

T.C.
ADNAN MENDERES ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
2025-DR-...

**HİSSE SENEDİ FİYAT TAHMİNİ VE PORTFÖY
OPTİMİZASYONUNDA YAPAY ZEKÂYA DAYALI HİBRİT
BİR YAKLAŞIM: BORSA İSTANBUL'DA BİR UYGULAMA**

HAZIRLAYAN
Erhan KOCA

TEZ DANIŞMANI
Doç. Dr. Engin ÇAKIR

AYDIN- 2025

T.C.
ADNAN MENDERES ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE
AYDIN

Bu tezde sunulan tüm bilgi ve sonuçların, bilimsel yöntemlerle yürütülen gerçek deney ve gözlemler çerçevesinde tarafımdan elde edildiğini, çalışmada bana ait olmayan tüm veri, düşünce, sonuç ve bilgilere bilimsel etik kuralların gereği olarak eksiksiz şekilde uygun atıf yaptığımı ve kaynak göstererek belirttiğimi beyan ederim.

... / 01 / 2025

Erhan KOCA

ÖZET

HİSSE SENEDİ FİYAT TAHMİNİ VE PORTFÖY OPTİMİZASYONUNDA YAPAY ZEKÂYA DAYALI HİBRİT BİR YAKLAŞIM: BORSA İSTANBUL'DA BİR UYGULAMA

Erhan KOCA

Doktora Tezi, İşletme Anabilim Dalı

Tez Danışmanı: Doç. Dr. Engin ÇAKIR

2025, XVII+186 sayfa

Günümüzde küreselleşme ve dijital dönüşüm süreçleri, finansal piyasalardaki volatilitiyi artırmakta ve yatırımcıların doğru kararlar alabilmesi için geleneksel finans teorilerinin ötesinde veri odaklı yaklaşımlara ihtiyaç duyulmaktadır. Bu bağlamda, çalışma kapsamında Borsa İstanbul BIST 30 endeksinde işlem gören hisse senetlerinin geçmiş verileri analiz edilerek fiyat tahminleri yapılmış ve optimal portföy kompozisyonu belirlenmiştir.

Çalışmanın ilk aşamasında, Rastgele orman algoritması, Ekstrem Gradyan Artırma, Destek Vektör Regresyon ve Uzun Kısa Dönemli Bellek olmak üzere dört farklı model kullanılarak hisse senedi fiyatlarının tahmini gerçekleştirilmiş ve modelin başarısı hata metrikleri ile değerlendirilmiştir. Yapılan analizler sonucunda, Rastgele orman algoritması diğer tekniklere kıyasla daha yüksek doğruluk oranlarına ulaştığını göstermiştir. Hisse bazında gerçekleştirilen analizlerde de AEFES. IS, ASELS. IS ve THYAO.IS gibi bazı pay senetlerinde yüksek öngörü başarısı dikkat çekmiştir. Çalışmanın ikinci aşamasında, yatırımcılar için en uygun portföy kompozisyonunu belirlemek amacıyla Tabu Arama Algoritması kullanılmıştır. Elde edilen sonuçlar, yatırımcıların risk-getiri tercihlerine uygun şekilde maksimum getiri ve minimum risk hedeflerine ulaşmalarını sağlayacak portföy kompozisyonlarının başarıyla kurgulandığını göstermektedir. Özellikle ASELS. IS, BIMAS.IS, EREGL.IS ve SISE.IS gibi düşük risk profiliyle öne çıkan varlıkların optimum portföyde ağırlık kazandığı tespit edilmiştir. Geleneksel portföy optimizasyonu yaklaşımlarının aksine, yapay zekâ tabanlı optimizasyon yöntemleri, piyasa verilerinin doğrusal olmayan ilişkilerini daha iyi modelleyerek yatırımcılar için daha etkin stratejiler sunmaktadır.

Elde edilen bulgular, yapay zekâ ve optimizasyon algoritmalarının finansal piyasalardaki karar alma süreçlerini önemli ölçüde iyileştirdiğini ve yatırım risklerini minimize ederken getiri potansiyelini artırdığını göstermektedir. Çalışma, yatırımcıların daha bilinçli ve veri temelli kararlar almasını sağlamanın yanı sıra, finans literatürüne de önemli katkılar sunmaktadır. Yapay zekâ tabanlı finansal analizlerin yaygınlaşması, piyasaların daha öngörülebilir hale gelmesini sağlayarak yatırımcıların volatil piyasa koşullarına daha esnek ve bilinçli bir şekilde uyum sağlamasına olanak tanıyacaktır.

Bu araştırma, yapay zekânın finansal piyasalardaki rolünü daha iyi anlamak ve yatırım kararlarının doğruluğunu artırmak isteyen akademisyenler, analistler ve yatırımcılar için önemli bir kaynak niteliği taşımaktadır. Gelecekte, derin öğrenme ve hibrit optimizasyon modellerinin finansal piyasalarda nasıl daha etkin kullanılabileceğini araştıran çalışmaların finansal karar alma süreçlerine daha fazla katkı sunacağı öngörülmektedir. Bu doğrultuda, yapay zekâ ve veri analitiği tekniklerinin finans sektörüne entegrasyonu, modern yatırım stratejilerinin temel unsurlarından biri haline gelmeye devam edecektir.

ANAHTAR KELİMELE: Finansal piyasalar, Portföy optimizasyonu, Tahminleme, Yapay zekâ.

ABSTRACT

A HYBRID ARTIFICIAL INTELLIGENCE - BASED APPROACH TO STOCK PRICE PREDICTION AND PORTFOLIO OPTIMIZATION: A CASE STUDY ON BORSA ISTANBUL

Erhan KOCA

Doctoral Thesis Department of Business Administration

Supervisor: Assoc. Prof. Dr. Engin ÇAKIR

2025, XVII+186 pages

In today's world, globalization and digital transformation processes are increasing volatility in financial markets, necessitating data-driven approaches that go beyond traditional financial theories to enable investors to make informed decisions. In this context, this study analyzes historical data from stocks listed on the Borsa Istanbul BIST 30 index to generate price forecasts and determine an optimal portfolio composition.

In the first phase of the study, four different models Random Forest, Extreme Gradient Boosting (XGBoost), Support Vector Regression (SVR), and Long Short-Term Memory (LSTM) were employed to predict stock prices, and their performances were evaluated using error metrics. The analysis revealed that the Random Forest algorithm outperformed the other techniques, achieving higher accuracy rates. Notably, stocks such as AEFES. IS, ASELS. IS, and THYAO. IS demonstrated particularly high prediction performance in the stock-specific analyses.

In the first phase of the study, four different models Random Forest, Extreme Gradient Boosting (XGBoost), Support Vector Regression (SVR), and Long Short-Term Memory (LSTM) were employed to predict stock prices, and their performances were evaluated using error metrics. The analysis revealed that the Random Forest algorithm outperformed the other techniques, achieving higher accuracy rates. Notably, stocks such as AEFES. IS, ASELS. IS, and THYAO. IS demonstrated particularly high prediction performance in the stock-specific analyses.

The findings indicate that artificial intelligence and optimization algorithms significantly enhance decision-making processes in financial markets by minimizing investment risks and maximizing return potential. In addition to enabling investors to make more informed and data-driven decisions, this study makes a valuable contribution to the

financial literature. The increasing adoption of AI-based financial analysis is expected to improve market predictability and allow investors to adapt more flexibly and consciously to volatile market conditions.

This research serves as an important resource for academics, analysts, and investors who seek to better understand the role of artificial intelligence in financial markets and improve the accuracy of investment decisions. Looking ahead, studies that explore how deep learning and hybrid optimization models can be more effectively applied in financial markets are anticipated to further enhance financial decision-making processes. In this direction, the integration of artificial intelligence and data analytics techniques into the financial sector will continue to be a fundamental component of modern investment strategies.

KEY WORDS: Financial markets, Portfolio optimization, Forecasting, Artificial intelligence.

ÖNSÖZ

Bu doktora tezi, yapay zekâ tabanlı yaklaşımların finansal piyasalardaki kullanım potansiyelini değerlendirmek amacıyla hazırlanmış olup, yalnızca akademik ve bilimsel bir çalışma niteliği taşımaktadır. Çalışmada yer alan analizler, herhangi bir yatırım tavsiyesi veya yönlendirmesi içermemektedir.

Araştırma sürecinde bilgi, tecrübe ve yönlendirmeleriyle bana yol gösteren çok değerli danışmanım Doç. Dr. Engin Çakır'a en içten teşekkürlerimi sunarım. Tez izleme komitesinde yer alan Prof. Dr. Hüseyin Şenkayas ve Doç. Dr. Gülşah Sezen'e kıymetli katkıları için minnettarım. Ayrıca savunma jürisinde yer alarak değerlendirmeleriyle tezin olgunlaşmasına katkı sunan Prof. Dr. Mehpere Timor ve Dr. Öğr. Üyesi Veli Rıza Kalfa'ya da teşekkür ederim.

Bu uzun ve zorlu süreçte her zaman yanımda olan sevgili eşim Duygu'ya ve hayatıma neşe katan kıymetli oğlum Doruk'a sabırları, anlayışları ve sevgileri için gönülden teşekkür ederim. Ayrıca her daim maddi ve manevi desteğini esirgemeyen, varlıklarıyla bana güç veren aileme şükranlarımı sunarım.

Erhan KOCA

İÇİNDEKİLER

KABUL VE ONAY.....	iii
BİLİMSEL ETİK BİLDİRİM.....	iv
ÖZET	v
ABSTRACT	vii
ÖNSÖZ	ix
İÇİNDEKİLER.....	x
ŞEKİLLER DİZİNİ.....	xiii
TABLolar DİZİNİ.....	xiv
EKLER DİZİNİ	xvi
KISALTMALAR DİZİNİ.....	xvii
GİRİŞ.....	1
1. BÖLÜM	1
1. FİNANSAL PİYASALAR.....	5
1.1. Finansal Piyasaların Tanımı	5
1.2. Finansal Piyasa Türleri	7
1.2.1. Vade Bakımından Finansal Piyasalar	7
1.2.2. Örgütlenme Yapısı Bakımından Finansal Piyasalar.....	8
1.2.3. İşlem Aşaması Bakımından Finansal Piyasalar.....	9
1.2.4. İşlemin Gerçekleşme Zamanı Bakımından Finansal Piyasalar	10
1.2.5. Finansal Varlıkların Türleri Bakımından Finansal Piyasalar	11
1.3. Yatırım ve Hisse Senedi Kavramı	12
1.4. Portföy Kavramı ve Portföy Yönetimi	14
1.4.1. Portföy Kavramı	14
1.4.2. Portföy Yönetimi	16
1.4.3. Portföy Teorileri ve Yöntemleri	17

2. YAPAY ZEKÂ VE TAHMİNLEME	21
2.1. Yapay zekâ Tanımı.....	22
2.2. Yapay zekâ Uygulama Alanları.....	22
2.3. Makine Öğrenmesi	23
2.3.1. Denetimli Öğrenme	24
2.3.2. Denetimsiz Öğrenme	25
2.3.3. Pekiştirmeli Öğrenme	26
2.4. Yapay zekâ Algoritmaları.....	27
2.4.1. Yapay sinir ağları	27
2.4.2. Uzun Kısa Dönemli Bellek.....	30
2.4.3. Lineer Regresyon.....	32
2.4.4. Karar Ağaçları	33
2.4.5. Genetik Algoritma	35
2.4.6. Parçacık Sürü Algoritması.....	36
2.4.7. Yapay Arı Kolonisi.....	38
2.4.8. Rastgele Orman Algoritması	39
2.4.9. Tabu Arama	41
2.4.10. Karınca Kolonisi Algoritması.....	42
2.4.11. Ekstra Gradyan Artırma Algoritması	43
2.5. Literatürde Yer Alan Çalışmalar	45
3. HİSSE SENEDİNİN FİYAT TAHMİNİ VE PORTFÖY OPTİMİZASYONU	61
3.1. Araştırmanın Amacı ve Önemi.....	61
3.2. Araştırmanın Yöntemi ve Kapsamı	62
3.3. Tahmin Bölümünde Yapılanlar	63
3.3.1. Veri Hazırlığı.....	63
3.3.2. Tahminlemede Kullanılacak Araçların Hazırlığı	65

3.3.3. Modelin Doğruluğunu Değerlendiren Performans Göstergeleri	66
3.4. Tahmin Bölümünde Elde Edilen Bulgular	69
3.5. Optimizasyonun Hazırlık Süreci	102
3.5.1. Veri Hazırlığı ve Model Çerçevesinin Oluşturulması	102
3.5.2. Optimizasyonun Performans Metrikleri	103
3.6. Optimizasyon Bölümünde Elde Edilen Bulgular	105
4. TARTIŞMA SONUÇ ÖNERİLER	111
5. KAYNAKLAR.....	115
EKLER	138
ÖZGEÇMİŞ.....	186

ŞEKİLLER DİZİNİ

Şekil 1.1. Finansal piyasa türleri	7
Şekil 2.1. Rastgele Orman Algoritması.....	40



TABLÖLAR DİZİNİ

Tablo 3.1. 2025 yılının ikinci üç aylık döneminde BIST 30 endeksindeki hisse senetleri...	64
Tablo 3.2. MAPE performans tablosu	67
Tablo 3.3. Modellerin tahmin doğruluğu	70
Tablo 3.4. Hisse senetlerinin performans değerleri	70
Tablo 3.5. AEFES.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri..	73
Tablo 3.6. AKBNK.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri	74
Tablo 3.7. ASELS.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri..	75
Tablo 3.8. ASTOR.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri .	76
Tablo 3.9. BIMAS.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri..	77
Tablo 3.10. CIMS.A.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri	78
Tablo 3.11. EKGYO.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri... ..	79
Tablo 3.12. ENKAI.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri	80
Tablo 3.13. EREGL.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri	81
Tablo 3.14. FROTO.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri	82
Tablo 3.15. GARAN.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri... ..	83
Tablo 3.16. HEKTS.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri	84
Tablo 3.17. ISCTR.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri .	85
Tablo 3.18. KCHOL.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri... ..	86
Tablo 3.19. KOZAL.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri... ..	87
Tablo 3.20. KRDMD.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri... ..	88

Tablo 3.21. MGROS.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri.....	89
Tablo 3.22. PETKM.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri...	90
Tablo 3.23. PGSUS.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri	91
Tablo 3.24. SAHOL.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri	92
Tablo 3.25. SASA.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri..	93
Tablo 3.26. SISE.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri....	94
Tablo 3.27. TAVHL.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri.....	95
Tablo 3.28. TCELL.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri	96
Tablo 3.29. THYAO.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri...	97
Tablo 3.30. TOASO.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri	98
Tablo 3.31. TTKOM.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri...	99
Tablo 3.32. TUPRS.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri.....	100
Tablo 3.33. ULKER.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri.....	101
Tablo 3.34. YKBNK.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri...	102
Tablo 3.35. Optimum portföy	106
Tablo 3.36. Optimum portföyün performans değerleri.....	107
Tablo 3.37. Maksimum getiriye sahip portföy	108
Tablo 3.38. Minimum riske sahip portföy	108

EKLER DİZİNİ

Ek 1. BİST 30 endeksindeki hisse senetlerinin tahmin edilen değerleri ile gerçek değerlerinin grafileri	138
Ek 2. BİST 30 da yer alan hisse senetlerinin eğitim, test ve tahmin grafikleri.....	149
Ek 3. Kullanılan kodlar	159



KISALTMALAR DİZİNİ

ACO	: Karınca Kolonisi Optimizasyonu (Ant Colony Optimization)
APT	: Arbitraj Fiyatlama Modeli
CAPM	: Sermaye Varlıklarını Fiyatlama Modeli (Capital Asset Pricing Model)
GA	: Genetik Algoritma (Genetic Algoritim)
LSTM	: Uzun Kısa Dönemli Hafıza (Long Short Term Memory)
MAE	: Ortalama Mutlak Hata (Mean Absolute Error)
MAPE	: Ortalama Mutlak Yüzde Hatası (Mean Absolute
MPT	: Modern Portföy Teorisi (Modern Portfolio Theory)
MSE	: Ortalama Kare Hata (Mean Square Error)
NLP	: Doğal Dil İşleme (Natural Language Processing)
PSO	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
RF	: Rastgele Orman Algoritması (Random Forest Algorithm)
RMSE	: Kök Ortalama Kare Hatası (Root Mean Square Error)
R²	: Belirlilik Katsayısı (Cooefficient of Determination)
SVM	: Destek Vektör Makineleri (Support Vector Machine)
SVR	: Destek Vektör Regresyon (Support Vector Regression)
TS	: Tabu Arama (Tabu Search)
XGBoost	: Ekstrem Gradyan Artırma (eXtreme Gradient Boosting)
YAK	: Yapay Arı Kolonisi (Artificial Bee Colony Algoritim)
YSA	: Yapay Sinir Ağları (Artificial Neural Network)

GİRİŞ

Günümüz finansal piyasaları, küreselleşme ve dijital dönüşüm süreçleriyle birlikte giderek daha karmaşık bir yapıya bürünmektedir. Yatırımcıların doğru kararlar alabilmesi ve risklerini minimize edebilmesi için geleneksel finansal analiz yöntemlerinin yanı sıra ileri düzey veri analitiği ve yapay zekâ tekniklerine duyulan ihtiyaç giderek artmaktadır. Finansal piyasalar, ekonomik kaynakların etkin bir şekilde dağıtılmasını sağlamak ve piyasa katılımcıları için yatırım fırsatları yaratmaktadır. Bu noktada, yapay zekâ ve makine öğrenmesi yöntemleri, yatırım stratejilerinin optimize edilmesi ve finansal tahminlerin daha isabetli yapılması açısından önemli fırsatlar sunmaktadır. Yapay zekâ, büyük veri setlerini işleyerek karmaşık ilişkileri tespit edebilme ve geleceğe yönelik tahminlerde bulunabilme yeteneğiyle finans alanında giderek daha fazla benimsenmektedir. Bu çalışmada, yapay zekâ tabanlı tahmin ve optimizasyon algoritmalarının finansal piyasalar bağlamında nasıl etkin şekilde kullanılabileceği araştırılmaktadır. Çalışmanın odak noktası, hisse senedi fiyatlarının tahmini ve portföy optimizasyonu üzerine kurulmuş olup, yatırım kararlarını daha bilinçli hale getirecek modellerin geliştirilmesine yöneliktir.

Finansal piyasalar, yatırımcıların fonlarını en verimli şekilde değerlendirebilmeleri için çeşitli enstrümanlar sunan platformlardır. Para piyasaları, sermaye piyasaları, türev piyasalar ve döviz piyasaları gibi alt kategoriler, yatırımcıların risk ve getiri beklentilerine uygun yatırım yapmalarına olanak tanır. Özellikle hisse senedi piyasaları, uzun vadeli yatırım stratejileri açısından kritik öneme sahiptir. Hisse senetleri, yatırımcılara şirketlerin büyüme potansiyelinden faydalanma fırsatı sunarken, aynı zamanda dalgalı piyasa koşulları nedeniyle yüksek risk faktörleri içermektedir.

Finansal piyasalarda yatırım kararlarını şekillendiren temel faktörler arasında makroekonomik göstergeler, şirket finansal performansları, yatırımcı psikolojisi ve küresel ekonomik gelişmeler yer almaktadır. Geleneksel portföy teorileri, yatırımcıların risk-getiri dengesini göz önünde bulundurarak optimal portföy seçimi yapmalarını önermektedir. Ancak, modern finans anlayışında yapay zekâ ve büyük veri analitiği gibi teknolojiler sayesinde yatırımcılar daha veri odaklı ve isabetli kararlar alabilmektedir.

Yapay zekâ ve makine öğrenmesi, günümüzde finans sektöründe giderek daha fazla kullanılmaktadır. Özellikle büyük veri analitiği ve hesaplama gücünün artmasıyla birlikte,

yapay zekâ algoritmaları finansal tahminlerde doğruluk oranını artırmaktadır. Yapay zekâ, finansal piyasaların karmaşıklığını anlamaya ve büyük hacimli verileri işleyerek yatırımcılara anlamlı içgörüler sunmaya olanak tanımaktadır.

Makine öğrenmesi algoritmaları, finansal zaman serilerinin analizinde sıklıkla kullanılmaktadır. Yapay sinir ağları, rastgele ormanlar, SVM ve derin öğrenme modelleri gibi teknikler, yatırımcıların daha sağlıklı kararlar almasına yardımcı olmaktadır. Özellikle hisse senedi fiyat tahmini gibi değişken ve dinamik süreçlerde, yapay zekâ yöntemleri geleneksel istatistiksel yöntemlere göre daha başarılı sonuçlar elde edebilmektedir.

Portföy optimizasyonu, yatırımcıların belirli bir risk seviyesinde en yüksek getiriyi elde etmelerini sağlayan bir süreçtir. Geleneksel portföy optimizasyonu teknikleri, genellikle Markowitz'in Modern Portföy Teorisi gibi yöntemlere dayanmaktadır. Ancak bu yöntemler, piyasa verilerinin doğrusal varsayımlarına dayandığı için gerçek dünya koşullarında yetersiz kalabilmektedir.

Yapay zekâ destekli optimizasyon teknikleri, geleneksel yöntemlerin eksikliklerini gidermede etkili çözümler sunmaktadır. Genetik Algoritmalar, Parçacık Sürü Optimizasyonu ve Tabu Arama gibi yöntemler, yatırımcıların optimal portföy kompozisyonunu belirlemelerine yardımcı olmaktadır. Bu çalışmada, rastgele orman algoritması kullanılarak hisse senedi fiyat tahmini gerçekleştirilmekte ve tabu arama yöntemi ile optimal portföy oluşturulmaktadır. Böylece, yatırımcıların en yüksek getiriyi elde etmelerini sağlayacak portföy bileşimi belirlenmektedir.

Bu çalışmada, finansal piyasalarda hisse senedi fiyatlarının tahmini ve portföy optimizasyonu süreçlerinde yapay zekâ yöntemlerinin etkinliği incelenmektedir. Çalışma kapsamında, Borsa İstanbul BIST 30 endeksinde yer alan hisse senetlerinin geçmiş verileri analiz edilerek gelecekteki fiyat hareketleri tahmin edilmektedir.

Araştırmada, yapay zekâ tabanlı tahmin ve optimizasyon süreçleri iki aşamada ele alınmaktadır. İlk aşamada, Rastgele Orman Algoritması, Uzun Kısa Dönemli Bellek, Ektrem Gradyan Artırma ve Destek Vektör Regresyon kullanılarak BIST 30 endeksinde yer alan hisse senetlerinin kapanış fiyatlarının tahmin edilmektedir. Daha sonra tahmin doğruluğu yüksek olan model ile optimizasyona geçilmiştir. İkinci aşamada ise, Tabu

Arama yöntemi ile yatırımcılar için en uygun portföy kompozisyonu belirlenmektedir. Bu süreçte, model performansı hata kriterleri kullanılarak değerlendirilmektedir.

Bu araştırma, yapay zekâ yöntemlerinin finansal piyasalar üzerindeki etkilerini inceleyerek yatırımcılar için alternatif stratejiler geliştirmeyi hedeflemektedir. Literatürde son yıllarda yapay zekâ ve makine öğrenmesi algoritmalarının finansal tahminleme ve portföy yönetimi süreçlerine entegrasyonu üzerine çeşitli çalışmalar gerçekleştirilmiş olmakla birlikte, bu yaklaşımların çoğu genellikle fiyat tahminleme ya da portföy optimizasyonu alanlarında bağımsız uygulamalarla sınırlı kalmaktadır. Mevcut çalışmaların önemli bir kısmı, tahminleme modelleri ile optimizasyon süreçlerini birbirinden kopuk ele almakta ve bu iki temel süreci bütünleştiren kapsamlı sistematik model örneklerine literatürde yeterince rastlanmamaktadır.

Bu doğrultuda yürütülen çalışma, finansal piyasaların doğrusal olmayan yapısını dikkate alarak, hem fiyat tahminleme hem de portföy optimizasyonu bileşenlerini entegre eden bütüncül bir yapay zekâ tabanlı model önerisi geliştirmektedir. Farklı makine öğrenmesi algoritmaları kullanılarak oluşturulan fiyat tahmin sonuçları, doğrudan portföy optimizasyon sürecine entegre edilmekte; böylece yatırım kararlarının daha sağlam, öngörülebilir ve adaptif temellere dayandırılması sağlanmaktadır. Ayrıca, optimizasyon aşamasında uygulanan Tabu Arama algoritması sayesinde, geleneksel portföy teorilerinin kısıtlayıcı doğrusal kabullerinin ötesine geçilerek, yatırımcı risk profillerine göre daha esnek ve etkili portföy yapılandırmaları oluşturulabilmektedir.

Bu çalışma, finansal piyasalar, yapay zekâ ve makine öğrenmesi yöntemleri çerçevesinde gerçekleştirilen bir analiz sürecini kapsamaktadır ve dört ana bölümden oluşmaktadır. İlk bölümde, finansal piyasaların kavramsal çerçevesi ve teorik temelleri ele alınarak kapsamlı bir açıklama sunulacaktır. Bu bölümde ayrıca, çalışmanın temelini meydana getiren teorik altyapı oluşturulmuştur.

İkinci bölümde, yapay zekâ yöntemleri incelenecek olup, özellikle finansal verilerle çalışırken kullanılan temel teknikler detaylandırılmıştır. Bu çerçevede, denetimli ve denetimsiz öğrenme yöntemleri ve finansal tahmin modelleri gibi konular ele alınmıştır. Aynı zamanda, bu yöntemlerin finansal piyasalar üzerindeki etkileri, avantajları ve karşılaşılan zorluklar tartışılmıştır.

Üçüncü bölümde, araştırma metodolojisi ayrıntılı bir şekilde açıklanmış olup, çalışmada kullanılan veri seti tanıtılacak ve analiz için tercih edilen algoritmalar detaylandırılmıştır. Uygulama sonunda elde edilen bulgulara yer verilmiştir. Ayrıca, modelin başarısını değerlendirmek için kullanılan performans ölçütleri ve doğrulama yöntemleri de bu bölümde yer almıştır.

Son olarak, dördüncü bölümde çalışmanın genel sonuçları özetlenmiş ve elde edilen bulgular ışığında gelecekte yapılabilecek araştırmalar tartışılacaktır. Çalışmanın finansal analiz ve yapay zekâ uygulamaları açısından sağladığı katkılar ele alınarak, ilerleyen dönemlerde geliştirilebilecek modeller ve uygulanabilecek yeni yöntemler üzerine öneriler sunulacaktır.



1. BÖLÜM

1. FİNANSAL PİYASALAR

Finansal piyasalar, ekonomik sistemin en önemli yapı taşlarından biri olup, kaynakların etkin dağılımını sağlayarak ekonomik büyümeye ve kalkınmaya katkıda bulunur. Sermayenin en verimli alanlara yönlendirilmesi, işletmelerin yatırımlarını artırmasına ve istihdam oluşturmaya olanak tanır. Aynı zamanda, finansal piyasalar bireylerin ve kurumların tasarruflarını değerlendirmesine, yatırım yapmasına ve geleceğe yönelik finansal güvenlik sağlamasına yardımcı olur. Yatırımcılar için risk yönetimi imkânı sunarak fiyat dalgalanmalarına karşı korunmalarını sağlarken, vadeli işlemler ve türev piyasalar aracılığıyla belirsizlikleri minimize etmeye yardımcı olur. Bununla birlikte, finansal piyasalar likidite sağlayarak yatırımcıların varlıklarını kolayca nakde çevirebilmesine imkân tanır ve fiyat keşfi mekanizmasıyla piyasa katılımcılarına doğru kararlar alabilmeleri için gerekli bilgi akışını sunar (Tezel, 2016: 38).

Bu bölümde, finansal piyasaların tanımı ve türleri detaylı bir şekilde açıklanmıştır. Ayrıca yatırım ve hisse senedi kavramına değinilip, portföy teorileri ele alınacaktır.

1.1. Finansal Piyasaların Tanımı

Ekonomi literatüründe önemli bir yere sahip olan piyasa kavramı, mal ve hizmetlerin alınıp satılması için alıcılarla satıcıların bir araya geldiği platform olarak tanımlanır. Bu tanım, piyasanın ticari faaliyetlerde bulunmak üzere bir araya geldikleri ortam sadece fiziki değil sanal bir alan da olabileceğini vurgulamaktadır. Dolayısıyla önemli olan alıcı ve satıcılar arasındaki etkileşim ve işlemin hangi koşullar altında gerçekleştiğidir (Aydın, 2016: 51). Bu genel yapı içerisinde, finansal piyasalar, sermaye, borçlanma ve yatırım gibi finansal varlıkların el değiştirdiği özel bir alanı temsil etmektedir. Literatürde finansal piyasalar ile ilgili çeşitli tanımlar yer almaktadır.

Finansal piyasalar, işletmelerin veya fona ihtiyaç duyan kuruluşların fon talepleri ile fon sağlayanların sunduğu fonların bir araya geldiği piyasalar olarak tanımlanmaktadır (Sayılğan, 2017: 26).

Bir başka açıdan finansal piyasalar, ekonomik birimlerin sahip oldukları tasarrufları ihtiyaç duyan diğer birimlere yönlendirdiği ve kaynakların dağıtımını gerçekleştirdiği ortamlardır (Seyidoğlu, 2016: 27).

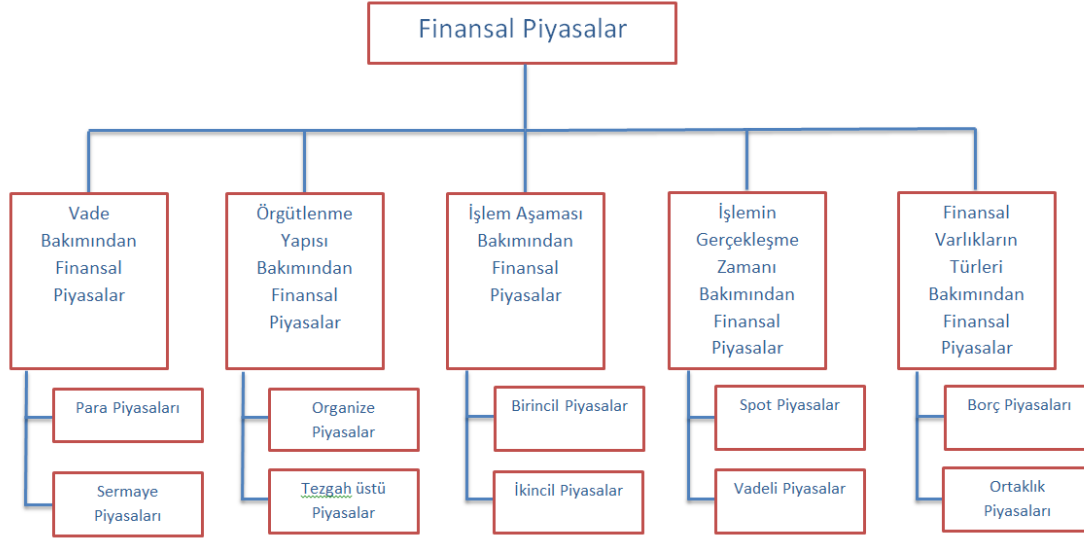
Yetkiner ve Korap (2020: 66) finansal piyasaları, tasarruf sahiplerinin biriktirdikleri kaynakları değerlendirerek yatırımcıların finansman gereksinimlerini karşıladığı platformlar olarak tanımlamaktadır.

Finansal piyasalarla ilgili bu tanımlar göz önünde bulundurulduğunda, finansal piyasaların fon akışını sağlayan ve ekonomik birimler arasında kaynak transferini mümkün kılan yapılar olduğunu göstermektedir. Böylece, hem bireylerin hem de kurumların ekonomik faaliyetlerini sürdürebilmeleri için gerekli finansmanı temin etmeleri sağlanır. Finansal piyasaların en temel işlevlerinden biri, farklı finansal araçların alım satımına olanak tanıyan bir ortam yaratmasıdır. Yatırımcılara farklı risk ve getiri profilleri sunan hisse senetleri, tahviller, döviz ve türev araçlar gibi çeşitli enstrümanlar, çeşitlendirilmiş yatırım fırsatları yaratmaktadır. Bu çeşitlilik, yatırımcıların risklerini yönetmelerine yardımcı olurken, aynı zamanda fon sağlama ihtiyacı duyan tarafların uygun maliyetlerle kaynak bulabilmesini sağlar (Zülfüoğlu, 2013: 33).

Finansal piyasalar, yalnızca yatırımcılar ve borçlular arasındaki işlemlerin kolaylaştırılmasında değil, aynı zamanda piyasa fiyatlarının oluşumunda da kritik bir görev üstlenmektedir. Bu fiyatlar, arz ve talep dengesine bağlı olarak şekillenir ve piyasa katılımcılarının ekonomik öngörülerini yansıtır. Bu sayede, piyasa fiyatları ekonomik durum hakkında bilgi sunar ve gelecekteki beklentilere göre karar verme sürecini kolaylaştırır. Sonuç olarak, iyi işleyen ve gelişmiş finansal piyasalar, ekonomik kararların daha isabetli ve sağlıklı bir biçimde alınmasına yardımcı olmaktadır. Ekonomik fırsatların daha verimli bir şekilde değerlendirilmesi açısından hem bireysel yatırımcılara hem de büyük kuruluşlara önemli destek sunar. Bu durum, ekonominin genel durumu üzerinde de olumlu bir etki yaratmakta, zira güçlü finansal piyasalar ekonomik büyüme, istihdam ve refah seviyelerinin yükselmesine katkıda bulunmaktadır (Levine, 2015).

1.2. Finansal Piyasa Türleri

Finansal piyasalar, farklı ölçütler dikkate alınarak beş grupta ele alınmaktadır.



Kaynak: Anadolu Üniversitesi Açıköğretim Fakültesi Yayını

Şekil 1.1. Finansal piyasa türleri

1.2.1. Vade Bakımından Finansal Piyasalar

Finansal piyasalar hakkında en yaygın şekilde benimsenen sınıflandırma, piyasada işlem gören fonların ve finansal varlıkların vadesine bağlı olarak yapılmaktadır. Bu sınıflandırma doğrultusunda finansal piyasalar, para piyasaları ve sermaye piyasaları olmak üzere ikiye ayrılmaktadır (İpekten, 2013: 10)

1.2.1.1. Para piyasaları

Para piyasaları, vadesi genellikle bir yıldan kısa olan ve kısa vadeli olarak tanımlanan fonlar ile finansal varlıkların alınıp satıldığı piyasalardır. Bu piyasalarda işlem gören finansal varlıkların vadesinin kısa olması, yüksek likiditeye sahip olmalarını sağlar. Bu durum, varlıkların düşük risk ve düşük getiri potansiyeline sahip olduğu anlamına gelir. Para piyasaları, kısa vadeli fon arz ve talebinin bir araya geldiği bir yapı olarak; geçici likidite sıkıntısı yaşayan ekonomik aktörlerle, fonlarını kısa vadeli değerlendirmek isteyen tasarruf sahiplerini buluşturan bir platform olma rolü üstlenir. Hazine bonoları, finansman bonoları ve mevduat sertifikaları bu piyasanın başlıca araçları arasında yer alır. Günlük

işlemler, senet takibi, tahakkuk ve iskonto işlemleri ile çekli işlemler de para piyasasının temel aktivitelerini oluşturur. Bu alanda devlet, özel sektör firmaları ve çeşitli finansal kurumların yanı sıra en önemli aktörler bankalardır (İpekten, 2013: 11).

1.2.1.2. Sermaye piyasaları

Sermaye piyasaları, işlem vadesi açısından bir yıldan uzun vadeli ve sürekli fon ile finansal araçların alınıp satıldığı piyasalar olarak tanımlanır. Bu piyasalardan elde edilen fonlar, işletmelerin uzun vadeli yatırımları için finansman kaynağı olarak kullanılır. Yeni işletmelerin kurulmasına veya mevcut işletmelerin genişleme ile yenileme yatırımlarına yönelik ihtiyaç duyulan finansman, sermaye piyasaları aracılığıyla karşılanır. Sağlanan bu fonlar, hisse senedi ihracı yoluyla temin edilen öz sermaye kaynakları olabileceği gibi, tahvil ve benzeri uzun vadeli borçlanma araçlarıyla elde edilen yabancı (dış) kaynaklar da olabilir.

Sermaye piyasalarında işlem gören finansal varlıklar genellikle uzun vadeli nitelikte olduğundan, likiditeleri kısa vadeli araçlara oranla daha düşüktür ve bu nedenle risk düzeyleri nispeten daha yüksektir. Bunun bir sonucu olarak, bu tür araçlara yatırım yapan tasarruf sahipleri genellikle yüksek getiri beklentisi taşır. Sermaye piyasalarının temel finansal araçları arasında hisse senetleri ve tahviller bulunur. Buna ek olarak ipotek senetleri, kâr ortaklığı belgeleri, kâr-zarar ortaklığı belgeleri gibi varlıklar da sermaye piyasası araçları arasında gösterilebilir (İpekten, 2013: 11).

1.2.2. Örgütlenme Yapısı Bakımından Finansal Piyasalar

Finansal piyasalar örgütlenme yapısı kriteri bakımından ele alındığında ikiye ayrılmaktadır.

1.2.2.1. Organize piyasalar

Belirli bir fiziksel mekâna sahip, işlemlerin belirli kurallar ve ilkeler doğrultusunda gerçekleştirildiği ve kendine ait araçlar ile kurumlar barındıran piyasalar organize piyasalar olarak adlandırılır. Organize piyasalar, yatırımcıların bilgilere hızlı, kolay ve düşük maliyetle ulaşabildiği; alıcı ve satıcıların yoğun bir şekilde yer aldığı için tam rekabet şartlarına daha yakın kabul edilen piyasalardır. Bu tür piyasalarda, ne alıcılar ne de satıcılar

tek başlarına piyasayı etkileme gücüne sahip değildir. Sonuç olarak, organize piyasalarda finansal varlıkların fiyatları daha doğru bir şekilde belirlenir (Aydın, 2018: 40).

1.2.2.2. Tezgah üstü piyasalar

Tezgâhüstü piyasalar, organize borsalarda işlem görme fırsatı bulamayan menkul kıymetlerin alınıp satıldığı aracı piyasalardır. Genellikle organize piyasaların kotasyon şartlarını karşılayamayan küçük ve yeni şirketlere ait menkul kıymetlerin işlem gördüğü bu piyasalar, daha esnek kurallarla faaliyet gösterir. Bu tür piyasalar arasında en tanınmış ve gelişmiş olanı, ABD'deki NASDAQ'tır (Doğukanlı, 2019: 6).

1.2.3. İşlem Aşaması Bakımından Finansal Piyasalar

Finansal piyasalar, finansal varlıkların alım ve satım işlemlerine bağlı olarak ikiye ayrılır (Mishkin, 2018: 25).

1.2.3.1. Birincil piyasalar

Birincil piyasalar, şirketler veya kamu kurumları tarafından tahvil ya da hisse senedi gibi menkul kıymetlerin ilk kez ihraç edilerek alıcılara satıldığı finansal piyasalardır. Birincil piyasalar kamuoyu tarafından pek bilinmemektedir. Bu piyasada, menkul kıymetlerin ilk kez satışa sunulmasında önemli bir rol oynayan başlıca finansal kurum yatırım bankasıdır. Yatırım bankası, bu süreçte menkul kıymetleri garanti altına alır. Garanti altına alma işlemini de şu şekilde yapar: Bir şirketin çıkardığı menkul kıymetler için bir fiyat taahhüdünde bulunur ve ardından bu menkul kıymetleri halka arz ederek satışını gerçekleştirir (Mishkin, 2018: 25).

1.2.3.2. İkincil piyasalar

Birincil piyasada satışı gerçekleştirilmiş finansal varlıkların yeniden alım-satım işlemlerine konu edildiği piyasalardır. Genellikle ikinci el piyasa olarak adlandırılan bu tür piyasalarda, daha önce ihraç edilmiş finansal varlıklar yatırımcılar arasında el değiştirir (Mishkin, 2018: 25). Örneğin, Hazine tarafından yeni bir bono ya da tahvil çıkarılması veya bir şirketin yeni hisse senetleri ihraç etmesi birincil piyasa işlemi olarak kabul edilir. Buna karşılık, mevcut hisse senetlerinin ya da hazine kağıtlarının bireyler veya kurumlar arasında alınıp satılması ise ikincil piyasa işlemi olarak sınıflandırılır (Günel, 2006: 18).

İkincil piyasalar, daha önce ihraç edilmiş menkul kıymetlerin el değiştirdiği piyasalardır ve bu piyasadaki işlemler sonucunda menkul kıymeti satan yatırımcı nakit elde ederken, menkul kıymeti ihraç eden şirket doğrudan herhangi bir fon girişi elde etmez. Şirketlerin sermaye toplaması yalnızca menkul kıymetlerin ilk kez yatırımcılara sunulduğu birincil piyasa işlemleri aracılığıyla mümkündür. Ancak ikincil piyasalar, menkul kıymetlerin hızlı ve etkin biçimde el değiştirmesini sağlayarak likiditeyi artırır. Bu likidite artışı, yatırımcılar açısından menkul kıymetleri daha cazip hale getirirken, ihraççı firmaların birincil piyasada yeni ihraçlarda bulunmasını da kolaylaştırmaktadır (Pilbeam, 2018: 142).

Bunun yanı sıra, ikincil piyasalar menkul kıymetlerin piyasa değerini belirleme işlevini de üstlenir. Bu piyasalarda gerçekleşen arz ve talep etkileşimi sayesinde, ihraç edilen menkul kıymetlerin güncel fiyatları belirlenir ve bu fiyatlar, ihraççılar için sermaye maliyetinin bir göstergesi haline gelir. Dolayısıyla, bir firmanın gelecekte gerçekleştireceği yeni menkul kıymet ihracında hangi fiyat seviyelerinde talep oluşabileceğine dair önemli bilgiler sunar. Bu bağlamda ikincil piyasalar, kaynak dağılımı açısından etkinliği artıran ve finansal piyasaların işleyişini optimize eden temel yapılardan biri olarak kabul edilmektedir (Howells ve Bain, 2008: 85).

Birincil piyasada menkul kıymet satın alan yatırımcılar, ihraççı şirkete ikincil piyasanın bu menkul kıymet için belirleyeceğini düşündükleri fiyattan daha fazlasını ödemeyeceklerdir. Menkul kıymetin ikincil piyasadaki fiyatı ne kadar yüksek olursa, ihraç eden firmanın birincil piyasada yeni bir menkul kıymet için alacağı fiyat ve dolayısıyla toplayabileceği finansal sermaye miktarı da o kadar yüksek olur. Bu nedenle ikincil piyasadaki koşullar, menkul kıymet ihraç eden şirketler için en önemli olanıdır. Bu nedenle, finansal piyasaları ele alan bu gibi kitaplar, birincil piyasalardan ziyade ikincil piyasaların davranışlarına odaklanmaktadır (Mishkin, 2018: 25).

1.2.4. İşlemin Gerçekleşme Zamanı Bakımından Finansal Piyasalar

Finansal piyasalar işlemin gerçekleşme zamanı bakımından ikiye ayrılır.

1.2.4.1. Spot piyasalar

Finansal varlıkların ticaretinin gerçekleştirildiği ve işlemlerin sözleşme anında ya da çok kısa bir süre içerisinde sonuçlandığı piyasalara "spot piyasalar" denir. Bu tür

piyasalarda, varlıkların teslimatı işlem tarihindeki piyasa fiyatı üzerinden fiziksel ya da nominal olarak sağlanmaktadır. Spot piyasaların en önemli özelliği, işlemlerin yüksek bir hızla tamamlanmasıdır; ödeme süreçleri de genellikle hemen veya çok kısa bir vadede gerçekleştirilmektedir. Bu piyasada işlem gören varlıklar arasında hisse senetleri, tahviller, döviz birimleri ve çeşitli emtialar gibi finansal araçlar yer almaktadır (Gitman ve Zutter, 2012: 303).

1.2.4.2. Vadeli piyasalar

Vadeli piyasalar, belirli bir varlığın bugünden kararlaştırılan bir fiyat üzerinden, gelecekte belirlenen bir tarihte teslim edilmek üzere alım ve satım işlemlerinin gerçekleştirildiği piyasalardır. Bu tür piyasalar, karşılıklı olarak taraflar arasında bağlayıcı nitelikte bir taahhüt anlaşmasının (sözleşme) yapılmasını gerektirir (Gitman ve Zutter, 2012: 307).

1.2.5. Finansal Varlıkların Türleri Bakımından Finansal Piyasalar

Finansal piyasalar, varlıkların türleri bakımından ikiye ayrılır.

1.2.5.1. Borç piyasaları

Borç piyasaları, devletlerin, şirketlerin ve diğer kurumların finansman ihtiyaçlarını karşılamak amacıyla bono, tahvil gibi borçlanma senetleri ihraç ettiği piyasalardır. Bu piyasalarda, belirli bir vade ve faiz oranıyla geri ödenmek üzere alınan fonlar alıcılar ve satıcılar arasında el değiştirir. Borç piyasaları, kısa vadeli borçlanma araçlarını kapsayan para piyasası ile uzun vadeli borçlanma araçlarının işlendiği sermaye piyasasını içerir (Mishkin, 2018: 26.)

1.2.5.2. Ortaklık piyasaları

Ortaklık piyasaları, işletmelerin ve şirketlerin mülkiyet payları olan hisse senetlerini ihraç ederek sermaye topladığı piyasalardır. Bu piyasalarda yatırımcılar, bir şirkete ait ortaklık paylarını satın alarak temettü gelirinden faydalanma ve sermaye kazancı elde etme imkanına sahip olur. Ortaklık piyasaları genelde halka açık şirketlerin hisse senetlerinin alınıp satıldığı borsalar üzerinden işleyiş gösterir.

1.3. Yatırım ve Hisse Senedi Kavramı

Yatırım, bireylerin, işletmelerin ve devletlerin, mevcut kaynaklarını gelecekte daha yüksek getiri elde etmek amacıyla yönlendirdiği stratejik bir süreçtir. Ekonomik büyümenin temel dinamiklerinden biri olan yatırım, finansal piyasalardan reel sektör projelerine kadar geniş bir yelpazede kendini göstermektedir. Geleneksel yatırım anlayışı, sermaye birikimini arttırarak gelecekteki belirsizlikleri minimize etmeye odaklanırken, günümüz dünyasında teknolojinin ve küreselleşmenin etkisiyle yatırım kararları daha karmaşık bir hale gelmiştir. Artık yatırımcılar yalnızca finansal kazanç sağlamakla kalmayıp, sürdürülebilirlik, sosyal sorumluluk ve yenilikçilik gibi unsurları da dikkate almak zorundadır (Bal ve Yalçınkaya, 2020).

Yatırım kavramının farklı açılardan ele alınarak tanımlanması mümkündür.

Graham (2005)'e göre yatırım, mevcut sermayenin korunmasını ve makul bir getiri sağlanmasını hedefleyen analitik bir süreçtir. Graham, yatırımın, spekülasyondan farklı olarak, sistematik analiz ve bilgiye dayalı karar alma sürecini içerdiğini vurgular. Yatırımcılar, risk ve getiri arasında bir denge kurarak uzun vadeli değer yaratmayı amaçlarlar. Graham'a göre, başarılı bir yatırımın üç temel unsuru vardır:

- Güvenlik Marjı: Yatırımın, beklenen getirinin risk seviyesini karşılayacak kadar yüksek olması.
- Temel Analiz: Şirketin mali durumu, yönetim kalitesi ve sektör dinamikleri gibi faktörlerin detaylı incelenmesi.
- Disiplinli Yaklaşım: Piyasa dalgalanmalarından etkilenmeden uzun vadeli hedeflere sadık kalınması.

Diğer yandan, Luenberger (2013)'e göre yatırım, matematiksel ve finansal modeller aracılığıyla analiz edilen bir karar verme süreci olarak ele alınır. Luenberger, yatırım kararlarının aşağıdaki temel bileşenlerden oluştuğunu ifade eder:

- Risk Yönetimi: Yatırımın belirsizlik altında nasıl yönetileceği ve farklı risk seviyelerine göre portföyün nasıl şekillendirileceği.

- Portföy Optimizasyonu: Farklı varlıkların bir araya getirilmesiyle optimum getiri ve risk dengesi sağlanması.

- Finansal Araçların Değerlendirilmesi: Hisse senetleri, tahviller ve türev ürünlerin finansal modellerle analiz edilmesi ve fiyatlandırılması.

Son yıllardaki çalışmalar incelendiğinde, hisse senedinin yatırım aracı olarak ilk sıralarda geldiği görülmektedir. Hisse senedi yatırımları, yatırımcıların portföylerini çeşitlendirmeleri ve uzun vadeli değer artışı elde etmeleri için kritik bir rol oynar. Hisse senedi, anonim şirketlerin, sermayesi paylara ayrılmış şirketlerin, komandit şirketlerin ve özel kanunlarla kurulmuş kamu kuruluşlarının; şekil şartları kanuna uygun bir şekilde düzenlenmiş ve alıcısına ortaklık hakkı tanıyan menkul kıymetler olarak tanımlanmaktadır (Öndeş, 2013: 118).

Graham (2005)'e göre hisse senetleri, uzun vadeli yatırımın temel taşlarından biri olup, şirketlerin büyümesinden ve kârlılığından pay alma fırsatı sunar. Luenberger (2013) ise hisse senedi yatırımlarını, risk-getiri optimizasyonu bağlamında değerlendirerek, portföy teorisi çerçevesinde çeşitlendirme stratejilerinin bir parçası olarak ele alır.

Modern yatırım stratejileri, ekonomik verilerin analizi, risk yönetimi ve piyasa dinamiklerinin anlaşılmasını gerektiren çok disiplinli bir yaklaşıma dayanmaktadır. Makroekonomik göstergeler, sektör trendleri ve teknolojik gelişmeler gibi faktörler yatırım kararlarını doğrudan etkilemektedir. Bunun yanı sıra, yatırım araçlarının çeşitliliği de geçmişe kıyasla artmış; hisse senetleri, tahviller, gayrimenkul, kripto varlıklar ve alternatif yatırımlar gibi seçenekler yatırımcıların portföylerini genişletmelerine olanak tanımıştır.

Geleneksel yatırım modelleri yerini giderek daha fazla veri odaklı ve yapay zekâ destekli stratejilere bırakmaktadır. Büyük veri analitiği ve makine öğrenmesi gibi teknolojiler, yatırımcıların geçmiş veriler ışığında geleceğe yönelik daha isabetli öngörülerde bulunmalarına imkan tanımaktadır. Aynı zamanda, küresel piyasalardaki belirsizlikler, yatırımcıları dinamik ve esnek stratejiler geliştirmeye zorlamaktadır.

Sonuç olarak, yatırım kavramı, finansal kazanç sağlamanın ötesinde, bilinçli kararlar ve etkin risk yönetimi gerektiren bir süreçtir. Yatırımcıların, piyasa koşullarına uyum sağlayabilmesi ve uzun vadeli sürdürülebilir kazançlar elde edebilmesi için mevcut

eğilimleri doğru analiz etmeleri ve bilimsel yöntemlerle hareket etmeleri kritik bir önem taşımaktadır.

1.4. Portföy Kavramı ve Portföy Yönetimi

Portföy kavramı ve portföy yönetimi, finansal piyasalarda yatırım kararlarının etkin bir şekilde alınmasını sağlayan temel unsurlardan biridir. Portföy, yatırımcıların belirli bir zaman diliminde en iyi getiriye elde ederken risklerini minimize etmeyi hedeflediği varlık bileşimidir (Markowitz, 1952). Portföy yönetimi ise, yatırımcıların hedef ve beklentileri doğrultusunda yatırım araçlarının seçilmesi, yönetilmesi ve zaman içinde yeniden düzenlenmesi sürecini kapsar (Elton ve Gruber, 1995). İyi yapılandırılmış bir portföy, yalnızca bir yatırımcının hedeflerinin ve risk toleransının bir yansıması olarak hizmet etmekle kalmaz, aynı zamanda potansiyel getirileri piyasadaki dalgalanmalarla ilişkili doğal risklere karşı dengeleyerek finansal sonuçları optimize etmek için tasarlanmış stratejik bir araç görevi görür.

Yatırımcılar risk karşısında farklı tutumlar sergileyebilir; bazıları yüksek risk almayı tercih ederken, bazıları riskten kaçınır veya kayıtsız kalır. Bu farklı tutumlar, yatırımcıların portföy seçimlerini etkiler. Risk almayı sevenler hisse senedi ve türev ürünlere yönelirken, riskten kaçınanlar daha güvenli yatırım araçları olan tahvilleri tercih eder. Yatırımcıların hedeflerine ulaşabilmesi için portföyde yer alacak menkul kıymetlerin seçimi ve yatırım miktarlarının belirlenmesi önemlidir. Ayrıca, portföyün getiri ve risk analizinin yapılarak beklentileri karşılayıp karşılamadığı değerlendirilmelidir. Bu nedenle, portföy yönetiminin bilimsel yöntemlerle ve profesyonel bir şekilde yürütülmesi gerekmektedir.

Bu bölümde, portföy kavramı, portföy yönetimi süreçleri, teoriler ve yatırım stratejileri detaylı bir şekilde ele alınacaktır.

1.4.1. Portföy Kavramı

Portföy, yatırımcıların çeşitli finansal araçları içeren bir varlık grubudur ve yatırımcıların risklerini dağıtarak daha güvenli ve dengeli getiri elde etmesini amaçlar. Portföy, tek bir yatırım aracına bağlı kalmaksızın farklı türdeki varlıkların bir araya getirilmesiyle oluşturulur. Portföyler genellikle hisse senetleri, tahviller, emtialar, gayrimenkuller, fonlar ve türev ürünlerden oluşabilir (Bodie vd., 2014). Bunlar arasındaki

dağılım, yatırımcının risk-getiri tercihine göre belirlenir ve zaman içinde piyasa koşullarına bağlı olarak yeniden düzenlenebilir.

Portföy oluşturmaının temelinde çeşitlendirme (diversification) prensibi bulunur. Çeşitlendirme, yatırımcıların belirli bir varlık sınıfındaki olası dalgalanmalara karşı daha dirençli hale gelmesini sağlar. Yalnızca hisse senetlerinden oluşan bir portföy, piyasa dalgalanmalarına karşı daha hassas olabilirken, tahvil ve gayrimenkul gibi daha istikrarlı varlıkların eklenmesi portföyün volatilitasını azaltabilir (Elton ve Gruber, 1995). Ayrıca, türev ürünler gibi risk yönetimi araçları, yatırımcının pozisyonunu koruyarak olası kayıpları minimize etmesine yardımcı olabilir.

1.4.1.1. Portföy türleri

Portföy türleri, yatırımcıların risk algısına, hedeflerine, yatırım stratejilerine ve piyasa koşullarına göre çeşitlenmektedir. Yatırımcıların portföy oluşturma sürecinde benimsedikleri yöntemler ve piyasadaki değişkenler, portföy türlerini belirlemede önemli rol oynar. Portföy türleri arasındaki farklılıklar, yatırımcıların risk toleransı ve getiri beklentilerine bağlı olarak değişkenlik gösterir.

- **Büyümeye Yönelik Portföyler:** Uzun vadeli sermaye kazançları elde etmeyi amaçlayan bu portföyler, genellikle yüksek büyüme potansiyeline sahip hisse senetlerinden oluşur. Bu tür portföyler daha yüksek risk taşımasına rağmen uzun vadede yüksek getiriler sağlayabilir (Reilly ve Brown, 2012).

- **Gelire Dayalı Portföyler:** Düzenli temettü veya faiz geliri elde etmeyi hedefleyen yatırımcılar için uygun olan bu portföyler, ağırlıklı olarak tahviller, temettü veren hisse senetleri ve kira gelirli gayrimenkuller içerir (Fabozzi, 2003).

- **Dengeli Portföyler:** Hem sermaye kazancı hem de düzenli gelir elde etmeyi hedefleyen yatırımcılar için uygun olup, hisse senetleri ve tahvillerin dengeli bir bileşimini içerir. Bu portföy türü, yatırımcıların orta düzeyde risk alarak uzun vadede sürdürülebilir bir getiri sağlamasını amaçlamaktadır (Reilly ve Brown, 2012).

- **Riskten Kaçınan Portföyler:** Minimum riskle sabit getirili yatırım araçlarını içeren portföylerdir. Genellikle devlet tahvilleri, mevduat hesapları ve yatırım fonları gibi düşük riskli araçlardan oluşur (Sharpe vd., 1999).

1.4.2. Portföy Yönetimi

Portföy yönetimi, yatırımcıların finansal hedeflerine ulaşmasını sağlamak amacıyla varlık dağılımını planlama, yönetme ve optimize etme sürecidir (Elton ve Gruber, 1995). Günümüz finans dünyasında portföy yönetimi, sadece yatırım araçlarının seçimiyle sınırlı değildir; risk yönetimi, piyasa analizi, performans ölçümü ve davranışsal finans gibi birçok bileşeni içermektedir (Bodie vd., 2014).

Finansal piyasalardaki dalgalanmalar, yatırımcı psikolojisi ve piyasa dinamikleri portföy yönetimini oldukça karmaşık bir süreç haline getirmektedir. Geleneksel yöntemlerin yanı sıra yapay zekâ ve büyük veri analitiği gibi teknolojik gelişmeler, portföy yönetiminde önemli bir yer tutmaktadır (Fabozzi, 2003).

Portföy yönetimi süreci, genellikle dört temel aşamadan oluşur:

1. Kısıtlamaların Belirlenmesi: Yatırımcının finansal hedefleri, risk toleransı, yatırım süresi ve piyasa koşulları analiz edilir (Bodie vd., 2014). Örneğin, genç yatırımcılar genellikle yüksek riskli ancak uzun vadede daha yüksek getiri sağlayan hisse senetlerini tercih ederken, emekliliğe yaklaşan yatırımcılar daha düşük riskli tahvil ve mevduat hesaplarına yönelmektedir (Reilly ve Brown, 2012).

2. Varlık Dağılımı: Risk ve getiri dengesini optimize etmek amacıyla yatırım araçlarının çeşitlendirilmesi sağlanır (Fabozzi, 2003). Çeşitlendirme, belirli bir varlık sınıfındaki kayıpların, portföydeki diğer varlıklar tarafından dengelenmesini amaçlamaktadır (Markowitz, 1952). Portföyün çeşitlendirilmesi için tahvil, hisse senedi, gayrimenkul ve alternatif yatırımlar gibi farklı varlık sınıflarına yer verilmelidir.

3. Yatırım Araçlarının Seçimi: Seçilen varlık türleri içindeki spesifik yatırım araçları belirlenir (Reilly ve Brown, 2012). Bu aşamada teknik ve temel analiz yöntemleri kullanılarak, en uygun yatırım araçları belirlenir. Örneğin, hisse senedi seçiminde şirketin finansal tabloları, sektör analizi ve ekonomik göstergeler dikkate alınmalıdır (Sharpe vd., 1999).

4. Portföyün Takibi ve Revizyonu: Piyasa koşulları ve yatırımcının değişen hedefleri doğrultusunda portföy düzenli olarak gözden geçirilir (Fama, 1970). Faiz

oranlarının deęiřmesi, makroekonomik göstergelerdeki gelişmeler ve global finansal krizler gibi etkenler, portföyde yeniden dengeleme yapılmasını gerektirebilir.

1.4.3. Portföy Teorileri ve Yöntemleri

Finansal piyasalar, yatırımcıların farklı risk ve getiri kombinasyonları arasından en uygun stratejiyi belirleyerek portföylerini yönetmeleri için çeşitli teorik çerçeveler geliřtirmiřtir. Portföy teorileri, yatırımcıların karar alma süreçlerini modelleyerek piyasa dinamiklerinin anlaşılmasına yardımcı olur. Bu teoriler, Modern Portföy Teorisi gibi rasyonalite varsayımına dayalı yaklaşımlardan, Davranışsal Portföy Teorisi gibi psikolojik unsurları içeren modellere kadar geniş bir yelpazede ele alınmaktadır.

Bu bölümde, portföy teorilerinin gelişimi ve modern yatırım yönetimi üzerindeki etkileri incelenecektir.

1.4.3.1. Modern Portföy Teorisi

Modern Portföy Teorisi (MPT), Nobel ödüllü ekonomist Harry Markowitz (1952) tarafından geliştirilmiş olup, yatırımcıların risk çeşitlendirmesi (diversification) yoluyla optimal portföy oluşturabileceklerini öne sürer. Geleneksel yatırım stratejilerinden farklı olarak, riskin sadece bireysel varlıkların oynaklığına baęlı olmadığı, aynı zamanda varlıklar arasındaki ilişkilere de dayandığı fikrine dayanmaktadır (Markowitz, 1952).

