REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

FORECASTING BIST RENEWABLE ENERGY STOCKS WITH DEEP LEARNING MODELS AND PORTFOLIO OPTIMIZATION

Ozan ALP

MASTER OF SCIENCE THESIS
Department of Statistics
Program of Statistics

Supervisor
Assoc. Prof. Dr. Gülder KEMALBAY

August, 2025

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

FORECASTING BIST RENEWABLE ENERGY STOCKS WITH DEEP LEARNING MODELS AND PORTFOLIO OPTIMIZATION

A thesis submitted by Ozan ALP in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 04.08.2025 in Department of Statistics, Program of Statistics.

Assoc. Prof. Dr. Gülder KEMALBAY
Yildiz Technical University
Supervisor

Approved By the Examining Committee

Assoc. Prof. Dr. Gülder KEMALBAY, Supervisor
Yildiz Technical University _____

Assoc. Prof. Dr. Atıf Ahmet EVREN, Member
Yildiz Technical University _____

Prof. Dr. Ozan KOCADAĞLI, Member
Mimar Sinan Fine Arts University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Forecasting BIST Renewable Energy Stocks with Deep Learning Models and Portfolio Optimization supervised by my supervisor, Assoc. Prof. Dr. Gülder KEMALBAY. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Ozan ALP

Signature

Dedicated to my family and friends

ACKNOWLEDGEMENTS

First, I want to thank Assoc. Prof. Dr. Gülder KEMALBAY. Her guidance, patience, and support played a major role in this study. Her thoughtful feedback helped me shape this research. I also want to show my appreciation to Metin DÜZENLİ for his ongoing involvement and the many meaningful discussions he shared during different parts of this work. Lastly, I want to thank my family and friends who gave me moral support and helpful talks when I needed them.

Ozan ALP

TABLE OF CONTENTS

LIST OF SYMBOLS vii

LIST OF ABBREVIATIONS viii

LIST OF FIGURES ix

LIST OF TABLES x

ABSTRACT xi

ÖZET xiii

1 INTRODUCTION 1

1.1 Literature Review . . . 1
1.2 Objective of the Thesis . . . 4
1.3 Hypothesis . . . 5

2 Financial Market Fundamentals 6

2.1 Stock . . . 6
2.1.1 Definition and Nature of Stocks . . . 6
2.1.2 Factors Influencing Stock Prices . . . 7
2.2 Stock Analysis Techniques . . . 7
2.2.1 Fundamental Analysis . . . 8
2.2.2 Technical Analysis . . . 8
2.3 Integrating Analysis into Investment Strategy . . . 11
2.4 Stock Exchange . . . 12
2.5 Stock Index . . . 12
2.6 Portfolio . . . 13
2.6.1 Importance of Portfolio . . . 13
2.7 Risk . . . 14
2.8 Return/Profit . . . 15
2.9 Performance Evaluation . . . 15
2.10 Portfolio Risk–Return Framework . . . 16
2.10.1 Portfolio Return . . . 16

2.10.2 Portfolio Risk (Variance and Std-Dev) . . . 16
2.10.3 Risk-Adjusted Performance Ratios . . . 16
2.10.4 Optimization Models . . . 19
2.10.5 Mean-Variance + Sharpe + AB (MV-SR-AB) . . . 20
2.11 Diversification . . . 21

3 Methodology **22**
3.1 Deep Learning . . . 22
3.2 Artificial Neural Networks (ANN) . . . 22
3.2.1 Activation Functions . . . 23
3.2.2 Loss Function . . . 24
3.2.3 Gradient Descent Optimization . . . 24
3.2.4 Backpropagation Algorithm . . . 25
3.2.5 Recurrent Neural Networks (RNN) . . . 25
3.3 Vanilla Recurrent Neural Network . . . 26
3.3.1 Vanishing and Exploding Gradient . . . 27
3.4 Long Short-Term Memory (LSTM) . . . 28
3.5 Gated Recurrent Unit (GRU) . . . 29
3.6 Amazon Chronos . . . 29
3.7 Hyperparameters . . . 30
3.7.1 Hyperparameter Tuning . . . 31
3.8 Data Preprocessing . . . 32
3.9 Evolutionary Portfolio Optimization Procedure . . . 36
3.9.1 Evolutionary Algorithm . . . 36

