



EGE ÜNİVERSİTESİ

LİSANS TEZİ

**BORSA ENDEKSİ HAREKETLERİNİN MAKİNE
ÖĞRENMESİ İLE ANALİZİ**

Berkay ÇETİN, Cemil CESUR, Ersin ELMAS

Tez Danışmanı : Prof. Dr. Aybars UĞUR

Bilgisayar Mühendisliği Anabilim Dalı

Sunuş Tarihi : 23.06.2023

**Bornova - İZMİR
2023**

EGE ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ

(LİSANS TEZİ)

**BORSA ENDEKSİ HAREKETLERİNİN MAKİNE
ÖĞRENMESİ İLE ANALİZİ**

Berkay ÇETİN, Cemil CESUR, Ersin ELMAS

Tez Danışmanı : Prof. Dr. Aybars UĞUR

Bilgisayar Mühendisliği Anabilim Dalı

Sunuş Tarihi : 23.06.2023

**Bornova-İZMİR
2023**

ÖZET

BORSA ENDEKSİ HAREKETLERİNİN MAKİNE ÖĞRENMESİ İLE ANALİZİ

CESUR, Cemil

ÇETİN, Berkay

ELMAS, Ersin

Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof.Dr. Aybars UĞUR

Haziran 2023

Gelişen teknolojiyle birlikte bilgisayar kullanımı, onun insanlardan üstün özelliklerinin varlığı sayesinde gün geçtikçe artmaktadır. Bilgisayarların hesaplamadaki kesinlik, hız, bellek, doğruluk vb. alanlardaki üstünlükleri sayesinde işlerimiz kolaylaşmakta, insanların kendi yapacağı işlerin otomatik ve daha kolay olarak gerçekleştirilmesi sağlanmaktadır. Geliştirmiş olduğumuz projede de bu istenen işlerin kolaylaştırılması ve daha tutarlı olmasına yardımcı olmak amaçlanmıştır. Projede amacımız, finansal zaman serilerinin doğrusal olmayan dinamik yapılarına, öngörülemeyen, düzensiz hareketlerine ve diğer birçok çevresel etmene karşı makine öğrenimi algoritmalarının kullanımıyla birlikte finansal zaman serilerinin tahminini yapmayı, yatırımcı veya yatırımcıların önünü görmesini ve elde edilen sonuçlara göre yatırım yapılacak doğru alanı bulmasına yardımcı olmaktır. Bu sayede yatırımcıların daha az risk alarak birikimlerini daha doğru bir şekilde kullanmaları sağlanmak istenmektedir.

Anahtar Kelimeler: Makine Öğrenmesi, Yapay Zeka, Derin Öğrenme, Veri Madenciliği, Yapay Sinir Ağları, Uzun Kısa Süreli Bellek, Destek Vektör Makineleri, Naive Bayes, Borsa Endeks Tahmini, Zaman Serileri, Python, numpy, pandas, math, yfinance, MinMaxScaler

ABSTRACT

ANALYSIS OF STOCK INDEX MOVEMENTS WITH MACHINE LEARNING

CESUR, Cemil
ÇETİN, Berkay
ELMAS, Ersin

Bachelor in Computer Eng.
Supervisor: Prof. Dr. Aybars UĞUR
June 2023

With the developing technology, the use of computers is increasing day by day thanks to its superior features over humans. Thanks to the superiority of computers in areas such as precision in calculation, speed, memory, accuracy, etc., our jobs are made easier, and the jobs that people can do themselves can be performed automatically and more easily. In the project we have developed, it is aimed to facilitate these desired jobs and help them to be more consistent. Our aim in the project is to predict financial time series with the use of machine learning algorithms against the non-linear dynamic structures of financial time series, unpredictable, irregular movements and many other environmental factors, to help investors or investors see the way ahead and to help them find the right area to invest according to the results obtained. In this way, it is aimed to enable investors to use their savings more accurately by taking less risk.

Keywords: Machine Learning, Artificial Intelligence, Deep Learning, Data Mining, Artificial Neural Networks, Long Short-Term Memory, Support Vector Machines, Naive Bayes, Stock Index Forecasting, Time Series, Python, numpy, pandas, math, yfinance, MinMaxScaler

TEŞEKKÜR

Tez çalışmamızın başarılı bir şekilde tamamlanmasında, bize değerli rehberliklerini sunan saygıdeğer danışmanımız Prof. Dr. Aybars UĞUR'a teşekkürlerimizi sunmak istiyoruz. Sizin değerli yönlendirmeleriniz, vizyoner bakış açınız ve profesyonel rehberliğiniz ile tezimizi başarıyla tamamlama fırsatı bulduk. Sizden gelen öneri ve geri bildirimler, çalışmamızdaki eksiklikleri giderme ve kalitesini artırma konusunda bize büyük bir ilham kaynağı oldu.

İÇİNDEKİLER

<u>ÖZET</u>	i
<u>ABSTRACT</u>	ii
<u>TEŞEKKÜR</u>	iii
<u>ŞEKİLLER DİZİNİ</u>	vi
<u>TABLolar DİZİNİ</u>	vii
<u>1. GİRİŞ</u>	1
<u>2. YAPAY ZEKA</u>	2
<u>2.1. Yapay Zeka Nedir?</u>	2
<u>2.2. Makine Öğrenmesi</u>	3
<u>2.3. Derin Öğrenme</u>	4
<u>2.4. Yapay Sinir Ağları</u>	5
<u>2.5. Yapay Zeka ve Borsa</u>	5
<u>3. BORSA</u>	6
<u>4. VERİ MADENCİLİĞİ</u>	7
<u>5. LİTERATÜR ÇALIŞMASI</u>	8
<u>6. YÖNTEM VE TEKNOLOJİLER</u>	11
<u>6.1. Sliding Window Tekniği</u>	11
<u>6.2. LSTM (Uzun Kısa Süreli Bellek)</u>	12
<u>6.3. Naive Bayes</u>	18
<u>6.4. Destek Vektör Makineleri (DVM)</u>	18
<u>6.5. ARIMA</u>	19
<u>6.5.1. ARIMA Kullanım Alanları</u>	20
<u>6.6. Bidirectional LSTM</u>	20
<u>6.7. Yapay Sinir Ağları</u>	21
<u>6.8. Kullanılan Dil, Ortam ve Kütüphaneler</u>	21
<u>7. GERÇEKLEŞTİRİM</u>	24
<u>7.1 Analiz</u>	24
<u>7.2 Tasarım</u>	24
<u>7.3 DVM Gerçekleştirimi</u>	25
<u>7.4 Naive Bayes Gerçekleştirimi</u>	26
<u>7.5 Stacked LSTM Gerçekleştirimi</u>	28
<u>7.6 Bidirectional LSTM Gerçekleştirimi</u>	30

<u>7.7 ARIMA Gerçekleştirimi</u>	30
<u>7.8 YSA Gerçekleştirimi</u>	31
<u>7.9 Modellerin Karşılaştırılması</u>	33
<u>8. SONUÇ VE ÇIKARIMLAR</u>	36
<u>KAYNAKÇA</u>	37

ŞEKİLLER DİZİNİ

<u>Şekil 2.1 Yapay Zeka ve Kapsama Alanı [22]</u>	2
<u>Şekil 2.2 Makine Öğrenmesine Genel Bir Bakış [10]</u>	3
<u>Şekil 2.3 Yapay Sinir Ağı Örneği [23]</u>	5
<u>Şekil 4.1 Veri Madenciliği ve Disiplinler [24]</u>	7
<u>Şekil 6.1 Sliding Window Tekniği [27]</u>	11
<u>Şekil 6.2 Bir LSTM'nin iç yapısı [28]</u>	13
<u>Şekil 6.3 THYAO 1 Yıllık Verisi</u>	14
<u>Şekil 6.4 Veri Ölçeklendirme</u>	14
<u>Şekil 6.5 Eğitim ve Test Verisinin Ayrılması</u>	15
<u>Şekil 6.6 Sliding Window Yöntemi ile Veri Seti Oluşturma</u>	15
<u>Şekil 6.7 Modelin Eğitimi</u>	16
<u>Şekil 6.8 Tahminleme ve Hata Oranları</u>	16
<u>Şekil 6.9 Genel Sonuç Grafiği</u>	17
<u>Şekil 6.10 Sonraki Gün Tahmini</u>	17
<u>Şekil 6.11 Bidirectional LSTM Modeli [29]</u>	21
<u>Şekil 7.1 Hisse Senedi Seçim Ekranı</u>	24
<u>Şekil 7.2 Veri Seti Ekran Görüntüsü</u>	25
<u>Şekil 7.3 DVM Çıktısı</u>	26
<u>Şekil 7.4 Veri Seti Ekran Görüntüsü</u>	27
<u>Şekil 7.5 Naive Bayes Çıktısı</u>	28
<u>Şekil 7.6 PETKM.IS Fiyat Kapanış Grafiği</u>	29
<u>Şekil 7.7 LSTM Çıktısı</u>	29
<u>Şekil 7.8 Bidirectional LSTM Çıktısı</u>	30
<u>Şekil 7.9 ARIMA Çıktısı</u>	31
<u>Şekil 7.10 KCHOL.IS Hissesinin Kapanış Değer Grafiği</u>	32
<u>Şekil 7.11 KCHOL.IS Hissesinin Tahmin Grafiği</u>	33
<u>Şekil 7.12 Performans Metrikleri</u>	33

TABLÖLER DİZİNİ

<u>Tablo 6.1 Kullanılan Ortam Gereksinimleri</u>	21
<u>Tablo 7.1 Literatür ile Karşılaştırma</u>	34
<u>Tablo 7.2 Modeller Arası Performans Karşılaştırılması</u>	35
<u>Tablo 7.3 Modeller Arası Yönlendirme Performansının Karşılaştırılması</u>	35

KISALTMALAR DİZİNİ

Kısaltmalar

ANN - Artificial Neural Networks

YSA - Yapay Sinir Ağları

LSTM - Long Short-Term Memory

SVM - Support Vector Machines

DVM - Destek Vektör Makineleri

NB - Naive Bayes

1. GİRİŞ

Bu tez projesi, doğrusal olmayan dinamik yapılara ve öngörülemeyen, düzensiz hareketlere sahip olan finansal zaman serilerinin tahmin yapmayı zorlaştıran doğasını yatırımcılar için basitleştirmeyi amaçlamaktadır. Finansal zaman serileri üzerindeki birçok faktör, borsa endeksleri üzerinde etkili olabilir. Bu nedenle doğru gelecek tahminleri ve borsa endeksi hareketleri üretmek için finansal zaman serilerinin karmaşık yapısını anlayıp bu yapıya dayalı tahminler yapıp, makine öğrenimi algoritmalarını kullanarak yatırımcılara yardımcı olmak, finansal verileri analiz eden ve tahminler yapan bir sistem geliştirmek hedeflenmektedir.

Proje kapsamında, finansal zaman serilerinin tahmini için kullanılabilecek farklı makine öğrenimi algoritmaları incelenmektedir. Bu algoritmalar arasında özellikle Yapay Sinir Ağları (ANN), Uzun Kısa Süreli Bellek (LSTM), Destek Vektör Makineleri (SVM) ve Naive Bayes gibi yöntemler üzerinde durulmaktadır. Ayrıca, finansal veri ön işleme yöntemleri, veri boyutunu azaltma teknikleri ve model performansının değerlendirilmesi gibi konular da proje kapsamında ele alınmaktadır.

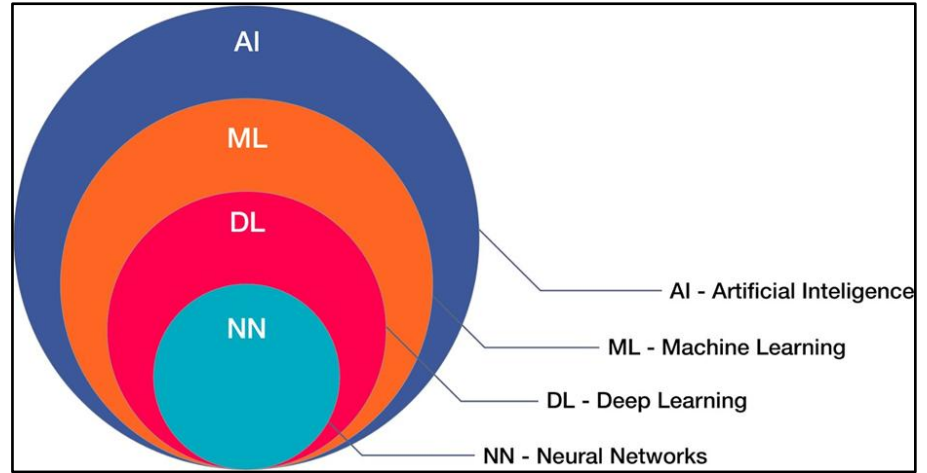
Bu çalışmanın hedefi, finansal zaman serilerinin tahmin edilmesinde kullanılan makine öğrenimi algoritmalarının performanslarını karşılaştırmak ve en iyi sonucu veren algoritmanın belirlenmesidir. Bu çalışmanın önemi ise finansal piyasalarda doğru tahminlerin yapılması için makine öğrenimi yöntemlerinin kullanımının giderek artmasıdır. Bu yöntemler, daha doğru finansal zaman serisi projeksiyonları üretmek için kullanılabilir ve yatırımcıların ve şirketlerin alacakları kararlarda önemli bir yardımcı olabilir. Ayrıca, bu çalışmanın, makine öğrenimi algoritmalarının finansal zaman serileri tahminlerindeki kullanımını daha iyi anlamak ve finansal piyasalardaki belirsizliği azaltmak için önemli bir katkı sağlaması hedeflenmektedir.

2. YAPAY ZEKA

2.1. Yapay Zeka Nedir?

Bir bilgisayar veya bilgisayar destekli makinenin insana özgü nitelikler, problem çözme, anlama, anlam çıkarma, genelleme yapma, geçmiş deneyimlerden öğrenme gibi daha yüksek mantıksal süreçlerle ilgili işlemleri yürütme yeteneği yapay zeka olarak bilinir (Nabiyev, 2012).