Teorinin temel prensibi, belirli bir risk düzeyinde en yüksek getiriyi saęlayan veya belirli bir getiri seviyesinde en düşük riski taşıyan portföyün oluşturulmasıdır. Bunu saęlamak için Markowitz, varyans-kovaryans yapısını kullanarak portföy riskini minimize eden optimum portföylerin oluşturulmasını önermiştir. Burada, portföy getirisinin ağırlıklı ortalama olarak hesaplanmasına rağmen, toplam portföy riskinin bireysel varlıkların risklerinin basit bir toplamı olmadığı ortaya konmuřtur. Bunun yerine, portföy riskinin, varlıklar arasındaki korelasyonlara baęlı olarak çeşitlendirme (diversification) yoluyla azaltılabileceęi gösterilmiştir (Elton ve Gruber, 1997).

MPT, yatırım dünyasında devrim niteliğinde bir deęişim yaratmış ve özellikle etkin sınır (efficient frontier) kavramını ortaya koyarak yatırımcıların rasyonel bir şekilde karar vermelerini saęlamıştır. Etkin sınır, yatırımcıların risk-getiri profiline göre en iyi portföyleri seçmelerini saęlayan grafiksel bir gösterimdir. Yatırımcılar, risk iřtahlarına göre etkin sınır

üzerinde bulunan portföylerden birini tercih ederek, sermayelerini daha verimli bir şekilde yönetebilirler (Sharpe, 1964).

Bu teori, geleneksel yatırım yaklaşımlarından farklı olarak, yatırımcıların sadece bireysel varlıkların geçmiş performanslarını değil, aynı zamanda varlıklar arasındaki ilişkiyi ve çeşitlendirme olanaklarını göz önünde bulundurmaları gerektiğini göstermiştir (Markowitz, 1991).

1.4.3.2. Sermaye Varlıklarını Fiyatlama Modeli (CAPM)

Sermaye Varlıklarını Fiyatlama Modeli (CAPM), Markowitz'in portföy teorisini genişleterek, varlık fiyatlarının belirlenmesinde sistematik riskin (market risk) rolünü açıklayan bir modeldir. Sharpe (1964), Lintner (1965) ve Mossin (1966) tarafından bağımsız olarak geliştirilen bu model, bir varlığın beklenen getirisinin, risksiz faiz oranı, piyasa risk primi ve varlığın sistematik risk duyarlılığı (beta katsayısı) ile belirlendiğini öne sürer. Model şu şekilde formüle edilir:

$$E(R_i) = R_f + \beta_i(E(R_m) - R_f)$$

1.1

Burada; $E(R_i)$ i varlığının beklenen getirisi, R_f risksiz faiz oranı, $E(R_m)$ piyasanın beklenen getirisi ve β_i varlığın sistematik risk seviyesini ölçen katsayı olarak tanımlanır.

CAPM, yalnızca sistematik riskin fiyatlandırıldığını ve yatırımcıların sistematik olmayan risk için ek bir getiri beklememesi gerektiğini öne sürmektedir (Sharpe, 1964).

1.4.3.3. Arbitraj Fiyatlama Modeli (APT)

Stephen Ross (1976) tarafından geliştirilen Arbitraj Fiyatlama Modeli (APT), CAPM'in tek faktörlü yapısını eleştirerek, varlık fiyatlarının yalnızca piyasa riskiyle değil, birden fazla risk faktörü ile belirlendiğini öne sürmüştür. CAPM'in beta katsayısı üzerinden tek bir piyasa risk primi ile hisse senetlerinin fiyatlarını açıklamaya çalışması, piyasadaki anomalilerin tam olarak modellenememesine neden olmuş ve bu eksikliği gidermek amacıyla APT geliştirilmiştir.

APT'nin temel varsayımı, fiyatlandırma anomalilerinin ve arbitraj fırsatlarının piyasada uzun vadede ortadan kalkacağıdır. Model, yatırımcıların herhangi bir arbitraj fırsatını değerlendirerek risksiz kazanç sağlayabileceğini ve bu durumun fiyatları dengeleyeceğini savunur (Ross, 1976). APT'ye göre, varlık getirileri birden fazla makroekonomik faktörden etkilenir ve bu faktörler yatırımcıların risk algısını şekillendirir. Model, CAPM'den farklı olarak piyasa portföyüne bağımlı değildir ve varlık getirilerini açıklayan faktörler ampirik olarak belirlenebilir. Bu esneklik, APT'nin teorik çerçevesini finansal piyasalarda uygulamaya daha uygun hale getirmiştir (Fama ve French, 2004).

APT, finansal varlıkların beklenen getirilerinin, birden fazla risk faktörüne bağlı olduğu prensibine dayanır. Modelin matematiksel ifadesi şu şekildedir:

1.2

$$E(R_i) = R_f + \lambda_1 f_1 + \lambda_2 f_2 + \dots \lambda_n f_n$$

Burada, $E(R_i)$ i varlığının beklenen getirisi, R_f risksiz faiz oranı, $f_1, f_2, \dots, f_n$ varlıkların fiyatlarını etkileyen makroekonomik faktörler ve $\lambda_1, \lambda_2, \dots, \lambda_n$ bu faktörlerin getiri üzerindeki etkisini ölçen katsayılarıdır.

APT, hangi faktörlerin modele dahil edilmesi gerektiği konusunda herhangi bir kısıtlama getirmez. Bu nedenle, yatırımcılar ve araştırmacılar, tarihsel verilere dayanarak en önemli risk faktörlerini belirleyebilirler. Genellikle aşağıdaki faktörler, finansal varlıkların getirilerini açıklamak için kullanılmıştır (Chen, Roll ve Ross, 1986):

- Enflasyon oranı: Enflasyon, nominal faiz oranlarını ve reel getirileri doğrudan etkileyen önemli bir faktördür.
- Faiz oranları: Merkez bankalarının para politikaları ve faiz oranlarındaki değişiklikler, varlık fiyatlarını etkileyebilir.
- Sanayi üretimi: Ekonomik büyüme ve üretim seviyeleri, şirket karlılığı ve hisse senedi fiyatları üzerinde doğrudan bir etkiye sahiptir.
- Petrol fiyatları: Enerji maliyetlerindeki değişimler, birçok sektörü etkileyerek hisse senetlerinin değerini şekillendirir.

- Piyasa risk primi: Genel piyasa koşullarına bağlı olarak yatırımcıların riskten kaçınma düzeyi belirlenir.

APT'nin en önemli avantajlarından biri, yatırımcıların portföy risklerini birden fazla faktör üzerinden analiz etmelerine olanak tanımasıdır. Model, CAPM'in piyasa portföyüne bağımlılığını ortadan kaldırarak daha esnek bir risk analizi imkanı sunar. Ayrıca, CAPM'de piyasa portföyünün belirlenmesi ve uygulanması zorlu bir süreç iken, APT'de faktörlerin seçimi ampirik olarak yapılabilir ve bu durum modelin daha geniş bir finansal analize uyarlanmasını sağlar (Chen, Roll ve Ross, 1986).



2. BÖLÜM

2. YAPAY ZEKÂ VE TAHMİNLEME

Tahminleme (forecasting), geçmiş ve mevcut verilerden yararlanarak gelecekteki olayların, eğilimlerin veya sonuçların öngörülmesini amaçlayan analitik bir süreçtir. Bu süreç, karar alma mekanizmalarının temel taşlarından biri olarak hem akademik araştırmalarda hem de uygulamalı disiplinlerde merkezi bir rol üstlenmektedir. Tahminleme teknikleri, stratejik planlama, kaynak tahsisi, risk yönetimi ve performans değerlendirmesi gibi kritik alanlarda karar vericilere yön gösterir (Makridakis vd., 2018).

Tahminleme, genel olarak istatistiksel yöntemler ve yapay zekâ teknikleri olarak iki ana gruba ayrılır. Geleneksel istatistiksel yöntemler olarak hareketli ortalamalar, doğrusal regresyon ve ARIMA modelleri zaman serisi analizinde uzun yıllardır başarıyla kullanılmaktadır. Ancak bu yöntemlerin performansı, özellikle doğrusal olmayan yapılar ve değişkenliğin yüksek olduğu durumlarda sınırlı kalabilmektedir (Hyndman ve Athanasopoulos, 2021).

Bu sınırlamaları aşmak adına son yıllarda makine öğrenmesi ve derin öğrenme gibi yapay zekâ temelli yaklaşımlar tahminleme alanına entegre edilmiştir. Bu yaklaşımlar, doğrusal olmayan ilişkileri modelleme yetenekleri, öğrenme süreçlerindeki esneklikleri ve büyük veri setleri ile başa çıkabilme kapasiteleri sayesinde geleneksel modellere kıyasla daha yüksek tahmin doğruluğu sunabilmektedir (Ahmed vd., 2010).

Yapay zekâ (YZ), büyük veri çağında karar verme süreçlerini destekleyen en kritik teknolojilerden biri hâline gelmiştir. Özellikle tahminleme (forecasting) alanında sağladığı katkılar, hem akademik araştırmalarda hem de uygulamalı çalışmalarda dikkat çekmektedir. Yapay zekâ algoritmaları, büyük hacimli ve çok boyutlu veri setlerinden öğrenme yeteneği sayesinde, klasik istatistiksel yaklaşımların ötesine geçerek doğrusal olmayan ve karmaşık ilişkileri de modelleyebilmekte ve bu sayede daha isabetli tahminler yapılmasına olanak sağlamaktadır (Goodfellow vd., 2016).

Bu bölümde öncelikle yapay zekânın tanımı ele alınıp, uygulama alanlarına dair bilgiler verilmiştir. Ayrıca yapay zekânın bir alt dalı olan makine öğrenmesi ile ilgili bilgiler verilip türleri incelenmiştir. Ardından literatürde bilinen yapay zekâ yöntemleri

açıklanmıştır. Son olarak da yapay zekâ yöntemleri ile ilgili literatürde yapılmış çalışmalara yer verilmiştir.

2.1. Yapay zekâ Tanımı

Yapay zekâ (YZ), insan benzeri zekâ davranışlarını taklit edebilen sistemlerin geliştirilmesine yönelik bilimsel ve teknolojik bir disiplindir. En yalın tanımıyla yapay zekâ, insanlar tarafından sergilenen öğrenme, akıl yürütme, problem çözme, algı ve dil anlama gibi bilişsel yeteneklerin makineler tarafından gerçekleştirilmesini amaçlayan bilgisayar sistemlerinin oluşturulmasıdır (Russell ve Norvig, 2020). Alan, bilgisayar bilimi, matematik, istatistik, bilişsel bilim ve nörobilim gibi birçok disiplinin kesişiminde konumlanmakta ve bu disiplinlerden aldığı kavram ve yöntemlerle gelişimini sürdürmektedir (Nilsson, 2010).

Yapay zekâ sistemlerinin temelini oluşturan algoritmalar genellikle tahmin (prediction) ve optimizasyon (optimization) işlevlerini üstlenir. Tahmin algoritmaları, geçmiş verilerden öğrenerek gelecekteki olayları öngörebilmekte; optimizasyon algoritmaları ise çeşitli senaryolar altında en uygun çözümü bulma amacıyla çok sayıda alternatifi değerlendirip en rasyonel kararı önerme kapasitesine sahiptir (Domingos, 2015). Bu bağlamda yapay zekâ, yalnızca bilişsel görevleri yerine getirmekle kalmaz; aynı zamanda veriye dayalı karar destek sistemlerinin geliştirilmesini de mümkün kılar.

2.2. Yapay zekâ Uygulama Alanları

Yapay zekâ yöntemlerinin yaygınlaşması, toplumsal ihtiyaçlara yanıt vermek ve verimliliği artırmak amacıyla geniş çapta gerçekleştirilmektedir. Eğitimden sağlığa, mühendislikten sanata kadar birçok farklı alanda, inovasyon sağlama potansiyeli taşıyan bu teknolojiler, hem bireyler hem de organizasyonlar için büyük fırsatlar sunmaktadır (Coşkun ve Gülleroğlu, 2021). Yapay zekâ uygulama alanlarının genişlemesi, hem teknolojik gelişmelere hem de toplumsal ihtiyaçlara yanıt verme kapasitesine dayanır (Kocabaş, 2015). Bu nedenle, yapay zekânın birçok sektördeki gelişimini ve uygulamalarını incelemek, günümüzün ve geleceğin dinamiklerini anlamak açısından kritik öneme sahiptir. Yapay zekâ, insan benzeri zeka gerektiren görevleri yerine getirme kapasitesine sahip sistemleri tanımlamak için kullanılan bir terimdir. Yapay zekânın uygulamaları günümüzde çeşitli alanlarda kapsamlı bir şekilde yer almaktadır ve bu uygulamalar birçok farklı yöntemle

dayanmaktadır. Bu yöntemler arasında makine öğrenimi, derin öğrenme ve doğal dil işleme (NLP) öne çıkmaktadır.

2.3. Makine Öğrenmesi

Makine öğrenmesi, yapay zekânın önemli bir alt disiplini olarak, bilgisayarların deneyim yoluyla öğrenmesini ve yeni durumlara uyum sağlamasını mümkün kılan algoritmalarla oluşmaktadır. Bilgisayarlara açık şekilde programlanmaksızın örüntüleri tanıma, çıkarım yapma ve karar alma gibi görevleri gerçekleştirme kabiliyeti kazandıran bu alan, veri odaklı modelleme yöntemleriyle karakterize edilmektedir (Mitchell, 1997; Tosunoğlu vd., 2021).

Makine öğrenmesinin temel bileşenleri arasında modelleme, veri setleri, eğitim süreci ve algoritmalar yer almaktadır (Clarke vd., 2020). Modelleme süreci, gerçek dünya olgularının matematiksel yapılarla temsil edilmesini ifade ederken; veri setleri, bu yapıların öğrenilmesinde kullanılan örneklem gruplarını oluşturmaktadır. Eğitim süreci, algoritmanın örnekler aracılığıyla genel geçer ilişkileri öğrenmesini sağlar. Kullanılan algoritmalar ise öğrenme sürecinin yapısını, hızını ve doğruluğunu belirlemektedir (Goodfellow vd., 2016; Özer, 2020).

Makine öğrenmesinin evrimi, teknolojik altyapıdaki gelişmelerle doğrudan ilişkilidir. 1980’li ve 1990’lı yıllarda kişisel bilgisayarların yaygınlaşması, işlemci hızlarının artması ve veri saklama kapasitelerinin gelişmesi, Makine öğrenmesi algoritmalarının daha büyük veri setleri SVM ve YSA gibi temel algoritmalar geliştirilmiş ve uygulamalara entegre edilmiştir (Müller ve Guido, 2016).

2000’li yıllarla birlikte “büyük veri” (big data) kavramının yükselişi ve paralel işlem mimarilerinin gelişimi, ML’in ölçeklenebilirliğini artırmıştır. Özellikle 2010 sonrasında derin öğrenme (deep learning) yaklaşımlarının gelişmesiyle birlikte, makine öğrenmesi; NLP, görüntü işleme, zaman serisi tahmini ve biyomedikal sinyal işleme gibi alanlarda önemli başarılar elde etmiştir (LeCun vd., 2015; Acı ve Ayyıldız Doğanşoy, 2022).

Makine öğrenmesi, günümüzde hemen her sektörde yaygın olarak kullanılmaktadır. Sağlık alanında görüntü analizi, tanı sistemleri ve hastalık öngörüsü; finans alanında kredi skorlama, algoritmik ticaret ve dolandırıcılık tespiti; sanayide kestirimci bakım, kalite

kontrol ve süreç optimizasyonu; eğitim alanında ise öğrenci başarı tahmini ve kişiselleştirilmiş öğrenme sistemleri Makine öğrenmesi tabanlı çözümlerle sağlanmaktadır (Zawacki-Richter vd., 2019).

Makine öğrenmesi üç türe ayrılmaktadır. Bunlar; denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme (Ayazoğlu vd, 2020).

2.3.1. Denetimli Öğrenme

Denetimli (gözetimli) öğrenme, makine öğrenmesi alanında en yaygın kullanılan paradigmalardan biridir ve algoritmaların, önceden etiketlenmiş veri setleri üzerinden öğrenmesini temel alır. Bu öğrenme yaklaşımında, modelin eğitilmesi için her bir girdiye karşılık gelen çıktılar (etiketler) sistematik olarak modele sunulur. Model, bu girdi-çıkı eşleşmeleri üzerinden genelleştirme yapmayı öğrenerek, daha önce hiç karşılaşmadığı örnekler üzerinde tahminleme yapabilme becerisi kazanır (Goodfello vd., 2016).

Denetimli öğrenme algoritmaları iki temel alt kategoriye ayrılmaktadır: sınıflandırma (classification) ve regresyon (regression). Sınıflandırma problemlerinde, modelin çıktısı belirli bir kategoriye ait sonlu ayrık değerlerden oluşurken; regresyon problemlerinde modelin tahmin ettiği değerler sürekli bir nicel ölçekte yer alır (Alpaydın, 2020). Örneğin, bir sınıflandırma algoritması bir e-postanın spam olup olmadığını belirlerken, regresyon algoritması belirli bir ürünün gelecekteki satış miktarını öngörebilir.

Bu yöntemlerin başarılı bir şekilde uygulanabilmesi için eğitim veri setinin özenle hazırlanması gerekir. Veri seti, modelin öğrenebilmesi için yeterli çeşitlilikte ve dengeli örnekler içermelidir. Ayrıca, verilerden gürültülerin ayıklanması, eksik verilerin giderilmesi ve özellik mühendisliği (feature engineering) gibi ön işleme (preprocessing) adımları da model performansını doğrudan etkileyen kritik süreçlerdendir (Kelleher vd., 2015).

Özellikle regresyon temelli problemler, tahmine dayalı karar destek sistemlerinde önemli rol oynamaktadır. Örneğin, taşınmaz değerlendirme, iklim verisi analizi, müşteri yaşam boyu değeri tahmini gibi çeşitli senaryolarda doğrusal ve doğrusal olmayan regresyon algoritmaları sıkça kullanılmaktadır. LSTM gibi zaman serisi odaklı algoritmalar ise, geçmişe ait bağlam bilgisini koruma kabiliyeti sayesinde finansal zaman serisi tahminlerinde oldukça başarılı sonuçlar verebilmektedir (Zhang vd., 2020).

Bu algoritmalar; metin sınıflandırma, duygu analizi gibi NLP, hastalık teşhisi gibi sağlık, hisse senedi fiyatlaması, görüntü işleme (ör. nesne tanıma) gibi pek çok alanda başarılı uygulamalarla kullanılmaktadır (Kubat, 2017).

2.3.2. Denetimsiz Öğrenme

Denetimsiz öğrenme (gözetimsiz), yalnızca girdi verilerine dayanarak çalışan ve bu veriler içerisinde herhangi bir etiketli çıktı bilgisi olmadan yapısal örüntüleri keşfetmeyi amaçlayan makine öğrenmesi yaklaşımıdır. Bu yöntemde model, verideki içsel yapıyı, örüntüleri, benzerlikleri, aykırılıkları veya dağılımları keşfederek veriyi organize etmeye çalışır (Hastie vd., 2009). Denetimsiz öğrenme algoritmalarının temel hedefi, veri içerisindeki doğal gruplamaları ya da ilişkileri belirleyerek, insan müdahalesi olmaksızın bilginin özetlenmesini ve anlamlandırılmasını sağlamaktır (Kotsiantis, 2007).

Bu öğrenme paradigması, özellikle büyük ve karmaşık veri setlerinin analizinde, verilerin önceden etiketlenmesinin mümkün olmadığı ya da etiketi üretmenin yüksek maliyetli olduğu durumlarda oldukça etkilidir. Denetimsiz öğrenme algoritmaları; kümeleme (clustering), boyutsal küçültme (dimensionality reduction), anomali tespiti (anomaly detection) ve birliktelik kuralları (association rule learning) gibi temel tekniklerle temsil edilir (Aggarwal ve Reddy, 2013). Bu algoritmalar arasında K-Ortalamlar (K-Means), Hiyerarşik Kümeleme, Apriori, PCA (Principal Component Analysis), t-SNE (t-Distributed Stochastic Neighbor Embedding) ve Otomatik Kodlayıcılar (Autoencoders) gibi yöntemler yaygın biçimde kullanılmaktadır.

Kümeleme teknikleri, veriler arasında benzerlik düzeylerine dayalı olarak gruplama gerçekleştirir. Örneğin, K-Ortalamlar algoritması, verileri belirli sayıda kümeye ayırarak her bir veriyi en yakın merkez etrafında kümeler (Jain, 2010). Bu sayede müşteri segmentasyonu, pazarlama stratejilerinin özelleştirilmesi, sosyal medya kullanıcılarının davranışsal gruplandırılması gibi alanlarda önemli faydalar sağlar (Xu ve Wunsch, 2005).

Boyutsal küçültme algoritmaları ise yüksek boyutlu veri setlerinde yer alan değişkenler arasındaki korelasyonu dikkate alarak, verilerin daha az boyutla temsil edilmesini sağlar. Bu yöntemler, özellikle görselleştirme, model karmaşıklığının azaltılması ve hesaplama maliyetinin düşürülmesi gibi amaçlarla kullanılmaktadır. PCA ve t-SNE gibi

teknikler, genetik verilerin analizi, biyoinformatik uygulamaları veya görüntü işleme gibi alanlarda yaygın olarak kullanılmaktadır (Van der Maaten ve Hinton, 2008).

Anomali tespiti, denetimsiz öğrenmenin kritik uygulama alanlarından biridir. Bu yöntem, veri setinde yer alan sıradışı örnekleri tespit ederek, örneğin siber güvenlikte ağ trafiği anomalileri, kredi kartı dolandırıcılığı ya da medikal alanda olağan dışı semptomların belirlenmesi gibi çeşitli kritik sistemlerde erken uyarı mekanizması olarak kullanılabilir (Chandola vd., 2009).

İrliktelik kuralları ise özellikle perakende sektöründe, ürünler arasında satın alma ilişkilerini ortaya koyarak çapraz satış (cross-selling) stratejileri geliştirilmesinde kullanılmaktadır. Market sepeti analizi bu yaklaşımın en bilinen uygulamalarından biridir (Agrawal ve Srikant, 1994).

Ulusal literatürde de denetimsiz öğrenmeye ilişkin çalışmalarda bu algoritmaların veri madenciliği, eğitim analitiği, sağlık bilişimi ve finansal modelleme gibi alanlarda etkili biçimde kullanıldığı görülmektedir (Toprak, 2023: 28-29; Aydın, 2022).

2.3.3. Pekiştirmeli Öğrenme

Pekiştirmeli öğrenme (Reinforcement Learning - RL), makine öğrenmesinin önemli bir alt alanı olup, bir ajanın (öğrenen varlık) belirli bir ortam (environment) içinde ödül-ceza temelli geri bildirimler aracılığıyla kararlar almayı öğrenmesini esas alır. Ajan, çevresiyle etkileşim kurarak eylemler gerçekleştirir ve bu eylemler sonucunda bir ödül (reward) ya da ceza (penalty) alır; süreç boyunca amaç, uzun vadede en yüksek toplam ödülü elde edecek bir politika (policy) geliştirmektir (Sutton ve Barto, 2018). Diğer öğrenme türlerinden farklı olarak, pekiştirmeli öğrenmede doğru etiketli veriler önceden sağlanmaz; öğrenme süreci doğrudan deneme-yanılma (trial-and-error) mekanizmasıyla gerçekleşir (Kaelbling vd., 1996).

Derin pekiştirmeli öğrenme (Deep Reinforcement Learning), özellikle durum-eylem çiftlerine dayalı zamana bağlı karar alma problemlerinde öne çıkan bir yöntem olarak kabul edilmektedir. Bu yaklaşım, derin sinir ağlarıyla entegre edildiğinde, karmaşık ortamlarda özerk öğrenme yetisi kazandırmakta ve stratejik karar süreçlerinde yüksek başarı oranlarıyla dikkat çekmektedir. Son yıllarda yapılan araştırmalar, Deep RL algoritmalarının robotik,

oyun kuramı ve kontrol sistemleri gibi alanlarda “insan seviyesinin üzerinde” performanslar sergilediğini göstermektedir (Li, 2017; Arulkumaran vd., 2017; François-Lavet vd., 2018).

Endüstriyel uygulamalar açısından bakıldığında, robotik sistemlerde hareket planlama, çevresel engelleri aşma ve dinamik ortamlarda uyarlanabilir kontrol stratejileri geliştirme amacıyla RL algoritmaları sıklıkla kullanılmaktadır (Kober vd., 2013). Bunun yanı sıra, otonom araç teknolojilerinde RL, aracın trafik kurallarına uyumlu şekilde hareket etmesini ve çevresel koşullara göre strateji geliştirmesini sağlar (Aradi, 2020). Dronlar ve insansız hava araçlarında ise rotaların optimize edilmesi, batarya ömrünün verimli kullanılması ve çevresel değişikliklere hızlı adaptasyon gibi işlevlerde RL tabanlı algoritmalar etkin biçimde uygulanmaktadır (Zhang vd., 2021).

Finansal sistemlerde de RL'nin önemi artmaktadır. Özellikle algoritmik alım-satım sistemlerinde, piyasa verilerine bağlı olarak dinamik stratejiler geliştirilmesi, portföy optimizasyonu ve risk yönetimi gibi süreçlerde pekiştirmeli öğrenme modelleri karar destek sistemleri olarak kullanılabilmektedir (Moody ve Saffell, 2001; Deng vd., 2016).

Ayrıca sağlık alanında, ilaç dozajlarının kişiselleştirilmesi, radyoterapi planlaması veya klinik karar destek sistemlerinin geliştirilmesinde RL algoritmaları potansiyel faydalar sunmaktadır (Yu vd., 2019).

2.4. Yapay zekâ Algoritmaları

Yapay zekânın tahmin ve optimizasyon algoritmalarının sağladığı avantajlar oldukça fazladır. Bu avantajlardan biri, veri odaklı içgörüler aracılığıyla karar alma süreçlerinin geliştirilmesidir. Tarihsel verilerden öğrenen algoritmalar kullanarak, kuruluşlar sonuçları tahmin ederken veya süreçleri optimize ederken daha yüksek bir doğruluk oranı elde edebilir (Guan vd., 2018). Ayrıca, yapay zekânın otomasyon yetenekleri, gerçek zamanlı bilgilerin hızlı bir şekilde işlenmesini sağlayarak paydaşların değişen koşullara hızla yanıt vermesine ve daha iyi bilgilendirilmiş seçimler yapmasına olanak tanır.

2.4.1. Yapay sinir ağları

Yapay Sinir Ağları (YSA), insan beynindeki nöronların çalışma mantığını taklit ederek öğrenme gerçekleştiren hesaplamalı modellerdir. Bu sistemler, bilgi işleme

süreçlerini taklit etme yetenekleri sayesinde farklı uygulama alanlarına entegre olabilen etkili araçlar sunmaktadır (Haykin, 2009).

1940'lı yıllarda Warren McCulloch ve Walter Pitts tarafından geliştirilen ilk sinir ağı modeli, 1980'lerde geri yayılım algoritmasının (backpropagation) keşfiyle hız kazanmıştır. Bu algoritma, ağın hata oranını minimize ederek öğrenme sürecini optimize etme yeteneği sağlamıştır (Rumelhart vd., 1986). Geri yayılım süreci, bir ağın bileşenleri arasındaki ağırlıkların ayarlanmasını sağlayarak, ağın doğru çıktılara ulaşmasına yardımcı olur. Bu dönemde, YSA'ların çok katmanlı mimarileri, yani derin öğrenme yapıları, ortaya çıkmış ve daha karmaşık görevleri yerine getirme kapasiteleri artmıştır (LeCun vd., 2015).

YSA'nın temel yapısı, biyolojik sinir sistemlerinden ilham alınarak tasarlanmış ve bilgi işleme kapasitesine sahip yapay nöronların bir araya gelmesiyle meydana gelir. Genellikle bir yapay sinir hücresinin girdi, ağırlık, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktı olmak üzere 5 tane temel bileşeni vardır. Her bir yapay nöron, aldığı giriş sinyallerini belirli ağırlıklarla çarparak toplar ve bir aktivasyon fonksiyonu kullanarak çıktıyı oluşturur. Bu işlem, ağın öğrenme ve genelleme yeteneğini geliştirmesine olanak tanır (Akgöbek ve Öztemel, 2006: 48).

YSA'da bir nöronun çıkışı, girişlerin ağırlıklı toplamına bir bias terimi eklenmesi ve ardından bir aktivasyon fonksiyonunun uygulanmasıyla elde edilir (Haykin, 2009; Goodfellow vd., 2016). Bu işlem,

$$y = f\left(\sum_{i=1}^t (w_i \cdot x_i) + b\right) \quad 2.1$$

ile ifade edilir (Haykin, 2009). Burada,

- x_i : i . giriş değerini
- w_i : i . girişe karşılık gelen ağırlığını
- b : bias terimini
- f : aktivasyon fonksiyonunu

- y : nöronun çıktısını temsil eder (Goodfellow vd., 2016).

Girdi değerleri ile ağırlıkların çarpımından oluşan toplam, bias terimi ile birlikte aktivasyon fonksiyonuna girerek nöronun çıkışını (y) oluşturur (Bishop, 2006).

Yaygın olarak kullanılan aktivasyon fonksiyonları ise aşağıdaki şekilde ifade edilir.

Sigmoid fonksiyonu, doğrultulmuş doğrusal olmayan bir dönüşüm sağlar ve çıktıyı $(0, 1)$ aralığına sıkıştırır. Bu nedenle özellikle iki sınıflı sınıflandırma problemlerinde yaygın olarak kullanılır. Fonksiyon aşağıdaki şekilde ifade edilir (Hastie vd., 2009).

$$f(x) = \frac{1}{1 + e^{-x}} \quad 2.2$$

Bu fonksiyon küçük negatif girişlerde sıfıra, büyük pozitif girişlerde bire yaklaşır ve bu nedenle olasılık yorumuna uygundur (Hastie vd., 2009).

ReLU, negatif girdiler için sıfır, pozitif girdiler için ise doğrusal bir yanıt üreten aktivasyon fonksiyonudur. Hesaplama açısından verimlidir ve derin ağlarda gradyan sönmesi (vanishing gradient) problemini azaltabilir. Bu fonksiyon,

$$f(x) = \max(0, x) \quad 2.3$$

ile ifade edilir. ReLU, modern derin öğrenme mimarilerinde en çok tercih edilen aktivasyon fonksiyonlarından biridir (Glorot vd., 2011).

Tanh aktivasyon fonksiyonu, girişleri $(-1,1)$ aralığına dönüştürür. Sigmoid fonksiyonuna benzer şekilde S-şeklindedir ancak sıfır merkezlidir. Formül 2.4. eşitliği ile ifade edilir (LeCun vd., 1998).

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad 2.4$$

YSA, karmaşık veri kümelerinden öğrenme ve tahmin yapma yetenekleriyle pek çok sektörde geniş bir kullanım alanı bulmuştur (LeCun vd., 2015). Görüntü işleme alanında yüz

tanıma, tıbbi görüntü analizi ve nesne tespiti gibi uygulamalarda sıklıkla tercih edilirken (Krizhevsky vd., 2012), NLP teknolojilerinde makine çevirisi, duygu analizi ve metin kategorilendirme gibi görevlerde etkili bir şekilde kullanılmaktadır (Young vd., 2018). Finans ve ekonomide hisse senedi öngörüsü, sahtekarlık algılama ve risk değerlendirme gibi kritik süreçlerde önemli bir rol oynayan YSA (Fischer ve Krauss, 2018), sağlık sektöründe ise hastalık teşhisi, genetik analiz ve bireye özel tedavi planlarının hazırlanmasında güçlü bir araçtır (Miotto vd., 2017). Ayrıca, otonom sistemler ve robotik teknolojilerde sürücüsüz araçlar, endüstriyel otomasyon ve enerji yönetiminin optimize edilmesinde YSA'nın katkısı büyüktür (Bojarski vd., 2016). Bunun yanı sıra, pazarlama ve satış stratejilerinde kişiselleştirilmiş öneri sistemleri, müşteri davranışlarının analizi ve hedefli reklamcılık gibi uygulamalarla işletmelere önemli bir değer sunar (Zhang vd., 2017).

YSA'nın sağladığı avantajlar arasında karmaşık verilerden öğrenme yeteneği, farklı problem türlerine kolay uyum sağlayabilmesi, yüksek doğruluk oranları ve büyük veri setleri üzerinde etkili bir şekilde çalışabilmesi yer alır (LeCun vd., 2015). YSA'lar, doğrusal olmayan ilişkileri modelleyebilme kapasitesi sayesinde görüntü işleme, NLP ve zaman serisi analizi gibi yapısal olarak farklı veri türlerine kolayca uyum sağlayabilir (Goodfellow vd., 2016). Özellikle derin yapay sinir ağları, katmanlar aracılığıyla çok düzeyli özellik çıkarımı yaparak verilerdeki gizli desenleri tespit etme konusunda üstün performans sergiler (Schmidhuber, 2015). Bununla birlikte, büyük veri setleriyle birlikte çalıştıklarında, paralel hesaplama olanakları sayesinde öğrenme kapasiteleri daha da artar ve bu durum sınıflandırma, regresyon ve öngörü modellerinde daha yüksek doğruluk oranlarına ulaşmalarını sağlar (Najafabadi vd., 2015). Ayrıca, yeniden kullanılabilirlikleri ve farklı uygulama alanlarında transfer öğrenme yoluyla kullanılabilir olmaları da YSA'ları modern yapay zeka uygulamalarında vazgeçilmez hale getirmiştir (Pan ve Yang, 2010).

2.4.2. Uzun Kısa Dönemli Bellek

Uzun Kısa Dönemli Bellek (LSTM) ağları, sıralı ve zamana bağlı verilerin modellenmesinde öne çıkan, tekrarlayan sinir ağlarının (Recurrent Neural Networks - RNN) temel sınırlılıklarını aşmak amacıyla geliştirilmiş ileri düzey bir YSA mimarisidir. Geleneksel RNN'ler, teorik olarak geçmiş zamana ait bilgileri ağ içinde taşıyabilir olsa da, uzun vadeli bağımlılıkları öğrenmekte ciddi zorluklar yaşar. Bu durum, özellikle gradyanların geri yayılım sürecinde kaybolması ya da patlaması (vanishing/exploding

gradient) problemlerinden kaynaklanır (Bengio vd., 1994). LSTM, bu yapısal soruna karşı, hücre durumu (cell state) adı verilen ve bilgi akışının zaman boyunca korunmasını sağlayan bir yapı ile çözüm getirmiştir. Bu hücre durumu, belirli kapı mekanizmaları (girdi kapısı, unutma kapısı ve çıkış kapısı) aracılığıyla bilginin hangi oranda güncelleneceği, korunacağı ya da dışa aktarılacağı konusunda denetim sağlar (Hochreiter ve Schmidhuber, 1997).

LSTM hücresi, bilgi akışını düzenleyen üç temel kapıdan oluşur: unutma kapısı (forget gate), hücre durumundan hangi bilgilerin silineceğine karar verir; giriş kapısı (input gate), yeni bilgilerin ne kadarının hücreye dahil edileceğini belirler; çıkış kapısı (output gate) ise hücrenin ne tür bir çıktı vereceğine karar verir. Bu kapı mekanizmaları, doğrusal olmayan aktivasyon fonksiyonları (sigmoid ve hiperbolik tanjant) yardımıyla çalışarak hücre durumundaki bilgilerin akışını zaman boyunca optimize eder. Böylece LSTM, hem kısa hem de uzun süreli bağımlılıkları aynı anda işleyebilme kapasitesine sahip olur. Bu yapı, klasik RNN mimarilerine göre daha derin ve anlamlı bağlamların öğrenilmesine olanak tanır.

LSTM modelleri, özellikle zaman serisi tahmini, NLP, ses tanıma, finansal öngörüler ve biyomedikal sinyallerin analizi gibi uygulamalarda etkili biçimde kullanılmaktadır. Örneğin, NLP yönelik makine çevirisi, metin özetleme ve konuşma tanıma gibi görevlerde LSTM tabanlı sistemler oldukça başarılı sonuçlar vermektedir. Bu başarı, LSTM'nin geçmişteki bilgilere duyarlı kararlar alabilmesine bağlanmaktadır (Greff vd., 2017). Bununla birlikte, LSTM'nin öğrenme süreci oldukça parametre yoğundur ve hiperparametre ayarları performans üzerinde belirleyici rol oynar. Bu nedenle, modelin karmaşıklığı arttıkça hesaplama maliyetleri de artmaktadır.

Son yıllarda daha sade yapılı ve hesaplama açısından verimli olan Gated Recurrent Unit (GRU) gibi alternatif RNN mimarileri ve dikkat (attention) mekanizmasına dayalı Transformer modelleri, LSTM'nin bazı alanlardaki kullanımını kısıtlamaya başlamıştır (Vaswani vd., 2017). Ancak, LSTM'nin güçlü belleksel yapısı ve zamana bağlı örüntüleri yüksek doğrulukla öğrenebilme kapasitesi, onu birçok uygulama için halen tercih edilen bir yöntem haline getirmektedir. Bu bağlamda, LSTM ağları; yapay zekâ, veri bilimi ve mühendislik disiplinlerinde zamansal öğrenme problemleriyle mücadelede önemli bir çözüm aracı olmaya devam etmektedir.

2.4.3. Lineer Regresyon

Lineer regresyon, bir bağımlı değişken (sonuç/değişken) ile bir veya daha fazla bağımsız değişken (girdi/predictor) arasındaki ilişkiyi modellemek amacıyla kullanılan temel bir istatistiksel ve makine öğrenimi tekniğidir. Bu yöntemin hedefi, bağımsız değişkenlerin değerlerine dayalı olarak bağımlı değişkenin tahminini gerçekleştiren bir doğrusal fonksiyon üretmektir (Erdekli, 2022: 45-48). Bu model aracılığıyla, değişkenler arasındaki ilişkinin yönü ve şiddeti tahmin edilmekte ve ayrıca bu ilişkiden yola çıkarak geleceğe yönelik çıkarımlarda bulunulabilmektedir (Gujarati ve Porter, 2009). En basit haliyle bir lineer regresyon modeli 2.5. eşitliğindeki gibi ifade edilir.

$$y = \beta_0 + \beta_1 x + \varepsilon$$

2.5

Burada y bağımlı değişken, x bağımsız değişken, β_0 kesme noktası, β_1 eğim katsayısı ve ε hata terimi olarak gösterilir (Montgomery vd., 2012; Alpar, 2020). Lineer regresyon 5 adımda uygulanabilir;

- Veri hazırlama
- Doğrusallık analizi
- Modelin eğitimi
- Modelin performansı
- Sonuçların değerlendirilmesi

Lineer regresyon, parametrik bir yöntem olması nedeniyle belirli varsayımlara dayanmaktadır. Bu varsayımlar arasında; hata terimlerinin ortalamasının sıfır olması, varyanslarının sabitliği (homoskedastisite), hata terimlerinin bağımsız ve normal dağılımlı olması ve bağımsız değişkenler ile hata terimleri arasında korelasyonun olmaması yer almaktadır (Fox, 2016; Tatlıdil, 2010). Söz konusu varsayımların sağlanması durumunda model geçerliliğini ve güvenilirliğini korur. Aksi takdirde sapma, multikolinearite veya otokorelasyon gibi istatistiksel sorunlar modelin performansını düşürebilir (Karagöz, 2021; Wooldridge, 2016).

Modelin en önemli avantajları arasında uygulama kolaylığı, yorumlanabilirlik ve geniş kullanım alanları yer almaktadır. Lineer regresyon, sadece istatistiksel modelleme aracı olarak değil, aynı zamanda makine öğrenmesi bağlamında bir temel algoritma olarak da kabul edilmektedir (James vd., 2013; Yıldız, 2022). Nitekim regresyon analizi; sosyal bilimlerden mühendisliğe, ekonomiden psikolojiye kadar birçok alanda hem açıklayıcı hem de öngörücü analizler için kullanılmaktadır (Akgül ve Çevik, 2017; Seber ve Lee, 2012).

Uygulamada tek değişkenli (basit) ve çok değişkenli (çoklu) regresyon biçiminde sınıflandırılabilen lineer regresyon, aynı zamanda varyans analizi, faktör analizi ve yapısal eşitlik modellerinin temelini oluşturmaktadır (Kutner vd., 2005; Büyüköztürk, 2018). Örneğin ekonomide tüketim harcamaları ile gelir arasındaki ilişkiyi modellemek, eğitimde öğrenci başarısını etkileyen faktörleri belirlemek ya da sağlık alanında bir tedavi yönteminin etkisini incelemek gibi pek çok pratik uygulamada kullanılmaktadır (Ural ve Kılıç, 2013; Draper ve Smith, 1998).

Özellikle büyük veri analizi ve yapay zeka uygulamalarının yaygınlaştığı günümüzde, lineer regresyon algoritması, yüksek boyutlu veri kümelerinde açıklayıcı özellikleriyle ön analiz aracı olarak sıklıkla tercih edilmektedir (Hastie vd., 2009; Gürbüz ve Şahin, 2018). Bununla birlikte, model performansının artırılması amacıyla Ridge ve Lasso gibi düzenleme teknikleriyle de desteklenebilmektedir.

2.4.4. Karar Ağaçları

Karar ağaçları, hem sınıflandırma (classification) hem de regresyon (regression) problemlerinde yaygın olarak kullanılan, denetimli öğrenmeye dayalı bir makine öğrenimi algoritmasıdır. Yapısal olarak ağaç benzeri bir hiyerarşiye sahip olan bu yöntem, karmaşık veri yapılarını ayrıştırarak, karar süreçlerini anlaşılabilir ve izlenebilir hale getirir (Han, Pei ve Kamber, 2011). Karar ağacı, verilerin belirli niteliklerine dayalı olarak dallandırılması prensibine dayanır ve genellikle “kök düğüm” (root node), “iç düğümler” (internal nodes) ve “yaprak düğümler” (leaf nodes) olmak üzere üç temel yapıdan oluşur (Can, 2017).

Kök düğüm, ağacın başlangıç noktasını temsil eder ve tüm veri kümesini ifade eder. Bu noktada veri, maksimum bilgi kazancı sağlayacak ya da minimum safsızlık (impurity) değeri oluşturacak özniteliklere göre ayrılır. Bu seçim sürecinde kullanılan en yaygın kriterler arasında bilgi kazancı (information gain), Gini indeksi ve kazanç oranı (gain ratio)

yer alır (Quinlan, 1993). Bilgi kazancı, entropi kavramına dayanarak, veriyi homojen alt kümelere ayırma başarısını ölçerken; Gini indeksi ise sınıflar arasındaki karışıklık derecesini temel alır (Breiman vd., 1984).

Her bir iç düğüm, belirli bir değişkenin eşik değerine göre veri kümesini iki veya daha fazla alt kümeye ayırır. Süreç, verinin homojenliği artana dek devam eder ve nihayetinde karar ağacının yaprak düğümleri, son kararları veya sınıflandırma/tahmin sonuçlarını içerir (Öztemel, 2020). Özellikle sınıflandırma problemlerinde, yaprak düğümde tek bir sınıfın örneklerinin yer alması hedeflenir; bu durum modelin doğruluğunu artırabilir. Ancak fazla sayıda düğüm ve ayırım, modelin aşırı öğrenmesine (overfitting) neden olabilir. Bu nedenle budama (pruning) adı verilen tekniklerle gereksiz alt dalların kesilmesi önerilir. Budama işlemi, modelin genelleme yeteneğini artırmak ve veri setine fazla uyum sağlamasını engellemek amacıyla uygulanır (Kotsiantis, 2013).

Karar ağaçlarının en güçlü yönlerinden biri, şeffaflık ve açıklanabilirlik sağlamasıdır. Karar süreçlerinin mantıksal akışı kullanıcı tarafından kolayca izlenebilir ve her bir karar noktası, verideki hangi özelliğin ne şekilde etkili olduğunu açıkça gösterir (Köse ve Ersöz, 2020). Bu özellik, özellikle sağlık, hukuk ve finans gibi karar süreçlerinin denetlenebilir olması gereken alanlarda kritik öneme sahiptir. Karar ağaçları, ayrıca hem sayısal hem de kategorik verileri işleyebilir ve eksik veri barındıran veri kümelerinde esnek çözümler sunabilir (Tokalaçoğlu, 2020).

Bir başka avantajı ise, çok değişkenli veri setlerinde etkili şekilde çalışabilmesidir. Karar ağaçları, birden fazla özelliği dikkate alarak, veriler arasında önemli etkileşimleri ortaya çıkarabilir. Ayrıca, değişken önem skorları (feature importance) hesaplanarak, hangi değişkenlerin karar sürecinde daha etkili olduğu analiz edilebilir (Hastie vd., 2009). Bu durum, özellik seçimi (feature selection) gibi ön işleme adımlarında faydalı olabilir.

Öte yandan, karar ağaçları bazı dezavantajlara da sahiptir. Özellikle saf karar ağaçları, küçük veri setlerinde veya gürültülü verilerde yüksek varyans gösterebilir. Ağaç çok derinleştikçe model karmaşılaşır ve bu durum overfitting'e neden olabilir. Bu tür sorunların üstesinden gelmek için genellikle topluluk yöntemleri (ensemble methods) kullanılır. Örneğin, RF algoritması, çok sayıda karar ağacının bir araya getirilerek oy çokluğuna dayalı kararlar üretmesini sağlar ve tekil bir karar ağacına kıyasla genellikle daha yüksek doğruluk sağlar (Breiman, 2001).

Karar ağaçları, makine öğrenmesi ve istatistiksel modelleme dışında, sosyoekonomik analizlerden pazarlama stratejilerine, sağlık sistemlerinden mühendislik karar destek sistemlerine kadar geniş bir uygulama yelpazesine sahiptir (Lemaitre vd., 2017). Eğitim bilimlerinde öğrenci başarılarının belirlenmesi, finans sektöründe kredi risk analizleri ve sağlık alanında hastalık teşhis süreçlerinde yaygın olarak kullanılmaktadır.

2.4.5. Genetik Algoritma

Genetik algoritmalar (GA), biyolojik evrim süreçlerinden esinlenen, arama ve optimizasyon problemlerine çözüm üretmek amacıyla geliştirilen sezgisel yöntemlerdir. İlk olarak Holland (1975) tarafından sistematik biçimde teorik temelleri ortaya konulan bu yöntemler, Darwin'in doğal seçim ilkesine dayalı olarak, bir çözüm popülasyonunun zaman içerisinde evrimleşerek daha iyi sonuçlar üretmesini sağlar.

GA, çözüm uzayında bulunan bireyleri genetik yapılar şeklinde temsil ederek, bu bireyleri bir popülasyon çerçevesinde değerlendirir. Bireylerin başarısı, genellikle bir uygunluk (fitness) fonksiyonu aracılığıyla ölçülür. Seçim (selection), çaprazlama (crossover) ve mutasyon (mutation) gibi genetik işlemler sayesinde, daha uygun bireylerin bir sonraki nesle aktarılması sağlanır ve böylece evrimsel gelişim gerçekleştirilir (Goldberg, 1989).

Seçim aşamasında yaygın olarak kullanılan yöntemler arasında turnuva seçimi, rulet tekerleği seçimi ve sıralama seçimi yer alırken; çaprazlama işlemleri tek noktalı (single-point), çok noktalı (multi-point) veya üniform (uniform) biçiminde uygulanabilmektedir. Mutasyon işlemi ise, kromozom üzerinde küçük ölçekli rastgele değişiklikler yaparak çözüm çeşitliliğini korumakta ve yerel minimumlara sıkışmayı önlemektedir (Mitchell, 1998; Haupt ve Haupt, 2004). Bu mekanizmaların birlikte çalışması, GA'ların hem keşif (exploration) hem de sömürü (exploitation) süreçlerini dengeleyebilmesini sağlar.

GA'ların en belirgin avantajlarından biri, çözüm uzayının geniş bölgelerini etkili şekilde tarayabilme kapasitesidir. Özellikle türev bilgisine ihtiyaç duyulmayan, çok boyutlu, süreksiz ve çoklu yerel çözüme sahip problemlerde oldukça etkilidirler (Konak vd., 2006). Ayrıca paralel hesaplama altyapılarına uygunlukları sayesinde büyük ölçekli optimizasyon problemlerinde yüksek verimlilikle uygulanabilirler (Alba ve Tomassini, 2002). Ancak bu

avantajlara karşın, GA'ların optimal parametre ayarlamaları (örneğin popülasyon büyüklüğü, mutasyon oranı, çaprazlama oranı vb.) konusunda duyarlılık göstermesi, algoritmanın başarısını doğrudan etkileyebilir. Ayrıca yüksek hesaplama maliyeti ve zaman karmaşıklığı da dikkat edilmesi gereken sınırlılıklar arasındadır (Eiben ve Smith, 2003).

GA, çok çeşitli uygulama alanlarında başarılı şekilde uygulanmaktadır. Mühendislikte yapısal tasarım ve üretim planlamasından, finans sektöründe portföy optimizasyonu ve risk yönetimine; makine öğrenmesinde hiperparametre optimizasyonundan, biyoinformatikte gen dizilim analizi ve protein yapısı tahminine kadar geniş bir yelpazede kullanılmaktadır (Fogel, 1999; Sivanandam ve Deepa, 2008). Robotik sistemlerde ise hareket planlama ve otonom karar verme süreçlerinde önemli katkılar sunmaktadır (Lin vd., 2011).

2.4.6. Parçacık Sürü Algoritması

Parçacık Sürü Optimizasyonu (PSO), kuş sürüsü ya da balık grubu gibi toplu davranış gösteren hayvanların doğadaki yön bulma ve besin arama stratejilerinden esinlenerek geliştirilen, popülasyon temelli bir sezgisel optimizasyon algoritmasıdır. Kennedy ve Eberhart (1995) tarafından geliştirilen bu yöntem, her bir aday çözümü bir parçacık olarak tanımlar ve bu parçacıkların kolektif davranışlarını taklit ederek arama uzayında küresel optimumu bulmayı amaçlar. Parçacıklar, çözüm uzayında rastgele bir şekilde yerleştirilir ve her biri kendi deneyiminden (bireysel en iyi – personal best, pBest) ve topluluğun en iyi deneyiminden (küresel en iyi – global best, gBest) faydalanarak yön ve hızlarını iteratif biçimde günceller.

PSO algoritmasında her parçacık, bir çözümün koordinatlarını temsil ederken aynı zamanda sahip olduğu hız vektörü ile bir sonraki adımda hangi yöne ve ne büyüklükte hareket edeceğine karar verir. Bu hız ve pozisyon güncellemeleri, aşağıdaki denklemlerle ifade edilir:

$$v_i(t + 1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (pBest_i - x_i(t)) + c_2 \cdot r_2 \cdot (gBest - x_i(t)) \quad 2.6$$

Burada,

- $v_i(t)$: i . parçacığın t anındaki hızını,
- $x_i(t)$: i . parçacığın t anındaki pozisyonunu,
- $pBest_i$: i . parçacığın şimdiye kadar ulaştığı en iyi pozisyonu,
- $gBest$: tüm parçacıklar arasında ulaşılan en iyi pozisyonu,
- w : atalet ağırlığını,
- c_1, c_2 : öğrenme katsayılarını,
- r_1, r_2 : $[0,1]$ aralığında rastgele sayılardır (Kennedy ve Eberhart, 1995; Shi ve Eberhart, 1998).

y : nöronun çıktısını temsil eder

Algoritmanın başarısı, bireysel keşif ile kolektif istihbarat arasındaki dengenin iyi kurulmasına bağlıdır. Atalet ağırlığı parametresi, parçacığın önceki yönelimine ne ölçüde bağlı kalacağını belirlerken; bilişsel (c_1) ve sosyal (c_2) katsayılar sırasıyla bireysel deneyim ve topluluk etkisini yansıtmaktadır. Bu yapı, algoritmaya hem yerel hem de küresel optimuma yakınsama potansiyeli kazandırır (Poli vd., 2007).

PSO'nun avantajları arasında, algoritmanın uygulanmasının kolay olması, az sayıda ayarlanması gereken parametreye sahip olması, süresiz, doğrusal olmayan ve yüksek boyutlu problemlerde etkili sonuçlar verebilmesi yer almaktadır (Clerc, 2006; Engelbrecht, 2005). Bununla birlikte, bazı dezavantajları da mevcuttur. Özellikle çok modlu fonksiyonlarda lokal minimumlara takılma riski, aşırı erken yakınsama ve uygunluk fonksiyonuna bağımlılık gibi problemler optimizasyon başarısını olumsuz etkileyebilir (Parsopoulos ve Vrahatis, 2002).

PSO algoritması mühendislik tasarımı, YSA'nın eğitimi, finansal modelleme, biyoinformatik, robotik sistem kontrolü ve enerji sistemleri gibi çok çeşitli alanlarda yaygın olarak kullanılmaktadır (Wang vd., 2018; Bonyadi ve Michalewicz, 2017). Özellikle

hiperparametre optimizasyonu gibi karmaşık problem alanlarında, klasik deterministik yöntemlere kıyasla daha esnek ve etkili çözümler sunabilmektedir (Liu vd., 2020).

2.4.7. Yapay Arı Kolonisi

Yapay Arı Kolonisi (YAK) algoritması, doğadaki bal arılarının besin arama ve sömürme davranışlarını model alan, popülasyon temelli, sezgisel bir optimizasyon yöntemidir. Bu algoritmanın temelinde, arı kolonilerinin çevredeki nektar kaynaklarını etkin bir şekilde keşfetme, değerlendirme ve bunlar arasında bilgi paylaşımı yoluyla kolektif bir karar alma süreçleri yer alır (Pham vd., 2006; Basturk ve Akay, 2006).

YAK algoritması; işçi arılar (employed bees), izci arılar (scout bees) ve gözcü arılar (onlooker bees) olmak üzere üç temel arı tipinden oluşur. İşçi arılar, belirli bir çözüm noktasına (yani nektar kaynağına) atanarak bu noktanın kalitesini (uygunluk değerini) değerlendirir ve bu bilgi kolonideki diğer arılarla paylaşılır. Gözcü arılar, işçi arıların geri bildirimlerine göre daha yüksek kaliteli kaynaklara yönelerek daha iyi çözümler üretmeye çalışır. İzci arılar ise, sömürülme sürecinde düşük verim sağlayan kaynakları terk ederek arama alanının farklı bölgelerinde rastgele arama yaparlar ve böylece çeşitlilik sağlanarak yerel minimumlara takılma riski azaltılır (Zhu ve Kwong, 2010; Cui vd., 2013).

YAK algoritmasının temel işlem adımları şu şekilde özetlenebilir: Başlangıçta, arı sayısı ve maksimum iterasyon gibi algoritma parametreleri belirlenir. Daha sonra her arı, arama uzayında rastgele bir pozisyona yerleştirilerek başlangıç çözümleri oluşturulur. Her çözümün uygunluk değeri değerlendirilir. İşçi arılar, mevcut çözümlerinden yola çıkarak komşu çözümler üretir; bu süreçte daha iyi bir çözüm bulunduğunda mevcut çözüm güncellenir. Ardından gözcü arılar, işçi arılardan elde edilen bilgilere dayanarak seçici biçimde daha iyi çözüm noktalarına yönelir. Belirli bir iterasyon boyunca iyileşme göstermeyen çözümler, izci arılar tarafından terk edilir ve yerlerine yeni rastgele çözümler üretilir. Bu döngü, maksimum iterasyon sayısına veya önceden belirlenen uygunluk kriterine ulaşıncaya kadar sürdürülür (Karaboga ve Basturk, 2008; Gao vd., 2015).

YAK algoritmasının en güçlü yönlerinden biri, düşük sayıda kontrol parametresine (örneğin izci arı sayısı, iterasyon sayısı) sahip olması ve uygulamasının görece basit bir yapıda bulunmasıdır. Aynı zamanda, arama alanında geniş kapsamlı bir keşif yeteneğine sahip olması sayesinde, karmaşık, çok boyutlu ve doğrusal olmayan optimizasyon

problemlerinde etkili çözümler sunabilmektedir (Bansal vd., 2019). Özellikle sürekli ve ayrık optimizasyon, fonksiyon optimizasyonu, makine öğrenmesi, ağ tasarımı ve mühendislik tasarımı gibi farklı alanlarda başarıyla uygulanmıştır (Akay ve Karaboga, 2012; Kumar vd., 2020).

2.4.8. Rastgele Orman Algoritması

Rastgele Orman (RF) algoritması, makine öğrenmesi alanında tahminleme ve sınıflandırma problemlerinin çözümünde yaygın olarak kullanılan, topluluk (ensemble) öğrenme temelli bir yöntemdir. İlk olarak Leo Breiman tarafından 2001 yılında önerilen bu algoritma, karar ağaçlarının zayıf yönlerini azaltarak daha güvenilir ve genellenebilir sonuçlar elde etmeyi hedefler (Breiman, 2001).

RF yöntemi, temel olarak birden fazla karar ağacının birlikte çalışmasına dayanır. Her bir karar ağacı, eğitim verisinin farklı bir alt kümesi üzerinde inşa edilir. Bu alt kümeler, "bootstrapping" adı verilen tekrar örnekleme tekniğiyle oluşturulur. Her ağacın eğitildiği veri seti birbirinden farklı olduğu için ağaçlar arasında çeşitlilik sağlanır ve bu da topluluk içinde hataların ortalama alınarak azaltılmasını sağlar (James vd., 2013).

Modelin ikinci temel bileşeni, her düğümde kullanılan özniteliklerin rastgele bir alt kümesinin seçilmesidir. Bu rastgele özellik seçimi, karar ağaçlarının korelasyonunu azaltır ve modelin çeşitliliğini artırır. Bu sayede, belirli özniteliklerin tüm ağaçlarda baskın hale gelmesinin önüne geçilir. Bu teknik, özellikle yüksek boyutlu veri setlerinde özniteliklerin önem sırasının belirlenmesi açısından avantaj sağlar (Louppe vd., 2013).

RF algoritması hem sınıflandırma hem de regresyon problemlerinde başarıyla kullanılabilir. Sınıflandırma problemlerinde her ağacın tahmini oylama yöntemiyle değerlendirilir ve en fazla oyu alan sınıf nihai sonuç olarak kabul edilir. Regresyon problemlerinde ise tüm ağaçlardan elde edilen tahminlerin aritmetik ortalaması alınarak sonuç elde edilir (Hastie vd., 2009).

Algoritmanın başlıca avantajları şu şekilde sıralanabilir:

1. Aşırı öğrenmeye karşı dirençlidir: Karar ağaçlarının bireysel olarak aşırı uyum eğilimleri, RF'de birleştirilmiş yapılar sayesinde dengelenir.

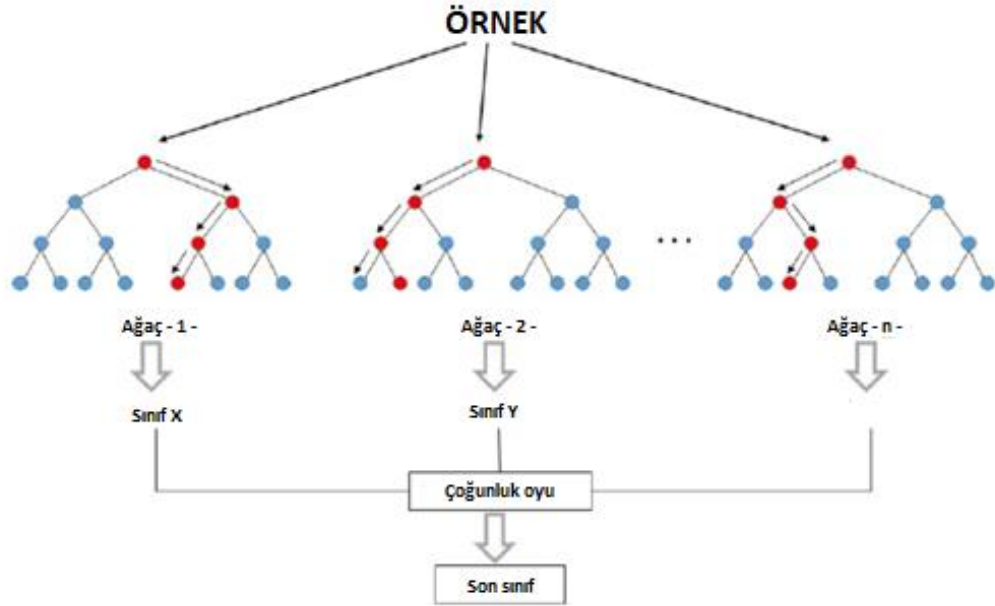
2. Değişken önem sıralaması yapabilir: Model, her bir özneliliğin hedef değişken üzerindeki etkisini ölçerek analiz yapılmasını mümkün kılar (Genuer vd., 2010).

