

T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ
BİLİM DALI

BORSA ANALİZİ VE TAHMİNİ İÇİN DERİN
ÖĞRENME AĞLARI

YÜKSEK LİSANS TEZİ

Seyda KALYONCU

İstanbul
Temmuz, 2020

T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR BİLİMLERİ ve MÜHENDİSLİĞİ
YÜKSEK LİSANS PROGRAMI

BORSA ANALİZİ VE TAHMİNİ İÇİN DERİN
ÖĞRENME AĞLARI

YÜKSEK LİSANS TEZİ

Seyda KALYONCU

Tez Danışmanı
Dr. Öğr. Üyesi Akhtar JAMIL

İstanbul
Temmuz, 2020

TEZ ONAYI

Lisansüstü Eğitim Enstitüsü Müdürlüğüne,

Bu çalışma, jürimiz tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Bilimleri ve Mühendisliği Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman

Dr. Öğr. Üyesi Akhtar JAMIL

Üye

Dr. Öğr. Üyesi Ali HAMİTOĞLU

Üye

Dr. Öğr. Üyesi Muhammad ILYAS

Onay

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.

Prof. Dr. Ali GÜNEŞ

Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİMİ

Yüksek lisans tezi olarak hazırladığım “**Borsa Analizi ve Tahmini için Derin Öğrenme Ağları**” adlı çalışmanın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

Seyda Kalyoncu

ÖNSÖZ

Araştırmamdaki her aşamada bana yardımcı olan değerli tez danışmanım Dr. Öğr. Üyesi Akhtar JAMIL'e, eğitim alanında dersleriyle bize vizyon katan çok değerli hocamız Dr. Öğr. Üyesi Yahya Şirin'e, lisans ve yüksek lisans eğitimim boyunca benden desteklerini esirgemeyen arkadaşlarıma, Murat Demirci'ye ve aileme teşekkürlerimi sunarım.

Seyda KALYONCU

İstanbul - 2020

ÖZET
BORSA ANALİZİ ve TAHMİNİ İÇİN DERİN ÖĞRENME
AĞLARI

Seyda Kalyoncu

Yüksek Lisans, Bilgisayar Bilimleri ve Mühendisliği

Tez danışmanı: Dr. Öğr. Üyesi Akhtar Jamil

Temmuz-2020, 61 Sayfa

Borsa, bir ülkenin ekonomik koşullarının kilit bir göstergesidir. Borsa, broker ve şirketlerin yatırım yapması için tarafsız bir zemin sağlar. Yüksek yatırım getirisi nedeniyle, insanlar geleneksel bankalardan ziyade borsalara yatırım yapma eğilimindedir. Ancak, döviz kurlarındaki yüksek dalgalanmalar nedeniyle borsalara yatırım yapma riski yüksektir. Bu nedenle, oldukça sağlam bir borsa tahmin sistemi geliştirmek, yatırımcıların yatırım hakkında daha iyi bir karar vermelerine yardımcı olabilir. Borsa tahmini, değişken piyasa durumlarıyla başa çıkmak için en ilginç araştırma alanı haline gelmiştir. Bu alanda bir dizi araştırma çalışması yayınlanmış ve farklı yöntemler önerilmiştir. Yapay zekâ (AI) alanındaki makine öğrenimi tekniklerinin başarısını takiben, borsa tahmini alanındaki verimliliklerini de gösterdiler. Bu tez çalışmasında, borsa fiyat tahmini için en popüler üç makine öğrenme algoritması ve gelecekteki piyasa değerini tahmin etmek için hisse senedi geçmiş verilerine derin öğrenme temelli bir yaklaşım uygulanmıştır. İncelenen yöntemler şunlardır: Yapay Sinir Ağı (YSA), k-En Yakın Komşular (KNN), Oto-Regresif Entegre Hareketli Ortalamalar (Auto ARIMA) ve Uzun Kısa Süreli Bellek (LSTM). Bu çalışmanın amacı, geçmiş hisse senedi fiyat verilerine bakarak gelecekte bir tahmin yapmaktır.

Anahtar Kelimeler: Borsa tahmini, makine öğrenmesi, LSTM, derin öğrenme, yapay sinir ağı, k-En Yakın Komşular, Auto ARIMA

ABSTRACT
DEEP LEARNING NETWORKS for STOCK MARKET
ANALYSIS

Seyda Kalyoncu

Master of Science, Computer Science and Engineering

Supervisor: Assist. Prof. Akhtar Jamil

July-2020, 61 Pages

The stock market is a key indicator of the economic conditions of a country. Stock exchange provides a neutral ground for brokers and companies to invest. Due to high investment return, people tend to invest in stock markets rather than traditional banks. However, there is high risk in investment in stock markets due to high fluctuations in exchange rates. Therefore, developing a highly robust stock prediction system can help investors to make a better decision about investment. A number of research works have been published in this area and different methods have been proposed. Following the success of machine learning techniques in the field of artificial intelligence (AI), they have also shown their efficiency in the field of stock market prediction. In this thesis, the three most popular machine learning algorithms for stock market price prediction and a deep learning based approach to stock historical data have been applied to predict future market value. The methods investigated include: Artificial Neural Network (ANN), k-Nearest Neighbors (KNN), Auto-Regressive Integrated Moving Averages (Auto ARIMA) ve Long Short Term Memory (LSTM). The aim of this study is to make a future forecast by looking at the historical stock price data.

Keywords: Stock market prediction, machine learning, LSTM, deep learning, machine learning, artificial neural network, k-Nearest Neighbors, Auto ARIMA

İÇİNDEKİLER

DIŞ KAPAK	
İÇ KAPAK	
TEZ ONAYI	i
BİLİMSEL ETİK BİLDİRİMİ	ii
ÖNSÖZ	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER	vi
TABLolar LİSTESİ	viii
ŞEKİLLER LİSTESİ	ix
KISALTMALAR LİSTESİ	xi
BİRİNCİ BÖLÜM	1
GİRİŞ 1	
1.1. Problem	1
1.2 Araştırma ve Yöntem	2
1.3 Amaç	3
İKİNCİ BÖLÜM	4
LİTERATÜR TARAMASI	4
ÜÇÜNCÜ BÖLÜM	8
VERİ SETİ ve ALGORİTMALAR	8
3.1 Veri Seti	8
3.1.1 Veri Önışleme	9
3.2 Algoritmalar	13
3.2.1 Yapay Sinir Ağları	13
3.2.2 K En Yakın Komşular.....	16
3.2.3 Oto-Regresif Entegre Hareketli Ortalamalar	17

3.2.4 Uzun Kısa Süreli Bellek.....	18
3.3 Aktivasyon Fonksiyonu	21
3.3.1 Sigmoid Fonksiyonu	22
3.3.2 Tanh Fonksiyonu.....	23
3.3.3 ReLu Fonksiyonu.....	23
DÖRDÜNCÜ BÖLÜM	25
YÖNTEM.....	25
4.1 Makine Öğrenme Yöntemleri	25
4.1.1 Yapay Sinir Ağları Modeli.....	25
4.1.2 K-En Yakın Komşu Modeli	28
4.1.3 Oto regresif Entegre Hareketli Ortalamalar Modeli	30
4.2 Derin Öğrenme Yöntemi.....	32
4.2.1 Uzun-Kısa Süreli Bellek	32
BEŞİNCİ BÖLÜM	35
DENEYSEL SONUÇLAR.....	35
5.1 Hata Oranı Metrik Hesaplamaları	35
5.2 Makine Öğrenme Algoritmalarının Sonuçları	36
5.2.1 Yapay Sinir Ağları Modeli Sonuçları	36
5.2.2 K-En Yakın Komşu Modeli Sonuçları.....	36
5.2.3 Oto regresif Entegre Hareketli Ortalamalar Modeli Sonuçları	37
5.3 Derin Öğrenme Yöntemi Sonuçları	39
5.3.1 Uzun-Kısa Süreli Bellek Modeli Sonuçları	39
ALTINCI BÖLÜM	42
TARTIŞMA VE DEĞERLENDİRME	42
KAYNAKÇA	44
ÖZGEÇMİŞ.....	48

TABLÖLAR LİSTESİ

Tablo 1. THYAO Hissesine Ait Veri Seti	8
Tablo 2. 5 Dakikalık THYAO Örnek Veri Seti.....	9
Tablo 3. Makine Öğrenimi Algoritmaları Doğruluk Oranları	38
Tablo 4. Veri Setinden Elde Edilen Doğruluk Oranlarının Özeti.....	39



ŞEKİLLER LİSTESİ

Şekil 1. TATAGLOBAL Hisse Senedi Veri Seti	9
Şekil 2. 5 Dakikalık Periyotlara Döndürme Fonksiyonu.....	9
Şekil 3. İndikatör Ekleme	10
Şekil 4. Veri Ön işlemeden sonra Borsa İstanbul Veri Seti.....	11
Şekil 5. AKBNK Hisse Senedi Kapanış Fiyatı.....	11
Şekil 6. ARCLK Hisse Senedi Kapanış Fiyatı	11
Şekil 7. ASELS Hisse Senedi Kapanış Fiyatı.....	12
Şekil 8. GARAN Hisse Senedi Kapanış Fiyatı.....	12
Şekil 9. THYAO Hisse Senedi Kapanış Fiyatı.....	12
Şekil 10. MLP Mimari Yapısı	14
Şekil 11. Bilgisayar Nöronu	14
Şekil 12. Örnek Yapay Sinir Ağı.....	15
Şekil 13. Tekrarlayan Sinir Ağı (RNN) Şeması	18
Şekil 14. Uzun-Kısa Sürekli Bellek (LSTM) Mimarisi.....	19
Şekil 15. Aktivasyon Fonksiyonu Şeması	21
Şekil 16. Sigmoid Fonksiyonu.....	22
Şekil 17. Tanh Fonksiyonu	23
Şekil 18. ReLu Fonksiyonu	24
Şekil 19. Eğitim Veri Seti Ön İşleme	26
Şekil 20. Test Veri Seti Ön İşleme	26
Şekil 21. Yapay Sinir Ağı için Gerekli Kütüphaneler	27
Şekil 22. YSA Modeli İkinci Katman.....	27
Şekil 23. YSA Modeli Son Katman.....	28
Şekil 24. YSA Modeli Optimizer ve Kayıp.....	28
Şekil 25. YSA Modeli Epoch ve Batch_size Sayısı	28
Şekil 26. k-En Yakın Komşu Modeli Kütüphaneler.....	29
Şekil 27. Veri Seti.....	29
Şekil 28. K-En Yakın Komşu Modeli Test-Eğitim Seti İşlemi	29
Şekil 29. K-En Yakın Komşu Modeli.....	30
Şekil 30. ARIMA Modeli için Gerekli Kütüphaneler	30

Şekil 31. TATAGLOBAL Oto korelasyon Grafiği	31
Şekil 32. TATAGLOBAL Kapanış Fiyatı.....	31
Şekil 33. LSTM Modeli Veri Seti Ayırma	32
Şekil 34. Zaman Aşımı Algoritması	33
Şekil 35. LSTM Modeli Katmanlar	33
Şekil 36. LSTM Modeli.....	34
Şekil 37. TATAGLOBAL YSA Model Sonuçları	36
Şekil 38. TATAGLOBAL K-NN Modeli Sonuçları	37
Şekil 39. TATAGLOBAL Auto ARIMA Model Sonuçları	37
Şekil 40. TATAGLOBAL Auto ARIMA Model Sonuçları	38
Şekil 41. LSTM kullanarak AKBNK Hisse Senedi Tahmini.....	40
Şekil 42. LSTM kullanarak ASELS Hisse Senedi Tahmini	40
Şekil 43. LSTM kullanarak GARAN Hisse Senedi Tahmini	41
Şekil 44. LSTM kullanarak THYAO Hisse Senedi Tahmini	41
Şekil 45. LSTM kullanarak ARCLK Hisse Senedi Tahmini.....	41

KISALTMALAR LİSTESİ

LSTM	: Uzun Kısa Süreli Bellek
ARIMA	: Oto-Regresif Entegre Hareketli Ortalamalar
K-NN	: k-En Yakın Komşular
YSA	: Yapay Sinir Ağı
RNN	: Tekrarlayan Sinir Ağı (Recurrent Neural Network)
RELU	: Düzleştirilmiş Doğrusal Birim Katmanı (Rectified Linear Units Layer)
ML	: Makine Öğrenmesi (Machine Learning)
DL	: Derin Öğrenme (Deep Learning)
MLP	: Çok Katmanlı Algılayıcı (Multi Layer Perceptron)
WMA	: Ağırlıklı Hareketli Ortalamalar (Weighted Moving Average)
MA	: Hareketli Ortalamalar

BİRİNCİ BÖLÜM

GİRİŞ

1.1. Problem

Borsa tahmini deęişken piyasa durumlarıyla başa çıkmak için en ilginç araştırma alanı haline gelmiştir. Dünya çapında tahmin edilmesi zor bir alan olmuştur. Borsa, halka açık şirketlerin alım, satım ve ihraç faaliyetlerinin düzenli olarak gerçekleştirildięi pazar anlamına gelir. Borsaya sahip her ülke bünyesinde yatırımcıları bir araya getirdięi, alım-satım işlemlerinin güvenle ve belli kurallar dahilinde yapılmasını sağladığı bir kurum bulundurur. Türkiye’de faaliyet gösteren bu kurum Borsa İstanbul’dur. Borsa İstanbul Türkiye’de faaliyet gösteren tek kurumdur. 1986 yılında İMKB (İstanbul Menkul Kıymetler Borsası) adı ile faaliyet göstermeye başlayan kurum 2013 yılında ismini Borsa İstanbul (BİST) olarak deęiştirmiştir. Borsada işlem yapmak isteyen bir kişi öncelikle Borsa İstanbul’un onay verdięi aracı kurumlardan bir yatırım hesabı açması gerekir. Yapılan alım-satım işlemleri bu sektörde hizmet veren aracı kurumlar sayesinde gerçekleşir. Alım satım işlemleri arz-talep ilişkisine dayanmaktadır. Kıymetli bir hisse senedi almak isteyen bir alıcı bu işlemi yapmak için karşılığında o hisse senedini satan bir satıcı olmalıdır. Yatırımcı henüz deęer kazanmamış bir hisse senedi olarak, deęer gördüğü zaman o hisse senedini sattığı zaman kar elde eder. Bu borsada görülen standart bir işlemdir. Hisse senedi sermaye şirketlerinin payını temsil eden bir yatırım türüdür. Yatırımcılar zaman içinde deęerleneceklerini düşündükleri hisse senetlerini satın alırlar. Hisse senedi deęeri, yukarıda bahsettiğimiz arz-talep ilişkine göre deęişiklik gösterir.

Yerel ve evrensel olarak borsayı etkileyen birçok faktör vardır. Bu nedenle, ekonomik denge ve fiziksel olaylar gibi borsaya birçok faktör dahil olduğundan, tahmini çok zor bir iştir. Bu zamana kadar yapılan çalışmalarda kesin bir yöntem ve teori yoktur (Liu et al., 2019). Bu nedenle, oldukça sağlam bir hisse tahmin sistemi geliştirmek, yatırımcıların yatırım hakkında daha iyi bir karar vermelerine yardımcı olabilir. Genel olarak, hisse tahmin yöntemleri üç ana kategoriye ayrılır: teknik analiz, temel analiz ve evrimsel analiz. Teknik analiz, geçmiş fiyat hareketlerine göre gelecekteki fiyat hareketlerini tahmin etmek için en sık kullanılan yöntemlerdir. Teknik analiz yöntemleri piyasaya göre deęişiklik göstermektedir. Bu istatistikler doğrudan giriş verilerinden öğrenilebilir ve analiz edilebilir. Teknik analiz, fiyat ve hacim dahil olmak

üzere geçmiş piyasa verilerinin incelenmesidir. Piyasa psikolojisi, davranışsal ekonomi ve niceliksel analizden elde edilen bilgileri kullanarak teknik analistler, gelecekteki piyasa davranışlarını tahmin etmek için geçmiş performansı kullanmayı hedefler. Teknik analizin en yaygın iki şekli grafik desenleri ve teknik (istatistiksel) indikatörlerdir. Teknik analiz göstergeleri indikatör olarak adlandırılmaktadır. İndikatörler, teknisyenlerin fiyat ve hacimlere çeşitli matematiksel formüller uyguladığı istatistiksel bir teknik analiz şeklidir. Teknik göstergeler, teknik analizleri takip eden yatırımcılar tarafından kullanılan bir menkul kıymetin veya sözleşmenin fiyatı, hacmi ve / veya açık faizi tarafından üretilen sezgisel veya örüntüye dayalı sinyallerdir. Tarihsel verileri analiz ederek, teknik analistler gelecekteki fiyat hareketlerini tahmin etmek için göstergeler kullanırlar. En yaygın teknik göstergeler, eğilimleri tespit etmeyi kolaylaştırmaya yardımcı olmak için fiyat verilerini düzgün hale getiren hareketli ortalamalardır. Hareketli Ortalama (MA) teknik analizde yaygın olarak kullanılan bir hisse senedi indikatörüdür. Bir hisse senedinin hareketli ortalamasını hesaplamamanın nedeni, sürekli güncellenen bir ortalama fiyat oluşturarak fiyat verilerinin düzeltilmesine yardımcı olmaktır. Hareketli ortalamalar (MA), sürekli güncellenen bir ortalama fiyat oluşturarak fiyat verilerini düzelten basit bir teknik analiz aracıdır. Ortalama, 10 gün, 20 dakika, 30 hafta veya tüccarın seçtiği herhangi bir zaman dilimi gibi belirli bir süre boyunca alınır. Hareketli ortalama stratejileri popülerdir ve hem uzun vadeli yatırımcılara hem de kısa vadeli yatırımcılara uygun herhangi bir zaman çerçevesine uyarlanabilir. Hareketli ortalamalar genellikle bir hisse senedinin eğilim yönünü veya destek ve direnç seviyelerini belirlemek için hesaplanır. Geçmiş fiyatlara dayandığı için trend takip eden veya gecikmeli bir göstergedir. Hareketli ortalama süresi ne kadar uzun olursa, gecikme o kadar büyük olur. Dolayısıyla, 200 günlük hareketli ortalama, son 200 günlük fiyatları içerdiği için 20 günlük MA'dan çok daha fazla gecikmeye sahip olacaktır. Hisse senetleri için 50 günlük ve 200 günlük hareketli ortalama rakamları büyük ölçüde yatırımcılar ve borsa simsarları izlemektedir ve önemli ticaret sinyalleri olarak kabul edilmektedir.