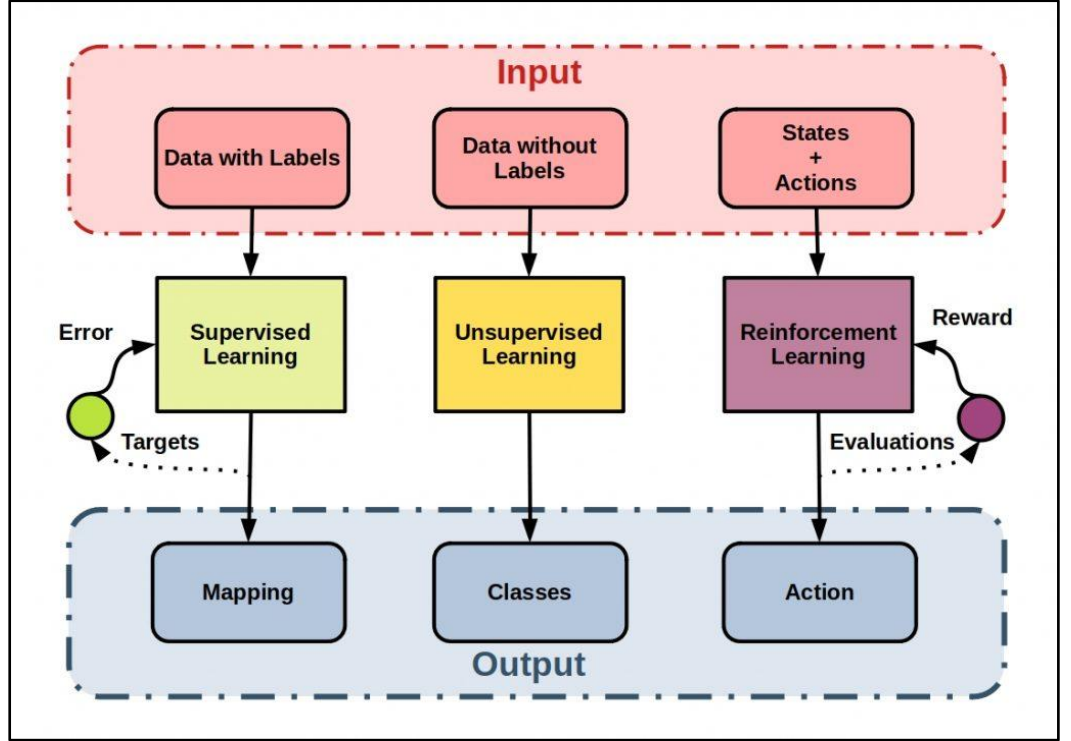
Slage'e göre sezgisel programlama, bir metodoloji olarak yapay zekanın temelidir (Andrew, 1991). Popov, yapay zekayı insan davranışını taklit eden makineler yapma bilimi olarak tanımlar (Popov, 1990). Axe, yapay zekanın akıllı yazılımlara odaklanan bir bilim olduğunu iddia etmektedir (Copelan, 1993). Yapay zeka, Genesereth ve Nilsson'un sözleriyle, akıllı davranışın incelenmesidir. Şekil 2.1'de yapay zekanın kapsama alanları görülebilmektedir. [1]



Şekil 2.1 Yapay Zeka ve Kapsama Alanı [22]

2.2. Makine Öğrenmesi

Matematik ve istatistik kullanılarak veriler üzerinde gerçekleştirilen işlemlerden sonuçlar çıkararak tahminler üreten sistemlerin modellenmesi makine öğrenimi olarak bilinmektedir.[8] Şekil 2.2’de makine öğrenimine genel bir bakış verilmiştir.



Şekil 2.2 Makine Öğrenmesine Genel Bir Bakış [10]

Algoritma ve halihazırda var olan veri seti kullanılarak model oluşturulur. Modellerden en iyi performansı elde etmek için makine öğrenimi kullanılır ve öğrenme türleri üç gruba ayrılır. Bunlar:

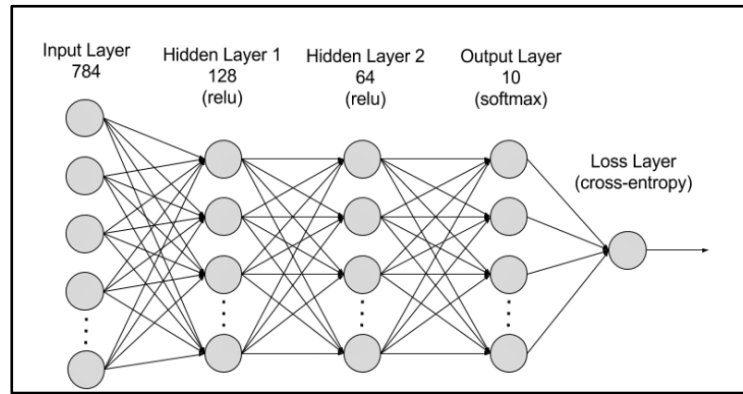
- **Denetimli Öğrenme:** Bu tür öğrenmede, girdileri çıktılara eşleyen fonksiyonu üretmek için regresyon veya sınıflandırma teknikleri kullanılır. Hem girdiler hem de çıktılar eğitim verilerine dahil edilir.
- **Denetimsiz Öğrenme:** Bu öğrenme tekniği kullanılarak bilinmeyen bir yapı tahmin edilir. Giriş verisinin sınıfı belirsizdir.
- **Takviyeli Öğrenme:** Bu öğrenme tekniğinde girdilerin çevresiyle etkileşime girerek ödül olarak bilinen geri bildirimi en üst düzeye çıkararak en iyi eylem yolu belirlenmeye çalışılır. Bu öğrenme türü diğer öğrenme türlerinden bu yönüyle ayrılır.[9]

2.3. Derin Öğrenme

Derin Öğrenme, Yapay Sinir Ağları olarak bilinen beynin yapısından ve işlemlerinden ilham alan bir makine öğrenimi algoritması biçimidir. Biyolojik nöronlar gibi yapay nöronlar, daha sonra toplanan, işlenen ve çıktılarına aktarılan giriş sinyallerini alır. Özellikle internetin gelişmesiyle birlikte dijital dünyada her saniye milyonlarca veri büyük hacimlerde üretilmekte ve depolanmaktadır. Eğitim için 5 kullanılan veri miktarı arttıkça, geleneksel makine öğrenimi algoritmalarının performansı büyük verilerle başa çıkmakta yetersiz kalmıştır. Makine öğrenimi algoritmaları hızlı çalışıp hızla gelişse de günümüzde internetin yaygınlaşmasıyla birlikte erişilebilen veri miktarı arttığından, hızlı bir sonuçtan çok daha yüksek bir performans değerine sahip olmak daha önemli hale gelmiştir. Derin öğrenme sistemleri ile bu büyük veriler etkili bir şekilde kullanılmakta ve verimli sonuçlar elde edilebilmektedir.[2] Diğer alanlardan farklı olarak, Derin öğrenme sınıflandırma özelliklerini ve görevlerini doğrudan veriden gerçekleştirmeyi öğrenir. Bazı modellerinde ilgili özelliklerin çıkarılmasına ihtiyaç duyulmadan doğrudan eğitim yapabilme olanağı bulunmaktadır. Derin öğrenme modelleri, insan düzeyinde öğrenmeyi başarmak ve hatta bu seviyeyi aşmak için farklı ve çok sayıda çok sinir ağı mimarisi kullanılarak eğitilebilir. Ayrıca Derin ağlar, sistemin yüksek kaliteli GPU'lar kullanarak büyük verilerle eğitilmesine izin verir. Bu sebepten derin öğrenme için üst düzey donanıma ihtiyaç vardır.

2.4. Yapay Sinir Ağları

Yapay sinir ağları (YSA), insan beyninin öğrenme yolunu taklit ederek beynin öğrenme, hatırlama, genelleme yapma yolu ile topladığı verilerden yeni veri üretebilme gibi temel işlevlerin gerçekleştirildiği bilgisayar yazılımlarıdır. Yapay sinir ağları; insan beyninden esinlenerek, öğrenme sürecinin matematiksel olarak modellenmesi uğraşı sonucu ortaya çıkmıştır (Kabalcı, 2014). Yapay sinir ağları, paralel dağıtılmış ağlar, bağlantılı ağlar, nuromorfik ağlar gibi adlarla da tanımlanmaktadır (Keskenler , 2017). [3] Şekil 2.3’de yapay sinir ağı örneği görülmektedir.



Şekil 2.3 Yapay Sinir Ağı Örneği [23]

2.5. Yapay Zeka ve Borsa

Finansal yatırım, yapay zeka uygulamalarının en sık uygulandığı alanlardandır. Chase Manhattan Bank (Marose, 1990), Peat Marwick, American Express (Zahedi, 1993), finansal yatırım problemlerinin çözümünde yapay zekayı etkili bir şekilde kullanan birçok şirketten sadece birkaçıdır. Yapay zekanın finasta en çok kullanıldığı uygulamalar, hisse senedi performansının tahmini ve hisse senedi piyasasında ticaret yapmak için hisse senetlerinin seçimidir. [4]

3. BORSA

Borsa, hisse senedi ve devlet tahvili, emtia, döviz, vadeli işlem ve opsiyon sözleşmeleri gibi varlıkların alınıp satıldığı, halka arz edildiği ve bu malların güvenli bir şekilde alınıp satılabildiği yapılandırılmış piyasadır. Tüm bunları düşündüğümüzde aklımıza ilk olarak ülkemizin organize borsası Borsa İstanbul (BİST) gelmektedir. Hisse senedi piyasaları, sermayenin değiş tokuş edildiği, şirketlerin ucuz fon bulabileceği ve yatırımcıların küçük birikimlerle bile şirket kazançlarına ortak olabildikleri bir pazardır. Ayrıca borsalar, sermayenin tabana yayılması, adil gelir dağılımı, kayıt dışı ekonomiyle mücadele gibi sosyo-ekonomik zorlukların yanı sıra finansal araçlar aracılığıyla reel sektöre uzun vadeli kaynak sağlamaktadır. [5] Ekonomiye kaynak ve likidite sağlama, güvenilirlik şeffaflık gibi misyonları da vardır. Sermaye Piyasası Kanunları, borsada işlem gören şirketlerin kotasyonu, halka arzları, piyasa işlem koşulları ve borsanın işleyişini düzenler. Yatırımcılar, borsada işlem gören şirketlerin kanunen yayınlamakla yükümlü olduğu mali tablolar ve diğer şirket bilgileri ile borsa fiyat ve bilgilerine, BİST'in yayımladığı basın ve dijital yayın organları aracılığıyla kolaylıkla ulaşabilirler.

Borsa İstanbul, Türkiye'de borsacılık faaliyetlerini yürütmek için kurulmuş bir piyasalardır. Borsada işlem gören şirketlerin kotasyonu, halka arzları, piyasa işlem koşulları ve borsanın işleyişi, Sermaye Piyasası Kanunları tarafından düzenlenmektedir.[6] Yatırımcılar, borsada işlem gören şirketlerin kanunen yayınlamakla yükümlü olduğu mali tablolar ve diğer şirket bilgileri ile borsa fiyat ve bilgilerine BİST'in yayımladığı basın ve dijital yayın organları aracılığıyla kolaylıkla ulaşabilirler. Bu sayede, yatırımcılar, borsada işlem gören şirketlerin finansal durumlarına ilişkin doğru bilgiye erişebilir ve yatırım kararlarını bu bilgilere göre verebilirler.

Borsalarda işlem gören yatırım araçları, belirli şirketlerin hisse senetleri, endeksler, emtialar, dövizler ve diğer finansal araçlar olabilir. Her borsanın farklı işleyiş yapısı ve kuralları vardır. Örneğin, Amerikan borsalarının işleyişi ile Türk borsalarının işleyişi ve takvimleri birbirinden farklıdır. [7] Şirketler, borsada listelenerek potansiyel yatırımcılara ulaşmak ve finansal kaynak sağlamak için halka arz edilebilirler

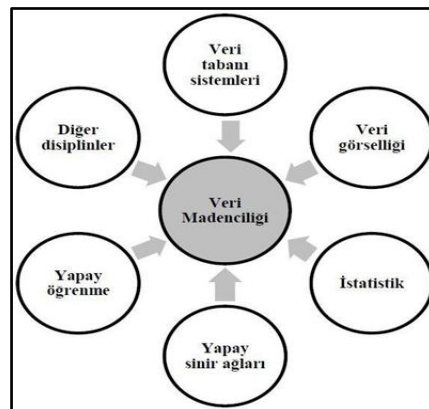
4. VERİ MADENCİLİĞİ

Bilgisayar tarafından üretilen veri, anlamlandırılması amacıyla belirli bir hedefe yönelik olarak işlenmektedir. Bilgi kavramı, verilerin bu amaç doğrultusunda işlenmesiyle gelişir. Bilgiye dayalı hareket etmek, kayıpları önlememize yardımcı olabilir ve gelecekte yapılan durumsal tahminlere dayalı olarak harekete geçme yeteneği sağlayabilir. İşte bu noktada büyük miktarlardaki verileri işleyebilen, ham veriyi bilgiye dönüştüren ve anlamlı hale getiren tekniklerden yararlanmak gerekir ve bunların başında veri madenciliği gelmektedir. Veri madenciliği, geleceğe yönelik tahmin analizlerine ve aramalara yardımcı olmak amacıyla büyük miktarlardaki verilerden ilgili ve pratik ilişkileri ve örüntüleri çıkarmak için bilgisayar sistemlerinin kullanılması sürecidir.[8]

Öte yandan veri madenciliği, bilgi keşfi sürecinin bir adımı olarak kabul edilmektedir. Aşağıda bilgi keşfi sürecindeki adımlar listelenmiştir.

1. Veri temizleme (gürültülü ve tutarsız verilerin giderilmesi)
2. Veri Entegrasyonu (birçok veri kaynağının birleştirilmesi)
3. Veri Seçimi (yapılacak analize ilişkin verilerin belirlenmesi)
4. Veri Dönüşümü (verinin veri madenciliği tekniklerinden yararlanılabilecek forma dönüşümünü gerçekleştirmek)
5. Veri madenciliği (veri kalıplarını yakalamak için akıllı yöntemlerin uygulanması)
6. Örüntü Değerlendirme (bazı ölçümlere göre elde edilen bilgileri temsil eden ilginç örüntülerin tanımlanması)
7. Bilgi Görüntüleme (çıkarılan bilginin kullanıcıya sunulması)

Veri madenciliği tekniklerinin tıp, eğitim, finans ve meteoroloji gibi birçok alanda bulunan veriler üzerinde uygulanmasıyla gelecekte olabilecek durumların tahminleri ve geçmiş verilerin sınıflandırmaları yapılabilmektedir. Bu sayede insanların ya da makinelerin istenilen alana yönelik sonuç, yorum çıkartmaları ve ileriye yönelik karar vermelerine önemli bir destek sağlanmaktadır.[9] Şekil 4.1’de veri madenciliği ve disiplinler şablonu görülebilmektedir.



Şekil 4.1 Veri Madenciliği ve Disiplinler [24]

5. LİTERATÜR ÇALIŞMASI

Bu bölümde hisse senedi gelecek fiyat tahmini konusunda daha önce yapılan çalışmalardan bazıları incelenmiş ve aşağıda yer aldığı şekilde özetlenmiştir.