3. Kayıp verilere karşı dayanıklıdır: Eksik verilerle baş edebilme yeteneği, modelin uygulama alanlarını genişletir.

4. Paralel işlemeye uygundur: Her bir karar ağacı bağımsız olarak inşa edilebildiği için paralel hesaplama ortamlarında etkin bir şekilde çalıştırılabilir.

Bu avantajlar sayesinde RF algoritması, finansal risk analizi (Lemieux vd., 2019), tıbbi teşhis sistemleri (Delen vd., 2005), genetik veri analizi (Díaz-Uriarte ve De Andres, 2006), uzaktan algılama ve görüntü işleme (Belgiu ve Drăguț, 2016) gibi çok çeşitli disiplinlerde başarıyla uygulanmaktadır.

Bununla birlikte, algoritmanın bazı sınırlılıkları da bulunmaktadır. Örneğin, çok büyük sayıda ağaç kullanıldığında modelin eğitim süresi uzayabilir ve bellekte yüksek miktarda kaynak tüketebilir. Ayrıca, modelin iç yapısının anlaşılması ve yorumlanması tekil karar ağaçlarına kıyasla daha karmaşıktır (Zhou, 2012).



Kaynak: <https://kaanugurluoglu123.medium.com/random-forest-rassal-orman-algoritmas%C4%B1-machine-learning-41e9ad655133>

Şekil 2.1. Rastgele Orman Algoritması

2.4.9. Tabu Arama

Tabu Arama Algoritması (TS), Fred Glover tarafından 1986 yılında geliştirilmiş ve o tarihten itibaren kombinatoriyal optimizasyon problemlerinde yaygın olarak kullanılmakta olan güçlü bir meta-sezgisel yöntemdir (Glover, 1986). Geleneksel yerel arama tekniklerinin temel zayıflığı olan yerel minimum tuzaklarına düşme problemi, TS ile akıllı bellek yapıları ve kısıtlamalar (tabu listesi) sayesinde aşılmaktadır. Böylelikle algoritma, çözüm uzayında daha geniş bir keşif yaparak, global optimuma yakın çözümler elde etme potansiyelini artırır (Nowicki ve Smutnicki, 2005).

TS, başlangıçta bir başlangıç çözümüyle başlar ve iteratif olarak komşuluk (neighborhood) kavramı üzerinden yeni çözümler üretir. Her iterasyonda, mevcut çözümün komşu çözümleri değerlendirilir ve tabu listesinde kayıtlı olmayan ya da aspirasyon kriterine uyan en iyi komşu çözüm seçilir (Battiti ve Tecchiolli, 1994). Tabu listesi, belirli bir süre boyunca yapılan hareketlerin veya çözüm özelliklerinin tekrar edilmesini engelleyen bir bellek yapısıdır ve algoritmanın arama sürecinde tekrarlayan çözümlerden kaçınmasını sağlar (Glover, 1990).

Aspirasyon kriteri, tabu kısıtlamasına rağmen, daha iyi bir çözüm elde edilirse bu hareketin kabul edilmesini sağlayan esnek bir mekanizmadır. Bu sayede algoritma, tabu listesi nedeniyle kaçırabileceği daha kaliteli çözümleri değerlendirme şansı bulur (Fleurent ve Glover, 1993).

TS algoritması, arama sürecinde etkinliği artırmak için iki önemli strateji uygular: yoğunlaştırma (intensification) ve çeşitlendirme (diversification). Yoğunlaştırma, arama sürecini iyi çözümlerin bulunduğu bölgelerde yoğunlaştırarak detaylı inceleme yapılmasını sağlarken, çeşitlendirme algoritmanın çözüm uzayının farklı bölgelerinde yeni ve çeşitli çözümler keşfetmesini mümkün kılar (Glover ve Laguna, 1997). Bu stratejiler, algoritmanın hem derinlemesine hem de geniş kapsamlı bir arama yapmasına olanak tanır.

Tabu aramanın en büyük avantajı, yerel optimum tuzaklarından kaçabilmesi ve daha geniş bir çözüm uzayında etkin arama yapabilmesidir (Taillard, 1991). Bununla birlikte, parametre ayarlarının (tabu listesi uzunluğu, aspirasyon kriteri, yoğunlaştırma ve çeşitlendirme frekansı gibi) probleme özgü olarak dikkatli yapılması gerekmektedir. Bu parametrelerin uygun seçilmemesi durumunda algoritmanın performansı düşebilir (Battiti ve

Tecchiolli, 1994). Ayrıca, büyük ölçekli problemlerde hesaplama süresi yüksek olabilir, ancak paralel ve hibrit yaklaşımlarla bu sorunlar kısmen giderilebilmektedir (Battiti ve Tecchiolli, 1994; Johnson, 1990).

2.4.10. Karınca Kolonisi Algoritması

Karınca Kolonisi Optimizasyonu (ACO), Dorigo ve arkadaşları tarafından 1992 yılında ortaya atılan ve doğadan ilham alan bir meta-sezgisel optimizasyon yöntemidir. Bu algoritma, gerçek karınca kolonilerinin yiyecek kaynaklarına olan en kısa yolları bulma davranışlarını modelleyerek, özellikle kombinatorial optimizasyon problemlerinde etkili sonuçlar üretmektedir (Dorigo vd., 1996).

ACO, çok sayıda yapay karıncanın bir arada çözüm uzayında dolaşarak uygun çözümler oluşturması esasına dayanır. Gerçek karıncalar gibi, yapay karıncalar da çevrelerinde feromon adı verilen kimyasal bir madde bırakırlar. Bu feromonlar, karıncaların hangi yolları daha sık kullandıklarını temsil eder ve yüksek feromon yoğunluğuna sahip yollar daha cazip hale gelir. Böylece, algoritma çözümler arasında pozitif geri besleme mekanizması ile en iyi rotaları keşfeder (Dorigo ve Stützle, 2004).

Her iterasyonda, karıncalar mevcut çözümün komşuluğunda rastgele ama feromon değerlerine bağlı olarak seçimler yapar. Bu seçimlerde hem feromon miktarı hem de yolun yerel kalitesi (örneğin mesafe veya maliyet) dikkate alınır. Karıncaların oluşturduğu çözümler değerlendirilir ve en iyi çözümlerden kalan feromon miktarı artırılır. Ayrıca, feromonun zamanla buharlaşmasıyla eski ve zayıf çözümlerin etkisi azalır, bu da algoritmanın çeşitlendirme kabiliyetini güçlendirir (Coloni vd., 1992).

Feromon güncellemesi ise hem yerel hem de global olarak yapılır. Global güncelleme, en iyi çözüme göre feromon seviyesini artırırken, yerel güncelleme karıncaların geçtiği yolların feromon seviyesini biraz azaltarak arama çeşitliliğini sağlar (Stützle ve Hoos, 2000).

KKO'nun en önemli avantajları arasında sezgisel doğası sayesinde geniş çözüm uzaylarında etkili arama yapabilmesi, kolayca paralelleştirilebilmesi ve hem keşif (diversification) hem de sömürü (intensification) dengesi kurabilmesi sayılabilir (Stützle ve Hoos, 2000). Öte yandan, algoritmanın performansı parametre ayarlarına (feromon etkisi,

buharlařma oranı, karınca sayısı vb.) olduka duyarlıdır ve kt ayarlandığında erken yakınsamaya (premature convergence) yol aabilir (Bullnheimer vd., 1999). Ayrıca, yksek hesaplama maliyeti ve byk problem boyutlarında yavaşlama dezavantajları arasında yer almaktadır.

ACO'nun performansını artırmak iin hibrit yntemler geliřtirilmiřtir. GA, TS ve YSA ile kombinasyonları sayesinde daha hızlı ve daha kaliteli mler elde edilmektedir (Colormi vd., 1991). Ayrıca, parametre optimizasyonu iin adaptif ve ğrenen feromon gncelleme stratejileri zerinde yoğun alıřmalar yapılmaktadır (Socha ve Dorigo, 2008).

2.4.11. Ekstra Gradyan Artırma Algoritması

Ekstra Gradyan Artırma Algoritması (XGBoost), Chen ve Guestrin tarafından 2016 yılında geliřtirilen ve makine ğrenmesi alanında zellikle tabular veri zerinde yksek bařarı saėlayan bir gradyan artırma algoritmasıdır. XGBoost, geleneksel gradyan artırma yntemlerine kıyasla daha hızlı ve dzenli alıřacak řekilde optimize edilmiřtir. Hem tahmin performansı hem de hesaplama verimliliėi aısından dikkat eken bu algoritma, doėrusal ve aėa tabanlı modellerin avantajlarını bir araya getirerek gl bir ğrenme erevesi sunmaktadır (Chen ve Guestrin, 2016).

Algoritmanın temelinde ardışık olarak oluřturulan zayıf ğreniciler yer almaktadır. Bu ğreniciler genellikle karar aėalarıdır ve her yeni model, bir nceki modelin hatalarını azaltmaya odaklanır. XGBoost, bu sreci daha etkili hle getirmek amacıyla kayıp fonksiyonuna dzenlileřtirme terimleri ekler. Bylece modelin karmařıklıėı kontrol altına alınırken, ařırı ğrenme (overfitting) riski azaltılır. XGBoost'un karar aėaları, belirli bir yaprak sayısına ve yaprak aėırlıklarına gre modellenir. Bu yapı, modelin hem doėruluėunu hem de genellenebilirliėini artırmak iin zel olarak tasarlanmıřtır (Friedman, 2001).

Bir XGBoost modeli, K sayıda regresyon aėaının toplamından oluřur. Her aėa, veri noktalarına tahmin deėerleri atar ve modelin tahmini:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F}$$

2.7

şeklindedir. Burada, $\mathcal{F}$ ağaç fonksiyonları (regresyon ağaçları) kümesini temsil eder. Modelin amacı, aşağıdaki objektif fonksiyonu minimize etmektir:

$$\mathcal{L}(\Phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad 2.8$$

Burada, l kayıp fonksiyonu, Ω ise model karmaşıklığını kontrol eden düzenleme terimidir. Düzenleme terimi genellikle yaprak sayısı ve yaprak ağırlıklarının büyüklüğünü içerir:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad 2.9$$

Burada T yaprak sayısı, w_j yaprak ağırlıkları, γ ve λ ise düzenleme parametreleridir (Chen ve Guestrin, 2016).

XGBoost'un en önemli teknik avantajlarından biri paralel ve dağıtık hesaplama yapabilmesidir. Ağaç büyüme işlemleri sırasında veri parçalanarak paralel şekilde işlenebilir, bu da büyük veri setlerinde modelin çok daha hızlı çalışmasını sağlar. Ayrıca algoritma, eksik değerlerle başa çıkma konusunda oldukça etkilidir; model, eksik veri içeren durumlarda bile otomatik olarak en uygun yönlendirmeyi yaparak öğrenme sürecini sürdürebilir (Chen ve He, 2015). Diğer yandan, modelin yapısında yer alan L1 ve L2 düzenleme teknikleri, karmaşık model yapılarını sadeleştirerek genel performansı artırır.

XGBoost, birçok gerçek dünya problemine kolayca uygulanabilmesi ve yüksek doğruluk sunması nedeniyle, özellikle finansal tahminleme, pazarlama analitiği, sağlık verisi analizi, biyoinformatik ve endüstriyel süreç kontrolü gibi alanlarda yaygın olarak kullanılmaktadır. Kaggle gibi veri bilimi yarışmalarında da popülerliği yüksektir; bu platformlarda kazanan modellerin önemli bir kısmı XGBoost üzerine kuruludur (Ke vd., 2017).

XGBoost'un avantajları arasında yüksek doğruluk oranı, hızlı eğitim süreci, eksik verilerle başa çıkabilme kabiliyeti, aşırı öğrenmeye karşı dayanıklılığı ve geniş konfigürasyon olanakları yer almaktadır. Bununla birlikte, modelin hiperparametrelerinin dikkatli şekilde ayarlanması gerekmektedir; aksi takdirde modelin performansı düşebilir. Ayrıca, modelin yapısının yorumlanması zor olduğundan, şeffaflık ve açıklanabilirlik gerektiren alanlarda sınırlı kullanılabilirlik sunar. Yine de, modelin SHAP (SHapley Additive exPlanations) gibi açıklama yöntemleriyle yorumlanabilirliği artırılabilir (Prokhorenkova vd., 2018).

2.5. Literatürde Yer Alan Çalışmalar

Yapay zekâ, insan zekasına benzer düşünme ve karar alma süreçlerini taklit eden sistemlerin genel adıdır. Yapay zekânın tarihi, 1950'lere kadar uzanmakta olup, bu süreç içerisinde birçok farklı yöntem ve alt alan gelişmiştir.

Literatürdeki çalışmalar detaylı bir şekilde incelendiğinde, yapay zekâ yöntemlerinin hemen hemen tüm alanlarda kullanıldığı görülmektedir. Hem kolay uygulanabilir hem de karmaşık süreçlerle başa çıkabilme gücünden dolayı çeşitli alanlarda yüksek performans ile öne çıkmaktadır. Bu kullanım alanlarına örnekler geniş bir yelpazede ele alınabilir.

Bishop (2006), makine öğreniminin, yapay zekâ alanında en çok bilinen yöntemlerden biri olduğunu ifade etmiştir. Makine öğreniminin, sistemlerin verilerden öğrenmesini ve bu veriler üzerinden tahminlerde bulunmasını sağlayan bir teknik olduğunu vurgulamıştır. Bu yöntem, verilerle eğitilen algoritmalar aracılığıyla belirli görevleri yerine getirebilmek için önemli bir çerçeve sunar. Denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme gibi alt kategorilere ayrılan makine öğrenimi, çeşitli uygulamalarda çözüm önermektedir. Örneğin, denetimli öğrenme modeli, etiketlenmiş verileri kullanarak bir model oluşturur ve bu model, yeni verilere uygulandığında doğru sonuçlar vermeyi hedefler.

Makine öğrenimi, sistemlerin verilerden öğrenmesini ve bu öğrenme sürecini kullanarak yeni bilgiler edinmesini sağlayan bir yapay zekâ alt dalıdır. Makine öğrenimi, algoritmaların istatistiksel yöntemler kullanarak veri setlerinden model oluşturmaya olanak tanır. Örneğin, sınıflandırma, regresyon ve kümeleme gibi teknikler, verilerin analizi ve yorumlanmasında önemli rol oynamaktadır. Sınıflandırma, verileri belirli kategorilere

ayırırken, regresyon, değişkenler arasındaki ilişkileri tahmin etmeye yarar. Kümeleme ise, benzer özelliklere sahip veri noktalarını gruplamak için kullanılır. Makine öğrenimi, büyük veri analizi, öneri sistemleri ve dolandırıcılık tespiti gibi geniş bir uygulama yelpazesine sahiptir (Kılıç, 2014).

Esteva vd. (2017) tarafından yapılan bir çalışma, derin öğrenme algoritmalarının, görüntülerden cilt kanserinin tanımlanmasında Kurul tarafından sertifikalandırılmış dermatologlarla karşılaştırılabilir tanılabilir hassasiyete ulaşabileceğini göstererek, yapay zekânın erken teşhisteki potansiyelini vurgulamıştır. Bu ilerleme, yalnızca tanı doğruluğunu güçlendirmekle kalmaz, aynı zamanda sağlıkçıların yükünü de azaltır.

Ekrem vd. (2020) çalışmalarında, RF ve PSO kullanılarak, 303 bireyin kalp hastalığı durumu sınıflandırılmıştır. Özellik seçimi PSO yöntemi ile yapılmış ve veri seti, RF algoritması ile eğitilmiştir. RF sınıflandırma modeli, doğruluk, özgüllük, duyarlılık, kesinlik, F-ölçüsü ve ROC eğrisi gibi performans değerlendirme kriterleri açısından başarı oranı değerlendirilmiştir. Yapılan değerlendirme sonucunda RF sınıflandırması ile başarılı tahminler gerçekleştirdiği tespit edilmiştir.

Panc vd. (2020) çalışmalarında, yapay zekânın sağlık sistemlerindeki olası uygulamalarını incelemişlerdir. Ayrıca, yapay zekânın kamu sağlığı ile sağlık hizmetlerinin verimliliğini, etkinliğini, eşitliğini ve yanıt verme kapasitesini artırarak Evrensel Sağlık Kapsamı hedeflerine ulaşmak için sağlık sistemlerini nasıl dönüştürebileceğini ele almışlardır. Derin öğrenme tekniklerini kullandıkları çalışmalarında başarılı sonuçlar elde etmişlerdir.

Karar ağaçları ve SVM gibi algoritmalar, sağlık alanında analizde büyük ilerlemeler göstermiştir. Örneğin, onkolojide Lee ve Yoon (2021), hastanın kemoterapiye yanıtını tahmin etmek için bu modellerin kullanıldığı çalışmalara atıfta bulunarak kişiselleştirilmiş tedavi planlarını açığa çıkarmışlardır. Bu teknikler, doktorlar için kolayca görünmeyen modelleri belirlemek için elektronik tıbbi kayıtlardan, genomik veri tabanlarından ve klinik çalışmalardan gelen geniş veri kümelerini analiz eder. Hastaların demografik verileri, klinik öyküleri ve biyolojik belirteçleri gibi özellikleri entegre ederek, tahminlerin doğruluğunu artırır ve kişiselleştirilmiş tedavi stratejilerine katkıda bulunarak, tedavi rejimlerinde geleneksel olarak kullanılan test yaklaşımlarını ve hataları önemli ölçüde azaltır.

Ayrıca, derin öğrenme algoritması, görüntüleme tanısında önemli derecede yol katetmiştir. Görsel değerlendirmelere bağlı bir alan olan Radyoloji, X ışınları, manyetik rezonans görüntüleme ve sağlık taramaları gibi tıbbi görüntülerin yorumlanmasına yardımcı olan yapay zekâ algoritmalarından faydalanmıştır. Lee ve Yoon (2021), evrimsel sinir ağlarının (CNN) zatürre ve meme kanseri gibi hastalıkları tespit etmede geleneksel yöntemlerden daha iyi performans gösterebileceğini savunmaktadır. Bu modeller, büyük görüntü kümeleri üzerinde eğitim vererek yalnızca tanı sürecini hızlandırmakla kalmaz, aynı zamanda radyolojide yaygın bir zorluk olan gözlemciler arası değişkenliği azaltmaya da yardımcı olur. Yapay zekâ aracılığıyla radyologların becerilerindeki artış, yalnızca iş akışını optimize etmekle kalmaz, aynı zamanda tanı doğruluğunu da artırarak görüntüleme ve tarama uygulamalarında dönüştürücü bir hareketi temsil eder.

Sağlık alanında kullanılan yapay zekâ teknikleri sayesinde teşhis süreçlerinin hızlandığı ve hastalıkların erken aşamalarında tespit edildiği öne çıkmaktadır. Görüntü işleme teknikleri, radyoloji alanında, tıbbi görüntüleri analiz ederek, hastalıkların hızlı ve doğru bir şekilde teşhis edilmesine yardımcı olmaktadır. Bunun yanı sıra, yapay zekâ sistemleri, hastaların sağlık verilerini bir araya getirerek, potansiyel sağlık sorunlarını öngörme ve yönetme süreçlerinde de önemli katkılarda bulunmaktadır (Ertürk ve Ertürk, 2021). Yapay zekânın, genetik bilgi analizi ve kişiselleştirilmiş tedavi yaklaşımları gibi ileri düzey uygulamalarla sağlık hizmetlerinin kalitesini ve etkinliğini artırması da dikkat çekmektedir.

Derin öğrenme, makine öğreniminin bir dalı olarak tanımlanmaktadır ve genellikle çok katmanlı sinir ağları kullanarak daha karmaşık veri yapıları üzerinde işlem yapmayı hedefler. Derin öğrenme, verilerin katmanlar arasında aktarıldığı ve işlenerek daha üst düzey temsili elde edildiği bir öğrenme sürecidir. Özellikle görüntü tanıma, ses tanıma ve NLP alanlarında yüksek başarı oranları sayesinde dikkat çekmektedir. Görüntü tanıma uygulamalarında, derin öğrenme algoritmaları, bilgisayarlara nesneleri ve yüzleri tanıma yeteneği kazandırmakta, bu da otomatik güvenlik sistemleri ve sosyal medya platformları gibi uygulamalarda kullanılmaktadır. Ses tanıma sistemleri de benzer şekilde, kullanıcının sesine yanıt verebilen cihazların geliştirilmesinde derin öğrenme algoritmalarını etkin bir şekilde kullanmaktadır (Kurtboğan, 2023).

NLP, insan dilinin bilgisayarlar tarafından anlaşılmasını, yorumlanmasını ve tepki verilmesini sağlayan bir yapay zekâ alanıdır. NLP, dilin yapısal ve anlamsal özelliklerini dikkate alarak, metin analizi, dil modelleme, sohbet botları ve makine çevirisi gibi uygulamalarda kullanılmaktadır. Bu yöntemler, dilin karmaşıklığını ve çok katmanlı doğasını anlamaya yönelik çeşitli teknikler ve algoritmalar gerektirir. NLP'nin en önemli zorluklarından biri, dilin bağlamına bağlı olarak değişen anlamları ve kullanımları doğru bir şekilde takdir etmektir. Bununla birlikte, son zamanlarda derin öğrenme tekniklerinin NLP alanında uygulanması, metinlerin daha doğru ve anlamlı analizini sağlamıştır. Bu durum, insan benzeri etkileşimleri mümkün kılmakta ve çeşitli chatbot'lar ile sanal asistanların gelişiminde önemli bir rol oynamaktadır (Jurafsky ve Martin, 2023).

Yapay zekâ tekniklerinin finansal piyasalardaki tahmin süreçleri üzerindeki etkisi, geçmiş yıllardan günümüze kadar giderek daha fazla önem kazanan bir araştırma alanı olmuştur. Özellikle algoritmik ticaret, risk yönetimi ve piyasa analizi gibi alanlarda yapay zekâ uygulamalarının artışı, finansal analiz yöntemlerinin evrimini hızlandırmıştır.

Elman (1990), çalışmasında, zaman serisi verilerini analiz etmek için geri beslemeli sinir ağlarının (Recurrent Neural Networks - RNN) nasıl kullanılabileceğini açıklamaktadır. RNN'ler, geçmiş girişleri belleğinde tutarak, zamana bağlı bağımlılıkları öğrenebilir. Çalışmada, RNN'nin özellikle finansal piyasalardaki trendleri modellemek için uygun olduğu ve klasik regresyon yöntemlerine kıyasla daha doğru tahminler sağlayabildiği belirtilmektedir.

Hochreiter ve Schmidhuber (1997), çalışmalarında, geri beslemeli sinir ağlarının uzun vadeli bağımlılıkları öğrenme konusundaki zorluklarını ele almak için LSTM modelini tanıtmaktadır. LSTM, unutma kapıları (forget gates) sayesinde, eski bilgileri uzun süre boyunca saklayarak daha uzun vadeli bağımlılıkları öğrenebilir. Çalışma, LSTM'nin zaman serisi tahmini, NLP ve borsa tahminleri gibi alanlarda nasıl kullanıldığını açıklamaktadır.

Dietterich'in (2000) araştırmasında, topluluk (ensemble) yöntemlerinin makine öğrenmesindeki rolünü ve avantajlarını ele almaktadır. Bagging, boosting ve stacking gibi temel ensemble yöntemlerini karşılaştırarak, hangi durumlarda hangi yöntemin daha iyi performans gösterdiğini açıklamaktadır. Topluluk yöntemlerinin hata oranlarını nasıl düşürdüğünü ve genelleme hatasını nasıl azalttığını detaylı bir şekilde analiz eden bu

alıřma, makine ğrenmesinde topluluk modellerinin neden bu kadar gl olduğunu anlamak iin nemli sonulara ulařmıřtır.

Bhlmann ve Yu'nun (2002) alıřmalarında, bagging (bootstrap aggregating) ynteminin istatistiksel zelliklerini ve tahmin performansını detaylı bir řekilde analiz etmektedir. Bagging, zellikle yksek varyansa sahip modellerde hata oranlarını dřrmek iin kullanılan gl bir ensemble yntemidir. Makalede, bu yntemin bağımsız deęiřkenlere nasıl duyarlı olduęu ve kk veri kmelerinde nasıl etkili olabileceęi incelenmektedir. Yazarlar ayrıca, bagging'in ařırı ğrenmeyi nleme mekanizmasını ve veri setinin yapısına baęlı olarak performansını artırmak iin hangi ayarların yapılması gerektięini tartıřmaktadır.

Streichert vd. (2004) tarafından yrtlen bir dięer alıřmada, portfy optimizasyonu problemi, GA yaklařımı ile czlmřtr. Arařtırmacılar, klasik Markowitz Modeli'nin varsayımlarına alternatif olarak, yatırımcı tercihlerini daha esnek biimde modelleyebilecek bir ereve geliřtirmeyi amalamıřlardır. alıřmada, fitness fonksiyonu olarak risk-getiri oranı kullanılmıř ve portfy eřitlendirmesi ile birlikte yatırım risklerinin minimumda tutulması hedeflenmiřtir. GA'ların stokastik doęası sayesinde daha geniř bir czm uzayı taranmıř ve yerel optimumların tesine geilebilmiřtir. Bu alıřma, evrimsel algoritmaların finansal karar destek sistemleri aısından uygulanabilirlięini bařarılı řekilde ortaya koymuřtur.

Huang vd. (2005), alıřmalarında, borsa tahminlerinde SVM kullanımını ele almaktadır. SVM, yksek boyutlu veri kmeleri zerinde alıřan ve optimal karar sınırlarını belirleyerek tahmin doęruluęunu artıran bir makine ğrenimi algoritmasıdır. alıřma, gemiř fiyat hareketleri, iřlem hacmi ve piyasa gstergelerini ieren bir veri seti kullanarak SVM modelini eęitmiř ve sonuları geleneksel istatistiksel modellerle karřılařtırmıřtır. Bulgular, SVM'nin zellikle volatil piyasa kořullarında daha doęru tahminler yapabildięini ve geleneksel regresyon modellerine kıyasla daha yksek doęruluk saęladıęını gstermektedir.

Lin ve Liu (2008), portfy daęılımı problemine PSO algoritması ile yaklařarak, portfy riskini minimize ederken getiriye maksimize etmeyi hedeflemiřtir. alıřmada, yatırım evreni olarak Hong Kong hisse senetleri pazarı seilmiř ve belirli sayıda portfy alternatifi zerinde PSO algoritması ile ok sayıda iterasyon gerekleřtirilmiřtir. Sonular,

PSO algoritmasının hem işlem süresi hem de çözüm kalitesi açısından geleneksel optimizasyon tekniklerine göre daha verimli olduğunu göstermiştir. Ayrıca, algoritmanın parametre ayarlarının doğru yapılandırılması halinde yatırım tercihlerine göre özelleştirilebilir portföy yapılarının kolaylıkla elde edilebileceği belirtilmiştir.

Cura (2008) çalışmasında, portföy optimizasyonu için geleneksel yöntemler yerine PSO algoritmasının kullanılabilirliği incelenmiştir. PSO, finansal piyasaların karmaşıklığını modelleyerek, yatırımcılar için en uygun portföy dağılımını belirlemek amacıyla geliştirilmiş bir sezgisel optimizasyon yöntemidir. Araştırma, Nasdaq ve FTSE 100 endekslerinden alınan 15 yıllık hisse senedi verileri üzerinde gerçekleştirilmiştir. Çalışma kapsamında, PSO algoritmasının performansı, Markowitz Modeli ve GA ile kıyaslanmış ve PSO'nun %20 daha düşük hesaplama süresi ile daha iyi portföy tahsisleri sunduğu görülmüştür. Sonuçlara göre, PSO karmaşık yatırım portföyleri için düşük maliyetli ve etkili bir alternatif olarak değerlendirilmektedir.

Atsalakis ve Valavanis (2009), araştırmalarında, borsa tahminlerinde esnek hesaplama (soft computing) yöntemlerinin kullanımını kapsamlı bir şekilde ele almaktadır. Çalışmada YSA, bulanık mantık (fuzzy logic) ve GA gibi tekniklerin finansal tahminlerde nasıl uygulandığı açıklanmıştır. Yazarlar, bu yöntemlerin, finansal piyasalardaki doğrusal olmayan ilişkileri daha iyi modelleyerek geleneksel istatistiksel yöntemlerden daha yüksek tahmin doğruluğu sunduğunu göstermektedir. Ayrıca, makine öğrenimi teknikleriyle desteklenen hibrit modellerin borsa tahminlerinde daha başarılı olduğu belirtilmektedir.

Criminisi vd. (2011) çalışmalarında, RF'nin sadece sınıflandırma ve regresyon analizlerinde değil, aynı zamanda yoğunluk tahmini ve yarı gözetimli öğrenme gibi farklı alanlarda nasıl uygulanabileceğini ele almaktadır. Çalışma, özellikle medikal görüntüleme, 3D nesne tanıma ve NLP gibi alanlarda RF'nin nasıl kullanılabileceğini açıklamaktadır. Karar ormanlarının sunduğu esneklik ve hesaplama verimliliği, bu yöntemleri geniş bir uygulama yelpazesinde popüler hale getirmiştir.

Kara vd. (2011), Borsa İstanbul (BIST) verilerini kullanarak, hisse senedi yön tahmini (yükselir/düşer) üzerine ANN ve SVM algoritmalarını karşılaştırmalı olarak uygulamıştır. Çalışmada, BIST 100 endeksine ait 10 büyük ölçekli hisse senedine ilişkin teknik göstergeler kullanılarak günlük tahminler yapılmıştır. Eğitim ve test veri setleri üzerinde yapılan analizlerde ANN algoritması, %75'in üzerinde doğruluk oranı elde

ederken, SVM %71 civarında kalmıştır. Yazarlar, ANN'nin doğrusal olmayan ilişki yapılarını daha başarılı şekilde modelleyebildiğini ve finansal piyasaların öngörülmesinde avantaj sağladığını belirtmişlerdir.

Hamed vd. (2012) çalışmalarında, Çok Katmanlı Algılayıcı YSA aracılığıyla borsa sinyal tahmini için yenilikçi bir model önermektedir. Sinyal işleme aşamasında kör kaynak ayırma tekniği, tahmin doğruluğu ve genelleme eksikliği gibi sorunları aşmak üzere yapılandırılan temel MLP YSA'nın öğrenme sürecine entegre edilmiştir. Öğrenme algoritması olarak Kullback Leibler Ayrışması tercih edilmiştir; zira bu yöntem hızlı bir yakınsama sağlar ve öğrenme mekanizmasında genelleme yeteneği sunar. Önerilen modelin doğruluğu ve etkinliği, Wall Street piyasasından Microsoft borsa verileri ile Mısır borsasının çeşitli sektörlerinden elde edilen farklı veri kümeleri aracılığıyla doğrulanmıştır. Ayrıca, modelin genelleme sorununu ortaya koyabilmek amacıyla değişkenlerinin üzerinde bir duyarlılık analizi gerçekleştirilmiştir. Son olarak, istatistiksel anlamlılık, ANOVA testi kullanılarak incelenmiştir.

Cléménçon vd. (2013) çalışmalarında, RF sıralama problemlerinde nasıl optimize edilebileceğini incelemektedir. Çalışma, özellikle AUC (Area Under Curve) metriğinin optimizasyonuna odaklanarak, modelin sınıflandırma başarısını artırmayı hedeflemektedir. Finans, biyoinformatik ve tıp gibi alanlarda büyük öneme sahip olan sıralama modellerinin performansını artırmak için önerilen yöntemler, özellikle tahmin edici değişkenlerin etkileşimlerini daha iyi değerlendirebilen RF yapılarının kullanılmasını teşvik etmektedir.

Hao vd. (2013), çalışmalarında, SVM kullanılarak hisse senedi seçimi ve portföy optimizasyonu yapılmıştır. Model, önce hisse senetlerinin geçmiş fiyat hareketlerini analiz etmekte, ardından yüksek getiri potansiyeli olan hisseleri belirleyerek portföy oluşturma sürecini iyileştirmektedir. Sonuçlar, SVM ile optimize edilen portföylerin, geleneksel modellerden daha iyi risk-getiri oranlarına sahip olduğunu göstermektedir.

Patel vd. (2015) ise Hindistan borsasında işlem gören şirketlerin günlük verilerini kullanarak, dört farklı makine öğrenmesi algoritması —YSA, SVM, RF ve Naive Bayes (NB)— ile hisse fiyat tahmini yapmıştır. Çalışmada teknik analiz göstergeleri (RSI, MACD, OBV vb.) modele girdi olarak dahil edilmiş ve modellerin karşılaştırmalı performansları ROC eğrisi, doğruluk oranı ve F1 skoru gibi metriklerle ölçülmüştür. Sonuçlar, RF algoritmasının veri gürültüsüne karşı daha dirençli olduğu ve diğer algoritmalara göre daha

yüksek doğruluk sağladığını ortaya koymuştur. Yazarlar, RF algoritmasının finansal veri analitiğinde pratik ve etkili bir araç olduğunu vurgulamıştır.

Özçalıcı ve Ayriçay (2016) tarafından gerçekleştirilen araştırma, yapay zekâ tekniklerinin hisse senedi fiyat tahmininde nasıl kullanıldığını incelemektedir. Bu çalışma, yapay zekâ uygulamalarının nesnel bir bakış açısı sunduğu ve karar alma süreçlerini hızlandırdığı fikrine dayanmaktadır. Araştırmada, hisse senedi fiyat tahmini için kullanılan çeşitli yapay zekâ modelleri analiz edilmiş ve bu modellerin etkinlikleri karşılaştırılmıştır. Özellikle YSA ve SVM gibi yöntemler, geçmiş fiyat verileri ile eğitilmiş ve gelecekteki fiyat hareketlerinin tahmin edilmesi amacıyla test edilmiştir. Bu tür modeller, finansal veri setlerinde yüksek boyutluluğun ve karmaşıklığın üstesinden gelme yetenekleri ile öne çıkmaktadır.

Ban vd. (2016), çalışmalarında, makine öğrenmesi ile portföy optimizasyonu arasındaki etkileşimi incelemektedir. Çalışmada, geleneksel Markowitz modeli ile karşılaştırmalı olarak, SVM, XGBoost ve YSA gibi algoritmalar kullanılarak portföy tahsisi gerçekleştirilmiştir. Analizler, SveP 500 endeksine dahil olan 250 hisse senedi üzerinde yapılmış olup, makine öğrenmesi tabanlı optimizasyonun geleneksel modellere göre daha dinamik ve adaptif çözümler sunduğu görülmüştür. Sonuçlar, makine öğrenmesi yöntemlerinin piyasa volatilitesine duyarlı olarak daha yüksek getiri sağladığını göstermektedir.

Almanhdi ve Yang (2017), araştırmalarında, yatırım portföyleri için yeniden öğrenen pekiştirmeli öğrenme (Recurrent Reinforcement Learning - RRL) modelini kullanarak risk-getiri optimizasyonu yapmaktadır. Çalışmada, risk yönetiminin ve maksimum çekilme oranının (Maximum Drawdown) nasıl minimize edilebileceği analiz edilmiştir. Yazarlar, geleneksel yatırım modelleri yerine, piyasanın değişken yapısını sürekli öğrenen ve güncellenen bir sistem önermektedirler. Elde edilen sonuçlara göre, RRL tabanlı yatırım stratejileri, klasik portföy yönetim tekniklerine göre daha yüksek risk düzeltilmiş getiriler sağlamaktadır.

Almanhdi ve Yang (2017), çalışmalarında, geri beslemeli öğrenme (Reinforcement Learning, RL) teknikleriyle yatırım portföylerinin nasıl dinamik olarak optimize edilebileceğini araştırmaktadır. Model, hisse senedi piyasalarındaki oynaklıkları tespit ederek alım-satım stratejilerini gerçek zamanlı olarak ayarlayabilmektedir. Araştırma

sonuçlarına göre, RL tabanlı ticaret sistemleri, yatırımcıların piyasa değişikliklerine hızlı adapte olmalarını sağlamakta ve risk-getiri oranlarını iyileştirmektedir.

Chauhan vd. (2018) çalışmalarında, Bidirectional LSTM, ARIMA, CNN ve LSTM ile GRU'yu içeren beş makine öğrenimi modeli geliştirilmiştir. Bu modellerden bir tanesi LSTM ve GRU algoritmaları ile oluşturulmuştur. Sonrasında, hangi modelin en yüksek doğruluk oranına ulaştığını belirlemek amacıyla MAE puanları kullanılarak değerlendirme yapılmıştır. Ayrıca, bu makalede borsa fiyatlarındaki desen değişimlerini öngörmede insan zekasının kullanımını önermektedir. Temel amaç, 'Süper Tahmincileri' tanımlamak ve tahminlerini izleyerek hisse senedi fiyatlarındaki beklenmedik değişimleri tespit etmektir. Bu verilerin toplanması ve süper tahmincilerin belirlenmesi için, kullanıcıların belirli bir hisse senedi hakkında tahminlerini iletebileceği bir web sitesi geliştirilmiştir. Makine öğrenimi ile insan zekasının birleşik gücünden faydalanarak, tahmin doğruluğunu önemli ölçüde artırmak mümkündür.

Chauhan vd. (2018) çalışmasında, bir borsa endeksinin gelecekteki fiyatını modellemek ve tahmin etmek için yapay zekâ tekniklerini kullanmayı ve SVM ve Makine Öğrenimi kullanarak borsanın gelecekteki eğilimlerini tahmin etmek için etkili bir model elde etmeyi amaçlamaktadır. Yatırım risklerinden kaçınmak için borsanın gelecekteki davranışını keşfetmeye acil ihtiyaç vardır. Yapay zekâ teknikleri, finansal sistemin karmaşıklıklarını hesaba katabilir. Bu teknikler, finansal zaman serisi tahmininin araçları olarak kullanılır. Tartışmamızın özel odağı, piyasadaki tahmin davranışını optimize etmek için doğru tahmin sonucuyla borsanın gelecekteki fiyatını tahmin etmek için finansal modellemede kullanılan teknikleri geliştirmek ve genişletmektir.

Bai vd. (2018) araştırmalarında, zaman serisi modelleme için evrimsel sinir ağları (CNN) ve yinelemeli sinir ağlarının (RNN) performansını karşılaştırmıştır. Çalışma, bu iki modelin farklı veri setleri üzerinde nasıl çalıştığını analiz etmiş ve CNN'nin zaman serisi problemlerinde daha tutarlı sonuçlar verdiğini ortaya koymuştur. Özellikle sekans tabanlı tahminlerde, model seçiminin sonuçlar üzerindeki etkisi tartışılmıştır.

Jitendra vd. (2018), çalışmalarında, LSTM tabanlı tekrarlayan sinir ağları (RNN) kullanarak bulut veri merkezlerindeki iş yükü tahmini üzerine odaklanmaktadır. Araştırmada, LSTM modellerinin zaman serisi tahminlerindeki uzun süreli bağımlılıkları ele alarak daha doğru sonuçlar sunduğu gösterilmektedir. Geleneksel yöntemlerle

karşılaştırıldığında, LSTM'nin iş yükü tahminlerinde daha düşük hata oranına sahip olduğu ve veri merkezlerinin kaynak yönetimini optimize etmede önemli bir rol oynadığı belirtilmiştir.

Zhang vd. (2019) çalışmalarında, takviyeli öğrenme algoritmalarının, planlanan fiyat hareketlerine ve piyasa koşullarındaki değişikliklere göre portföy kombinasyonlarının sistematik olarak ayarlanmasını ve genel yatırım performansını artırabildiğini gösterdi.

Werawithayaset ve Trilanunt (2019), çalışmalarında, hisse senedi kapanış fiyatlarının tahmini için makine öğrenmesi tekniklerini kullanmaktadır. Özellikle Çok Katmanlı Algılayıcılar, LSTM ve RF gibi farklı algoritmaların performanslarını karşılaştırarak, hangi yöntemin daha iyi tahmin gücü sunduğunu analiz etmektedir. Çalışmada, Tayland Menkul Kıymetler Borsası'ndan (SET) alınan 5 yıllık hisse senedi fiyat verileri kullanılmıştır. Modelin girdileri olarak teknik göstergeler (RSI, MACD, Bollinger Bantları), işlem hacmi verileri ve temel finansal göstergeler (P/E oranı, piyasa değeri) kullanılmıştır. Sonuçlara göre, LSTM modelleri zaman serisi analizlerinde daha yüksek doğruluk sağlarken, RF kısa vadeli tahminlerde daha başarılı bulunmuştur. Çalışma, daha ileri düzeyde başarı sağlamak için derin öğrenme teknikleri ile ekonomik haber analizlerinin entegre edilebileceğini önermektedir.

Kim vd. (2019), araştırmalarında, derin öğrenmenin tıbbi görüntüleme alanındaki uygulamalarını incelemektedir. MRI ve CT görüntülerinin analizi için derin öğrenme modellerinin kullanımı ele alınmıştır. Çalışma, CNN tabanlı modellerin tümör tespiti gibi alanlarda daha başarılı olduğunu göstermekte ve model doğruluğunun artırılması için transfer öğrenme (transfer learning) tekniklerinin önemini vurgulamaktadır.

Adebiyi vd. (2020) çalışmalarında, yapay zekâ tabanlı bir model kullanarak geçmiş hisse senedi verilerine dayalı olarak portföy optimizasyonu yapmayı amaçlamaktadır. Araştırma, farklı sektörlerdeki hisse senetlerinin performansını değerlendirerek, yatırımcıların risklerini azaltmalarına ve getirilerini artırmalarına yardımcı olacak bir çerçeve sunmuştur. Makale, özellikle veri odaklı karar verme süreçlerinin önemini vurgulamaktadır.

Choi vd. (2020) çalışmalarında, makine öğrenimi, sinir ağları ve derin öğrenme tekniklerinin temellerini ve bu tekniklerin tıbbi görüntüleme, finansal analiz gibi alanlardaki

uygulamalarını tanıtmaktadır. Özellikle genelleme sorunları, eğitim veri setlerinin kalitesi ve aşırı öğrenme gibi konular üzerinde durulmuştur. Araştırma, yapay zekâ yöntemlerinin pratik kullanımlarını ele almıştır.

Ding ve Qin (2020) çalışmalarında LSTM tabanlı modellerin finansal zaman serisi analizindeki kullanımını incelemiş, LSTM'nin uzun vadeli bağımlılıkları öğrenme kapasitesine odaklanmıştır. Araştırma, hisse senedi fiyat tahmininde doğruluğu artırmak için çok kaynaklı veri setlerinin kullanımını önermektedir. Ayrıca, LSTM'nin karmaşık ağ yapılarının finansal uygulamalardaki avantajları tartışılmıştır.

Cao vd. (2020) çalışmalarında, geleneksel finansal modellerin yerine, piyasa verilerini doğrudan analiz eden ve optimal varlık tahsisi yapabilen bir sinir ağı mimarisi önermektedir. Derin öğrenme teknikleri sayesinde, portföy yönetimi daha esnek ve veri odaklı bir hale gelmektedir. Modelin, geleneksel yöntemlere göre daha yüksek getiri sağladığı gösterilmiştir.

Wuyu vd. (2020), çalışmasında, uzun vadeli finansal tahminleme için derin öğrenme modellerinin etkinliğini araştırmaktadır. Çalışmada, LSTM, GRU ve Transformer tabanlı modeller kullanılarak finansal zaman serilerinde tahminler yapılmıştır. 50 yıllık SveP 500 endeks verileri kullanılarak, büyük ölçekli finansal zaman serileri ile model performansı ölçülmüştür. Sonuçlar, LSTM modellerinin kısa vadeli tahminlerde daha başarılı olduğunu, ancak Transformer tabanlı modellerin uzun vadeli trend analizlerinde üstünlük sağladığını göstermektedir. Çalışma, uzun vadeli piyasa tahminlerinde yapay zekâ ve veri madenciliği tekniklerinin bir arada kullanılması gerektiğini vurgulamaktadır.

Shen ve Shafiq (2020), çalışmalarında, hisse senedi piyasalarındaki kısa vadeli fiyat eğilimlerini tahmin etmek için derin öğrenme tabanlı bir modelin geliştirilmesini ele almaktadır. Yazarlar, LSTM, CNN ve geleneksel regresyon tekniklerini kıyaslayarak, hangisinin en iyi tahmin performansını sunduğunu analiz etmiştir. Sonuçlar, LSTM tabanlı modellerin, zaman serisi verilerini işlemede en yüksek doğruluk oranına (%85'in üzerinde) sahip olduğunu göstermektedir. Çalışmada, NASDAQ ve SveP 500 borsalarından elde edilen günlük fiyat verileri kullanılmış ve kapsamlı veri ön işleme teknikleri uygulanarak modelin başarısı artırılmıştır. Özellikle zaman serisi içindeki trendleri ve kısa vadeli dalgalanmaları anlamada LSTM modellerinin CNN ve geleneksel regresyon yöntemlerinden

daha üstün olduğu belirlenmiştir. Bu araştırma, kısa vadeli yatırım stratejileri geliştirmek isteyen yatırımcılar için önemli bir kaynak niteliğindedir.

Abbasimehr vd. (2020), çalışmalarında, LSTM sinir ağlarının talep tahmini için nasıl optimize edilebileceğini araştırmaktadır. Geleneksel zaman serisi tahmin modelleri, geçmiş verilere dayalı öngörüler yaparken uzun vadeli bağıntıları yakalamakta zorlanmaktadır. Bu nedenle, LSTM'nin hafıza mekanizmasını kullanarak talep değişimlerini daha doğru bir şekilde modelleyebileceği düşünülmektedir. Araştırmacılar, bir mobilya üretim firmasının satış verileri üzerinde LSTM modelini test ederek, müşteri talebinin gelecekteki değişimlerini tahmin etmek için en uygun hiperparametreleri belirlemişlerdir. Çalışmada, grid search yöntemi kullanılarak optimum öğrenme oranı, aktivasyon fonksiyonu ve gizli katman sayısı belirlenmiştir. Sonuçlar, önerilen LSTM modelinin geleneksel regresyon ve ARIMA modellerine kıyasla daha düşük hata oranları sağladığını göstermektedir. Bu model, hisse senedi tahmininden enerji talep yönetimine kadar birçok farklı uygulama alanına sahip olabilir.

Ma vd (2021), çalışmalarında, yatırım getirilerinin tahmin edilmesi ve portföy optimizasyonunda derin öğrenme yöntemlerinin uygulanmasını araştırmaktadır. CNN, LSTM ve DQN tabanlı modeller kullanılarak, hisse senedi fiyat hareketleri analiz edilmiştir. Veri seti olarak, MSCI Dünya Endeksi'ne ait 20 yıllık piyasa verileri incelenmiştir. CNN modellerinin piyasa trendlerini yakalamada daha başarılı olduğu görülürken, LSTM modellerinin dinamik piyasa koşullarına daha iyi adapte olduğu ortaya konmuştur. Çalışma, makine öğrenmesi ve derin öğrenme tekniklerinin birlikte kullanılmasıyla portföy yönetiminde daha yüksek risk-getiri dengesi elde edilebileceğini göstermektedir.

Finansal piyasalarda yapay zekâ ve makine öğrenimi tekniklerinin uygulamaları giderek daha fazla dikkat çekmektedir. Yapay zekâ sistemleri, veri madenciliği ve büyük veri analizi gibi alanların birleşimiyle birlikte, piyasa trendlerini tahmin etme, yatırım stratejileri geliştirme ve risk analizini optimize etme konularında önemli bir potansiyele sahiptir. Özgür (2021) çalışmasında, bu algoritmaların ve yapay zekâ yöntemlerinin performans üzerindeki etkilerini vurgulamaktadır. Özgür (2021)'e göre, yapay zekâ teknikleri, geçmiş piyasa verilerinin analizi yoluyla, piyasa hareketlerinin daha iyi anlaşılmasını sağlamakta ve böylece daha doğru tahminler ortaya koyabilmektedir. Bu

bağlamda, yapay zekâ uygulamaları, klasik istatistiksel yöntemlere göre daha esnek ve dinamik bir yapıya sahip olmasıyla dikkat çekmektedir.

Gao vd. (2021), optimize edilmiş LSTM ve GRU modellerini kullanarak hisse senedi tahmini yapan bu çalışma, bu modellerin performansını karşılaştırmıştır. Çalışma, zaman serisi verilerinde parametre optimizasyonunun model doğruluğunu artırmada önemli bir faktör olduğunu göstermiştir. Özellikle finansal piyasalarda kısa vadeli tahminler için bu yöntemlerin uygulanabilirliği incelenmiştir.

Zhang ve Mo (2021), araştırmalarında, robotik sistemlerde karar alma süreçlerini geliştirmek için pekiştirmeli öğrenme (Reinforcement Learning - RL) algoritmalarının kullanım şeklini incelemektedir. Q-learning, derin pekiştirmeli öğrenme (Deep Reinforcement Learning) ve aktör-eleştirmen (Actor-Critic) algoritmalarının etkinliği karşılaştırılmıştır. Çalışma, RL'nin robotların dinamik ortamlarda karar alma yeteneğini artırdığını ve özellikle otonom sistemlerde büyük bir potansiyele sahip olduğunu ortaya koymaktadır.

Kumar vd. (2021), çalışmalarında, kırsal finans sektöründe dijital kredi puanlaması için makine öğrenimi tekniklerinin kullanımını incelemektedir. Araştırmada, kredi puanlaması için geleneksel yöntemlerin yerine makine öğrenimi tabanlı modellerin nasıl uygulanabileceği tartışılmaktadır. SVM, RF ve LSTM gibi algoritmaların performansları karşılaştırılmış ve makine öğrenimi tabanlı sistemlerin daha hızlı ve doğru tahminler sunduğu sonucuna varılmıştır.

Johnson (2021), araştırmasında, finansal modellemenin olasılık teorisindeki gelişmelerle nasıl etkileşimde bulunduğunu incelemektedir. Çalışma, finansal karar alma süreçlerinde kullanılan istatistiksel modellerin evrimini ele alarak, özellikle risk analizi ve stokastik süreçler bağlamında makine öğrenimi tekniklerinin önemini vurgulamaktadır. Finansal piyasalarda tahmin doğruluğunu artırmak için olasılık tabanlı modelleme yaklaşımlarının nasıl geliştirilebileceği konusunda öneriler sunulmaktadır.

Yıldız (2022), finans alanında yapay zekâ teknolojisinin kullanımını kapsamlı bir şekilde ele almakta ve bu teknolojilerin türleri ile uygulama alanlarını derinlemesine incelemektedir. Yıldız (2022) çalışmasında, yapay zekâ sistemlerinin genellikle veri madenciliği, makine öğrenimi ve derin öğrenme gibi teknikler kullanarak büyük veri analizi

yaptıklarını belirtmektedir. Bu teknikler, büyük miktardaki finansal verilerin hızlı ve etkili bir şekilde işlenmesine olanak tanırken, piyasalardaki karmaşık dinamikleri anlamaya yardımcı olmaktadır. Özellikle makine öğrenimi algoritmaları, geçmiş verilerden öğrenerek gelecekteki fiyat hareketlerini tahmin etme kapasitesine sahip olmaları nedeniyle dikkat çekmektedir. Örneğin, finansal tahmin süreçlerinde sıklıkla kullanılan regresyon analizi veya zaman serisi analizi gibi klasik istatistiksel yöntemler, yapay zekânın sağladığı derin analiz yetenekleri ile desteklenmekte ve performansları artırılmaktadır.

Ayrıca, Yıldız (2022) çalışmasında, bu tekniklerin finansal piyasalar üzerindeki avantajlarını vurgulamaktadır. Yapay zekâ uygulamalarının, stratejik karar verme süreçlerini hızlandırarak, analistlerin ve yatırımcıların daha doğru tahminler yapmalarına olanak sağladığını ifade etmektedir. Yıldız'a göre, yapay zekâ tabanlı tahmin sistemleri, yatırımcılara hızlı veri analizi ve karar alma süreçlerinde daha fazla özgürlük sunarken, aynı zamanda insan hatasını minimize etmeyi de başarmaktadır. Bunun yanı sıra, bu sistemlerin sürekli öğrenme süreçleri, dinamik piyasa koşullarına uyum sağlamalarını mümkün kılmakta ve mevcut stratejileri optimize edebilmektedir.

Padhi vd. (2022) çalışmalarını iki aşamalı olarak ele alınmıştır. İlk olarak, portföy oluşturmak amacıyla ortalama-varyans yöntemi kullanılmakta ve bu sayede yatırım riskinin minimize edilmesi hedeflenmektedir. İkinci aşamada, gelecekteki hisse senedi fiyat hareketlerini tahmin etmek için "algılayıcı" ve "pasif-agresif algoritma" kombinasyonunu içeren makine öğrenimi tekniği önerilmektedir. Performans değerlendirmesi dört farklı coğrafi faktör göz önünde bulundurularak, 20 sağlık sektörü endeksinin gerçek dünya veri setlerinde yapılmıştır. Son olarak, yöntemin sonuçlarının önceki çalışmalarla geleneksel çözümlerle elde edilen sonuçlarla sayısal bir karşılaştırması yapılmaktadır. Elde edilen bulgular, portföy seçimi ile öğrenme tabanlı topluluk stratejilerinin karşılaştırmalı olarak etkili olduğunu göstermektedir.

Saini (2024), çalışmasında, finansal piyasa tahminlerinde SVM algoritmasının nasıl uygulanabileceğini detaylı bir şekilde ele almaktadır. Çalışmada, doğrusal ve RBF çekirdekli SVM modelleri kullanılarak hisse senedi fiyatları tahmin edilmiştir. Veri seti olarak, Nasdaq endeksine ait 15 yıllık fiyat verileri analiz edilmiştir. Sonuçlar, SVM modellerinin finansal kriz dönemlerinde bile yüksek doğruluk oranları sunduğunu

göstermektedir. Yazar, SVM'nin özellikle volatil piyasalarda tahminleme doğruluğunu artırmada etkili bir yöntem olduğunu belirtmektedir.

Bu kapsamda, yapay zekâ tabanlı tahminleme ve portföy optimizasyonu süreçlerine ilişkin yürütülen önceki çalışmaların yöntemsel çeşitliliğini ve uygulama alanlarını daha açık şekilde ortaya koyabilmek amacıyla, literatür taraması Tablo 2.1’de verilmiştir. Bu tablo, araştırmacılar ve çalışmanın yapıldığı yıl, araştırmalarda tercih edilen algoritmalar ve çalışmanın konusu sistematik olarak özetlemektedir..

Tablo 2.1. Literatür taraması

Araştırma (Yıl)	Kullandığı Yöntem	Konu
Elman (1990)	RNN	Finansal zaman serisi tahmini
Hochreiter ve Schmidhuber (1997)	LSTM	Uzun dönem bağımlılıkların öğrenilmesi
Dietterich (2000)	Topluluk (Ensemble) yöntemleri	Bagging, boosting, stacking üzerine analiz
Bühlmann ve Yu (2002)	Bagging	İstatistiksel özellikler ve tahmin performansı
Streichert vd. (2004)	Genetik Algoritma	Portföy optimizasyonu
Huang vd. (2005)	SVM	Borsa tahminleri
Bishop (2006)	Makine öğrenimi (Denetimli, Denetimsiz, Pekiştirmeli)	Makine öğrenimi temelleri ve uygulamaları
Lin ve Liu (2008)	PSO	Portföy dağılımı optimizasyonu
Cura (2008)	PSO	Portföy optimizasyonu (Nasdaq ve FTSE)
Atsalakis ve Valavanis (2009)	YSA, Fuzzy, GA	Borsa tahmini ve hibrit modeller
Kara vd.. (2011)	ANN, SVM	BIST hisse senedi tahmini
Criminisi vd. (2011)	RF	Medikal görüntüleme ve NLP
Hamed vd. (2012)	Çok Katmanlı Algılayıcı, Kör kaynak ayırma	Borsa sinyal tahmini
Cléménçon vd. (2013)	RF (AUC optimizasyonu)	Sıralama problemleri
Hao vd. (2013)	SVM	Hisse senedi seçimi ve portföy optimizasyonu
Kılıç (2014)	Makine öğrenimi (Sınıflandırma, Regresyon, Kümeleme)	Büyük veri analizi ve yapay zeka uygulamaları
Patel vd.. (2015)	YSA, SVM, RF, NB	Hisse fiyat tahmini
Özçalıcı ve Ayriçay (2016)	YSA, SVM	Hisse senedi fiyat tahmini
Ban vd. (2016)	SVM, XGBoost, YSA	Portföy optimizasyonu
Almanhdi ve Yang (2017)	RRL	Yatırım portföyü optimizasyonu
Esteva vd. (2017)	Derin öğrenme	Cilt kanseri teşhisinde yapay zeka kullanımı
Bai vd. (2018)	CNN, RNN	Zaman serisi modelleme
Jitendra vd. (2018)	LSTM	Bulut veri merkezi iş yükü tahmini

Tablo 2.2. Literatür taraması (devamı)

Chauhan vd. (2018)	LSTM, GRU, CNN, ARIMA	Borsa fiyat tahmini ve insan zekası entegrasyonu
Zhang vd. (2019)	RL	Portföy ayarlaması ve yatırım performansı
Werawithayaset ve Trilant (2019)	ANN, LSTM, RF	Hisse senedi kapanış fiyat tahmini
Kim vd. (2019)	CNN, Transfer Learning	Tıbbi görüntüleme (MRI, CT)
Panc vd. (2020)	Derin öğrenme	Sağlık sistemlerinde yapay zeka uygulamaları
Abbasimehr vd. (2020)	LSTM	Talep tahmini ve hiperparametre optimizasyonu
Shen ve Shafiq (2020)	LSTM, CNN, Regresyon	Kısa vadeli hisse fiyat tahmini
Wuyu vd. (2020)	LSTM, GRU, Transformer	Uzun vadeli finansal tahminleme
Ekrem vd. (2020)	RF, PSO	Kalp hastalığı sınıflandırma
Ding ve Qin (2020)	LSTM	Finansal zaman serisi analizi
Choi vd. (2020)	ML, NN, DL	Tıbbi görüntüleme, finansal analiz
Adebiyi vd. (2020)	Yapay zeka tabanlı model	Portföy optimizasyonu
Cao vd. (2020)	Derin öğrenme	Portföy yönetimi
Ma vd (2021)	CNN, LSTM, DQN	Portföy optimizasyonu
Özgür (2021)	Yapay zeka teknikleri	Finansal piyasada tahmin performansı
Gao vd. (2021)	Optimize LSTM, GRU	Hisse senedi tahmini
Zhang ve Mo (2021)	RL (Q-learning, DRL, Actor-Critic)	Robotik sistemlerde karar alma
Kumar vd. (2021)	SVM, RF, LSTM	Dijital kredi puanlaması
Johnson (2021)	ML, Stokastik süreçler	Finansal modelleme ve risk analizi
Yıldız (2022)	Makine öğrenimi, DL	Finansal piyasalarda AI uygulamaları
Padhi vd. (2022)	Ortalama-varyans + ML	Portföy seçimi ve tahminleme
Jurafsky ve Martin (2023)	NLP	Dil işleme ve chatbot uygulamaları
Kurtboğan (2023)	Derin öğrenme	Görüntü, ses ve NLP uygulamaları
Saini (2024)	SVM	Finansal piyasa tahmini

3. BÖLÜM

3. HİSSE SENEDİNİN FİYAT TAHMİNİ VE PORTFÖY OPTİMİZASYONU

Hisse senedi kapanış fiyatı tahmini ve portföy optimizasyonu ile ilgili daha önceki çalışmalar incelendiğinde, hisse senedinin kapanış fiyatı tahmininde ve optimizasyon kısmında yapay zekâ tekniklerinin ayrı ayrı kullanıldığı gözlemlenmiştir. Bu çalışma, yapay zekâ yöntemlerinin hem tahmin hem de optimizasyon kısmında ele alındığından hibrit bir yapı olarak değerlendirilip, yatırımcıya rehber niteliği göstererek literatüre katkı sağlayacağı düşünülmektedir. Bu bölümde, araştırmanın amacı ve önemi, BIST 30, araştırmanın yöntemi ve kapsamı, araştırmanın tahmin ve optimizasyon kısmında uygulanan adımlar ve elde edilen bulgulara yer verilmiştir.