1.2 Araştırma ve Yöntem

Görüldüğü üzere teknik analiz istatistiksel analize dayanmaktadır (Liu et al., 2019). Ancak, hisselerin ait olduğu şirketin durumu, siyasi düzen ve evrensel olaylar gibi birçok dış faktör borsa trendini etkilemektedir. Bu nedenle, borsa zaman serileri

genellikle doğrusal olmayan ve dinamik bir süreçtir. Aynı zamanda, borsadaki dalgalanma oldukça şiddetlidir. Buna göre, veriler çok gürültülüdür. Piyasa tahmini yaparken riski düşük tutmak çok önemlidir. Bununla birlikte, güçlü bilgi işlem teknolojisinin ve son teknoloji makine öğrenme tekniklerinin geliştirilmesi, borsa değeri tahminlerinin yüksek doğrulukla yapılmasını mümkün kılmıştır. Son çalışmaların çoğu yapay zekâ uygulamaları ve makine öğrenme teknikleri ile yapılmaktadır. Bu tekniklerin başarı oranları diğer istatistiksel modellerden çok daha yüksektir. Yapılan araştırmalar makine öğrenmesi algoritmaları ve derin öğrenme üzerine olmuştur. Bu tez çalışmasında 3 önemli makine algoritması olan Yapay Sinir Ağları, Oto-Regresif Entegre Hareketli Ortalamalar (ARIMA), k-En Yakın Komşu algoritması ve derin öğrenme ağı olan Uzun-Kısa Süreli Bellek (LSTM) yöntemi uygulanmıştır.

1.3 Amaç

Bu çalışmanın amacı büyük ilgi alanı haline gelen borsa alanında yatırımcılara ışık tutacak tahmin çalışmaları yapmaktır. Amaç geçmiş hisse senedi fiyat verilerine bakarak gelecekteki bir tahmin yapmaktır. Derin öğrenme ve makine öğrenme algoritmaları sayesinde riski düşük tutmak en büyük amaçlardan biridir. Yapay zekâ alanına olan ilgiyle beraber makine öğrenme algoritmaları ve derin öğrenme temelli modellerin bu alan için uygunluğunun ispatlanması hedeflenmektedir. Tezin bundan sonraki kısımları ilk olarak daha önce yapılan çalışmaların anlatıldığı literatür taraması bölümü bulunmaktadır. Ardından kullanılan algoritmalar ve işlem yapılacak veri setleri hakkında bir bölüm bulunur. Daha sonra uygulanan algoritma modellerinin anlatıldığı yöntem bölümü, oluşturulan modellerin deneysel sonuçları ve son olarak değerlendirme ve tartışma bölümü olmak üzere toplamda 6 bölüm bulunmaktadır.

İKİNCİ BÖLÜM

LİTERATÜR TARAMASI

Borsa, bir ülkenin ekonomik koşullarının kilit bir göstergesidir. Borsa, broker ve şirketlerin yatırım yapması için tarafsız bir zemin sağlar. Yüksek yatırım getirisi nedeniyle, insanlar geleneksel bankalardan ziyade borsalara yatırım yapma eğilimindedir. Ancak, döviz kurlarındaki yüksek dalgalanmalar nedeniyle borsalara yatırım yapma riski yüksektir. Bu nedenle, oldukça sağlam bir borsa tahmin sistemi geliştirmek, yatırımcıların yatırım hakkında daha iyi bir karar vermelerine yardımcı olabilir. Borsaya olan ilgi ve gelişen teknoloji sayesinde bu alanda yapılan çalışmalara yönelim oldukça yüksektir. Makine öğrenme teknikleri, yapay zekâ uygulamaları borsa tahmini alanının önünü açmıştır. Başarı oranları birçok istatistiksel modellerden çok daha yüksektir. Son yıllarda, yapay sinir ağı bu alanda sıcak bir araştırma alanı haline gelmiştir. Bu alanda kullanılan birçok makine öğrenme algoritması ve derin öğrenme tekniği bulunmaktadır.

Yapay sinir ağı (YSA) doğrusal olmayan ilişkiye güçlü doğrusal olmayan yaklaşım yeteneğine sahiptir. YSA, karmaşık verilerdeki doğrusal olmayan ilişkiyi öğrenmek için başarıyla uygulanmıştır. Hisse senedi fiyatı, hacmi ve diğer veriler hisse senedi fiyatı değişikliklerini etkileyen çok sayıda bilgi içerdiğinden, YSA hisse senedi fiyatlarının etkilerini bulmak için hisse senedinin geçmiş verilerini öğrenebilir. Bilgisayarlar ve işletmeler gibi farklı alanlardaki araştırmacılar da hisse senedi piyasası tahmini üzerine birçok araştırmada bulmuşlardır (Sadia et al., 2019). Fatima ve arkadaşları (Fatima & Hussain, 2008) çalışmalarında ANN ve ARIMA kullanarak yaklaşık 3 yıllık verilere sahip KSE-100 Endeksini tahmin etmişlerdir. Bir başka çalışmada Asya pazarlarını kullanarak piyasaya somut etkiler bulmak için araştırmalar yapılmıştır (Aurangzeb, 2012). Chirag Modi ve arkadaşları Ulusal Menkul Kıymetler Borsası (NSE) kapsamındaki şirketlerin YSA hisse senetlerini kullanarak bir tahmin çalışması yapmıştır. Yapay sinir ağlarının hisse senedi fiyat tahmininde başarılı olduklarını kanıtlamışlardır (Patel & Yalamalle, 2014). Y. Bing, J. Hao ve S. Zhang, borsa endekslerini tahmin etmek için YSA kullandılar (Bing et al., 2012). Son n gün içinde hisse değerlerine dayalı bir tahmin geliştirdiler. Bu çalışma sonucunda %96 doğruluk oranı elde ettiler. Wanjawa ve arkadaşları hata geri yayılımı ile ileri beslemeli çok katmanlı algılayıcı olan Yapay Sinir Ağının kullanarak New York

Menkul Kıymetler Borsası hisselerini tahmin edebilir sonuçlar elde ettiler (Wanjawa & Muchemi, 2014). Malav Shastri ve arkadaşları ilk olarak hisse senedi fiyatlarını tahmin etmek için önce Naive Bayes sınıflandırıcısı aracılığıyla duygu puanlarını hesaplayarak bir teknik önermiş ve bundan sonra hem duygu puanlarına hem de tarihsel stok veri kümesine sinir ağı uygulamıştır. Duyarlılık analizinden gelen girdiler ile nöral bir ağ modeli ve fiyatları tahmin etmek için geçmiş veriler kullanılıyor. Deneylemlerden, doğruluk seviyesinin maksimum durumlarda %90'ın üzerine çıktığı ve ayrıca son verilerle eğitildiğinde modelin daha sağlam bir temel olacağı gözlemlenmiştir (Shastri et al., 2018). Moghaddam ve arkadaşları yapay sinir ağının (YSA) günlük NASDAQ borsa oranını tahmin etme yeteneğini araştırmışlardır. Geri yayılma algoritması tarafından eğitilen birkaç ileri beslemeli YSA değerlendirilmiştir. Bu çalışmada kullanılan metodoloji, kısa vadeli tarihsel hisse senedi fiyatlarını ve haftanın gününü girdi olarak değerlendirmiştir. Model çıktıları ile gerçek veriler arasında belirgin bir fark olmadığı ve yüksek doğruluk oranı sonucuna varmışlardır (Moghaddam et al., 2016). Mehak Usmani ve arkadaşları Karachi Menkul Kıymetler Borsası'nın (KSE) gün kapanışında farklı makine öğrenme teknikleri kullanarak piyasa performansını tahmin etmişlerdir. Tek Katmanlı Algılayıcı (SLP), Çok Katmanlı Algılayıcı (MLP), Radyal Temelli Fonksiyon (RBF) ve Destek Vektör Makinesi (SVM) dahil makine öğrenme teknikleri karşılaştırılmıştır. YSA'nın temeli olan MLP algoritması diğer tekniklerle karşılaştırıldığında en iyi performansı göstermiştir (Sadia et al., 2019).

Diğer araştırılan yöntemlerden biri de k-En Yakın Komşu Algoritmasıdır. Kısaca k-NN olarak adlandırdığımız algoritma en temel makine algoritmalarından birisidir. Sınıflandırma algoritması olarak da bilinmektedir. Finans alanında doğrusal olmayan süreci modelleme de kullanılmaktadır. Khalid Alkhatib ve arkadaşları Ürdün pazarındaki altı şirketin hisse senedi fiyatlarını tahmin etmek için k-NN algoritmasını kullandılar (Alkhatib et al., 2013). Sonuçlar k-NN algoritmasının çok küçük hata paylarına ve gerçek verilere çok yakın değerlere sahip olduğu sonucuna varılmıştır. Başka bir k-NN çalışmada, Endonezya pazarındaki LQ45 endeksinin fiyat tahmini yapılmıştır (Tanuwijaya & Hansun, 2019). Çalışmada, k-NN, hareketli ortalama yönteminden daha yüksek bir doğruluk oranı elde edilmiştir. Patil ve diğ. K-NN algoritması ve geri yayılım öğrenme tekniği kullanarak hisse senedi fiyatlarını tahmin etmiştir (Sonawane, 2015). K-NN algoritmasının fiyat tahmininde daha gerçekçi

sonular verdiđini grdler. Rodrigues ve arkadařları k-NN algoritması kullanarak dokuz farklı dvizin gelecekteki hareketlerini tahmin eden bir model geliřtirmişlerdir (Fernández-Rodríguez et al., 1999). 1978-1994 yılları arasındaki gnlk verilerle alıřma yapmışlardır. Dođrusal olmayan veriler zerinde yaptıkları alıřmada başarılı sonular elde etmişlerdir. Aparicio ve arkadařları ABD Menkul Kıymetler Piyasası'ndan alınan 5 hisse senedine k-NN algoritması uygulamışlardır (Aparicio et al., 2002). Yapılan alıřma sonucunda elde edilen tahminlerin kalitesi lmlmř ve tatmin edici sonular ile karřılařmışlardır. Kenan İlarıslan k-NN algoritması kullanarak hisse senedi fiyatlarının bir gn sonraki fiyatını tahmin etmeye alıřmıştır. BIST'de iřlem gren Afyon imento hisse senedinin 2014 yılı kapanıř fiyatlarına k-NN uygulayarak 2015 yılının ilk 3 ayının fiyatlarını tahmin alıřması yapmıştır. alıřmalarının sonucunda %97 oranında bir başarı elde etmiştir (İLARSLAN, 2016) .

Auto ARIMA, zaman serisi tahmini iin ok popler bir istatistiksel yntemdir. Gemiř veriler gelecekteki deđerleri tahmin etmek iin kullanılır. Auto ARIMA birok alıřmada kullanılmıştır ve geliřtirilmiştir. S. Wadi, M. Almasarweh ve A. Alsarairah, 2010'dan 2018'e Amman Menkul Kıymetler Borsası'ndan (ASE) elde edilen kapanıř fiyatlarına otomatik ARIMA modelini uyguladı (Wadi et al., 2018). Sonu olarak, modelin nemli sonular verdiđini grdler. Bir bařka alıřmada otoregresif entegre hareketli ortalama (ARIMA) modelinin avantajları gsterilmiştir (Almasarweh & Wadi, 2018). alıřmalarında, 1993'ten 2017'ye kadar gnlk verileri olan rdn piyasasında Amman borsası verileri kullanıldı. Sonu olarak, ARIMA modelinin kısa vadeli piyasa tahminlerinde nemli sonular verdiđi gsterilmiştir. Adebıyi ve arkadařları New York Menkul Kıymetler Borsası ve Nijerya Menkul Kıymetler borsasından elde edilen hisse senedi verileriyle ARIMA kullanarak hisse senedi tahmin modeli oluřturmuşlardır. Elde edilen sonular, ARIMA modelinin kısa vadeli tahmin iin gl bir potansiyele sahip olduđunu ve hisse senedi fiyat tahmini iin mevcut tekniklerle olumlu rekabet edebileceđini gstermiştir (Adebıyi et al., 2014).

Son zamanlarda, derin đrenme tabanlı yaklařımlar hisse verilerinin sınıflandırılması iin iyi bir dođruluk gstermiştir. zellikle, Tekrarlayan Sinir Ađları (RNN'ler), zaman serisi veri analizi iin yaygın olarak kullanılmaktadır (Roondiwala et al., 2015). rneđin, Xiao ve ark. olaya dayalı borsa tahmini iin derin bir evriřimli sinir ađı kullandı (El-Masry et al., 2002). Bengio ve diđerleri hisse senedi fiyat tahmininde

LSTM modelini kullanmıştır (Baek & Kim, 2018). LSTM, RNN'nin gelişmiş bir versiyonudur. Sıralı verilerle başa çıkma yeteneğine sahiptir ve borsa değer tahmininin eğitimi ve testi için son derece uygundur. Hisse senedi tahminleri için Chen, Zhou ve Dai (Chen et al., 2015) Çin borsa verileri üzerine LSTM tabanlı bir model önermiştir. Piyasa endeksleri ve hisse senetlerinin tarihsel fiyat verilerini kullandılar. Sonuçlar LSTM'nin umut verici olduğunu gösterdi. LSTM, mevcut verileri genişletmek için farklı türdeki sıralı verileri diğer ağlara besleyebilir. Benzer şekilde, Li, Bu ve Wu (Jiahong Li et al., 2017), yatırımcı hisselerini ve CSI300 hisse senedini tahmin etmek için LSTM'i kullanarak tarihi piyasa verileri içeren bir ağ beslemiştir. Böylece, LSTM'in destek vektör makinelerinin karşılaştırma modellerinden daha iyi performans gösterdiği ve fiyat tahmininde daha iyi sonuçlar verdiği sonucuna vardılar. David M. Q. Nelson ve arkadaşları, yakın gelecekte belirli bir hisse senedinin yükselip yükselmeyeceğini LSTM modelini kullanarak bir tahmin çalışması yaptılar (Nelson et al., 2017). Modelin yüksek varyasyonları tahmin etmede daha başarılı olduğunu buldular. Ortalama %55,9'luk bir doğruluk elde ettiler. Wei Bao ve meslektaşları, hisse senedi fiyatı tahmini için LSTM ve otomatik kodlayıcıları (SAE) modellerini birleştirerek üç aşamalı derin öğrenme çerçevesi oluşturduklarını (Bao et al., 2017). Ertesi günün kapanış fiyatı tahmini için LSTM modelini oluşturduklarını. Sonuçlar, önerilen modelin hem öngörme doğruluğu hem de karlılık performansında diğer benzer modellerden daha iyi performans gösterdiğini göstermektedir. Pengfei Yu ve Xuesong Yan hisse senedi fiyatlarını tahmin etmek için LSTM'e dayanan derin bir sinir ağı modeli oluşturduklarını (Yu & Yan, 2020). Önerilen ve diğer bazı tahmin modellerini karşılaştırdıklarında, önerilen tahmin modelinin daha yüksek bir tahmin doğruluğuna sahip olduğunu gösterir. Thomas Fischer ve Christopher Krauss, 1992'den 2015'e kadar S&P 500 stok hareketlerini tahmin etmek için LSTM, rastgele orman (RAF), derin bir sinir ağı (DNN) ve lojistik regresyon sınıflandırıcı (LOG) modellerini kullandılar (Fischer & Krauss, 2017). LSTM sonuçlarını rastgele orman, standart bir derin ağ ve basit bir lojistik regresyon ile karşılaştırdılar. Bu alan için kendiliğinden uygun olan LSTM modelinin, standart bir net ve lojistik regresyonun açık bir farkla üstesinden geldiği sonucuna varmışlardır. Çoğu zaman (küresel kriz hariç), rastgele orman modelinden daha iyi performans gösterdiklerini gördüler.