ARSLANKAYA, S., & TOPRAK, Ş. (2021)'de Ereğli Demir ve Çelik Fabrikaları T. A. Ş.'nin hisse senedi fiyat projeksiyonlarını analiz etmek için makine öğrenme yöntemlerinden Polinom Regresyon ve Rastgele Orman Regresyon algoritmalarının yanı sıra derin öğrenme yöntemlerinden RNN ve LSTM algoritmaları kullanılmıştır. Algoritmalar Spyder (Anaconda3) editörü ve Python 3.7 programlama dili kullanılarak yazılmıştır. Araştırmanın veri tabanı, günlerin ve hisse senedi kapanış fiyatlarının yer aldığı bir Excel 2013 dosyasıdır. Toplamda 1619 veri bulunmaktadır. MSE, MAE ve RMSE hata ölçütleri kullanılarak tahmin verilerinin istatistiksel analizi yapılmıştır. Modeller hata seviyelerine göre sıralandığında sırasıyla Rastgele Orman Regresyon modeli, LSTM modeli, RNN modeli ve Polinom Regresyon modeli olduğu görülmüştür. Hisse senedi fiyatlarını etkileyebilecek ek borsa hareketlerinin entegre edilmesi ve veri miktarının artırılmasıyla, modellerin daha doğru ve güvenilir bulgular sunabileceği düşünülmektedir.[9]

AKŞEHİR, Z. D., & KILIÇ, E. (2019)'da, 1 Ocak 2016 ile 9 Mayıs 2019 tarihleri arasındaki veriler kullanarak çeşitli banka hisselerinin bir gün sonraki kapanış fiyatları tahmin edilmiştir. Tahmin modeli, hisse senedi hareketleri doğrusal modeller sergilediği için karar ağaçları, çoklu regresyon ve rastgele orman teknikleri kullanılarak oluşturulmuştur. Bu teknikler kullanılarak oluşturulan modellerde iki farklı girdi değişkeni seti kullanılmıştır. İlk veri setinde 46'sı teknik ve 4'ü temel olmak üzere toplam 50 gösterge bulunmaktadır. Yapılan indirgemeler sonucunda ikinci veri seti 29 teknik ve 4 temel göstergeden oluşmaktadır. Bu iki modelin performansları R^2 ölçütü ile değerlendirildiğinde teknik göstergelerde yapılan indirgemenin performansı olumlu yönde etkilediği görülmüştür. Çalışmanın sonucunda regresyon, karar ağaçları ve rastgele orman yöntemleri ile oluşturulan modellerin banka hisselerinin gelecek fiyatlarını tahminlemede oldukça başarılı olduğu görülmüştür.[10]

Şişmanoğlu, G., Koçer, F., ÖNDE, M. A., & SAHİNGOZ, O. K. (2020)'de, üç farklı derin öğrenme modeli -LSTM, GRU ve BLSTM- bir hisse senedi tahmin sistemi oluşturmak için kullanılmış ve modeller karşılaştırılmıştır. Spekülatif dalgalanmaları önlemek amacıyla veri seti olarak 1968-2018 yılları arasında New York Borsasından alınan hisse senetlerinin zaman serisi değerleri kullanılmıştır. Özellikle IBM hisse senedi kullanılarak testler gerçekleştirilmiştir. Deneyisel bulgulara göre, önerilen modeller arasında 5

günlük girdiler ile eğitilen BLSTM %63,54'lük yönsel doğruluk oranı ve 0,01'lik RMSE değeri ile en iyi sonucu vermiş, 10 günün sonunda %13,47 kar getirdiği tespit edilmiştir.[11]

DALKIRAN, İ., & Mehmet, O. Z. A. N. (2022)'de, Borsa İstanbul'da işlem gören ISCTR hisse senedi, uzun vadeli veriler incelenerek günlük değer tahmini yapmak için kullanılmıştır. Tahmini geliştirmek için ilgili hisse senedinin yanı sıra VAKBN, GARAN, 9 QNBFB ve AKBNK hisse senetleri ile USD/TRY, BIST30 ve BANKX endekslerine ait geçmiş veriler de kullanılmıştır. Çalışmada Adam ve RMSProp optimize edicileri ile Uzun Kısa Dönem Hafıza (LSTM) yöntemi kullanılmış ve performansı değerlendirilmiştir. Elde edilen sonuçlar ve sağlanan görseller neticesinde derin öğrenme modelleri, iletişimin son derece hızlı olduğu günümüz dünyasında her türlü askeri, siyasi, ekonomik ve jeopolitik haber akışından önemli ölçüde etkilenen borsadaki günlük fiyat hareketlerini kabul edilebilecek hatalar dahilinde tahmin edebilmiştir.[12]

Arslan, M. E., & KIRCI, P. (2021)'de, önümüzdeki günlerde bitcoin verilerini görselleştiren ve borsa hareketlerinin momentumuna uygun olarak kullanıcı için al ve sat sinyalleri oluşturan bir sistem, makine öğrenmesi ve geçmiş yıllardan gelen veriler kullanılarak öğretilmiştir. Hedef olarak, günümüzde ve gelecekte en kazançlı borsalardan biri olmaya aday Bitcoin borsası seçilmiştir. Bitcoin'in günlük en yüksek, en düşük, hacim, arz ve talep verilerine dayanarak, doğrusal regresyon algoritması alım ve satım sinyalleri oluşturmuştur. Quandl veritabanı, bu verileri toplamak için bitcoin ticaret borsası Bitfinex tarafından kullanılmıştır. Test bulgularına göre %95,5 doğruluk oranına sahip olmasına rağmen, sistem kurulduktan sonra yapılan incelemelerde, bir sosyal medya paylaşımının ardından bitcoin değerinin kısa sürede beklenen değer aralığından uzaklaştığı keşfedilmiştir. Bu verilere göre, bitcoin üzerindeki manipülasyon hareketleri %30'luk bir değişim oranına yaklaşabilirken, diğer uluslararası borsalardaki manipülasyon hareketleri tipik olarak %5 ile %10 arasındadır. Bu da bitcoin borsasının diğer borsalara kıyasla ne kadar değişken ve manipüle edilebilir olduğunu göstermektedir.[13]

Koç Ustalı, N. , Tosun, N. & Tosun, Ö. (2021)'in amacı, Borsa İstanbul Anonim Şirketi'nin BIST 30 Endeksi'nde yer alan şirketler için hisse senedi fiyatlarının tahmin edilmesidir. Bunun için öncelikle BIST 30 Endeksi şirketlerinin 2010-2019 yılları arasındaki üçer aylık finansal tabloları sunulmuş ve bu tablolar kullanılarak şirketlerin finansal oranları oluşturulmuştur. Ayrıca şirketlerin finansal oranlarına göre üçer aylık ortalamalar hesaplanmış ve şirket hisse senetlerinin aylık kapanış değerlerine ulaşılmıştır. Veriler toplandıktan sonra, her bir şirketin hisse senedinin gelecekteki değeri XGBoost, Rastgele Orman ve Yapay Sinir Ağları (YSA) algoritmaları kullanılarak tahmin edilmiştir. Her bir stratejiden elde edilen tahmin sonuçları daha sonra

karşılaştırılmıştır. XGBoost ve Rastgele Orman algoritmaları birbirlerine yakın sonuçlar vermelerine rağmen XGBoost algoritması en iyi sonucu vermektedir. Ayrıca her iki modelin de YSA'ya göre daha yüksek performans gösterdiği tespit edilmiştir. [14]

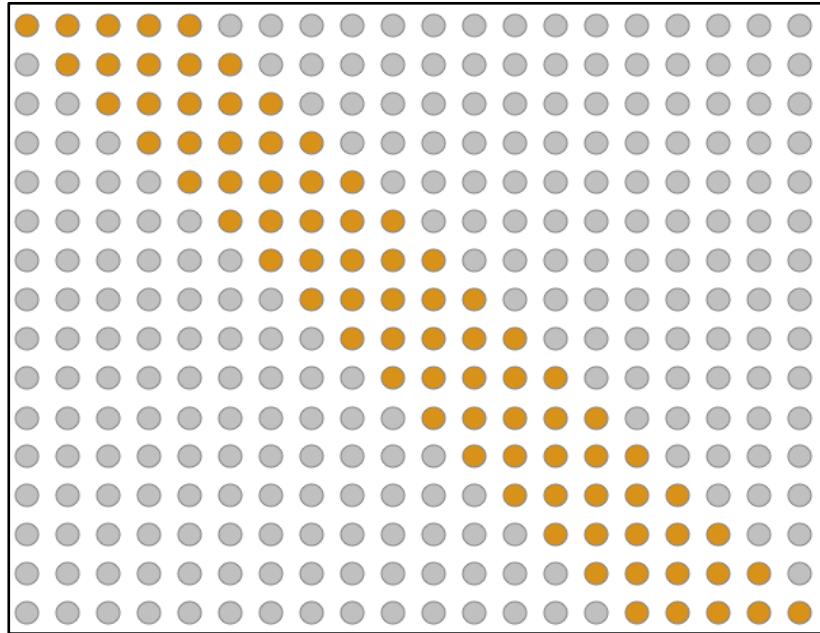
6. YÖNTEM VE TEKNOLOJİLER

Bu bölümde, Bitirme Tezinde kullanılacak yöntem literatüre bağlı olarak anlatılacak, tez çalışmasında kullanılacak teknolojiler ve tez çalışması kapsamında seçilen yaklaşım açıklanacaktır.

6.1. Sliding Window Tekniği

Sliding Window tekniği, genellikle makine öğrenmesi ve görüntü işleme bağlamında kullanılan bir veri dizisini yinlemek için bir yöntemdir. Verileri sabit boyutta örtüşen pencerelere bölerek ve her pencereyi bağımsız olarak işleyerek çalışır. Bu şekilde, kullanılan algoritmanın karmaşıklığını azaltır. [25] Alt dizinin sürekli bir sırayı takip etmeniz gereken problemleri çözmek için sıklıkla kullanılır, örneğin bir alt dizinin değerlerini toplamak veya yalnızca benzersiz karakterler içeren en uzun alt dizgeyi bulmak gibi. [26]

Bizim konumuz hakkında bir örnek vermek gerekirse: 1.000 günlük bir hisse senedi fiyatı zaman serisini ele alalım ve kayan pencere tekniğini kullanarak giriş-çıkış çiftlerini oluşturmak isteyelim. Eğer pencere boyutu 30 ve adım büyüklüğü 1 olarak belirlenirse, ilk adımda 1-30. günler arasındaki veri pencereye alınır ve bir sonraki adımda 2-31. günler arasındaki veri pencereye alınır. Bu işlem veri setinin sonuna kadar devam eder. Şekil 6.1'de Sliding Window tekniği verilmiştir.



Şekil 6.1 Sliding Window Tekniği [27]

6.2. LSTM (Uzun Kısa Süreli Bellek)

Çok sayıda tekrarlayan sinir ağı (RNN) çeşidinden biri olan uzun kısa süreli bellek (LSTM), önceki aşamalardan girdiyi yakalayabilir ve gelecek hakkında tahminler yapmak için kullanabilir. Bir yapay sinir ağının (YSA) tipik olarak üç katmanı vardır:

1. Giriş katmanı,
2. Gizli katman,
3. Çıkış katmanı.

Tek bir gizli katmana sahip bir YSA'nın giriş katmanındaki düğüm sayısı her zaman verilerin boyutuna bağlıdır ve giriş katmanının düğümleri "sinaps" olarak bilinen bağlantılarla gizli katmana bağlanır. (Girişten gizli katmana) her iki düğümlü ilişki, sinyallerin nasıl işleneceğini belirleyen ağırlık adı verilen bir katsayı içerir. Öğrenme süreci doğal olarak ağırlıkların sürekli olarak ayarlanmasıdır, öğrenme süreci tamamlandıktan sonra YSA her bir sinaps için optimum ağırlıklara sahip olacaktır.

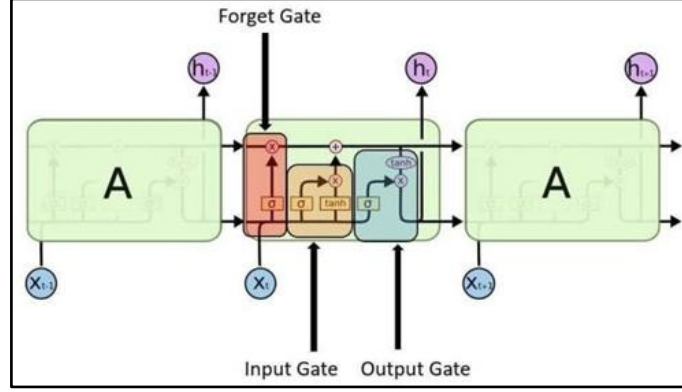
SoftMax fonksiyonu kullanılarak eğitime ve test verileri arasında hata oranı azaltılmış değerler üreten bir dönüşüm olan aktivasyon fonksiyonu, girdi katmanından gelen ağırlıkların toplamına gizli katman düğümleri tarafından uygulanır. Bu dönüşümün ardından elde edilen değerler YSA'mızın çıkış katmanını oluşturur; ancak bu değerler en iyisi olmayabilir. Bu durumda, optimum hata değerini hedeflemek için bir geri yayılım işlemi kullanılacaktır; bu işlem çıkış katmanını gizli katmana bağlar ve seçilen epoch sayısı için optimum hata ile en iyi ağırlığa uygun bir sinyal gönderir. Bu süreç, tahminlerimizi iyileştirmek ve tahmin hatasını en aza indirmek için tekrarlanır.

Bu işlem tamamlandıktan sonra model eğitilecektir. Tekrarlayan Sinir Ağları (RNN), verilerden öğrenmek ve gelecekteki eğilimleri tahmin etmek için önceki aşamaları kullanan bir NN sınıfıdır. Önceki gözlem dizilerine dayanarak gelecekteki değerleri tahmin eder.

Gelecekteki değerleri tahmin etmek ve tahmin etmek için, verilerin önceki aşamalarını akılda tutmak önemlidir; bu durumda gizli katman, sıralı verilerden gelen geçmiş veriler için bir depo görevi görür. Gelecekteki verileri tahmin etmek için önceki dizilerin bileşenlerini kullanma yöntemi, tekrarlayan olarak adlandırılır.

"Hafıza çizgisine" dayalı Uzun Kısa Vadeli Hafıza (LSTM), RNN'nin uzun süreli hafızayı depolayamaması nedeniyle uzun süreli veriler içeren tahmin senaryolarında oldukça yararlı bulunmuştur. Entegre bir hafıza hattına sahip bir

LSTM'nin kapıları, önceki aşamalardan bilgileri geri çağırmak için kullanılabilir. LSTM düğümlerinin yapısı aşağıdaki şekilde gösterilmektedir. Şekil 6.2'de LSTM'in iç yapısının modeli görülmektedir.

