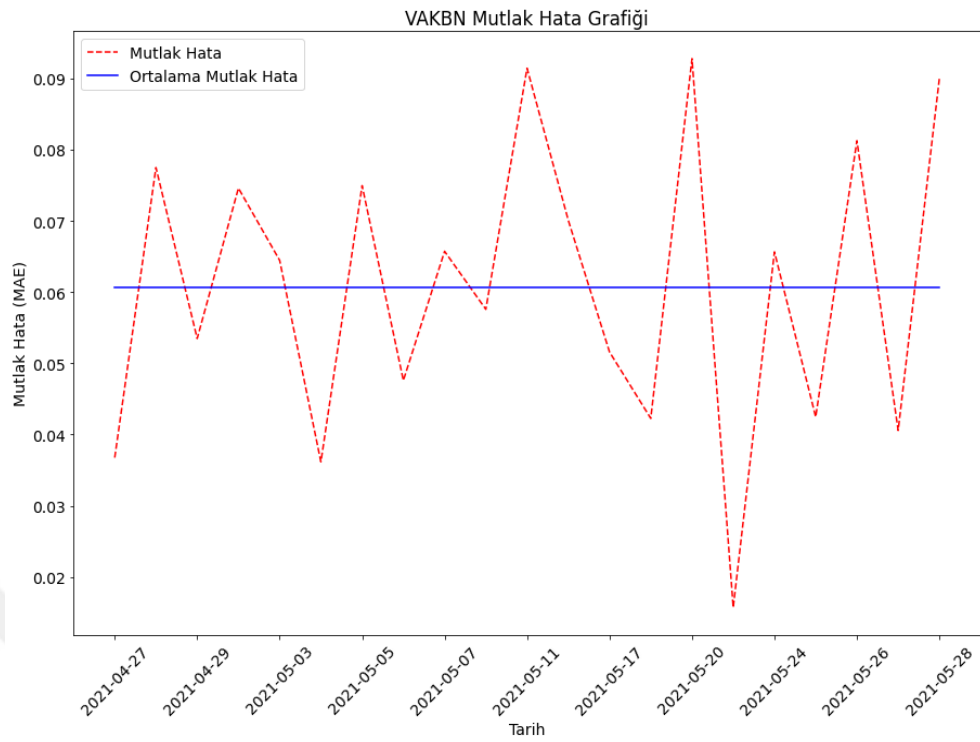
Şekil 4. 40. VAKBN hisse senedinin nisan ayı test sonuçlarına ait mutlak hata grafiği



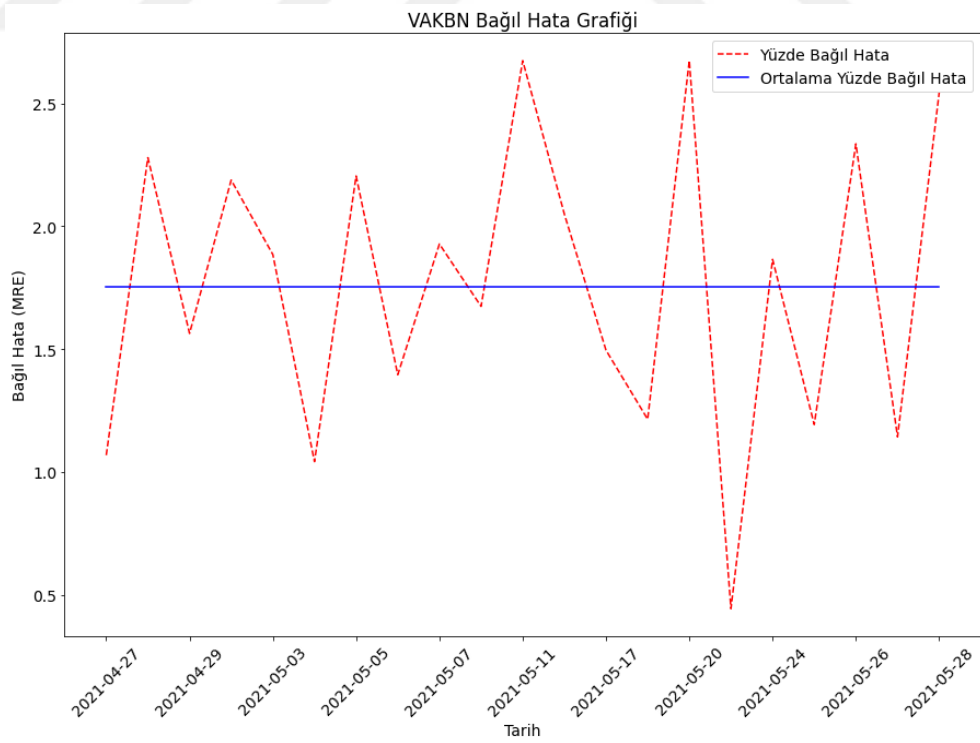
Şekil 4. 41. VAKBN hisse senedinin nisan ayı test sonuçlarına ait bağıl hata grafiği.