ÜÇÜNCÜ BÖLÜM

VERİ SETİ ve ALGORİTMALAR

Bu bölümde borsa tahmini için kullanılacak algoritmaların tanımlamaları, veri seti ve veri setine uygulanan işlemler yer alacaktır.

3.1 Veri Seti

Yapılan tez çalışmasında iki farklı veri seti üzerinde çalışmalar yapılmıştır. İlk veri setimiz olan borsada tanınmış beş Türk şirketin hisse senedi değerini tahmin etmek için Uzun Kısa Süreli Bellek (LSTM) kullandık. Bu çalışmada kullanacağımız veri seti, Borsa İstanbul'un BIST 30 listesinde yer alan 5 hisseden oluşuyor. Bu şirketler arasında Borsa İstanbul'un BIST 30 listesindeki Türk Hava Yolları (THYAO), Akbank (AKBNK), Arçelik (ARCLK), Aselsan (ASELS) ve Garanti (GARAN) şirketleri bulunmaktadır. Veriler, 2014-2019 yıllarına ait son 5 yıllık verilerdir. Veri setimizdeki her bir hisse için veriler bir dakikalık verilerden oluşur. Tablo 1, işlem öncesi durumu göstermektedir.

Tablo 1. THYAO Hissesine Ait Veri Seti

Symbol	Signal_Time	Date_Time	Close	High	Low	Open	Total Quantity	Volume	Weighted Average
THYAO	1412145300000	2014-10-01T09:35:00+03:00	6.51	6.52	6.49	6.49	533009	3.464895	6.50063
THYAO	1412145360000	2014-10-01T09:36:00+03:00	6.53	6.53	6.51	6.51	267332	1.742998	6.51998
THYAO	1412145420000	2014-10-01T09:37:00+03:00	6.52	6.53	6.52	6.52	165859	1.082718	6.52794
THYAO	1412145480000	2014-10-01T09:38:00+03:00	6.52	6.53	6.52	6.53	90008	5.869022	6.52056

LSTM modelini uygulamadan önce veri setimizde bazı düzenlemeler yaptık. Veri setinin doğru hazırlanması, doğru verilerin kullanılması ve gürültünün giderilmesi uygulanacak modelin doğruluğunu artırır.

İkinci veri setimiz Hindistan piyasasındaki Tata Global Beverages Limited şirketinden elde edilmiştir. Bu veriler, 2010-2019 yılları arasında hisse senedinin açılış, kapanış, en yüksek ve en düşük değerlerinden oluşmaktadır. Diğer veri setine göre farkı günlük verilerden oluşmaktadır. Bu veri setimizi makine algoritmalarını uygulamak için kullandık. TATAGLOBAL şirketine ait veri seti Şekil 1'de gösterilmiştir.

	Date	Open	High	Low	Last	Close	Total Trade	Quantity	Turnover (Lacs)
0	2010-07-21	122.1	123.00	121.05	121.10	121.55		658666.0	803.56
1	2010-07-22	120.3	122.00	120.25	120.75	120.90		293312.0	355.17
2	2010-07-23	121.8	121.95	120.25	120.35	120.65		281312.0	340.31
3	2010-07-26	120.1	121.00	117.10	117.10	117.60		658440.0	780.01
4	2010-07-27	117.6	119.50	112.00	118.80	118.65		586100.0	694.98

Şekil 1. TATAGLOBAL Hisse Senedi Veri Seti

3.1.1 Veri Önışleme

Veri önışleme işlemlerini sadece Borsa İstanbul'un BIST 30 listesindeki 5 hisse senedine ait olan veri setine uyguladık. Veri kümesinde, ilk olarak verilerdeki tarih sütununu bir zaman dizisi olduđu için dizine ekledik.

```
def bes_dk(x):
    if x.minute%10==0 or x.minute%10==5:
        return x
    else:
        pass

data['date_time']=data['date_time'].apply(bes_dk)
```

Şekil 2. 5 Dakikalık Periyotlara Döndürme Fonksiyonu

Veri kümesi bir dakikalık çubuklardan oluşuyordu. Bir dakikalık veriler çok gürültülü bir veri olacağından GMT değerini kaldırdık ve beş dakikalık periyotlara böldük (Tablo 2).

Tablo 2. 5 Dakikalık THYAO Örnek Veri Seti

Date_Time	Symbol
2014-10-01 09:35:00	THYAO
2014-10-01 09:40:00	THYAO
2014-10-01 09:45:00	THYAO

Finansal piyasalardaki geçmiş fiyat hareketlerine bakıldığında, gelecekteki fiyat hareketleri için tahmin yöntemlerine teknik analiz denir. Teknik analizi kullanırken ana yöntemler oluşumlar ve göstergelerdir. Göstergelere teknik analiz göstergeleri (indikatör) denir. Hisse senetleri hakkında alım satım sinyalleri vererek yatırımcılara yardımcı olur. Göstergeler hisse senedi fiyatı ve hacim verileri kullanılarak hesaplanır. Çalışmamızda en çok kullanılan gösterge türlerinden biri olan Ağırlıklı Hareketli

Ortalama (WMA) göstergesini kullandık. WMA, geçen dönemin ortalamasını gösterir ve mevcut fiyatın trende ne kadar olduğunu gösterir.

```
for i in range(15):
    indicator.append(0)

for i in range(16, len(data)):

    x=[]

    x=data.iloc[i-16:i,3].values
    x=np.array(x)
    carp_topla=[]

    for z in range(1,16):
        carp_topla.append(x[z]*z)
    carp_topla=np.array(carp_topla)
    indicator.append(carp_topla.sum()/120)

indicator=pd.Series(indicator)
```

Şekil 3. İndikatör Ekleme

Verilerdeki yakın değeri kullanarak 15 dönemlik sürenin ortalamasını alarak WMA indikatör veri kümesine ekledik (Şekil 3). İndikatör değerlerini veri setimize ekledikten sonra, 5 dakikalık periyotlarda bir önceki çizginin önceki değerleri ile kapanış değerlerinin farkını alarak yönü (eğilim) belirledik. Bu eğilim değerini pozitifse 1, negatifse 0'a ekledik. Veri setimizin son versiyonu Şekil 4'de gösterilmektedir. Her bir hissenin kapanış değeri, Şekil 5, Şekil 6, Şekil 7, Şekil 8 ve Şekil 9' gösterilmiştir.

date_time	symbol	signal_time	close	high	low	open	total_quantity	volume	weighted_average	indicator	trends
2014-10-01 10:50:00	THYAO	1412149800000	6.51	6.53	6.51	6.53	75214	490394.29859	6.51999	6.518750	0
2014-10-01 10:55:00	THYAO	1412150100000	6.52	6.52	6.51	6.51	3140	20441.46072	6.51002	6.519083	1
2014-10-01 11:00:00	THYAO	1412150400000	6.50	6.50	6.50	6.50	1334	8671.00000	6.50000	6.517000	0
2014-10-01 11:05:00	THYAO	1412150700000	6.52	6.52	6.51	6.51	93352	607804.83930	6.51089	6.517500	1
2014-10-01 11:10:00	THYAO	1412151000000	6.51	6.52	6.51	6.51	50186	326741.08074	6.51060	6.516750	0

Şekil 4. Veri Ön işlemeden sonra Borsa İstanbul Veri Seti



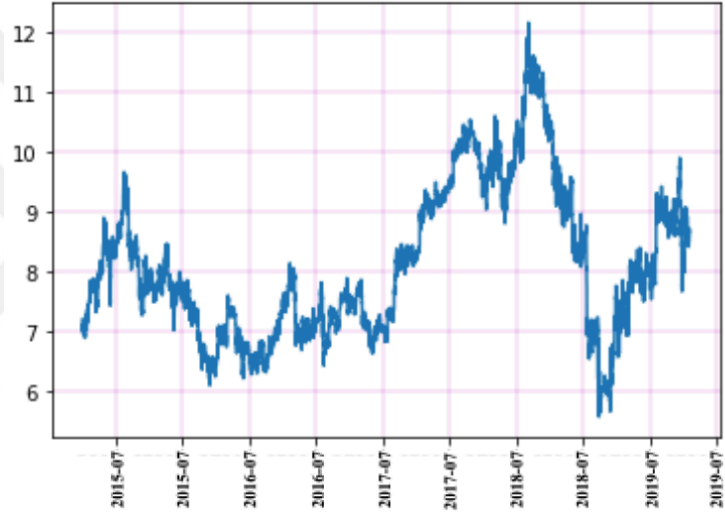
Şekil 5. AKBNK Hisse Senedi Kapanış Fiyatı



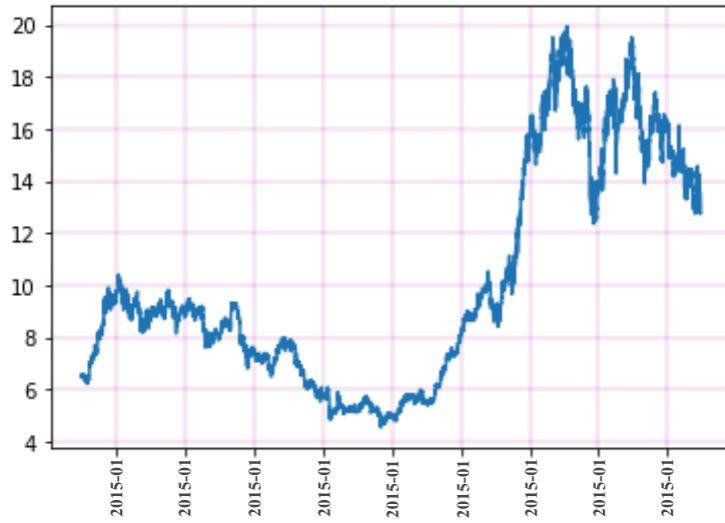
Şekil 6. ARCLK Hisse Senedi Kapanış Fiyatı



Şekil 7. ASELS Hisse Senedi Kapanış Fiyatı



Şekil 8. GARAN Hisse Senedi Kapanış Fiyatı



Şekil 9. THYAO Hisse Senedi Kapanış Fiyatı

İkinci veri seti olan Hindistan piyasasındaki TATAGLOBAL şirketine ait hisse senedi verileri günlük veriler olduğundan sadece min-max ölçekleyici dediğimiz normalizasyon işlemi uygulandı.

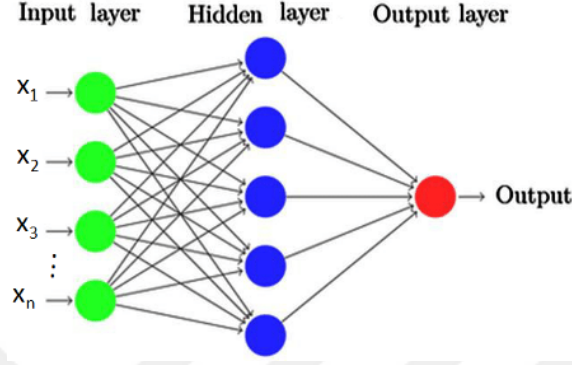
3.2 Algoritmalar

Makine öğrenimi (ML) algoritmaları yapay zekâ alanındaki en popüler yaklaşımlardan biridir. Veri miktarı arttıkça, makine öğrenimindeki teknolojik ilerlemeler hız kazanmıştır. Son on yılda, Makine Öğrenimi hayatımızın ayrılmaz parçalarından biri haline geldi. İnsan el yazısını tanımak kadar basit veya kendi kendini süren arabalar kadar karmaşık bir görevde uygulanır. Ayrıca, birkaç on yıl içinde, daha mekanik tekrarlayan görevin sona ereceği de bekleniyor. Artan veri miktarı, Makine Öğreniminin teknolojik ilerleme için gerekli bir unsur olarak daha yaygın hale geleceğine inanmak için iyi bir nedendir. ML'nin büyük bir etki yarattığı birçok önemli endüstri vardır: Finansal hizmetler, Teslimat, Pazarlama ve Satış, Sağlık Hizmetleri bunlardan birkaçı. ML uzun zamandır borsa tahmininde kullanılmaktadır. ML algoritmalarının borsa endeks hareketlerinde ve gelecekteki projeksiyonlarda başarılı olduğu bilinmektedir. ML'nin alt dalı olan derin öğrenme en az ML algoritmaları kadar başarılıdır. Son zamanlarda, derin öğrenme tabanlı yaklaşımlar stok verilerinin sınıflandırılması için iyi bir doğruluk göstermiştir. Özellikle, Tekrarlayan Sinir Ağları (RNN'ler), zaman serisi veri analizi için yaygın olarak kullanılmaktadır. LSTM, RNN'nin gelişmiş bir versiyonudur. Sıralı verilerle başa çıkma yeteneğine sahiptir ve borsa değer tahmininin eğitimi ve testi için son derece uygundur. Bu bölümde kısaca en popüler üç makine öğrenme algoritması ve bir derin öğrenme yöntemi hakkında konuşacağız. Bunlar Yapay Sinir Ağı (YSA), k-En Yakın Komşu ve Oto-Regresif Entegre Hareketli Ortalamalar (Oto ARIMA) ve derin öğrenme yöntemi olan LSTM'dir.

3.2.1 Yapay Sinir Ağları

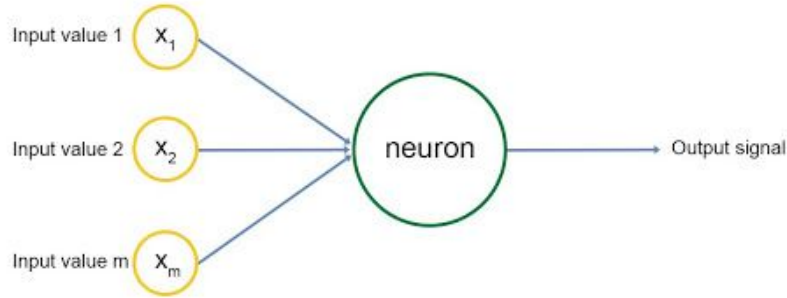
Yapay Sinir Ağları (YSA) örneklerle, tarihsel verilerle bilgi toplar ve genellemeler yapar. Daha sonra, hiç görmediği örneklere kıyasla örnek bilgilerini kullanarak bu örnekler hakkında karar verir. Bu, karmaşık sorunları başarıyla çözme yeteneğini gösterir. İnsan beyni muhtemelen dünyanın en karmaşık makinesidir. Rekor sürede sonuçlara ulaşmada çok etkilidir. YSA yapısı insan beynininkine çok benzer. YSA'nın

çeşitli mimarileri vardır, ancak en popüler olanı çok katmanlı algılayıcıdır (Şekil 10). Nöronlar üç katman halinde düzenlenmiştir: giriş, gizli ve çıkış katmanları. Giriş katmanındaki nöron sayısı, özelliklerle aynıdır ve çıkış katmanındaki nöron sayısı, sınıf sayısı ile aynıdır. Gizli katmandaki nöronlar, sınıflandırıcının doğruluğuna bağlı olarak değişebilir.



Şekil 10. MLP Mimari Yapısı

Bir nöronun üç bileşeni vardır, dendritler, akson ve nöronun ana gövdesi. Dendritler sinyalin alıcılarıdır ve akson vericidir. Tek başına, bir nöron fazla kullanışlı değildir, ancak diğer nöronlara bağlandığında, birkaç karmaşık hesaplama yapar ve gezegenimizdeki en karmaşık makineyi, insan vücudunu çalıştırmaya yardımcı olur.