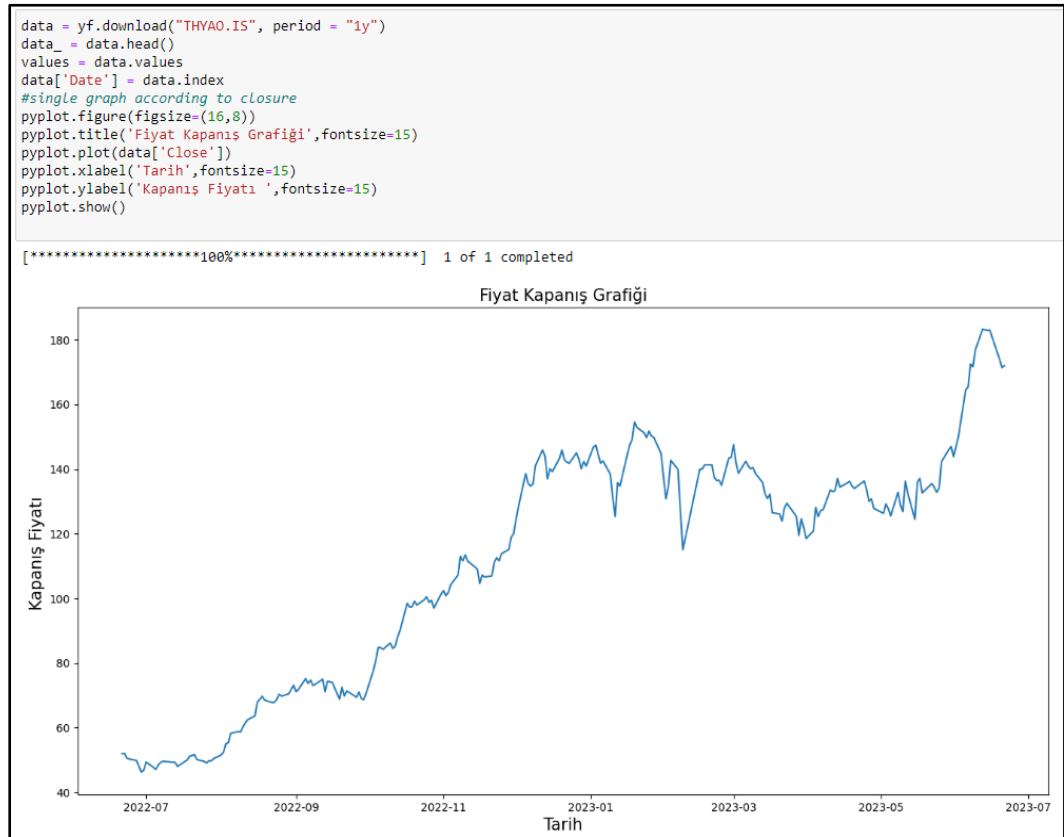


Şekil 6.2 Bir LSTM'nin iç yapısı [28]

LSTM, veri dizilerini ezberleme kapasitesine sahip olduğu için benzersiz bir RNN sınıfıdır. Her hücredeki üst çizgi, modelleri bir taşıma hattı olarak birbirine bağlar ve geçmişten gelen verileri şimdikilere aktarır. Hücrelerin bağımsızlığı, bir hücreden diğerine değer ekleyerek modelin elden çıkarma filtresine yardımcı olur. Her LSTM düğümü, aktarılan veri akışlarını depolamaktan sorumlu bir dizi hücreden oluşmalıdır. Verileri atarak veya akmasına izin vererek, kapıları oluşturan sigmoidal sinir ağı katmanı nihayetinde hücreyi ideal bir değere yönlendirir. Her sigmoid katman 0 ya da 1 ikili değerine sahiptir; burada 0 "hiçbir şeyin geçmemesine" ve 1 "her şeyin geçmesine" izin verir. Her bir hücrenin durumunu kontrol etmek için kapılar aşağıdaki şekilde yönetilir:

- Unut Kapısı 0 ile 1 arasında bir değer üretir; 1 "bunu tamamen sakla" ve 0 "bunu tamamen göz ardı et" anlamına gelir.
- Hafıza Kapısı hücreye hangi yeni verinin kaydedileceğine karar verir. Bir sigmoid katmanın ilk "giriş kapısı katmanı" değiştirilecek değerleri seçer. Daha sonra bir "tanh" katmanı tarafından duruma eklenebilecek potansiyel yeni aday değerlerin bir vektörü oluşturulur.
- Her hücrenin çıkışı çıkış kapısı tarafından belirlenir. Hücre durumu ile birlikte filtrelenmiş ve en son eklenen veriler, çıkış değerini belirlemek için kullanılacaktır.[15]

Aşağıdaki şekiller 6.3 ve şekil 6.10 arasında, projede kullanılmak üzere LSTM yöntemi ile 'Türk Hava Yolları Anonim Ortaklığı' hissesinin fiyat tahmininin gerçekleştirimi verilmiştir:



Şekil 6.3 THYAO 1 Yıllık Verisi

```
values = data['Close'].values.reshape(-1,1)
values = values.astype('float32')
scaler = MinMaxScaler(feature_range=(0, 1))
dataset = scaler.fit_transform(values)
# Birkaç Değere Bakalım
dataset[0:10]
```

```
array([[0.04171529],
       [0.04280922],
       [0.03186989],
       [0.02968204],
       [0.02640024],
       [0.01356477],
       [0.         ],
       [0.00452158],
       [0.02275378],
       [0.01093933]], dtype=float32)
```

Şekil 6.4 Veri Ölçeklendirme


```
# %70 Train % 30 Test
TRAIN_SIZE = 0.70
train_size = int(len(dataset) * TRAIN_SIZE)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size, :], dataset[train_size:len(dataset), :]
print("Gün Sayıları (training set, test set): " + str((len(train), len(test))))
```

Gün Sayıları (training set, test set): (175, 75)

Şekil 6.5 Eğitim ve Test Verisinin Ayrılması

```
def create_dataset(dataset, window_size = 1):
    data_X, data_Y = [], []
    for i in range(len(dataset) - window_size - 1):
        a = dataset[i:(i + window_size), 0]
        data_X.append(a)
        data_Y.append(dataset[i + window_size, 0])
    return(np.array(data_X), np.array(data_Y))
```

```
# Verisetlerimizi Oluşturalım
window_size = 1
train_X, train_Y = create_dataset(train, window_size)
test_X, test_Y = create_dataset(test, window_size)
print("Original training data shape:")
print(train_X.shape)
# Yeni verisetinin şekline bakalım.
train_X = np.reshape(train_X, (train_X.shape[0], 1, train_X.shape[1]))
test_X = np.reshape(test_X, (test_X.shape[0], 1, test_X.shape[1]))
print("New training data shape:")
print(train_X.shape)
print(test_X.shape)
```

Original training data shape:
(173, 1)
New training data shape:
(173, 1, 1)
(73, 1, 1)

Şekil 6.6 Sliding Window Yöntemi ile Veri Seti Oluşturma

```

def fit_model(train_X, train_Y, window_size = 1):
    model = Sequential()
    # Modelin iki layerlı şekilde kurulacak.
    model.add(LSTM(units=100, return_sequences=True, input_shape=(train_X.shape[1], 1)))
    model.add(LSTM(units=100, dropout= 0.1))
    model.add(Dense(units=1))
    model.compile(loss = "mean_squared_error",
                  optimizer = "adam")
    #100 epoch yani 100 kere verisetine bakılacak.
    model.fit(train_X,
              train_Y,
              epochs = 100,
              batch_size = 16,
              verbose = 1)

    return(model)
model1 = fit_model(train_X, train_Y, window_size)

```

Şekil 6.7 Modelin Eğitimi

```

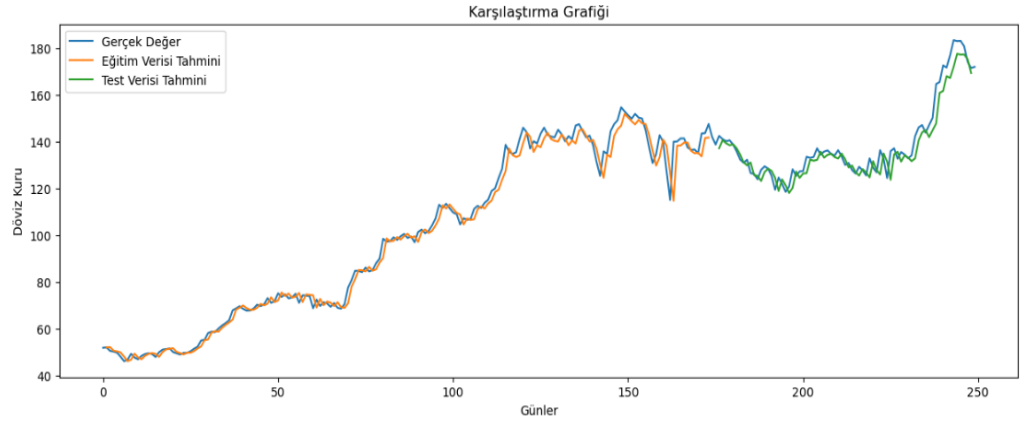
def predict_and_score(model, X, Y):
    # Şimdi tahminleri 0-1 ile scale edilmiş halinden geri çeviriyoruz.
    pred = scaler.inverse_transform(model.predict(X))
    orig_data = scaler.inverse_transform([Y])
    # Rmse değerlerini ölçüyoruz.
    score = math.sqrt(mean_squared_error(orig_data[0], pred[:, 0]))
    return(score, pred)
rmse_train, train_predict = predict_and_score(model1, train_X, train_Y)
rmse_test, test_predict = predict_and_score(model1, test_X, test_Y)
print("Training data score: %.2f RMSE" % rmse_train)
print("Test data score: %.2f RMSE" % rmse_test)

```

6/6 [=====] - 0s 1ms/step
 3/3 [=====] - 0s 1ms/step
 Training data score: 3.93 RMSE
 Test data score: 4.89 RMSE

Şekil 6.8 Tahminleme ve Hata Oranları

```
# Öğrendiklerini tahminletip ekliyoruz.
train_predict_plot = np.empty_like(dataset)
train_predict_plot[:, :] = np.nan
train_predict_plot[window_size:len(train_predict) + window_size, :] = train_predict
# Şimdi ise testleri tahminletiyoruz.
test_predict_plot = np.empty_like(dataset)
test_predict_plot[:, :] = np.nan
test_predict_plot[len(train_predict) + (window_size * 2) + 1:len(dataset) - 1, :] = test_predict
# Plot'u oluşturalım.
plt.figure(figsize = (15, 5))
plt.plot(scaler.inverse_transform(dataset), label = "Gerçek Değer")
plt.plot(train_predict_plot, label = "Eğitim Verisi Tahmini")
plt.plot(test_predict_plot, label = "Test Verisi Tahmini")
plt.xlabel("Günler")
plt.ylabel("Döviz Kuru")
plt.title("Karşılaştırma Grafiği")
plt.legend()
plt.show()
```



Şekil 6.9 Genel Sonuç Grafiği

```
# Sonraki günün tahminini yapma
last_day = dataset[-1].reshape(1, 1, 1)
next_day_prediction = model1.predict(last_day)
next_day_prediction = scaler.inverse_transform(next_day_prediction)
x="{:.2f}".format(next_day_prediction[0][0])
print("Sonraki günün tahmini: ", x)
```

```
1/1 [=====] - 0s 13ms/step
Sonraki günün tahmini: 167.40
```

Şekil 6.10 Sonraki Gün Tahmini

6.3. Naive Bayes

Bir sınıflandırma/kategorilendirme algoritması olan Naive Bayes sınıflandırması, olasılık ilkelerine göre tanımlanmış bir dizi hesaplama ile, sisteme sunulan verilerin sınıfını yani kategorisini tespit etmeyi amaçlar.

Naive Bayes sınıflandırmasında algoritmaya belirli miktarda öğretilen veri sağlanır (Örn: 100 adet). Öğretim için sağlanan veriler bir sınıfa/kategoriye sahip olmalıdır. Öğretilen veriler üzerindeki olasılık işlemleri, daha önce öğrenilen olasılık değerlerine göre sisteme sağlanan yeni test verileri üzerinde hareket eder ve mevcut test verilerinin hangi kategoriye ait olduğunu belirlemeye çalışır. Elbette, öğretilen veri miktarı ne kadar büyük olursa, doğru test verisi grubunun tanımlanması o kadar kesin olur. [16]

Naive Bayes yöntemi, uygulanmasının basit olması, çok büyük veri kümeleri için kullanılabilmesi ve çok karmaşık sınıflandırma yaklaşımlarından daha iyi performans gösterdiğinin bilinmesi gibi avantajlara sahiptir. Bu yöntemde izlenecek temel aşamalar şunlardır:

- Sağlanan veri setini bir frekans tablosuna dönüştürün. 2. Oluşma olasılığını belirleyin ve olasılıklara dayalı bir olabilirlik tablosu oluşturun.
- Naive Bayes denklemini kullanarak her sınıf için sonsal olasılığı belirleyin.
- Tahminin sonucu, en yüksek arka olasılığa sahip sınıftır.

6.4. Destek Vektör Makineleri (DVM)

Karar fonksiyonunun kapasite kontrolü, kernel fonksiyonlarının kullanımı ve çözümün seyrekliği, özel bir öğrenme yöntemi olan destek vektör makinesinin (DVM) tanımlayıcı özellikleridir. DVM'nin aşırı uyum sorununa karşı son derece dirençli olduğu ve sonuçta yüksek bir genelleme performansı elde ettiği kanıtlanmıştır. DVM, genelleme hatasının bir üst sınırını en aza indirerek bir fonksiyonu tahmin etmek için yapısal risk minimizasyonu prensibinin yeni teorisine dayanmaktadır. Doğrusal olmayan optimizasyona ihtiyaç duyan ve yerel minimumlarda sıkışıp kalma riski taşıyan sinir ağı eğitiminin aksine, DVM eğitimi doğrusal olarak kısıtlanmış ikinci dereceden bir programlama problemini çözmeye benzer, dolayısıyla DVM'nin çözümü her zaman benzersiz ve global olarak optimaldir.