Şekil 4. 42. VAKBN hisse senedinin mayıs ayına ait test sonuçlarının grafiği.



Şekil 4. 43. VAKBN hisse senedinin mayıs ayı test sonuçlarına ait mutlak hata grafiđi.



Şekil 4. 44. VAKBN hisse senedinin mayıs ayı test sonuçlarına ait bađıl hata grafiđi.

Borsa İstanbul'da işlem gören 5 bankanı hisselerine ilişkin fiyat hareketlerinin tahmini için eğitilen derin öğrenme modelinin test sonuçları sayısal formda Tablo 4.2'de verilmiştir. Model performansının doğru yorumlanabilmesi için MSE, MAE ve MRE metrikleri birlikte kullanılmıştır. Farklı hisse senetlerine ait modelin tahmin performansını kıyaslamak için hisse değerlerinin genlik aralıklarının farklı olması sebebiyle MRE metriği kullanılmıştır. Bu bağlamda en düşük validasyon performansı 0.01686 ortalama bağıl hata ile QNBFB hisse değeri üzerinde görmekteyiz. Bunun sebebi Şekil 4.11'de görüleceği üzere volatilitenin ve düzensizliğin en yüksek olduğu hisselerin yine QNBFB ye ait olması olarak yorumlanabilir. Nisan ayı test döneminde ortalama 0.01498 MRE ve mayıs ayına ait test döneminde ortalama 0.01350 MRE hata değerleri elde edilmiştir.

Tablo 4. 2 Modele ait validasyon ve test sonuçları.

	HATA TÜRÜ	VALİDASYON	TEST	TEST
		2021-03-01 2021-04-01	2021-04-01 2021-04-30	2021-05-01 2021-05-28
AKBNK	MSE	0.03120	0.00335	0.00255
	MAE	0.08190	0.04863	0.04194
	MRE	0.01600	0.01001	0.00823
GARAN	MSE	0.05507	0.00956	0.00495
	MAE	0.15264	0.08577	0.05582
	MRE	0.01519	0.01240	0.00721
ISCTR	MSE	0.01844	0.00269	0.00233
	MAE	0.07342	0.04379	0.04068
	MRE	0.01465	0.00900	0.00813
QNBFB	MSE	2.39283	2.62041	2.19611
	MAE	1.04160	1.33437	1.24201
	MRE	0.01686	0.02815	0.02646
VAKBN	MSE	0.00898	0.00342	0.00406
	MAE	0.05694	0.05301	0.06035
	MRE	0.01495	0.01537	0.01748
ORTALAMA	MSE	0.50130	0.52788	0.44200
	MAE	0.28130	0.31311	0.28816
	MRE	0.01633	0.01498	0.01350

5. BÖLÜM

SONUÇ

5.1 Sonuçların Değerlendirilmesi

Bu tez çalışmasında, BORSA İSTANBUL'da işlem gören 5 farklı bankanın hisselerinin kısa vade değerlerinin derin öğrenme modelleri ile tahmin edilmesi amaçlanmıştır. Bu kapsamda iki farklı bilgisayar ortamı, iki farklı derin öğrenme modeli ve farklı hiper parametreler kullanılarak toplamda doksan altı farklı model ortaya konmuş ve en uygun model-hiper parametre kombinasyonu tespit edilmeye çalışılmıştır. Bu tez çalışması kapsamında Tablo 4.1'de verilen MSE cinsinden hata değerleri, LSTM modelinin GRU modeline göre daha düşük hata ile fiyat tahmini yaptığını göstermiştir.

Şekil 4.1 ve Şekil 4.2 dikkate alındığında önerilen modelin eğitilme ortamı da tahmin başarısını ve hızını etkilemektedir. Google firmasının yapay öğrenme uygulamalarında kullanılması amacıyla geliştirmiş olduğu Tensör İşlem Birimlerinin (TPU) her bir iterasyonda üç saniye kadar daha hızlı olduğu açıkça görülmektedir.