Şekil 11. Bilgisayar Nöronu

Bilgisayar nöronu Şekil 11'de gösterildiği gibi benzer şekilde oluşturulur. Sarı dairelerle işaretlenmiş parçalar nöronun girişleridir. Nöron, bir hesaplamadan sonra bir çıkış sinyali yayar. Giriş katmanı nöronun dendritleridir. Çıkış sinyali akson. Her giriş sinyalinin bir ağırlığı vardır (w_i). Bu ağırlık giriş değeri ile çarpıldıktan sonra, nöron

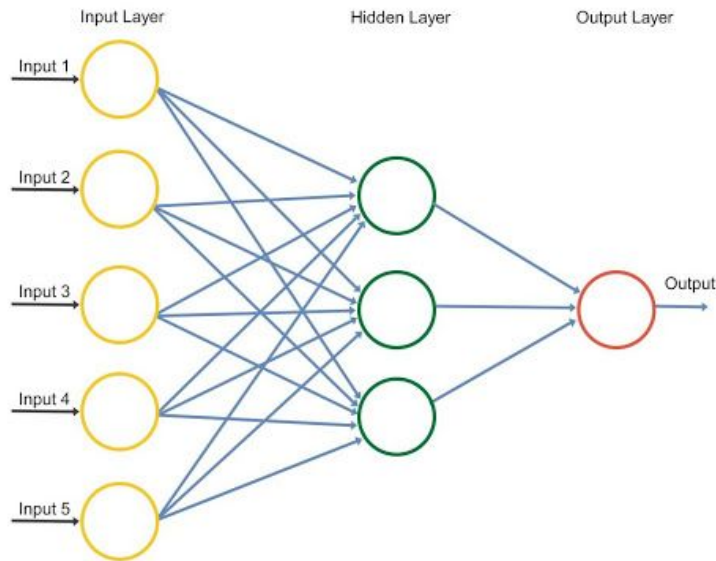
tüm giriş değişkenlerinin ağırlıklı toplamını saklar. Sinir öğrenme eğitimi aşamasında, bu ağırlıklar geri yayılım ve eğitim inişi kavramlarıyla hesaplanır. Aktivasyon fonksiyonu daha sonra elde edilen ağırlıklı toplamaya uygulanır. Daha sonra nöronun nöral çıkış sinyali ile sonuçlanır.

Sinir ağlarının çalışmasını anlamak için bir örneğe bakacağız. Giriş katmanı, bir çıkış değerine ulaşmamıza veya bir tahmin yapmamıza yardımcı olacak parametrelerden oluşur. Beynimiz aslında dokunmak, duymak, görmek, koklamak ve tatmak için duyularımız olan beş temel giriş parametresine sahiptir.

Beynimizdeki nöronlar, bu temel girdi parametrelerinden duygular ve duygular gibi daha karmaşık parametreler oluşturur. Duyularımız ve hislerimiz, temelde beynimizin sinir ağının çıktısı olan harekete geçmemize veya karar almamıza neden olur. Bu nedenle, bu durumda bir karar vermeden önce iki hesaplama katmanı vardır.

İlk katman beş duyuyu girdi olarak alır ve çıktının bir karar veya eylem olduğu bir sonraki hesaplama katmanının girdisi olan duygu ve hislerle sonuçlanır.

Bu nedenle, insan beyninin çalışmasının bu son derece basit modelinde, bir giriş katmanımız, iki gizli katmanımız ve bir çıkış katmanımız var. Tabii ki deneyimlerimizden, hepimiz beynin bundan çok daha karmaşık olduğunu biliyoruz, ama aslında beynimizde hesaplamalar bu şekilde yapılır.



Şekil 12. Örnek Yapay Sinir Ağı

Sinir ağı öğreticisinde alınan örnekte, Şekil 12’de gösterildiği gibi beş giriş parametresi vardır. Gizli katman 3 nörondan oluşur ve çıktı katmanındaki sonuç hisse senedi fiyatı için tahmindir. Gizli katmandaki 3 nöron, beş giriş parametresinin her biri için farklı ağırlıklara sahip olacak ve giriş parametrelerini girişlerin çeşitli kombinasyonlarına göre etkinleştirecek farklı aktivasyon fonksiyonlarına sahip olabilir. Örneğin, ilk nöron hacmi ve kapanış fiyatı ile açılış fiyatı arasındaki farkı inceliyor olabilir ve yüksek ve düşük fiyatlarını görmezden gelebilir. Bu durumda, yüksek ve düşük fiyatlar için ağırlıklar sıfır olacaktır. Modelin kendini elde etmek için eğittiği ağırlıklara dayanarak, nörondaki ağırlıklı toplama bir aktivasyon fonksiyonu uygulanacaktır, bu belirli nöron için bir çıkış değeri ile sonuçlanacaktır. Benzer şekilde, diğer iki nöron, bireysel aktivasyon fonksiyonlarına ve ağırlıklarına göre bir çıkış değeri ile sonuçlanacaktır. Son olarak, hisse senedi fiyatının çıktı değeri veya öngörülen değeri, her bir nöronun üç çıktı değerinin toplamı olacaktır. Sinir ağı hisse senedi fiyatlarını tahmin etmek için bu şekilde çalışacaktır. Yapay Sinir Ağı'nın bir hisse senedi fiyatının hareketini tahmin etmek için kendini nasıl eğiteceğini öğreniyor.

3.2.2 K En Yakın Komşular

K-En Yakın Mahalle (k-NN) Algoritması, makine öğrenme algoritmalarından en basit ve en yaygın olarak kullanılan sınıflandırma algoritmalarından biridir. Hem sınıflandırma hem de regresyon problemlerinde kullanılmasına rağmen, çoğunlukla sektördeki sınıflandırma problemlerinin çözümünde kullanılmaktadır. K-NN algoritması, T. M. Cover ve P. E. Hart tarafından 1967’de önerilen bir yöntemdir. K-NN algoritması, gürültülü verilere dirençli olduğu için en çok tercih edilen makine öğrenme algoritmalarından biridir. Algoritma, belirli bir sınıfla örnek veri kümesi kullanılarak kullanılır. Örnek veri setine dahil edilecek yeni veriler mevcut verilerden istiflendikten sonra, k yakın komşuları kontrol edilir. Mesafe hesaplamaları için üç farklı mesafe fonksiyonu kullanılır. Bunlar “Öklid” Uzaklığı, “Manhattan” Uzaklığı ve “Minkowski” Uzaklığıdır. K-NN verilerini kullanarak yeni veri noktalarını bu mesafe fonksiyonlarına göre sınıflandırır. Veriler en yakın komşu sınıfa atanır. En yakın komşuların sayısını artırdıkça doğruluk oranı artabilir, yani k. K-NN algoritmasında, önce k değeri belirlenir. k, belirli bir noktaya yakın olan komşuların sayısıdır. K = 3 dersek, en yakın 3 komşuya göre sınıflandırılır. Daha sonra, mevcut verilere ayarlanan veriye eklenecek yeni verilerin mesafesi, ayrı mesafe fonksiyonları

ile hesaplanır. En yakın k komşuları elde edilen mesafelerden alınır. Özellik değerlerine göre, k komşuya veya komşu sınıfına atanır.

3.2.3 Oto-Regresif Entegre Hareketli Ortalamalar

ARIMA, zaman serisi tahmini için çok popüler bir istatistiksel yöntemdir. ARIMA, Otomatik Regresif Entegre Hareketli Ortalamalar anlamına gelir. Otoregresif Entegre Hareketli Ortalamalar (ARIMA), zaman serisi verilerini tahmin etmek ve analiz etmek için kullanılan istatistiksel bir modeldir. Sağlam bir zaman serisini tahmin etmek için güçlü bir yol sağlar. Veri serisi sabittir, yani ortalama ve varyans zamanla değişmemelidir. Bir dizi, günlük dönüşümü veya seriyi farklılaştırarak durağan hale getirilebilir. Giriş olarak sağlanan veriler tek değişkenli bir seri olmalıdır, çünkü ARIMA gelecekteki değerleri tahmin etmek için geçmiş değerleri kullanır. ARIMA, model yapıyı tanımlamak için iyi bir kısaltma olmuştur. ARIMA'nın üç bileşeni vardır: AR (otoregresif terim), I (farklılaşma terimi) ve MA (hareketli ortalama terim).

AR terimi, sonraki değeri tahmin etmek için kullanılan geçmiş değerleri ifade eder. AR terimi, arima'daki 'p' parametresi ile tanımlanır.

I: Bu entegrasyon demektir. Zaman serisini hareketsizleştirmek için gözlem farklılaştırmasının tanımıdır.

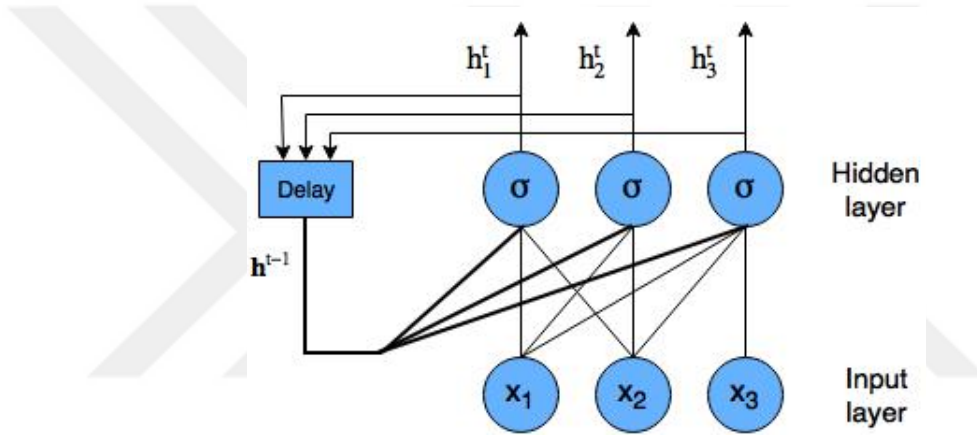
MA: Hareketli Ortalama anlamına gelir. Bir gözlem ile gecikmeli bir gözlem için uygulanan hareketli ortalama modelinden geriye kalan bir hata arasındaki bağımlılığı kullanan bir modeldir.

Bu bileşenlere eşdeğer bir standart temsil p , d , q vardır. p , bahsettiğimiz gecikme gözlemlerinin sayısıdır. d , ham gözlemlerin farklılaşma sayısıdır. q , hareketli ortalama penceresinin boyutudur. Mevsimsel olmayan ARIMA modelleri genellikle ARIMA (p , d , q) olarak belirtilir, burada p , d ve q parametreleri negatif olmayan tamsayılardır, p otoregresif modelin sırası (zaman gecikmesi sayısı) d 'dir. Fark (verilerin geçmiş değerlerin kaç kez çıkarıldığı) ve q , hareketli ortalama modelinin sırasıdır. Mevsimsel ARIMA modelleri genellikle ARIMA (p , d , q) (P , D , Q) m olarak belirtilir, burada m her mevsimdeki periyot sayısını ifade eder ve büyük P , D , Q otoregresif, farklılaşmayı ve ARIMA modelinin mevsimsel kısmı için hareketli ortalama terimleri ifade eder.

Belirtilen terimleri içeren bir model oluşturulur. Verileri durağan hale getirmek, yani modeli olumsuz etkileyen eğilimleri kaldırmak için belirli bir derece farkla hazırlanır. Modelin kullanmayacağı öğeye sahip parametre için 0 değeri kullanılabilir. Böylece ARIMA modeli yapılandırılabilir.

3.2.4 Uzun Kısa Süreli Bellek

LSTM, daha uzun geçmiş verileri hatırlama yeteneğine sahip özel bir Tekrarlayan Sinir Ağı (RNN) türüdür. Tekrarlayan bir sinir ağı, dil, hisse senedi fiyatları, elektrik talebi ve benzeri gibi zamana veya sıraya bağlı davranışları modellemeye çalışan bir sinir ağıdır. Bu, t 'deki bir nöral ağ katmanının çıkışının, $t + 1$ 'deki aynı ağ katmanının girişine geri beslenmesiyle gerçekleştirilir.

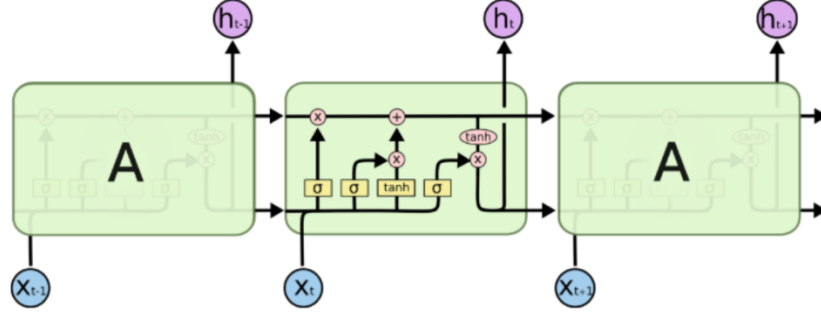


Şekil 13. Tekrarlayan Sinir Ağı (RNN) Şeması

Geleneksel bir tekrarlayan sinir ağında, degrade geri yayılma fazı sırasında, degrade sinyali, tekrarlanan gizli katmanın nöronları arasındaki bağlantılarla ilişkili ağırlık matrisi ile çok sayıda çarpılabilir. Bu, geçiş matrisindeki ağırlıkların büyüklüğünün öğrenme süreci üzerinde güçlü bir etkisi olabileceği anlamına gelir. Bu matristeki ağırlıklar küçükse, degrade sinyalinin o kadar küçük hale geldiği, yok olma gradyanları adı verilen bir duruma yol açabilir veya öğrenme çok yavaş olur veya tamamen çalışmayı durdurur. Ayrıca verilerdeki uzun vadeli bağımlılıkları öğrenme görevini daha da zorlaştırabilir. Tersine, bu matristeki ağırlıklar büyükse, gradyan sinyalinin o kadar büyük olduğu bir duruma yol açabilir ve öğrenmenin ayrışmasına neden olabilir. Bu genellikle patlayan degradeler olarak adlandırılır.

Bu konular, bellek hücresi adı verilen yeni bir yapı getiren LSTM modelinin arkasındaki ana motivasyonlardır. Ağdaki gizli katmanın yapay nöronlarının yerini

alan bir hesaplama birimi olan bellek hücrelerini sunar. Bu bellek hücreleri, ağı geçmişle etkili bir şekilde ilişkilendirilmesini sağlar. Bu nedenle, yüksek tahmin kabiliyeti sayesinde verilerin zaman içinde yapısını dinamik olarak kavrayabilir ve tahmin edebilir.



Şekil 14. Uzun-Kısa Sürekli Bellek (LSTM) Mimarisi

Bir bellek hücresi dört ana elemandan oluşur: bir giriş kapısı, kendi kendine tekrarlayan bir bağlantıya sahip bir nöron (kendisine bir bağlantı), bir unutmaya kapısı ve bir çıkış kapısı. Kendi kendini tekrarlayan bağlantının ağırlığı 1.0'dır ve dışarıdan gelen herhangi bir paraziti engellemek için bellek hücresinin durumunun bir zaman adımından diğerine sabit kalmasını sağlar. Kapılar, bellek hücresinin kendisi ve çevresi arasındaki etkileşimleri modüle etmeye yarar. Giriş kapısı, gelen sinyalin bellek hücresinin durumunu değiştirmesine veya bloke etmesine izin verebilir. Öte yandan, çıkış kapısı, bellek hücresinin durumunun diğer nöronlar üzerinde bir etkiye sahip olmasına ya da önlenmesine izin verebilir. Son olarak, unutmaya kapısı, bellek hücresinin kendi kendini tekrarlayan bağlantısını modüle ederek, hücrenin gerektiği gibi önceki durumunu hatırlamasına veya unutmaya izin verebilir.

Standart bir sinir ağı ünitesi, sadece bir aktivasyon fonksiyonu tarafından kullanıldığında ilgili giriş aktivasyonundan ve çıkış aktivasyonundan oluşur (Sundermeyer et al., 2012).

$$b_i = \tanh(a_i) \quad (1)$$

Aşağıdaki denklemler bir bellek hücreleri katmanının t adımında güncellenir. Bu denklemlerde:

- x_t , t zamanında bellek hücresi katmanına yapılan girdidir.

- $W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o$ ve V_o ağırlık matrisleridir.
- b_i, b_f, b_c ve b_o sapma vektörleridir.