4 Data Preparation, Model Training, and Portfolio Implementation **39**
4.1 Environment . . . 39
4.2 Dataset Description . . . 40
4.2.1 Data Cleaning and Interpolation of Stock Price Series . . . 41
4.2.2 Time Series Preparation and Data Loader Construction . . . 41
4.3 Hyperparameter Optimization . . . 42
4.4 Model Selection and Forecast Evaluation . . . 44
4.4.1 Forecast Visualization of Best Performing Models . . . 46
4.5 Portfolio Construction Using Risk-Adjusted Optimization Strategies 50

5 CONCLUSION **55**

REFERENCES **56**

PUBLICATIONS FROM THE THESIS **60**

LIST OF SYMBOLS

AB	Asymmmetric Beta
ANN	Artificial Neural Network
BIST	Borsa İstanbul
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MDA	Mean Directional Accuracy
MSE	Mean Squared Error
MV	Mean-Variance model
MV-SR	Mean-Variance-Sharpe model
MV-AB	Mean-Variance-Asymmmetric Beta model
MV-AB-SR	Mean-Variance-Asymmmetric Beta-Sharpe model
MVO	Mean-Variance Optimization
NSGA-II	Non-dominated Sorting Genetic Algorithm II
SR	Sharpe Ratio model
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RSI	Relative Strength Index
XELKT	BIST Electricity Index

LIST OF ABBREVIATIONS

β	Beta ratio (upside beta / downside beta)
β_i^+	Upside beta of asset i
β_i^-	Downside beta of asset i
h	Hidden state in recurrent neural networks
X	Input data or input sequence
L	Loss function
μ	Mean
R_p	Portfolio return
$\hat{y}_i$	Predicted value of the ith observation
S	Sharpe Ratio
σ	Standard deviation (volatility)
y_i	True value of the ith observation
W	Weight matrix in neural network models
w	Weight of an asset in the portfolio or model

LIST OF FIGURES

Figure 3.1 Activation function graphs for Sigmoid, Tanh, ReLU, and LReLU 23
Figure 3.2 Recurrent neural network loop [25] 26
Figure 3.3 Vanilla RNN cell [27] 26
Figure 3.4 Gradient formulas. Left: RNN, Right: LSTM [31] 27
Figure 3.5 LSTM cell structure [25] 28
Figure 3.6 GRU cell structure [34] 29
Figure 3.7 High-level depiction of Chronos. 30
Figure 3.8 Left:Grid search. Right: Random search [38] 32
Figure 3.9 Algorithm flowcharts of NSGA-II. 37
Figure 4.1 Best parameters for each model 43
Figure 4.2 Best forecasting models per stock based on MSE and RMSE . . 44
Figure 4.3 Best forecasting models per stock based on MAE 45
Figure 4.4 Best forecasting models per stock based on MAPE 45
Figure 4.5 Best forecasting models per stock based on MDA 45
Figure 4.6 Forecast & ground truth for ALFAS.IS and SMRTG.IS 46
Figure 4.7 Forecast & ground truth for HUNER.IS and MAGEN.IS 47
Figure 4.8 Forecast & ground truth for KARYE.IS and BIOEN.IS 47
Figure 4.9 Forecast & ground truth for AYDEM.IS and GWIND.IS 48
Figure 4.10 Forecast & ground truth for NATEN.IS and PAMEL.IS 48
Figure 4.11 Forecast & ground truth for ZOREN.IS and ENJSA.IS 49
Figure 4.12 Forecast & ground truth for XELKT.IS 49
Figure 4.13 2D Pareto front visualization using return and variance objective values of MV and MV-SR model. 51
Figure 4.14 2D Pareto front visualization using return and variance objective values of MV-AB and MV-AB-SR model. 52
Figure 4.15 Cumulative returns of Mean-Variance included strategies. 54
Figure 4.16 Beta Ratios of each strategy portfolios 54