Elde edilen bu deneysel sonuçlar ışığında ISCTR, VAKBN, GARAN, QNBFB ve AKBNK hisse değerlerinin fiyat tahmini için LSTM tabanlı modeller ve hiper parametreler tercih edilmiş ve bu doğrultuda tahmin çalışmaları ağırlık kazanmıştır. Bu tahminler yapılırken USD/TRY, XBANK ve BIST30 endekslerine ait veriler de modele girdi olarak dahil edilmiştir. LSTM modeli çoktan teke yapıda kullanılmış olup altmış dört günlük kayar pencere yapısında kullanılmış ve önceki altmış dört güne ait veri altmış beşinci günün değerinin tahmininde kullanılmıştır.

Fiyat tahmini yapılan borsa hisselerinin değer aralıklarının ve değer genliklerinin farklı olmaları sebebiyle eğitilen modellerin tahmin performansı değerlendirilirken 3.Bölümde

tarif edilen ortalama bağıl hata, ortalama karesel hata ve ortalama mutlak hata metrikleri kullanılmıştır. Önerilen modelin genelliğinin kontrol edilebilmesi amacı ile veri kümesi eğitim, validasyon ve iki farklı test olmak üzere dört parçaya bölünmüştür ve iki ayaklı test süreci için ayrı ayrı hata değerleri hesaplanmıştır.

Tablo 4.2 dikkate alındığında, yapılan çalışmalar neticesinde önerilen LSTM tabanlı derin öğrenme modeli, Nisan 2021 dönemi için 5 farklı hisse arasında en düşük hata ile ISCTR hissesine ilişkin sonuçlar üretmiştir. ISCTR'ye ait test verilerinin tahmininde MRE cinsinden en düşük 0.009 hataya ulaşılmıştır. Ayrıca ISCTR hissesi için MSE ve MAE cinsinden hata değerleri de sırasıyla 0.00269 ve 0.04379 şeklinde hesaplanmıştır. Aynı model, Mayıs 2021 dönemi için ise GARAN hissesine ilişkin fiyat tahminlerinde en düşük hataya ulaşmıştır. GARAN hissesi için MRE cinsinden en düşük 0.00721 hata ile sonuç üretilmiştir. Benzer olarak GARAN hissesi için MSE ve MAE cinsinden test hataları da 0.00956 ve 0.08577 şeklindedir.

Yine Tablo 4.2 referans alındığında Nisan 2021 dönemi için fiyat tahmini yapılan VAKBN, QNBFB ve AKBNK hisselerine ilişkin MRE cinsinden hata değerleri de sırasıyla 0.01537, 0.02815 ve 0.01001 olarak hesaplanmıştır. Mayıs 2021 dönemi içinde aynı hisselerle ait MRE cinsinden test hataları sırasıyla 0.01748, 0.02646 ve 0.00823 şeklindedir.

Ayrıca Tablo 4.2'de Nisan 2021 dönemi için hesaplanan MRE, MAE ve MSE cinsinden ortalama hata değerleri de 0.01498, 0.31311, 0.52788 ve Mayıs 2021 dönemi için hesaplanan MRE, MAE ve MSE cinsinden ortalama hata değerleri de 0.0135, 0.28816, 0.442 şeklinde verilmiştir. Ortalama MRE, MAE ve MSE değerleri her bir hisseye ait ilgili hata metriklerinin ortalamaları alınarak hesaplanmıştır.

5.2 Öneriler ve Gelecek Çalışmalar

Bölüm 4.5'te paylaşılmış hata grafikleri incelendiğinde, test dönemlerine ilişkin hata değerlerinin de kendi aralarında ciddi farklılıklar gösterdiği görülmüştür. Model başarısının en çok düştüğü tarihlerde, borsa endeksinde sıra dışı hareketlerin olduğu görülmektedir. Şekil 4.24'te 2021-05-20 ile 2021-05-24 tarihleri arasında görülen sıra dışı yükseliş buna örnek olarak gösterilebilir. Bu noktadaki tahmin başarısını

geliřtirmek adına önemli basın yayın organlarında veya sosyal medyada yer alan ekonomi haberleri, duygu analizi metotları kullanılarak analiz edilebilir ve önerilen modele girdi olarak dahil edilebilir. Bu durumda sosyal etkilerden kaynaklanan radikal deęişiklikler öngörülebilir ve model başarısına olumlu katkıda bulunulabilir.