Son zamanlarda, finansal tahminlerle ilgili konularda bazı DVM uygulamaları tespit edilmiştir. Çoğu zaman, tahminler ile gözlemlenen değerler arasındaki farklar, tahminlerin doğruluk ve kabul edilebilirlik düzeyi için bir

gösterge görevi görmektedir. Tahmin hatasını azaltmaya odaklanan tahmin teknikleri, finans sektörü uygulayıcılarının hedeflerine ulaşmaları için yeterli olmayabilir. Başka bir deyişle, küçük bir tahmin hatasına sahip bir tahmine dayalı alım satım, hareketin yönüne ilişkin kesin bir öngörüye dayalı alım satım kadar karlı olmayabilir. [17]

Destek vektör makinesi (SVM) olarak bilinen denetimli öğrenme tekniği ile girdi-çıkı eşleme fonksiyonları oluşturmak için bir dizi etiketli eğitim verisi kullanılır. Sınıflandırma fonksiyonu veya girdi verilerinin kategorisi ya da regresyon fonksiyonunun her ikisi de eşleme fonksiyonu olarak kullanılabilir. Doğrusal olmayan çekirdek fonksiyonları, girdi verilerini, girdi verilerinin orijinal girdi uzayına göre daha kolay ayrılabilirdiği yüksek boyutlu bir özellik uzayına dönüştürmek için sınıflandırmada sıklıkla kullanılır. Ardından, maksimum marjlı hiper düzlemler oluşturulur. Bu şekilde oluşturulan model, eğitim verilerinin yalnızca sınıf sınırlarına yakın bir kısmına bağımlıdır. Buna benzer şekilde, model tahminine yeterince yakın olan herhangi bir eğitim verisi SVR modeli tarafından göz ardı edilir. DVM'ler "kernel yaklaşımları" olarak da anılmaktadır.

DVM'ler, biyoinformatik, metin madenciliği, yüz tanıma ve görüntü işleme gibi çeşitli gerçek dünya uygulamalarındaki yüksek rekabetçi performansları sayesinde, sinir ağları ve bulanık sistemler gibi diğer yumuşak hesaplama teknikleriyle birlikte makine öğrenimi ve veri madenciliği için en gelişmiş araçlardan biri olarak kendilerini kanıtlamışlardır. [18]

6.5. ARIMA

ARIMA (Otokorelasyonlu Entegre Hareketli Ortalama), istatistik ve ekonometride, özellikle zaman serisi analizinde, otokorelasyonlu hareketli ortalama (ARMA) modelinin bir genelleştirilmesidir. Verileri daha iyi anlamak veya gelecek serisi noktalarını tahmin etmek için bu modeller zaman serisi verilerine uygulanır. [30]

Trend ve sezonsallıktan arındırılmış olan durağan zaman serileri için ARIMA öngörüsü doğrusal (doğrusal regresyon gibi) denklemden başka bir şey değildir. Tahmini, ARIMA modelinin parametrelerine (p, d, q) bağlıdır.

(AR) Otomatik Regresyon terimi, bir modelin bir gözlem ile bazı gecikmeli gözlemler arasındaki bağımlı ilişkiyi kullanmasını ifade eder.

p: AR modelinde kullanılan gecikmeli gözlem sayısı veya gecikme derecesi olarak adlandırılır. Bu değer, bağımlı değişkenin ne kadar önceki gözlemlerle ilişkili olduğunu belirler.

Hareketli Ortalama (MA) terimi, bir modelin bir gözlem ile bağımlılık arasındaki gecikme gözlemlerine uygulanan hareketli bir ortalama modelinden kalan bir hata arasındaki ilişkiyi kullanmasını ifade eder.

q: MA modelinde kullanılan hareketli ortalama sırası veya hareketli ortalama penceresinin boyutu olarak adlandırılır. Bu değer, hatanın ne kadar önceki gözlemlerle ilişkili olduğunu belirler.

I (Entegrasyon) Farkların Sayısı (d): Ham gözlemlerin farklılaştırılmasıyla zaman serilerini hareketsiz hale getirmek için kullanılan bir yöntemdir (örneğin, bir önceki zaman adımındaki gözlemden bir sonraki zaman adımındaki gözlem çıkarılarak).

d: Ham gözlemlerin farklılaştırılması ile elde edilen fark sayısı, aynı zamanda farklılaşma derecesi olarak da adlandırılır. [31]

6.5.1. ARIMA Kullanım Alanları

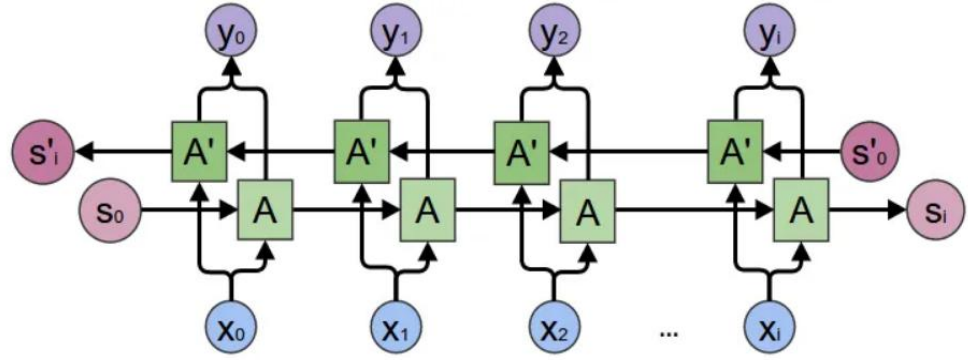
ARIMA modelinin veri seti üzerinde uygulanabilirliği, verilerin durağan (stationary) veya durağan olmayan (non-stationary) olup olmadığına, tek değişkenli olduğuna ve trend, mevsimsellik, döngü gibi her türlü veri modeliyle uyumlu olması durumuna bağlıdır. ARIMA modeli genellikle kısa vadeli tahminler için önerilir. Ancak, ARIMA modeli yalnızca tek değişkenli verilerin tahmininde kullanılır ve bağımlı değişkeni etkileyen diğer değişkenlerin etkisini dikkate almaz.

Bununla birlikte, küçük veri setleri üzerinde (dar tarih aralığı) ARIMA modelini uygularken doğru sonuçlar elde etmek zor olabilir. [32]

6.6. Bidirectional LSTM

Çift yönlü olan tekrarlayan sinir ağları temelde iki ayrı RNN'nin birleştirilmesinden ibarettir. Ağlar bu yapı sayesinde her zaman adımında diziyle ilgili hem ileri hem de geri bilgiye erişebilir. [33]

Çift yönlü kullanıldığında, girdileriniz iki farklı yönde işlenecektir: Biri bugünden geleceğe ve diğeri gelecekte bugüne. Bu yöntemin tek yönlüden farkı, gelecekte gelen bilgilerin geriye doğru çalışan LSTM'de korunması ve iki gizli durumu birleştirerek herhangi bir zamanda hem şimdiki hem de gelecekte gelen verileri koruyabilmenizdir. Bidirectional LSTM modeli şekil 6.11'de sembolik olarak görülebilmektedir.



Şekil 6.11 Bidirectional LSTM Modeli [29]

6.7. Yapay Sinir Ağları

Biyolojik sinir sisteminin matematiksel bir temsili olan yapay sinir ağları, karmaşık bilgisayar işlemlerini gerçekleştirme yeteneğine sahiptir. [34] Yapay sinir ağları veri taşıyan küçük yapılardan oluşur. Düzlemler olarak adlandırılan bu küçük yapı katmanlarını içerir. Bu yapılar yalnızca bir önceki katmanla ve bir sonraki katmandaki daha küçük yapılarla etkileşime girer; kendi katmanları içinde birbirleriyle iletişim kurmazlar. Ancak modele bağlı olarak katman sayısı değişebilir. Burada önemli olan mümkün olan en yüksek tahmin değerlerini bulmaktır. Bu da daha fazla katman eklenmesine neden olabilir. Sinir ağı modeli en kesin sonucu belirleyene kadar hesaplamaya devam edecektir. Model, gerçek değerlerle aynı olan değerleri bulduğunda çalışmayı ve öğrenmeyi durduracaktır. [35] Tablo 6.1 kullanılan ortam gereksinimleri görülmektedir.

6.8. Kullanılan Dil, Ortam ve Kütüphaneler

Tablo 6.1 Kullanılan Ortam Gereksinimleri

Donanım	PC
İşletim Sistemi	Windows 10 Pro
Geliştirme Ortamı	Jupyter Notebook, Spyder
Programlama Dili	Python
Kütüphaneler	yFinance, Scikit-Learn, Keras, Tensorflow, Matplotlib

Jupyter Notebook: Jupyter Notebook, veri bilimcileri tarafından canlı kod, denklemler, görselleştirmeler ve anlatı metni içeren belgeler oluşturmak ve paylaşmak için kullanılan açık kaynaklı bir web uygulamasıdır.[36] Basit ve akıcı bir belge merkezli deneyim sunar ve kod ile çıktısını görselleştirmeler, anlatı metni, matematiksel denklemler ve diğer zengin medyayı birleştiren tek bir belgede birleştirir. [37].

Python: Python, Guido Van Rossum tarafından geliştirilen bir programlama dilidir. Açık kaynak kodlu olması, geniş bir kütüphaneye sahip olması, birçok modül eklenebilmesi gibi avantajları bulunmaktadır. [38]

Web geliştirme, veri bilimi, makine öğrenimi, sistem otomasyonu, API geliştirme, GUI geliştirme, oyun geliştirme, bilimsel hesaplama, yapay zeka, veri analizi, veritabanı programlama, ağ programlama, IoT, mikrodenetleyici programlama, test otomasyonu, araç geliştirme, yazılım otomasyonu ve daha birçok alanda kullanılabilir. [39]

yFinance: yFinance, Yahoo Finance API'sini kullanarak finansal verileri almak için Python kütüphanesidir. Bu kütüphane, bir hisse senedi veya kripto para birimi gibi bir finansal varlıkla ilgili verileri toplamak için kullanılabilir. yFinance, tarihsel fiyat verileri, hisse senedi fiyatları, finansal raporlar ve diğer finansal veriler gibi birçok farklı veri tipini destekler. Bu kütüphane, finansal verileri hızlı ve kolay bir şekilde almak için kullanışlı bir araçtır. [40]

Scikit Learn: Scikit-learn, Python dilinde kullanılan açık kaynak kodlu bir makine öğrenmesi kütüphanesidir. NumPy, SciPy ve Matplotlib gibi kütüphanelerin üzerine inşa edilmiştir ve sınıflandırma, regresyon, kümeleme ve boyut azaltma gibi makine öğrenimi algoritmaları için çeşitli fonksiyonlar sunar. Bu kütüphane, verileri okuma ve ön işleme, model seçimi, model eğitimi ve sonuçların değerlendirilmesi gibi birçok işlemi kolaylaştıran API'ler sunar. [41]

Keras: Keras, Python dilinde yazılmış, Theano ve TensorFlow üzerine oluşturulmuş bir derin öğrenme kütüphanesidir. Keras, neredeyse her tür derin öğrenme modelini tanımlamak ve eğitmek için uygun bir yol sağlayan bir üst düzey sinir ağları API'sidir. Keras, sinir ağları oluşturmak ve eğitmek için başlık altındaki karmaşık ayrıntıların çoğunu gizleyen basit ve modüler bir API sağlar. [42]

Tensorflow: Google Brain ekibi tarafından oluşturulan ve ilk kez 2015 yılında halka açıklanan TensorFlow, sayısal hesaplama ve büyük ölçekli makine öğrenmesi için açık kaynaklı bir kütüphanedir. TensorFlow, bir dizi makine öğrenmesi ve derin öğrenme modelini ve algoritmayı (neural ağlar olarak da bilinir) bir araya getirir ve ortak programlama metaforları aracılığıyla kullanılabilir hale getirir. [43]

Matplotlib: Matplotlib, Python programlama dilinde veri görselleřtirmesi için kullanılan bir kütüphanedir. Verileri grafikler, diyagramlar ve çizimlerle görsel olarak temsil etmek için bir dizi fonksiyon sunar. Matplotlib, çizgi grafikleri, sütun grafikleri, dağılım grafikleri ve daha birçok grafik türünü oluşturmak için kullanılabilir. [44]

7. GERÇEKLEŞTİRİM

7.1 Analiz

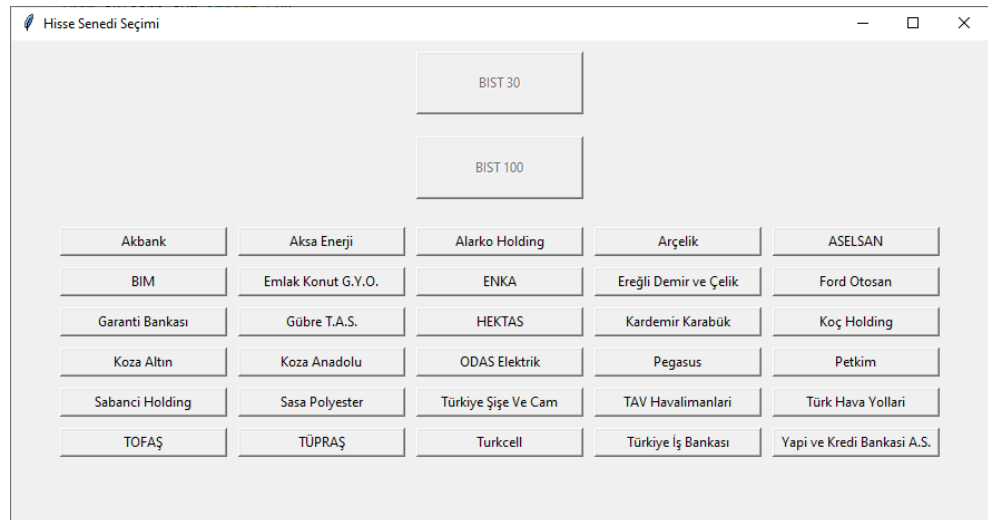
Finansal zaman serileri konusunda, bu bölümde derinlemesine bir inceleme yapılmıştır. Analiz kapsamında, finansal verilerin özellikleri, geçmiş performansları, gelecekteki eğilimleri ve dalgalanmaları gibi konular ayrıntılı bir şekilde incelenmiştir. Bu analizler, finansal verilerin nasıl işleneceği, yorumlanacağı ve kullanılacağı konusunda önemli bir temel sağlamaktadır.

Projede kullanmak üzere BIST30'daki hisse senetleriyle ilgili verilerin çekilmesi için Yahoo Finance API kullanılmıştır.

Bununla birlikte, bu bölümde kullanılacak olan makine öğrenimi algoritmaları hakkında da bir inceleme yapılmıştır. LSTM, SVM, ANN ve Naive Bayes modellerinin özellikleri araştırılmış ve uygunluğu değerlendirilmiştir. Bu inceleme, finansal verilerin daha doğru bir şekilde analiz edilmesi ve gelecekteki eğilimlerin tahmin edilmesi için kullanılan teknikler hakkında kapsamlı bir bilgi sağlamaktadır.