LIST OF TABLES

Table 4.1 Calculated portfolio weights using forecast values. 53
Table 4.2 Portfolio comparison of strategies, evaluated with ground truth values. 53

ABSTRACT

Forecasting BIST Renewable Energy Stocks with Deep Learning Models and Portfolio Optimization

Ozan ALP

Department of Statistics
Master of Science Thesis

Supervisor: Assoc. Prof. Dr. Gülder KEMALBAY

This thesis investigates portfolio optimization within the renewable energy sector of Borsa Istanbul (BIST) by integrating deep learning-based return forecasting with multi-objective portfolio models. Daily closing prices of twelve renewable energy companies listed in the BIST Electricity Index between November 16, 2022, and March 19, 2025, were used as the empirical foundation. For one-day-ahead stock return prediction, four time series forecasting models were implemented: Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Amazon's Chronos framework.

Forecasting performance was evaluated using five metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Directional Accuracy (MDA). For each stock, the best-performing model based on MAPE was selected and its return predictions were incorporated into portfolio construction.

Four distinct multi-objective portfolio optimization models were developed and solved using the Non-dominated Sorting Genetic Algorithm-II: (i) Mean-Variance (MV), (ii) Mean-Variance with Sharpe ratio maximization (MV-SR), (iii) Mean-Variance with Asymmetric Beta minimization (MV-AB), and (iv) Mean-Variance with both Sharpe and Asymmetric Beta objectives (MV-SR-AB). All portfolios were subject to full-investment and no-short-selling constraints.

The empirical findings demonstrate that combining deep learning–based forecasts with multi-objective portfolio optimization enhances risk-adjusted returns. In particular, the MV-SR-AB model delivered consistently superior performance in terms of Sharpe ratio, Sortino ratio, and long-term growth, illustrating the added value of incorporating downside-risk sensitivity alongside traditional mean-variance criteria.

Keywords: Deep learning, stock price forecasting, portfolio optimization, recurrent neural networks, genetic algorithm

ÖZET

BIST Yenilenebilir Enerji Hisselerinin Derin Öğrenme Modelleriyle Tahmini ve Portföy Optimizasyonu

Ozan ALP

İstatistik Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Gülder KEMALBAY

Bu tez, Borsa İstanbul (BIST) yenilenebilir enerji sektöründe portföy optimizasyonunu, derin öğrenme tabanlı getiri tahmini ile çok amaçlı portföy modellerini entegre ederek incelemektedir. Ampirik analiz, 16 Kasım 2022 ile 19 Mart 2025 tarihleri arasında BIST Elektrik Endeksi'nde işlem gören on iki yenilenebilir enerji şirketinin günlük kapanış fiyatlarına dayanmaktadır. Hisse senetlerinin bir gün sonraki getirilerini tahmin etmek için dört zaman serisi tahmin modeli kullanılmıştır: Tekrarlayan Sinir Ağı (RNN), Gated Recurrent Unit (GRU), Uzun Kısa Süreli Bellek (LSTM) ve Amazon'un Chronos çerçevesi.

Tahmin performansı, Beş farklı ölçüt kullanılarak değerlendirilmiştir: Ortalama Mutlak Hata (MAE), Ortalama Kare Hata (MSE), Kök Ortalama Kare Hata (RMSE), Ortalama Mutlak Yüzde Hata (MAPE) ve Ortalama Yön Doğruluğu (MDA). Her hisse için, MAPE değerine göre en iyi performansı gösteren model seçilmiş ve bu modelin getiri tahminleri portföy oluşturma sürecinde kullanılmıştır.