KAYNAKÇA

- [1] Alpaydın E., 2020. Introduction to Machine Learning, **Adaptive Computation and Machine Learning Series** Massachusetts Institute of Technology, 324-340.
- [2] Moghar A., 2020. Stock Market Prediction Using LSTM Recurrent Neural, **Network International Workshop on Statistical Methods and Artificial Intelligence, 6-9**, 5-6.
- [3] Raşo H., Demirci M., 2019. Predicting the Turkish Stock Market BIST 30 Index using Deep Learning, **Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi, 11**:7-8.
- [4] Konstantinou R., 2017. Stock Market prediction using Artificial Neural Networks, Department of Computer Science and Engineering Chalmers University of Technology, MSc Thesis, Göteborg/İsveç, 90 s.
- [5] İnce T., 2019. Predicting the Bitcoin Trend Using Technical Indicators for Deep Learning Algorithmic Features, Department of Computer Science and Engineering Boğaziçi University, MSc Thesis , İstanbul/Türkiye , 250 s.
- [6] Linzie A., 2017. Financial Analysis with Artificial Neural Networks Short-term Stock Market Forecasting, Gardner-Webb University, Undergraduate Honors Theses, Honors Program, North Carolina/Amerika Birleşik Devletleri, 47-55.
- [7] Falinouss P., 2007. Stock trend prediction using news articles: a text mining approach. Lulea University of Technology, Master Thesis, Lulea/İsveç, 85 s.
- [8] Gunduz, H., Cataltepe, Z., Yaslan, Y., 2017. Stock market direction prediction using deep neural networks, pp. 1-4. *Signal Processing and Communications Applications Conference (SIU)*, 25 May, 2017, IEEE.
- [9] Site A., 2020. Stock Market Prediction Using Machine Learning Models. Dokuz Eylül Üniversitesi. Yüksek Lisans Tezi. İzmir/Türkiye, 68 s.
- [10] Fama, E. F., 1970. Efficient capital markets: a review of theory and empirical work. *The Journal of Finance*, **25**: 383-417.

- [11] Irmak S., Çetin K., 2009. Hisse senetlerinin korelasyon uzaklıklarına dayalı olarak kümelenmesi. Süleyman Demirel Üniversitesi, İktisadi ve İdari Bilimler Fakültesi Dergisi., 1: 395-406.
- [12] Wu S., 2013. A review on coarse warranty data and analysis. **Reliability Engineering and System**, 114: 1-11
- [13] Rahm E., Hong H., 2000. Data Cleaning: Problems and Current Approaches, University of Leipzig, Almanya, 214 s.
- [14] Ioffe S., Szegedy C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, Cornell University, arXiv:1502.03167.
- [15] Morris K, 1972. Mathematical thought from ancient to modern times, Vol. 3. Oxford University Press, 1122-1127, ISBN 0195061373.
- [16] Çakır Ö., 2019. Derin Öğrenme Nedir? (Web sayfası: <https://www.yapayzekatr.com/2019/12/16/derinogrenme-nedir/>), (Erişim tarihi: 25 Mayıs 2021).
- [17] Jeffcok P., 2018. What's the Difference Between AI, Machine Learning, and Deep Learning? (Web sayfası: <https://blogs.oracle.com/bigdata/post/whatx27s-the-difference-between-aimachine-learning-and-deep-learning>), (Erişim Tarihi: 30 Mayıs 2021).
- [18] Valueva V., Nagornov N., Lyakhov A., Valuev V., Chervyakov I., 2020. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*. Elsevier Convolutional neural networks are a promising tool for solving the problem of pattern recognition. BV. 177: 232–243. doi: 10.1016/j.matcom.2020.04.031. ISSN 0378- 4754.
- [19] Van den Oord, Aaron; Dieleman, Sander; Schrauwen, Benjami,. Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; Weinberger, K. Q. 2013. Deep content-based music recommendation . Curran Associates, Inc, 2643–2651.
- [20] Collobert, Ronan; Weston, Jason 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. Proceedings of the 25th

International Conference on Machine Learning. pp. 160–167. ICML '08. New York, NY, USA: 69 ACM. Doi:10.1145/1390156.1390177. ISBN 978-1-60558-205-4. S2CID 2617020.