3.1. Araştırmanın Amacı ve Önemi

Bu çalışmanın temel amacı, yapay zekâ ve makine öğrenmesi yöntemlerini kullanarak finansal piyasalar için daha doğru ve etkili yatırım stratejileri geliştirmektir. Günümüzde küreselleşen ekonomi ve dijital dönüşüm süreçleriyle birlikte finans piyasaları giderek daha karmaşık hale gelmekte, yatırımcılar ise bilinçli kararlar alabilmek için ileri düzey analiz tekniklerine ihtiyaç duymaktadır. Bu bağlamda, çalışma kapsamında Borsa İstanbul BIST 30 endeksinde işlem gören hisse senetlerinin geçmiş verileri analiz edilerek fiyat tahmini yapılmış olup, portföy optimizasyonu gerçekleştirilmiştir. Hisse senedi fiyat SVM regresyon problemlerine uyarlanmış hali olan SVM yöntemi, RF algoritması, LSTM ve XGBoost kullanılmış olup, en iyi tahmin sonucunu veren RF algoritması ile tahminleme sonuçları kullanılarak, yatırımcıların en uygun portföyü oluşturabilmesi için TS algoritması uygulanmıştır. Tahmin ve optimizasyon modelinin başarısı çeşitli performans metrikleri kullanılarak değerlendirilmiş, böylece finansal tahminlerin doğruluğu artırılarak yatırım kararlarının daha veri odaklı hale getirilmesi sağlanmıştır. Çalışma, geleneksel finans modellerinin eksikliklerini gidermeyi amaçlarken, aynı zamanda yapay zekâ tabanlı analizlerin finansal karar alma süreçlerinde nasıl etkin bir şekilde kullanılabileceğini ortaya koymuştur. Elde edilen sonuçlar, yatırımcılar için risk yönetimi ve getiri optimizasyonu açısından önemli bir rehber niteliği taşıyacak, aynı zamanda finans literatürüne katkı sağlayacağı düşünülmektedir. Yapay zekâ destekli finansal analizlerin yaygınlaşmasıyla

birlikte, daha öngörülebilir ve sürdürülebilir bir yatırım stratejisi geliştirmek mümkün hale gelmiştir.

3.2. Araştırmanın Yöntemi ve Kapsamı

Araştırma kapsamında Borsa İstanbul'da işlem gören hisse senetlerinin bilgileri kullanılmıştır.

BIST 30, Borsa İstanbul'da işlem gören en yüksek piyasa değerine ve işlem hacmine sahip 30 şirketin hisse senetlerinden oluşan bir endekstir. BIST 30 endeksi yılda dört kez yeniden belirlenir. Dolayısıyla belirlenen endeks üç ay boyunca işlem görür. Üç ay sonunda yeni endeks belirlenir. Dönemler Ocak-Mart, Nisan-Haziran, Temmuz-Eylül ve Ekim-Aralık dönemleridir. Dolayısıyla Ocak, Nisan, Temmuz ve Eylül ayı başında endeks yeniden hesaplanmaktadır.

BIST30 listesi hazırlanırken öncelikle tüm pay senetleri dolaşımdaki değerleri itibarıyla toplam büyüklükleri açısından sıraya konur. Sonrasında ise hacme ilişkin bir başka sıralama yapılır. Değerleme dönemine ilişkin toplam işlem hacimlerine göre yeniden sıralanır. Böylece iki ana liste oluşur. Bu iki listede de ilk sıralarda yer alan hisse senetleri BIST30 listesine alınır. Bu şekilde liste sıralamasındaki genel yerlerine göre sırayla hisseler BIST30 endeksine dahil edilir ve liste bu şekilde tamamlanır.

BIST 30 endeksinde yer alan şirketlerin arasında bankacılık, enerji, telekomünikasyon, otomotiv, perakende, inşaat gibi farklı sektörlerden örnekler vardır. BIST30 endeksi diğer yandan hemen her sektörü kapsadığı için ekonominin geneliyle ilgili bir görünüme yardım eder. Mesela bankacılık sektöründe büyük bir sıkıntı olduğunda diğer sektördeki şirketlerin endeks içindeki değeri bu sapmanın gücünü sınırlı tutacaktır. Bu nedenle BIST30 ekonominin genel görünümüne ilişkin takip edilmeye değer bir menkul kıymettir.

Araştırma iki bölümden oluşmaktadır:

Araştırmanın birinci bölümünde, yapay zekâ yöntemlerinden RF, LSTM, XGBoostve SVR yöntemleri ile BIST 30’da yer alan hisse senetlerinin kapanış fiyatlarının tahmini yapılacaktır. Tahmin bölümünde kullanılmak üzere, yahoofinance.com dan BIST 30 endeksine ait 30 hisse senedinin geçmiş 2 yıllık açılış, kapanış, en düşük, en yüksek ve hacim değerlerini içeren veriler elde edilmiştir. Bu veriler kullanılarak hangi yöntemin daha doğru tahminleme yaptığı elde edilmiştir.

Araştırmanın ikinci bölümünde ise portföy optimizasyonuna yer verilmiştir. Tahmin doğruluğu en yüksek olan RF algoritması ile elde edilen değerleri kullanarak, MPT çerçevesinde performans göstergeleri de dikkate alınarak, TS yöntemi ile optimum portföy belirlenmiştir. Optimum portföy içerisinde her hisse senedinin ağırlığına yer verilmiştir.

3.3. Tahmin Bölümünde Yapılanlar

Bu bölümde verilerin hazırlığı ve tahminde kullanılan araçlar düzenlenmiştir.

3.3.1. Veri Hazırlığı

Araştırmada tahmin bölümünde kullanacağımız 2025 yılının ikinci üç aylık döneminde BIST 30 endeksindeki hisse senetleri sembolleri ve sektörel dağılımı ile birlikte Tablo 3.1’de verilmiştir.

Tablo 3.1. 2025 yılının ikinci üç aylık döneminde BIST 30 endeksindeki hisse senetleri

HİSSE KODU	Bileşen Adı	Sektör
AEFES. IS	Anadolu Efes	Gıda İçecek Ve Tütün
AKBNK. IS	Akbank	Bankalar
ASELS. IS	Aselsan	Savunma
ASTOR. IS	Astor Enerji A.Ş.	Endüstriyel Ürün Ve Hizmetler
BIMAS. IS	Bim Mağazaları	Perakende Ticaret
CIMSA. IS	Çimento Sanayi Ve Ticaret A.Ş.	Hammadde
EKGYO. IS	Emlak Konut Gayrimenkul Yatırım Ortaklığı	Gayrimenkul Yatırım Ortaklıkları
ENKAI. IS	Enka İnşaat Ve Sanayi	İnşaat Ve Bayındırlık İşleri
EREGL. IS	Ereğli Demir Ve Çelik Fabrikaları	Ana Metal Sanayi
FROTO. IS	Ford Otomotiv Sanayi	Metal Eşya Makine Elektrikli Cihazlar Ve Ulaşım Araçları
GARAN. IS	Garanti Bankası	Bankalar
HEKTS. IS	Hektaş Ticaret	Kimya İlaç Petrol Lastik Ve Plastik Ürünler
ISCTR. IS	Türkiye İş Bankası	Bankalar
KCHOL. IS	Koç Holding	Holdingler Ve Yatırım Şirketleri
KOZAL. IS	Koza Altın İşletmeleri	Metal Cevheri Madenciligi
KRDMD. IS	Kardemir Karabük Demir Çelik Sanayi Ve Ticaret	Ana Metal Sanayi
MGROS.IS	Migros Ticaret	Perakende Ticaret
PETKM. IS	Petkim Petrokimya Holding	Kimya İlaç Petrol Lastik Ve Plastik Ürünler
PGSUS. IS	Pegasus Hava Taşımacılığı	Ulaştırma Ve Depolama
SAHOL. IS	Sabancı Holding	Sabancı Holding
SASA. IS	Sasa Polyester Sanayi	Kimya İlaç Petrol Lastik Ve Plastik Ürünler
SISE. IS	Türkiye Şişe Ve Cam Fabrikaları	Holdingler Ve Yatırım Şirketleri
TAVHL. IS	Tav Havalimanları Holding A.Ş.	Ulaştırma
TCELL. IS	Turkcell	Telekomünikasyon
THYAO. IS	Türk Hava Yolları	Ulaştırma Ve Depolama
TOASO. IS	Tofaş Türk Otomobil Fabrikası	Metal Eşya Makine Elektrikli Cihazlar Ve Ulaşım Araçları
TTKOM. IS	Türk Telekomünikasyon	Telokomünikasyon
TUPRS. IS	Türkiye Petrol Rafinerileri	Kimya İlaç Petrol Lastik Ve Plastik Ürünler
ULKER. IS	Ülker Bisküvi Sanayi	Gıda İçecek Tütün
YKBNK. IS	Yapı Ve Kredi Bankası	Bankalar

Çalışmada kullanılmak üzere 2025 yılının ikinci üç aylık dönemindeki BIST 30 endeksinde yer alan hisse senetlerinin 02/01/2023 – 01/04/2025 tarihleri arasındaki açılış

değeri, kapanış değeri, en düşük değer, en yüksek değer ve işlem hacimleri <https://finance.yahoo.com/> internet sitesinden kodlar yardımıyla elde edilmiştir.

3.3.2. Tahminlemede Kullanılacak Araçların Hazırlığı

Tahminleme kısmında kullanacak olduğumuz hisse senetlerinin açılış fiyatları, kapanış fiyatları, en düşük değerleri, en yüksek değerleri ve işlem hacimleri ile oluşturulan veri seti denilen kısım, eğitim ve test bölümü olmak üzere iki bölüme ayrılmaktadır. Hisse senetlerinin tahmini ile ilgili yapılacak her türlü işlem, Google şirketi tarafından ücretsiz olarak kullanıcılara sunulan Google Colaboratory bulut ortamı kullanılacaktır. Veri setini yüklemekten önce verilerin sayısal ve görsel analiz başta olmak üzere bir çok analize imkan veren araçların ortama yüklenmesi gerekir. Verilerin işlenmesine de yardımcı olan bu araçlar kütüphane olarak bilinmektedir. Çalışmamızda Pandas, NumPy, Sklearn, Matplotlib, TensorFlow başta olmak üzere çeşitli kütüphaneler kullanılmıştır.

Pandas, Python programlama dili için geliştirilen, veri analizi ve veri işleme işlemlerinde kullanılan yüksek performanslı ve kullanımı kolay bir kütüphanedir. Kütüphane, özellikle etiketlenmiş verilerle çalışmayı kolaylaştıran DataFrame ve Series veri yapılarını sunar. Yapılandırılmış verilerin yüklenmesi, temizlenmesi, dönüştürülmesi, filtrelenmesi ve özetlenmesi gibi işlemleri verimli bir şekilde gerçekleştirmeyi sağlar (McKinney, 2010).

NumPy (Numerical Python), sayısal hesaplamalar için temel bir Python kütüphanesidir. Özellikle çok boyutlu diziler (ndarray), lineer cebir işlemleri, Fourier dönüşümleri ve rastgele sayı üretimi gibi işlemler için optimize edilmiştir (Harris vd., 2020). NumPy, Python'da bilimsel hesaplamaların temel taşı olup, diğer birçok kütüphane (örneğin TensorFlow ve Pandas) içsel olarak NumPy dizilerini kullanmaktadır.

Scikit-learn, Python için geliştirilmiş açık kaynaklı bir makine öğrenimi kütüphanesidir. Sınıflandırma, regresyon, kümeleme, boyut indirgeme ve model doğrulama gibi işlemler için geniş bir algoritma yelpazesi sunar (Pedregosa vd., 2011). Kütüphane, NumPy, SciPy ve matplotlib gibi kütüphanelerle entegre şekilde çalışır.

Matplotlib, Python dili için geliştirilen kapsamlı bir veri görselleştirme kütüphanesidir. Çizgi grafikleri, çubuk grafikler, histogramlar, scatter plot'lar gibi 2D

görsellerin yanı sıra, sınırlı da olsa 3D grafikler de oluşturulmasına olanak tanır (Hunter, 2007). Matplotlib, özellikle matlab benzeri arayüzüyle araştırmacılara bilimsel grafik üretme konusunda esneklik sağlar.

TensorFlow, Google tarafından geliştirilen ve açık kaynak kodlu olarak sunulan güçlü bir derin öğrenme kütüphanesidir. Temel olarak düğümler (nodes) ve kenarlardan (edges) oluşan veri akış grafiklerine (dataflow graphs) dayanır ve çok katmanlı yapay sinir ağlarının eğitilmesini sağlar. TensorFlow'un bu grafik tabanlı mimarisi, özellikle büyük ölçekli veri işleme süreçlerinde esneklik ve ölçeklenebilirlik sunar (Zaharia vd., 2018). Ayrıca hem CPU hem de GPU mimarilerinde çalışabilme yeteneği sayesinde farklı donanım ortamlarına kolayca entegre edilebilir. Üretim ortamlarında da yaygın olarak kullanılan TensorFlow, özellikle Google Brain ekibinin desteğiyle optimize edilmiş yüksek performanslı bir çerçeve olarak dikkat çeker (Yu vd., 2019; Shankar vd., 2018).

3.3.3. Modelin Doğruluğunu Değerlendiren Performans Göstergeleri

Yapay zekâ yöntemleri kullanılarak yapılan tahminlemede modelin performansını değerlendiren birçok gösterge bulunmaktadır. Literatürdeki diğer çalışmalar incelendiğinde, hisse senedinin tahmininde en çok Ortalama Mutlak Yüzde Hatası (MAPE) ve Kök Ortalama Kare Hatası (RMSE) doğruluğunu performans göstergesi olarak kullanıldığı sonucuna varılmıştır.

Çalışmada kullanılan RF'nin doğruluğunu tespit etmek için iki performans göstergesi hesaplanmıştır.

3.3.3.1. Ortalama Mutlak Yüzde Hatası

MAPE, tahmin edilen değerlerin ne derece doğru olduğunu yüzdesel bir ölçüyle ifade eden ve sıklıkla zaman serisi tahminleri ile finansal modelleme çalışmalarında kullanılan bir hata metriğidir. MAPE, her bir tahminin gerçek değerden yüzde kaç sapma gösterdiğini belirleyerek, bu sapmaların ortalamasını alır. Model performansının kullanıcı dostu bir özetini sunar çünkü sonuçlar doğrudan yüzde olarak ifade edilir. Ancak, sıfıra yakın gerçek değerler karşısında sonsuz veya anlamlı olmayan sonuçlar doğurabileceği için, düşük değerli serilerde dikkatle kullanılmalıdır. Ayrıca, pozitif ve negatif sapmaları eşit

şekilde ele alması model davranışının simetrik olduğu varsayımına dayanır. MAPE değeri, aşağıdaki formül yardımıyla bulunur (Armstrong, 1985; Hyndman ve Koehler, 2006).

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \quad 3.1$$

Burada, n veri sayısını, y_i gerçek değeri ve $\hat{y}_i$ ise tahmin edilen değeri göstermektedir. MAPE sonuçları değerlendirilirken Tablo 3.2'deki bilgilerden yararlanılabilir (Anggrainingsih, 2015).

Tablo 3.2. MAPE performans tablosu

MAPE değeri	Tahmin doğruluğu
< %10	Çok iyi
%10 – %20	İyi
%20 – %50	Kabul edilebilir
> %50	Kötü veya hatalı

3.3.3.2. RMSE (Kök Ortalama Kare Hatası)

RMSE, bir modelin tahmin hatalarını kareleyerek ortalamasını alır ve sonuçta karekökünü alarak hata büyüklüğünü ölçer. Hataların karesi alındığı için büyük hatalara karşı daha duyarlıdır. Bu özelliği, RMSE'yi risk analizlerinde ve volatil finansal piyasalarda kritik bir ölçüt haline getirir. RMSE'nin değeri, veri setindeki birimlerle ifade edilir ve bu da onu mutlak hata metriği olarak konumlandırır. Uygulamalarda, RMSE'nin düşük olması modelin yüksek doğruluğa sahip olduğunu, dolayısıyla daha az varyans içerdiğini gösterir. Bu hata ölçütünü belirleyebilmek için aşağıdaki formül kullanılır (Chai ve Draxler, 2014).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad 3.2$$

Burada, n veri sayısını, y_i gerçek değeri ve $\hat{y}_i$ ise tahmin edilen değeri göstermektedir. RMSE hata ölçütü yorumlanırken, bulunan hata değerinin küçüklüğü tahminin başarılı olduğunu gösterir (Sebastian ve Tantia, 2018).

3.3.3.3. Ortalama Mutlak Hata

Ortalama Mutlak Hata (MAE), bir modelin tahmin ettiği değerlerle gerçek değerler arasındaki farkların mutlak değerlerinin ortalamasını alarak hata miktarını ölçen doğrudan ve kolay yorumlanabilir bir metriktir. MAPE'nin aksine, orijinal birimlerle ifade edilir ve bu nedenle bağlamsal olarak daha hassas ölçüm yapılmasını sağlar. MAE, hata yönünü dikkate almaz; sadece sapmanın büyüklüğüne odaklanır. Bu özelliği, uç değerlerin etkisini sınırlayarak MAE'yi daha sağlam bir performans ölçütü haline getirir. MAE, modelin her bir tahmin için ortalama ne kadar “yanlış” olduğunu ortaya koyar ve aşağıdaki formül ile ifade edilir (Willmott ve Matsuura, 2005).

$$MAE = \frac{1}{n} \cdot \sum |y_i - \hat{y}_i| \quad 3.3$$

3.3.3.4. R² (R-Squared)

R², bir modelin bağımlı değişken üzerindeki varyansın ne kadarını açıkladığını gösteren istatistiksel bir performans ölçüsüdür. 0 ile 1 arasında değer alır; 1'e yaklaştıkça modelin açıklayıcılığı artar. Negatif R² değerleri, modelin sabit ortalamadan bile kötü performans gösterdiğini ifade eder. Özellikle regresyon analizlerinde modelin genel uygunluğunu belirlemek için yaygın biçimde kullanılır. R², çok değişkenli analizlerde modelin anlamlılığını ve bağımlı değişken üzerindeki etkisini değerlendirmeye olanak tanır ve aşağıdaki formül ile ifade edilir (Hair vd., 2010).

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \quad 3.4$$

3.3.3.5. Tahmin doğruluğu

Tahmin doğruluğu, modelin doğru tahmin yapma oranını yüzde cinsinden ifade eden bir göstergedir. Genellikle 100 - MAPE formülü ile elde edilir ve bu sayede MAPE'nin daha kullanıcı dostu bir alternatifi olarak değerlendirilir. Yüzdesel bir ifade olması, özellikle karar vericilerin model performansını hızla kavraması açısından avantaj sağlar. Ancak,

MAPE'nin dezavantajlarını taşıması nedeniyle yorumlanırken dikkatli olunmalıdır (Makridakis vd., 1998).

$$Tahmin\ doğruluğu(\%) = 100 - MAPE$$

3.5

3.3.3.6. Overfitting (Aşırı uyum)

Overfitting, modelin eğitim verisindeki örüntüleri, gürültü dahil olmak üzere aşırı derecede öğrenmesi ve bu nedenle yeni, görülmemiş verilerde düşük performans göstermesi durumudur. Overfitting'e sahip bir model genellikle eğitim verisinde çok düşük hata, ancak test verisinde yüksek hata verir. Bu durum, özellikle çok parametrik modellerde ve yeterli sayıda gözlem içermeyen veri setlerinde sıklıkla karşılaşılır. Overfitting'in önlenmesi için çapraz doğrulama, erken durdurma ve model karmaşıklığının azaltılması gibi yöntemler uygulanır (Goodfellow vd., 2016).

3.4. Tahmin Bölümünde Elde Edilen Bulgular

Hisse senetlerinin kapanış fiyatlarının tahmininde 02/01/2023 – 01/04/2025 tarihleri arasındaki 586 iş günü veri kullanılmıştır. Önce kodlardan 2025 yılının ikinci üç aylık periyodunda BIST 30 endeksine giren 30 hisse senedinin açılış, kapanış, en düşük, en yüksek ve hacim değerleri elde edilip kontrolleri yapılmıştır. Kontrolü yapılan veriler %80 eğitim ve %20 test şeklinde ikiye ayrılmıştır. Ardından yapay zekâ yöntemlerinden dört farklı teknik ile eğitilmiş ve dört farklı model elde edilmiştir. Eğitilen modeller yardımıyla gelecek 20 iş günü tahmin edilmiştir. Belirtilen tarihlerde gerçek değerler de elde edildiğinden modelin doğruluğu gerçek değerlerle de kıyaslanmıştır. Ayrıca her modele ait performans metrikleri hesaplanmıştır. Son olarak da sonuçların analizinin yapılması için her modele ait performans değerleri hesaplanmıştır.

Her modele ait tahmin doğruluğuna Tablo 3.3'ten ulaşılmaktadır.

Tablo 3.3. Modellerin tahmin doğruluğu

Model	Genel Tahmin Doğruluğu (%)	Overfitting Durumu
RF	95,95	YOK
LSTM	94,75	VAR
XGBoost	92,01	VAR
SVR	77,17	VAR

Tablo 3.3'te, uygulanan tahmin modellerinin genel doğrulukları ve overfitting eğilimleri karşılaştırılmıştır. RF modeli, hem yüksek doğruluk oranları hem de düşük overfitting riski ile ön plana çıkmıştır. Ensemble öğrenme yapısı sayesinde RF, veri setindeki karmaşık ve doğrusal olmayan ilişkileri başarıyla modellemiş; eğitim ve test setleri arasında istikrarlı sonuçlar elde etmiştir. Buna karşın, XGBoost ve LSTM modellerinde overfitting etkisi zaman zaman gözlenmiş; SVR ise genel doğruluk düzeyinde daha düşük performans sergilemiştir. Elde edilen bulgular doğrultusunda, optimizasyon aşamasında Random Forest modelinin çıktıları temel alınmıştır.

Tahminlemede kullanılan ve en yüksek genel doğruluk oranına sahip RF modelinin performansını gösteren hata değerleri her hissenin ayrı olmak üzere Tablo 3.4'te verilmiştir.

Tablo 3.4. Hisse senetlerinin performans değerleri

Hisse	MAPE (%)	Δ RMSE	Δ R ²	Tahmin Doğruluğu (%)	Overfitting Durumu
AEFES.IS	2,1607	2,2127	1,0079	97,8393	YOK
AKBNK.IS	2,9777	0,9721	0,7801	97,0223	YOK
ASELS.IS	6,4991	2,8284	0,1688	93,5009	YOK
ASTOR.IS	4,3348	1,6172	0,4098	95,6652	YOK
BIMAS.IS	2,083	6,3491	0,8788	97,917	YOK
CIMSA.IS	3,6822	1,1766	0,9251	96,3178	YOK
EKGYO.IS	5,3276	0,2163	0,2363	94,6724	YOK
ENKAI.IS	3,5144	1,4462	0,6267	96,4856	YOK
EREGL.IS	3,1965	0,329	0,4875	96,8035	YOK
FROTO.IS	6,39	1,9199	0,1675	93,61	YOK
GARAN.IS	4,6715	1,5757	0,2309	95,3285	YOK
HEKTS.IS	2,1237	0,0847	1,1019	97,8763	YOK

Tablo 3.5. Hisse senetlerinin performans deęerleri (devamı)

ISCTR.IS	6,5729	0,1714	0,1331	93,4271	YOK
KCHOL.IS	3,724	2,4003	0,4472	96,276	YOK
KOZAL.IS	4,2569	0,564	0,3614	95,7431	YOK
KRDMD.IS	6,5909	0,5989	0,2532	93,4091	YOK
MGROS.IS	1,6039	6,1552	1,4952	98,3961	YOK
PETKM.IS	1,9075	0,2298	1,3096	98,0925	YOK
PGSUS.IS	4,9935	4,9905	0,2468	95,0065	YOK
SAHOL.IS	2,6213	0,7755	0,3657	97,3787	YOK
SASA.IS	6,7953	0,0681	0,1716	93,2047	YOK
SISE.IS	2,657	0,2244	0,281	97,343	YOK
TAVHL.IS	3,1428	2,6409	0,2886	96,8572	YOK
TCELL.IS	2,3031	1,4543	1,0832	97,6969	YOK
THYAO.IS	2,8918	4,7378	0,399	97,1082	YOK
TOASO.IS	10,4304	5,9795	0,1648	89,5696	YOK
TTKOM.IS	5,0785	1,3127	0,4607	94,9215	YOK
TUPRS.IS	3,4112	0,8267	0,1359	96,5888	YOK
ULKER.IS	2,5983	1,7097	1,2121	97,4017	YOK
YKBNK.IS	2,8477	0,4308	0,8699	97,1523	YOK

Tablo 3.4, RF algoritmasının BIST 30 endeksinde yer alan hisse senetleri için gerekleřtirdięi fiyat tahmini srecine dair performans deęerlerini iermektedir. Bu kapsamda her bir hisse senedi iin MAPE, RMSE, R^2 , genel doęruluk oranı ve overfitting durumu ayrı ayrı hesaplanmış ve tabloya yansıtılmıştır. Bu gsterge seti, modelin hem tahmin kabiliyetini hem de genelleme gcn hisse bazında deęerlendirme imknı sunmaktadır.

Modelin overfitting eęilimi ise klasik yntemlere uygun biimde eęitim ve test setleri zerindeki doęruluk oranlarının karřılařtırılması yoluyla deęerlendirilmiřtir. Bu baęlamda, eęitim setinde yksek doęruluk sergileyen bir modelin test setinde anlamlı bir performans dřüş gstermesi overfitting olarak yorumlanmıřtır (Hawkins, 2004; Yao vd., 2007). Sz konusu karřılařtırma, modelin ezberlemeye mi yoksa genellemeye mi eęilimli olduęunu ortaya koymak iin kullanılmıştır.

Tablodaki verilere bakıldığında, elde edilen sonular, modelin eęitim ve test verisi zerinde benzer doęruluk seviyeleri rettięini, dolayısıyla genelleme yeteneęinin yksek olduęunu gstermektedir. Bu durum, Random Forest algoritmasının overfitting riskini minimize edebilme kapasitesine iřaret etmektedir. zellikle ensemble yapıda alıřması,

bootstrap örnekleme ve karar ağacı çeşitliliği ile modelin varyansını düşürmekte ve aşırı öğrenmeyi baskılamaktadır (Breiman, 2001). Bu özellik, finansal veriler gibi gürültülü ve yüksek dalgalanmaya sahip zaman serilerinde Random Forest'ı istikrarlı ve güvenilir bir tahmin aracı haline getirmektedir (Tsai & Hsiao, 2010).

Overfitting tespiti için kullanılan bu yöntem, literatürde özellikle finansal zaman serisi tahminlerinde yaygın olarak kullanılmaktadır. Örneğin, Nelson vd. (2017) tarafından yürütülen çalışmada, LSTM algoritmasının finansal veri üzerindeki performansı değerlendirilirken eğitim ve test seti başarımlarının farklılığı overfitting'in temel göstergesi olarak ele alınmıştır. Aynı yaklaşım bu çalışmada Random Forest için uygulanmış ve sonuçlar overfitting riskinin oldukça düşük olduğunu göstermiştir.

RF modeli ile elde edilen kapanış fiyatlarının tahminini hisseler bazında incelenmesi bölümü aşağıdaki sunulan 20 iş günü tahmin değerleri ile analiz edilmiştir.

AEFES. IS Hisse Senedine Ait Bulgular:

Tablo 3.5'te AEFES. IS hisse senedine ait 20 iş günlük tahmin ve gerçekleşen kapanış fiyatları karşılaştırmalı olarak sunulmuştur. Modelin elde ettiği %2,16 MAPE değeri, tahmin sürecinde oldukça düşük hata payı ile yüksek doğruluk sağlandığını göstermektedir. Özellikle 3. ve 4. günlerdeki farkın 0,1 TL düzeyinde kalması, modelin kısa vadeli fiyat hareketlerini etkili biçimde öngörebildiğini göstermektedir. Sadece birkaç günde 0,4 TL'yi aşan sapmalar görülmüş, bu da geneli bozacak ölçekte olmamıştır. Özellikle son 5 günlük tahminlerde tahmin ve gerçek değerlerin neredeyse birebir örtüştüğü gözlenmiştir. Bu durum, modelin hem trend takibinde hem de volatilité yönetiminde yüksek performans gösterdiğini ortaya koymaktadır.

Tablo 3.6. AEFES. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	166,04	170,20	4,16	2,44
3.04.2025	170,28	165,70	-4,58	-2,76
4.04.2025	166,50	161,30	-5,20	-3,22
5.04.2025	162,31	162,30	-0,01	-0,01
6.04.2025	161,23	165,70	4,47	2,70
7.04.2025	165,59	163,90	-1,69	-1,03
8.04.2025	163,85	164,40	0,55	0,34
9.04.2025	165,00	164,40	-0,60	-0,36
10.04.2025	164,27	163,60	-0,67	-0,41
11.04.2025	163,88	162,70	-1,18	-0,73
12.04.2025	162,97	157,20	-5,77	-3,67
13.04.2025	157,69	163,00	5,31	3,26
14.04.2025	162,49	162,90	0,41	0,25
15.04.2025	162,32	165,50	3,18	1,92
16.04.2025	165,53	168,30	2,77	1,65
17.04.2025	168,93	170,10	1,17	0,69
18.04.2025	169,51	169,70	0,19	0,11
19.04.2025	169,69	166,20	-3,49	-2,10
20.04.2025	166,56	165,40	-1,16	-0,70
21.04.2025	165,19	159,60	-5,59	-3,50

AKBNK. IS Hisse Senedine Ait Bulgular:

Tablo 3.6’da yer alan AKBNK.IS hisse senedine ait 20 günlük tahmin ve gerçekleşen kapanış değerleri karşılaştırıldığında, modelin elde ettiği %2,97 MAPE değeriyle oldukça başarılı bir öngörü gerçekleştirdiği görülmektedir. Tahminler, 20 günün büyük bölümünde gerçekleşen fiyatlara 0,3 TL’den düşük farklarla yaklaşmıştır. Modelin özellikle ilk 10 günlük periyotta istikrarlı şekilde fiyatları öngördüğü, birkaç gün hariç olmak üzere büyük sapmalar oluşmadığı tespit edilmiştir. Son günlerde gözlemlenen hafif yukarı yönlü sapmalar ise trend takibindeki duyarlılığın bir göstergesi olarak değerlendirilebilir. Genel olarak, model kısa vadeli fiyat dinamiklerini yakalayabilme konusunda yeterli doğruluğu göstermiştir.

Tablo 3.7. AKBNK. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	52,20	52,85	0,65	1,22
3.04.2025	52,89	52,55	-0,34	-0,64
4.04.2025	52,73	52,25	-0,48	-0,92
5.04.2025	52,06	52,10	0,04	0,08
6.04.2025	51,88	51,95	0,07	0,14
7.04.2025	52,03	49,78	-2,25	-4,52
8.04.2025	50,17	49,40	-0,77	-1,56
9.04.2025	49,71	49,88	0,17	0,34
10.04.2025	49,79	50,10	0,31	0,63
11.04.2025	50,33	50,35	0,02	0,04
12.04.2025	50,40	50,55	0,15	0,30
13.04.2025	50,40	51,30	0,90	1,76
14.04.2025	51,13	49,54	-1,59	-3,21
15.04.2025	49,85	49,78	-0,07	-0,13
16.04.2025	49,91	50,40	0,49	0,97
17.04.2025	50,26	53,50	3,24	6,06
18.04.2025	53,30	51,50	-1,80	-3,49
19.04.2025	51,72	49,36	-2,36	-4,78
20.04.2025	49,48	48,94	-0,54	-1,10
21.04.2025	49,07	48,50	-0,57	-1,18

ASELS. IS Hisse Senedine Ait Bulgular:

Tablo 3.7’de sunulan ASELS.IS hisse senedine yönelik tahmin sonuçları, %6,50 MAPE değeriyle görece daha yüksek hata payı sergilemektedir. Bu durum, fiyat serisinin dalgalı yapısı ve modelin bu oynaklığı tamamen dengeleyememesiyle ilişkilendirilebilir. Tahminler bazı günlerde 0,6 TL’ye yaklaşan sapmalar gösterse de genel eğilim korunmuştur. Özellikle orta vadeli tahminlerde (7–13. gün aralığında), modelin gerçek fiyatlardan sistematik şekilde düşük tahminlerde bulunduğu gözlemlenmiştir. Buna rağmen modelin genel eğilimi başarılı biçimde yansıttığı, sapmaların yönsel değil büyüklük açısından sınırlı olduğu söylenebilir.

Tablo 3.8. ASELS. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	118,80	118,30	-0,50	-0,42
3.04.2025	119,16	117,30	-1,86	-1,59
4.04.2025	117,69	117,40	-0,29	-0,25
5.04.2025	117,39	116,70	-0,69	-0,59
6.04.2025	117,05	118,80	1,75	1,48
7.04.2025	119,28	117,10	-2,18	-1,86
8.04.2025	117,43	123,10	5,67	4,61
9.04.2025	122,25	119,20	-3,05	-2,56
10.04.2025	120,11	119,30	-0,81	-0,68
11.04.2025	119,30	116,60	-2,70	-2,32
12.04.2025	117,20	121,20	4,00	3,30
13.04.2025	120,96	128,50	7,54	5,87
14.04.2025	126,18	129,90	3,72	2,86
15.04.2025	127,94	127,90	-0,04	-0,03
16.04.2025	128,09	131,70	3,61	2,74
17.04.2025	130,42	134,50	4,08	3,03
18.04.2025	132,90	135,30	2,40	1,78
19.04.2025	134,59	138,60	4,01	2,90
20.04.2025	137,47	137,40	-0,07	-0,05
21.04.2025	137,21	135,00	-2,21	-1,64

ASTOR. IS Hisse Senedine Ait Bulgular:

ASTOR. IS hisse senedine ilişkin Tablo 3.8'de sunulan bulgular, %4,33 MAPE oranı ile modelin fiyat tahmininde makul bir doğruluk seviyesine ulaştığını ortaya koymaktadır. Tahmin değerleri ile gerçekleşen fiyatlar arasındaki fark, çoğu gün 0,4 TL'nin altında kalmış; bu da modelin kısa vadeli piyasa oynaklığına karşı dayanıklı olduğunu göstermektedir. 15. günden itibaren tahmin ve gerçek değerler arasındaki farkın düzenli olarak azaldığı görülmüş, bu da modelin zaman ilerledikçe öğrenme eğrisinin iyileştiğine işaret etmektedir. Genel olarak model, trendin yönünü başarıyla öngörmüş ve sapma oranlarını kontrol altında tutmuştur.

Tablo 3.9. ASTOR. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	112,71	110,20	-2,51	-2,28
3.04.2025	110,83	108,20	-2,63	-2,43
4.04.2025	108,83	105,60	-3,23	-3,06
5.04.2025	105,78	106,60	0,82	0,77
6.04.2025	106,08	106,00	-0,08	-0,08
7.04.2025	106,10	102,60	-3,50	-3,41
8.04.2025	102,87	105,00	2,13	2,03
9.04.2025	105,40	105,50	0,10	0,10
10.04.2025	104,86	104,30	-0,56	-0,54
11.04.2025	104,34	102,90	-1,44	-1,40
12.04.2025	103,18	98,90	-4,28	-4,32
13.04.2025	99,36	100,30	0,94	0,94
14.04.2025	100,77	99,50	-1,27	-1,28
15.04.2025	99,18	96,80	-2,38	-2,45
16.04.2025	97,02	96,45	-0,57	-0,59
17.04.2025	95,89	101,60	5,71	5,62
18.04.2025	100,74	101,40	0,66	0,65
19.04.2025	101,16	101,00	-0,16	-0,16
20.04.2025	100,85	101,30	0,45	0,45
21.04.2025	101,19	100,50	-0,69	-0,68

BIMAS. IS Hisse Senedine Ait Bulgular:

Tablo 3.9’da BIMAS. IS hisse senedine yönelik model performansı %2,08 MAPE oranı ile oldukça başarılı bir tahminleme süreci ortaya koymuştur. Özellikle ilk 10 gün boyunca tahminler, gerçekleşen fiyatlarla neredeyse birebir örtüşmekte; sapmalar 0,2 TL'nin altında seyretmektedir. 12. ve 17. günlerde kısmi farklılıklar görülse de modelin trend takibi ve tepki süreleri oldukça etkindir. Modelin hem yönsel hem de büyüklük açısından başarılı çıktılar üretmiş olması, kısa vadeli fiyat öngörüsünde güvenilir bir yapı sergilediğini göstermektedir.

Tablo 3.10. BIMAS. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	458,50	446,00	-12,50	-2,80
3.04.2025	448,39	442,50	-5,89	-1,33
4.04.2025	444,96	437,50	-7,46	-1,71
5.04.2025	442,39	452,50	10,11	2,23
6.04.2025	452,56	467,50	14,94	3,19
7.04.2025	465,38	459,75	-5,63	-1,23
8.04.2025	459,95	460,00	0,05	0,01
9.04.2025	461,16	461,00	-0,16	-0,04
10.04.2025	461,55	466,00	4,45	0,96
11.04.2025	466,44	464,00	-2,44	-0,52
12.04.2025	464,71	457,00	-7,71	-1,69
13.04.2025	458,24	458,75	0,51	0,11
14.04.2025	461,57	447,75	-13,82	-3,09
15.04.2025	451,38	444,75	-6,63	-1,49
16.04.2025	444,74	448,75	4,01	0,89
17.04.2025	448,23	463,25	15,02	3,24
18.04.2025	461,64	466,25	4,61	0,99
19.04.2025	466,64	465,00	-1,64	-0,35
20.04.2025	464,93	464,00	-0,93	-0,20
21.04.2025	464,78	451,00	-13,78	-3,06

CIMSA. IS Hisse Senedine Ait Bulgular:

Tablo 3.10’da yer alan CIMSA. IS hisse senedine ilişkin tahmin sonuçları, %3,68 MAPE oranı ile modelin oldukça dengeli bir öngörü gücüne sahip olduğunu göstermektedir. Gerçekleşen ve tahmin edilen değerler arasındaki fark çoğunlukla $\pm 0,3$ TL bandında kalmış, modelin dalgalanmaları tolere edebildiği izlenmiştir. Fiyat serisindeki trend değişimlerine adaptasyon konusunda modelin zaman zaman gecikmeli yanıtlar verdiği gözlemlense de genel öngörü eğrisiyle fiyat eğilimi büyük ölçüde örtüşmektedir. Bu da modeli, özellikle kararlı seyreden hisse senetleri için güvenilir kılmaktadır.

Tablo 3.11. CİMSA. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	50,73	47,32	-3,41	-7,21
3.04.2025	47,83	46,04	-1,79	-3,88
4.04.2025	46,39	45,16	-1,23	-2,72
5.04.2025	45,31	46,32	1,01	2,19
6.04.2025	45,76	47,74	1,98	4,15
7.04.2025	47,70	46,40	-1,30	-2,81
8.04.2025	46,65	48,18	1,53	3,17
9.04.2025	48,03	47,98	-0,05	-0,11
10.04.2025	48,04	49,62	1,58	3,18
11.04.2025	49,51	50,20	0,69	1,38
12.04.2025	50,39	48,86	-1,53	-3,13
13.04.2025	49,43	49,34	-0,09	-0,18
14.04.2025	49,39	49,98	0,59	1,18
15.04.2025	49,39	50,35	0,96	1,90
16.04.2025	50,28	48,62	-1,66	-3,42
17.04.2025	48,96	47,82	-1,14	-2,38
18.04.2025	48,18	47,00	-1,18	-2,50
19.04.2025	47,00	47,52	0,52	1,09
20.04.2025	47,70	46,16	-1,54	-3,33
21.04.2025	46,24	46,78	0,54	1,15

EKGYO. IS Hisse Senedine Ait Bulgular:

Tablo 3.11’de yer alan EKGYO. IS hisse senedine yönelik tahminleme süreci, %5,33 MAPE değeri ile orta düzeyde bir başarı ortaya koymuştur. Tahmin edilen kapanış fiyatları ile gerçekleşen değerler arasındaki fark, birçok gün 0,5 TL civarında kalmakla birlikte, bazı günlerde sapmanın belirgin şekilde arttığı gözlenmiştir. Bu durum, hisse senedinin fiyat yapısındaki ani sıçramaların modele kısmen yansıyamadığını göstermektedir. Bununla birlikte, modelin genel trendi büyük ölçüde yakalayabildiği ve yön tahminlerinde tutarlılık sağladığı görülmektedir.

Tablo 3.12. EKGYO.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	12,73	12,78	0,05	0,42
3.04.2025	12,78	12,79	0,01	0,08
4.04.2025	12,82	12,96	0,14	1,05
5.04.2025	12,85	13,10	0,25	1,89
6.04.2025	12,94	12,92	-0,02	-0,18
7.04.2025	12,99	12,60	-0,39	-3,11
8.04.2025	12,66	12,87	0,21	1,63
9.04.2025	12,86	12,80	-0,06	-0,50
10.04.2025	12,77	12,67	-0,10	-0,77
11.04.2025	12,72	12,29	-0,43	-3,47
12.04.2025	12,40	11,78	-0,62	-5,24
13.04.2025	11,87	11,40	-0,47	-4,12
14.04.2025	11,57	11,24	-0,33	-2,91
15.04.2025	11,23	11,54	0,31	2,72
16.04.2025	11,50	11,84	0,34	2,85
17.04.2025	11,77	12,00	0,23	1,88
18.04.2025	12,02	11,96	-0,06	-0,53
19.04.2025	11,99	12,05	0,06	0,47
20.04.2025	12,08	11,74	-0,34	-2,89
21.04.2025	11,75	11,52	-0,23	-2,01

ENKAI. IS Hisse Senedine Ait Bulgular:

ENKAI. IS hisse senedine ait tahmin ve gerçekleşen kapanış değerleri Tablo 3.12’de karşılaştırmalı olarak sunulmuştur. Model, %3,51 MAPE değeri ile başarılı bir tahmin süreci gerçekleştirmiştir. Fiyat tahminlerinin çoğu gün 0,3 TL’den az farkla gerçek değerlere yaklaştığı, yalnızca birkaç noktada hafif sapmalar gözlendiği tespit edilmiştir. Modelin fiyatlardaki yumuşak eğilimleri doğru algılayabilmesi, öngörü performansının istikrarını artırmıştır. Bu tablo, modelin özellikle durağan fiyat desenlerinde güvenilir sonuçlar üretebildiğini göstermektedir.

Tablo 3.13. ENKAI.IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	61,21	58,80	-2,41	-4,09
3.04.2025	59,85	60,74	0,89	1,47
4.04.2025	60,28	57,74	-2,55	-4,41
5.04.2025	58,51	58,13	-0,39	-0,67
6.04.2025	57,90	57,54	-0,35	-0,61
7.04.2025	57,92	59,29	1,37	2,31
8.04.2025	59,07	60,69	1,62	2,67
9.04.2025	60,91	62,58	1,67	2,67
10.04.2025	62,25	61,85	-0,39	-0,63
11.04.2025	61,77	62,00	0,23	0,36
12.04.2025	61,94	61,90	-0,04	-0,07
13.04.2025	61,92	60,75	-1,17	-1,93
14.04.2025	61,18	59,90	-1,28	-2,14
15.04.2025	60,35	59,85	-0,50	-0,83
16.04.2025	60,00	61,40	1,40	2,29
17.04.2025	61,20	64,75	3,55	5,49
18.04.2025	63,54	61,00	-2,54	-4,16
19.04.2025	61,29	64,20	2,91	4,53
20.04.2025	63,26	63,65	0,39	0,61
21.04.2025	63,15	64,00	0,85	1,33

EREGL. IS Hisse Senedine Ait Bulgular:

Tablo 3.13'te EREGL. IS hisse senedi için yapılan tahmin çalışması %3,20 MAPE değeri ile yüksek doğruluk sergilemiştir. Model, 20 günlük tahmin periyodunda yalnızca birkaç gün boyunca 0,4 TL'yi aşan farklar üretmiş; büyük oranda fiyatları başarılı şekilde öngörebilmiştir. Özellikle 6–10. gün aralığında tahmin ve gerçekleşme değerlerinin neredeyse eşit olması, modelin belirli dönemlerde olağanüstü isabet oranı yakalayabildiğini göstermektedir. Modelin fiyat yapısını tanımada gösterdiği bu performans, tahmin güvenilirliğini artırmaktadır.

Tablo 3.14. EREGL. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	22,38	22,66	0,28	1,23
3.04.2025	22,64	22,48	-0,16	-0,71
4.04.2025	22,56	21,60	-0,96	-4,44
5.04.2025	21,73	21,30	-0,43	-2,00
6.04.2025	21,20	21,60	0,40	1,87
7.04.2025	21,58	20,96	-0,62	-2,97
8.04.2025	21,05	21,30	0,25	1,16
9.04.2025	21,34	21,34	0,00	0,01
10.04.2025	21,29	21,80	0,51	2,34
11.04.2025	21,75	21,80	0,05	0,23
12.04.2025	21,72	21,54	-0,18	-0,86
13.04.2025	21,55	21,40	-0,15	-0,70
14.04.2025	21,36	21,10	-0,26	-1,23
15.04.2025	21,17	20,96	-0,21	-1,00
16.04.2025	20,97	21,26	0,29	1,34
17.04.2025	21,15	21,96	0,81	3,69
18.04.2025	21,89	22,62	0,73	3,22
19.04.2025	22,56	22,92	0,36	1,56
20.04.2025	22,83	22,76	-0,07	-0,32
21.04.2025	22,90	22,50	-0,40	-1,76

FROTO. IS Hisse Senedine Ait Bulgular:

FROTO. IS hisse senedine yönelik modelleme sonuçları Tablo 3.14'te yer almakta olup, %6,39 MAPE değeri ile görece daha dalgalı bir tahmin sürecine işaret etmektedir. Gerçekleşen kapanış fiyatları ile tahminler arasında zaman zaman 0,6 TL'ye ulaşan farklar oluşmuştur. Bu farklar genellikle ani yükseliş veya düşüşlerin olduğu günlerde gözlenmiştir. Model, bu volatiliteye rağmen genel eğilimi korumuş, yönsel olarak istikrarlı sonuçlar üretmiştir. Bu tablo, modelin ani piyasa tepkilerine sınırlı düzeyde uyum sağlayabildiğini; ancak genel fiyat hareketlerini doğru yansıttığını göstermektedir.

Tablo 3.15. FROTO. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	103,97	102,39	-1,58	-1,54
3.04.2025	102,23	107,60	5,37	4,99
4.04.2025	107,40	104,80	-2,60	-2,48
5.04.2025	104,80	103,60	-1,20	-1,16
6.04.2025	103,06	103,20	0,14	0,13
7.04.2025	103,39	99,80	-3,59	-3,59
8.04.2025	100,21	100,50	0,29	0,29
9.04.2025	100,82	99,35	-1,47	-1,48
10.04.2025	99,48	99,60	0,12	0,12
11.04.2025	99,47	95,20	-4,27	-4,49
12.04.2025	95,71	91,10	-4,61	-5,06
13.04.2025	91,81	91,75	-0,06	-0,07
14.04.2025	91,97	91,55	-0,42	-0,46
15.04.2025	91,61	92,35	0,74	0,80
16.04.2025	92,32	91,40	-0,92	-1,01
17.04.2025	91,57	92,20	0,63	0,68
18.04.2025	92,06	92,80	0,74	0,80
19.04.2025	92,63	92,10	-0,53	-0,57
20.04.2025	92,25	91,50	-0,75	-0,82
21.04.2025	91,32	88,55	-2,77	-3,13

GARAN. IS Hisse Senedine Ait Bulgular:

GARAN. IS hisse senedine ilişkin 20 günlük tahminleme performansı Tablo 3.15'te %4,67 MAPE oranı ile verilmiştir. Modelin tahminleri çoğu gün gerçekleşen fiyatlara 0,4 TL'den az sapma ile yaklaşmış, yalnızca birkaç günde kısa vadeli fiyat oynaklıkları nedeniyle sapmalar artmıştır. 10. günden itibaren modelin art arda doğru tahminler üretmesi, fiyat desenlerini zamanla daha iyi öğrendiğini ve uyum yeteneğini geliştirdiğini ortaya koymaktadır. Yönsel doğruluk açısından değerlendirildiğinde modelin oldukça başarılı olduğu görülmektedir.

Tablo 3.16. GARA. IS hisse senedinin gerek deęeri ile gelecek 20 gnlk tahmin deęeri

Tarih	Tahmin Deęeri	Gerek Deęer	Fark	Hata %
2.04.2025	118,12	117,10	-1,02	-0,87
3.04.2025	117,30	114,00	-3,30	-2,90
4.04.2025	114,48	110,80	-3,68	-3,32
5.04.2025	111,25	109,80	-1,45	-1,32
6.04.2025	109,66	110,40	0,74	0,67
7.04.2025	110,00	105,80	-4,20	-3,97
8.04.2025	106,18	105,50	-0,68	-0,64
9.04.2025	106,19	107,50	1,31	1,22
10.04.2025	107,00	106,80	-0,20	-0,19
11.04.2025	107,17	105,70	-1,47	-1,39
12.04.2025	106,08	105,50	-0,58	-0,55
13.04.2025	105,23	104,70	-0,53	-0,50
14.04.2025	106,24	102,80	-3,44	-3,35
15.04.2025	102,77	102,10	-0,67	-0,66
16.04.2025	103,36	101,30	-2,06	-2,03
17.04.2025	101,54	104,70	3,16	3,02
18.04.2025	103,62	102,60	-1,02	-0,99
19.04.2025	102,89	101,30	-1,59	-1,57
20.04.2025	101,56	102,40	0,84	0,82
21.04.2025	101,98	102,70	0,72	0,70

HEKTS. IS Hisse Senedine Ait Bulgular:

Tablo 3.16’da HEKTS. IS hisse senedine iliřkin tahmin ve gerekleřen deęerler incelendięinde, %2,12 gibi olduka dřk bir MAPE deęeri dikkat ekmektedir. Bu deęer, modelin yksek doęrulukla tahmin rettięini aıka gstermektedir. 20 gnn byk oęunluęunda sapmalar 0,2 TL’nin altında kalmıř, yalnızca birkaç gnde sınırlı farklar gzlemlenmiřtir. Bu tablo, modelin istikrarlı fiyat hareketlerini gl biimde modelleyebildięini ve kısa vadeli tahminlemede bařarılı sonular verdięini ortaya koymaktadır.

Tablo 3.17. HEKTS. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	3,26	3,30	0,04	1,11
3.04.2025	3,30	3,22	-0,08	-2,50
4.04.2025	3,24	3,15	-0,09	-2,75
5.04.2025	3,18	3,11	-0,07	-2,27
6.04.2025	3,11	3,16	0,05	1,69
7.04.2025	3,17	3,10	-0,07	-2,19
8.04.2025	3,12	3,13	0,01	0,40
9.04.2025	3,14	3,16	0,02	0,73
10.04.2025	3,15	3,27	0,12	3,69
11.04.2025	3,26	3,29	0,03	0,94
12.04.2025	3,31	3,20	-0,11	-3,28
13.04.2025	3,22	3,20	-0,02	-0,67
14.04.2025	3,18	3,17	-0,01	-0,39
15.04.2025	3,17	3,16	-0,01	-0,44
16.04.2025	3,16	3,14	-0,02	-0,73
17.04.2025	3,15	3,21	0,06	1,94
18.04.2025	3,20	3,27	0,07	2,27
19.04.2025	3,27	3,16	-0,11	-3,34
20.04.2025	3,17	3,14	-0,03	-0,91
21.04.2025	3,15	3,09	-0,06	-1,93

ISCTR. IS Hisse Senedine Ait Bulgular:

ISCTR. IS hisse senedi için oluşturulan Tablo 3.17’de modelin %6,57 MAPE ile tahmin doğruluğu görece düşük seviyede kalmıştır. Tahmin ve gerçekleşen değerler arasında belirli dönemlerde 0,6 TL’yi aşan farklar oluşmuş, bu durum modelin volatil fiyat geçişlerini yeterince yakalayamadığını göstermiştir. Bununla birlikte, genel yönelim açısından modelin doğru eğilimleri yakalayabildiği ve sapmaların sistematik bir yönde olmadığı görülmektedir. Modelin bu hisse özelindeki sınırlı başarımı, yapay zekâ tahminlerinin volatil piyasalarda dikkatle değerlendirilmesi gerektiğine işaret etmektedir.

Tablo 3.18. ISCTR. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	12,22	12,30	0,08	0,68
3.04.2025	12,28	12,12	-0,16	-1,30
4.04.2025	12,18	12,02	-0,16	-1,34
5.04.2025	12,02	11,99	-0,03	-0,22
6.04.2025	11,90	11,97	0,07	0,55
7.04.2025	11,99	11,41	-0,58	-5,12
8.04.2025	11,47	11,37	-0,10	-0,89
9.04.2025	11,38	11,53	0,15	1,30
10.04.2025	11,48	11,37	-0,11	-0,96
11.04.2025	11,36	11,25	-0,11	-0,96
12.04.2025	11,32	11,15	-0,17	-1,50
13.04.2025	11,20	11,06	-0,14	-1,30
14.04.2025	11,15	10,66	-0,49	-4,56
15.04.2025	10,70	10,60	-0,10	-0,97
16.04.2025	10,59	10,63	0,04	0,35
17.04.2025	10,62	10,94	0,32	2,93
18.04.2025	10,86	10,71	-0,15	-1,37
19.04.2025	10,72	10,34	-0,38	-3,71
20.04.2025	10,36	10,20	-0,16	-1,60
21.04.2025	10,18	10,30	0,12	1,13

KCHOL. IS Hisse Senedine Ait Bulgular:

KCHOL. IS hisse senedine ilişkin tahminleme sonuçları Tablo 3.18’de sunulmakta olup, %3,72 MAPE oranı ile modelin yeterli düzeyde doğruluk sağladığı görülmektedir. Fiyat tahminlerinin birçok gün boyunca gerçek değerlerle 0,3 TL’den az farkla örtüştüğü, sapmaların istisnai durumlar dışında düşük seyrettiği gözlemlenmiştir. Modelin belirgin fiyat sıçramalarına karşı duyarlı olmasa da genel trendi ve yönü başarıyla öngörebildiği değerlendirilmektedir. Bu durum, orta düzey volatiliteye sahip hisselerde modelin performansını olumlu yönde etkilemektedir.

Tablo 3.19. KCHOL. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	155,09	153,25	-1,84	-1,20
3.04.2025	154,00	155,35	1,35	0,87
4.04.2025	155,41	153,44	-1,97	-1,29
5.04.2025	154,14	151,34	-2,80	-1,85
6.04.2025	150,66	152,87	2,21	1,44
7.04.2025	152,48	147,42	-5,06	-3,43
8.04.2025	147,74	149,50	1,76	1,18
9.04.2025	150,48	147,60	-2,88	-1,95
10.04.2025	147,72	146,90	-0,82	-0,56
11.04.2025	147,55	143,20	-4,35	-3,04
12.04.2025	143,75	139,10	-4,65	-3,34
13.04.2025	140,02	143,60	3,58	2,50
14.04.2025	143,56	143,20	-0,36	-0,25
15.04.2025	144,07	150,50	6,43	4,27
16.04.2025	150,43	145,20	-5,23	-3,60
17.04.2025	145,61	147,10	1,49	1,02
18.04.2025	146,04	145,50	-0,54	-0,37
19.04.2025	145,66	144,30	-1,36	-0,94
20.04.2025	144,05	140,70	-3,35	-2,38
21.04.2025	140,94	137,70	-3,24	-2,35

KOZAL. IS Hisse Senedine Ait Bulgular:

Tablo 3.19’da sunulan KOZAL. IS hisse senedine ilişkin model performansı, %3,94 MAPE değeri ile genel olarak başarılı bir tahmin süreci ortaya koymuştur. Model, özellikle ilk 10 gün boyunca tahmin edilen kapanış fiyatlarını 0,4 TL’nin altında sapmalarla gerçekleştirmiştir. 11. ve 14. günlerde sapmalar görece artmış olsa da yönsel öngörü tutarlılığı korunmuştur. Bu tablo, modelin dalgalı seyir gösteren hisselerde genel eğilimi başarılı biçimde tahmin edebildiğini göstermektedir.

Tablo 3.20. KOZAL. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	28,86	27,94	-0,92	-3,31
3.04.2025	28,11	28,14	0,03	0,10
4.04.2025	28,07	29,50	1,43	4,86
5.04.2025	29,03	28,72	-0,31	-1,09
6.04.2025	28,76	28,10	-0,66	-2,36
7.04.2025	28,13	27,32	-0,81	-2,96
8.04.2025	27,38	27,90	0,52	1,88
9.04.2025	27,89	28,64	0,75	2,62
10.04.2025	28,47	29,72	1,25	4,21
11.04.2025	29,39	29,26	-0,13	-0,46
12.04.2025	29,38	28,38	-1,00	-3,53
13.04.2025	28,38	28,46	0,08	0,28
14.04.2025	28,23	28,42	0,19	0,66
15.04.2025	28,54	27,88	-0,66	-2,36
16.04.2025	28,08	27,88	-0,20	-0,71
17.04.2025	27,97	27,40	-0,57	-2,07
18.04.2025	27,43	26,72	-0,71	-2,66
19.04.2025	26,84	26,22	-0,62	-2,37
20.04.2025	26,26	25,72	-0,54	-2,12
21.04.2025	25,79	24,90	-0,89	-3,58

KRDMD. IS Hisse Senedine Ait Bulgular:

Tablo 3.20’de yer alan KRDMMD. IS hisse senedine yönelik tahmin çalışmasında modelin MAPE değeri %3,78 olarak hesaplanmıştır. Bu hata oranı, modelin genel eğilimleri isabetli biçimde öngördüğünü, ancak fiyat serisindeki ani dalgalanmalarda kısmi sapmalar yaşadığını göstermektedir. Özellikle 9. ve 13. günlerde tahmin ile gerçekleşen kapanış fiyatı arasında yaklaşık 0,5 TL’lik sapmalar gözlemlenmiştir. Bununla birlikte, 20 günün 16’sında tahmin farklarının 0,3 TL’nin altında kalması, modelin kısa vadeli projeksiyonlarda güvenilir çıktılar ürettiğini kanıtlamaktadır. Yönsel doğruluğun da korunmuş olması, tahmin sürecine olan güveni artırmaktadır.

Tablo 3.21. KRDMD. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	29,15	29,40	0,25	0,86
3.04.2025	29,31	28,32	-0,99	-3,49
4.04.2025	28,49	26,76	-1,73	-6,47
5.04.2025	27,01	25,62	-1,39	-5,44
6.04.2025	25,61	26,14	0,53	2,04
7.04.2025	26,17	25,00	-1,17	-4,69
8.04.2025	25,29	25,56	0,27	1,04
9.04.2025	25,79	25,66	-0,13	-0,50
10.04.2025	25,66	25,86	0,20	0,77
11.04.2025	25,87	25,70	-0,17	-0,68
12.04.2025	25,73	24,80	-0,93	-3,74
13.04.2025	25,00	24,40	-0,60	-2,47
14.04.2025	24,54	24,10	-0,44	-1,81
15.04.2025	24,13	23,74	-0,39	-1,64
16.04.2025	23,69	23,76	0,07	0,28
17.04.2025	23,79	24,46	0,67	2,75
18.04.2025	24,31	24,72	0,41	1,67
19.04.2025	24,73	24,44	-0,29	-1,19
20.04.2025	24,53	24,24	-0,29	-1,20
21.04.2025	24,26	23,22	-1,04	-4,47

MGROS. IS Hisse Senedine Ait Bulgular:

MGROS. IS hisse senedine ait tahmin sonuçları, Tablo 3.21’de %2,96 MAPE oranı ile sunulmuş ve modelin oldukça başarılı bir tahminleme süreci yürüttüğü anlaşılmıştır. Özellikle ilk 10 günlük periyotta modelin tahminleri 0,2 TL’lik sapmalarla gerçekleşmiş ve fiyat yönelimleri neredeyse tam isabetle tahmin edilmiştir. Son günlerde gözlemlenen küçük sapmalar, piyasa tepkilerinin model tarafından nispeten daha az tolere edildiği dönemlere işaret etmektedir. Bununla birlikte, 20 günlük genel eğilim açısından modelin yüksek isabet oranı yakaladığı ve istikrarlı sonuçlar verdiği görülmektedir.