Dört farklı çok amaçlı portföy optimizasyon modeli geliştirilmiş ve Baskın Olmayan Sıralama Genetik Algoritma-II kullanılarak çözülmüştür: (i) Ortalama-Varyans (MV), (ii) Ortalama-Varyans ve Sharpe oranı maksimizasyonu (MV-SR), (iii) Ortalama-Varyans ve Asimetrik Beta minimizasyonu (MV-AB) ve (iv) Ortalama-Varyans ile birlikte hem Sharpe oranı hem Asimetrik Beta hedeflerini içeren model (MV-SR-AB). Tüm portföyler, tam yatırım ve kısa satış yasağı kısıtları altında değerlendirilmiştir.

Ampirik bulgular, derin öğrenme tabanlı getiri tahminlerinin çok amaçlı portföy optimizasyonu ile birleştirilmesinin, risk düzeltilmiş getirileri artırdığını göstermektedir. Özellikle MV-SR-AB modeli, Sharpe oranı, Sortino oranı ve uzun vadeli büyüme açısından tutarlı şekilde üstün performans sergilemiş; bu da geleneksel ortalama-varyans kriterlerine ek olarak aşağı yönlü risk hassasiyetinin dahil edilmesinin sağladığı katma değeri ortaya koymuştur.

Anahtar Kelimeler: Derin öğrenme, hisse senedi fiyat tahmini, portföy optimizasyonu, tekrarlayan sinir ağları, genetik algoritma

1
INTRODUCTION

1.1 Literature Review

Portfolio optimization stands as a key issue in finance seeking to distribute assets in a way that strikes a balance between risk and return for the best results. Over the years, the main methods for portfolio optimization have used mean-variance analysis, which Harry Markowitz brought to light in the 1950s. This approach assumes that investors want to get the most returns for a defined risk level or to lower risk as much as possible for a given return. The mean-variance optimization (MVO) model builds portfolios by looking at the expected returns and how assets move together. It aims to find the efficient frontier a set of portfolios yielding the highest expected return for a specific risk level or the lowest risk for a certain return level [1].

But old-school MVO and similar approaches run into a few problems. One big issue is how they are thrown off by the numbers you put in, like expected returns and how assets move together. These figures often come from past data, which can be wrong and might not show what's going to happen in the market. What's more, MVO thinks asset returns follow a normal pattern and that investors always play it safe. In real life, these ideas don't always hold up.

Over the last few years deep learning has shown up as a useful tool to optimize portfolios. Unlike old-school methods that depend on clear guesses about asset returns and risk deep learning models can spot tricky patterns and links straight from big financial datasets. Deep neural networks can handle many kinds of info such as market data, core business data, feelings from news and social media, and other unusual data sources. By picking up on twisty relationships and time-based trends in data deep learning models might be better at predicting asset prices and making portfolios work harder.

Researchers have tried many deep learning models to optimize portfolios. LSTM network, which is a recurrent neural network variant, is very effective in identifying trends within sequential data like daily stock prices or intraday trends. Convolutional neural networks are used to pick out key features from structured inputs such as company financial reports or major economic data. People also use reinforcement learning methods like deep Q-networks and policy gradients to figure out the best strategies for making decisions when outcomes are uncertain.

In his research on deep reinforcement learning, Le Trung Hieu explores examines how different Deep Reinforcement Learning (DRL) methods are applied to improving stock portfolios. The research stands out by moving beyond past studies, which focused more on the trend-heavy and volatile cryptocurrency market, and instead tackling the intricate and two-way nature of the stock market. It introduces practical considerations like transaction fees and risks into the DRL systems' state and reward setups. The study evaluates top continuous policy gradient methods such as Deep Deterministic Policy Gradient (DDPG). The study also applies Minimum Variance Portfolio Theory to pick stocks and uses Wavelet Transform to clean up price data noise, showcasing real-world progress in applying DRL to financial markets [2].

Zhang et al. [3] introduce a new deep learning-based method that tackles the flaws in old-school portfolio optimization techniques. These traditional methods follow two steps: first, they predict asset returns, and then they estimate the covariance matrix to calculate the best portfolio weights. But these steps often lead to errors due to simplified guesses like assuming returns follow a normal pattern. Zhang and others suggest an all-in-one deep learning model that figures out portfolio weights . This skips the need to predict returns and estimate risks.