[21] Oleksii A., Sebastien R., Anton P., 2020. Deep Learning Techniques to Improve Intraoperative Awareness Detection from Electroencephalographic Signals. 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). Montreal, QC, Canada: IEEE:142–145. doi:10.1109/EMBC44109.2020.9176228. ISBN 978-1-7281-1990-8. PMID 33017950. S2CID 221386616.

[22] Avraam T., Nikolaos P., 2017. Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks. IEEE 19th Conference on Business Informatics (CBI). Thessaloniki, Greece: 7–12. doi:10.1109/CBI.2017.23. ISBN 978-1-5386-3035-8. S2CID 4950757.

[23] Saha S., 2018. A Comprehensive Guide to Convolutional Neural Networks, (Web sayfası: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>), (Erişim Tarihi: 30 Mayıs 2021).

[24] Samuel D., 2019. A thorough review on the current advance of neural network structures. **Annual Reviews in Control**, **14**: 200–230.

[25] Isaac A., 2018. State-of-the-art in artificial neural network applications: A survey. **Heliyon**. **4** (11): e00938. doi: 10.1016/j.heliyon.2018.e00938. ISSN 2405-8440. PMC 6260436. PMID 30519653.

[26] Ahmed T., 2018. Time series forecasting using artificial neural networks methodologies: A systematic review. **Future Computing and Informatics Journal**. **3** (2): 334–340. doi: 10.1016/j.fcij.2018.10.003. ISSN 2314-7288.

[27] Alex G., Marcus L., Santiago F., 2009. A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. **31** (5): 855–868. CiteSeerX 10.1.1.139.4502. Doi:10.1109/tpami.2008.137. PMID 19299860. S2CID 14635907.

- [28] Haşim S., 2014. Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling, 70 s.
- [29] Xiangang L., Xihong W., 2014. Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition. ArXiv:1410.4281
- [30] Hochreiter, S.; Bengio, Y.; Frasconi, P.; Schmidhuber, J., 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. **IEEE Press**. ISBN 0-7803-5369-2.
- [31] Olah, C., 2015. Understanding LSTM Networks. (Web Sayfası: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>), (Erişim tarihi: 25 Mayıs 2021).
- [32] Cho K., Merriënboer B., Schwenk H., Bougares F., Bengio Y., Bahdanau D., Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. ArXiv:1406.1078v3 [cs.CL]
- [33] <https://keras.io/about/>, (Erişim tarihi: 25 Mayıs 2021)
- [34] <https://numpy.org/about/>, (Erişim tarihi: 25 Mayıs 2021)
- [35] <https://pandas.pydata.org/about/>, (Erişim tarihi: 25 Mayıs 2021)
- [36] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15 1929-1958.
- [37] Claeskens, G.; Hjort, N.L., 2008. Model Selection and Model Averaging, Cambridge University Press.
- [38] Brownlee J., 2019. A Gentle Introduction to Early Stopping to Avoid Overtraining Neural Networks, (Web sayfası: <https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/>), (Erişim Tarihi: 06 Haziran 2021).
- [39] Lemaréchal C., 2012. Cauchy and the Gradient Method", Doc Math Extra: 251–254.

[40] Gandhi R., 2018. A Look at Gradient Descent and RMSprop Optimizers,(Web sayfası: <https://towardsdatascience.com/a-look-at-gradient-descent-and-rmsprop-optimizersf77d483ef08b>), (Erişim Tarihi: 06 Haziran 2021).

[41] Gibbon C.,1990. Performance indicators, BERA Dialogues (2), ISBN 978-1-85359-092-4.

[42] Kızrak A., 2019. Step-by-Step Use of Google Colab's Free TPU, (Web sayfası: <https://blogs.oracle.com/bigdata/post/whatx27s-the-difference-between-ai-machinelearning-and-deep-learning>), (Erişim Tarihi: 06 Haziran 2021).



ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Mehmet OZAN

Uyruğu: Türkiye (T.C)

Doğum Tarihi ve Yeri:

e-mail:

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi, Elektrik Elektronik Müh.	-
Lisans	Erciyes Üniversitesi, Elektrik Elektronik Müh.	2017
Lise	Nuh Mehmet Baldöktü A.L., KAYSERİ	2013

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2021-Halen	Türkiye İş Bankası	Veri Mühendisi
2017-2021	BTC Bilişim Hizmetleri A.Ş.	BT Uzmanı

YABANCI DİL

İngilizce