İlk olarak bunun için giriş kapısını ve \tilde{C}_t değerleri t zamanında bellek hücrelerinin durumları için hesaplanıyor:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

$$C_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (3)$$

İkincisi, hafıza hücresinin zamandaki unutulmuş kapılarının aktivasyonu olan f_t değeri hesaplanıyor:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (4)$$

Giriş kapısı aktivasyonu i_t , unut kapısı aktivasyonu f_t ve aday durum değeri \tilde{C}_t 'nin değeri göz önüne alındığında, \tilde{C}_t bellek hücrelerinin yeni durumu hesaplanabilir:

$$C_t = i_t * \tilde{C}_t + f_t * C_{t-1} \quad (5)$$

Bellek hücrelerinin yeni durumu ile, çıkış kapılarının değeri ve daha sonra çıktıları hesaplanabilir (Mu, 2019):

$$o_t = \sigma(W_o x_t + U_o h_t + V_o C_t + b_o) \quad (6)$$

$$h_t = o_t * \tanh(C_t) \quad (7)$$

LSTM doğal olarak borsa tahmini için uygundur. Veriler verildiğinde, uygun verilere bağlı olarak modelleri otomatik olarak kavrayabilir. Bellek hücreleri, bilgileri bir süre saklayarak yok olma eğilimini önleyebilir. Uzun süreli bağımlılıkları öğrenebildiği için RNN'den daha avantajlıdır. LSTM zaman ve katman arasında ters geçiş için hata tutabilir. LSTM hatayı daha kararlı bir seviyede tutar.

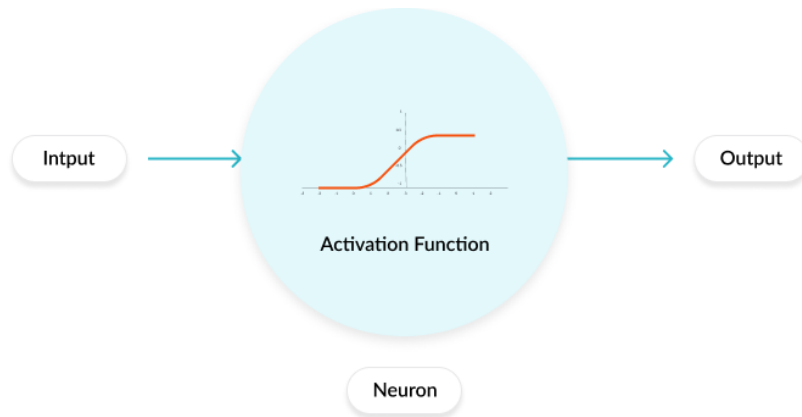
Tüm özyinelemeli sinir ağları, sinir ağının özyinelemeli modüller zinciridir. Standart RNN'lerde, bu özyinelemeli modül, tek tanh tabakası gibi çok basit bir yapıya sahip olacaktır. LSTM'ler de bu zincir benzeri yapıya sahiptir, ancak özyinelemeli modül

farklı bir yapıya sahiptir. Tek bir sinir ağı katmanına sahip olmak yerine, dört tane vardır ve çok özel bir şekilde etkileşime girerler.

3.3 Aktivasyon Fonksiyonu

Aktivasyon fonksiyonları, bir sinir ağının çıktısını belirleyen matematiksel denklemlerdir. İşlev, ağdaki her bir nörona bağlanır ve her nöronun girdisinin modelin tahmini ile ilgili olup olmadığına bağlı olarak etkinleştirilmesi (“tetiklenmesi”) gerekip gerekmediğini belirler. Aktivasyon fonksiyonları ayrıca her nöronun çıkışını 1 ile 0 arasında veya -1 ile 1 arasında normalleştirmeye yardımcı olur. Aktivasyon fonksiyonlarının ek bir yönü, her bir veri örneği için binlerce hatta milyonlarca nöron arasında hesaplandıkları için hesaplama açısından verimli olmalarıdır. Modern sinir ağları, aktivasyon fonksiyonuna ve türev fonksiyonuna artan hesaplama yükü yerleştiren modeli eğitmek için backpropagation adı verilen bir teknik kullanır. Hız ihtiyacı ReLu ve Swish gibi yeni işlevlerin geliştirilmesine yol açmıştır.

Bir sinir ağında, girdi adı verilen sayısal veri noktaları girdi katmanındaki nöronlara beslenir. Her nöronun bir ağırlığı vardır ve giriş sayısının ağırlıkla çarpılması, bir sonraki katmana aktarılan nöronun çıktısını verir. Aktivasyon fonksiyonu, mevcut nöronu besleyen giriş ile bir sonraki katmana giden çıkışı arasındaki matematiksel bir "geçit" tir. Bir kurala veya eşiğe bağlı olarak nöron çıkışını açıp kapatan bir basamak fonksiyonu kadar basit olabilir. Ya da giriş sinyallerini sinir ağının çalışması için gereken çıkış sinyallerine eşleyen bir dönüşüm olabilir.

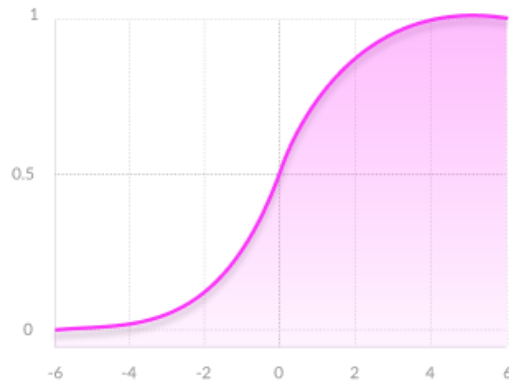


Şekil 15. Aktivasyon Fonksiyonu Şeması

Sinir ağırları giderek artan bir şekilde, ağın karmaşık verileri öğrenmesine, bir soruyu temsil eden hemen hemen her işlevi hesaplamasına ve öğrenmesine ve doğru tahminler yapmasına yardımcı olabilecek doğrusal olmayan etkinleştirme işlevlerini kullanır. Modern sinir ağı modelleri doğrusal olmayan aktivasyon fonksiyonlarını kullanır. Modelin, ağın girdi ve çıktıları arasında, doğrusal olmayan veya yüksek boyuta sahip görüntüler, video, ses ve veri setleri gibi karmaşık verileri öğrenmek ve modellemek için gerekli olan karmaşık eşlemeler oluşturmasına izin verir. Akla getirilebilen hemen hemen her işlem, aktivasyon fonksiyonunun doğrusal olmaması şartıyla, bir sinir ağında fonksiyonel bir hesaplama olarak temsil edilebilir. Doğrusal olmayan işlevler doğrusal etkinleştirme işlevinin sorunlarını giderir. Girişlerle ilgili bir türev fonksiyonuna sahip oldukları için geri çoğalmaya izin verirler. Derin bir sinir ağı oluşturmak için birden fazla nöron katmanının “istiflenmesine” izin verirler. Yüksek doğruluk seviyesine sahip karmaşık veri setlerini öğrenmek için çoklu gizli nöron katmanlarına ihtiyaç vardır.

3.3.1 Sigmoid Fonksiyonu

Sigmoid fonksiyonlar günümüzde en yaygın kullanılan aktivasyon fonksiyonlarından biridir. 'S' şekilli grafik olarak çizilen doğrusal olmayan bir fonksiyondur. X değerlerinin -2 ile 2 arasında olduğuna dikkat edilirse, Y değerleri çok diktir. Bu, x'deki küçük değişikliklerin Y değerinde büyük değişiklikler getireceği anlamına gelir.



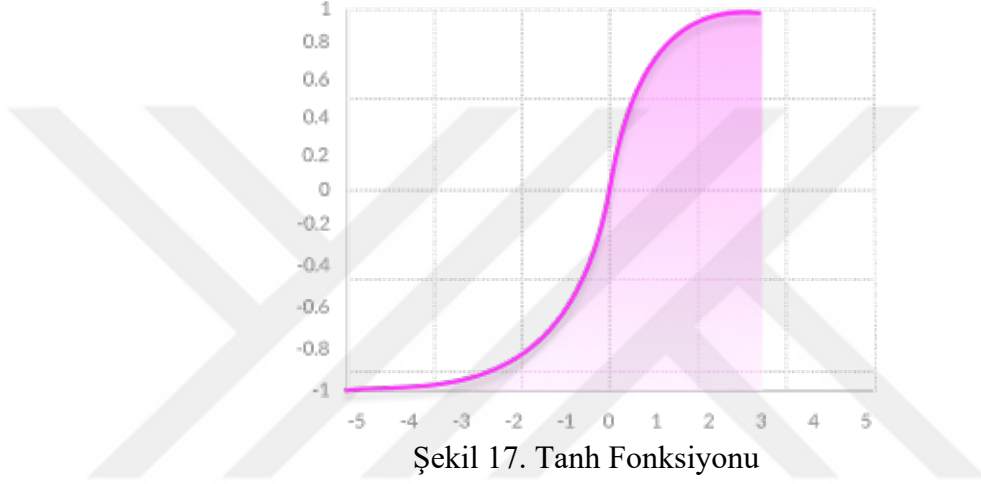
Şekil 16. Sigmoid Fonksiyonu

$$A = \frac{1}{1 + e^{-x}} \quad (8)$$

Sigmoid fonksiyonu Genellikle sonucun 0 veya 1 olduđu ikili bir sınıflandırmanın çıktı katmanında kullanılır, çünkü sigmoid fonksiyonunun değeri sadece 0 ile 1 arasındadır, aksi halde değeri 0,5 ve 0'dan büyükse, sonuç kolayca 1 olarak tahmin edilebilir.

3.3.2 Tanh Fonksiyonu

Hemen hemen her zaman sigmoid fonksiyonundan daha iyi çalışan aktivasyon Tanh fonksiyonudur, ayrıca Teğet Hiperbolik fonksiyon olarak da bilinir. Aslında sigmoid fonksiyonunun matematiksel olarak değiştirilmiş versiyonudur. Her ikisi de benzerdir ve birbirinden türetilebilir.

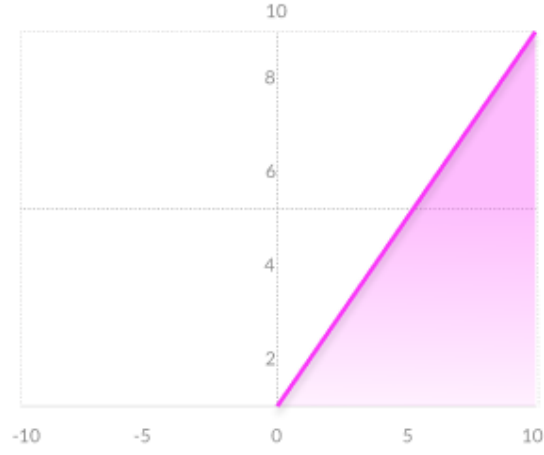


$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (9)$$

Genellikle bir sinir ağının gizli katmanlarında değeri -1 ile 1 arasında olduđu için kullanılır, bu nedenle gizli katman için ortalama 0 veya çok yakın olur, bu nedenle ortalamaları 0'a yakın getirerek verileri ortalamaya yardımcı olur. Sonraki katman için öğrenme çok daha kolay.

3.3.3 ReLu Fonksiyonu

ReLu aktivasyon fonksiyonu doğrultulmuş doğrusal birim anlamına gelir. En yaygın kullanılan aktivasyon fonksiyonudur. Temel olarak Sinir ağının gizli katmanlarında uygulanır.



Şekil 18. ReLu Fonksiyonu

$$A(x) = \max(0, x) \quad (10)$$

ReLu işlevi yukarıda gösterildiği gibidir. X pozitifse x, aksi takdirde 0 çıktısı verir. ReLu, daha basit matematiksel işlemleri içerdiğinden, tanh ve sigmoid'den daha az hesaplama açısından pahalıdır. Bir seferde sadece birkaç nöron aktive edilir, bu da ağı seyrek yapar ve hesaplama için verimli ve kolay hale getirir.

DÖRDÜNCÜ BÖLÜM

YÖNTEM

Bu bölümde makine öğrenme algoritmalarının ve LSTM modelinin borsa tahmini için uygulanış şekilleri ve oluşturulan tahmin modelleri anlatılacaktır.

4.1 Makine Öğrenme Yöntemleri

4.1.1 Yapay Sinir Ağları Modeli

Yapay Sinir Ağlarının (YSA) gelecekteki tahminde tarihsel verileri öğrenerek çalıştığı sonucuna varıldı. YSA Modelleri, zaman serisi tahminlerinde tercih edilen yöntem olmuştur. Veri seti, 2010-2019 yılları arasında TATA-GLOBAL şirketine ait 9 yıllık günlük verilerden oluşmaktadır. İlk olarak gerekli birkaç kütüphaneler programa aktarıldı. Modeli eğitmek ve test etmek için veri kümesini içe aktarmamıza ve hazırlamamıza yardımcı olacak kütüphaneler içe aktarıldı. Numpy bilimsel hesaplama için temel bir pakettir, bu kütüphaneyi veri setindeki hesaplamalar için kullanıldı. Kütüphane np diğer adı kullanılarak içe aktarılır. Pandas kütüphanesi, Python'daki yapay sinir ağını oluşturmak için kod boyunca kullanılacak güçlü veri çerçevesi nesnesini kullanmamıza yardımcı olur. Daha sonra "NSE-TATAGLOBAL.csv" adlı csv dosyasında depolanan veri seti içe aktarıldı. Bu panda kütüphanesi kullanılarak yapılır. Veriler veri kümesi adlı bir veri çerçevesinde saklanır. Veri seti %70 eğitim ve %30 test verilerine ayrıldı. Veri önışlemedeki diğer bir önemli adım, veri kümesini standartlaştırmaktır. Bu işlem, tüm giriş özelliklerinin ortalamasını sıfıra eşitle ve varyanslarını 1'e dönüştürür. Bu, tüm giriş özelliklerinin farklı ölçekleri nedeniyle modeli eğitirken herhangi bir yanlılık olmamasını sağlar. Bu yapılmazsa, sinir ağı karışabilir ve diğerlerinden daha yüksek ortalama değere sahip özelliklere daha fazla ağırlık verebilir. Bu adım sklearn.preprocessing kütüphanesinden Min-Max Scaler yöntemi içe aktarılarak uygulandı. scaler değişkeni Min-Max Scaler () işleviyle başlatıldı. Bundan sonra, x ve y veri kümelerinde bu değişiklikleri uygulamak için fittransform işlevi kullanılıyor. Veri kümeleri hazır olduğuna göre, Keras kütüphanesini kullanarak Yapay Sinir Ağını oluşturmaya devam edebiliriz.

```

dataset_train = pd.read_csv("NSE-TATAGLOBAL.csv")
train = dataset_train.iloc[:,3:].values
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0, 1))
scaler2 = MinMaxScaler(feature_range = (0, 1))
y=train[:,0]
x=train[:,1:]
x_scaled=scaler.fit_transform(x)
y_scaled=scaler2.fit_transform(y.reshape(-1,1))
train_scaled=np.concatenate((y_scaled,x_scaled),axis=1)
train_scaled.shape

```

Şekil 19. Eğitim Veri Seti Ön İşleme

```

dataset_test = pd.read_csv("NSE-TATAGLOBAL_test.csv")
test = dataset_test.iloc[:,3:].values

scaler_testx = MinMaxScaler(feature_range = (0, 1))
scaler_testy = MinMaxScaler(feature_range = (0, 1))
ytest=test[:,0]
xtest=test[:,1:]
xtest_scaled=scaler_testx.fit_transform(xtest)
ytest_scaled=scaler_testy.fit_transform(ytest.reshape(-1,1))
test_scaled=np.concatenate((ytest_scaled,xtest_scaled),axis=1)
test_scaled.shape

```

Şekil 20. Test Veri Seti Ön İşleme

Şimdi yapay sinir ağını oluşturmak için kullanılacak fonksiyonlar içe aktarıldı. Sıralı yöntemi tensorflow.python.keras.models kütüphanesinden içe aktarıyoruz. Bu, sinir ağları öğreniminin katmanlarını sırayla oluşturmak için kullanılacaktır. İçe aktardığımız bir sonraki yöntem tensorflow.python.keras.layers kitaplığındaki Dense işlevi olacaktır. Bu yöntem yapay sinir ağıımızın katmanlarını oluşturmak için kullanılacaktır.