Their framework is highly flexible, enabling optimization of different objectives such as the Sharpe ratio and mean-variance trade-off, and supports practical constraints like cardinality constraints (limiting the number of assets in the portfolio) and maximum position constraints (restricting the proportion allocated to single asset). The model leverages a deep neural network architecture to learn complicated and non-linear patterns within data, enabling more precise and reliable portfolio optimization [3].

Zhang et al.'s method was throughly evaluated using both simulated and real market datasets, demonstrating its effectiveness with scalability. The empirical results show that their deep learning-based approach consistently outperforms traditional portfolio optimization methods across various performance metrics. The study's

extensive testing period, spanning over 20 years, highlights the model's robustness and adaptability to different market conditions. This research indicates an important step forward in applying deep learning to financial modeling, offering a more holistic and effective solution to portfolio optimization challenges [3].

Ta et al. [4] proposed using a LSTM based model used to estimate future stock prices and subsequent portfolio optimization. Their approach integrates multiple techniques of portfolio optimization , including a model with equal weights, mean-variance optimization, and Monte Carlo simulation, to construct efficient portfolios. The study demonstrated that the LSTM model outperformed classic methods like linear regression and support vector machines, achieving higher accuracy in stock prediction and yielding superior portfolio performance. The optimized portfolios not only showed significant improvements in returns and Sharpe ratios but also surpassed the Standard & Poor's 500 index in active returns and risk management .

Research shows that deep learning methods hold potential when compared to traditional techniques. These models tend to deliver stronger results in terms of balancing risk and returns while staying stable during market swings. But there are hurdles like needing a lot of clean data, understanding how the models work, and dealing with their high computing demands.

To manage portfolios new methods keep pushing boundaries as statistical techniques, machine learning, and deep learning improve. Mean-variance optimization still plays a key role, but innovations like Bayesian statistics and deep learning bring fresh ideas to refine strategies. Researchers will aim to make deep learning models more reliable and scalable. Efforts to combine these models with older methods and solve practical issues in real-world finance will take center stage going forward.

1.2 Objective of the Thesis

This thesis aims to compare four multi-objective portfolio-optimization strategies for equities in the BIST renewable-energy universe. All strategies rely on one-day-ahead return forecasts generated by deep-learning models (LSTM, GRU, Transformer variants) and are subject to full-investment $(\sum w_i = 1)$ and no-short-selling $(w_i \geq 0)$ constraints.

Using the NSGA-II evolutionary algorithm to approximate the Pareto frontier, the study evaluates whether incorporating Sharpe-ratio maximisation and/or an Asymmetric-Beta penalty—alongside the classical return-variance objectives—yields portfolios with superior out-of-sample risk-adjusted performance relative to the benchmark mean-variance formulation.

1.3 Hypothesis

H_1: The MV–SR–AB model achieves at least a 10% improvement in Sharpe ratio compared to the benchmark MV model. It also delivers at least a 25% improvement in Sortino ratio and significantly higher long-term return growth, as measured by CAGR.

H_2: The combined use of Sharpe ratio maximisation and Asymmetric Beta minimisation in the multi-objective formulation contributes to more robust and risk-aware portfolio structures. This is reflected in improved performance across multiple evaluation metrics, including risk-adjusted return and drawdown control.

2
Financial Market Fundamentals

2.1 Stock

2.1.1 Definition and Nature of Stocks

Stocks, also referred to as shares or equities, are financial instruments that give investors a partial stake in a publicly traded company [5]. When individuals purchase a stock, they essentially acquire a right to a portion of the company's earnings and assets based on the quantity of shares owned. Companies issue stocks to raise capital, typically via an Initial Public Offering (IPO), and subsequently traded on stock exchanges like New York Stock Exchange (NYSE).

Stockholders can benefit in two main ways:

- **Capital Gains**: Profit realized when a stock is sold with bigger price than it is purchased.
- **Dividends**: Periodic payments distributed from the company's profits to shareholders.