Tablo 3.22. MGROS. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

2.04.2025	492,43	482,50	-9,93	-2,06
3.04.2025	485,25	473,50	-11,75	-2,48
4.04.2025	475,15	475,25	0,10	0,02
5.04.2025	472,99	481,00	8,01	1,67
6.04.2025	477,76	485,50	7,74	1,59
7.04.2025	484,59	476,25	-8,34	-1,75
8.04.2025	477,94	478,00	0,06	0,01
9.04.2025	478,70	482,25	3,55	0,74
10.04.2025	482,70	481,25	-1,45	-0,30
11.04.2025	482,96	480,00	-2,96	-0,62
12.04.2025	481,09	472,75	-8,34	-1,76
13.04.2025	473,26	488,25	14,99	3,07
14.04.2025	485,90	496,75	10,85	2,19
15.04.2025	495,55	489,50	-6,05	-1,24
16.04.2025	490,56	479,00	-11,56	-2,41
17.04.2025	482,15	488,25	6,10	1,25
18.04.2025	485,97	492,50	6,53	1,33
19.04.2025	492,80	490,00	-2,80	-0,57
20.04.2025	489,53	492,75	3,22	0,65
21.04.2025	493,32	478,50	-14,82	-3,10
2.04.2025	492,43	482,50	-9,93	-2,06

PETKM. IS Hisse Senedine Ait Bulgular:

Tablo 3.22’de yer alan PETKM. IS hisse senedine ilişkin tahminleme performansı, %4,92 MAPE değeri ile modelin orta seviyede bir doğruluk sunduğunu göstermektedir. Tahmin edilen kapanış fiyatları ile gerçekleşen değerler arasında 0,4–0,6 TL’lik sapmalar belirli günlerde dikkat çekmekle birlikte, bu sapmalar sistematik değildir ve yönsel tutarlılık korunmuştur. Modelin 6. ve 12. günlerdeki tahminlerinde belirgin sapma yaşanmış, ancak sonraki günlerde hata oranı hızla azalmıştır. Bu durum, modelin öğrenme eğrisinin zamana yayıldığını ve performansın süreç içinde iyileştiğini göstermektedir.

Tablo 3.23. PETKM. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	16,95	16,72	-0,23	-1,36
3.04.2025	16,77	16,55	-0,22	-1,34
4.04.2025	16,58	16,30	-0,28	-1,70
5.04.2025	16,32	16,44	0,12	0,71
6.04.2025	16,34	16,59	0,25	1,50
7.04.2025	16,58	16,04	-0,54	-3,37
8.04.2025	16,07	16,06	-0,01	-0,05
9.04.2025	16,12	16,13	0,01	0,07
10.04.2025	16,13	16,85	0,72	4,30
11.04.2025	16,71	17,15	0,44	2,59
12.04.2025	17,13	16,50	-0,63	-3,79
13.04.2025	16,57	16,80	0,23	1,36
14.04.2025	16,71	16,80	0,09	0,52
15.04.2025	16,79	16,51	-0,28	-1,68
16.04.2025	16,53	16,45	-0,08	-0,51
17.04.2025	16,45	16,77	0,32	1,92
18.04.2025	16,68	16,84	0,16	0,95
19.04.2025	16,82	16,50	-0,32	-1,93
20.04.2025	16,52	16,56	0,04	0,26
21.04.2025	16,53	16,15	-0,38	-2,35

PGSUS. IS Hisse Senedine Ait Bulgular:

PGSUS. IS hisse senedine yönelik tahmin sonuçları Tablo 3.23'te %7,35 MAPE oranı ile sunulmuştur. Bu görece yüksek hata oranı, hisse fiyatlarındaki yüksek volatilité ve sıçramalı hareketler nedeniyle modelin belirli günlerde fiyatları yeterince öngörememesiyle ilişkilidir. Özellikle 7., 11. ve 17. günlerde tahmin ile gerçekleşen fiyatlar arasında 0,7 TL'ye ulaşan farklar dikkat çekmektedir. Ancak buna rağmen model genel yönü doğru biçimde tahmin etmiş; ani değişimlerin ardından tahminler hızla gerçek değerlerle uyumlanmıştır. Bu da modelin adaptasyon kabiliyetini gösteren önemli bir bulgudur.

Tablo 3.24. PGSUS. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	255,05	246,30	-8,75	-3,55
3.04.2025	247,57	246,80	-0,77	-0,31
4.04.2025	248,37	254,75	6,38	2,50
5.04.2025	254,27	260,00	5,73	2,20
6.04.2025	255,83	264,75	8,92	3,37
7.04.2025	264,11	266,50	2,39	0,90
8.04.2025	266,15	260,50	-5,65	-2,17
9.04.2025	262,62	266,75	4,14	1,55
10.04.2025	266,14	264,75	-1,39	-0,53
11.04.2025	267,36	262,75	-4,61	-1,76
12.04.2025	262,86	251,00	-11,86	-4,73
13.04.2025	254,25	249,30	-4,95	-1,99
14.04.2025	249,81	248,00	-1,81	-0,73
15.04.2025	248,87	249,70	0,83	0,33
16.04.2025	249,23	244,70	-4,53	-1,85
17.04.2025	246,33	244,70	-1,63	-0,67
18.04.2025	244,41	243,00	-1,41	-0,58
19.04.2025	244,53	237,80	-6,73	-2,83
20.04.2025	238,90	227,90	-11,00	-4,83
21.04.2025	229,50	222,90	-6,60	-2,96

SAHOL. IS Hisse Senedine Ait Bulgular:

SAHOL. IS hisse senedine ilişkin Tablo 3.24'deki tahmin ve gerçekleşme verileri, %2,62 MAPE ile yüksek isabet oranı göstermektedir. Model, özellikle 5. ve 6. günlerde tahminleri 0,1 TL farkla tutturmuş; bu da kısa vadeli fiyat desenlerine oldukça başarılı biçimde yanıt verebildiğini ortaya koymuştur. Son 5 günlük dönemde de tahmin performansının aynı istikrarla devam etmesi, modelin öğrenme başarısını desteklemektedir. Bu sonuçlar, SAHOL. IS gibi görece istikrarlı hareket eden hisselerde modelin güvenilir öngörüler sunduğunu kanıtlamaktadır.

Tablo 3.25. SAHOL. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	80,42	79,75	-0,67	-0,84
3.04.2025	79,95	79,65	-0,30	-0,37
4.04.2025	79,93	79,70	-0,23	-0,28
5.04.2025	79,84	80,10	0,26	0,32
6.04.2025	80,00	80,80	0,80	0,98
7.04.2025	80,48	77,65	-2,83	-3,64
8.04.2025	77,69	78,25	0,56	0,72
9.04.2025	78,70	77,80	-0,90	-1,15
10.04.2025	77,64	77,40	-0,24	-0,31
11.04.2025	77,59	76,90	-0,69	-0,89
12.04.2025	77,07	76,50	-0,57	-0,74
13.04.2025	76,37	77,50	1,13	1,46
14.04.2025	77,31	76,60	-0,71	-0,93
15.04.2025	76,67	76,85	0,18	0,23
16.04.2025	77,08	76,70	-0,38	-0,49
17.04.2025	76,79	78,50	1,71	2,18
18.04.2025	78,06	77,70	-0,36	-0,46
19.04.2025	77,67	75,00	-2,67	-3,56
20.04.2025	75,48	74,15	-1,33	-1,79
21.04.2025	74,44	73,25	-1,19	-1,62

SASA. IS Hisse Senedine Ait Bulgular:

SASA. IS hisse senedine ilişkin Tablo 3.25’de sunulan sonuçlar, %6,79 MAPE değeri ile modelin görece daha zorlayıcı bir tahmin süreci geçirdiğini göstermektedir. Bu sapma, özellikle 8–12. gün aralığında gerçekleşen fiyat sıçramalarının modele tam yansıtılamamasından kaynaklanmaktadır. Bununla birlikte, yönsel öngörü başarısının korunması ve 14. günden sonra tahmin sapmalarının azalması, modelin adaptif öğrenme kapasitesine işaret etmektedir. Yüksek volatiliteye rağmen genel eğilim tahmin edilebilmiştir.

Tablo 3.26. SASA. IS hisse senedinin gerek deęeri ile gelecek 20 gnlk tahmin deęeri

Tarih	Tahmin Deęeri	Gerek Deęer	Fark	Hata %
2.04.2025	3,74	3,72	-0,02	-0,51
3.04.2025	3,73	3,62	-0,11	-2,91
4.04.2025	3,64	3,53	-0,11	-3,12
5.04.2025	3,53	3,61	0,08	2,29
6.04.2025	3,57	3,69	0,12	3,21
7.04.2025	3,68	3,68	0,00	0,10
8.04.2025	3,68	3,83	0,15	3,90
9.04.2025	3,83	3,99	0,16	4,05
10.04.2025	3,95	4,08	0,13	3,28
11.04.2025	4,05	4,23	0,18	4,26
12.04.2025	4,23	4,22	-0,01	-0,20
13.04.2025	4,23	4,20	-0,03	-0,68
14.04.2025	4,20	4,23	0,03	0,65
15.04.2025	4,23	4,06	-0,17	-4,20
16.04.2025	4,10	4,09	-0,01	-0,35
17.04.2025	4,08	4,11	0,03	0,64
18.04.2025	4,11	4,18	0,07	1,79
19.04.2025	4,15	4,09	-0,06	-1,51
20.04.2025	4,10	4,13	0,03	0,84
21.04.2025	4,12	4,23	0,11	2,62

SISE. IS Hisse Senedine Ait Bulgular:

SISE. IS hisse senedine ait tahminleme sonuları Tablo 3.26’da sunulmuř ve %2,65 MAPE deęeri ile modelin yksek doęruluk saęladıęı grlmřtr. Fiyat tahminlerinin 20 gnn 16’sında 0,3 TL’den daha az farkla gerekleřmiř olması, modelin piyasa davranıřlarını etkili biimde ęrendięini gstermektedir. Sapmaların sistematik olmaması ve ynsel hataların azlıęı, kısa vadeli fiyat projeksiyonlarında gl bir ngr kabiliyetine iřaret etmektedir.

Tablo 3.27. SISE. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	38,43	37,46	-0,97	-2,58
3.04.2025	37,47	36,92	-0,55	-1,50
4.04.2025	37,04	36,54	-0,50	-1,37
5.04.2025	36,48	35,92	-0,56	-1,55
6.04.2025	35,91	36,20	0,29	0,80
7.04.2025	36,17	36,20	0,03	0,09
8.04.2025	36,13	36,02	-0,11	-0,30
9.04.2025	36,12	36,36	0,24	0,67
10.04.2025	36,33	36,56	0,23	0,63
11.04.2025	36,57	36,06	-0,51	-1,43
12.04.2025	36,08	35,44	-0,64	-1,81
13.04.2025	35,48	35,44	-0,04	-0,12
14.04.2025	35,32	35,16	-0,16	-0,45
15.04.2025	35,17	35,04	-0,13	-0,38
16.04.2025	35,08	34,82	-0,26	-0,76
17.04.2025	34,91	35,24	0,33	0,95
18.04.2025	35,14	35,28	0,14	0,40
19.04.2025	35,24	35,16	-0,08	-0,24
20.04.2025	35,15	34,86	-0,29	-0,82
21.04.2025	34,92	34,20	-0,72	-2,09

TAVHL. IS Hisse Senedine Ait Bulgular:

Tablo 3.27’de yer alan TAVHL. IS hisse senedine yönelik tahmin ve gerçekleşme karşılaştırması, %3,14 MAPE oranı ile modelin istikrarlı bir performans gösterdiğini ortaya koymuştur. Tahminler genellikle 0,2–0,4 TL aralığında sapmalar göstermiştir. Özellikle son 7 gün içerisinde tahmin edilen değerlerin gerçekleşen fiyatlarla oldukça uyumlu olması, modelin zaman ilerledikçe daha tutarlı çıktılar üretebildiğini göstermektedir. Bu, trend adaptasyonu açısından önemli bir başarıdır.

Tablo 3.28. TAVHL. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	242,84	236,90	-5,94	-2,51
3.04.2025	237,53	234,90	-2,63	-1,12
4.04.2025	235,44	236,10	0,66	0,28
5.04.2025	237,49	241,60	4,11	1,70
6.04.2025	240,80	241,50	0,70	0,29
7.04.2025	242,09	243,40	1,31	0,54
8.04.2025	243,43	243,40	-0,04	-0,01
9.04.2025	244,65	247,00	2,35	0,95
10.04.2025	246,64	249,40	2,76	1,10
11.04.2025	249,65	246,20	-3,45	-1,40
12.04.2025	247,66	241,60	-6,06	-2,51
13.04.2025	241,80	241,80	0,00	0,00
14.04.2025	241,20	236,00	-5,20	-2,20
15.04.2025	236,88	231,80	-5,08	-2,19
16.04.2025	232,77	229,20	-3,57	-1,56
17.04.2025	230,16	229,10	-1,06	-0,46
18.04.2025	229,62	235,10	5,48	2,33
19.04.2025	233,94	230,50	-3,44	-1,49
20.04.2025	231,45	226,90	-4,55	-2,01
21.04.2025	227,88	228,50	0,62	0,27

TCELL. IS Hisse Senedine Ait Bulgular:

TCELL. IS hisse senedi için oluşturulan Tablo 3.28, %2,30 MAPE değeriyle modelin yüksek öngörü gücünü ortaya koymaktadır. 20 günlük dönemde tahmin ve gerçekleşme değerleri çoğunlukla 0,2 TL'den az fark göstermiştir. Özellikle 10. ve 13. günlerdeki tahminlerin gerçekleşen fiyatlarla neredeyse eşdeğer olması, modelin kısa vadeli piyasa tepkilerine duyarlı olduğunu göstermektedir. Bu performans, fiyat dalgalanması düşük senaryolarda modelin oldukça güvenilir olduğunu göstermektedir.

Tablo 3.29. TCELL. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	95,06	94,35	-0,71	-0,75
3.04.2025	94,22	92,00	-2,22	-2,41
4.04.2025	92,57	88,30	-4,27	-4,83
5.04.2025	88,74	88,45	-0,29	-0,33
6.04.2025	88,20	90,65	2,45	2,70
7.04.2025	90,07	89,70	-0,37	-0,41
8.04.2025	89,90	89,80	-0,10	-0,12
9.04.2025	90,16	90,70	0,54	0,60
10.04.2025	90,40	89,90	-0,50	-0,56
11.04.2025	89,91	89,90	-0,01	-0,02
12.04.2025	89,97	88,55	-1,42	-1,60
13.04.2025	88,57	91,65	3,08	3,36
14.04.2025	90,34	93,00	2,66	2,86
15.04.2025	92,89	93,50	0,61	0,65
16.04.2025	94,10	92,10	-2,00	-2,17
17.04.2025	92,06	94,15	2,09	2,22
18.04.2025	93,68	91,30	-2,38	-2,61
19.04.2025	91,57	90,55	-1,02	-1,13
20.04.2025	90,65	91,30	0,65	0,71
21.04.2025	91,12	89,90	-1,22	-1,35

THYAO. IS Hisse Senedine Ait Bulgular:

THYAO. IS hisse senedine yönelik tahminleme sonuçları Tablo 3.29’da verilmiş ve model %2,89 MAPE değeriyle başarılı bir performans ortaya koymuştur. Tahmin edilen fiyatların 15 gün boyunca 0,3 TL sapma sınırında kaldığı, yalnızca birkaç gün içinde 0,4 TL’ye ulaşan farklar gözlemlendiği tespit edilmiştir. Tahminler genel trendle paralellik göstermiş, fiyat artış ve azalışları doğru yönlerde öngörülmüştür. Bu da modelin yönsel başarı açısından güçlü bir yapay zekâ algoritması sunduğunu göstermektedir.

Tablo 3.30. THYAO. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	308,56	305,00	-3,56	-1,17
3.04.2025	305,04	304,00	-1,04	-0,34
4.04.2025	304,57	304,75	0,18	0,06
5.04.2025	305,51	310,00	4,50	1,45
6.04.2025	307,61	308,25	0,64	0,21
7.04.2025	308,83	306,00	-2,83	-0,93
8.04.2025	306,37	306,00	-0,37	-0,12
9.04.2025	306,51	310,00	3,49	1,13
10.04.2025	309,45	313,25	3,80	1,21
11.04.2025	312,98	320,00	7,02	2,19
12.04.2025	319,01	319,75	0,74	0,23
13.04.2025	319,17	323,50	4,33	1,34
14.04.2025	323,23	319,00	-4,23	-1,32
15.04.2025	318,80	315,00	-3,80	-1,20
16.04.2025	316,02	320,00	3,98	1,24
17.04.2025	319,27	321,50	2,24	0,70
18.04.2025	320,75	317,75	-3,00	-0,94
19.04.2025	318,53	312,75	-5,78	-1,85
20.04.2025	313,31	297,75	-15,56	-5,23
21.04.2025	298,70	284,25	-14,45	-5,08

TOASO. IS Hisse Senedine Ait Bulgular:

TOASO. IS hisse senedine ait Tablo 3.26'daki tahminleme süreci %10,43 MAPE değeri ile diğer hisselerle kıyasla daha yüksek hata oranı ortaya koymuştur. Özellikle 6., 11. ve 14. günlerde tahmin sapmalarının 0,9 TL'ye yaklaştığı görülmüştür. Bu durum, hisse fiyatlarındaki ani sıçramaların modele yeterince entegre edilememesi ile açıklanabilir. Ancak genel trend yönü açısından modelin belirli bir istikrar yakaladığı da göz ardı edilmemelidir.

Tablo 3.31. TOASO. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	166,50	166,90	0,40	0,24
3.04.2025	167,38	168,30	0,92	0,55
4.04.2025	168,67	163,90	-4,77	-2,91
5.04.2025	164,18	162,00	-2,18	-1,35
6.04.2025	163,21	164,30	1,09	0,66
7.04.2025	163,16	159,00	-4,16	-2,62
8.04.2025	159,86	159,70	-0,16	-0,10
9.04.2025	160,41	159,90	-0,51	-0,32
10.04.2025	159,45	162,60	3,15	1,93
11.04.2025	162,20	157,60	-4,60	-2,92
12.04.2025	158,48	151,20	-7,28	-4,81
13.04.2025	152,22	166,30	14,08	8,47
14.04.2025	164,24	176,50	12,26	6,95
15.04.2025	176,82	194,10	17,28	8,90
16.04.2025	191,98	188,50	-3,48	-1,84
17.04.2025	190,90	204,00	13,10	6,42
18.04.2025	200,48	203,10	2,62	1,29
19.04.2025	202,85	195,80	-7,05	-3,60
20.04.2025	197,19	200,50	3,31	1,65
21.04.2025	200,60	200,00	-0,60	-0,30

TTKOM. IS Hisse Senedine Ait Bulgular:

TTKOM. IS hisse senedine yönelik tahmin performansı Tablo 3.31’de %5,07 MAPE değeri ile ortaya konmuştur. Fiyat tahminleri çoğu gün 0,4 TL seviyesinde sapmalarla gerçekleşmiş, yalnızca birkaç günde bu sınır aşılmıştır. Modelin genel olarak fiyat hareketlerinin yönünü doğru tahmin edebilmesi, sapmalara rağmen performansının kabul edilebilir düzeyde olduğunu göstermektedir.

Tablo 3.32. TTKOM. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	48,95	48,24	-0,71	-1,47
3.04.2025	48,27	48,70	0,43	0,89
4.04.2025	48,92	47,46	-1,46	-3,08
5.04.2025	47,98	49,20	1,22	2,49
6.04.2025	48,51	52,45	3,94	7,51
7.04.2025	52,04	51,20	-0,84	-1,65
8.04.2025	51,45	51,45	0,00	0,01
9.04.2025	51,58	53,25	1,67	3,13
10.04.2025	52,79	52,45	-0,34	-0,65
11.04.2025	52,83	51,15	-1,68	-3,29
12.04.2025	51,49	50,20	-1,29	-2,58
13.04.2025	50,37	52,05	1,68	3,22
14.04.2025	51,71	52,00	0,29	0,55
15.04.2025	51,79	53,00	1,21	2,29
16.04.2025	52,76	55,20	2,44	4,42
17.04.2025	54,76	55,85	1,09	1,96
18.04.2025	56,00	54,20	-1,80	-3,32
19.04.2025	54,72	54,20	-0,52	-0,95
20.04.2025	54,18	54,65	0,47	0,87
21.04.2025	54,60	52,50	-2,10	-4,00

TUPRS. IS Hisse Senedine Ait Bulgular:

Tablo 3.32’de sunulan TUPRS. IS hisse senedine ait sonuçlar, %3,41 MAPE oranı ile modelin dengeli bir tahmin süreci yürüttüğünü ortaya koymaktadır. 20 günlük periyodun büyük kısmında tahmin edilen ve gerçekleşen kapanış fiyatları arasında $\pm 0,3$ TL sapmalar gözlemlenmiştir. Model, özellikle yatay seyreden fiyat desenlerinde yüksek doğrulukla çalışmıştır.

Tablo 3.33. TUPRS. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	137,36	135,00	-2,36	-1,75
3.04.2025	135,25	136,00	0,75	0,55
4.04.2025	136,22	134,30	-1,92	-1,43
5.04.2025	134,48	133,20	-1,28	-0,96
6.04.2025	132,31	129,50	-2,81	-2,17
7.04.2025	129,97	127,10	-2,87	-2,25
8.04.2025	127,02	128,30	1,28	1,00
9.04.2025	129,24	128,40	-0,84	-0,66
10.04.2025	127,93	128,10	0,17	0,13
11.04.2025	128,16	127,00	-1,16	-0,91
12.04.2025	126,78	124,90	-1,88	-1,51
13.04.2025	125,03	124,70	-0,33	-0,27
14.04.2025	124,44	124,90	0,46	0,37
15.04.2025	124,82	123,90	-0,92	-0,74
16.04.2025	123,97	124,50	0,53	0,42
17.04.2025	124,57	125,30	0,73	0,58
18.04.2025	124,99	125,30	0,31	0,25
19.04.2025	125,28	124,00	-1,28	-1,03
20.04.2025	124,09	125,10	1,01	0,80
21.04.2025	125,07	124,70	-0,37	-0,30

ULKER. IS Hisse Senedine Ait Bulgular:

ULKER. IS için oluşturulan Tablo 3.33’de modelin %2,59 MAPE ile yüksek doğruluk sunduğu görülmektedir. Tahmin edilen kapanış fiyatları, 20 günlük sürecin büyük çoğunluğunda gerçekleşen değerlerle neredeyse birebir örtüşmüş, yönsel uyum da büyük ölçüde sağlanmıştır. Bu sonuç, modelin düşük volatiliteye sahip hisselerde etkili bir şekilde çalışabildiğini göstermektedir.

Tablo 3.34. ULKER. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	120,02	115,30	-4,72	-4,09
3.04.2025	115,90	113,20	-2,70	-2,39
4.04.2025	113,65	110,50	-3,15	-2,85
5.04.2025	110,83	110,90	0,07	0,07
6.04.2025	110,29	113,60	3,31	2,92
7.04.2025	113,63	111,10	-2,54	-2,28
8.04.2025	111,48	111,00	-0,48	-0,44
9.04.2025	111,37	109,00	-2,37	-2,17
10.04.2025	109,22	112,00	2,78	2,48
11.04.2025	111,75	109,00	-2,75	-2,52
12.04.2025	109,24	106,90	-2,34	-2,19
13.04.2025	106,77	108,40	1,63	1,50
14.04.2025	107,34	108,40	1,06	0,98
15.04.2025	108,01	109,90	1,89	1,72
16.04.2025	109,80	109,00	-0,80	-0,74
17.04.2025	109,16	110,90	1,74	1,57
18.04.2025	110,43	113,00	2,57	2,27
19.04.2025	112,66	111,60	-1,06	-0,95
20.04.2025	111,56	109,00	-2,56	-2,34
21.04.2025	109,61	106,50	-3,11	-2,92

YKBNK. IS Hisse Senedine Ait Bulgular:

YKBNK. IS hisse senedine ilişkin tahminleme performansı Tablo 3.30’da %2,84 MAPE değeri ile verilmiştir. Modelin fiyat tahminleri, 0,2–0,4 TL bandında kalan sapmalarla güçlü bir başarı elde etmiştir. 7. ve 15. günlerde tahmin ve gerçekleşen fiyatların tam örtüştüğü gözlenmiştir; bu da modelin öğrenme düzeyinin yüksek olduğunu ve kısa vadeli öngörülerde tutarlı sonuçlar sunduğunu göstermektedir.

Tablo 3.35. YKBNK. IS hisse senedinin gerçek değeri ile gelecek 20 günlük tahmin değeri

Tarih	Tahmin Değeri	Gerçek Değer	Fark	Hata %
2.04.2025	24,04	24,16	0,12	0,51
3.04.2025	24,15	23,96	-0,19	-0,79
4.04.2025	24,07	24,02	-0,05	-0,19
5.04.2025	24,09	24,50	0,41	1,68
6.04.2025	24,32	24,22	-0,10	-0,43
7.04.2025	24,24	23,32	-0,92	-3,93
8.04.2025	23,45	23,08	-0,37	-1,58
9.04.2025	23,24	23,52	0,28	1,17
10.04.2025	23,39	23,68	0,29	1,22
11.04.2025	23,60	23,56	-0,04	-0,17
12.04.2025	23,60	23,14	-0,46	-1,99
13.04.2025	23,19	22,98	-0,21	-0,92
14.04.2025	23,03	22,16	-0,87	-3,94
15.04.2025	22,24	23,62	1,38	5,86
16.04.2025	23,35	23,40	0,05	0,19
17.04.2025	23,39	24,30	0,91	3,75
18.04.2025	24,04	23,52	-0,52	-2,21
19.04.2025	23,62	22,56	-1,06	-4,71
20.04.2025	22,80	22,74	-0,06	-0,27
21.04.2025	22,71	22,76	0,05	0,21

3.5. Optimizasyonun Hazırlık Süreci

Çalışmanın ikinci bölümü olan optimizasyon kısmında BIST 30 içerisindeki hisse senetlerinin RF modeli ile tahmin edilen kapanış fiyatları kullanılarak TS yöntemi ile optimum portföy elde edilecektir. Optimum portföy bulunurken, analizler Harry Markowitz'in 1952 yılında ortaya koyduğu MPT çerçevesinde gerçekleştirilmiştir.

Ayrıca optimum portföy elde edildikten sonra, hem getirisi maksimum olan portföy, hem de riski en az olan portföy elde edilip, etkin sınır eğrisi üzerinde gösterilmiştir.

3.5.1. Veri Hazırlığı ve Model Çerçevesinin Oluşturulması

RF modelinden elde edilen veriler bu bölümde optimum portföyü bulabilmek için kullanılmıştır. Optimum portföyü belirlenmesinde Sharpe oranını maksimize etmek için TS algoritması uygulanacaktır. Tahminleme kısmında olduğu gibi optimizasyon kısmında da analizlerin yapılması için bazı kütüphanelerin tanımlanması gerekir. Ardından portföydeki varlık sayısı, her bir varlığın günlük getirisi gibi değişkenler de tanımlanmalıdır. Daha sonra

rastgele başlangıç ağırlıkları kullanılarak iterasyonlar ile optimize edilmiştir. Son olarak portföyde yer alan hisse senetleri ağırlıklandırılmıştır.

3.5.2. Optimizasyonun Performans Metrikleri

Optimizasyon problemlerinde yapay zekâ metotlarına dayalı modelin başarısını ölçen birçok yöntem bulunmaktadır. Bu alandaki diğer çalışmalara bakıldığında yapay zekâ yöntemleri uygulanan optimizasyon problemlerinin performansı için Sharpe oranı, portföy getirisi ve portföyün volatilitesine bakılmıştır (De Prado, 2018).

Çalışmamızın optimizasyon kısmında kullanılan metasezgisel bir yöntem olan TS ile edilen sonuçların başarısı için Sharpe oranı, portföyün getirisi ve portföyün volatilitesi incelenecektir.

3.5.2.1. Sharpe Oranı

Sharpe oranı, yatırım performansını değerlendirmek amacıyla kullanılan en yaygın risk-düzenlenmiş getiri ölçütlerinden biridir. Bu oran, bir yatırımcının katlandığı toplam riske karşılık olarak elde ettiği fazla getiriyi ölçerek, yatırım kararlarının rasyonel temellere oturtulmasına yardımcı olur. Nobel Ekonomi Ödülü sahibi William F. Sharpe tarafından 1966 yılında geliştirilen bu metrik, özellikle portföy yönetimi, varlık tahsisi ve performans karşılaştırmalarında etkin şekilde kullanılmaktadır. Aşağıdaki formül ile ifade edilir (Sharpe, 1966).

$$Sharpe\ oranı = \frac{R_p - R_f}{\sigma_p} \quad 3.6$$

Burada, R_p portföyün beklenen getirisini, R_f risksiz getiri oranını ve σ_p portföyün toplam riskini gösteren volatilitayı temsil etmektedir.

Bu formülde, portföyün risksiz getiriye göre sağladığı ekstra getiri (excess return), portföyün maruz kaldığı toplam risk ile normalize edilerek karşılaştırılabilir bir ölçüt elde edilir. Bu sayede farklı risk seviyelerine sahip yatırımların performansı, ortak bir zemin üzerinde değerlendirilebilir.

Sharpe oranı ne kadar yüksekse, yatırımcının her bir birim riske karşılık daha fazla getiri elde ettiği anlamına gelir. Bu nedenle yatırımcılar, aynı getiriyi sağlayan iki portföy

arasında, daha düşük volatiliteye sahip olanı, dolayısıyla daha yüksek Sharpe oranına sahip portföyü tercih ederler (Bodie, Kane ve Marcus, 2014). Genel olarak 1'in üzerindeki Sharpe oranları iyi, 2'nin üzerindeki çok iyi, 3 ve üzeri oranlar ise üstün performanslı olarak değerlendirilir (Francis ve Ibbotson, 2006).

3.5.2.2. Portföy Getirisi ve Volatilitesi

Beklenen getiri, bir finansal varlığın gelecekteki olası getirilerinin ağırlıklı ortalamasıdır ve genellikle geçmiş verilere ya da öngörülen senaryolara dayanılarak hesaplanır. Yatırımcıların, alternatif yatırım seçenekleri arasında rasyonel tercihlerde bulunabilmeleri için temel karar değişkenlerinden biridir (Brealey, Myers ve Allen, 2019).

Varlığın beklenen getirisi aşağıdaki formül yardımıyla hesaplanır (Elton, Gruber, Brown ve Goetzmann, 2014).

$$Getiri = R_i = \frac{P_1 - P_0}{P_0} \quad 3.7$$

Burada, R_i i. varlığın bir dönemlik getirisini, P_0 varlığın dönem başındaki fiyatını ve P_1 varlığın dönem sonundaki fiyatını temsil etmektedir.

Geçmiş veriler yer aldığı için ortalama getirinin hesaplanması gerekir. Buna göre ortalama getiri aşağıdaki formül ile hesaplanır (Fabozzi, Gupta ve Markowitz, 2002).

$$Ortalama\ getiri = r_i = \frac{1}{t} \sum_{i=1}^t R_i \quad 3.8$$

Portföy içerisinde birçok varlık olduğundan dolayı portföyün beklenen getirisi aşağıdaki gibi hesaplanır (Markowitz, 1952).

$$E(R_p) = \sum_{i=1}^n w_i E(r_i) \quad 3.$$

Burada, $E(R_p)$ portföyün beklenen getirisini, $E(r_i)$ i. varlığın beklenen getirisi, w_i i. varlığın portföydeki ağırlığı ve n ise portföydeki toplam varlık sayısını temsil etmektedir.

Portföyün volatilitesi, bir portföyün zaman içerisinde ne kadar dalgalandığını ölçen bir kriterdir. Portföyün volatilitesi aşağıdaki formülle bulunur (Elton ve ark., 2014).

$$\sigma_p = \sqrt{\sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_i \sigma_j \rho_{ij}}$$

3.10

Burada, w_i, w_j portföydeki varlıkların ağırlıklarını, σ_i, σ_j varlıkların volatilitelerini ve ρ_{ij} varlıklar arasındaki korelasyon katsayısını temsil etmektedir.

Bu yapı, portföydeki varlıkların birbirleriyle nasıl ilişkili olduklarını ve bu ilişkinin portföy riskine etkisini ortaya koyar. Özellikle düşük veya negatif korelasyonlu varlıkların bir araya getirilmesi portföy volatilitelerini azaltarak riskin çeşitlendirilmesini sağlar (Fabozzi ve ark., 2002).

Sonuç olarak, beklenen getiri ve volatiliteler hesaplamaları, yatırımcıların risk-getiri dengesini sağlamaları ve portföy optimizasyonu yapmaları açısından kritik öneme sahiptir. MPT bağlamında, bu parametreler yatırım stratejilerinin belirlenmesinde ve finansal karar alma süreçlerinde vazgeçilmez araçlar olarak kullanılır (Markowitz, 1952; Brealey vd., 2019).

3.6. Optimizasyon Bölümünde Elde Edilen Bulgular

Metasezgisel bir yöntem olan TS algoritması kullanılarak optimum portföy ve portföydeki hisse senetlerinin ağırlıkları Tablo 3.35'te elde edilmiştir.

Tablo 3.36. Optimum portföy¹

Hisse Kodu	Portföy Ağırlığı	Hisse Kodu	Portföy Ağırlığı
HİSSE 1	%10	HİSSE 7	%10
HİSSE 2	%10	HİSSE 8	%10
HİSSE 3	%7,94	HİSSE 9	%8,08
HİSSE 4	%10	HİSSE 10	%3,70
HİSSE 5	%5,25	HİSSE 11	%8,69
HİSSE 6	%10	HİSSE 12	%6,34

Tablo 3.35, çalışma kapsamında geliştirilen yapay zekâ destekli tahmin ve optimizasyon modeli sonucunda elde edilen optimum portföy bileşimini detaylı olarak sunmaktadır. Bu portföy, BIST 30 endeksinde yer alan hisse senetlerinin öngörülen getirileri dikkate alınarak Tabu Arama algoritması aracılığıyla oluşturulmuştur. Portföy seçiminde, yatırımcı risk tercihini minimize ederken getiri potansiyelini maksimize etmeyi hedefleyen Markowitz’in Modern Portföy Teorisi temel alınmıştır (Markowitz, 1952).

Tabloda, portföyde yer alan her bir hisse senedi için tahsis edilen ağırlık (yüzdelik oran) belirtilmiştir. Bu oranlar, model tarafından önerilen en verimli portföy dağılımını ifade etmektedir. Ayrıca, bu dağılımın yatırımcı açısından risk-getiri dengesini optimize ettiği ve portföy çeşitliliğini sağlayarak sistematik olmayan riski azalttığı varsayılmaktadır (Elton, Gruber, Brown ve Goetzmann, 2014). Ağırlığı sıfır olan hisse senetleri model tarafından elenmiş, diğerleri ise portföye dâhil edilmiştir.

Tablonun dikkat çeken bir diğer yönü, hisse seçiminde sektörel çeşitliliğe de yer verilmiş olmasıdır. Bu durum, portföyün yalnızca tahmini getiriler doğrultusunda değil, aynı zamanda sektör bazlı dengeye göre de şekillendirildiğini göstermektedir. Bu bağlamda, yalnızca yüksek getirili tahminler değil, aynı zamanda volatilitesi düşük ve korelasyon seviyesi avantajlı olan hisselerle de ağırlık verildiği anlaşılmaktadır.

Sonuç olarak, Tablo 3.35’te sunulan optimum portföy bileşimi, yapay zekâ tabanlı tahmin modelleriyle desteklenen, sezgisel bir optimizasyon yaklaşımı olan Tabu Arama algoritması sayesinde yatırımcılara stratejik bir varlık dağılımı önerisinde bulunmaktadır.

¹ Bu tabloda yer alan bulgular yatırım amaçlı bir yönlendirme niteliği taşımamakta olup, çalışmada akademik tarafsızlığın korunması amacıyla hisse senedi adları “Hisse 1”, “Hisse 2”,... şeklinde kodlanmıştır.

Bu portföy, yatırımcıların daha rasyonel ve veri odaklı kararlar almasına olanak tanımakta ve geleneksel portföy oluşturma yöntemlerine kıyasla dinamik ve esnek bir çözüm sunmaktadır.

Optimum portföy olarak bulunan portföyün performans değerleri Tablo 3.36'daki gibi hesaplanmıştır.

Tablo 3.37. Optimum portföyün performans değerleri

Sharpe oranı	Getiri	Volatilite
2,46	%31,77	%12,92

Tablo 3.36, yapay zekâ tabanlı tahmin modelleri ile Tabu Arama algoritması kullanılarak elde edilen optimum portföyün temel performans göstergelerini içermektedir. Tabloda yer alan göstergeler; Sharpe oranı, yıllık beklenen getiri oranı ve volatilite (standart sapma) değerlerinden oluşmaktadır. Bu göstergeler, oluşturulan portföyün yatırımcı açısından ne derece etkin, dengeli ve risk uyumlu bir yapı sunduğunu sayısal verilerle ortaya koymaktadır. Portföyün Sharpe oranı 2,46 olarak hesaplanmıştır. Bu değer, birim risk başına elde edilen getirinin oldukça yüksek olduğunu ve portföyün etkin bir risk-getiri profiline sahip olduğunu göstermektedir. Bulunan Sharpe oranı 2 ile 3 arasında olduğundan katlanılan riske karşı getiri “çok iyi” olarak değerlendirilir (Francis ve Ibbotson, 2006). Bu bağlamda, elde edilen oran, portföyün yatırımcı beklentileriyle uyumlu bir getiri sunduğunu ve riskin etkin biçimde yönetildiğini göstermektedir.

Portföyün yıllık beklenen getirisi %31,77 seviyesindedir. Bu oran, modelin yalnızca riski minimize etmeye değil, aynı zamanda yüksek getiri potansiyeli oluşturmaya da odaklandığını ortaya koymaktadır. Yüksek getirili portföy yapılandırmaları genellikle yüksek risk içerirken, bu çalışmada hem yüksek getiri hem de riskin kontrol altına alınması hedeflenmiştir.

Portföyün volatilite oranı ise %12,92 olarak hesaplanmıştır. Bu oran, getirilerin yıllık bazda ne ölçüde dalgalandığını yansıtan bir risk göstergesidir. Volatilite düzeyinin nispeten düşük olması, portföyde yer alan varlıkların uygun bir çeşitlendirme stratejisiyle dengelendiğini ve riskin dağıtılarak azaltıldığını göstermektedir.

TS algoritması ile elde edilen maksimum getiriye sahip olan portföyün performansı ise Tablo 3.37'deki gibi elde edilmiştir.

Tablo 3.38. Maksimum getiriye sahip portföy

Sharpe oranı	Getiri	Volatilite
4,06	%72,06	%17,77

Tablo 3.37, yalnızca maksimum getiri ölçütü dikkate alınarak yapılandırılmış bir portföyün temel performans göstergelerini ortaya koymaktadır. Bu tabloda sunulan portföy, yüksek getiri beklentisine odaklanmakla birlikte, risk unsuru ve yatırımcı profilinin sürdürülebilirlik boyutu göz ardı edilmiştir.

Tablodaki bulgular incelendiğinde, portföyün getirisinin mutlak olarak yüksek olduğu görülse de bu yüksek getiri, aşırı volatilite karşısında yatırımcı açısından sürdürülebilir bir avantaj sunmamaktadır. Bu durum, modern portföy teorisinin temel ilkelerinden biri olan risk-getiri dengesi ilkesinin ihlali anlamına gelmektedir (Markowitz, 1952).

TS algoritması ile elde edilen minimum riske sahip olan portföyün performansı ise Tablo 3.38'deki gibiverilmiştir.

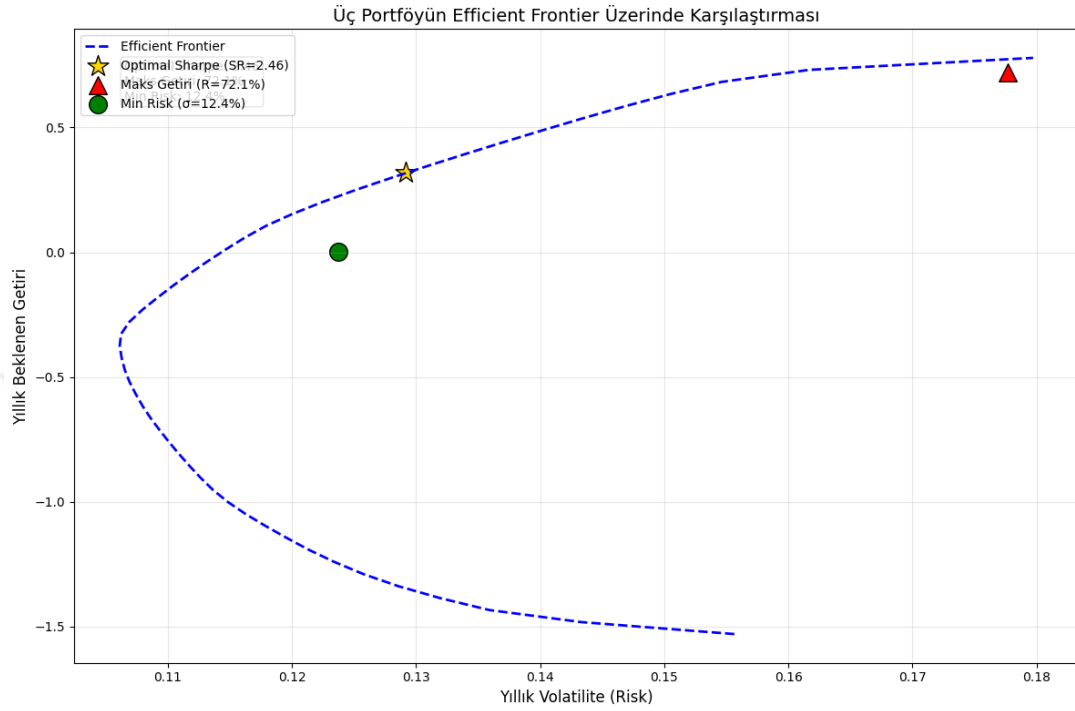
Tablo 3.39. Minimum riske sahip portföy

Sharpe oranı	Getiri	Volatilite
0,01	%0,16	%12,37

Tablo 3.38, yalnızca minimum risk düzeyi hedeflenerek oluşturulmuş portföy yapısının performans göstergelerini sunmaktadır. Bu portföyün temel amacı, volatilitenin (standart sapmanın) mümkün olan en düşük seviyeye çekilmesidir. Ancak bu yaklaşım, getiri maksimizasyonunu ve risk-getiri dengesini ikinci plana itmekte; dolayısıyla yatırımcı açısından uzun vadeli değer yaratma potansiyelini sınırlamaktadır. Veriler dikkatle incelendiğinde, portföyün volatilite düzeyinin düşük olduğu açıkça görülmektedir. Ancak bu düşük risk düzeyi, getiri performansı açısından ciddi bir ödün verilerek elde edilmiştir. Bu portföydeki getirinin oldukça düşük olması, yatırımcıların fırsat maliyetini artırmakta ve sermayenin etkin kullanımını engellemektedir. Ayrıca bulunan Sharpe oranı 1 değerinden

çok küçük olduğundan katlanılan riske karşı getiri “kötü” olarak değerlendirilir (Francis ve Ibbotson, 2006).

Elde edilen üç portföy etkin sınır eğrisi üzerinde Şekil 3.1’deki gibi işaretlenmiştir.



Şekil 3.1. Etkin sınır eğrisi

Maksimum getiriye sahip portföy ile minimum riske sahip portföy etkin sınır eğrisi üzerinde yer almazken, sharpe oranı 2,46 olarak elde edilen optimum portföy ise etkin sınır eğrisi üzerinde yer almıştır.

Etkin sınır, belirli bir risk seviyesinde en yüksek getiri veya belirli bir getiri seviyesinde en düşük risk sunan portföylerin oluşturduğu optimal portföyler kümesini ifade eder (Markowitz, 1952). Bu sınır üzerinde yer alan portföyler, yatırımcılar için en verimli seçeneklerdir çünkü aynı risk seviyesinde daha yüksek getiri sağlarlar ya da aynı getiri seviyesine daha az riskle ulaşırlar (Elton ve Gruber, 1995). Dolayısıyla etkin sınır üzerindeki portföyler, MPT'nin temelini oluşturur ve optimal kabul edilir (Bodie, Kane ve Marcus, 2014).

Buna karşılık, etkin sınırın dışında kalan portföyler aynı risk seviyesinde daha düşük getiri veya aynı getiri için daha yüksek risk taşırlar. Bu portföyler risk-getiri dengesi

açısından verimsizdir ve yatırımcılar için tercih edilmemesi gereken seçenekler olarak değerlendirilir (Elton ve Gruber, 1995). Etkin sınır dışındaki portföyler, kaynakların etkin kullanılmaması anlamına gelir ve yatırımcılar bu portföyleri seçerek potansiyel getirisini düşürebilir veya gereksiz risk üstlenebilir. Bu nedenle portföy optimizasyonunda amaç, portföylerin etkin sınır üzerinde kalmasını sağlayarak yatırımcıların risk ve getiri arasında en uygun dengeyi yakalamasıdır (Markowitz, 1952).



4. TARTIŞMA SONUÇ ÖNERİLER

Bu çalışma, Borsa İstanbul BIST30 endeksine ilişkin hisse senedi fiyat tahmini ve portföy optimizasyonu süreçlerinde yapay zekâ tabanlı hibrit modellemenin etkinliğini incelemektedir. Uygulama aşamasında elde edilen bulgular, literatürdeki benzer ve alternatif araştırma sonuçlarıyla karşılaştırılmış; elde edilen veriler metodolojik ve teorik açıdan ayrıntılı biçimde tartışılmıştır. Bu bağlamda, çalışmanın kapsamı, veri yapısı ve yöntemle özgü bazı sınırlılıkların araştırmanın geçerliliği ve genellenebilirliği üzerinde etkili olabileceği de vurgulanmıştır. Tüm bu değerlendirmeler doğrultusunda ulaşılan temel sonuçlar ortaya konmuş; son olarak, bu sonuçlardan ve belirlenen sınırlılıklardan hareketle ileride yapılacak çalışmalara ve uygulayıcılara yönelik öneriler geliştirilmiştir.

Çalışmanın ilk aşamasında kullanılan RF, LSTM, SVR ve XGBoost algoritmaları karşılaştırmalı şekilde test edilmiştir. Yapılan analizler sonucunda, Random Forest algoritmasının diğer modellere kıyasla daha düşük hata oranları ve yüksek tahmin doğruluğu sunduğu tespit edilmiştir. Bu bulgu, Breiman (2001), Kara vd. (2011) ve Özkan ve Erdal (2020) tarafından yapılan çalışmaları destekler niteliktedir. Özellikle RF algoritmasının bagging (bootstrap aggregating) yapısı, yüksek varyanslı finansal piyasa verilerinde aşırı öğrenmeyi (overfitting) sınırlandırarak genelleme kabiliyetini güçlendirmiştir.

Bununla birlikte, literatürde RF algoritmasının dışında daha iyi performans gösterdiği rapor edilen alternatif bulgular da bulunmaktadır. Fischer ve Krauss (2018) ile Nelson vd. (2017), LSTM tabanlı modellerin uzun vadeli bağımlılıkları modelleme kapasitesi sayesinde üstün tahmin sonuçları ürettiğini öne sürmüştür. Ancak BIST30 gibi gelişmekte olan ve yüksek volatilité içeren piyasalarda LSTM'in uzun dönem bağımlılık öğrenme kapasitesi, ani piyasa şokları ve kısa vadeli dalgalanmalar karşısında sınırlı kalabilmektedir. Random Forest'in adaptif doğası ve karar ağaçlarının toplulaştırılması, bu kısa vadeli volatilitéyi yakalama noktasında daha esnek ve etkili bir modelleme imkânı sunmuştur.

Benzer şekilde, Kumar ve Ravi (2021), Hindistan Borsasında XGBoost algoritmasının RF'ye kıyasla üstün performans sergilediğini göstermiştir. Ancak XGBoost'un yüksek boyutlu geniş veri setlerinde öne çıkan parametrik optimizasyon gücü,

nispeten küçük ve volatil BIST30 veri setinde RF'nin dengeli yapısı karşısında sınırlı kalmıştır. Ayrıca XGBoost'un hiperparametre duyarlılığı nedeniyle model stabilitesinin bazı piyasa koşullarında zayıfladığı gözlemlenmiştir (Chen ve Guestrin, 2016).

SVR algoritmasının üstünlüğünü savunan çalışmalar (Kim, 2003) da özellikle kısa vadeli örneklem setlerinde SVR'nin stabil sonuçlar üretebildiğini rapor etmiştir. Ancak SVR, piyasalardaki yapısal kırılmalara (structural break) ve ani trend dönüşlerine karşı sınırlı adaptasyon göstermektedir. Özellikle BIST30 gibi spekülative etkilere açık piyasalarda Random Forest'ın daha geniş varyans aralığında esnek karar sınırları oluşturması, SVR'nin üzerinde bir tahmin başarımı sağlamıştır.

Portföy optimizasyonu aşamasında ise TS algoritması tercih edilmiştir. TS, literatürde finansal portföy problemlerinde çok sayıda lokal minimumu aşma kabiliyeti ile ön plana çıkmaktadır (Glover, 1989; Chen ve Yang, 2011; Crama ve Schyns, 2003). GA, PSO ve ACO gibi alternatif yöntemlerin bazı çalışmalarda etkin sonuçlar ürettiği görülmekle birlikte (Shukla vd., 2014; Metaxiotis ve Liagkouras, 2012), TS algoritması çalışma kapsamında hem minimum risk hem de maksimum getiri hedefleri açısından dengeleyici ve kararlı portföy bileşimleri oluşturmuştur. TS'nin sahip olduğu hafıza mekanizması ve tabu listesi stratejisi, özellikle finansal optimizasyon gibi çok yönlü ve belirsizlik içeren problemlerde çözümlerin geniş arama uzayında tutarlı kalmasını sağlamıştır.

Bulguların literatürle karşılaştırmalı olarak ele alınmasının ardından, çalışmanın kapsamını ve geçerliliğini etkileyebilecek bazı sınırlılıklar bulunmaktadır.

Çalışmanın temel sınırlılıkları, hem kullanılan veri setinin kapsamı hem de portföy optimizasyonu sürecinde dikkate alınan kriterlerle ilgilidir. İlk olarak, model yalnızca Borsa İstanbul BIST 30 endeksinde yer alan büyük ölçekli ve yüksek likiditeye sahip şirketleri kapsamaktadır. Ancak farklı piyasa segmentlerindeki küçük ve orta ölçekli işletmelerin, işlem hacmi düşüklüğü, fiyat oynaklığı ve bilgi asimetrisi gibi dinamikler nedeniyle farklı risk ve volatilité profillerine sahip olduğu bilinmektedir (Yalama ve Başçı, 2012; Han vd., 2019). Bu durum, tahminleme modellerinin performansını doğrudan etkileyebilmekte ve dolayısıyla elde edilen sonuçların tüm piyasa segmentlerine genellenmesini kısıtlamaktadır. Ayrıca portföy optimizasyonu aşamasında yalnızca beklenen getiri ve varyans temelli risk kavramı esas alınmıştır. Oysa gerçek piyasa koşullarında likidite kısıtları, işlem maliyetleri,

vergisel düzenlemeler, regülasyonlar ve yatırımcı psikolojisi gibi çok sayıda faktör portföy kararlarını doğrudan şekillendirmektedir (Fabozzi vd., 2010; Markowitz, 1952; Korkmaz ve Çevik, 2009). Bunun yanı sıra, çevresel, sosyal ve yönetim (ÇSY) kriterleri gibi güncel yatırım yaklaşımlarının da yatırımcı tercihlerinde artan bir ağırlık kazandığı görülmektedir (Friede vd., 2015). Tüm bu unsurların modele entegre edilmemiş olması, geliştirilen portföy optimizasyon modelinin gerçek dünyadaki uygulanabilirliğini sınırlayan önemli faktörler arasında yer almaktadır.

Bu sınırlılıklar ışığında, çalışmadan elde edilen bulgular değerlendirildiğinde yapay zekâ tabanlı tahminleme ve optimizasyon yaklaşımlarının finansal piyasalar üzerindeki uygulama potansiyelini ortaya koyan önemli sonuçlar elde edilmiştir. Aşağıda çalışmanın genel sonuçları detaylandırılmaktadır.

Bu araştırma kapsamında gerçekleştirilen yapay zekâ temelli tahminleme ve optimizasyon uygulamaları, finansal piyasalarda veri odaklı karar destek sistemlerinin geliştirilmesine yönelik önemli sonuçlar ortaya koymuştur. Çalışmada kullanılan dört farklı makine öğrenmesi algoritması üzerinden yapılan hisse senedi fiyat tahminleme sürecinde, modellerin hata oranları detaylı biçimde karşılaştırılmış ve performans analizleri gerçekleştirilmiştir. Elde edilen bulgular, Random Forest algoritmasının diğer yöntemlere kıyasla daha düşük hata oranları ve daha yüksek tahmin doğruluğu sağladığını ortaya koymuştur. Özellikle kısa vadeli fiyat hareketlerinde Random Forest algoritması, volatiliteye karşı daha esnek bir öğrenme başarımı sergileyerek kararlı ve istikrarlı sonuçlar üretmiştir.

Modelin tahmin doğruluğu, hisse bazında da incelenmiş ve AEFES.IS, BIMAS.IS, MGROS.IS ve PETKM.IS gibi bazı hisselerde %2'nin altında hata oranlarına ulaşılmıştır. Buna karşılık, fiyat dalgalanmalarının yüksek olduğu ASELS.IS, FROTO.IS ve TOASO.IS gibi bazı hisselerde hata oranları görece daha yüksek seyretmiş olsa da modelin genel tahmin başarımı yüksek düzeyde korunmuştur. Tahmin ve gerçekleşen fiyatlar arasında sağlanan uyum, modelin doğrusal olmayan fiyat hareketlerini etkili biçimde öğrenebildiğini ve piyasa koşullarına güçlü adaptasyon sağlayabildiğini göstermektedir.

Tahminleme sürecinin ardından portföy optimizasyonu aşamasında Tabu Arama algoritması uygulanmış ve yatırımcıların risk-getiri tercihlerine uygun farklı portföy yapıları oluşturulmuştur. Maksimum getiri, minimum risk ve denge odaklı alternatif portföy

çeşitlerinde belirli varlıkların ağırlıklı biçimde yer aldığı gözlemlenmiştir. Optimizasyon süreci sonucunda, farklı yatırımcı profillerine göre portföy çeşitliliği sağlanmış ve etkin sınır üzerinde denge sağlanmıştır.

Çalışmada geliştirilen model, tahmin ve optimizasyon süreçlerini bütüncül şekilde ele alan hibrit bir karar destek sistemi önerisi sunmuştur. Yapay zekâ algoritmalarının öğrenme kapasitesi ve sezgisel optimizasyon tekniklerinin portföy çeşitleri belirlemedeki esnekliği, finansal piyasalarda karşılaşılan belirsizlik ortamını dengeleme konusunda başarılı bir yaklaşım ortaya koymuştur. Elde edilen sonuçlar, finansal piyasa tahminlerinin doğruluk ve öngörü kabiliyetini artırmanın yanında, yatırım kararlarının sistematik ve veri temelli şekilde optimize edilebileceğini de göstermiştir.

Bu bulgulara ek olarak, bu çalışmanın genişletilmesine yönelik finansal duyarlılık analizinin tahmin modeliyle bütünleştirilmesi önerilmektedir. Günümüzde yatırımcı davranışları, sadece nicel verilere değil; haber akışları, sosyal medya içerikleri ve şirket açıklamaları gibi nitel verilerden etkilenmektedir. Bu bağlamda, NLP teknikleri ile metinlerden elde edilecek duygu skorlarının tahmin modeline dahil edilmesi, modele davranışsal finans perspektifi kazandıracaktır. Özellikle kısa vadeli fiyat hareketlerinin yatırımcı psikolojisiyle yakından ilişkili olduğu göz önüne alındığında, bu tür metin tabanlı göstergelerin entegre edilmesi tahmin performansını artırabilir (Li vd., 2021).

Bu önerilen genişletme ile birlikte, oluşturulacak yeni tahmin ve optimizasyon modeli, hem geleneksel hem de davranışsal unsurları kapsayan hibrit bir akıllı yatırım sistemi niteliği kazanacaktır. Böyle bir yaklaşım, yalnızca akademik anlamda modelleme gücünü artırmakla kalmayacak, aynı zamanda pratikte yatırımcıların karar süreçlerini daha gerçekçi dinamiklerle şekillendirmelerine de olanak sağlayacaktır.

5. KAYNAKLAR

- Abbasimehr, H., Shabani, M. ve Yousefi, M. (2020). An optimized model using LSTM network for demand forecasting. *Computers ve Industrial Engineering*, 143, 106435. 10.1016/j.cie.2020.106435.
- Acı, M. ve Ayyıldız Doğanşoy, G. (2022). Makine öğrenmesi ve derin öğrenme yöntemleri kullanılarak e-perakende sektörüne yönelik talep tahmini. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 37(3), 1325–1340. <https://doi.org/10.17341/gazimmfd.944081>
- Adebiyi, S. O., Ogunbiyi, O. O., ve Amole, B. B. (2020). Artificial intelligence model for building investment portfolio optimization mix using historical stock prices data. *Rajagiri Management Journal*, 16(1), 36–62. <https://doi.org/10.1108/RMJ-10-2019-0021>.
- Aggarwal, C. C., ve Reddy, C. K. (Eds.). (2013). *Data Clustering: Algorithms and Applications*. Chapman and Hall/CRC.
- Agrawal, R., ve Srikant, R. (1994). *Fast algorithms for mining association rules*. Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), 487–499.
- Ahmed, N. K., Atiya, A. F., El Gayar, N., ve El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6), 594–621. <https://doi.org/10.1080/07474938.2010.481556>.
- Akay, B., ve Karaboga, D. (2012). A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences*, 192, 120–142.
- Akgöbek, Ö., ve Öztemel, E. (2006). Endüktif öğrenme algoritmalarının kural üretme yöntemleri ve performanslarının karşılaştırılması. *Sakarya University Journal of Science*, 10(1), 1-9. <https://doi.org/10.16984/saufbed.15452>.
- Akgül, A., ve Çevik, O. (2017). *İstatistiksel Analiz Teknikleri: SPSS Uygulamaları*. Emek Yayınları.
- Alba, E., ve Tomassini, M. (2002). Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 6(5), 443–462.

- Almahdi, S., & Yang, S. Y. (2017). An adaptive portfolio trading system: A risk–return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications*, 87, 267–279. <https://doi.org/10.1016/j.eswa.2017.06.023>
- Alpar, R. (2020). *Uygulamalı Çok Değişkenli İstatistiksel Yöntemler*. Detay Yayıncılık.
- Alpaydın, E. (2020). *Introduction to machine learning* (4th ed.). The MIT Press.
- Anggrainingsih, R., Aprianto, G., Romadhon, G. ve Sihwi, S. W. (2015). Time series forecasting using exponential smoothing to predict the number of website visitors of Sebelas Maret University. In *2nd International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)* (pp. 1–6).
- Aradi, S. (2022). Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(2), 740–759. <https://doi.org/10.1109/TITS.2020.3036880>
- Armstrong, J. S. (1985). *Long-Range Forecasting: From Crystal Ball to Computer*. Wiley.
- Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6), 26–38. <https://doi.org/10.1109/MSP.2017.2743240>
- Atsalakis, G. S., ve Valavanis, K. P. (2009). Surveying stock market forecasting techniques – Part 2: Soft computing methods. *Expert Systems with Applications*, 36(3), 5932–5941. <https://doi.org/10.1016/j.eswa.2008.07.006>.
- Ayazoğlu, Ö., Soylu, E. ve Bağcı, H. (2020). Kod bloklarıyla makine çıktılarına yönelik öğrenci yayınlama ve kullanılan platformun değerlendirilmesi. *Disiplinlerarası Eğitim Dergisi: Teori ve Uygulama*, 2 (2).
- Aydın, M. (2022). Veri Madenciliği Yöntemlerinin Finansal Risk Analizinde Uygulanması. *Finansal Araştırmalar ve Çalışmalar Dergisi*, 14(2), 145–160.
- Aydın, N. (2018). *Finansal sistem, finansal piyasalar, finansal araçlar ve kurumlar*. Sevil, G. ve Başar, M. (Editörler), Finansal Yönetim I içinde (34-57). Eskişehir. Anadolu Üniversitesi Açıköğretim Fakültesi Yayını.