Oluşturulan YSA Modeli, 6 giriş parametresinden (Açık-Yüksek-Düşük-Kapat-Toplam Miktar), iki gizli katman ve bir çıkış katmanından oluşur. Gizli katmanlarının birim değeri 128 olarak ayarlandı. Birimler değeri, gizli katmandaki nöron veya düğümlerin sayısıdır. Çıktı katmanındaki nöron, hisse senedi kapanış fiyatı tahminidir. Gizli katmandaki nöronlar, bu giriş parametrelerini etkinleştirmek için altı giriş parametresinin ve aktivasyon fonksiyonlarının her biri için farklı ağırlıklara sahip olacaktır.

```
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense
from tensorflow.python.keras.layers import Flatten

model = Sequential()

model.add(Flatten())
model.add(Dense(units = 128, kernel_initializer = 'uniform', activation = 'relu', input_dim = X_train.shape[1]))
```

Şekil 21. Yapay Sinir Ağı için Gerekli Kütüphaneler

Oluşturduğumuz model değişkeni Sequential() işlevi ile başlatıldı. Bu değişken daha sonra python'da yapay sinir ağı öğreniminin katmanlarını oluşturmak için kullanılacaktır. Modele katman eklemek için add() işlevi kullanıldı. Add işlevinin bağımsız değişkeni, Dense () işlevidir ve sırayla aşağıdaki bağımsız değişkenlere sahiptir:

Units: Bu, belirli bir katmandaki düğüm veya nöron sayısını tanımlar. Bu değer 128 olarak belirlendi, yani gizli katmanda 128 nöron olacak.

Kernel_initializer: Gizli katmandaki farklı nöronların ağırlıkları için başlangıç değerlerini tanımlar. Bunu "aynı, değişmeyen" olarak tanımladık, yani ağırlıklar tekdüze bir dağılımdan değerler ile başlatılacak.

Activation: Bu, belirli gizli katmandaki nöronlar için aktivasyon fonksiyonudur. Burada fonksiyon rektifiye edilmiş Doğrusal Birim fonksiyonu veya "ReLU" olarak tanımlanıyor.

Input_dim: Bu, gizli katmana giriş sayısını tanımlar, bu değeri, girdi özelliği veri çerçevesinin sütun sayısına eşit olarak tanımlandı. Model, önceki katmanın kaç çıktı ürettiğini bileceğinden, bu katman sonraki katmanlarda gerekli olmayacaktır.

```
model.add(Dense(units = 128, kernel_initializer = 'uniform', activation = 'relu'))
```

Şekil 22. YSA Modeli İkinci Katman

Daha sonra, etkinleştirme işlevi olarak düzgün bir çekirdek başlatıcı ve aktivasyon fonksiyonu "ReLU" olan 128 nöronlu ikinci bir katman ekleniyor. Model, kendisini elde etmek için eğittiği ağırlıklara dayanarak nöronda ağırlıklı bir ilave aktivasyon fonksiyonu uygulayacaktır ve bu, belirli nöron için çıkış değeri olacaktır. Benzer şekilde, diğer nöronların bireysel aktivasyon fonksiyonlarına ve ağırlıklarına göre bir çıkış değeri olacaktır. Bu sinir ağında sadece iki gizli katman oluşturuluyor.

```
model.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))
```

Şekil 23. YSA Modeli Son Katman

Yapılan bir sonraki katman, tek bir çıktıya ihtiyaç duyulan çıktı katmanı olacak. Bu nedenle, geçen birimler 1'dir ve aktivasyon fonksiyonu Sigmoid işlevi olarak seçilir, çünkü tahminin pazarın yukarı doğru hareket etme olasılığı olması istenilir.

```
model.compile(optimizer = 'adam', loss = 'mean_squared_error')
```

Şekil 24. YSA Modeli Optimizer ve Kayıp

Son olarak, aşağıdaki argümanları ileterek model derlendi:

Optimizer: Optimizer, stokastik gradyan inişinin bir uzantısı olan "adam" olarak seçilir.

Loss: Bu, eğitim süresi boyunca optimize edilecek kaybı tanımlar. Bu kayıp ortalama kare hatası olarak tanımlandı.

```
model.fit(X_train, y_train, batch_size = 10, epochs = 50)
```

Şekil 25. YSA Modeli Epoch ve Batch_size Sayısı

Şimdi yaratılan sinir ağı eğitim veri setlerine uygulanması gerekiyor. Bu, Xtrain, ytrain, batch_size ve fit () işlevindeki dönem sayısını geçirerek yapılır. Batch_size, modelin hataları geri yaymadan ve ağırlıklarda değişiklik yapmadan önce hatayı hesaplamak için kullandığı veri noktalarının sayısını ifade eder. Epoch sayısı, modelin eğitiminin eğitim veri kümesinde gerçekleştirilme sayısını temsil eder. Geriye dönük işlem 50 Epoch olarak tekrarlandı. Oluşturulan YSA modeli hisse senedi fiyatı tahmini için bu şekilde çalışmaktadır.

4.1.2 K-En Yakın Komşu Modeli

Python'da KNN Algoritmasını uygulamak için gereken kütüphaneleri içe aktarıldı.

```

import numpy as np
import pandas as pd
import quandl

import matplotlib.pyplot as plt

from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

```

Şekil 26. k-En Yakın Komşu Modeli Kütüphaneler

Bilimsel hesaplama için numpy kütüphaneleri eklendi. Ardından, grafiği çizmek için matplotlib.pyplot kütüphanesi içe aktarıldı. Doğruluk sınıflandırma puanı için sklearn.metrics'ten k-en yakın komşu oyu ve accuracyscore'u uygulamak için sklearn.neighbors'tan iki makine öğrenme kütüphanesi KNeighborsClassifier içe aktarıldı.

```

df = quandl.get("NSE/TATAGLOBAL", authtoken="drwupy8C2ciy4Hu4LxCf")
df.dropna()
df = df[['Open', 'High', 'Low', 'Close']]
df.head()

```

	Open	High	Low	Close
Date				
2010-07-21	122.1	123.00	121.05	121.55
2010-07-22	120.3	122.00	120.25	120.90
2010-07-23	121.8	121.95	120.25	120.65
2010-07-26	120.1	121.00	117.10	117.60
2010-07-27	117.6	119.50	112.00	118.65

Şekil 27. Veri Seti

Daha sonra “NSE-TATAGLOBAL.csv” adlı csv dosyasında depolanan veri seti içe aktarıldı. Bu panda kütüphanesi kullanılarak yapılır. Veriler df adlı bir veri çerçevesinde saklanır.

```

split_percentage = 0.7
split = int(split_percentage*len(df))

X_train = X[:split]
Y_train = Y[:split]

X_test = X[split:]
Y_test = Y[split:]

```

Şekil 28. K-En Yakın Komşu Modeli Test-Eğitim Seti İşlemi

Şimdi, veri kümesi eğitim veri kümesine ve test veri kümesine ayrıldı. Verilerin %70'i eğitilmek, geri kalan %30'u test etmek için kullanıldı. Bunu yapmak için, veri çerçevesini 70-30 oranında bölecek bir bölünmüş parametre oluşturuldu. "Xtrain" ve "Ytrain" eğitim veri kümesidir. "Xtest" ve "Ytest" test veri kümesidir.

```
knn = KNeighborsClassifier(n_neighbors=5)

knn.fit(X_train, Y_train)

accuracy_train = accuracy_score(Y_train, knn.predict(X_train))
accuracy_test = accuracy_score(Y_test, knn.predict(X_test))

print('Train_data Accuracy: %.2f' %accuracy_train)
print('Test_data Accuracy: %.2f' %accuracy_test)
```

Şekil 29. K-En Yakın Komşu Modeli

Veri kümesi eğitim ve test veri kümesine bölündükten sonra, en yakın k sınıflandırıcısı başlatıldı. Burada "k = 5" kullanıldı, k değerini değiştirebilir ve sonuçtaki değişikliği fark edebilir. Ardından, 'fit' işlevi kullanılarak eğitim verileri sığdırılıyor. Ardından, 'accuracy_score' işlevi kullanılarak eğitim ve test doğruluğu hesaplandı.

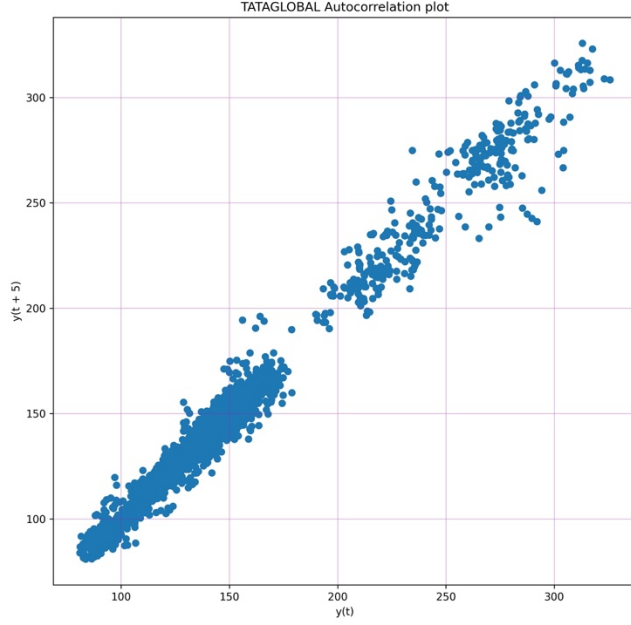
4.1.3 Otoresif Entegre Hareketli Ortalamalar Modeli

Bu bölümde ARIMA kullanarak zaman serisi tahmini için nasıl bir model oluşturulduğu anlatılacaktır. ARIMA modeli kısa vadeli tahminlerde daha iyi sonuçlar verebilir. İlk olarak gerekli kütüphaneleri projemize ekledik.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pandas.plotting import lag_plot
from pandas import datetime
from statsmodels.tsa.arima_model import ARIMA
from sklearn.metrics import mean_squared_error
```

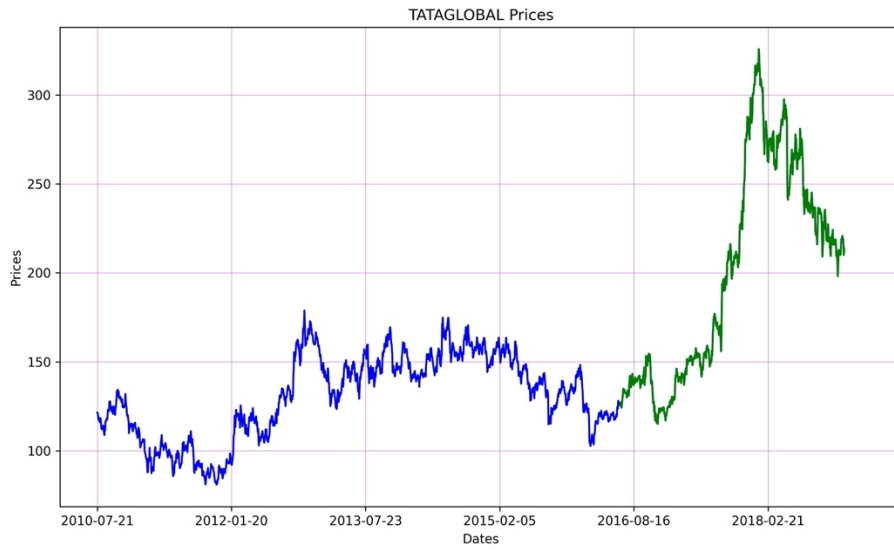
Şekil 30. ARIMA Modeli için Gerekli Kütüphaneler

Veri setimiz ARIMA modeliyle uyumludur. Bu veri seti yedi farklı özellikten oluşmaktadır (Şekil 1). Bu modelde, veri kümesindeki kapanış değeri incelendi. Aynı analiz diğer özelliklerin çoğu için tekrarlanabilir.



Şekil 31. TATAGLOBAL Oto korelasyon Grafiği

Zaman serilerini analiz etmeden önce, kapanış değerinin oto korelasyon grafiğini (Şekil 31) sabit bir gecikme süresine göre analiz edildi. Şekil 10'daki sonuçlara baktığımızda, ARIMA'yı uygulamak için uygun verilerin olduğunu görüyoruz. Daha sonra verileri %70 eğitim ve %30 test setlerine ayırdı. Zaman serisinin nasıl görüldüğünü görmek için eğitim ve test veri kümeleri çizdirildi (Şekil 32).



Şekil 32. TATAGLOBAL Kapanış Fiyatı

Veri setini analiz ettikten sonra, ARIMA modelini oluşturmak için p, d, q değerlerinin belirlenmesi gerekir. Bu değerleri belirlemek için korelasyon grafiklerine veya otomatik korelasyon grafiklerine bakmak gibi birçok yöntem vardır. Akaike Bilgi Ölçütü (AIC) yöntemini kullanarak p, d, q için en uygun değeri bulduk. AIC, bir veri kümesi için kalite modeli ölçüsüdür. Veriler için bir model koleksiyonu verildiğinde AIC, her modelin kalitesini, diğer modellerin her birine göre tahmin eder. En küçük AIC değerine sahip model, en uygun model olarak seçilir. Bu hesaplamayı python'da pmdarima.arima kütüphanesi ile hızlı bir şekilde yapılmasını sağlar. Bu kütüphane, bir dizi p, d, q, P, D ve Q değeri ayarlanmasını ve ardından tüm olası kombinasyonlar için modellerin sığdırılmasını sağlayan bir auto_arima işlevi içerir. Daha sonra model, en iyi AIC değerini bildiren kombinasyonu koruyacaktır. AIC değeri en düşük olan modelin p, d, q değerleri (1,1,1) olarak belirlenmiştir. Bu değerler modelde kullanıldı. Modeli test ve eğitim olarak ayrılan veri setlerinden eğitim setine uygulandı.

4.2 Derin Öğrenme Yöntemi

4.2.1 Uzun-Kısa Süreli Bellek

LSTM modeli oluşturulmadan önce verileri eğitim ve test olarak ayrıldı. Ama bir zaman serisi olduğu için shuffle = False yapmak zorundaydık.

```
dataset_train = pd.read_csv("thy_labeled_version.csv")
train = dataset_train.iloc[:,3:].values
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0, 1))
scaler2 = MinMaxScaler(feature_range = (0, 1))
y=train[:,0]
x=train[:,1:]
x_scaled=scaler.fit_transform(x)
y_scaled=scaler2.fit_transform(y.reshape(-1,1))
train_scaled=np.concatenate((y_scaled,x_scaled),axis=1)
train_scaled.shape
```

Şekil 33. LSTM Modeli Veri Seti Ayırma

Veri miktarı yüksek olduğundan, %0,3'ü test verisi olarak ayrıldı. Test ve eğitim olarak iki ayrı dosya oluşturuldu. Oluşturulan dosyalara Min-Max ölçekleyici normalleştirme yöntemi uygulandı.

```

X_test = []
y_test = []
timesteps = 50
for i in range(timesteps, len(test_scaled)):
    X_test.append(test_scaled[i-timesteps:i,1:])

    y_test.append(test_scaled[i, 0])
X_test, y_test = np.array(X_test), np.array(y_test)

```

Şekil 34. Zaman Aşımı Algoritması

Yine, zaman serilerinde verimli olmayı öğrenmek için gerekli olan zaman aşımı algoritmasını uyguladık (Şekil 34). Algoritmanın çalışma mantığı, belirttiğiniz veri sayısını gruplandırır ve önceki değerlerin belirlediği sayıdan sonraki değeri tahmin etmeye çalışır. Projemizde bu değeri 50 olarak belirledik. Her biri 50 olarak gruplandırdıktan sonra, önceki 50 değerle karşılaştırıldığında 51'inci tahmin etmeye çalışır. Bu özellikle LSTM modeli için çok uygundur. LSTM bu gruplamayı Uzun Süreli Bellekte saklar.

```

from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense
from tensorflow.python.keras.layers import LSTM
from tensorflow.python.keras.layers import Dropout

model = Sequential()

model.add(LSTM(units = 50,activation='tanh', return_sequences = True,
              input_shape = (X_train.shape[1], 8)))
model.add(Dropout(0.2))

model.add(LSTM(units = 50,activation='tanh', return_sequences = True))
model.add(Dropout(0.2))

model.add(LSTM(units = 50,activation='tanh', return_sequences = True))
model.add(Dropout(0.2))

model.add(LSTM(units = 50,return_sequences=False))
model.add(Dropout(0.2))

model.add(Dense(units = 1))

```

Şekil 35. LSTM Modeli Katmanlar

LSTM, Tekrarlayan Sinir Ağının (RNN) gelişmiş bir versiyonudur. Sıralı verilerle başa çıkma yeteneğine sahiptir ve borsa değer tahmininin eğitimi ve testi için son

derece uygundur. Model 1 giriş, 4 gizli ve 1 çıkış olmak üzere 6 katmandan oluşuyor. Deney düzeneğinde aktivasyon olarak hiperbolik tanjant fonksiyonuna sahip her katta 50 ünite içeren dört LSTM katmanı kullandık. LSTM modeline uyan gizli katmanlar nedeniyle hiperbolik tanjant fonksiyonunu kullandık. Teğet fonksiyon türevinin avantajı 0'ın sonlarına yaklaşmaktır ve daha uzun öğrenme fırsatları sunar.

Derin Öğrenmenin en büyük sorunu kuşkusuz aşırı uyum sağlamaktır. Verileri ezberlemek, eğitim aşamasında düşük kayıp ve yüksek doğruluk sağlarken, test aşamasında tam bir felakete dönüşür. Bunu önlemek için, aşırı sığdırma problemini önlemek için LSTM katmanlarının her birine %2'lik bir dropout ekledik. Dropout bazı düğümleri rastgele kaldırır. Böylece aynı değerlerin ezberlenmesi sorunu önlenmiş olur.