7.2 Tasarım

Kullanıcının hisse seçimi için basit bir arayüz yapılacak ve seçilen hissenin verileri elde edilecektir. Toplanan veriler, eksik veya yanlış verileri düzeltmek için kontrol edilecektir. Ayrıca, verilerin doğru bir şekilde analiz edilmesi için ölçeklendirme, normalizasyon ve diğer ön işleme teknikleri uygulanacaktır. Verilerin boyutu, gereksiz gürültüleri ve özellikleri kaldırmak için azaltılacaktır. Şekil 7.1'de kullanıcı seçim ekranının demo görüntüsü görülmektedir.



Şekil 7.1 Hisse Senedi Seçim Ekranı

Finansal zaman serilerinin tahmini için Yapay Sinir Ağları (YSA), Uzun Kısa Süreli Bellek (LSTM), Destek Vektör Makineleri (SVM) ve Naive Bayes gibi farklı makine öğrenimi algoritmaları kullanılacaktır. Bu algoritmaların performansı, önceden ayrılmış veri setleri kullanılarak eğitilerek karşılaştırılacaktır. En iyi performansı veren model, hiperparametre optimizasyonu teknikleri kullanılarak optimize edilecektir.

Eğitilen modellerin performansı test edilecektir. Test verileri kullanarak, model tahminlerinin doğruluğu, hassasiyeti, hata oranı gibi metrikler kullanılarak değerlendirilecektir.

7.3 DVM Gerçekleştirimi

DVM modeli kullanarak BIST100 endeksinin değer tahminini yaptığımız programın önemli adımları aşağıda verilmiştir:

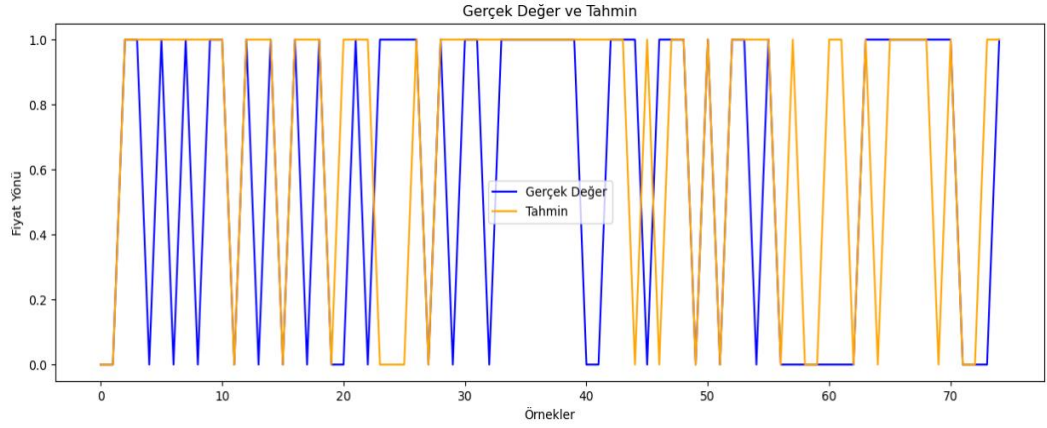
- "XU100.IS" sembolüne sahip BIST100 endeksinin 1 yıllık verileri indirildi ve df isimli bir DataFrame'e atandı.
- df DataFrame'ine "daily_change" isimli yeni bir sütun eklendi. Bu sütun, endeksin günlük yüzde değişimini ifade etmektedir.
- DataFrame'e "target" isimli bir sütun daha eklendi. Bu sütun, endeksin günlük değişiminin pozitif (artış) veya negatif (azalış) olduğunu belirtmektedir. Eğer "daily_change" sütunu 0'dan büyükse 1, aksi takdirde 0 olarak belirlenir. Şekil 7.2'de DataFrame görülebilmektedir.

Date	Open	High	Low	Close	Adj Close	Volume	daily_change	target
2022-06-20	2549.000000	2555.100098	2511.300049	2539.199951	2539.199951	2620212300	NaN	0
2022-06-21	2557.100098	2582.199951	2550.800049	2575.899902	2575.899902	2721684000	0.014453	1
2022-06-22	2572.199951	2586.899902	2552.100098	2574.300049	2574.300049	2424694900	-0.000621	0
2022-06-23	2576.100098	2588.699951	2533.899902	2543.199951	2543.199951	2624314200	-0.012081	0
2022-06-24	2554.800049	2570.600098	2525.199951	2554.100098	2554.100098	2511887500	0.004286	1
...
2023-06-13	5568.100098	5595.200195	5392.799805	5429.100098	5429.100098	3280041000	-0.028610	0
2023-06-14	5437.100098	5490.899902	5322.000000	5360.799805	5360.799805	3562526600	-0.012580	0
2023-06-15	5397.799805	5495.500000	5378.299805	5495.500000	5495.500000	2789147400	0.025127	1
2023-06-16	5518.500000	5566.100098	5467.000000	5475.500000	5475.500000	2959480100	-0.003639	0
2023-06-19	5485.470215	5485.470215	5359.910156	5405.890137	5405.890137	0	-0.012713	0

250 rows x 8 columns

Şekil 7.2 Veri Seti Ekran Görüntüsü

- X ve y değişkenleri tanımlandı. X, bağımsız değişkenleri (açılış fiyatı, en yüksek fiyat, en düşük fiyat, kapanış fiyatı, düzeltilmiş kapanış fiyatı ve işlem hacmi) içeren bir DataFrame olarak belirlenir. y ise hedef değişkeni temsil eder ve "target" sütununu içerir.
- Eğitim ve test verileri oluşturmak için train_test_split() fonksiyonu kullanılır. X ve y verileri, %70 eğitim ve %30 test verisi olarak bölündü.
- SVC sınıfı kullanılarak SVM modeli oluşturuldu. C=1.0 parametresi, hata toleransının kontrol edilmesini sağlar. random_state=1 parametresi ise tekrarlanabilir sonuçlar elde etmek için rastgelelik durumunu sabitler. kernel='linear' parametresi, doğrusal bir çekirdek kullanılacağını belirtir.
- Model eğitilip tahminleme yaptırıldı. Modelin performansı, accuracy_score() fonksiyonu kullanılarak gerçek test etiketleri (y_test) ile tahmin edilen etiketler (y_predict) arasındaki doğruluk oranıyla ölçüldü. Modelin performansı %68 olarak bulundu. Şekil 7.3'de DVM çıktı grafiksel görseli verilmiştir.



Şekil 7.3 DVM Çıktısı

7.4 Naive Bayes Gerçekleştirimi

Naive Bayes modeli kullanılarak BIST30 endeksinin değer tahminini yaptığımız programın önemli adımları aşağıda verilmiştir:

- "XU030.IS" sembolüne sahip BIST30 endeksinin 1 yıllık verileri indirildi ve "dataset" isimli bir DataFrame'e atandı.
- "dataset" DataFrame'ine "daily_change" isimli yeni bir sütun eklendi. Bu sütun, endeksin günlük yüzde değişimini ifade etmektedir.

- DataFrame'e "target" isimli bir sütun daha eklendi. Bu sütun, endeksin günlük değişiminin pozitif (artış) veya negatif (azalış) olduğunu belirtmektedir. Eğer "daily_change" sütunu 0'dan büyükse 1, aksi takdirde 0 olarak belirlenir. Şekil 7.4'de DataFrame görülebilmektedir.

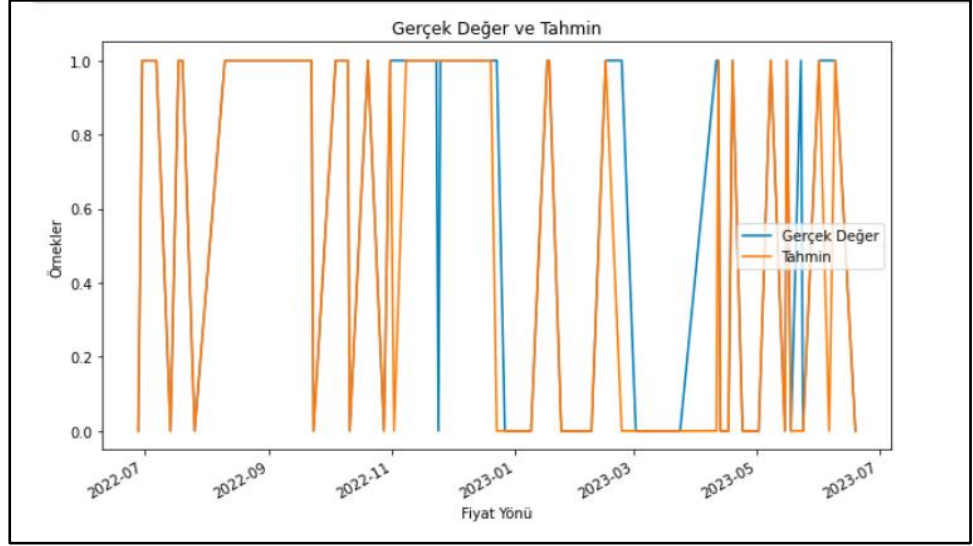
	Open	High	Low	Close	Adj Close	Volume	daily_change	target
Date								
2022-06-21	2805.300049	2840.300049	2795.300049	2831.300049	2831.300049	1557400	0.016552	1
2022-06-22	2826.800049	2839.800049	2800.199951	2821.399902	2821.399902	1403600	-0.003497	0
2022-06-23	2823.100098	2834.699951	2766.699951	2774.300049	2774.300049	1396900	-0.016694	0
2022-06-24	2786.800049	2805.100098	2748.100098	2783.000000	2783.000000	1425000	0.003136	1
2022-06-27	2766.800049	2778.100098	2701.899902	2754.100098	2754.100098	1289100	-0.010384	0
...
2023-06-13	6191.500000	6235.500000	5994.299805	6037.600098	6037.600098	2118100	-0.029917	0
2023-06-14	6045.799805	6117.600098	5914.399902	5967.600098	5967.600098	2381400	-0.011594	0
2023-06-15	6013.299805	6133.899902	5982.700195	6133.899902	6133.899902	1803400	0.027867	1
2023-06-16	6159.399902	6212.799805	6095.000000	6104.799805	6104.799805	1955600	-0.004744	0
2023-06-19	6115.410156	6115.410156	5974.529785	6015.410156	6015.410156	0	-0.014643	0

Şekil 7.4 Veri Seti Ekran Görüntüsü

- X ve y değişkenleri tanımlandı. X için 'dataset' adlı bir veri kümesinden(açılış fiyatı, en yüksek fiyat, en düşük fiyat, kapanış fiyatı, düzeltilmiş kapanış fiyatı, işlem hacmi ve günlük değişim) sütunlarını seçilir. y ise hedef değişkeni temsil eder ve "target" sütununu içerir.
- Eğitim ve test verileri oluşturmak için train_test_split() fonksiyonu kullanılır. X ve y verileri, %70 eğitim ve %30 test verisi olarak bölünür.
- 'StandardScaler' sınıfı, veri özelliklerini standartlaştırma yöntemiyle ölçeklendirmek için kullanılır. Standartlaştırma işlemi, her bir özelliği ayrı ayrı ele alır ve her özelliğin ortalamasını 0, standart sapmasını ise 1 olarak dönüştürür. Bu, özellikler arasındaki farklı ölçekleri dengeleyerek, bir modelin daha iyi performans göstermesine yardımcı olur.
- 'fit_transform' yöntemi, eğitim veri setine uygulanarak, ölçeklendirme için kullanılan ortalama ve standart sapma değerlerini hesaplar ve eğitim veri setini dönüştürür. 'transform' yöntemi ise bu hesaplanan değerleri kullanarak, test veri setini aynı ölçeklendirmeye tabi tutar. Sonuç olarak, hem eğitim veri seti hem de test veri seti ölçeklendirilir ve özellikler arasındaki karşılaştırmalar daha tutarlı bir şekilde yapılabilir.
- 'GaussianNB' sınıfı, Bayes teoremine dayanan bir makine öğrenme algoritması olan Gaussian Naive Bayes kullanılarak 'classifier' adında bir örnek oluşturulur ve bu örnek eğitim veri seti (x_train ve y_train)

üzerinde eğitilir. Eğitim, sınıflandırıcıya veri setindeki özelliklerin ve hedef değerlerin ilişkisini öğretir. Sonuç olarak, eğitilmiş sınıflandırıcı modeli oluşur ve daha sonra yeni örneklerin sınıflandırılması için kullanılabilir.

- Model eğitilip tahminleme yaptırılır. Modelin performansı, `accuracy_score()` fonksiyonu kullanılarak gerçek test etiketleri (`y_test`) ile tahmin edilen etiketler (`y_predict`) arasındaki doğruluk oranıyla ölçülür. Modelin performansı %85 olarak bulundu. Şekil 7.5’de grafiksel çıktı görülebilmektedir.



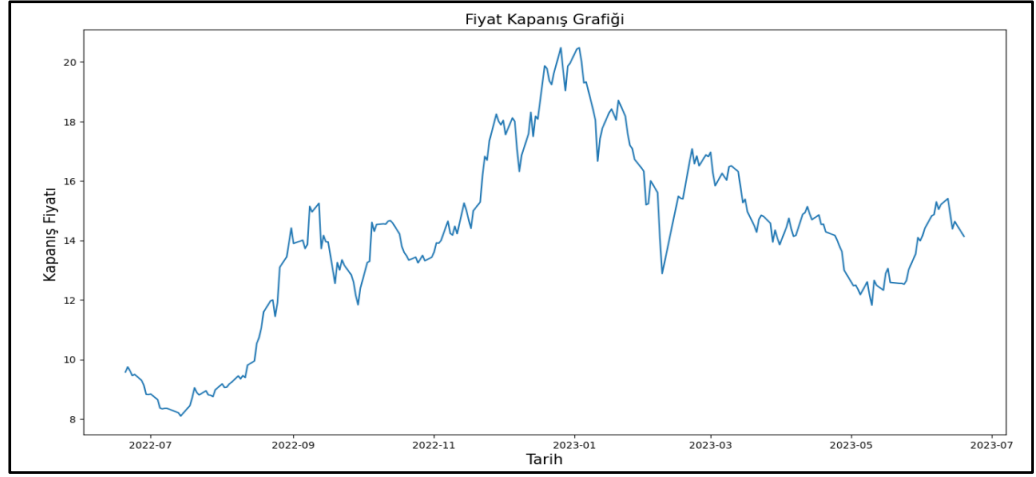
Şekil 7.5 Naive Bayes Çıktısı

7.5 Stacked LSTM Gerçekleştirimi

- "yfinance" kütüphanesi kullanılarak "PETKM.IS" sembolüne ait son 1 yılın verisinin indirildi. (Kendi çalışmalarımızda son 1 yıllık veriler üzerinden ilerledik. Ancak literatürdeki çalışmalarla karşılaştırma yaptığımız durumlarda benzer çalışmaların aralıklarını kullanarak ilerledik)

- Verilerin görselleştirilmesi için bir grafik oluşturuldu. Grafik, kapanış fiyatlarını temsil ediyor.