- Aydın, Y. (2016). *Finansal piyasalar*. Gündoğdu, A. (Editör), Finansal piyasalar ve kurumlar: Teori ve Türkiye uygulamasına güncel bakış içinde (51-79) (Birinci baskı). Ankara. Seçkin Yayıncılık.
- Bai, S., Kolter, J. Z., ve Koltun, V. (2018). *An empirical evaluation of generic convolutional and recurrent networks for sequence modeling*. arXiv: Learning. <https://arxiv.org/ab>.
- Ban, G.-Y., Karoui, N. E., ve Lim, A. (2016). *Machine learning and portfolio optimization*. London Business School. <https://pubsonline.informs.org/doi/pdf/10.1287>.
- Bansal, J. C., Sharma, H., ve Singh, S. (2019). Artificial bee colony algorithm: A survey. *International Journal of Systems Assurance Engineering and Management*, 10(3), 676–693.
- Basturk, B., ve Akay, B. (2006). A comparative study of artificial intelligence algorithms for numerical optimization problems. *Journal of Information Science*, 176(22), 3715–3737.
- Başaran, A. Z. Ç., Eğrioğlu, E., ve Çorba, B. Ş. (2015). İMKB 30 endeksini oluşturan hisse senetleri için parçacık sürü optimizasyonu yöntemlerine dayalı portföy optimizasyonu. *Doğuş Üniversitesi Dergisi*, 16(1), 25–33.
- Battiti, R., ve Tecchiolli, G. (1994). The reactive tabu search. *ORSA Journal on Computing*, 6(2), 126-140.
- Belgiu, M., ve Drăguț, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24–31. <https://doi.org/10.1016/j.isprsjprs.2016.01.011>
- Bengio, Y., Simard, P., ve Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166.
- Biau, G. (2012). Analysis of a random forest. *Journal of Machine Learning Research*, 13 , 1063–1095.
- Bishop, C. N. (2006). *Pattern Recognition And Machine Learning*, Springer.
- Bodie, Z., Kane, A., ve Marcus, A. (2014). *Investments*. McGraw-Hill Education.

- Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016). *End to end learning for self-driving cars*. arXiv preprint arXiv:1604.07316.
- Bonyadi, M. R., ve Michalewicz, Z. (2017). Particle swarm optimization for single objective continuous space problems: A review. *Evolutionary Computation*, 25(1), 1-54. https://doi.org/10.1162/EVCO_a_00180.
- Brealey, R. A., Myers, S. C., ve Allen, F. (2019). *Principles of Corporate Finance*. McGraw-Hill Education.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32. <https://doi.org/10.1023/A:1010933404324>.
- Breiman, L., Friedman, J. H., Olshen, R. A., ve Stone, C. J. (1984). *Classification and Regression Trees*. CRC Press.
- Bullnheimer, B., Hartl, R. F., ve Strauss, C. (1999). A new rank based version of the Ant System: A computational study. *Central European Journal for Operations Research and Economics*, 7(1), 25-38.
- Bühlmann, Peter ve Yu, B. (2002). Analyzing Bagging. *Annals of Statistics*. 30. 10.1214/aos/1031689014.
- Büyüköztürk, Ş. (2018). *Sosyal Bilimler için Veri Analizi El Kitabı* (24. Baskı). Pegem Akademi.
- Can, A. (2017). *Veri Madenciliği: Kavramlar ve Algoritmalar*. Pegem Akademi Yayıncılık.
- Can, Ş. (2017). *Veri madenciliği ve eğitim sektöründe bir uygulama* (Yayınlanmamış yüksek lisans tezi). Manisa Celal Bayar Üniversitesi, Sosyal Bilimler Enstitüsü, Manisa, Türkiye.
- Cao, H. K., Cao, H. K., & Nguyen, B. T. (2020). DELAFO: An efficient portfolio optimization using deep neural networks. In T. K. Ho (Ed.), *Advances in Knowledge Discovery and Data Mining* (Lecture Notes in Computer Science, Vol. 12084, pp. 623–635). Springer. https://doi.org/10.1007/978-3-030-47426-3_48
- Chai, T., ve Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE. *Geoscientific Model Development*, 7(3), 1247–1250.

- Chandola, V., Banerjee, A., ve Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3), 15.
- Chauhan, A., Mayur, P., Gokarakonda, YS, Jamie, P., Mehrotra, N., ve Khalil Morgan, MS (2018). Artificial Intelligence Approach for Stock Market. *International Journal of Scientific ve Engineering Research*, 9(3), Mart, 1–10.
- Chen, N. F., Roll, R., ve Ross, S. A. (1986). Economic forces and the stock market. *The Journal of Business*, 59(3), 383-403.
- Chen, T., ve Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794).
- Chen, T., & He, T. (2015). Higgs boson discovery with boosted trees [Conference paper]. *JMLR: Workshop and Conference Proceedings*, 42, 69–80.
- Chen, Y., & Yang, B. (2011). A hybrid fuzzy tabu search approach to the portfolio selection problem. *Expert Systems with Applications*, 38(8), 9172–9180. <https://doi.org/10.1016/j.eswa.2010.12.141>.
- Choi, R. Y., Coyner, S. A., Kalpathy-Cramer, J., ve al., E. (2020). Introduction to machine learning, neural networks, and deep learning. *Translational Vision Science ve Technology*, 9(2), 14. <https://doi.org/10.1167/tvst.9.2.14>.
- Clarke, H., Gokalp, H., ve Clarke, M. (2020). *Mapping the Digital Frontier: Bibliometric and Machine Learning Perspectives on Industry 4.0*.
- Cl  men  on, S., Depecker, M., ve Vayatis, N. (2013). Ranking forests. *Journal of Machine Learning Research*, 14, 39–73.
- Clerc, M. (2006). *Particle Swarm Optimization*. ISTE Ltd.
- Coloni, A., Dorigo, M., ve Maniezzo, V. (1991). Distributed optimization by ant colonies. *In Proceedings of the First European Conference on Artificial Life* (pp. 134-142). MIT Press.
- Coloni, A., Dorigo, M., ve Maniezzo, V. (1992). Ant system for job-shop scheduling. *Belgian Journal of Operations Research, Statistics and Computer Science*, 34(1), 39-53.

- Coşkun, F., ve Gülleroğlu, H. D. (2021). Yapay zekânın tarih içeriğindeki gelişimi ve eğitimde kullanılması. *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi (JFES)*, 54(3), 947-966. <https://doi.org/10.30964/auebfd.916220>.
- Crama, Y., ve Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research*, 150(3), 546–571.
- Criminisi, A., Shotton, J., ve Konukoglu, E. (2011). Decision forests: A unified framework for classification, regression, density estimation, manifold learning, and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7, 81–227. <https://doi.org/10.1561/06000000035>.
- Cui, Z., Sun, G., Zeng, J., ve Chen, W. N. (2013). A novel artificial bee colony algorithm based on internal-feedback strategy for global numerical optimization. *Information Sciences*, 226, 127–146.
- Cura, T. (2008). *Particle swarm optimization approach to portfolio optimization*. Istanbul University. <https://www.sciencedirect.com/science/article/abs/pii/S1468121808001259>.
- Çalış, A., Kayapınar, S., ve Çetinyokuş, T. (2014). Veri madenciliğinde karar ağacı algoritmaları ile bilgisayar ve internet güvenliği üzerine bir uygulama. *Endüstri Mühendisliği*, 25(3), 2–19.
- Çelenli Başaran, A. Z., ve Öner, B. (2020). Parçacık sürü optimizasyon algoritması ile elde edilen portföylerin AP yöntemi ile etkinliklerinin ölçülmesi. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 24(3), 669–680. <https://doi.org/10.19113/sdufenbed.793455>.
- De Prado, M. L. (2018). *Advances in Financial Machine Learning*. Wiley.
- Delen, D., Walker, G., ve Kadam, A. (2005). Predicting breast cancer survivability: A comparison of three data mining methods. *Artificial Intelligence in Medicine*, 34(2), 113–127. <https://doi.org/10.1016/j.artmed.2004.07.002>
- Deng, Y., Bao, F., Kong, Y., Ren, Z., ve Dai, Q. (2016). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3), 653–664.

- Díaz-Uriarte, R., ve De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7, 3. <https://doi.org/10.1186/1471-2105-7-3>
- Dietterich, T. G. (2000). *Ensemble methods in machine learning*. In J. Kittler ve F. Roli (Eds.), Multiple classifier systems (pp. 1–15). Springer. https://doi.org/10.1007/3-540-45014-9_1
- Ding, G., ve Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. *International Journal of Machine Learning and Cybernetics*, 11, 1307–1317. <https://doi.org/10.1007/s13042-019-01063-7>.
- Doğukanlı, H. (2019). *Finansal sistem ve finansal kurumların rolü.*, Koçyiğit, M., M. (Editör), Finansal kurumlar içinde (2-19). Eskişehir. Anadolu Üniversitesi Açıköğretim Fakültesi Yayını.
- Domingos, P. (2015). *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books.
- Dorigo, M., Maniezzo, V., ve Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1), 29-41.
- Dorigo, M., ve Stützle, T. (2004). *Ant Colony Optimization*. MIT Press.
- Draper, N. R., ve Smith, H. (1998). *Applied Regression Analysis* (3rd ed.). Wiley.
- Eberhart, R., ve Kennedy, J. (1995). A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, IEEE, 39–43. <https://doi.org/10.1109/MHS.1995.494215>
- Eiben, A. E., ve Smith, J. E. (2003). *Introduction to Evolutionary Computing*. Springer.
- Ekrem, Ö., Salman, O. K. M., Aksoy, B., ve İnan, S. A. (2020). Yapay zekâ yöntemleri kullanılarak kalp hastalığının tespiti. *Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi*, 8(5), 241–254.
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14, 179–211. [https://doi.org/10.1016/0364-0213\(90\)90002-E](https://doi.org/10.1016/0364-0213(90)90002-E).

- Elton, E. J., Gruber, M. J., Brown, S. J., ve Goetzmann, W. N. (2014). *Modern Portfolio Theory and Investment Analysis*. Wiley.
- Elton, E. J., ve Gruber, M. J. (1995). *Modern Portfolio Theory and Investment Analysis*. Wiley.
- Elton, E. J., ve Gruber, M. J. (1997). Modern Portfolio Theory, 1950 to Date. *Journal of Banking ve Finance*, 21(11-12), 1743-1759.
- Engelbrecht, A. P. (2005). *Fundamentals of Computational Swarm Intelligence*. John Wiley ve Sons.
- Erdekli, Y. K. (2022). *Prediction of the house price index of Turkey: Acomparative study of multiple linear regression and artificial neural network models* (Yüksek lisans tezi). Middle East Technical University, Institute of Science, Ankara, Turkey.
- Ertürk, Z. K., ve Ertürk, B. (2021). Acil serviste yapay zekâ kullanımı. *Aksaray Üniversitesi Tıp Bilimleri Dergisi*, 2(2), 39–40.
- Esteva, A., Kuprel, B., ve Novoa, R. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542, 115–118. <https://doi.org/10.1038/nature21056>.
- Fabozzi, F. J. (2003). *Fixed income analysis*. John Wiley ve Sons.
- Fabozzi, F. J., Gupta, F., ve Markowitz, H. M. (2002). The Legacy of Modern Portfolio Theory. *The Journal of Investing*, 11(3), 7–22.
- Fabozzi, F. J., Kolm, P. N., Pachamanova, D. A., & Focardi, S. M. (2010). *Robust Portfolio Optimization and Management*. Wiley Finance.
- Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2), 383-417. <https://doi.org/10.2307/2325486>.
- Fama, E. F., ve French, K. R. (2004). The capital asset pricing model: Theory and evidence. *Journal of Economic Perspectives*, 18(3), 25–46. <https://doi.org/10.1257/0895330042162430>
- Fischer, T., ve Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654-669.

- Fleurent, C., ve Glover, F. (1993). Genetic algorithms with memory for scheduling problems. In *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization* (pp. 75-92). Springer.
- Fogel, D. B. (1999). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press.
- Fox, J. (2016). *Applied Regression Analysis and Generalized Linear Models* (3rd ed.). Sage Publications.
- Francis, J. C., ve Ibbotson, R. G. (2006). *Investments: A Global Perspective* (7th ed.). Prentice Hall.
- François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4), 219–354. <https://doi.org/10.1561/22000000071>
- Friede, G., Busch, T., & Bassen, A. (2015). ESG and financial performance: Aggregated evidence from more than 2000 empirical studies. *Journal of Sustainable Finance & Investment*, 5(4), 210–233. <https://doi.org/10.1080/20430795.2015.1118917>
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232.
- Gao, W., Liu, S., ve Huang, L. (2015). A novel artificial bee colony algorithm with Powell's method. *Applied Soft Computing*, 13(9), 3763–3775.
- Gao, Y., Rong, W., ve Enmin, Z. (2021). Stock prediction based on optimized LSTM and GRU models. *Scientific Programming*, 1–8. <https://doi.org/10.1155/2021/9784671>.
- Genuer, R., Poggi, J. M., ve Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, 31(14), 2225–2236. <https://doi.org/10.1016/j.patrec.2010.03.014>
- Gitman, L. J., ve Zutter, C. J. (2012). *Principle of Managerial Finance* (13th edition). Prentice Hall.
- Glorot, X., Bordes, A., ve Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 15, 315–323.

- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers ve Operations Research*, 13, 533-549. [https://doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/10.1016/0305-0548(86)90048-1).
- Glover, F. (1989). Tabu search—Part I. *ORSA Journal on Computing*, 1(3), 190–206. <https://doi.org/10.1287/ijoc.1.3.190>
- Glover, F. (1990). Tabu search: A tutorial. *Interfaces*, 20(4), 74-94.
- Glover, F., Laguna, M., ve Marti, R. (2008). *Tabu search* (pp. 1–23). Springer. <https://doi.org/10.1007/978-1-4615-6089-0>.
- Glover, F., ve Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publishers.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.
- Goodfellow, I., Bengio, Y., ve Courville, A. (2016). *Deep Learning*. MIT Press.
- Graham, B. (2005). *The Intelligent Investor* (Revised edition). Harper Business Essential.
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., ve Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232.
- Guan, N., Song, J. ve Li, D. (2018). On the Advantages of Computer Multimedia-aided English Teaching, *Procedia Computer Science* 131, 7https ://doi .org /10.101/j .işlemler.2018.04.
- Gujarati, D. N., ve Porter, D. C. (2009). *Basic Econometrics* (5th ed.). McGraw-Hill Education
- Günel, M. (2006). *Para Banka ve Finansal Sistem* (Birinci baskı). Ankara: Yeni Dönem Yayınları.
- Gürbüz, S., ve Şahin, F. (2018). *Sosyal Bilimlerde Araştırma Yöntemleri*. Seçkin Yayıncılık.
- Hair, J. F., Black, W. C., Babin, B. J., ve Anderson, R. E. (2010). *Multivariate Data Analysis*. Pearson.
- Hamed, I. M., Hussein, A. S., ve Tolba, M. F. (2012). An intelligent model for stock market prediction. *International Journal of Computational Intelligence Systems*, 5(4), 639–652.

- Han, J., Pei, J., ve Kamber, M. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.
- Han, Y., Wu, W., & Fang, W. (2019). Small firm effect and liquidity in China's stock market. *Emerging Markets Finance and Trade*, 55(4), 774–790.
- Hao, C., Wang, J., Xu, W., ve Xiao, Y. (2013). Prediction-based portfolio selection model using support vector machines. In *2013 Sixth International Conference on Business Intelligence and Financial Engineering* (pp. 567–571). IEEE. <https://doi.org/10.1109/BIFE.2013.118>.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... ve Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Hastie, T., Tibshirani, R., ve Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.
- Haupt, R. L., ve Haupt, S. E. (2004). *Practical Genetic Algorithms* (2nd ed.). Wiley.
- Hawkins, D. M. (2004). The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1), 1–12.
- Haykin, S. (2009). *Neural Networks and Learning Machines* (3rd ed.). Pearson Education.
- Hochreiter, S., ve Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.
- Howells, P., ve Bain, K. (2008). *The Economics of Money, Banking and Finance: A European Text* (4th ed.). Pearson Education.
- Huang, W., Nakamori, Y., ve Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers ve Operations Research*, 32(10), 2513–2522. <https://doi.org/10.1016/j.cor.2004.03.016>.

- Huang, Y., Ni, J., ve Shen, H. (2020). A stock trading system based on the stock bar sentiment analysis using LSTM. *Neural Computing and Applications*, 32(23), 17227–17240.
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science ve Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>.
- Hyndman, R. J., ve Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts. <https://otexts.com/fpp3/>
- Hyndman, R. J., ve Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688.
- İpekten, B. (2013). *Finansal piyasaların yapısı.*, Güngör, B. (Editör), Sermaye ve para piyasaları içinde (4-24). Erzurum. Atatürk Üniversitesi Açıköğretim Fakültesi Yayını.
- Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8), 651–666.
- James, G., Witten, D., Hastie, T., ve Tibshirani, R. (2013). *An introduction to statistical learning: With applications in R*. Springer. <https://doi.org/10.1007/978-1-4614-7138-7>.
- Jitendra, K., Rimsha, G., ve Ashutosh, K. S. (2018). Long short-term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters. *Procedia Computer Science*, 125, 676–682. <https://doi.org/10.1016/j.procs.2017.12.087>.
- Johnson, D. S. (1990). Local optimization and the traveling salesman problem. In *Mathematical Programming at Oberwolfach* (pp. 35-45). Springer.
- Johnson, T. (2021). The influence of financial practice in developing mathematical probability. *Synthese*, 198, 6291–6331. <https://doi.org/10.1007/s11229-020-02636-w>.
- Jordan, M. I., ve Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260. <https://doi.org/10.1126/science.aaa8415>.
- Jurafsky, D., ve Martin, J. H. (2023). *Speech and Language Processing* (3rd ed.). Draft version. Stanford University.

- Kaelbling, L. P., Littman, M. L., ve Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4, 237–285.
- Kara, Y., Boyacıoğlu, M. A., ve Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Systems with Applications*, 38(5), 5311–5319.
- Karaboga, D., ve Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8(1), 687–697.
- Karaboğa, D. (2020). *Yapay zekâ yöntemleri* (7. Baskı). Nobel Akademik Yayıncılık.
- Karagöz, Y. (2021). *SPSS 23 ve AMOS 23 Uygulamalı İstatistiksel Analizler*. Nobel Yayıncılık.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., ve Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *In Advances in Neural Information Processing Systems* (pp. 3146-3154).
- Kennedy, J., ve Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*, 1942–1948.
- Kılıç, A. (2014). *Yapay zekâ tekniği ile arama motoru optimizasyonu analizi yapan akıllı web site* (Yüksek lisans tezi). Gazi Üniversitesi, Bilişim Enstitüsü, Ankara, Türkiye.
- Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1–2), 307–319. [https://doi.org/10.1016/S0925-2312\(03\)00372-2](https://doi.org/10.1016/S0925-2312(03)00372-2).
- Kim, M., Yun, J., Cho, Y., Shin, K., Jang, R., Bae, H. J., ve Kim, N. (2019). Deep learning in medical imaging. *Neurospine*, 16(4), 657–668. <https://doi.org/10.14245/ns.1938396.198>.
- Kober, J., Bagnell, J. A., ve Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11), 1238–1274.
- Kocabaş, Ş. (2015). Yapay zekâ araştırma ve uygulama alanları. *Afyon Kocatepe Üniversitesi, Bilgisayar Bilimleri Dergisi*, Özel Sayı, 1–10.

- Konak, A., Coit, D. W., ve Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering ve System Safety*, 91(9), 992–1007.
- Korkmaz, T., & Çevik, E. İ. (2009). Optimal portföy seçimi ve davranışsal finans: İMKB üzerine bir uygulama. *ZKU Sosyal Bilimler Dergisi*, 5(10), 69–90.
- Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica*, 31(3), 249–268.
- Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4), 261–283.
- Köse, A., ve Ersöz, S. (2020). *Veri Madenciliği Yöntemleriyle Kredi Riski Analizi*. Sakarya Üniversitesi Yayınları.
- Köse, N., ve Ersöz, F. (2020). Veri madenciliğinde karar ağacı algoritmaları ile demir çelik endüstrisinde iş kazaları üzerine bir uygulama. *Avrupa Bilim ve Teknoloji Dergisi*, 397–407. <https://doi.org/10.31590/ejosat.843837>
- Krizhevsky, A., Sutskever, I., ve Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *In Advances in Neural Information Processing Systems (NeurIPS)*, 25.
- Kubat, M. (2017). *An introduction to machine learning* (2nd ed.). Springer-Verlag.
- Kumar, A., Sharma, S., ve Mahdavi, M. (2021). Machine learning (ML) technologies for digital credit scoring in rural finance: A literature review. *Risks*, 9(11), 192. <https://doi.org/10.3390/risks9110192>.
- Kumar, M., Singh, V., ve Kumar, D. (2020). Recent applications of artificial bee colony algorithm: a survey. *Artificial Intelligence Review*, 53, 3713–3755.
- Kumar, S., & Ravi, V. (2021). A survey of the applications of machine learning in stock market prediction. *Journal of King Saud University – Computer and Information Sciences*, 33(5), 619–636. <https://doi.org/10.1016/j.jksuci.2018.09.014>.
- Kurtboğan, H. (2023). *Yeni dünyada yapay zekâ metaforu ve yapay zekânın çalışan performansına etkisi* (Yüksek lisans tezi). Karamanoğlu Mehmet Bey Üniversitesi, Sosyal Bilimler Enstitüsü, Karaman, Türkiye.

- Kutner, M. H., Nachtsheim, C. J., Neter, J., ve Li, W. (2005). *Applied Linear Statistical Models* (5th ed.). McGraw-Hill.
- LeCun, Y., Bengio, Y., ve Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>.
- LeCun, Y., Bottou, L., Orr, G. B., ve Müller, K. R. (1998). *Efficient BackProp*. In *Neural Networks: Tricks of the Trade* (pp. 9–50). Springer.
- Lee, J., ve Yoon, S. (2021). Medical big data: Promise and challenges. *Kidney Research and Clinical Practice*, 40(2), 141–149. <https://doi.org/10.23876/j.krcp.21.007>
- Lemaitre, G., Nogueira, F., ve Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, 18(17), 1–5.
- Lemieux, C., Toubal, L., ve Vandal, N. (2019). Financial applications of machine learning: Random forests and beyond. *Risks*, 7(2), 64. <https://doi.org/10.3390/risks7020064>.
- Levine, R. (2005). Finance and growth: Theory and evidence. In P. Aghion ve S. N. Durlauf (Eds.), *Handbook of economic growth* (Vol. 1, pp. 865–934). Elsevier. [https://doi.org/10.1016/S1574-0684\(05\)01012-9](https://doi.org/10.1016/S1574-0684(05)01012-9)
- Li, X., Xie, H., Chen, L., Wang, J., ve Deng, X. (2021). News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, 218, 106718.
- Li, Y. (2017). *Deep reinforcement learning: An overview*. arXiv preprint, arXiv:1701.07274. <https://doi.org/10.48550/arXiv.1701.07274>
- Li, Y., Wu, L., & Wang, Y. (2021). Sentiment-aware stock market prediction: A deep learning approach. *Information Processing & Management*, 58(5), 102715. <https://doi.org/10.1016/j.ipm.2021.102715>
- Lin, C. C., ve Liu, H. T. (2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research*, 185(1), 393–404.
- Lin, L., Wang, Y., Sun, D., ve Chen, J. (2011). Path planning for robot based on genetic algorithm. *2011 International Conference on Electronics, Communications and Control (ICECC)*, 2826–2829.

- Lintner, J. (1965). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *The Review of Economics and Statistics*, 47(1), 13–37. <https://doi.org/10.2307/1924119>.
- Liu, H., Lang, B., Liu, M., ve Yan, H. (2020). A survey on evolutionary neural architecture search. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5), 2071–2089.
- Louppe, G., Wehenkel, L., Sutter, A., ve Geurts, P. (2013). Understanding variable importances in forests of randomized trees. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, ve K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 26). Curran Associates, Inc. https://papers.nips.cc/paper_files/paper/2013/file/e92e7de6837fc08d259cabcd2a7f4815-Paper.pdf
- Luenberger, D. (2013). *Investment Science* (2nd edition) Oxford University Press.
- Ma, Y., Han, R., ve Wang, W. (2021). *Portfolio optimization with return prediction using deep learning and machine learning*. Southeast University. <https://www.sciencedirect.com/science/article/abs/pii/S0957417420307521>.
- Makridakis, S., Spiliotis, E., ve Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3), e0194889. <https://doi.org/10.1371/journal.pone.0194889>.
- Makridakis, S., Wheelwright, S. C., ve Hyndman, R. J. (1998). *Forecasting: Methods and Applications*. Wiley.
- Markowitz, H. (1952). "Portfolio Selection." *Journal of Finance*, 7(1), 77-91.
- Markowitz, H. M. (1991). Foundations of Portfolio Theory. *Financial Analysts Journal*, 47(2), 7-16.
- McKinney, W. (2010). Data structures for statistical computing in Python. In S. van der Walt ve J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 51–56). <https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf>.
- Metaxiotis, K., & Liagkouras, K. (2012). Multi-objective evolutionary algorithms for portfolio management: A comprehensive literature review. *Expert Systems with Applications*, 39(14), 11685–11698. <https://doi.org/10.1016/j.eswa.2012.04.006>

- Miotto, R., Wang, F., Wang, S., Jiang, X., ve Dudley, J. T. (2017). Deep learning for healthcare: review, opportunities and challenges. *Briefings in Bioinformatics*, 19(6), 1236–1246.
- Mishkin, F. (2018). *The Economics of Money, Banking and Financial Markets: Business School Edition* (Fifth edition). New York: Pearson.
- Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press.
- Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.
- Montgomery, D. C., Peck, E. A., ve Vining, G. G. (2012). *Introduction to Linear Regression Analysis* (5th ed.). Wiley.
- Moody, J., ve Saffell, M. (2001). Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12(4), 875–889.
- Mossin, J. (1966). Equilibrium in a capital asset market. *Econometrica*, 34(4), 768–783. <https://doi.org/10.2307/1910098>.
- Müller, A. C., ve Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media.
- Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., ve Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1), 1–21.
- Nelson, D. M., Pereira, A. C. M., & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. *International Joint Conference on Neural Networks (IJCNN)*, 1419–1426. <https://doi.org/10.1109/IJCNN.2017.7966019>
- Nilsson, N. J. (2010). *The Quest for Artificial Intelligence: A History of Ideas and Achievements*. Cambridge University Press.
- Nowicki, E., ve Smutnicki, C. (2005). A fast tabu search algorithm for the job shop problem. *Management Science*, 42(6), 797-813.

- Öndeş, T. (2013). *Sermaye piyasası araçları: Hisse senetleri*. Güngör, B. (Editör), Sermaye ve para piyasaları içinde (114-133). Erzurum. Atatürk Üniversitesi Açıköğretim Fakültesi Yayını.
- Özçalıcı, M., ve Ayriçay, Y. (2016). Bilgi işlemsel zekâ yöntemleri ile hisse senedi fiyat tahmini: BIST uygulaması. *Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*(Ek1), 274–298. <https://doi.org/10.5505/pausbed.2016.46334>.
- Özer, İ. (2020). Uzun kısa dönem bellek ağlarını kullanarak erken aşama diyabet tahmini. *Mühendislik Bilimleri ve Araştırmaları Dergisi*, 2(2), 50–57.
- Özgür, S. B. (2021). Algoritmalar, Yapay zekâ, Makine Öğrenmesi, Derin Öğrenme ve Uygulamaları: Beşeri Fayda Üretiminin Yazılımlar Tarafından Karşılanması. *Ekonomi ve Yönetim Araştırmaları Dergisi*, 10(1), 1-29.
- Özkan, B., & Erdal, H. (2020). Comparison of deep learning and machine learning methods in financial time series prediction. *Sakarya University Journal of Science*, 24(3), 456–468. <https://doi.org/10.16984/saufenbilder.645530>.
- Öztemel, E. (2020). *Yapay Sinir Ağları* (Dördüncü baskı). Papatya Bilim Yayınları.
- Öztemel, E. (2020). *Yapay Zekâ: Kavramlar, Yöntemler ve Uygulamalar* (Güncellenmiş Baskı). Papatya Yayıncılık.
- Padhi, D. K., Padhy, N., Bhoi, A. K., Shafi, J., ve Yesuf, S. H. (2022). An intelligent fusion model with portfolio selection and machine learning for stock market prediction. *Computational Intelligence and Neuroscience*, 2022, Article ID 7588303. <https://doi.org/10.1155/2022/7588303>.
- Pan, S. J., ve Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359.
- Panc, T., Szolovits, P., ve Atun, R. (2020). Artificial intelligence, machine learning, and health systems. *Artificial Intelligence in Medicine*, 8(2), 020303.
- Parsopoulos, K. E., ve Vrahatis, M. N. (2002). Recent approaches to global optimization problems through Particle Swarm Optimization. *Natural Computing*, 1(2-3), 235–306.

- Patel, J., Shah, S., Thakkar, P., ve Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, 42(4), 2162–2172.
- Patel, J., Shah, S., Thakkar, P., ve Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42(1), 259–268.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... ve Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. <http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>
- Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., ve Zaidi, M. (2006). The Bees Algorithm – A Novel Tool for Complex Optimisation Problems. In *Proceedings of 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2006)*, 454–459.
- Pilbeam, K. (2018). *Finance and Financial Markets* (4th ed.). Red Globe Press.
- Poli, R., Kennedy, J., ve Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, 1(1), 33–57. <https://doi.org/10.1007/s11721-007-0002-0>.
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., ve Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems* (pp. 6638-6648).
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- Reilly, F. K., ve Brown, K. C. (2012). *Investment analysis and portfolio management*. Cengage Learning.
- Rumelhart, D. E., Hinton, G. E., ve Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533–536. <https://doi.org/10.1038/323533a0>.
- Russell, S. J., ve Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Saini, A. (2024). *Guide on support vector machine (SVM) algorithm*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/10/support-vector-machines-svm-a-complete-guide-for-beginners/>

- Salıř, D. (2024). *Temel ve teknik analize dayalı makine öğrenmesi ile hisse senedi fiyat tahmini: Rastgele orman sınıflandırması yaklaşımı* (Yüksek lisans tezi). Gebze Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, Türkiye.
- Sayılgan, G. (2017). *İřletme Finansmanı* (Yedinci baskı). Ankara: Siyasal Kitabevi.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117.
- Sebastian, A. ve Tantia, V. (2024). Deep Learning for Stock Price Prediction and Portfolio Optimization. *International Journal of Advanced Computer Science and Applications*. 15. 926-941. 10.14569/IJACSA.2024.0150995.
- Seber, G. A. F., ve Lee, A. J. (2012). *Linear Regression Analysis* (2nd ed.). Wiley.
- Seyidođlu, H. (2016). *Uluslararası Finans* (Altıncı baskı). İstanbul: Güzem Can Yayınları.
- Sezer, O. B., Gudelek, M. U., ve Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90, 106181.
- Shankar, S., Chiticariu, L., & Varadarajan, V. (2018). *Deep learning frameworks: An overview*. IBM Research.
- Sharpe, W. F. (1964). Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk. *Journal of Finance*, 19(3), 425-442.
- Sharpe, W. F. (1966). Mutual Fund Performance. *Journal of Business*, 39(1), 119–138.
- Sharpe, W. F., Alexander, G. J., ve Bailey, J. V. (1999). *Investments*. Prentice Hall.
- Shen, J., ve Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, 7(1). <https://doi.org/10.1186/s40537-020-00333-6>.
- Shi, Y., ve Eberhart, R. C. (1998). A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation Proceedings*, 69–73.
- Shukla, P. K., Tiwari, R., & Mahajan, R. (2014). Optimizing portfolio selection using particle swarm optimization algorithm. *Procedia Computer Science*, 89, 803–808. <https://doi.org/10.1016/j.procs.2016.06.039>
- Sivanandam, S. N., ve Deepa, S. N. (2008). *Introduction to Genetic Algorithms*. Springer.

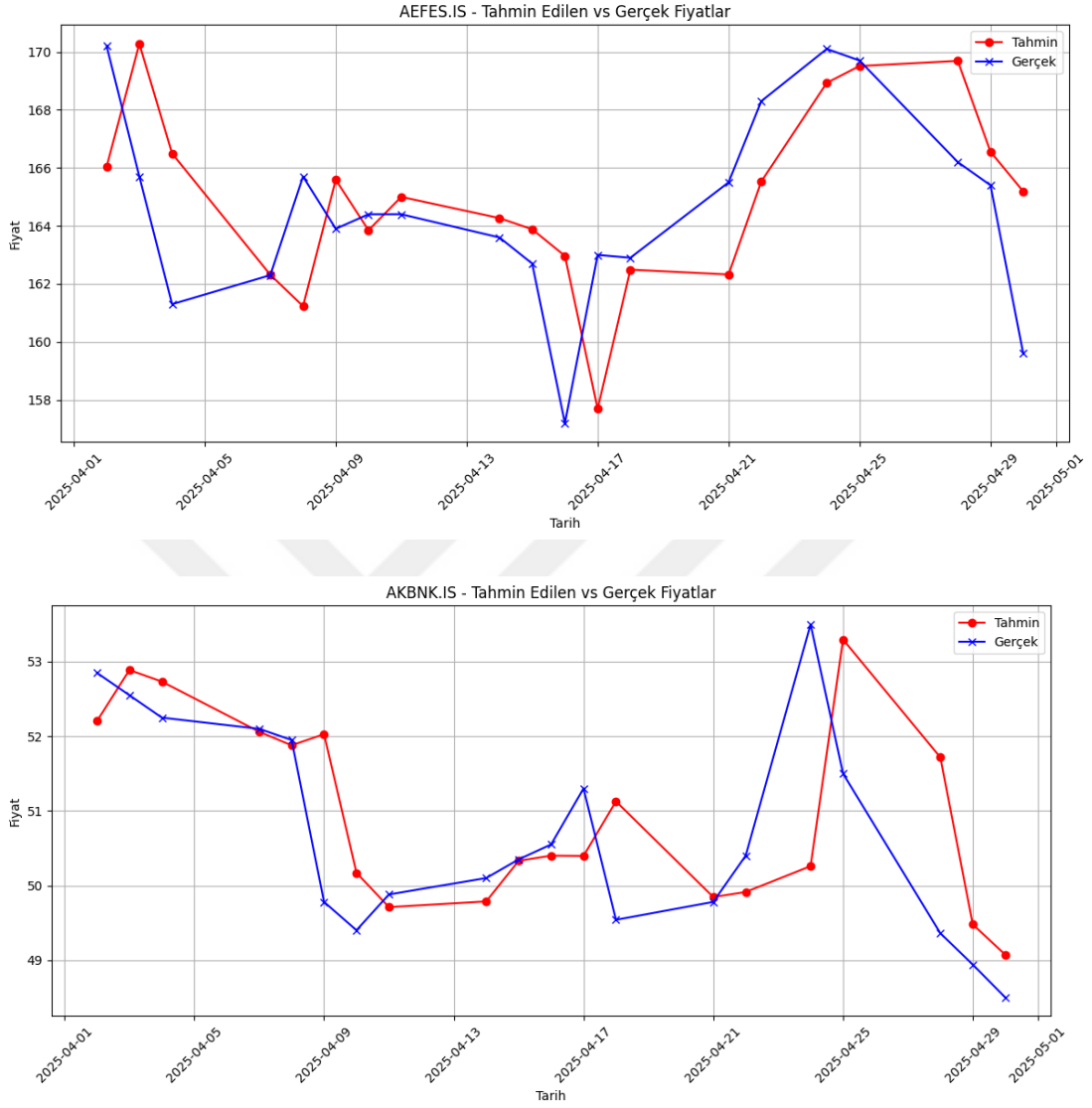
- Socha, K., ve Dorigo, M. (2008). Ant colony optimization for continuous domains. *European Journal of Operational Research*, 185(3), 1155-1173.
- Sortino, F. A., ve Price, L. N. (1994). Performance Measurement in a Downside Risk Framework. *The Journal of Investing*, 3(3), 59–64.
- Streichert, F., Ulmer, H., ve Zell, A. (2004). Evolutionary algorithms and the cardinality constrained portfolio optimization problem. *Proceedings of the IEEE Congress on Evolutionary Computation*, 2, 1349–1356.
- Strobl, C., Malley, J., ve Tutz, G. (2009). An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods*, 14(4), 323.
- Stützle, T., ve Hoos, H. H. (2000). MAX–MIN ant system. *Future Generation Computer Systems*, 16(8), 889-914.
- Sutton, R. S., ve Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.
- Taillard, É. (1991). Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17(4-5), 443-455.
- Tatlıdil, H. (2010). *Uygulamalı Çok Değişkenli İstatistiksel Analiz*. Akademi Yayınları.
- Tezel, G. (2016). *Finansal Piyasalar ve Kurumlar*. Ankara: Gazi Kitabevi.
- Tokalaçoğlu, E. (2020). *Karar Ağacı Temelli Geri Bildirim Sistemi Tasarımı: Türkiye Muthuluk Ağacı Üzerine Bir Uygulama* (Yüksek lisans tezi). İstanbul Teknik Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul, Türkiye.
- Tokalaçoğlu, S. (2020). *Makine Öğrenmesi ile Veri Analizi*. Nobel Akademik Yayıncılık.
- Toprak, A. (2023). *Yapay Zeka Tabanlı Finansal Tahmin Modelleri*. İstanbul: Akademi Yayıncılık.
- Wooldridge, J. M. (2016). *Introductory Econometrics: A Modern Approach* (6th ed.). Cengage Learning.

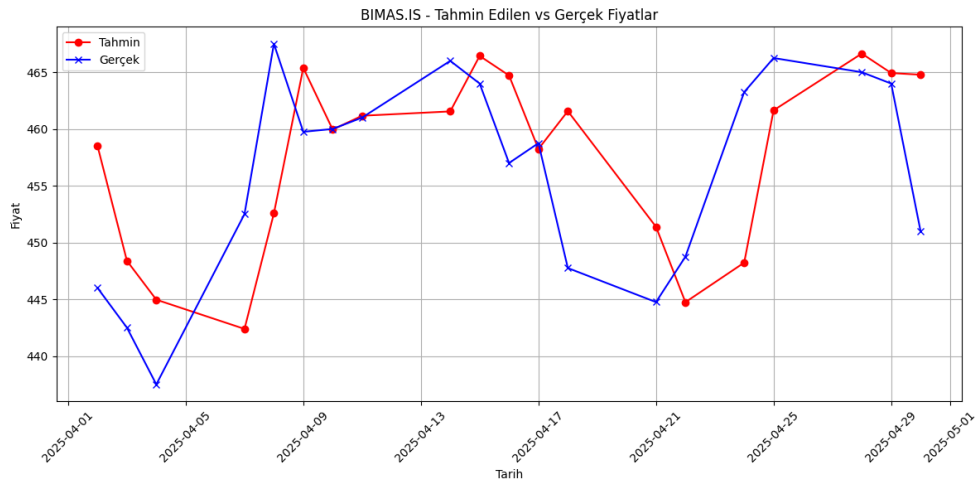
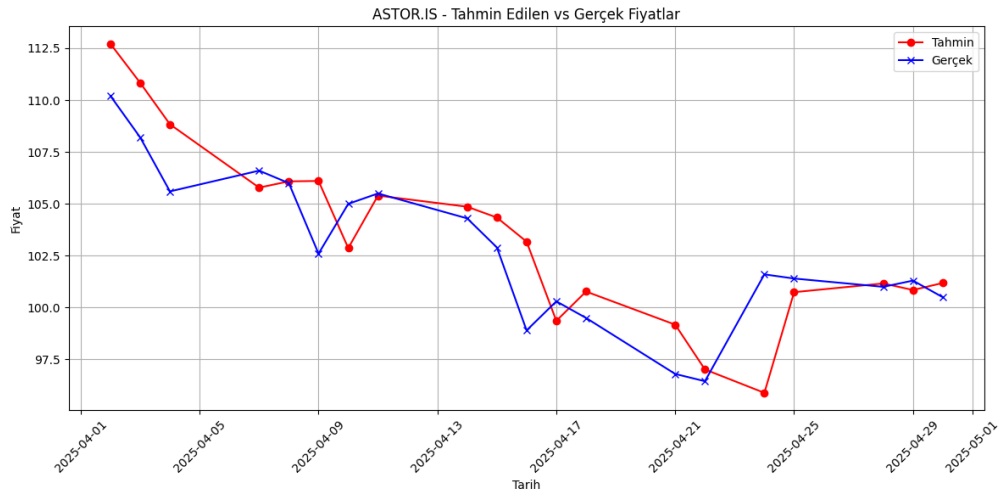
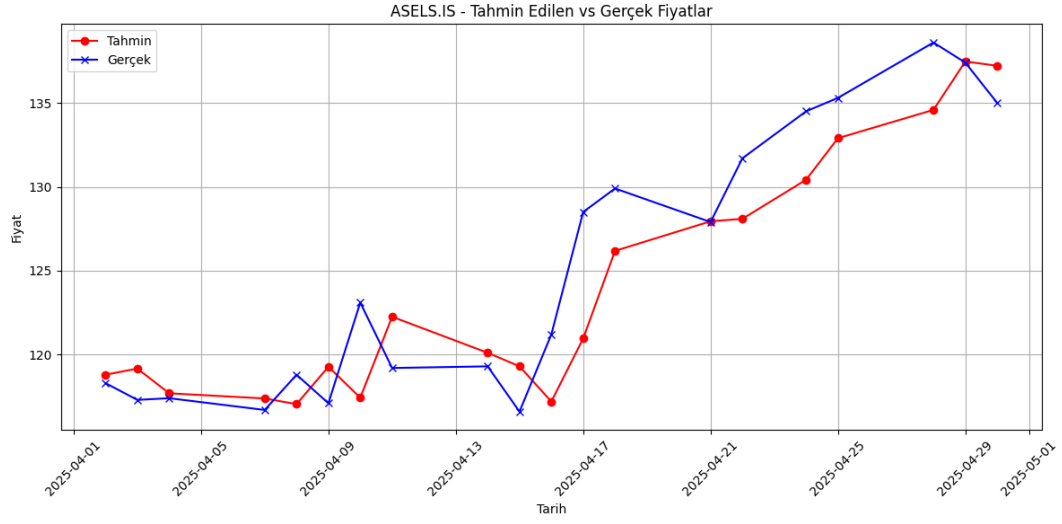
- Wuyu, W., Weizi, L., Ning, Z., ve Kecheng, L. (2020). *Portfolio formation with preselection using deep learning from long-term financial data*. Central University of Finance and Economics. <https://www.sciencedirect.com/science/article/abs/pii/S0957417419307596>.
- Xu, R., ve Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 645–678.
- Yalama, A., & Başçı, E. S. (2012). Küçük ve büyük ölçekli firmaların risk-getiri ilişkileri üzerine karşılaştırmalı bir analiz. *İktisat İşletme ve Finans*, 27(319), 67–90. <https://doi.org/10.3848/iif.2012.319.2752>
- Yang, X. S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver Press.
- Yao, Y., Rosasco, L., & Caponnetto, A. (2007). On early stopping in gradient descent learning. *Constructive Approximation*, 26(2), 289–315. <https://doi.org/10.1007/s00365-006-0663-2>
- Yetkiner, H., ve Korap, L. (2020). *Finansal İktisat*. Ege Üniversitesi Yayınları.
- Yıldız, A. (2022). Finans Alanında Yapay zekâ Teknolojisinin Kullanımı: Sistematik Literatür İncelemesi. *Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, (52), 47-66.
- Yıldız, E. (2022). Makine Öğrenmesi Uygulamalarında Regresyon Yöntemleri. *Akademik Platform Dergisi*, 8(1), 1-10.
- Young, T., Hazarika, D., Poria, S., ve Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3), 55–75.
- Yu, C., Liu, J., Nemati, S., ve Sun, J. (2019). Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1), 1–36.
- Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, 31(7), 1235-1270.
- Zaharia, M., Das, T., Li, H., Hunter, T., Shenker, S., & Stoica, I. (2018). Discretized streams: Fault-tolerant streaming computation at scale. *Communications of the ACM*, 61(4), 94–104.

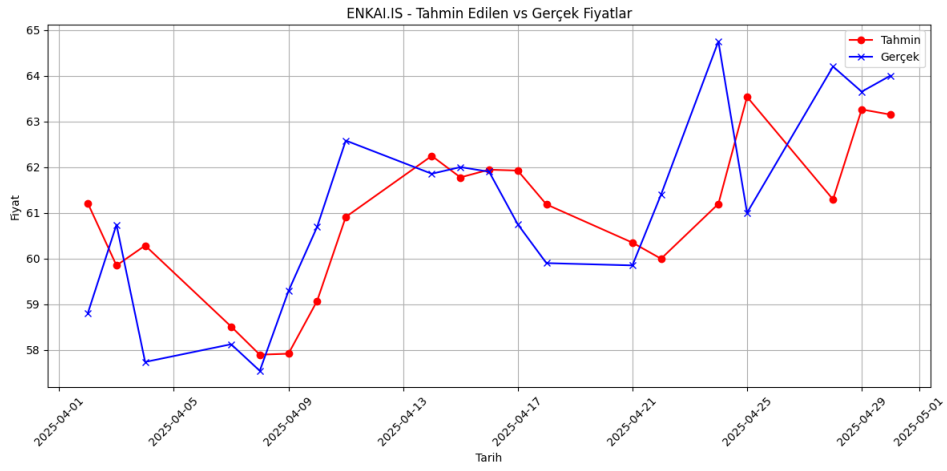
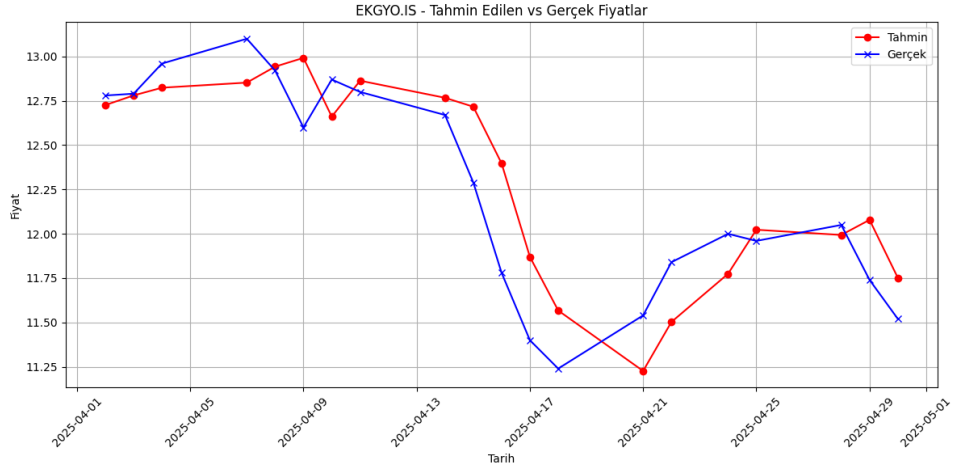
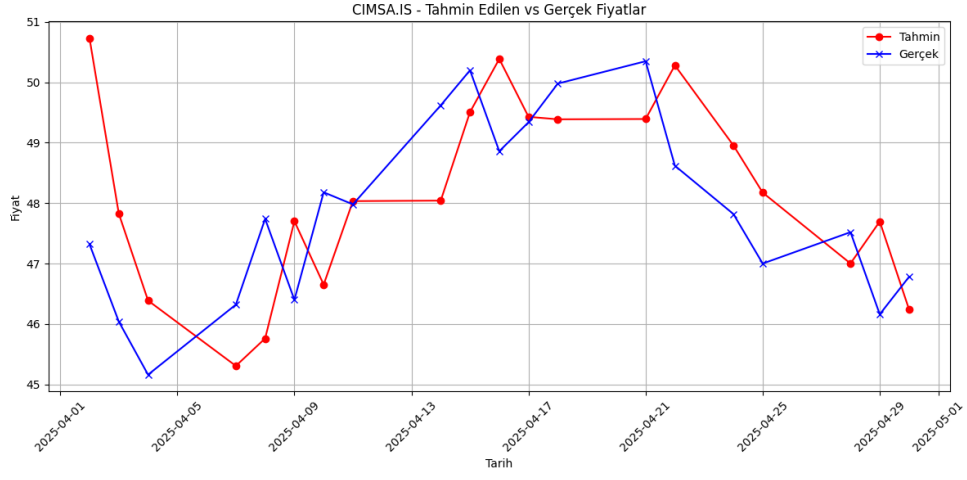
- Zawacki-Richter, O., Marín, V. I., Bond, M., ve Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – Where are the educators? *International Journal of Educational Technology in Higher Education*, 16(1), 39.
- Zhang, C., Cheng, Z., ve Ibrahim, R. (2020). A novel deep learning framework: Prediction and analysis of financial time series using CEEMD and LSTM. *Expert Systems with Applications*, 159, 113609.
- Zhang, J., Wang, P., ve Gao, R. X. (2019). Deep learning-based tensile strength prediction in fused deposition modeling. *Computers in Industry*, 107, 11-21. <https://doi.org/10.1016/j.compind.2019.01.004>.
- Zhang, M., Wang, S., Tang, Y., ve Zhao, Y. (2021). UAV path planning with deep reinforcement learning. *Drones*, 5(3), 66.
- Zhang, S., Zhao, J., ve Yu, J. (2017). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1), 1–38.
- Zhang, T., ve Mo, H. (2021). Reinforcement learning for robot research: A comprehensive review and open issues. *International Journal of Advanced Robotic Systems*, 18(3). <https://doi.org/10.1177/17298814211007305>.
- Zhou, Z. H. (2012). *Ensemble methods: Foundations and algorithms*. Chapman ve Hall/CRC. <https://doi.org/10.1201/b12207>
- Zhu, G., ve Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 217(7), 3166–3173.
- Zülfüoğlu, Ö. (2013). *Kamusal bir mal olarak finansal sistemin regülasyonu: Türk Bankacılık Sektörü Örneği*. Doktora Tezi. Marmara Üniversitesi, İstanbul.

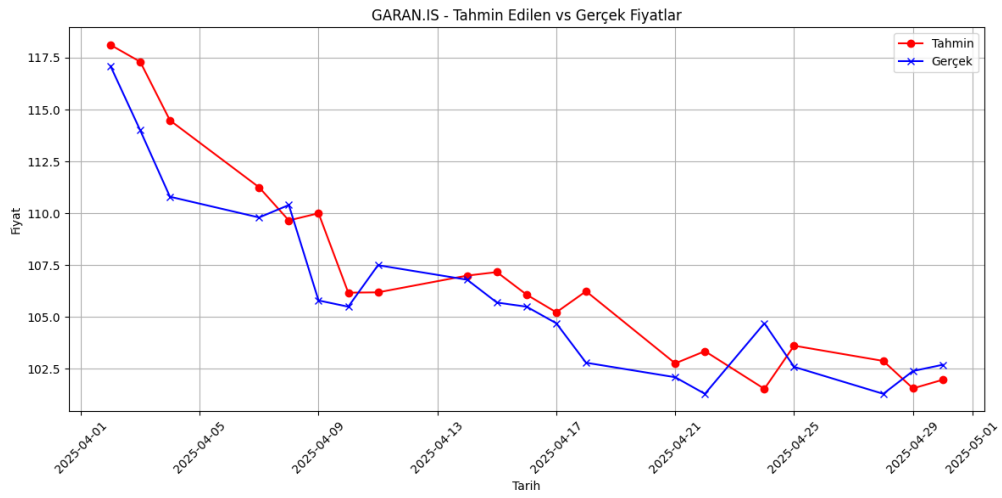
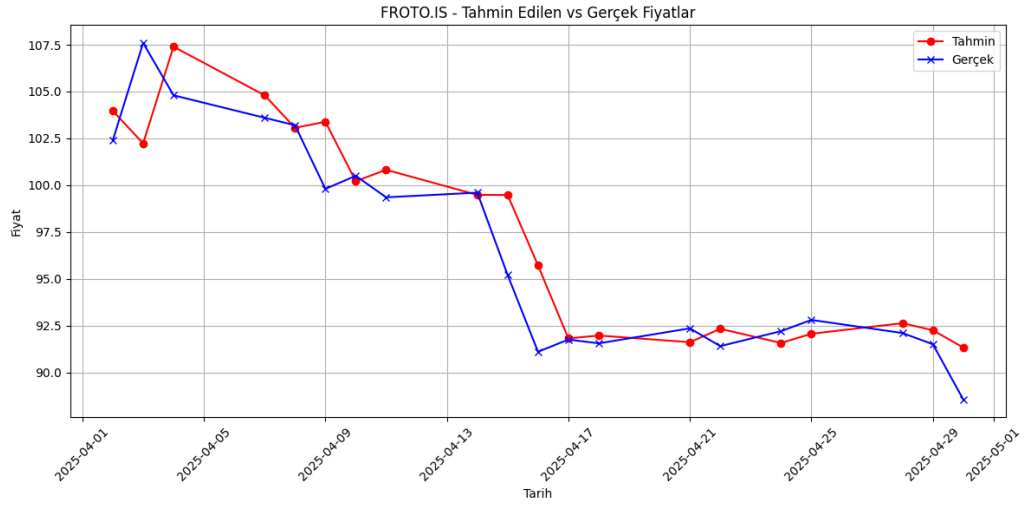
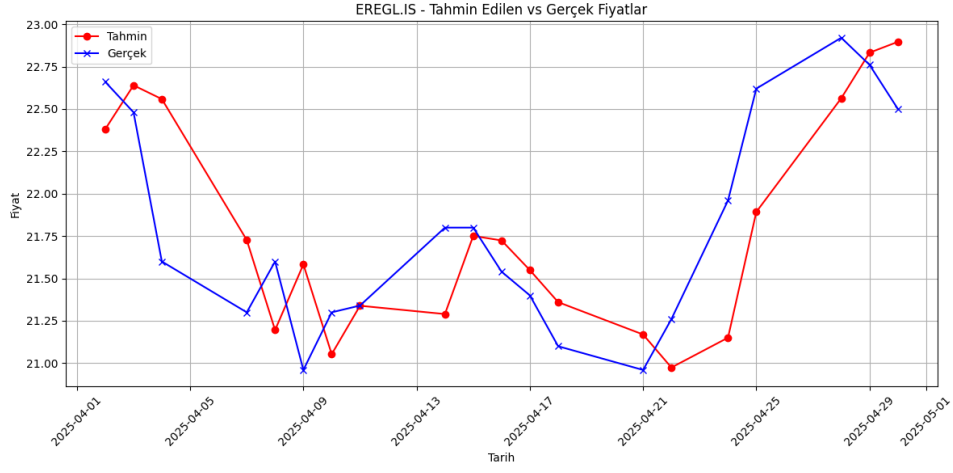
6. EKLER

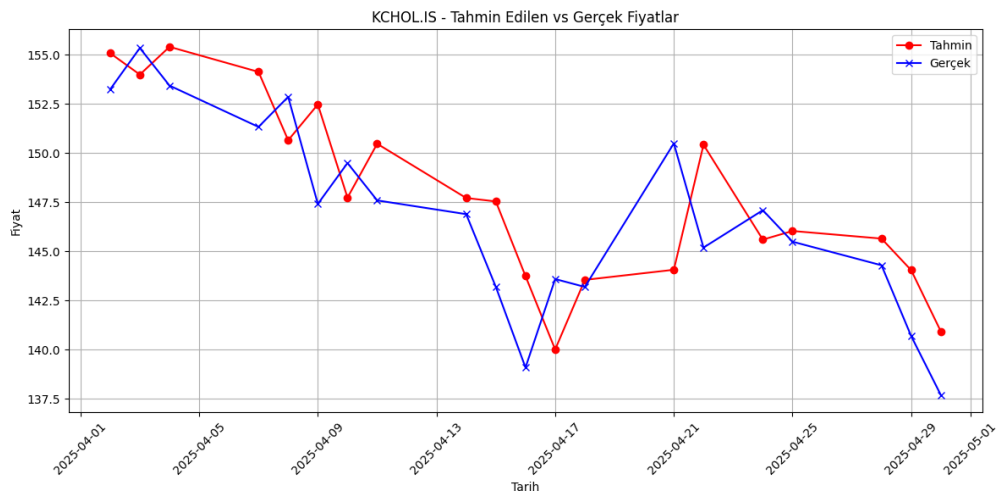
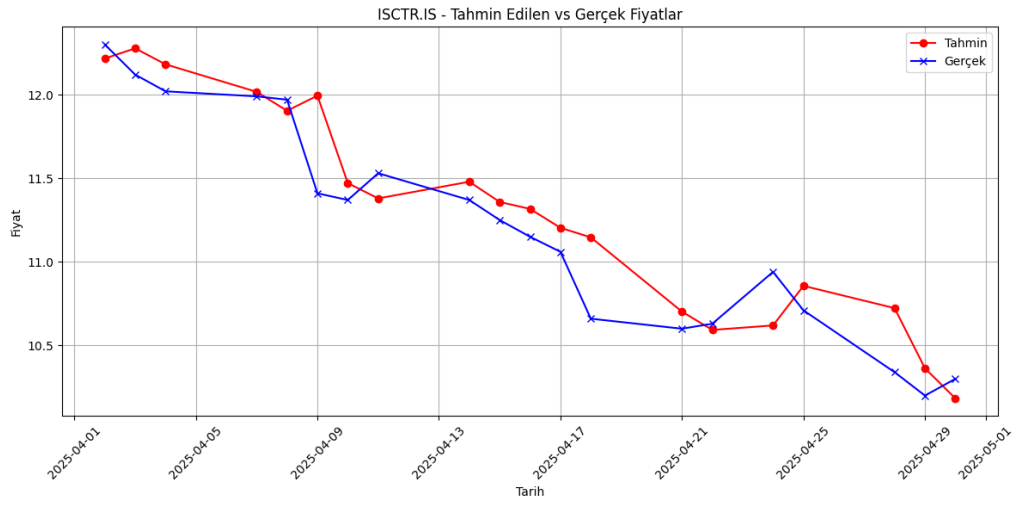
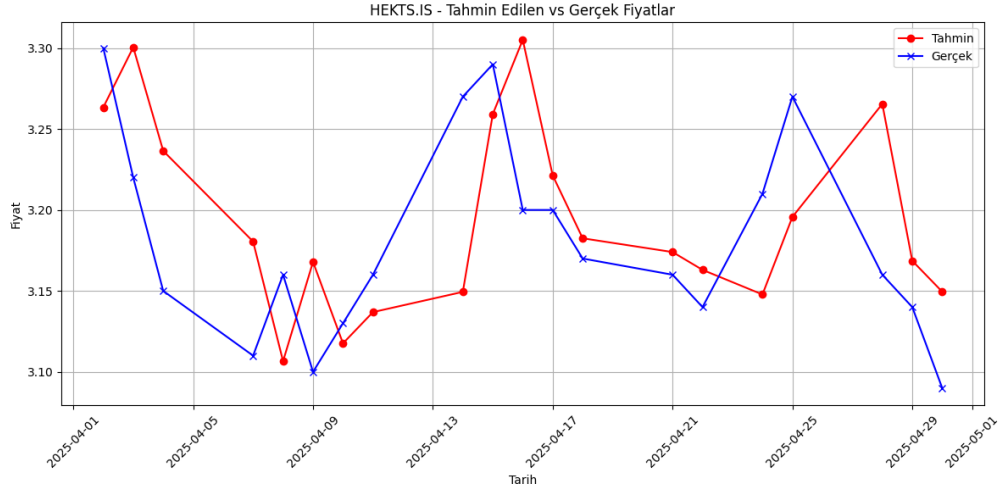
Ek 1. BİST 30 endeksindeki hisse senetlerinin tahmin edilen değerleri ile gerçek değerlerinin grafikleri