There are generally two types of stock:

- **Common Stocks**: Provide variable dividends and voting rights.
- **Preferred Stocks**: Generally have no voting rights but offer fixed dividends and higher claim than common stock during asset liquidation.

Stocks play a central role in capital markets, enabling companies to fund operations and growth, while providing investors with investment opportunities.

2.1.2 Factors Influencing Stock Prices

Stock prices are influenced by numerous dynamic factors including:

- **Corporate Performance**: Includes financial results such as revenue, earnings, and profit margins. A company that consistently delivers strong financial performance is often rewarded by investors through rising share prices.
- **Macroeconomic Indicators**: These are broad economic statistics that reflect the general condition of an economy:
  - **Inflation**: The pace at which the overall cost of goods and services goes up [6]. High inflation erodes purchasing power and may impact corporate profitability.
  - **Interest Rates**: The cost of borrowing money. An increase in interest rates usually leads to a drop in stock prices as they increase financing costs and reduce consumer spending.
  - **Gross Domestic Product (GDP)**: A key gauge of national economic activity. An increase in GDP growth reflects a healthier and more resilient economy, which can boost investor confidence [7].
- **Political and Geopolitical Events**: Elections, wars, regulatory changes, and trade tensions can all impact investor confidence and market direction.
- **Industry Trends**: Trends and developments specific to certain industries, such as advancements in renewable energy or fluctuations in commodity prices, can significantly influence stock prices within that sector.
- **Market Sentiment**: Refers to the overall mood or outlook of investors toward a certain security or financial market, often shaped by news, media coverage or behavior of large institutional investors.

These variables make the stock market complex and often unpredictable, prompting investors to rely on structured analysis methods.

2.2 Stock Analysis Techniques

Stock Analysis is vital for making investment choices and typically involves two main methodologies: fundamental analysis and technical analysis.

2.2.1 Fundamental Analysis

It aims to determine a stock's intrinsic value based on financial and economic factors, primarily through evaluating financial reports, including the balance sheet and cash flow statement [8].

Key Metrics in Fundamental Analysis

- **Earnings Per Share**: Measures how much of a company's net income is attributed to each individual share.
- **Price-to-Earnings Ratio**: Indicates the price investors are ready to pay for each dollar of a company's earnings.
- **Return on Equity**: Reveals the ability of a company to turn shareholders' equity into earnings.
- **Dividend Yield**: Annual dividend shown as a proportion of the share price.
- **Debt-to-Equity Ratio**: Assesses financial leverage.
- **Revenue Growth Rate**: Shows how effectively a company can expand its revenues.
- **Operating Margin**: Percentage of revenue remaining after variable costs.

2.2.2 Technical Analysis

It uses past price and trading volume data to forecast future prices, based on the idea that all available information is already reflected in the current price and that price patterns often recur

Basic OHLC Structure: Technical analysis often relies on daily price summaries called Open, High, Low, and Close (OHLC). These values are typically visualized with candlestick or bar charts:

- **Open**: Price when trading started.
- **High**: The maximum price observed.
- **Low**: The minimum price observed.
- **Close**: The price at which the session concluded.

Trend Indicators

- **Simple Moving Average (SMA)**: Calculates the average closing price of a stock over a defined time period.

$$\text{SMA}t = \frac{1}{n} \sum i = 0^{n-1} P_{t-i} \tag{2.1}$$

P_t: Closing price at time t
n: Period count
SMA_t: SMA at time t

- **Exponential Moving Average (EMA)**: Works like the SMA, but emphasizes recent prices more, making it quicker to adjust to new information. It is helpful for spotting trend shifts earlier.

$$\text{EMA}_t = \alpha \cdot P_t + (1 - \alpha) \cdot \text{EMA}_{t-1} \tag{2.2}$$

EMA_t: EMA at time t
P_t: Closing price at time t
α: Smoothing constant
EMA_{t-1}: Previous EMA value
$\alpha = \frac{2}{n+1}$: Derived from the chosen period n

- **Golden Cross / Death Cross**: The Golden Cross takes place when a long-term moving average is surpassed by a short-term moving average, indicating a bullish trend. The Death Cross is the opposite and may indicate a bearish trend.