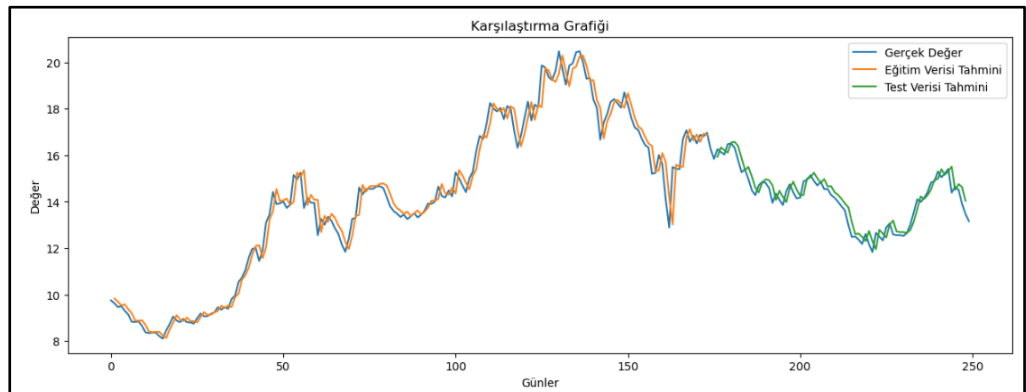
Şekil 7.6’da PETKM.IS’in fiyat, tahmin verileri grafiksel olarak gösterilmiştir.



Şekil 7.6 PETKM.IS Fiyat Kapanış Grafiği

- Veri setinin normalizasyonu için Min-Max ölçeklendirme yönteminin uygulandı.
- Veri setinin eğitim ve test verisi olarak ayrılması. Bu aşamada veri setinin %70'i eğitim verisi, %30'u ise test verisi olarak kullanıldı.
- LSTM modeli için girdi ve çıktı veri setlerinin oluşturuldu. Bu aşamada bir "pencere boyutu" belirlendi ve bu pencere boyutu kadar önceki veriler kullanarak bir sonraki verinin tahmini yapıldı.
- LSTM modelinin oluşturuldu, derlendi ve eğitildi. Model, 2 LSTM katmanı ve bir Dense katmanı içermektedir.
- Eğitilmiş modelin kullanarak eğitim ve test veri setleri üzerinde tahminler yapıldı ve hata hesaplamaları gerçekleştirildi. Hata hesaplaması için kök ortalama karesel hata (RMSE) kullanıldı.

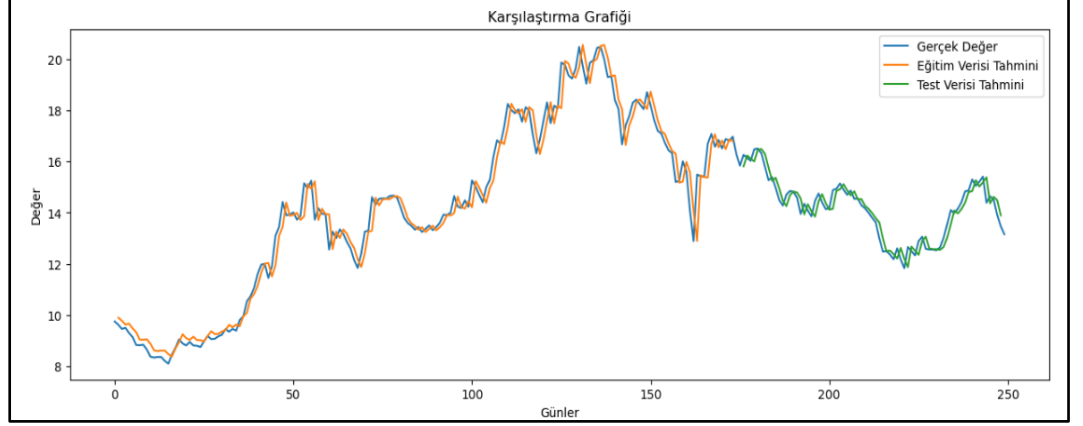
Şekil 7.7'de LSTM çıktısı verilmiştir.



Şekil 7.7 LSTM Çıktısı

7.6 Bidirectional LSTM Gerçekleştirimi

Birçok kısmı stacked LSTM ile aynı şekilde gerçekleştirilmektedir. Yalnızca model oluşturulurken 1 tane bidirectional LSTM katmanı ve 1 tane de Dense katmanı kullanılmıştır. Şekil 7.8’de Bidirectional LSTM çıktısı verilmiştir



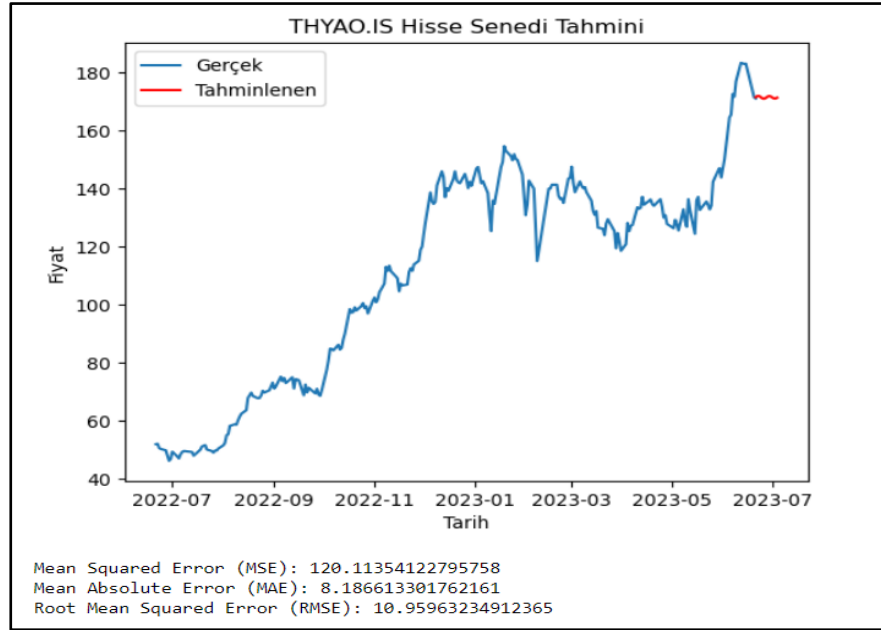
Şekil 7.8 Bidirectional LSTM Çıktısı

Stacked LSTM ile aynı veriler eşliğinde çalıştırıldığında; Stacked LSTM 0.39 RMSE değerini verirken bidirectional LSTM ise 0.36 RMSE değerini vermiştir.

7.7 ARIMA Gerçekleştirimi

- İlk olarak gerekli kütüphaneler (Pandas, NumPy, yfinance, statsmodels, matplotlib) import edilir.
- Ardından yf.download fonksiyonu kullanılarak Yahoo Finans'tan "THYAO.IS" adlı hisse senedinin 1 yıllık verisi indirilir.
- ARIMA modeli oluşturulur. ARIMA sınıfı df["Close"] sütununu hedef değişken olarak kullanır ve (2, 1, 2) parametreleriyle çağrılır. Bu parametreler, ARIMA modelinin otoregresif (AR), entegre (I) ve hareketli ortalama (MA) bileşenlerinin derecelerini belirtir.
- Oluşturulan model, model.fit() fonksiyonu ile verilere uyum sağlar.
- Sonraki 14 gün için tahmin yapılır.
- Tarihsel ve tahmin edilen veriler, plt.plot() fonksiyonuyla bir grafik üzerinde görselleştirilir.

- Model performansı değerlendirilir. Son 14 gün tahmini ve gerçek değerler alınır. Ardından ortalama karesel hata (MSE), ortalama mutlak hata (MAE) ve kök ortalama karesel hata (RMSE) hesaplanır. Şekil 7.9’da ARIMA çıktısı verilmiştir.



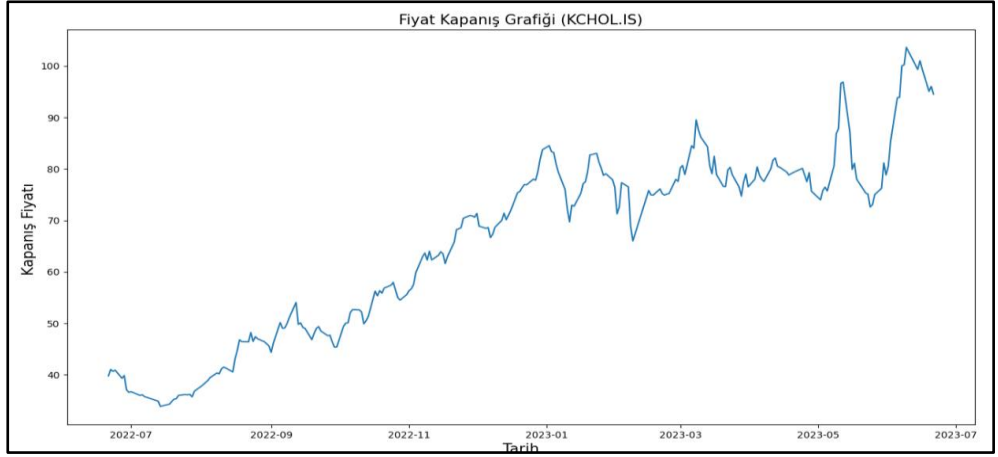
Şekil 7.9 ARIMA Çıktısı

7.8 YSA Gerçekleştirimi

Yapay sinir ağları modeli ile tahminleme yaptığımız programın gerçekleştirmesindeki önemli adımlar aşağıda verilmiştir:

- Örnek veriseti olarak KCHOL.IS hissesinin 1 yıllık kapanış değerleri kullanılır.

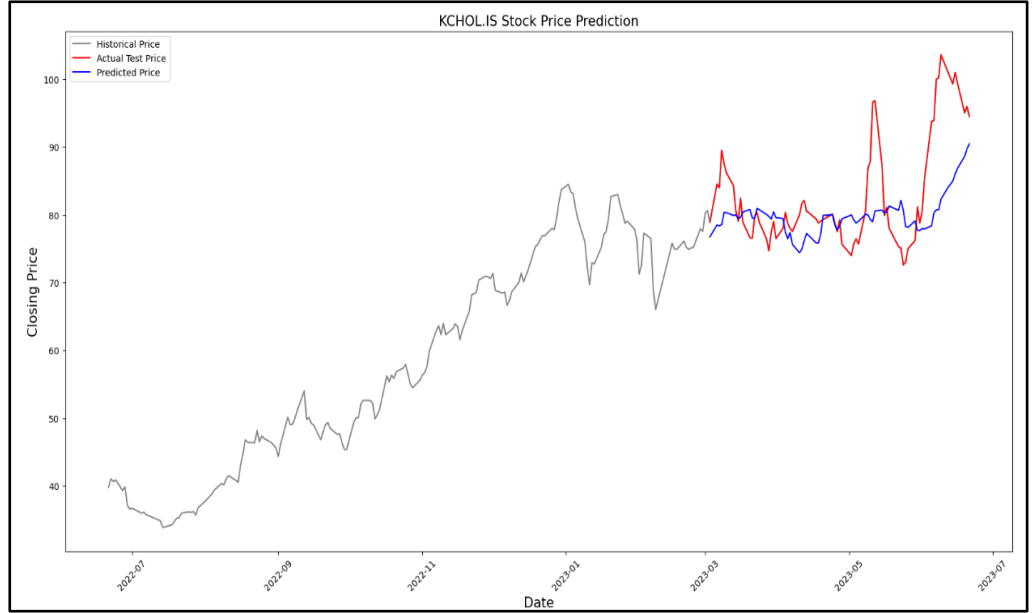
Şekil 7.10’da KCHOL.IS hissesinin grafiksel fiyat, tahmin grafiği görülmektedir.



Şekil 7.10 KCHOL.IS Hissenin Kapanış Değer Grafiği

- Veri seti, %70 eğitim ve %30 test verisi olacak şekilde ayrılır. İndeksleme kullanılarak, verinin belirli bir yüzdesi eğitim verisi olarak seçilirken, geri kalanı test verisi olarak seçilir.
- Veri, Min-Max ölçeklendirme yöntemi kullanılarak $[0, 1]$ aralığına ölçeklendirilir. Bu, verinin tüm değerlerini belirli bir aralığa dönüştürerek modelin daha iyi performans göstermesini sağlar.
- Eğitim verisi, zaman serisi öngörü modeli için uygun hale getirilir. 60 günlük bir pencere kullanarak, her biri 60 gün süren bir örnek olacak şekilde veri dilimleri oluşturulur. X_{train} , her bir örnekteki girdi verilerini (60 günlük pencere), y_{train} ise hedef çıktıları (bir sonraki kapanış fiyatı) temsil eder.
- MLPRegressor sınıfı kullanılarak bir çok katmanlı algılayıcı (MLP) regresyon modeli oluşturulur ve eğitilir. 100'şer nöron içeren 2 gizli katman kullanılır ve aktivasyon fonksiyonu olarak ReLU kullanılır.
- Test verisi, eğitim verisiyle aynı dönüşümlerden geçirilir. Son 60 gün veri alınarak modelin tahmin yapması için gerekli olan giriş verileri oluşturulur.
- Gerçek ve tahmin edilen hisse senedi fiyatlarının bir zaman serisi grafiği oluşturulur. Gerçek fiyatlar gri renkte gösterilirken, tahmin edilen fiyatlar mavi renkte gösterilir.

Şekil 7.11'de KCHOL.IS hissesinin tahmin grafiği verilmiştir.



Şekil 7.11 KCHOL.IS Hissenin Tahmin Grafiği

Modelin performansını değerlendirmek için bazı hata ölçütleri hesaplanır. MAE (Mean Absolute Error) ve MSE (Mean Squared Error), tahminlerin gerçek değerlerden ne kadar uzak olduğunu ölçerken, MAPE (Mean Absolute Percentage Error), tahmin hatalarının yüzdesel olarak ne kadar olduğunu ölçer. Sonuçlar ekrana yazdırılır. Şekil 7.12’de performans metrikleri görülmektedir.

```
Mean Absolute Error (MAE): 5.437
Mean Squared Error (MSE): 56.825
Mean Absolute Percentage Error (MAPE): 6.163
```

Şekil 7.12 Performans Metrikleri

7.9 Modellerin Karşılaştırılması

Aşağıdaki Tablo 7.1’de literatürdeki çalışmalardaki metrikler ile bizim çalışmamızdaki metriklerin karşılaştırılması sunulmuştur. Bu karşılaştırmalarda literatür çalışmalarındaki aynı veri setleri kullanılmıştır.