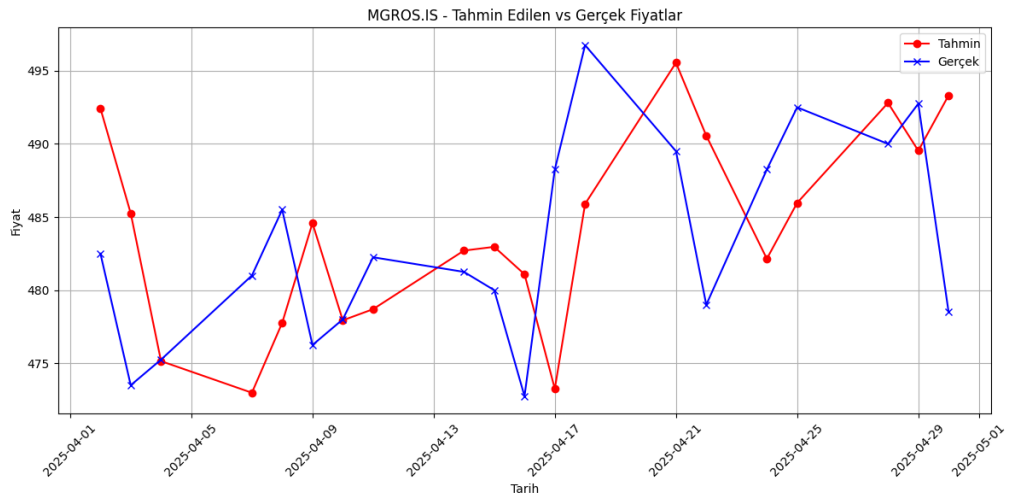
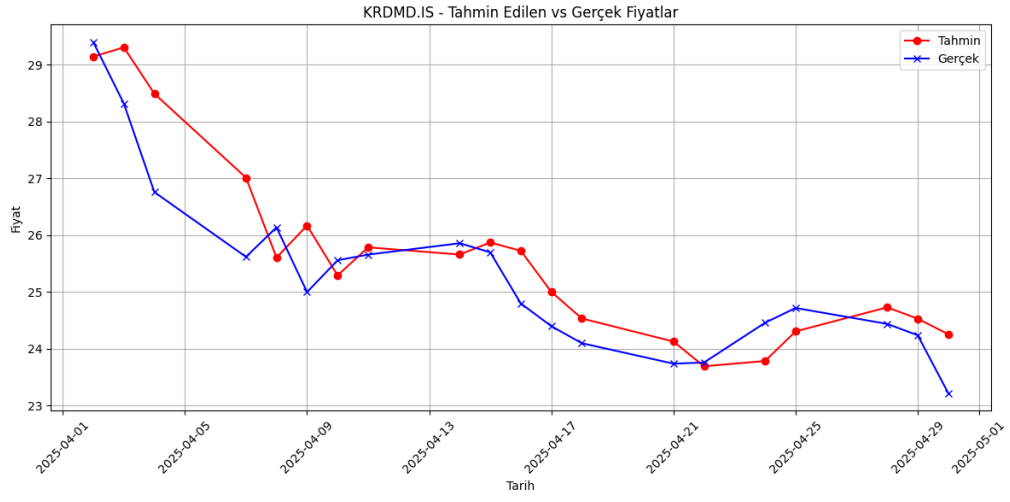
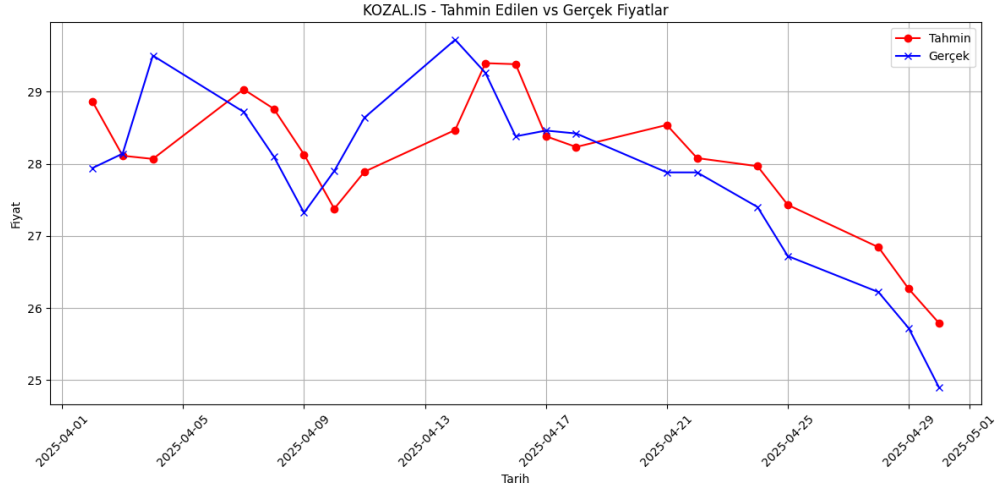


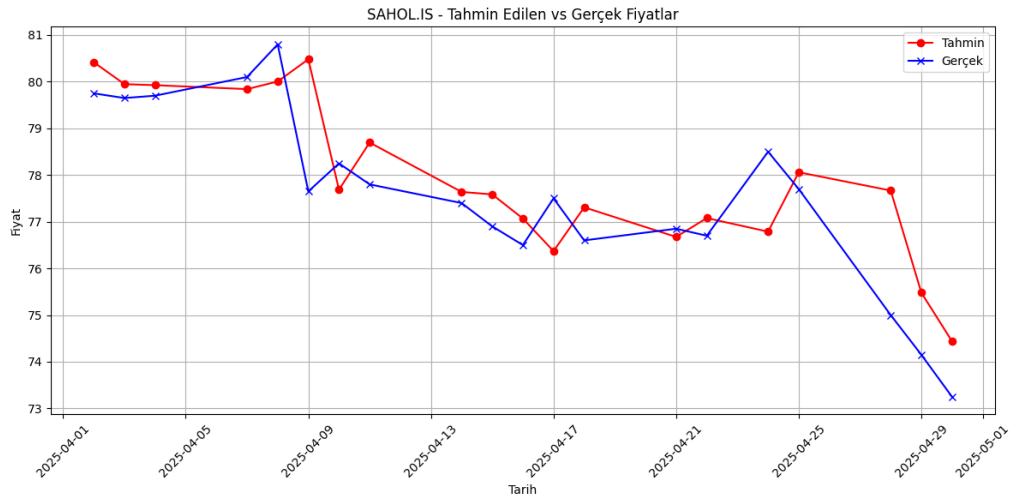
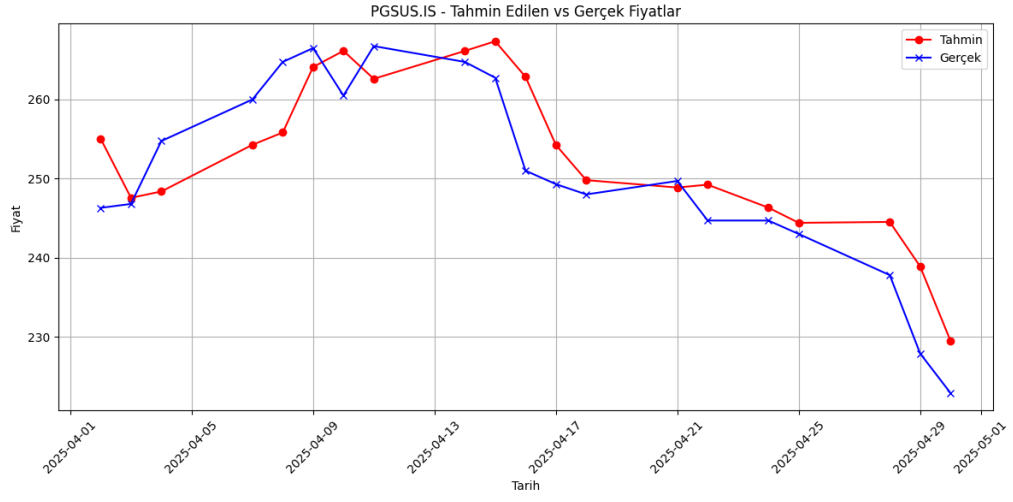
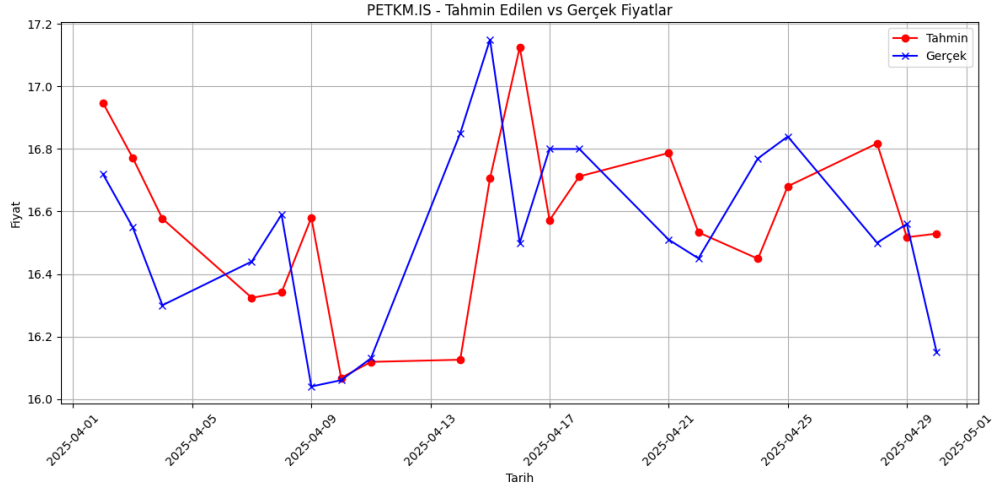


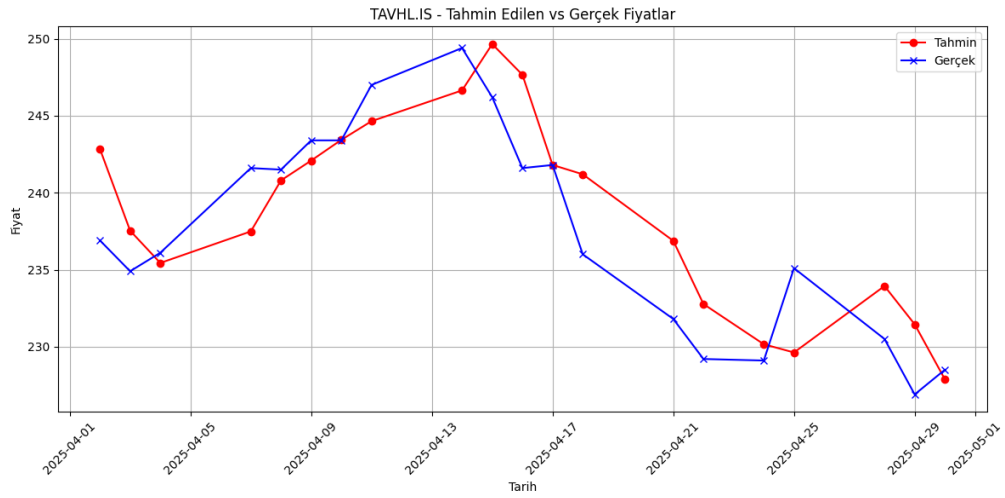
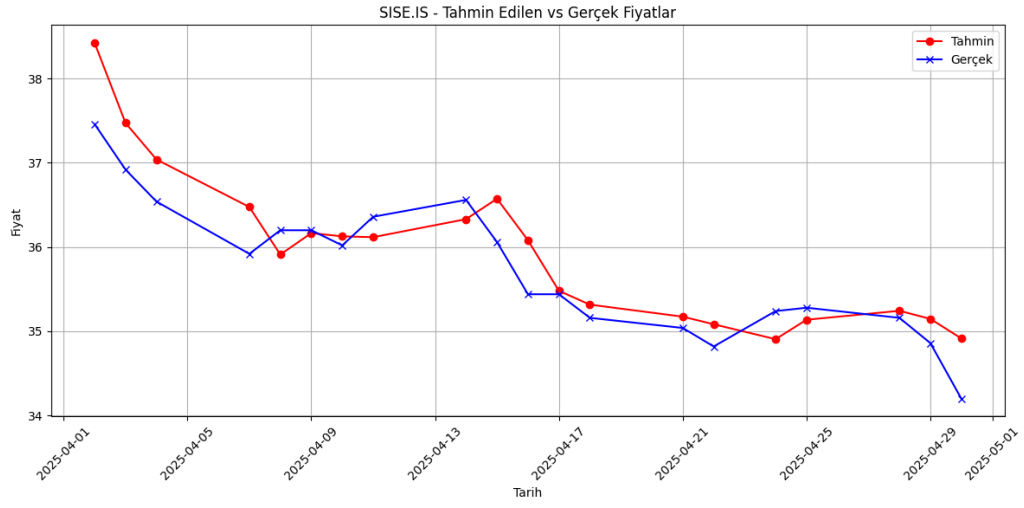
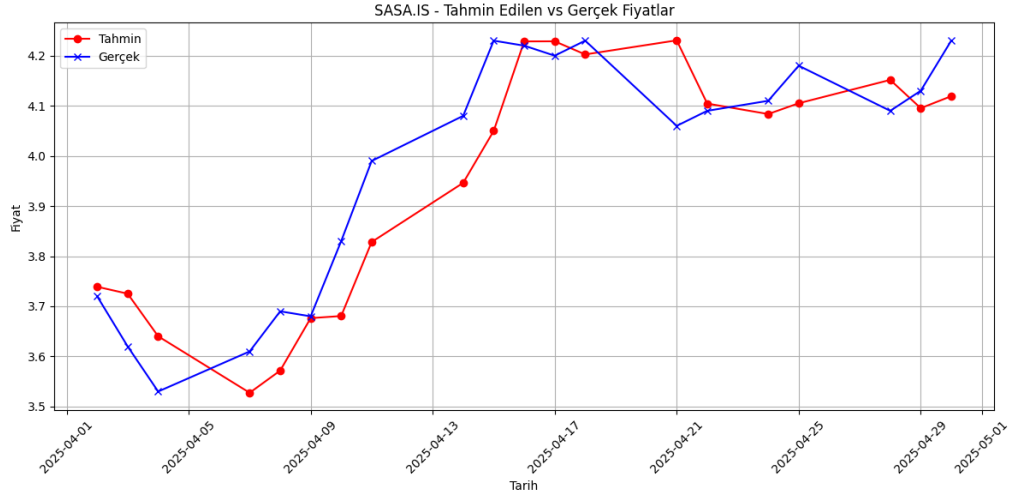


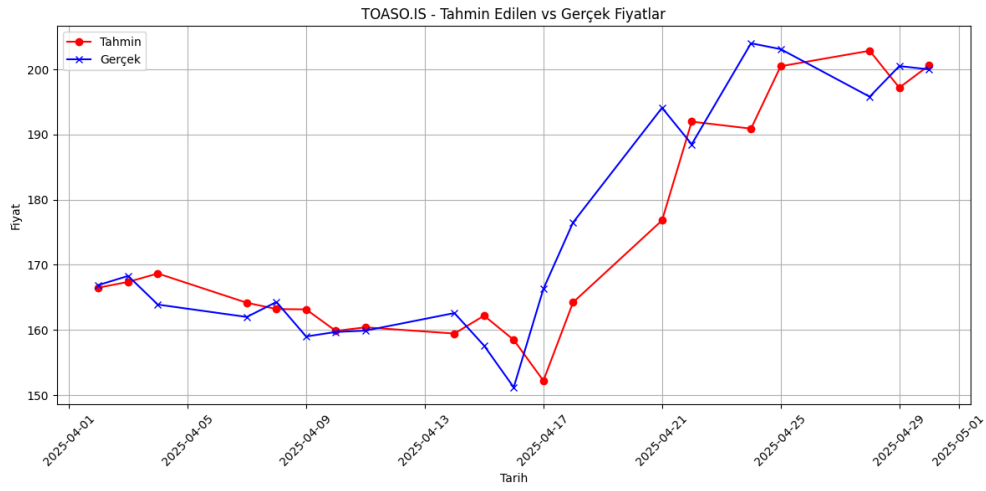
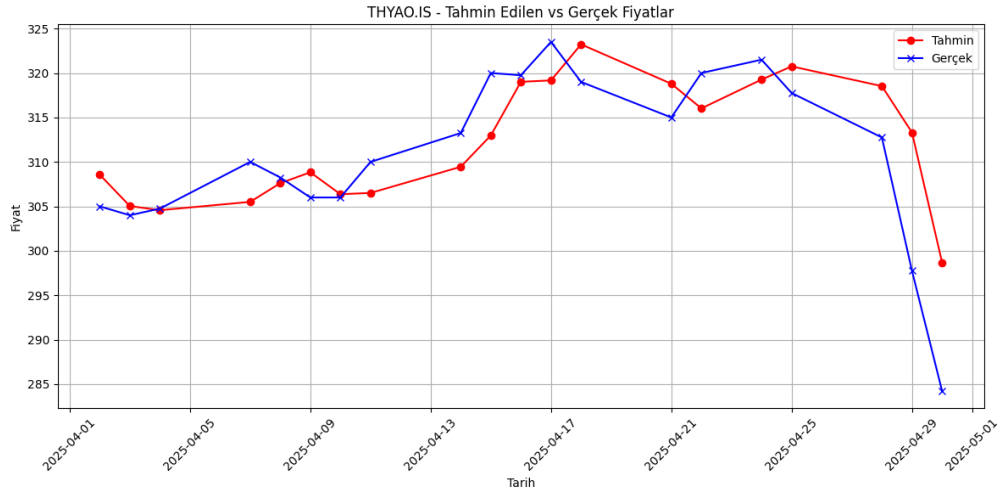
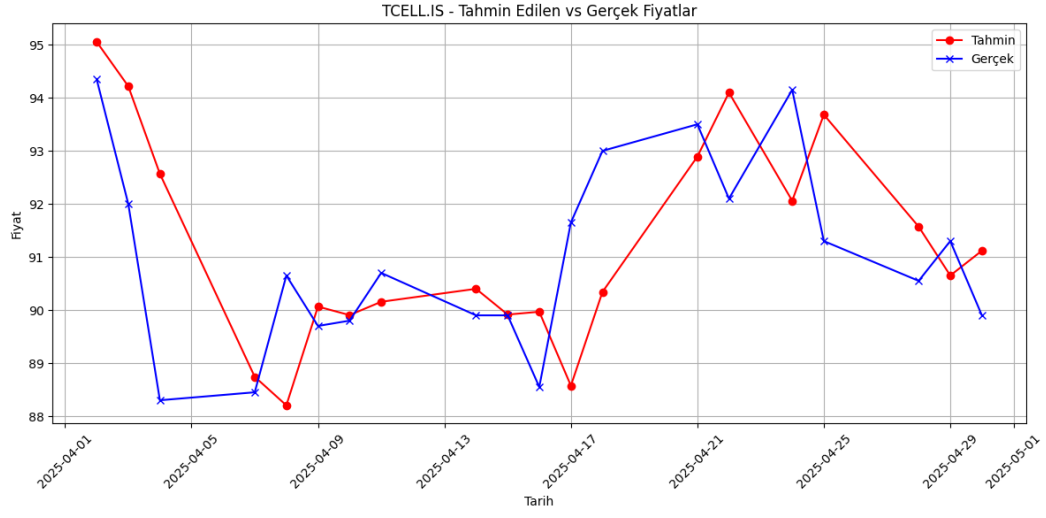


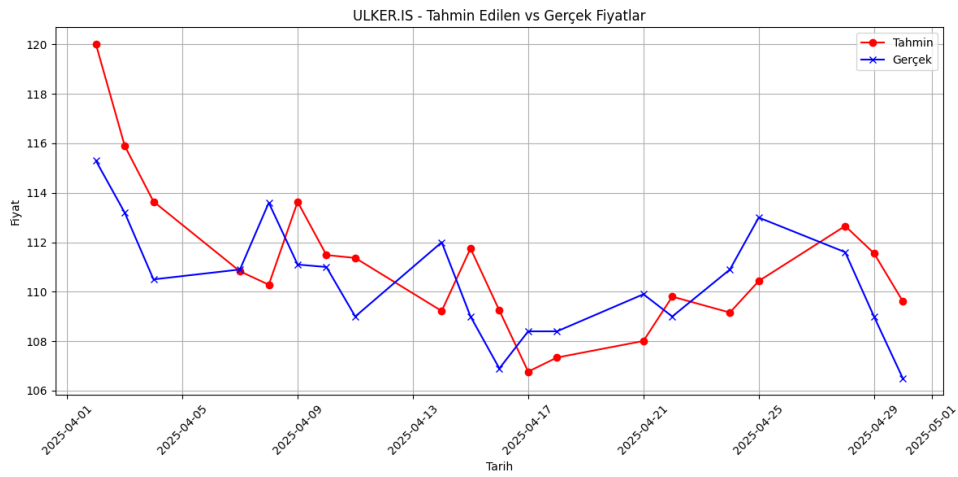
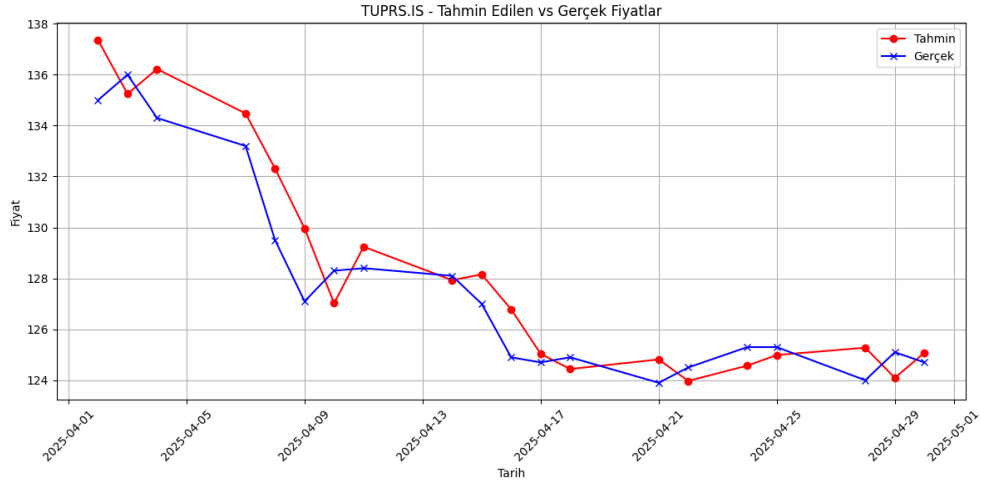
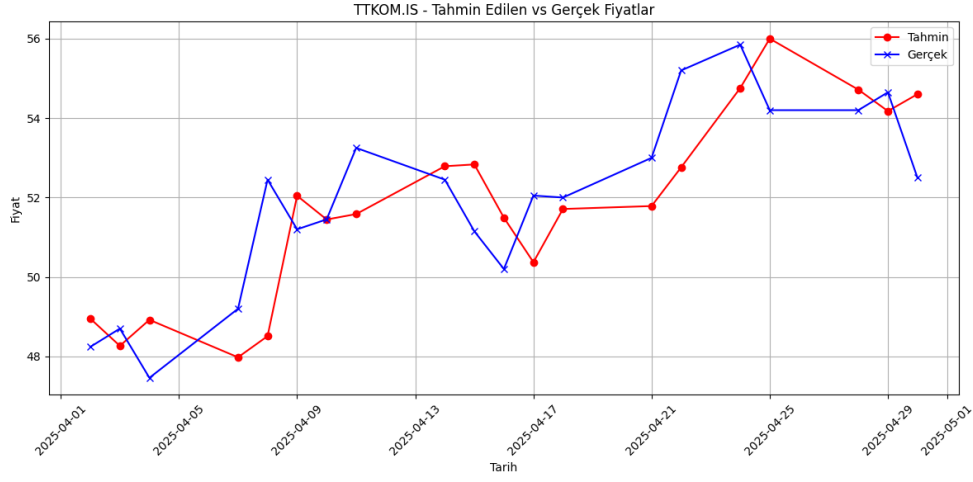


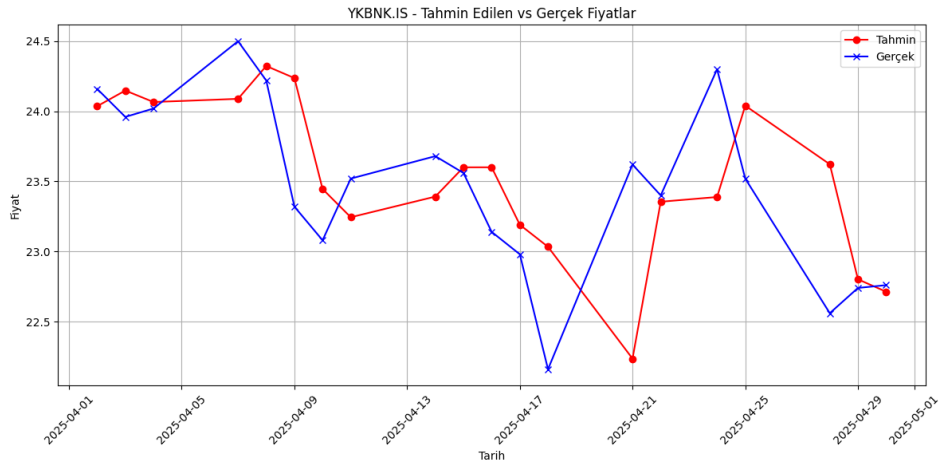




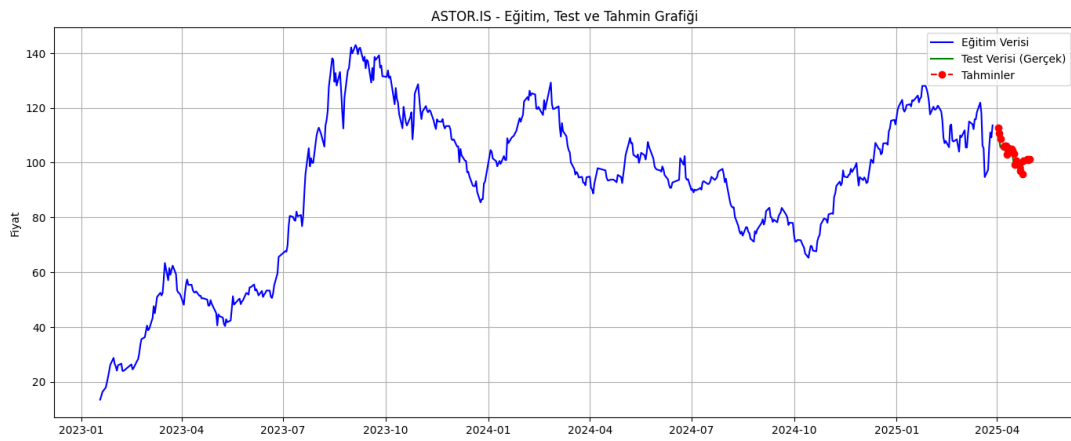
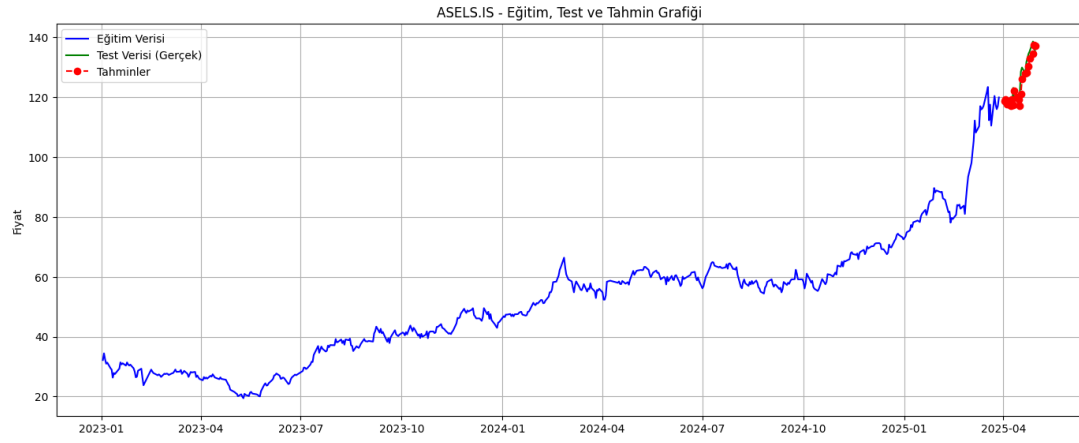
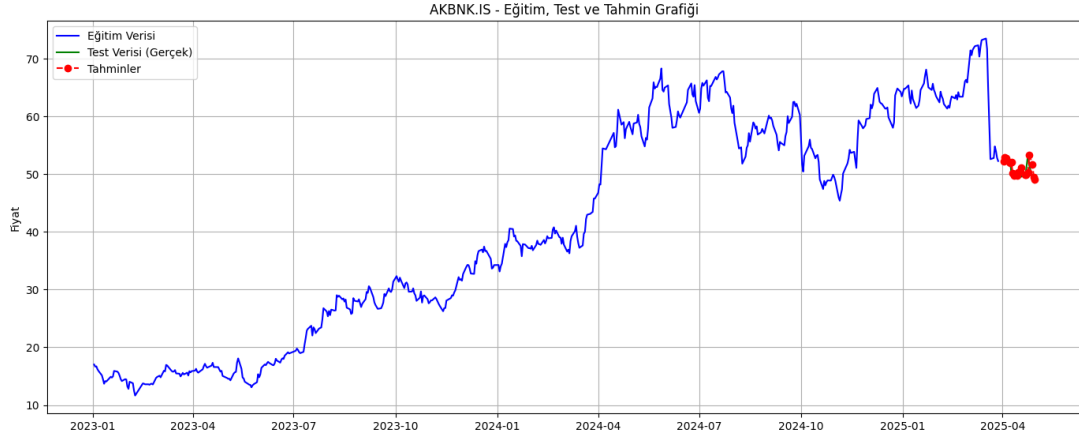


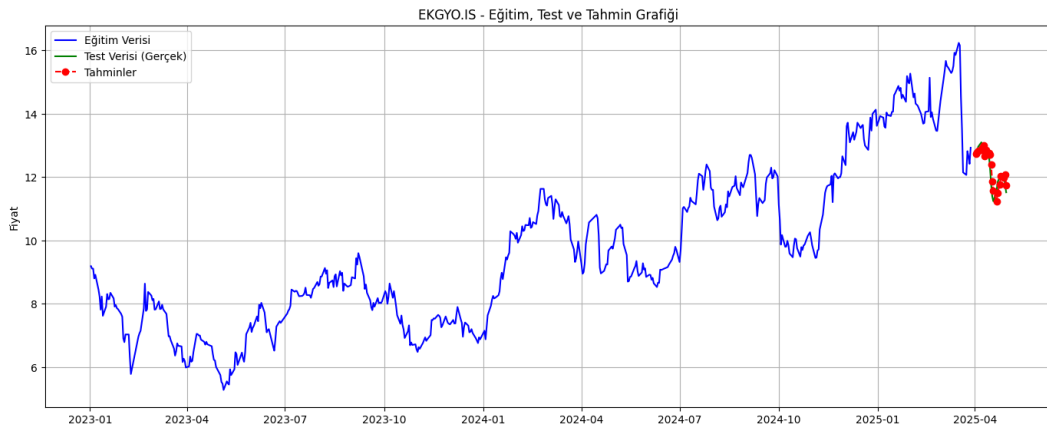
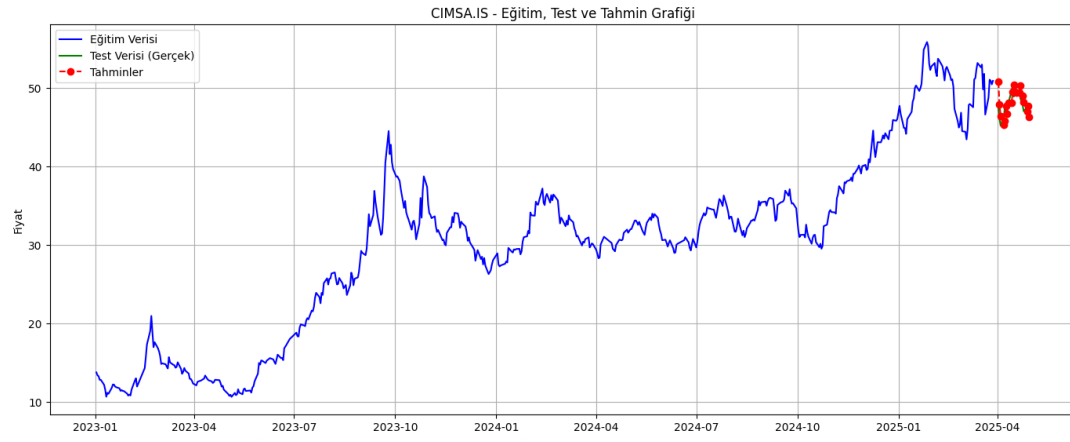
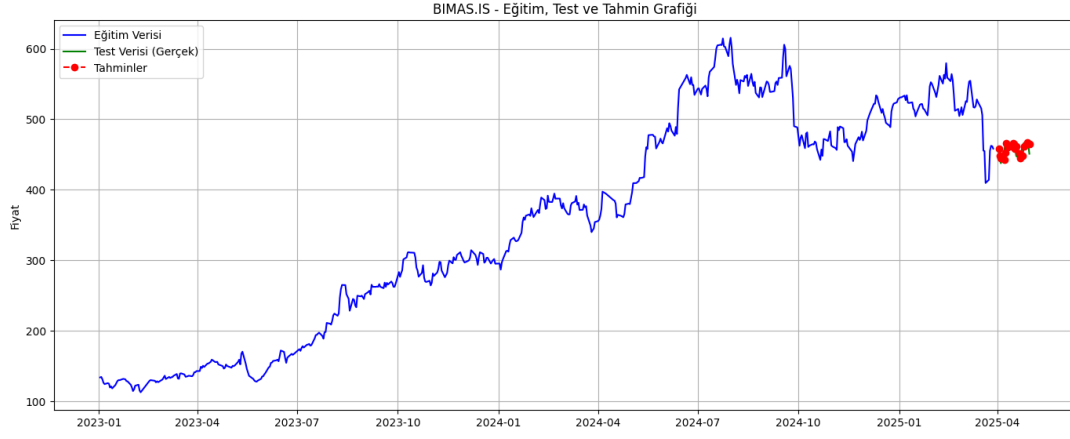


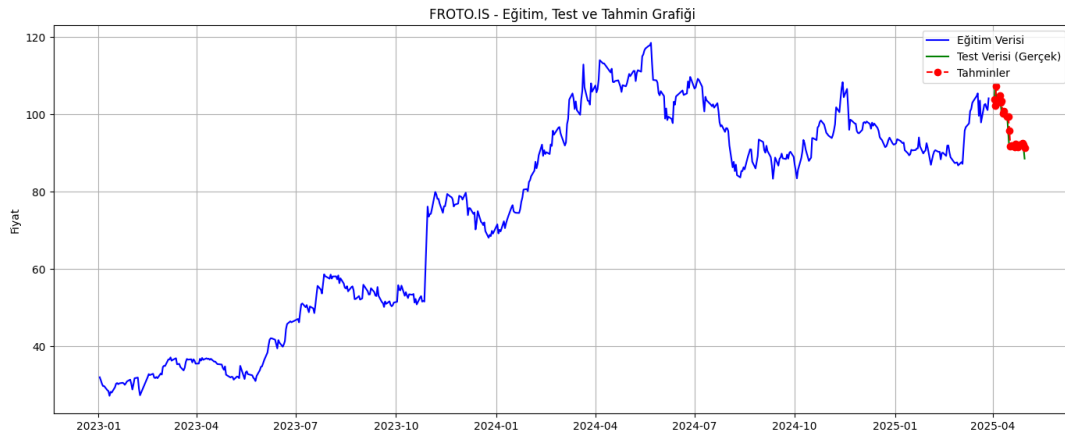
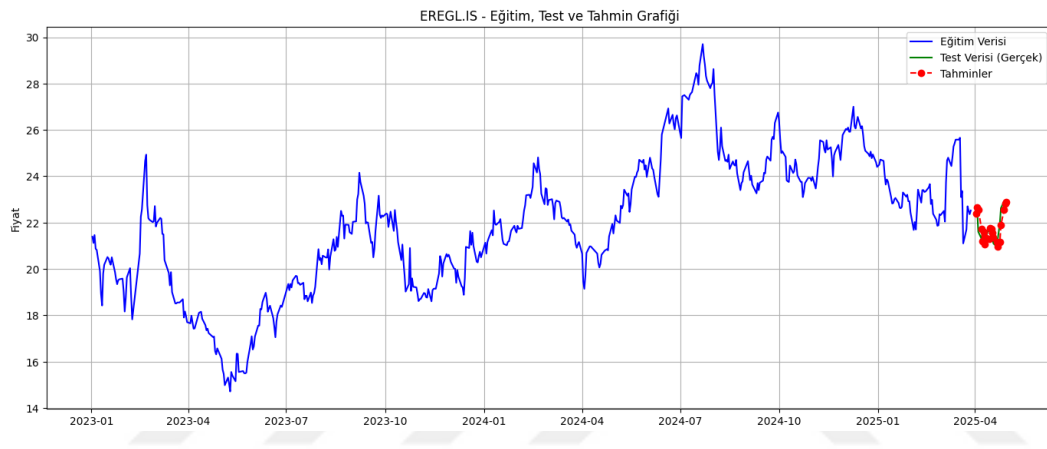
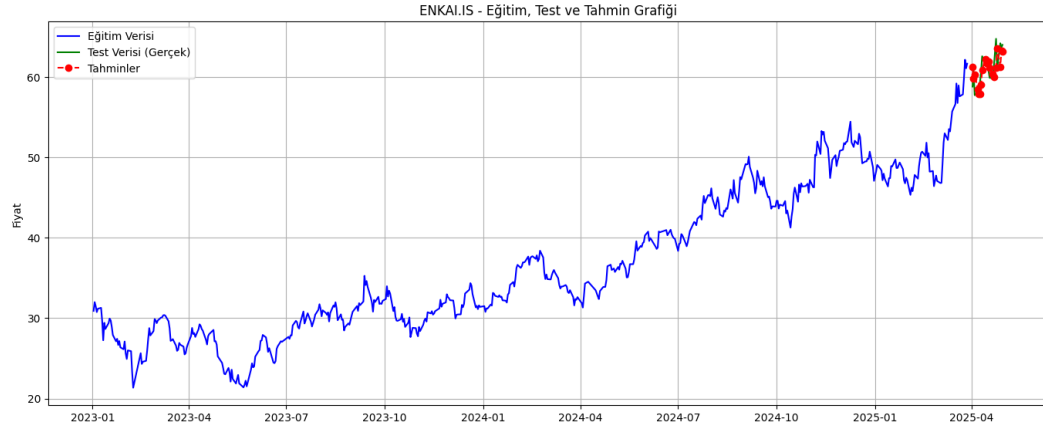


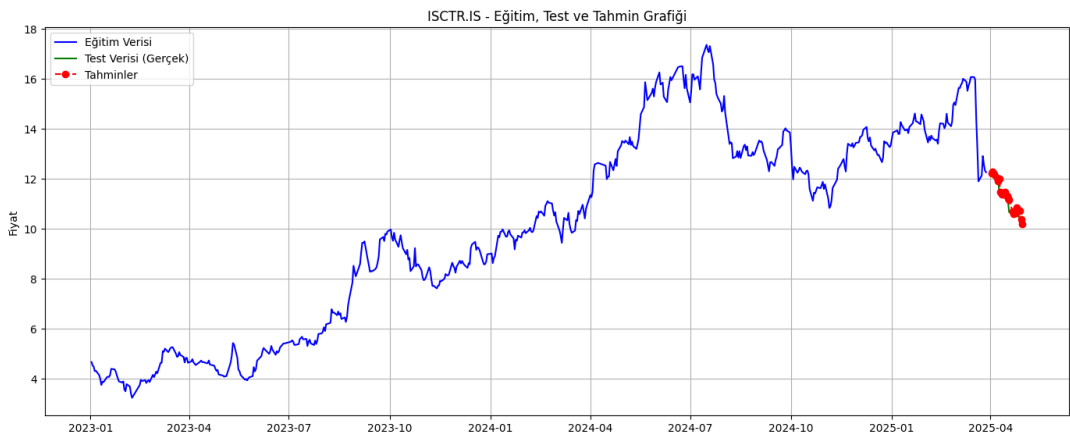
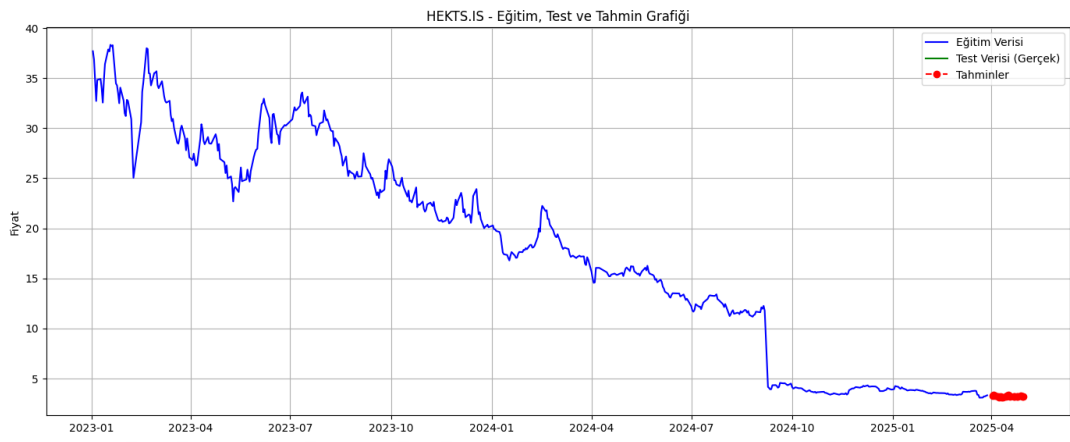
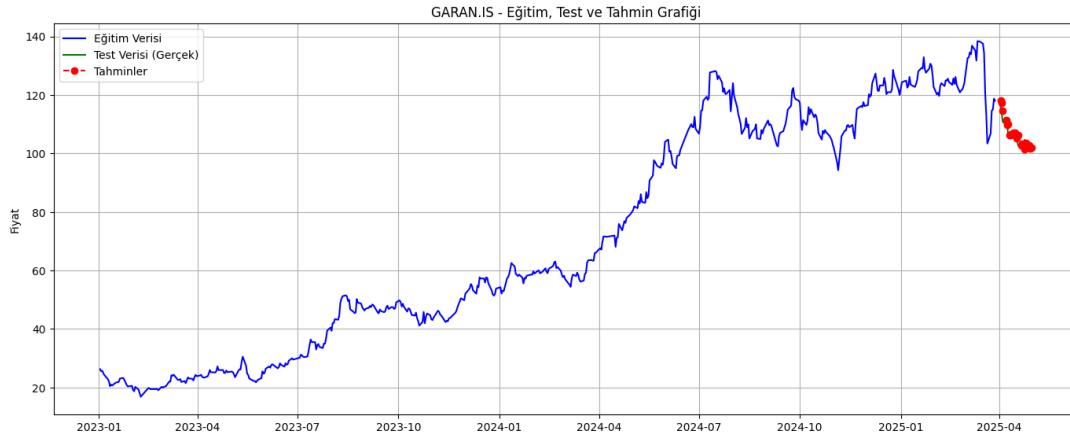


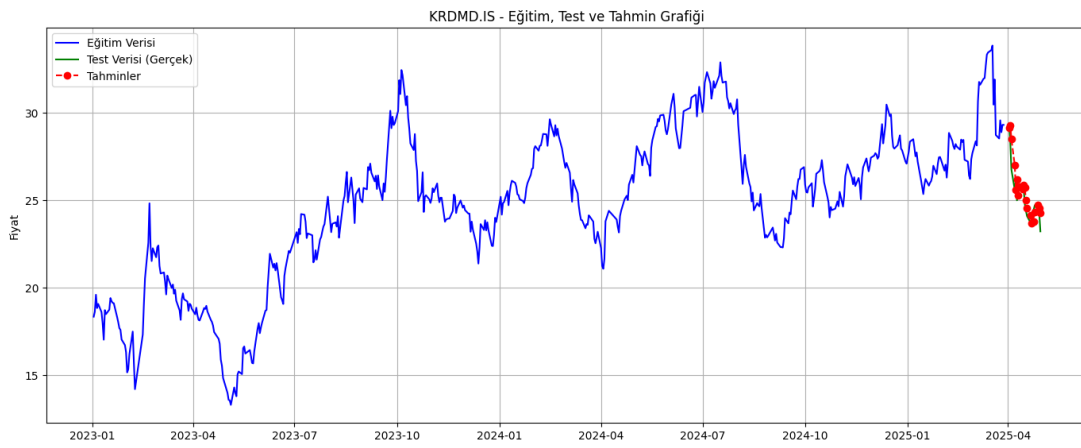
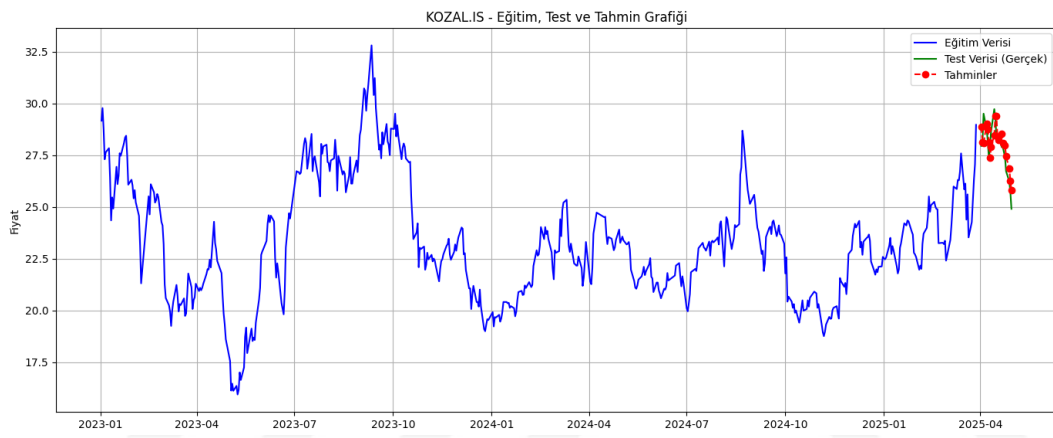
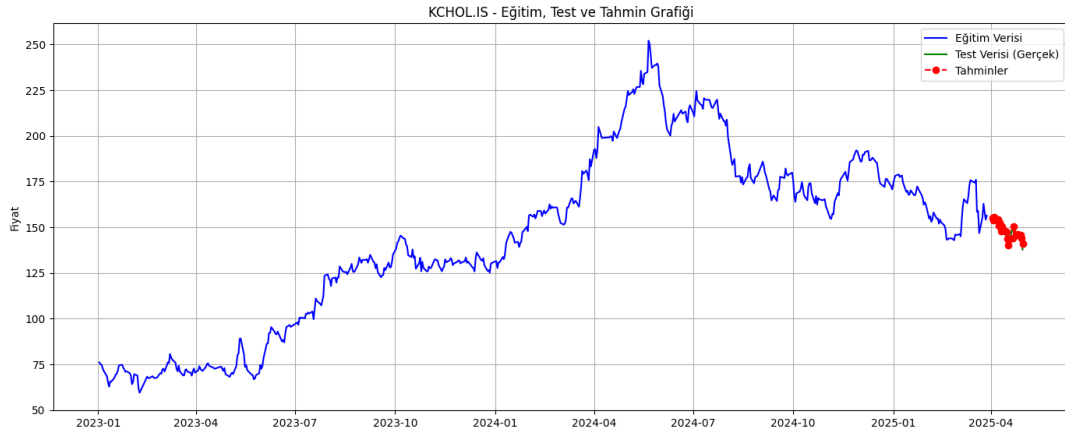
Ek 2. BİST 30 da yer alan hisse senetlerinin eğitim, test ve tahmin grafikleri

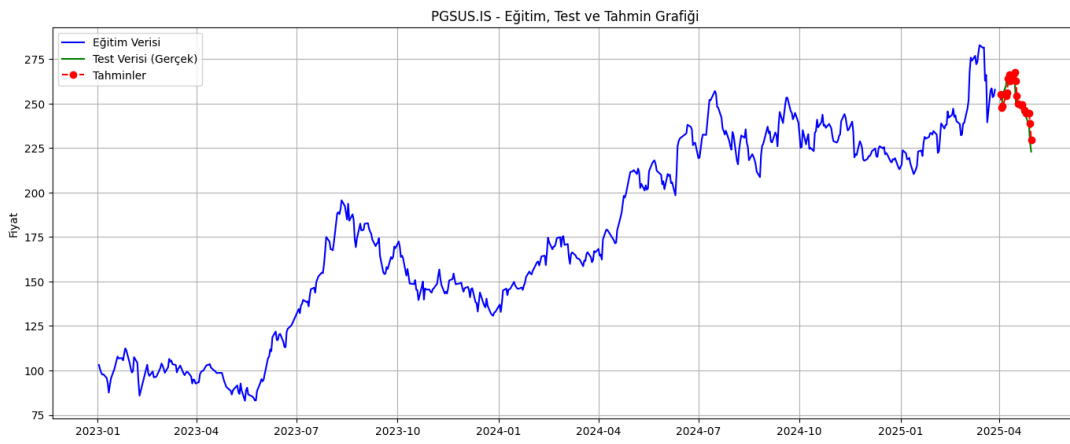
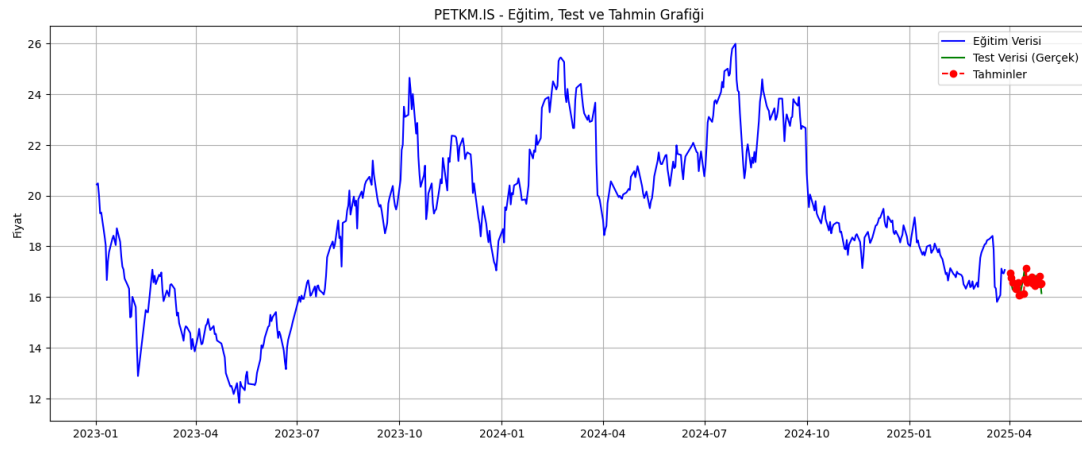
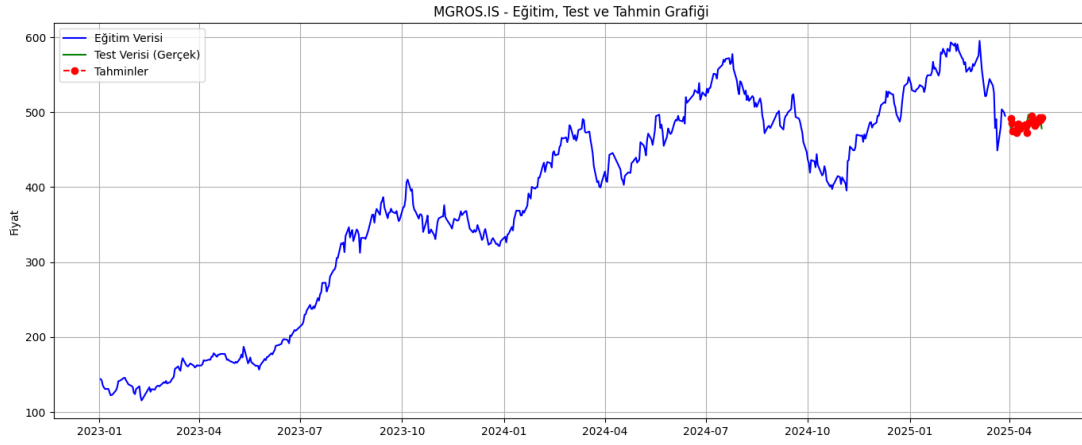


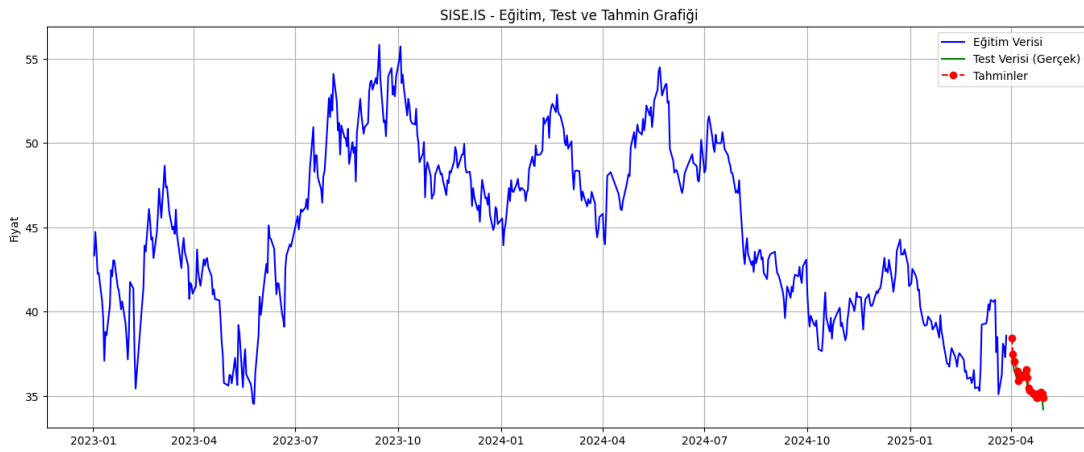
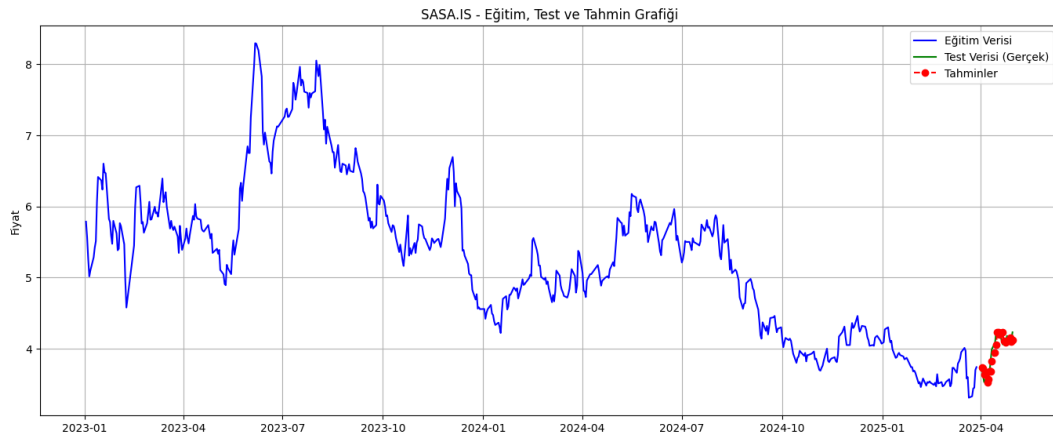
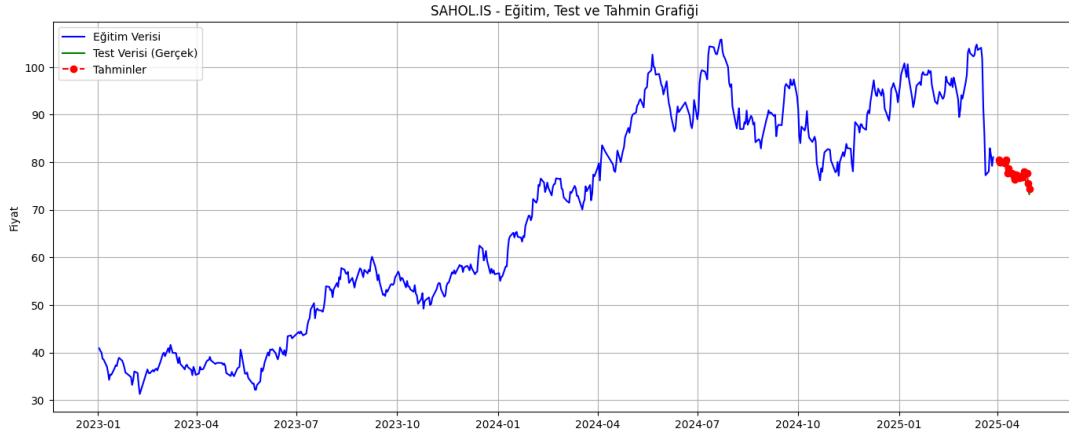


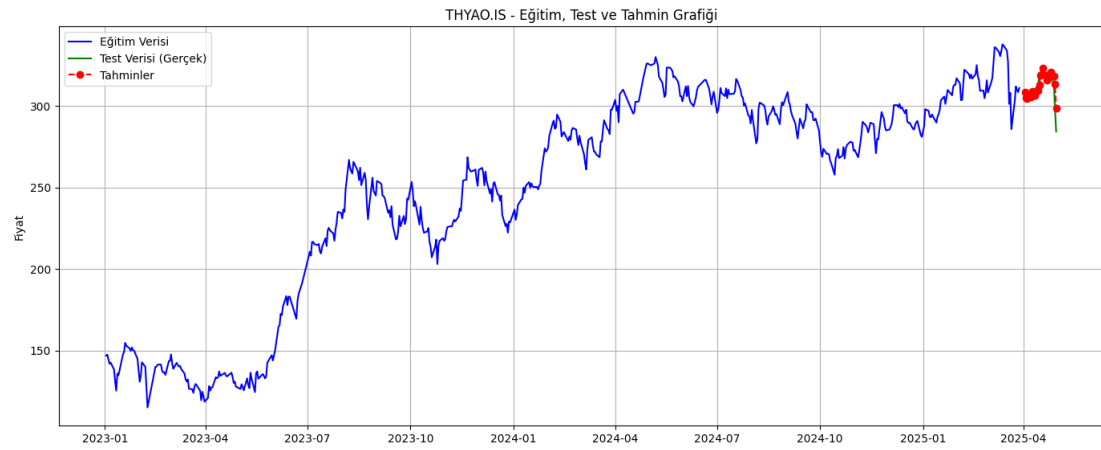
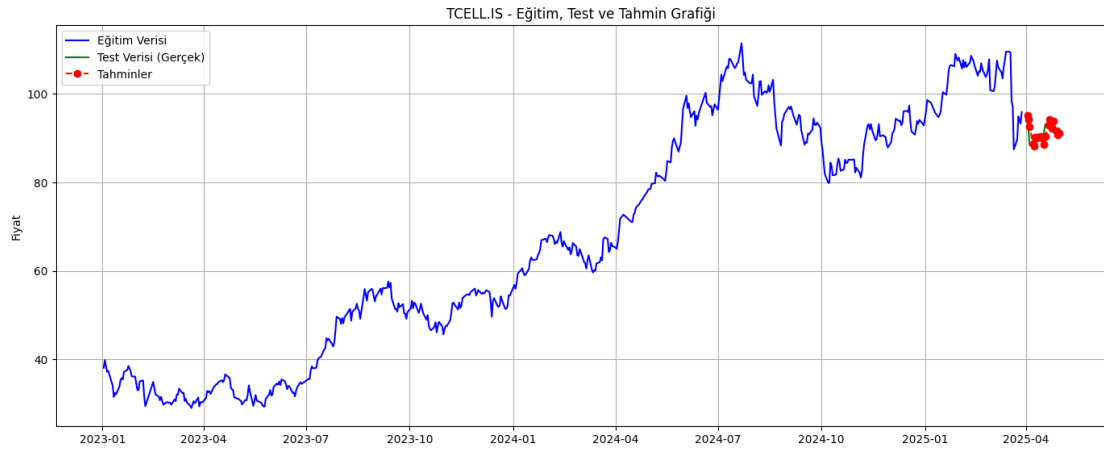
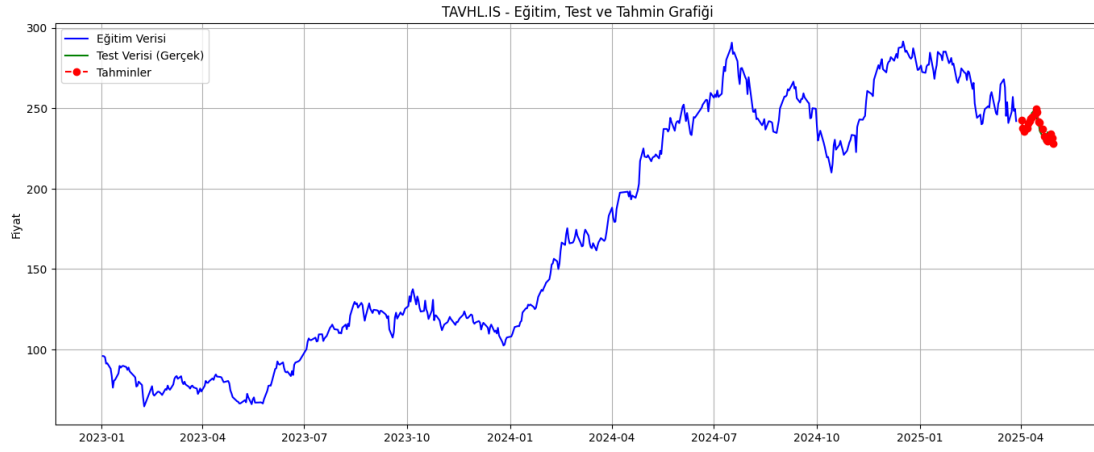