```
model.compile(optimizer = 'adam', loss = 'mean_squared_error')  
model.fit(X_train, y_train, epochs = 100, batch_size = 32)
```

```
Epoch 1/100  
3124/3124 [=====] - 261s 84ms/step - loss: 0.0015  
Epoch 2/100  
3124/3124 [=====] - 260s 83ms/step - loss: 6.7475e-04  
Epoch 3/100  
3124/3124 [=====] - 254s 81ms/step - loss: 5.8053e-04
```

Şekil 36. LSTM Modeli

Ağın son katmanı, regresyon çıkışı için bir birim içeren yoğun bir katmandan oluşuyordu. Kayıp fonksiyonu olarak ortalama kare hatası kullanıldığında varsayılan Adam optimizer'ı kullandık. Öğrenme oranını düşürdüğü için daha iyi bir öğrenme süreci sunar. Kayıp değerlendirmesinde, ortalama kare hata fonksiyonunu kullandık. Eğitim sürecini hızlandırmak için toplam dönem sayısı (Epoch sayısı) 100'e ve parti büyüklüğü (batch_size) 32'ye ayarlanmıştır.

BEŞİNCİ BÖLÜM

DENEYSEL SONUÇLAR

Bu bölümde uyguladığımız yöntemlerden elde ettiğimiz sonuçlar, başarı oranları, karşılaştırmalar ve hata oranlarını hesapladığımız metrikler anlatılacaktır.

5.1 Hata Oranı Metrik Hesaplamaları

Model oluşturmak için kullanılan algoritmalar, optimizasyonları gerçekleştirmek için hata metriğini kullanır. Hata metriği seçimi son modeli ve performansını değerlendirme şeklini etkiler, bu nedenle hata metrikleri arasındaki farkı anlamak önemlidir. Ortalama Mutlak Hata (MAE) ve Kök ortalama kare hatası (RMSE), sürekli değişkenlerin doğruluğunu ölçmek için kullanılan hem istatistiklerde hem de makine öğreniminde regresyondaki en yaygın metriklerden ikisidir. Aynı görev için kullanılmasına rağmen (tahminlerdeki hataları anlamak) ikisi arasında önemli farklılıklar vardır.

RMSE, hatanın ortalama büyüklüğünü ölçen ikinci dereceden bir puanlama kuralıdır. Tahmin ve gerçek gözlem arasındaki ortalama kare farklılıklarının kareköküdür. RMSE, borsa trendlerindeki tahmin hatasının ortalama büyüklüğünü şu şekilde hesaplayabilir:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (forecast(t) - actual(t))^2}{n}} \quad (11)$$

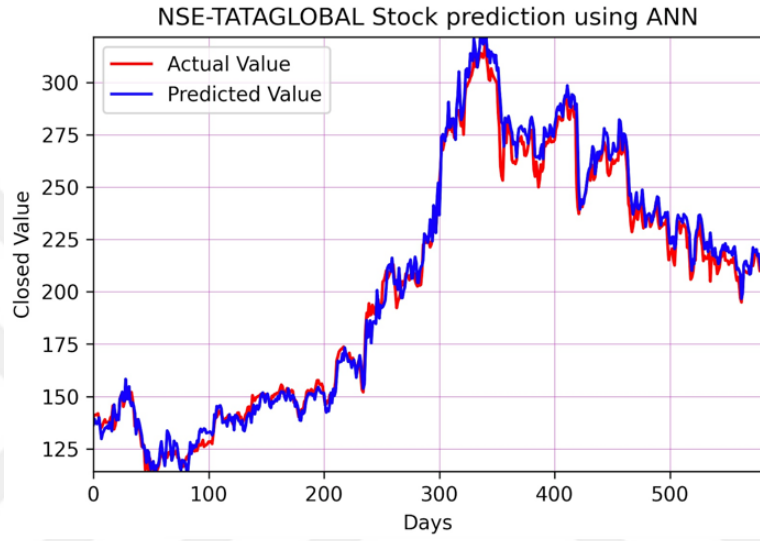
Formülü kelimelerle ifade edersek, tahmin ve karşılık gelen gözlenen değerler arasındaki farkın her birinin karesi alınır ve daha sonra numune üzerinden ortalaması alınır. Son olarak, ortalamanın kare kökü alınır. Hataların ortalaması alınmadan önce kareler alındığından, RMSE büyük hatalara nispeten yüksek bir ağırlık verir. Bu, RMSE'nin özellikle istenmeyen büyük hatalar olduğunda en yararlı olduğu anlamına gelir. Ortalama mutlak hata (MAE), beklenen tüm değerler ile öngörülen değerler arasındaki farkların mutlak değerinin toplamının toplam tahmin sayısına bölünmesiyle elde edilir. MAE, borsa endekslerinin tahminindeki ortalama hata ölçüsünü şu şekilde hesaplanır:

$$MAE = \frac{\sum_{t=1}^n |forecast(t) - actual(t)|}{n} \quad (12)$$

5.2 Makine Öğrenme Algoritmalarının Sonuçları

5.2.1 Yapay Sinir Ağları Modeli Sonuçları

Oluşturulan YSA modeli ile oldukça iyi sonuçlar elde edildi. YSA yapısı, tarihsel verileri öğrenerek geleceğe ilişkin tahminlerde bulunur ve zaman serisi tahmini için ne kadar uygun olduğunu gösterir. Elde edilen YSA modeline göre yakın değerleri karşılaştıran grafik Şekil 37'deki gibidir.



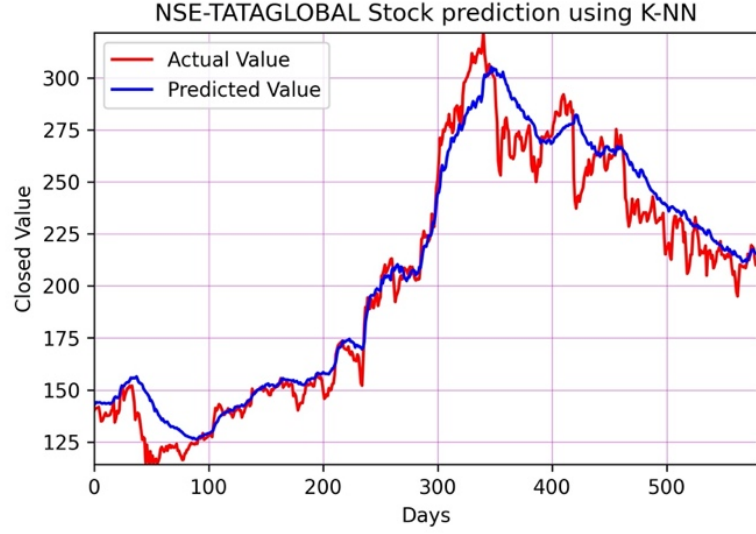
Şekil 37. TATAGLOBAL YSA Model Sonuçları

Oluşturulan YSA modelinin doğruluk oranı grafiklerde görüldüğü gibi yüksek bir başarıya sahiptir. Doğruluk oranını ve grafikler dışındaki Kök Ortalama Kare Hatası (RMSE) ve Ortalama Mutlak Hata (MAE) metrikleri değerlendirildi. YSA modelinin RMSE değerini 0,22 ve MAE değerini 0,16 olarak hesaplandı. Hata oranını hesaplayan RMSE değerinin oldukça düşük olduğunu görüyoruz. YSA modelinin zaman serisindeki başarısı oldukça yüksektir. YSA eğitiminin sonuna gelindiğinde, artık Python'da kendi Yapay Sinir Ağını oluşturabileceğinize ve makinelerinizin gücünü ve zekasını kullanarak ticarete başlayabileceğinize inanıyoruz.

5.2.2 K-En Yakın Komşu Modeli Sonuçları

KNN algoritması, makine öğrenme algoritmalarında büyük öneme sahiptir. Sınıflandırma problemlerinde çok yüksek bir başarıya sahiptir. Oluşturulan k-NN algoritması ile YSA modeline göre daha düşük bir tahmin oranı elde edildi. YSA

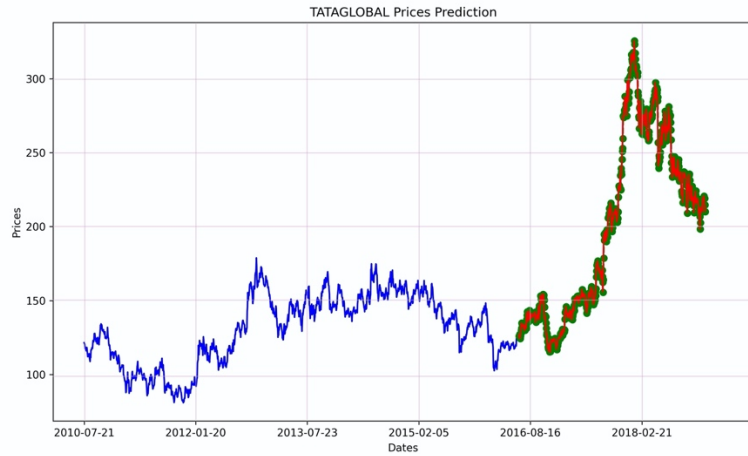
modelinden daha düşük olmasına rağmen, gerçek değerlere yakın değerler elde edildi. K-NN modelinin RMSE değeri 3.10 ve MAE değeri 2.88'dir.



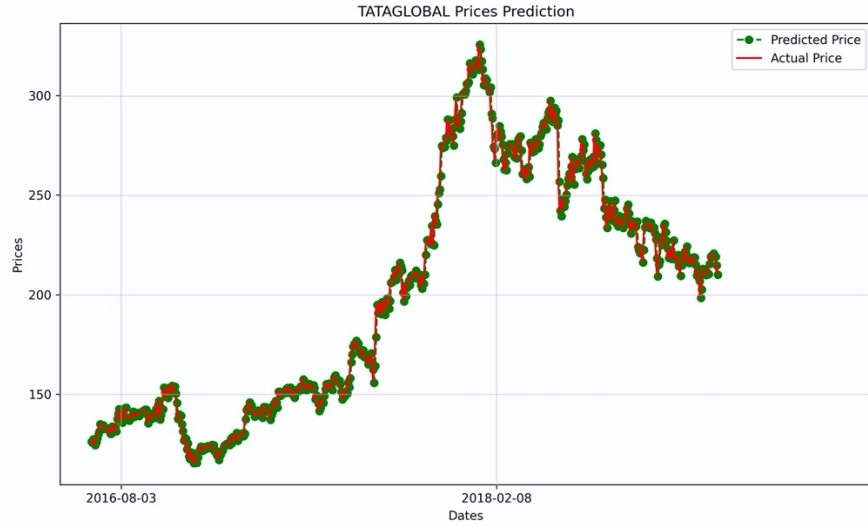
Şekil 38. TATAGLOBAL K-NN Modeli Sonuçları

5.2.3 Oto regresif Entegre Hareketli Ortalamalar Modeli Sonuçları

Auto ARIMA Model, önceki çalışmalarda zaman serilerinde yüksek bir başarı oranı göstermiştir. Oluşturulan ARIMA modeli ile AIC yöntemi ile hesaplanan p, d, q parametreleri ile yüksek doğruluk oranına ulaşıldı. Oluşturulan tahmin değerleri orijinal verilerle karşılaştırıldı. Tahminin, gerçek verileri olan test veri setiyle ne kadar iyi eşleştiğini görmek için iki farklı grafik çizdirildi (Şekil 39 ve Şekil 40).



Şekil 39. TATAGLOBAL Auto ARIMA Model Sonuçları



Şekil 40. TATAGLOBAL Auto ARIMA Model Sonuçları

ARIMA modelinin RMSE değerini 0.096 ve MAE değerini 0,051 olarak hesapladık. ML algoritmalarına ait sonuçların olduğu tablo Tablo 3'te verilmiştir. Burda yapılan çalışmaların haricinde Davidsson (Vainionpää & Davidsson, 2014), Wanjawaya (Wanjawa & Muchemi, 2014) ve Pengfei Yu (Yu & Yan, 2020)'a ait 3 farklı benzer çalışmaya ait sonuçlar da Tablo3'te gösterilmektedir.

Tablo 3. Makine Öğrenimi Algoritmaları Doğruluk Oranları

Modeller	Doğruluk (%)	RMSE	MAE
TATA-GLOBAL			
ANN	97.52	0.22	0.16
Auto-ARIMA	98.91	0.096	0.051
k-NN	88.02	3.10	2.88
COCA COLA			
ANN	-	1.83	MAPE: 0.71
S&P 500			
ARIMA	51.16	4.53	MAPE: 0.33
H&M			
k-NN	-	2.44	MPE: 0.763

Bu tezde yapılan ML çalışmaları ile diğer yapılan benzer çalışmaların karşılaştırılması yapılmıştır. RMSE değerlerinden hareket edecek olursak ANN ve Auto-ARIMA modelimizin Coca-Colaya ve ASE veri setiyle çalışan modellerden daha yüksek bir başarı elde edildiği sonucuna varılmaktadır. Bir diğer benzer çalışma olan H&M hissesi kullanılarak oluşturulan k-NN modelinin burada yapılan çalışmaya kıyasla daha başarılı olduğu görülmektedir.

5.3 Derin Öğrenme Yöntemi Sonuçları

5.3.1 Uzun-Kısa Süreli Bellek Modeli Sonuçları

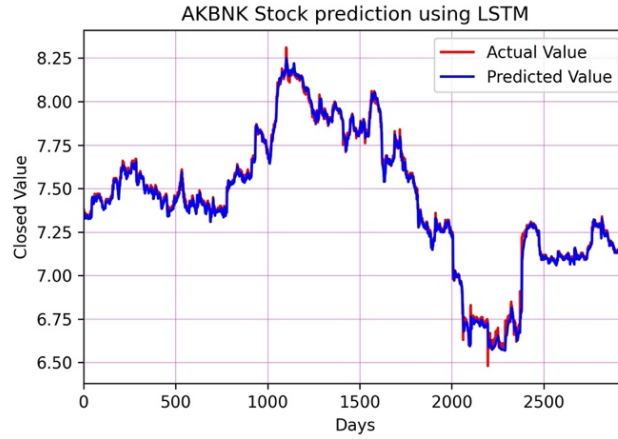
Önerilen yöntemin verimliliğini test etmek için bir dizi deney gerçekleştirildi. Aşağıda sunulan sonuçlar her hisse senedi için doğruluk oranlarıdır. Önerilen sistemin performansı, gerçek değer ile öngörülen değerler karşılaştırılarak değerlendirilir. Sonuçların doğruluğu genel doğruluk, Kök Ortalama Kare Hatası (RMSE) ve Ortalama Mutlak Hata (MAE) metrikleri ile değerlendirildi. Tablo 4, kullanılan veri kümesinde önerilen yöntem için elde edilen sonuçları özetlemektedir. Tablodan da görülebileceği gibi, RMSE ve MAE değerleri elde edilen LSTM modelinin yüksek bir başarı elde ettiğini göstermektedir. Bu değerlere dayanarak doğruluk değeri oldukça yüksektir. Yapılan çalışmadaki doğruluk ve hata oranları aynı zamanda benzer bir çalışma sergileyen Pengfei Yu ve Xuesong'un yaptığı çalışma sonuçlarıyla karşılaştırılmıştır. Pengfei Yu çalışmasında aynı şekilde 5 farklı hisse senedi üzerinde çalışma yapmıştır (Yu & Yan, 2020). 5 hisse senedinin de başarı ve hata oranları birbirine yakındır. Yaptıkları çalışmada 5 hisse senedinin de doğruluk oranları %60'ın altında kalıyor. Hata oranları ise bu tez çalışmasına ait LSTM modelinin hata oranlarından oldukça yüksektir. Tablo 4'de görüldüğü üzere bu yapılan tez çalışmasının daha yüksek bir başarı oranına sahip olduğu görülmektedir.

Tablo 4. Veri Setinden Elde Edilen Doğruluk Oranlarının Özeti

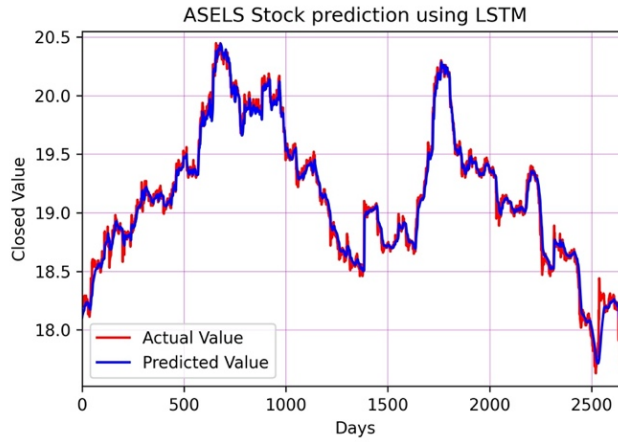
Şirket	Doğruluk (%)	RMSE	MAE
THY	97.52	0.0068	0.034
AKBANK	98.91	0.0078	0.016
ASELSAN	98.02	0.0050	0.047
GARANTI	98.35	0.024	0.033
ARCELİK	97.55	0.0048	0.043

Şirket	Doğruluk (%)	RMSE	MAPE (%)
S&P 500	58.07	7.92	1.30
DJIA	57.79	5.88	0.57
N 225	55.54	5.60	0.83
CSI 300	56.13	5.92	0.93
HSI	56.78	5.25	0.48

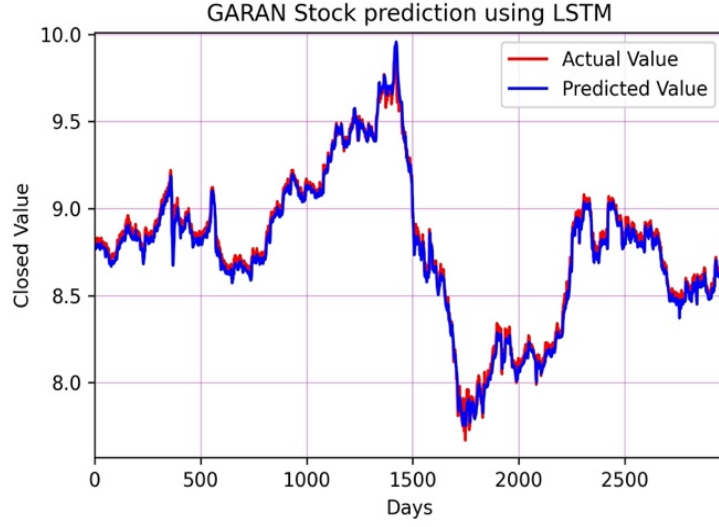
Her bir hisse senedi için grafiksel sonuçlar Şekil 41, Şekil 42, Şekil 43, Şekil 44 ve Şekil 45’de verilmiştir. Oluşturulan grafik, yakın fiyatın gerçek değerinin tahmini değere oranını göstermektedir. Tahmin edilen değerler gerçek değerlere çok yakındır. Her hisse senedinde yapılan çalışmada, gerçek değere çok yakın değerler elde edilmiştir.



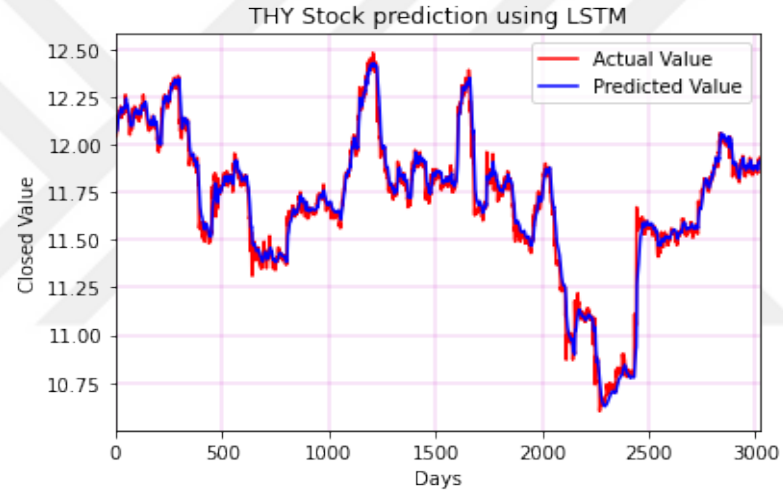
Şekil 41. LSTM kullanarak AKBNK Hisse Senedi Tahmini



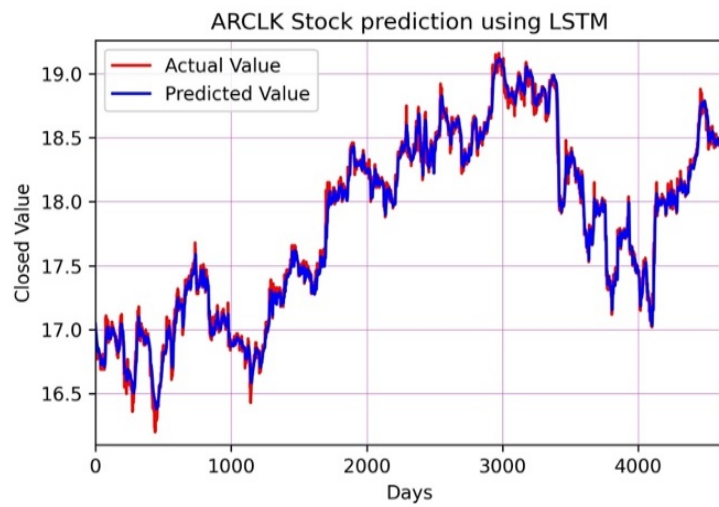
Şekil 42. LSTM kullanarak ASELS Hisse Senedi Tahmini



Şekil 43. LSTM kullanarak GARAN Hisse Senedi Tahmini



Şekil 44. LSTM kullanarak THYAO Hisse Senedi Tahmini



Şekil 45. LSTM kullanarak ARCLK Hisse Senedi Tahmini

ALTINCI BÖLÜM

TARTIŞMA VE DEĞERLENDİRME