Tablo 7.1 Literatür ile Karşılaştırma

Model	Bizim Başarımımız	Önceki Başarım(lar)	Kaynak
DVM	Accuracy: %68	Accuracy: %70	Z. Hu, J. Zhu and K. Tse, "Stocks market prediction using Support Vector Machine," 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering, Xi'an, China, 2013, pp. 115-118, doi: 10.1109/ICIIM.2013.6703096.
NB	Accuracy: %85	Accuracy: %77	Ampomah, E. K., Nyame, G., Qin, Z., Addo, P. C., Gyamfi, E. O., & Gyan, M. (2021). Stock market prediction with gaussian naïve bayes machine learning algorithm. <i>Informatica</i> , 45(2).
ANN	RMSE: 553.13	RMSE: 1311.65	KARAATLI, M., Güngör, İ., Demir, Y., & KALAYCI, Ş. (2005). Hisse senedi fiyat hareketlerinin yapay sinir ağıları yöntemi ile tahmin edilmesi. <i>Journal of Management and Economics Research</i> , 3(3), 38-48.
LSTM	RMSE: 0.10	RMSE: 0.13	TOPRAK, Ş., ÇAĞIL, G., & KÖKÇAM, A. H. (2023). Stock Closing Price Prediction with Machine Learning Algorithms: PETKM Stock Example In BIST. <i>Düzce Üniversitesi Bilim ve Teknoloji Dergisi</i> , 11(2), 958-976.
ARIMA	RMSE: 87.3	RMSE: 148.8	TEJOSANTOSO, C. K. (2023). <i>EVALUATING COAL STOCKS WITH ANN, RANDOM FOREST AND ARIMA MODEL</i> (Doctoral dissertation, Universitas Katolik Soegijapranata Semarang).

Aşağıdaki Tablo 7.2 ve Tablo 7.3’de ise projede kullanılan makine öğrenmesi tekniklerinin çeşitli endeksler üzerindeki tahminleri sonucu ortaya çıkan metriklerin karşılaştırması sunulmuştur.

Tablo 7.2 Modeller Arası Performans Karşılaştırılması

Metrik	Model	BIST30	BIST100	HALKB.IS
RMSE	DVM	367.67	340.17	1.34
	YSA	340.26	284.29	0.81
	LSTM	129.55	117.11	0.36
	ARIMA	772.61	675.26	2.36

Tablo 7.3 Modeller Arası Yönlendirme Performansının Karşılaştırılması

Metrik	Model	BIST30	BIST100	HALKB.IS
RMSE	DVM	0.53	0.52	0.48
	NB	0.28	0.20	0.36

8. SONUÇ VE ÇIKARIMLAR

Bu tez çalışması, hisse senedi fiyat tahminlemesi için farklı makine öğrenmesi tekniklerinin uygulanabilirliğini araştırmıştır. Farklı hisse senetleri üzerinde gerçekleştirilen deneylerle, modellerin performansı değerlendirilmiş ve aşağıdaki sonuçlar elde edilmiştir:

- DVM modeli, fiyat tahminlemesinde ve yönlendirmede göstermiş olduğu performans diğer modellere göre daha düşük olsada, performansının yeterli ve kullanılabilir olduğu açıktır.
- YSA modeli, tatmin edici bir performans göstermiştir. Ancak, daha yüksek doğruluk oranları elde edebilmek için daha fazla veri ve daha derin ağ yapılarına ihtiyaç duyulabilir.
- LSTM modeli ise en düşük hata oranına sahip olmuştur. Bu sonuç, LSTM'in hisse senedi fiyat tahmininde diğer modellere göre daha başarılı olduğunu göstermektedir. LSTM'in zaman içindeki gelişimleri takip etme yeteneği, daha doğru tahminler yapabilmesini sağlamaktadır.
- ARIMA modeli ise diğer modellere göre daha düşük performans göstermiştir. Bu durum, ARIMA'nın hisse senedi fiyat tahmininde diğer makine öğrenmesi tekniklerine kıyasla daha sınırlı bir başarıya sahip olduğunu göstermektedir.
- NB modeli, yönlendirmede en yüksek doğruluk oranına sahiptir. Bu sonuç, NB'nin hisse senedi fiyat artış azalış yönlendirmesinde başarılı bir seçenek olduğunu göstermektedir.

Sonuç olarak bu çalışma hisse senedi fiyat tahminlemesi için farklı makine öğrenmesi tekniklerini değerlendirmiştir. LSTM modeli fiyat tahmininde, NB modeli ise yönlendirme tahmininde diğer modellere göre daha yüksek doğruluk oranları ve daha düşük hata değerleriyle dikkat çekmişlerdir. Bu bulgular, hisse senedi piyasasında daha bilinçli kararlar almak isteyen yatırımcılara yardımcı olabilecek potansiyele sahiptir.

KAYNAKÇA

- [1] Öztürk, K., & Şahin, M. E. (2018). Yapay sinir ağları ve yapay zekâ'ya genel bir bakış. *Takvim-i Vekayi*, 6(2), 25-36.
- [2] Şişmanoğlu, G., Koçer, F., ÖNDE, M. A., & SAHİNGOZ, O. K. (2020). Derin öğrenme yöntemleri ile borsada fiyat tahmini. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 9(1), 434-445.
- [3] Elif Yıldırım. "Yapay sinir ağları", <https://www.veribilimiokulu.com/yapay-sinir-agiartificial-neural-network-nedir/> (2020) (Son Erişim: 8 Aralık 2022)
- [4] Vural, B. B. (2007). YAPAY SİNİR AĞLARI İLE FİNANSAL TAHMİN.
- [5] "Borsa Nedir? Borsada Nasıl İşlem Yapılır?", <https://www.gcmyatirim.com.tr/borsa/borsa-nedir> (Son Erişim: 8 Aralık 2022)
- [6] Evren Deniz. "Borsa İstanbul'da Devre Kesici Nedir? BİST Devre Kesici Nasıl Çalışır?" <https://www.gcmyatirim.com.tr/egitim/makaleler/borsa-istanbul-da-devre-kesici-nedir-bist-devre-kesici-nasil-calisir> (2021) (Son Erişim: 2 Haziran 2023)
- [7] Midas Finansal Teknolojiler. "Borsa Nedir? Borsa Çeşitleri Nelerdir?" <https://www.getmidas.com/midas-akademi/borsa-nedir/> (Son Erişim: 2 Haziran 2023)
- [8] "Veri Madenciliği", https://ybs.deu.edu.tr/e-kitap/veri_madenciligi_kitap.pdf (Son Erişim : 8 Aralık 2022)
- [9] Ünsal, Ö. (2020). VERİ MADENCİLİĞİ TEKNİKLERİ İLE HİSSE SENETLERİ ARASINDAKİ FİYAT ETKİLEŞİMLERİNİN BELİRLENMESİ . *Mühendislik Bilimleri ve Tasarım Dergisi* , Special Issue: International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAME 2020) , 106-112.
- [10] Aksoy, B. (2021). Pay senedi fiyat yönünün makine öğrenmesi yöntemleri ile tahmini: Borsa İstanbul örneği. *Business and Economics Research Journal*, 12(1), 89-110.
- [11] ARSLANKAYA, S., & TOPRAK, Ş. (2021). Makine Öğrenmesi ve Derin Öğrenme Algoritmalarını Kullanarak Hisse Senedi Fiyat Tahmini. *International Journal of Engineering Research and Development*, 13(1), 178-192.
- [12] "Supervised vs Unsupervised vs Reinforcement Learning – The fundamental differences" , <https://starship-knowledge.com/supervised-vs-unsupervised-vs-reinforcement> (Son Erişim: 10.06.2023)
- [13] AKŞEHİR, Z. D., & KILIÇ, E. (2019). Makine öğrenmesi teknikleri ile banka hisse senetlerinin fiyat tahmini. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 12(2), 30-39.
- [14] Şişmanoğlu, G., Koçer, F., ÖNDE, M. A., & SAHİNGOZ, O. K. (2020). Derin öğrenme yöntemleri ile borsada fiyat tahmini. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 9(1), 434-445.

- [15] DALKIRAN, İ., & Mehmet, O. Z. A. N. (2022). Derin Öğrenme Teknikleri Kullanılarak Borsadaki Hisse Değerlerinin Tahmin Edilmesi. *Avrupa Bilim ve Teknoloji Dergisi*, (39), 143-148.
- [16] Arslan, M. E., & KIRCI, P. (2021). Makine Öğrenmesi İle Borsa Analizi. *Avrupa Bilim ve Teknoloji Dergisi*, (28), 1117-1120.
- [17] Koç Ustalı, N. , Tosun, N. & Tosun, Ö. (2021). Makine Öğrenmesi Teknikleri İle Hisse Senedi Fiyat Tahmini . *Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Dergisi* , 16 (1) , 1-16.
- [18] Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. *Procedia Computer Science*, 170, 1168-1173.
- [19] “Naïve Bayes Sınıflandırma Algoritması” , <https://kodedu.com/2014/05/naive-bayessiniflandirma-algoritmasi/> (Son Erişim: 27 Aralık 2022)
- [20] Huang, W., Nakamori, Y., & Wang, S. Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & operations research*, 32(10), 2513- 2522.
- [21] Wang, L. (Ed.). (2005). Support vector machines: theory and applications (Vol. 177). Springer Science & Business Media.
- [22] “<https://www.dermedya.com/storage/app/media/uploaded-files/15-08zauaonxsdiflzwu.png>” (Erişim tarihi: 3 Mart 2023)
- [23] Amazon Web Services. “Sinir Ağı Nedir?”, <https://aws.amazon.com/tr/what-is/neural-network/> (Erişim tarihi: 3 Nisan 2023)
- [24] Savaş, S., Topaloğlu, N., & YILMAZ, M. (2012). Veri madenciliği ve Türkiye’deki uygulama örnekleri. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 11(21), 1-23.
- [25] Data Overload. “Sliding Window Technique — reduce the complexity of your algorithm”, <https://medium.com/@data-overload/sliding-window-technique-reduce-the-complexity-of-your-algorithm-5badb2cf432f> (2022) (Erişim tarihi: 22 Haziran 2023)
- [26] Annie Liao. “Understanding the Sliding Window Technique in Algorithms”, <https://levelup.gitconnected.com/understanding-the-sliding-window-technique-in-algorithms-c6c3bf8226da> (2020) (Erişim tarihi: 20 Haziran 2023)
- [27] “<https://i.stack.imgur.com/2Dneo.png>” (Erişim tarihi: 19 Haziran 2023)
- [28] “https://files.codingninjas.in/article_images/understanding-an-rnn-cell-2-1640774918.webp” (Erişim tarihi:20 Haziran 2023)

- [29] “https://miro.medium.com/v2/resize:fit:764/1*6QnPUSv_t9BY9Fv8_aLb-Q.png” (Erişim tarihi: 19 Haziran 2023)
- [30] Wikipedia. “Autoregressive integrated moving average”, https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average (Erişim tarihi: 20 Haziran 2023)
- [31] Seray Erol. “Zaman Serilerinde Tahminleme Analizleri : ARIMA”, <https://www.martechsociety.com/zaman-serilerinde-tahminleme-analizleri-arima/> (2020) (Erişim tarihi: 21 Haziran 2023)
- [32] Seray Erol. “Zaman Serilerinde Tahminleme Analizleri : ARIMA”, <https://www.martechsociety.com/zaman-serilerinde-tahminleme-analizleri-arima/> (2020) (Erişim tarihi: 21 Haziran 2023)
- [33] Raghav Aggarwal. “Bi-LSTM”, <https://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0> (2019) (Erişim tarihi: 20 Haziran 2023)
- [34] “AKBAŞ, M. Ç. YAPAY SİNİR AĞLARI TAHMİNLEME MODELİ İLE HİSSE SENEDİ FİYAT ENDEKSİ TAHMİNLEMESİ ÜZERİNE BİR ÇALIŞMA. *Antalya Bilim Üniversitesi Uluslararası Sosyal Bilimler Dergisi*, 3(2), 97-110.”
- [35] “Balcıoğlu, Y. S., & Gözel269, A. 2000-2019 Yılları Arasındaki İspanya Borsa Verilerinden Telefonice Firmasının Yapay Sinir Ağları Yöntemleri ile Karşılaştırmalı Analizi.”
- [36] Databricks Inc. “What is a Jupyter Notebook?”, <https://www.databricks.com/glossary/jupyter-notebook> (Erişim tarihi: 21 Haziran 2023)
- [37] Benjamin Pryke. “How to Use Jupyter Notebook: A Beginner’s Tutorial”, <https://www.dataquest.io/blog/jupyter-notebook-tutorial/> (2020) (Erişim tarihi:21 Haziran 2023)
- [38] Python Software Foundation. <https://www.python.org/> (Erişim tarihi: 21 Haziran 2023)
- [39] Coursera. “What Is Python Used For? A Beginner’s Guide”, <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python> (2023) (Erişim tarihi: 21 Haziran 2023)
- [40] Python Software Foundation. “yfinance 0.2.21”, <https://pypi.org/project/yfinance/> (Erişim tarihi: 21 Haziran 2023)
- [41] Burcu Koçulu. “Scikit-Learn Yapısı Kolay Açıklamalı Anlatım (Örneklerle)”, <https://burcukoculu.medium.com/scikit-learn-yap%C4%B1s%C4%B1-kolay-a%C3%A7%C4%B1klamal%C4%B1-anlat%C4%B1m-%C3%B6rneklerle-afbb4593e5> (2021) (Erişim tarihi: 3 Mart 2009)
- [42] Özgür Doğan. “Keras Kütüphanesi Nedir? Derin Öğrenme Modeli Oluşturma”, <https://teknoloji.org/keras-kutuphanesi-nedir-derin-ogrenme-modeli-olusturma/> (2020) (Erişim tarihi: 21 Haziran 2023)
- [43]Serdar Yegulalp. “What is TensorFlow? The machine learning library explained”, <https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html> (2022) (Erişim tarihi: 21 Haziran 2023)

[44] Onur KIZILARSLAN. “Matplotlib Kütüphanesi Nedir?”, <https://imonur.github.io/blog/matplotlib/2023/03/07/matplotlib-nedir.html> (2023) (Erişim tarihi: 21 Haziran 2023)