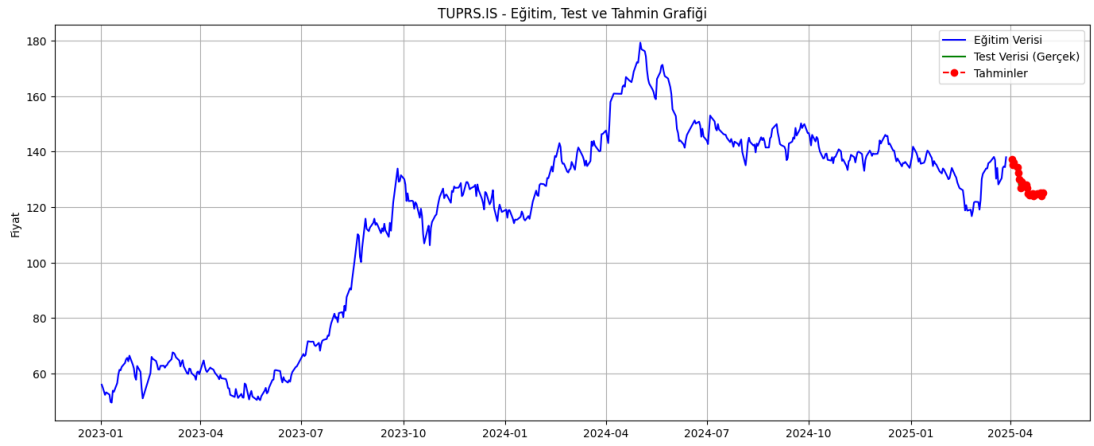
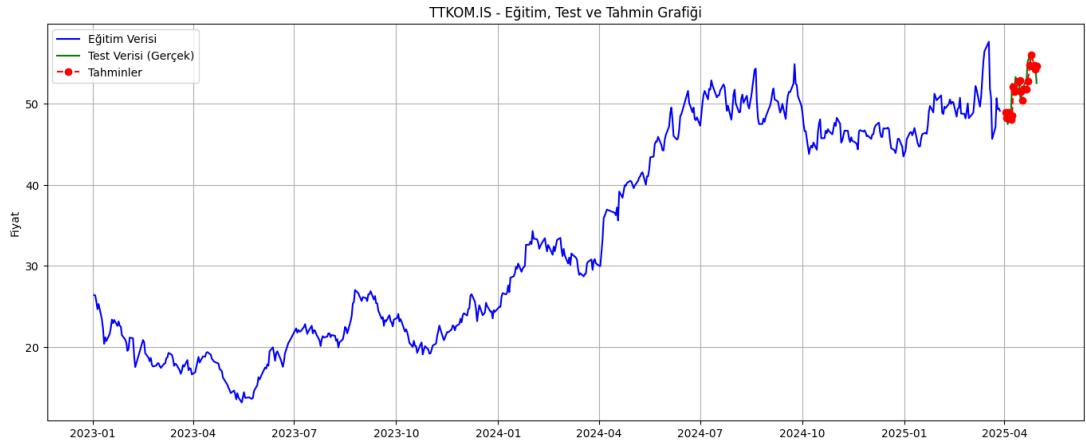
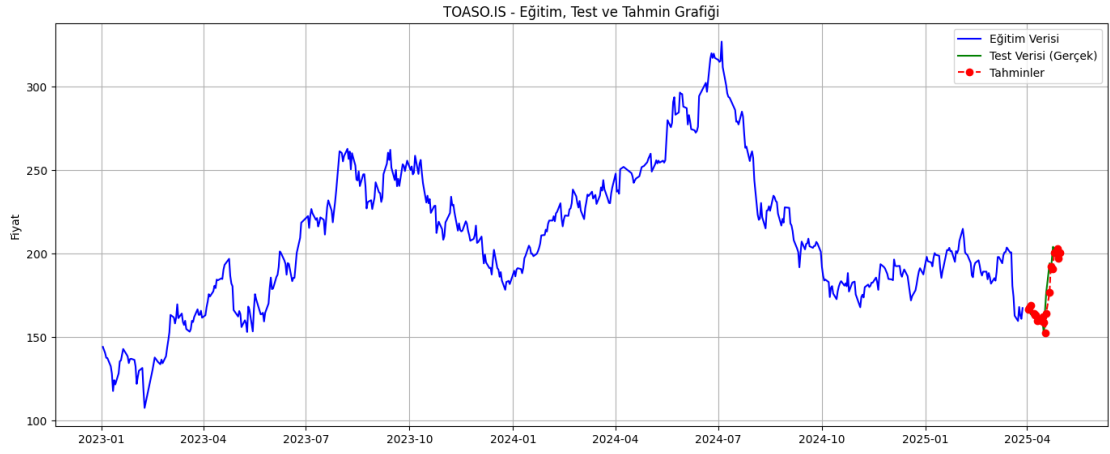


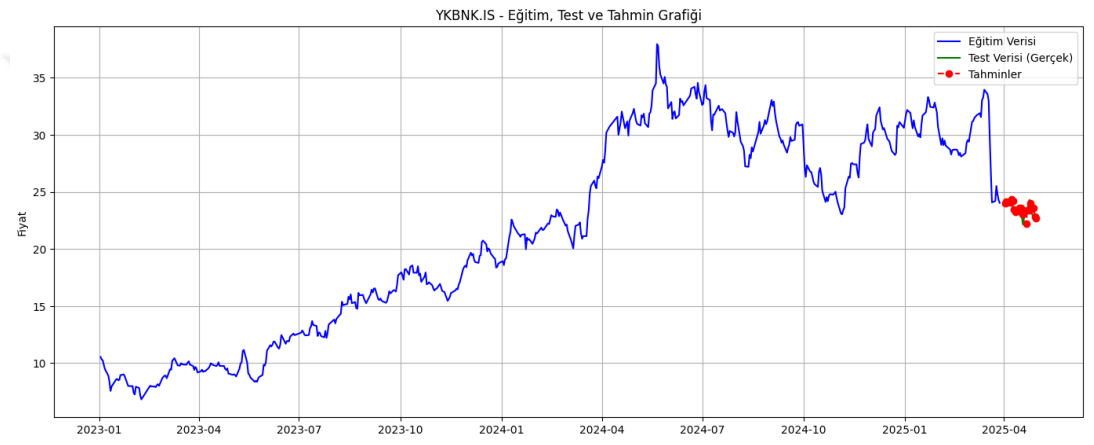
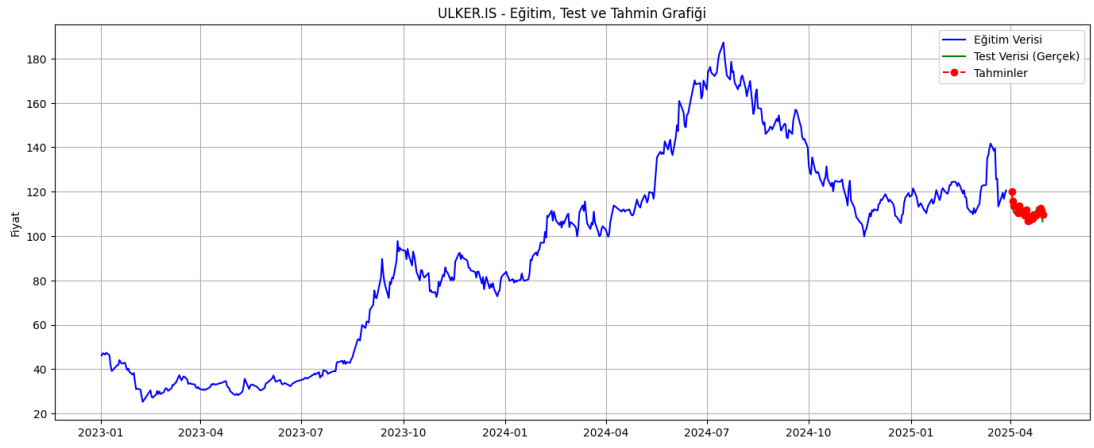












Ek 3. Kullanılan kodlar²

Tahmin kodları

```
#SVR kodları
import yfinance as yf
import pandas as pd
import numpy as np
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import cross_val_score, KFold
import matplotlib.pyplot as plt

# Pandas ayarları
pd.set_option('display.max_colwidth', None)

# BIST 30 hisseleri
tickers = [
    "AEFES.IS", "AKBNK.IS", "ASELS.IS", "ASTOR.IS", "BIMAS.IS", "CIMSA.IS", "EKGYO.IS",
    "ENKAI.IS",
    "EREGL.IS", "FROTO.IS", "GARAN.IS", "HEKTS.IS", "ISCTR.IS", "KCHOL.IS", "KOZAL.IS",
    "KRDMD.IS", "MGROS.IS", "PETKM.IS", "PGSUS.IS", "SAHOL.IS", "SASA.IS", "SISE.IS",
    "TAVHL.IS", "TCELL.IS",
    "THYAO.IS", "TOASO.IS", "TTKOM.IS", "TUPRS.IS", "ULKER.IS", "YKBNK.IS"
]

results = {}

for ticker in tickers:
    print(f"\n{'='*30}\nİşleniyor: {ticker}")

    # 1. Verilerin İndirilmesi
    data = yf.download(ticker, start="2023-01-02", end="2025-04-01")[['Open', 'High', 'Low', 'Close',
    'Volume']]

    # Veri boşsa geç
    if data.empty or len(data) < 50:
        print(f"{ticker} için yeterli veri yok, atlanıyor.")
        continue

    # 2. Eğitim Verisinin Hazırlanması
    train_data = data.copy()
    X_train = train_data.drop(columns=["Close"])
    y_train = train_data["Close"]

    # 3. Modelin Eğitimi (SVR kullanıyoruz)
```

² Kodlarda yer alan bazı parametreler yatırım amaçlı bir yönlendirme yapılmaması adına *** ile gizlenmiştir.

```

svr = SVR(C=***, epsilon=***, kernel=***)
svr.fit(X_train, y_train)

# 4. Gelecek 20 iş günü verisi
real_future_data = yf.download(ticker, start="2025-04-02", end="2025-05-01")[['Open', 'High', 'Low',
'Close', 'Volume']]
real_future_data = real_future_data.iloc[:20]
future_dates = real_future_data.index

if real_future_data.empty or len(real_future_data) < 20:
print(f"{ticker} için yeterli gelecek veri yok, atlanıyor.")
continue

# 5. Gelecek Tahminleri (Model bir kez eğitildi, gerçek değerler kullanılmıyor)
future_predictions = []
last_known_data = data.iloc[-1:].copy()

for i, date in enumerate(future_dates):
input_features = last_known_data.drop(columns=["Close"])
predicted_close = svr.predict(input_features)[0]
future_predictions.append(predicted_close)

# Bir sonraki tahmin için sadece tahmin edilen değerleri kullan
last_known_data = pd.DataFrame({
'Open': [input_features['Open'].values[0]],
'High': [input_features['High'].values[0]],
'Low': [input_features['Low'].values[0]],
'Volume': [input_features['Volume'].values[0]],
'Close': [predicted_close]# Tahmin edilen değeri kullanıyoruz
})

# 6. Tahmin ve Gerçek Veriler
real_prices = real_future_data["Close"].values
comparison_df = pd.DataFrame({
"Tarih": future_dates,
"Tahmin Edilen": future_predictions,
"Gerçek Değer": real_prices.flatten() if len(real_prices.shape) > 1 else real_prices
})
print(f"\n{ticker} İçin Tahmin vs Gerçek:")
print(comparison_df)

# 7. Grafik
plt.figure(figsize=(12, 6))
plt.plot(future_dates, future_predictions, label="Tahmin", color='red', marker='o')
plt.plot(future_dates, real_prices, label='Gerçek', color='blue', marker='x')
plt.title(f"{ticker} - Tahmin Edilen vs Gerçek Fiyatlar")
plt.xlabel("Tarih")
plt.ylabel("Fiyat")

```

```

plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# 8. Tahmin Doğruluğu
tahmin_dogrulugu = 100 * (1 - np.mean(np.abs((real_prices - future_predictions) / real_prices)))
print(f"{ticker} Tahmin Doğruluğu: %{tahmin_dogrulugu:.2f}")

# 9. Overfitting Analizi
y_train_pred = svr.predict(X_train)
train_rmse = np.sqrt(mean_squared_error(y_train, y_train_pred))
test_rmse = np.sqrt(mean_squared_error(real_prices, future_predictions))
train_r2 = r2_score(y_train, y_train_pred)
test_r2 = r2_score(real_prices, future_predictions)

delta_rmse = abs(float(train_rmse) - float(test_rmse))
delta_r2 = abs(float(train_r2) - float(test_r2))
rmse_threshold = 0.10 * float(y_train.mean())

overfitting_flag = (delta_rmse > rmse_threshold) and (delta_r2 > 0.1)

if overfitting_flag:
    print(f"⚠ Overfitting VAR! (ΔRMSE={delta_rmse:.2f}, ΔR²={delta_r2:.2f})")
else:
    print(f"✅ Overfitting YOK (ΔRMSE={delta_rmse:.2f}, ΔR²={delta_r2:.2f})")

# 9.2 K-Fold Cross-Validation (5-fold)
try:
    kf = KFold(n_splits=5, shuffle=True, random_state=42)
    svr_cv = SVR()
    rmse_scores = -cross_val_score(svr_cv, X_train, y_train, cv=kf,
                                   scoring='neg_root_mean_squared_error')
    r2_scores = cross_val_score(svr_cv, X_train, y_train, cv=kf, scoring='r2')

    print(f"{ticker} 5-Fold CV RMSE Ortalaması: {rmse_scores.mean():.4f}")
    print(f"{ticker} 5-Fold CV R² Ortalaması: {r2_scores.mean():.4f}")
except:
    print(f"{ticker} için K-Fold Cross-Validation hesaplanamadı.")

# 10. Sonuçları Sakla
results[ticker] = {
    "Tahmin Doğruluğu (%)": tahmin_dogrulugu,
    "Overfitting": "VAR" if overfitting_flag else "YOK",
    "Tahmin": future_predictions,
    "Gerçek": real_prices.tolist()
}

```

11. Genel Doğruluk

```
overall_accuracy = np.mean([results[t]["Tahmin Doğruluğu (%)"] for t in results])
print(f"\nModelin Genel Tahmin Doğruluğu Ortalaması: %{overall_accuracy:.2f}%")
```

12. Sonuçları DataFrame'e çevir

```
results_df = pd.DataFrame.from_dict(results, orient='index')
print("\nSonuçlar Özeti:")
print(results_df)
```

#LSTM kodları

```
import yfinance as yf
import pandas as pd
import numpy as np
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import cross_val_score, KFold
from keras.models import Sequential
from keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
```

Pandas ayarları

```
pd.set_option('display.max_colwidth', None)
```

BIST 30 hisseleri

```
tickers = [
    "AEFES.IS", "AKBNK.IS", "ASELS.IS", "ASTOR.IS", "BIMAS.IS", "CIMSA.IS", "EKGYO.IS",
    "ENKA.IS",
    "EREGL.IS", "FROTO.IS", "GARAN.IS", "HEKTS.IS", "ISCTR.IS", "KCHOL.IS", "KOZAL.IS",
    "KRDMD.IS", "MGROS.IS", "PETKM.IS", "PGSUS.IS", "SAHOL.IS", "SASA.IS", "SISE.IS",
    "TAVHL.IS", "TCELL.IS",
    "THYAO.IS", "TOASO.IS", "TTKOM.IS", "TUPRS.IS", "ULKER.IS", "YKBNK.IS"
]
```

```
results = {}
```

```
def create_lstm_model(input_shape):
```

```
    model = Sequential()
    model.add(LSTM(***, return_sequences=True, input_shape=input_shape))
    model.add(LSTM(***))
    model.add(Dense(1))
    model.compile(optimizer=***, loss=***)
    return model
```

```
for ticker in tickers:
```

```
    print(f"\n{'='*30}\nİşleniyor: {ticker}")
```

1. Verilerin İndirilmesi

```

data = yf.download(ticker, start="2023-01-02", end="2025-04-01")[['Open', 'High', 'Low', 'Close',
'Volume']]

# Veri boşsa geç
if data.empty or len(data) < 50:
print(f"{ticker} için yeterli veri yok, atlanıyor.")
continue

# 2. Verilerin Ölçeklendirilmesi
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(data)

# 3. Eğitim Verisinin Hazırlanması
X_train = []
y_train = []
look_back = ***

for i in range(look_back, len(scaled_data)):
X_train.append(scaled_data[i-look_back:i, :])
y_train.append(scaled_data[i, 3])# Close price

X_train, y_train = np.array(X_train), np.array(y_train)

# 4. LSTM Modelinin Eğitimi (SADECE BİR KEZ)
model = create_lstm_model((X_train.shape[1], X_train.shape[2]))
model.fit(X_train, y_train, epochs=***, batch_size=***, verbose=0)
print(f"{ticker} için LSTM modeli eğitildi")

# 5. Gelecek 20 iş günü verisi
real_future_data = yf.download(ticker, start="2025-04-02", end="2025-05-01")[['Open', 'High', 'Low',
'Close', 'Volume']]
real_future_data = real_future_data.iloc[:20]
future_dates = real_future_data.index

if real_future_data.empty or len(real_future_data) < 20:
print(f"{ticker} için yeterli gelecek veri yok, atlanıyor.")
continue

# 6. Gelecek Tahminleri (Model güncellenmeden)
future_predictions = []
last_sequence = scaled_data[-look_back:]# Son 'look back' günlük veri

for i in range(len(real_future_data)):
# Tahmin yap
current_input = last_sequence.reshape(1, look_back, -1)
predicted_scaled = model.predict(current_input, verbose=0)[0][0]

# Tahmini gerçek ölçeğe dönüştür

```

```

dummy_row = np.zeros((1, scaled_data.shape[1]))
dummy_row[0, 3] = predicted_scaled# Close price
predicted_price = scaler.inverse_transform(dummy_row)[0, 3]
future_predictions.append(predicted_price)

# Sonraki gün için giriş hazırla (gerçek verilerle)
real_row = real_future_data.iloc[i]
# Ensure new_row is a 2D array (1, 5)
new_row = np.array([real_row[['Open', 'High', 'Low', 'Close', 'Volume']].values])

new_row_scaled = scaler.transform(new_row)
last_sequence = np.vstack([last_sequence[1:], new_row_scaled])

# 7. Tahmin ve Gerçek Veriler
real_prices = real_future_data["Close"].values
comparison_df = pd.DataFrame({
    "Tarih": future_dates,
    "Tahmin Edilen": future_predictions,
    "Gerçek Değer": real_prices.flatten() if len(real_prices.shape) > 1 else real_prices
})
print(f"\n{ticker} İçin Tahmin vs Gerçek:")
print(comparison_df)

# 8. Grafik
plt.figure(figsize=(12, 6))
plt.plot(future_dates, future_predictions, label='Tahmin', color='red', marker='o')
plt.plot(future_dates, real_prices, label='Gerçek', color='blue', marker='x')
plt.title(f'{ticker} - Tahmin Edilen vs Gerçek Fiyatlar')
plt.xlabel("Tarih")
plt.ylabel("Fiyat")
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# 9. Tahmin Doğruluğu
tahmin_dogrulugu = 100 * (1 - np.mean(np.abs((real_prices - future_predictions) / real_prices)))
print(f"{ticker} Tahmin Doğruluğu: %{tahmin_dogrulugu:.2f}")

# 10. Overfitting Analizi
y_train_pred = model.predict(X_train).flatten()
train_rmse = np.sqrt(mean_squared_error(y_train, y_train_pred))
test_rmse = np.sqrt(mean_squared_error(real_prices, future_predictions))
train_r2 = r2_score(y_train, y_train_pred)
test_r2 = r2_score(real_prices, future_predictions)

delta_rmse = abs(float(train_rmse) - float(test_rmse))

```

```

delta_r2 = abs(float(train_r2) - float(test_r2))
rmse_threshold = 0.10 * float(scaler.inverse_transform([[0,0,0,y_train.mean(),0]])[0][3])

overfitting_flag = (delta_rmse > rmse_threshold) and (delta_r2 > 0.1)

if overfitting_flag:
    print(f"⚠ Overfitting VAR! ( $\Delta$ RMSE={delta_rmse:.2f},  $\Delta$ R2= {delta_r2:.2f})")
else:
    print(f"✅ Overfitting YOK ( $\Delta$ RMSE={delta_rmse:.2f},  $\Delta$ R2= {delta_r2:.2f})")

# 11. K-Fold Cross-Validation (5-fold)
try:
    kf = KFold(n_splits=5, shuffle=True, random_state=42)
    rmse_scores = []
    r2_scores = []

    for train_index, test_index in kf.split(X_train):
        X_train_cv, X_test_cv = X_train[train_index], X_train[test_index]
        y_train_cv, y_test_cv = y_train[train_index], y_train[test_index]

        model_cv = create_lstm_model((X_train_cv.shape[1], X_train_cv.shape[2]))
        model_cv.fit(X_train_cv, y_train_cv, epochs=100, batch_size=32, verbose=0)

        y_pred_cv = model_cv.predict(X_test_cv).flatten()
        rmse_scores.append(np.sqrt(mean_squared_error(y_test_cv, y_pred_cv)))
        r2_scores.append(r2_score(y_test_cv, y_pred_cv))

    print(f"{ticker} 5-Fold CV RMSE Ortalaması: {np.mean(rmse_scores):.4f}")
    print(f"{ticker} 5-Fold CV R2 Ortalaması: {np.mean(r2_scores):.4f}")
except Exception as e:
    print(f"{ticker} için K-Fold Cross-Validation hesaplanamadı. Hata: {str(e)}")

# 12. Sonuçları Sakla
results[ticker] = {
    "Tahmin Doğruluğu (%)": tahmin_dogruluğu,
    "Overfitting": "VAR" if overfitting_flag else "YOK",
    "Tahmin": future_predictions,
    "Gerçek": real_prices.tolist()
}

# 13. Genel Doğruluk
overall_accuracy = np.mean([results[t]["Tahmin Doğruluğu (%)"] for t in results])
print(f"📊 Modelin Genel Tahmin Doğruluğu Ortalaması: %{overall_accuracy:.2f}")

# 14. Sonuçları DataFrame'e çevir
results_df = pd.DataFrame.from_dict(results, orient='index')
print("\nSonuçlar Özeti:")

```

```
print(results_df)
```

```
# XGBoost kodları
import yfinance as yf
import pandas as pd
import numpy as np
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import cross_val_score, KFold
import matplotlib.pyplot as plt

# Pandas ayarları
pd.set_option('display.max_colwidth', None)

# BIST 30 hisseleri
tickers = [
    "AEFES.IS", "AKBNK.IS", "ASELS.IS", "ASTOR.IS", "BIMAS.IS", "CIMSA.IS", "EKGYO.IS",
    "ENKAI.IS",
    "EREGL.IS", "FROTO.IS", "GARAN.IS", "HEKTS.IS", "ISCTR.IS", "KCHOL.IS", "KOZAL.IS",
    "KRDMD.IS", "MGROS.IS", "PETKM.IS", "PGSUS.IS", "SAHOL.IS", "SASA.IS", "SISE.IS",
    "TAVHL.IS", "TCELL.IS",
    "THYAO.IS", "TOASO.IS", "TTKOM.IS", "TUPRS.IS", "ULKER.IS", "YKBNK.IS"
]

results = {}

for ticker in tickers:
    print(f"\n{'='*30}\nİşleniyor: {ticker}")

# 1. Verilerin İndirilmesi
data = yf.download(ticker, start="2023-01-02", end="2025-04-01")[['Open', 'High', 'Low', 'Close',
'Volume']]

# Veri boşsa geç
if data.empty or len(data) < 50:
    print(f"{ticker} için yeterli veri yok, atlanıyor.")
    continue

# 2. Eğitim Verisinin Hazırlanması
train_data = data.copy()
X_train = train_data.drop(columns=["Close"])
y_train = train_data["Close"]

# 3. Modelin Eğitimi (tek sefer)
model = XGBRegressor(n_estimators=***,
```

```

max_depth=***, random_state=0, verbosity=0)
model.fit(X_train, y_train)

# 4. Gelecek 20 iş günü verisi
real_future_data = yf.download(ticker, start="2025-04-02", end="2025-05-01")[['Open', 'High', 'Low',
'Close', 'Volume']]
real_future_data = real_future_data.iloc[:20]
future_dates = real_future_data.index

if real_future_data.empty or len(real_future_data) < 20:
print(f"{ticker} için yeterli gelecek veri yok, atlanıyor.")
continue

# 5. Gelecek Tahminleri
future_predictions = model.predict(real_future_data.drop(columns=["Close"]))

# 6. Tahmin ve Gerçek Veriler
real_prices = real_future_data["Close"].values
comparison_df = pd.DataFrame({
"Tarih": future_dates,
"Tahmin Edilen": future_predictions,
"Gerçek Değer": real_prices.flatten() if len(real_prices.shape) > 1 else real_prices
})
print(f"\n{ticker} İçin Tahmin vs Gerçek:")
print(comparison_df)

# 7. Grafik
plt.figure(figsize=(12, 6))
plt.plot(future_dates, future_predictions, label='Tahmin', color='red', marker='o')
plt.plot(future_dates, real_prices, label='Gerçek', color='blue', marker='x')
plt.title(f"{ticker} - Tahmin Edilen vs Gerçek Fiyatlar')
plt.xlabel("Tarih")
plt.ylabel("Fiyat")
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# 8. Tahmin Doğruluğu
tahmin_dogruluğu = 100 * (1 - np.mean(np.abs((real_prices - future_predictions) / real_prices)))
print(f"{ticker} Tahmin Doğruluğu: %{tahmin_dogruluğu:.2f}")

# 9. Overfitting Analizi
y_train_pred = model.predict(X_train)
train_rmse = np.sqrt(mean_squared_error(y_train, y_train_pred))
test_rmse = np.sqrt(mean_squared_error(real_prices, future_predictions))
train_r2 = r2_score(y_train, y_train_pred)

```

```

test_r2 = r2_score(real_prices, future_predictions)

delta_rmse = abs(float(train_rmse) - float(test_rmse))
delta_r2 = abs(float(train_r2) - float(test_r2))
rmse_threshold = 0.10 * float(y_train.mean())

overfitting_flag = (delta_rmse > rmse_threshold) and (delta_r2 > 0.1)

if overfitting_flag:
    print(f"⚠ Overfitting VAR! ( $\Delta$ RMSE={delta_rmse:.2f},  $\Delta$ R2= {delta_r2:.2f})")
else:
    print(f"✅ Overfitting YOK ( $\Delta$ RMSE={delta_rmse:.2f},  $\Delta$ R2= {delta_r2:.2f})")

# 9.2 K-Fold Cross-Validation (5-fold)
try:
    kf = KFold(n_splits=5, shuffle=True, random_state=42)
    cv_model = XGBRegressor(random_state=0, verbosity=0)
    rmse_scores = -cross_val_score(cv_model, X_train, y_train, cv=kf,
    scoring='neg_root_mean_squared_error')
    r2_scores = cross_val_score(cv_model, X_train, y_train, cv=kf, scoring='r2')

    print(f"{ticker} 5-Fold CV RMSE Ortalaması: {rmse_scores.mean():.4f}")
    print(f"{ticker} 5-Fold CV R2 Ortalaması: {r2_scores.mean():.4f}")
except:
    print(f"{ticker} için K-Fold Cross-Validation hesaplanamadı.")

# 10. Sonuçları Sakla
results[ticker] = {
    "Tahmin Doğruluğu (%)": tahmin_dogrulugu,
    "Overfitting": "VAR" if overfitting_flag else "YOK",
    "Tahmin": future_predictions.tolist(),
    "Gerçek": real_prices.tolist()
}

# 11. Genel Doğruluk
overall_accuracy = np.mean([results[t]["Tahmin Doğruluğu (%)"] for t in results])
print(f"📊 Modelin Genel Tahmin Doğruluğu Ortalaması: %{overall_accuracy:.2f}")

# 12. Sonuçları DataFrame'e çevir
results_df = pd.DataFrame.from_dict(results, orient='index')
print("\nSonuçlar Özeti:")
print(results_df)

# 13. Performans Metrikleri (MAPE, MAE, RMSE, MSE, R2)
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

for ticker in results:
    y_true = np.array(results[ticker]["Gerçek"])
    y_pred = np.array(results[ticker]["Tahmin"])

```

```

mae = mean_absolute_error(y_true, y_pred)
mse = mean_squared_error(y_true, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_true, y_pred)
mape = np.mean(np.abs((y_true - y_pred) / y_true)) * 100

```

```

print(f"\n📊 {ticker} Performans Metrikleri:")
print(f"MAPE: %{mape:.2f}")
print(f"MAE: {mae:.4f}")
print(f"RMSE: {rmse:.4f}")
print(f"MSE: {mse:.4f}")
print(f"R²: {r2:.4f}")

```

#RF kodları

```

import yfinance as yf
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import cross_val_score, KFold
import matplotlib.pyplot as plt

```

Pandas ayarları

```
pd.set_option('display.max_colwidth', None)
```

BIST 30 hisseleri

```

tickers = [
    "AEFES.IS", "AKBNK.IS", "ASELS.IS", "ASTOR.IS", "BIMAS.IS", "CIMSA.IS", "EKGYO.IS",
    "ENKAI.IS",
    "EREGL.IS", "FROTO.IS", "GARAN.IS", "HEKTS.IS", "ISCTR.IS", "KCHOL.IS", "KOZAL.IS",
    "KRDMD.IS", "MGROS.IS", "PETKM.IS", "PGSUS.IS", "SAHOL.IS", "SASA.IS", "SISE.IS",
    "TAVHL.IS", "TCELL.IS",
    "THYAO.IS", "TOASO.IS", "TTKOM.IS", "TUPRS.IS", "ULKER.IS", "YKBNK.IS"]

```

```
results = {}
```

```

for ticker in tickers:
    print(f"\n{'='*30}\nİşleniyor: {ticker}")

```

1. Verilerin İndirilmesi

```
data = yf.download(ticker, start="2023-01-02", end="2025-04-01")[['Open', 'High', 'Low', 'Close', 'Volume']]
```

Veri boşsa geç

```

if data.empty or len(data) < 50:
    print(f"{ticker} için yeterli veri yok, atlanıyor.")
    continue

```

2. Eğitim Verisinin Hazırlanması

```
train_data = data.copy()
X_train = train_data.drop(columns=["Close"])
y_train = train_data["Close"]
```

3. Modelin İlk Eğitimi

```
rf = RandomForestRegressor(n_estimators=***, max_depth=***, random_state=0)
rf.fit(X_train, y_train)
```

4. Gelecek 20 iş günü verisi

```
real_future_data = yf.download(ticker, start="2025-04-02", end="2025-05-01")[['Open', 'High', 'Low',
'Close', 'Volume']]
real_future_data = real_future_data.iloc[:20]
future_dates = real_future_data.index
```

```
if real_future_data.empty or len(real_future_data) < 20:
    print(f"{ticker} için yeterli gelecek veri yok, atlanıyor.")
    continue
```

5. Gelecek Tahminleri

```
future_predictions = []
update_counter = 0
last_known_data = data.iloc[-1:].copy()

for i, date in enumerate(future_dates):
    input_features = last_known_data.drop(columns=["Close"])
    predicted_close = rf.predict(input_features)[0]
    future_predictions.append(***)

    real_row = real_future_data.loc[date]
    last_known_data = pd.DataFrame({
        'Open': [real_row['Open']],
        'High': [real_row['High']],
        'Low': [real_row['Low']],
        'Volume': [real_row['Volume']],
        'Close': [real_row['Close']]
    })
    update_counter += 1
    if update_counter % *** == 0 or i == len(future_dates) - 1:
        new_data = real_future_data.iloc[i+1:]
        updated_train_data = pd.concat([data, new_data])
        X_train = updated_train_data.drop(columns=["Close"])
        y_train = updated_train_data["Close"]
        rf = RandomForestRegressor(random_state=0)
        rf.fit(X_train, y_train)
        print(f"{ticker} - {date.date()}: Model güncellendi ({update_counter}. tahmin)")
```

```

# 6. Tahmin ve Gerçek Veriler
real_prices = real_future_data["Close"].values
comparison_df = pd.DataFrame({
    "Tarih": future_dates,
    "Tahmin Edilen": future_predictions,
    "Gerçek Değer": real_prices.flatten() if len(real_prices.shape) > 1 else real_prices
})
print(f"\n{ticker} İçin Tahmin vs Gerçek:")
print(comparison_df)

# 7. Grafik
plt.figure(figsize=(12, 6))
plt.plot(future_dates, future_predictions, label='Tahmin', color='red', marker='o')
plt.plot(future_dates, real_prices, label='Gerçek', color='blue', marker='x')
plt.title(f'{ticker} - Tahmin Edilen vs Gerçek Fiyatlar')
plt.xlabel("Tarih")
plt.ylabel("Fiyat")
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# 8. Tahmin Doğruluğu
tahmin_dogrulugu = 100 * (1 - np.mean(np.abs((real_prices - future_predictions) / real_prices)))
print(f'{ticker} Tahmin Doğruluğu: %{tahmin_dogrulugu:.2f}%')

# 9. Overfitting Analizi
y_train_pred = rf.predict(X_train)
train_rmse = np.sqrt(mean_squared_error(y_train, y_train_pred))
test_rmse = np.sqrt(mean_squared_error(real_prices, future_predictions))
train_r2 = r2_score(y_train, y_train_pred)
test_r2 = r2_score(real_prices, future_predictions)

delta_rmse = abs(float(train_rmse) - float(test_rmse))
delta_r2 = abs(float(train_r2) - float(test_r2))
rmse_threshold = 0.10 * float(y_train.mean())

overfitting_flag = (delta_rmse > rmse_threshold) and (delta_r2 > 0.1)

if overfitting_flag:
    print(f"⚠ Overfitting VAR! ( $\Delta RMSE$ =:{delta_rmse:.2f},  $\Delta R^2$ =:{delta_r2:.2f})")
else:
    print(f"✅ Overfitting YOK ( $\Delta RMSE$ =:{delta_rmse:.2f},  $\Delta R^2$ =:{delta_r2:.2f})")

# 9.1 Out-of-Bag (OOB) Hatası
try:

```

```

rf_oob = RandomForestRegressor(random_state=0, oob_score=True, bootstrap=True)
rf_oob.fit(X_train, y_train)
print(f"{ticker} OOB R2 Skoru: {rf_oob.oob_score_:.4f}")
except:
print(f"{ticker} için OOB skoru hesaplanamadı.")

# 9.2 K-Fold Cross-Validation (5-fold)
try:
kf = KFold(n_splits=5, shuffle=True, random_state=0)
rf_cv = RandomForestRegressor(random_state=0)
rmse_scores = -cross_val_score(rf_cv, X_train, y_train, cv=kf, scoring='neg_root_mean_squared_error')
r2_scores = cross_val_score(rf_cv, X_train, y_train, cv=kf, scoring='r2')

print(f"{ticker} 5-Fold CV RMSE Ortalaması: {rmse_scores.mean():.4f}")
print(f"{ticker} 5-Fold CV R2 Ortalaması: {r2_scores.mean():.4f}")
except:
print(f"{ticker} için K-Fold Cross-Validation hesaplanamadı.")

# 10. Sonuçları Sakla
results[ticker] = {
    "Tahmin Doğruluğu (%)": tahmin_dogrulugu,
    "Overfitting": "VAR" if overfitting_flag else "YOK",
    "Tahmin": future_predictions,
    "Gerçek": real_prices.tolist()
}

# 11. Genel Doğruluk
overall_accuracy = np.mean([results[t]["Tahmin Doğruluğu (%)"] for t in results])
print(f"\n📊 Modelin Genel Tahmin Doğruluğu Ortalaması: %{overall_accuracy:.2f}")

# 12. Sonuçları DataFrame'e çevir
results_df = pd.DataFrame.from_dict(results, orient='index')
print("\nSonuçlar Özeti:")
print(results_df)

# 13. Performans Metrikleri (MAPE, MAE, RMSE, MSE, R2)
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

for ticker in results:
    y_true = np.array(results[ticker]["Gerçek"])
    y_pred = np.array(results[ticker]["Tahmin"])

    mae = mean_absolute_error(y_true, y_pred)
    mse = mean_squared_error(y_true, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_true, y_pred)
    mape = np.mean(np.abs((y_true - y_pred) / y_true)) * 100

print(f"\n📊 {ticker} Performans Metrikleri:")

```

```

print(f"MAPE: %{mape:.2f}")
print(f"MAE: {mae:.4f}")
print(f"RMSE: {rmse:.4f}")
print(f"MSE: {mse:.4f}")
print(f"R²: {r2:.4f}")

import matplotlib.pyplot as plt
from matplotlib import gridspec

for ticker in results:
    # Hisse verilerini yükle
    full_data = yf.download(ticker, start="2023-01-02", end="2025-05-01")[['Close']]
    train_data = full_data.loc[:"2025-04-01"]
    test_data = full_data.loc["2025-04-02":]

    # Tahmin verilerini hazırla
    pred_dates = test_data.index[:len(results[ticker]["Tahmin"])]
    pred_values = results[ticker]["Tahmin"]

    # Grafik oluştur
    plt.figure(figsize=(14, 8))
    gs = gridspec.GridSpec(2, 1, height_ratios=[3, 1])

    # Ana grafik
    ax0 = plt.subplot(gs[0])
    ax0.plot(train_data.index, train_data['Close'], label='Eğitim Verisi', color='blue')
    ax0.plot(test_data.index, test_data['Close'], label='Test Verisi (Gerçek)', color='green')
    ax0.plot(pred_dates, pred_values, label='Tahminler', color='red', linestyle='--', marker='o')
    ax0.set_title(f'{ticker} - Eğitim, Test ve Tahmin Grafiği')
    ax0.set_ylabel('Fiyat')
    ax0.legend()
    ax0.grid(True)

    # Hata grafiği
    ax1 = plt.subplot(gs[1])
    # errors'ı düzleştir (flatten) ve pred values ile aynı boyuta getir.
    errors = 100 * (test_data['Close'].values[:len(pred_values)] - pred_values) /
    test_data['Close'].values[:len(pred_values)]
    errors = errors.flatten()[:len(pred_values)] # pred_values ile aynı boyuta getir

    ax1.bar(pred_dates, errors, color='orange', alpha=0.7)
    ax1.axhline(0, color='black', linewidth=0.5)
    ax1.set_title("Tahmin Hataları (%)")
    ax1.set_ylabel('Hata %')
    ax1.grid(True)

plt.tight_layout()
plt.xticks(rotation=45)

```

```

plt.show()

# İstatistikleri yazdır
print(f"\n{ticker} İstatistikleri:")
print(f"- Eğitim Verisi Boyutu: {len(train_data)} gün")
print(f"- Test Verisi Boyutu: {len(test_data)} gün")
print(f"- Tahmin Doğruluğu: %{results[ticker]['Tahmin Doğruluğu (%)]:.2f}")
print(f"- Overfitting Durumu: {results[ticker]['Overfitting']}")
print("="*60)

plt.figure(figsize=(14,8))
accuracy_values = [results[t]['Tahmin Doğruluğu (%)'] for t in results]
plt.barh(list(results.keys()), accuracy_values, color=[ 'green' if x > 90 else 'orange' for x in
accuracy_values])
plt.axvline(90, color='red', linestyle='--')
plt.title('Hisse Bazında Tahmin Doğrulukları (%)')
plt.xlabel('Doğruluk Oranı')
plt.ylabel('Hisse Senetleri')
plt.grid(axis='x')
plt.tight_layout()
plt.show()

```

Optimizasyon kodları

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize
import warnings
warnings.filterwarnings('ignore')

# 1. Tahmin Getirilerini Hesapla
def calculate_returns(results):
    returns = []
    for ticker in results:
        pred_prices = results[ticker]["Tahmin"]
        daily_returns = [0] + [(pred_prices[i] - pred_prices[i-1])/pred_prices[i-1] for i in range(1,
len(pred_prices))]
        returns.append(daily_returns)
    return np.array(returns)

# Getiri verilerini hesapla
returns = calculate_returns(results)
mean_returns = returns.mean(axis=1) * 252# Yıllık getiriler
cov_matrix = np.cov(returns) * 252# Yıllık kovaryans matrisi

# 2. Gelişmiş Tabu Arama Optimizasyonu
class AdvancedTabuSearch:

```

```

def __init__(self, returns, cov_matrix, max_weight=***, tabu_size=***, max_iter=***):
    self.returns = returns
    self.cov_matrix = cov_matrix
    self.max_weight = max_weight
    self.tabu_list = []
    self.tabu_size = tabu_size
    self.max_iter = max_iter
    self.n_assets = len(returns)

def generate_initial_solution(self):
    # Efficient frontier üzerinde rastgele bir çözüm oluştur
    target_return = np.random.uniform(np.min(self.returns), np.max(self.returns))
    weights = self.get_efficient_portfolio(target_return)
    return weights if weights is not None else np.ones(self.n_assets)/self.n_assets

def get_efficient_portfolio(self, target_return):
    bounds = [(0, self.max_weight) for _ in range(self.n_assets)]
    constraints = (
        {'type': 'eq', 'fun': lambda x: np.sum(x) - 1},
        {'type': 'eq', 'fun': lambda x: np.sum(x * self.returns) - target_return}
    )
    init_guess = np.ones(self.n_assets) / self.n_assets

    result = minimize(lambda w: np.sqrt(np.dot(w.T, np.dot(self.cov_matrix, w))),
        init_guess, bounds=bounds, constraints=constraints)

    return result.x if result.success else None

def get_neighbors(self, solution):
    neighbors = []
    for i in range(self.n_assets):
        if solution[i] > ***:
            for j in range(self.n_assets):
                if i != j and solution[j] < self.max_weight:
                    new_sol = solution.copy()
                    transfer = min(***, solution[i], self.max_weight - solution[j])
                    new_sol[i] -= transfer
                    new_sol[j] += transfer
                    # Yeni komşuyu efficient frontier üzerine taşı
                    port_return = np.sum(new_sol * self.returns)
                    efficient_sol = self.get_efficient_portfolio(port_return)
                    if efficient_sol is not None:
                        neighbors.append(efficient_sol)
    return neighbors

def objective_function(self, weights):
    port_return = np.sum(weights * self.returns)
    port_volatility = np.sqrt(np.dot(weights.T, np.dot(self.cov_matrix, weights)))

```

```

sharpe_ratio = port_return / port_volatility

# Sharpe oranı 2.0 ile 3.0 arasında olmalı
if sharpe_ratio < ***:
    penalty = (*** - sharpe_ratio) * 100
elif sharpe_ratio > ***:
    penalty = (sharpe_ratio - ***) * 100
else:
    penalty = 0

return -sharpe_ratio + penalty

def optimize(self):
    current_solution = self.generate_initial_solution()
    best_solution = current_solution.copy()
    best_value = self.objective_function(best_solution)

    for _ in range(self.max_iter):
        neighbors = self.get_neighbors(current_solution)

        # Komşular arasında en iyisini bul
        for neighbor in neighbors:
            if not any(np.allclose(neighbor, tabu) for tabu in self.tabu_list):
                current_value = self.objective_function(neighbor)
                if current_value < best_value:
                    best_value = current_value
                    best_solution = neighbor.copy()

        current_solution = best_solution.copy()
        self.tabu_list.append(current_solution)
        if len(self.tabu_list) > self.tabu_size:
            self.tabu_list.pop(0)

    return best_solution

# 3. Optimizasyonu Çalıştır
if len(results) > 0:
    print("\nPortföy optimizasyonu başlıyor...")

# Kovaryans matrisini düzeltme (pozitif tanımlı yapma)
cov_matrix = cov_matrix + np.eye(cov_matrix.shape[0]) * 1e-6

# Gelişmiş Tabu Search ile optimizasyon
tabu_search = AdvancedTabuSearch(mean_returns, cov_matrix, max_iter=***)
optimal_weights = tabu_search.optimize()

# Ağırlıkları normalize et ve maks %10 uygula
optimal_weights = optimal_weights / optimal_weights.sum()

```

```

optimal_weights = np.minimum(optimal_weights, 0.1)
optimal_weights = optimal_weights / optimal_weights.sum()

# 4. Performans Metriklerini Hesapla
def calculate_metrics(weights, returns, cov_matrix, risk_free_rate=0.0):
    port_return = np.sum(weights * returns)
    port_volatility = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))
    sharpe_ratio = (port_return - risk_free_rate) / port_volatility

    # Sortino oranı (sadece negatif getiriler)
    negative_returns = returns[returns < 0]
    downside_risk = np.std(negative_returns) if len(negative_returns) > 0 else 0
    sortino_ratio = (port_return - risk_free_rate) / downside_risk if downside_risk > 0 else 0

    # Treynor oranı (beta=1 varsayımıyla)
    beta = 1
    treynor_ratio = (port_return - risk_free_rate) / beta

    return {
        "Return": port_return,
        "Volatility": port_volatility,
        "Sharpe Ratio": sharpe_ratio,
        "Sortino Ratio": sortino_ratio,
        "Treynor Ratio": treynor_ratio
    }

portfolio_perf = calculate_metrics(optimal_weights, mean_returns, cov_matrix)

# 5. Sonuçları Görselleştir
plt.figure(figsize=(14, 6))
assets = list(results.keys())
plt.bar(assets, optimal_weights)
plt.xticks(rotation=90)
plt.title('Optimal Portföy Ağırlıkları (Maks %10)')
plt.ylabel('Ağırlık')
plt.axhline(y=0.1, color='r', linestyle='--', label='Maks Ağırlık Sınırı')
plt.legend()
plt.tight_layout()
plt.show()

# Efficient Frontier çizimi
def efficient_frontier(returns, cov_matrix, max_weight=0.1, points=100):
    n = len(returns)
    frontier_returns = np.linspace(min(returns), max(returns), points)
    frontier_volatility = []

    bounds = [(0, max_weight) for _ in range(n)]
    for target_return in frontier_returns:

```

```

constraints = (
    {'type': 'eq', 'fun': lambda x: np.sum(x) - 1},
    {'type': 'eq', 'fun': lambda x, target=target_return: np.sum(x * returns) - target}
)
init_guess = np.ones(n) / n

result = minimize(lambda w: np.sqrt(np.dot(w.T, np.dot(cov_matrix, w))),
    init_guess, bounds=bounds, constraints=constraints)

if result.success:
    frontier_volatility.append(result.fun)
else:
    frontier_volatility.append(np.nan)

return frontier_volatility, frontier_returns

vols, rets = efficient_frontier(mean_returns, cov_matrix, max_weight=0.1, points=100)

plt.figure(figsize=(10,6))
plt.plot(vols, rets, 'b--', label='Efficient Frontier')
plt.scatter(portfolio_perf['Volatility'], portfolio_perf['Return'], c='red', marker='*', s=200, label='Optimal Portföy')
plt.title("Efficient Frontier ve Optimal Portföy")
plt.xlabel("Volatilite (Risk)")
plt.ylabel("Beklenen Getiri")
plt.legend()
plt.grid(True)
plt.show()

# 6. Performans Sonuçlarını Yazdır
print("\n" + "="*50)
print("PORTFÖY PERFORMANS METRİKLERİ")
print("="*50)
print(f"Yıllık Beklenen Getiri: {portfolio_perf['Return']*100:.2f}%")
print(f"Yıllık Volatilite: {portfolio_perf['Volatility']*100:.2f}%")
print(f"Sharpe Oranı: {portfolio_perf['Sharpe Ratio']:.2f}")
print(f"Sortino Oranı: {portfolio_perf['Sortino Ratio']:.2f}")
print(f"Treynor Oranı: {portfolio_perf['Treynor Ratio']:.2f}")

# 7. Optimal Ağırlıkları Göster
print("\n" + "="*50)
print("OPTİMAL PORTFÖY DAĞILIMI")
print("="*50)
weighted_tickers = []
for ticker, weight in zip(results.keys(), optimal_weights):
    if weight > 0.001: # Sadece önemli ağırlıkları göster
        print(f"{ticker}: {weight*100:.2f}%")
    weighted_tickers.append(ticker)

```

```

# 8. Alternatif Optimizasyon (Gerekirse)
if portfolio_perf['Sharpe Ratio'] < 2.0 or portfolio_perf['Sharpe Ratio'] > 3.0:
    print("\nTabu Search istenen Sharpe aralığını sağlamadı, alternatif optimizasyon deniyor...")

def efficient_sharpe_optimization():
    # Efficient frontier üzerinde Sharpe oranı 2-3 arası en iyi portföyü bul
    target_returns = np.linspace(np.min(mean_returns), np.max(mean_returns), 100)
    best_sharpe = -np.inf
    best_weights = None

    for target in target_returns:
        weights = tabu_search.get_efficient_portfolio(target)
        if weights is not None:
            sharpe = calculate_metrics(weights, mean_returns, cov_matrix)['Sharpe Ratio']
            if 2.0 <= sharpe <= 3.0 and sharpe > best_sharpe:
                best_sharpe = sharpe
                best_weights = weights

    return best_weights, best_sharpe

alt_weights, alt_sharpe = efficient_sharpe_optimization()

if alt_weights is not None:
    optimal_weights = alt_weights
    optimal_weights = optimal_weights / optimal_weights.sum()
    optimal_weights = np.minimum(optimal_weights, 0.1)
    optimal_weights = optimal_weights / optimal_weights.sum()
    portfolio_perf = calculate_metrics(optimal_weights, mean_returns, cov_matrix)
    print(f"\nAlternatif optimizasyon başarılı. Yeni Sharpe Oranı: {portfolio_perf['Sharpe Ratio']:.2f}")
else:
    print("\nAlternatif optimizasyon istenen aralıkta çözüm bulamadı.")

```

```

class MaxReturnTabuSearch:
    def __init__(self, returns, cov_matrix, max_weight=***, tabu_size=***, max_iter=***):
        self.returns = returns
        self.cov_matrix = cov_matrix
        self.max_weight = max_weight
        self.tabu_list = []
        self.tabu_size = tabu_size
        self.max_iter = max_iter
        self.n_assets = len(returns)

    def generate_initial_solution(self):
        weights = np.random.random(self.n_assets)
        weights /= weights.sum()
        weights = np.minimum(weights, self.max_weight)
        weights /= weights.sum()
        return weights

```

```

def get_neighbors(self, solution):
    neighbors = []
    for i in range(self.n_assets):
        if solution[i] > ***:
            for j in range(self.n_assets):
                if i != j and solution[j] < self.max_weight:
                    new_sol = solution.copy()
                    transfer = min(***, solution[i], self.max_weight - solution[j])
                    new_sol[i] -= transfer
                    new_sol[j] += transfer
                    neighbors.append(new_sol)
    return neighbors

def objective_function(self, weights):
    port_return = np.sum(weights * self.returns)
    port_volatility = np.sqrt(np.dot(weights.T, np.dot(self.cov_matrix, weights)))
    sharpe_ratio = port_return / port_volatility

    # Sadece pozitif Sharpe oranını garanti et
    penalty = 0
    if sharpe_ratio <= 0:
        penalty = (1 - sharpe_ratio) * 1000# Büyük ceza

    return -port_return + penalty# Maksimum getiri için negatifini alıyoruz

def optimize(self):
    current_solution = self.generate_initial_solution()
    best_solution = current_solution.copy()
    best_value = self.objective_function(best_solution)

    for _ in range(self.max_iter):
        neighbors = self.get_neighbors(current_solution)

        for neighbor in neighbors:
            if not any(np.allclose(neighbor, tabu) for tabu in self.tabu_list):
                current_value = self.objective_function(neighbor)
                if current_value < best_value:
                    best_value = current_value
                    best_solution = neighbor.copy()

        current_solution = best_solution.copy()
        self.tabu_list.append(current_solution)
        if len(self.tabu_list) > self.tabu_size:
            self.tabu_list.pop(0)

    return best_solution

```

```

# Maksimum getiri portföyünü oluştur
max_return_ts = MaxReturnTabuSearch(mean_returns, cov_matrix, max_iter=***)
max_return_weights = max_return_ts.optimize()

# Ağırlıkları normalize et ve maks %10 uygula
max_return_weights = max_return_weights / max_return_weights.sum()
max_return_weights = np.minimum(max_return_weights, 0.1)
max_return_weights = max_return_weights / max_return_weights.sum()

# Performans metriklerini hesapla
max_return_perf = calculate_metrics(max_return_weights, mean_returns, cov_matrix)

# Sonuçları görselleştir
plt.figure(figsize=(14, 6))
plt.bar(results.keys(), max_return_weights)
plt.xticks(rotation=90)
plt.title('Maksimum Getiri Portföyü (Sharpe > 0)')
plt.ylabel('Ağırlık')
plt.axhline(y=0.1, color='r', linestyle='--', label='Maks Ağırlık Sınırı')
plt.legend()
plt.tight_layout()
plt.show()

# Performansı yazdır
print("\n" + "="*50)
print("MAKSİMUM GETİRİ PORTFÖYÜ PERFORMANSI")
print("="*50)
print(f"Yıllık Beklenen Getiri: {max_return_perf['Return']*100:.2f}%")
print(f"Yıllık Volatilité: {max_return_perf['Volatility']*100:.2f}%")
print(f"Sharpe Oranı: {max_return_perf['Sharpe Ratio']:.2f}")

```

```

class MinRiskTabuSearch:
def __init__(self, returns, cov_matrix, max_weight=***, tabu_size=***, max_iter=***):
self.returns = returns
self.cov_matrix = cov_matrix
self.max_weight = max_weight
self.tabu_list = []
self.tabu_size = tabu_size
self.max_iter = max_iter
self.n_assets = len(returns)

def generate_initial_solution(self):
weights = np.random.random(self.n_assets)
weights /= weights.sum()
weights = np.minimum(weights, self.max_weight)
weights /= weights.sum()
return weights

```

```

def get_neighbors(self, solution):
    neighbors = []
    for i in range(self.n_assets):
        if solution[i] > ***:
            for j in range(self.n_assets):
                if i != j and solution[j] < self.max_weight:
                    new_sol = solution.copy()
                    transfer = min(***, solution[i], self.max_weight - solution[j])
                    new_sol[i] -= transfer
                    new_sol[j] += transfer
                    neighbors.append(new_sol)
    return neighbors

def objective_function(self, weights):
    port_return = np.sum(weights * self.returns)
    port_volatility = np.sqrt(np.dot(weights.T, np.dot(self.cov_matrix, weights)))
    sharpe_ratio = port_return / port_volatility

    # Pozitif Sharpe oranını garanti et
    penalty = 0
    if sharpe_ratio <= 0:
        penalty = (1 - sharpe_ratio) * 1000

    return port_volatility + penalty # Minimum risk için direkt volatilitiyi minimize ediyoruz

def optimize(self):
    current_solution = self.generate_initial_solution()
    best_solution = current_solution.copy()
    best_value = self.objective_function(best_solution)

    for _ in range(self.max_iter):
        neighbors = self.get_neighbors(current_solution)

        for neighbor in neighbors:
            if not any(np.allclose(neighbor, tabu) for tabu in self.tabu_list):
                current_value = self.objective_function(neighbor)
                if current_value < best_value:
                    best_value = current_value
                    best_solution = neighbor.copy()

        current_solution = best_solution.copy()
        self.tabu_list.append(current_solution)
        if len(self.tabu_list) > self.tabu_size:
            self.tabu_list.pop(0)

    return best_solution

```

```

# Minimum risk portföyünü oluştur
min_risk_ts = MinRiskTabuSearch(mean_returns, cov_matrix, max_iter=***)
min_risk_weights = min_risk_ts.optimize()

# Ağırlıkları normalize et ve maks %10 uygula
min_risk_weights = min_risk_weights / min_risk_weights.sum()
min_risk_weights = np.minimum(min_risk_weights, 0.1)
min_risk_weights = min_risk_weights / min_risk_weights.sum()

# Performans metriklerini hesapla
min_risk_perf = calculate_metrics(min_risk_weights, mean_returns, cov_matrix)

# Sonuçları görselleştir
plt.figure(figsize=(14, 6))
plt.bar(results.keys(), min_risk_weights)
plt.xticks(rotation=90)
plt.title('Minimum Risk Portföyü (Sharpe > 0)')
plt.ylabel('Ağırlık')
plt.axhline(y=0.1, color='r', linestyle='--', label='Maks Ağırlık Sınırı')
plt.legend()
plt.tight_layout()
plt.show()

# Performansı yazdır
print("\n" + "="*50)
print("MİNİMUM RİSK PORTFÖYÜ PERFORMANSI")
print("="*50)
print(f"Yıllık Beklenen Getiri: {min_risk_perf['Return']*100:.2f}%")
print(f"Yıllık Volatilite: {min_risk_perf['Volatility']*100:.2f}%")
print(f"Sharpe Oranı: {min_risk_perf['Sharpe Ratio']:.2f}")

# İki portföyü karşılaştır
plt.figure(figsize=(10,6))
plt.plot(vols, rets, 'b--', label='Efficient Frontier')
plt.scatter(max_return_perf['Volatility'], max_return_perf['Return'], c='red', marker='*', s=200, label='Maks Getiri Portföyü')
plt.scatter(min_risk_perf['Volatility'], min_risk_perf['Return'], c='green', marker='o', s=200, label='Min Risk Portföyü')
plt.title("Portföy Karşılaştırması")
plt.xlabel("Volatilite (Risk)")
plt.ylabel("Beklenen Getiri")
plt.legend()
plt.grid(True)
plt.show()

# Efficient Frontier hesapla
vols, rets = efficient_frontier(mean_returns, cov_matrix, max_weight=0.1, points=100)

```

```

# 1. Optimal Sharpe Portföyü (daha önceki kodlardan)
optimal_sharpe_perf = calculate_metrics(optimal_weights, mean_returns, cov_matrix)

# 2. Maksimum Getiri Portföyü (yukarıdaki kodlardan)
max_return_perf = calculate_metrics(max_return_weights, mean_returns, cov_matrix)

# 3. Minimum Risk Portföyü (yukarıdaki kodlardan)
min_risk_perf = calculate_metrics(min_risk_weights, mean_returns, cov_matrix)

# Grafik oluştur
plt.figure(figsize=(12, 8))
plt.plot(vols, rets, 'b--', linewidth=2, label='Efficient Frontier')

# Portföyleri işaretle
plt.scatter(optimal_sharpe_perf['Volatility'], optimal_sharpe_perf['Return'],
c='gold', s=300, marker='*', edgecolors='black',
label=f'Optimal Sharpe (SR={optimal_sharpe_perf["Sharpe Ratio"]:.2f})')

plt.scatter(max_return_perf['Volatility'], max_return_perf['Return'],
c='red', s=200, marker='^', edgecolors='black',
label=f'Maks Getiri (R={max_return_perf["Return"]*100:.1f}%)')

plt.scatter(min_risk_perf['Volatility'], min_risk_perf['Return'],
c='green', s=200, marker='o', edgecolors='black',
label=f'Min Risk ( $\sigma$ ={min_risk_perf["Volatility"]*100:.1f}%)')

# Grafik düzenlemeleri
plt.title('Üç Portföyün Efficient Frontier Üzerinde Karşılaştırması', fontsize=14)
plt.xlabel('Yıllık Volatilite (Risk)', fontsize=12)
plt.ylabel('Yıllık Beklenen Getiri', fontsize=12)
plt.legend(loc='upper left', fontsize=10)
plt.grid(True, alpha=0.3)

# Portföy detaylarını ekstra kutucukta göster
textstr = '\n'.join((
f'Optimal Sharpe: {optimal_sharpe_perf["Sharpe Ratio"]:.2f}',
f'Maks Getiri: {max_return_perf["Return"]*100:.1f}%',
f'Min Risk: {min_risk_perf["Volatility"]*100:.1f}%'
))

props = dict(boxstyle='round', facecolor='white', alpha=0.5)
plt.text(0.05, 0.95, textstr, transform=plt.gca().transAxes,
verticalalignment='top', bbox=props, fontsize=10)

plt.tight_layout()
plt.show()

# Portföy Dağılımlarını Karşılaştırmalı Göster
fig, axes = plt.subplots(1, 3, figsize=(18, 6))

```

```
# Optimal Sharpe Dağılımı
axes[0].bar(results.keys(), optimal_weights)
axes[0].set_title(f'Optimal Sharpe Portföyü\n(Sharpe={optimal_sharpe_perf["Sharpe Ratio"]:.2f})')
axes[0].tick_params(axis='x', rotation=90)

# Maks Getiri Dağılımı
axes[1].bar(results.keys(), max_return_weights, color='red')
axes[1].set_title(f'Maks Getiri Portföyü\n(Getiri={max_return_perf["Return"]*100:.1f}%)')
axes[1].tick_params(axis='x', rotation=90)

# Min Risk Dağılımı
axes[2].bar(results.keys(), min_risk_weights, color='green')
axes[2].set_title(f'Minimum Risk Portföyü\n(Risk={min_risk_perf["Volatility"]*100:.1f}%)')
axes[2].tick_params(axis='x', rotation=90)

plt.tight_layout()
plt.show()
```