Borsa tahmin edilmesi oldukça bir zor alandır. Borsa fiyatı bir ülkenin ekonomik büyümesinin en önemli göstergelerinden biridir. Bu nedenle hisse senedi piyasası fiyatının kesin hareketlerini belirlemek oldukça önemlidir. Geçmiş verilerin bilinmesinden ziyade siyasi, ekonomik, küresel anlamda yaşanan olaylar borsayı etkileyen faktörlerden bazılarıdır. Bununla birlikte, borsadaki karmaşık ve belirsiz davranışlar kesin belirlemeyi imkânsız kılar ve dolayısıyla yatırımcıların finansal karar alma süreci için güçlü tahmin modelleri derinden istenir. Borsa tahmini için temel ve teknik analiz kullanılmaktadır. Yapılan çalışmalar teknik analize ışık tutmaktadır. Yapılan tez çalışmasında iki farklı veri setiyle çalışılan iki çalışma grubu bulunmaktadır. İlki makine öğrenimi algoritmalarıyla yapılan çalışma diğeri makine öğreniminin alt dalı olan derin öğrenme yöntemiyle yapılan çalışmadır. Yapılan tüm çalışmalar tek tek ele alınmıştır. Elde edilen sonuçlar oldukça umut vericidir.

Borsa tahmini birçok ML algoritmasına açıktır. Aynı zamanda, zaman serisi verileri nedeniyle uygulanabilecek yüksek başarıya sahip modeller vardır. Bu çalışmada, Hindistan piyasasına ait TATAGLOBAL veri setine üç önemli makine öğrenimi algoritması uygulandı. Sonuçlardan görüldüğü üzere başarı oranları yüksektir. Hata geri yayımlı ileri beslemeli ağ kullanan çok katmanlı algılayıcı (MLP) gibi bu YSA modeller Python, C# gibi programlama dilleri kullanılarak prototipler halinde geliştirilebilir. YSA araçları, tahmin kapasitesine sahip olmalarını sağlamak için eğitim için önemli verilere ihtiyaç duyar. YSA ve ARIMA modellerinin bu konuda k-NN algoritmasına göre daha başarılı olduğunu söyleyebiliriz. Çalışmaların yatırımcılara ışık tutacağı umuluyor. Çalışmanın sonuçları karşılaştırılarak diğeri ML algoritmaları ile değerlendirilmesi, başka bir araştırma konusu olarak önerilebilir.

Son yıllarda, derin öğrenmeye dayalı yöntemlerin karmaşık sorunları çözebilecekleri için benimsenmiş olduğu görülmüştür. Derin öğrenme ağları daha büyük ve daha derin olduğundan, yeterli verilerin doğruluk oranını artırması daha olasıdır. Oluşturduğumuz LSTM modeli Borsa İstanbul'da işlem gören beş popüler hisse senedinin kapanış fiyatını tahmin etmek için bir teklif sunmaktadır. Her şeyden önce veri kümesine kapanış değerine göre hesaplanan WMA endeksi eklendi. Daha sonra eğilim değeri, önceki kapanış değerinin geçerli satırdaki yakını değerden çıkarılmasıyla

hesaplanır. Oluşturulan veri seti eğitim ve test olarak ayrıldıktan sonra, min-maks ölçekleyici normelleştirme işlemi ayrı olarak uygulandı ve LSTM modeli için uygun hale getirildi. Tahmin, yukarıda önerilen model yapısı oluşturularak yapılmıştır. LSTM'nin Tablo 3'te belirtilen RMSE ve MAE değerleri doğrultusunda başarılı sonuçlar verdiği görülmektedir. LSTM tabanlı modelin daha az risk sunduğu sonucuna varmak mümkündür. Sinir ağı mimarisindeki değişiklikleri ve girdi verilerinin ön-işlenmesine yönelik farklı yaklaşımları inceleyerek ve yeni farklı özellikler ekleyerek oluşturulan modeli ve tahminleri geliştirmenin yolları araştırma konusu olabilir.



KAYNAKÇA

- Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Stock price prediction using the ARIMA model. *Proceedings - UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, UKSim 2014*, 106–112.
<https://doi.org/10.1109/UKSim.2014.67>
- Alkhatib, K., Najadat, H., Hmeidi, I., & Shatnawi, M. K. A. (2013). Stock Price Prediction Using K-Nearest Neighbor Algorithm. *International Journal of Business, Humanities and Technology*, 3(3), 32–44.
- Almasarweh, M., & Wadi, S. AL. (2018). ARIMA Model in Predicting Banking Stock Market Data. *Modern Applied Science*, 12(11), 309.
<https://doi.org/10.5539/mas.v12n11p309>
- Aparicio, T., Pozo, E., & Saura, D. (2002). The nearest neighbour method as a test for detecting complex dynamics in financial series. An empirical application. *Applied Financial Economics*, 12(7), 517–525.
<https://doi.org/10.1080/09603100010007986>
- Aurangzeb, D. (2012). Factors Affecting Performance of Stock Market : Evidence from South Asian Countries. *International Journal of Academic Research in Business and Social Sciences*, 2(9), 1–15.
- Baek, Y., & Kim, H. Y. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications*, 113, 457–480.
<https://doi.org/10.1016/j.eswa.2018.07.019>
- Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE*, 12(7). <https://doi.org/10.1371/journal.pone.0180944>
- Bing, Y., Hao, J. K., & Zhang, S. C. (2012). Stock Market Prediction Using Artificial Neural Networks. *Advanced Engineering Forum*, 6–7, 1055–1060.
<https://doi.org/10.4028/www.scientific.net/aef.6-7.1055>

- Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015*, 2823–2824. <https://doi.org/10.1109/BigData.2015.7364089>
- El-Masry, A. M., Ghaly, M. F., Khalafallah, M. A., & El-Fayed, Y. A. (2002). Deep Learning for Event-Driven Stock Prediction Xiao. *Journal of Scientific and Industrial Research*, 61(9), 719–725.
- Fatima, S., & Hussain, G. (2008). Statistical models of KSE100 index using hybrid financial systems. *Neurocomputing*, 71(13–15), 2742–2746. <https://doi.org/10.1016/j.neucom.2007.11.044>
- Fernández-Rodríguez, F., Sosvilla-Rivero, S., & Andrada-Félix, J. (1999). Exchange-rate forecasts with simultaneous nearest-neighbour methods: Evidence from the EMS. *International Journal of Forecasting*, 15(4), 383–392. [https://doi.org/10.1016/S0169-2070\(99\)00003-5](https://doi.org/10.1016/S0169-2070(99)00003-5)
- Fischer, T., & Krauss, C. (2017). Networks for Financial Market Predictions. *FAU Discussion Papers in Economics, No. 11/2017, Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics, Erlangen*, 1–34.
- Jiahong Li, Hui Bu, & Junjie Wu. (2017). Sentiment-aware stock market prediction: A deep learning method. *2017 International Conference on Service Systems and Service Management*, 1–6. <https://doi.org/10.1109/ICSSSM.2017.7996306>
- Liu, J., Chao, F., Lin, Y.-C., & Lin, C.-M. (2019). *Stock Prices Prediction using Deep Learning Models*. <http://arxiv.org/abs/1909.12227>
- Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). Predicción del índice del mercado bursátil utilizando una red neuronal artificial. *Journal of Economics, Finance and Administrative Science*, 21(41), 89–93. <https://doi.org/10.1016/j.jefas.2016.07.002>
- Mu, A. (2019). 濟無No Title No Title. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9). <https://doi.org/10.1017/CBO9781107415324.004>

- Nelson, D. M. Q., Pereira, A. C. M., & De Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. *Proceedings of the International Joint Conference on Neural Networks, 2017-May(Dcc)*, 1419–1426. <https://doi.org/10.1109/IJCNN.2017.7966019>
- Patel, M. B., & Yalamalle, S. R. (2014). *Stock Price Prediction Using Artificial*. 3(6), 13755–13762.
- Roondiwala, M., Patel, H., & Varma, S. (2015). Predicting Stock Prices Using LSTM. *International Journal of Science and Research*, 6(4), 2319–7064. <https://www.quandl.com/data/NSE>
- Sadia, K. H., Sharma, A., Paul, A., Padhi, S., & Sanyal, S. (2019). Stock market prediction using machine learning algorithms. *International Journal of Recent Technology and Engineering*, 8(4), 280–283.
- Shastri, M., Roy, S., & Mittal, M. (2018). Stock Price Prediction using Artificial Neural Model: An Application of Big Data. *ICST Transactions on Scalable Information Systems*, 0(0), 156085. <https://doi.org/10.4108/eai.19-12-2018.156085>
- Sonawane, D. (2015). Prediction Using Back Propagation and k- Nearest Neighbor (k-NN) Algorithm. *International Journal of Innovative Reserach in Computer and Communication Engineering*, 3(4), 3209–3213.
- Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. *13th Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012, 1*, 194–197.
- Tanuwijaya, J., & Hansun, S. (2019). LQ45 stock index prediction using k-nearest neighbors regression. *International Journal of Recent Technology and Engineering*, 8(3), 2388–2391. <https://doi.org/10.35940/ijrte.C4663.098319>
- Vainionpää, I., & Davidsson, S. (2014). *Stock market prediction using the K Nearest Neighbours algorithm and a comparison with the moving average formula*. April.

Wadi, S. AL, Almasarweh, M., & Alsaraireh, A. A. (2018). Predicting Closed Price Time Series Data Using ARIMA Model. *Modern Applied Science*, 12(11), 181. <https://doi.org/10.5539/mas.v12n11p181>

Wanjawa, B. W., & Muchemi, L. (2014). *ANN Model to Predict Stock Prices at Stock Exchange Markets*. 1–23. <http://arxiv.org/abs/1502.06434>

Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks. *Neural Computing and Applications*, 32(6), 1609–1628. <https://doi.org/10.1007/s00521-019-04212-x>



ÖZGEÇMİŞ

B. EĞİTİM

Yüksek Lisans: İstanbul Sabahattin Zaim Üniversitesi Bilgisayar Bilimleri ve Mühendisliği (%30 İngilizce)

Lisans: İstanbul Sabahattin Zaim Üniversitesi Bilgisayar Mühendisliği (%100 Burslu)

C. MESLEKİ DENEYİM

2017 Bursa Büyükşehir Belediyesi Bilgi İşlem Daire Başkanlığı (Staj)

2018 ULUTEK Serim Yönetim Sistemleri Şirketi (3 ay)

2018 – Devam Ediyor İTÜ Teknopark Matriks Bilgi Dağıtım Hizmetleri A.Ş. iOS Developer

D. BİLDİRİLER ve YAYINLAR

Bildiriler: 1- Machine Learning Methods for Stock Market Analysis- ICONDATA Konferansı

2- Stock Market Forecasting Using Deep Learning Method- ICONDATA Konferansı

Yayınlar: 1- Stock Market Value Prediction Using Deep Learning- Data Science and Applications Journal