Momentum Indicators

Relative Strength Index (RSI): Assesses the rate and extent of price fluctuations to determine overbought or oversold situations.

$$\text{RSI} = 100 - \frac{100}{1 + RS} \tag{2.3}$$

$RS = \frac{\text{Average Gain}}{\text{Average Loss}}$
RSI: Relative Strength Index
RS: Relative Strength ratio based on average gains and losses

Stochastic Oscillator: Evaluates a stock's closing price in relation to its price range during a specified timeframe.

$$K = \frac{C - Ln}{Hn - Ln} \times 100 \tag{2.4}$$

K: Stochastic value (momentum indicator)
C: Latest closing price
Ln: Lowest price of n periods
Hn: Highest price of n periods

MACD (Moving Average Convergence Divergence): Uses two EMAs to detect momentum shifts.

$$\text{MACD} = \text{EMA}12 - \text{EMA}26, \quad \text{Signal Line} = \text{EMA}9(\text{MACD}) \tag{2.5}$$

EMA_{12}: EMA of 12 days
EMA_{26}: EMA of 26 days
$\text{EMA}_9(\text{MACD})$: 9-period EMA of the MACD line (signal line)

Volatility Indicators

- **Bollinger Bands**: Consisting of a central band (SMA) and two external bands at a predetermined standard deviation above and below. When prices touch the outer bands, it may suggest high volatility or potential price reversals.

$$\text{Upper Band} = \text{SMA}_n + k \cdot \sigma, \quad \text{Lower Band} = \text{SMA}_n - k \cdot \sigma \tag{2.6}$$

- **Average True Range (ATR)**: Calculates the average range between high and low prices over a specified duration, offering an understanding of market fluctuations.

$$\text{TR} = \max(H - L, |H - C_{\text{prev}}|, |L - C_{\text{prev}}|) \tag{2.7}$$

$$\text{ATR}t = \frac{1}{n} \sum i = 1^n TR_i \tag{2.8}$$

Volume Indicators

- **On-Balance Volume (OBV)**: Tracks cumulative volume based on whether prices close higher or lower. *Formula:*

$$\text{If } C_t > C_{t-1}, \quad OBV_t = OBV_{t-1} + V_t \tag{2.9}$$

$$\text{If } C_t < C_{t-1}, \quad OBV_t = OBV_{t-1} - V_t \tag{2.10}$$

$$\text{Otherwise}, \quad OBV_t = OBV_{t-1} \tag{2.11}$$

- **Volume Moving Average**: A moving average applied to volume data. It helps determine whether current trading activity is above or below average. Formula is similar to the SMA but applied to volume.

Support, Resistance, and Candlestick Patterns

- **Support and Resistance Levels**: Support refers to the price level at which demand typically inhibits further declines in price, whereas resistance denotes the level at which supply restricts any further increases in price.
- **Candlestick Patterns**: Visual formations in price charts that represent trader behavior. Common patterns include:
  - **Doji**: Indicates market indecision when opening and closing prices are almost identical.
  - **Hammer**: A bullish reversal pattern that shows buyers are pushing back after selling pressure.
  - **Engulfing**: Occurs when a larger candle fully contains the previous candle, signaling a strong reversal.

2.3 Integrating Analysis into Investment Strategy

Many investors blend multiple analysis strategies. In this thesis, technical indicators such as moving averages, RSI, and MACD may be integrated into deep learning models to enhance prediction accuracy. These predicted returns are used for portfolio optimization to maximize the gains.

2.4 Stock Exchange

A stock exchange is a type of specialized market for the sale and purchase of various financial securities such as stocks and bonds, as well as commodities and currency options [9]. A stock exchange also serves as a balance facilitator that maintains equilibrium where it ensures that there is sufficient trading activity and ease of trading within the market, while at the same time maintaining competition among buyers and sellers. The stock exchange also enables funds to be efficiently allocated in the economy through investments, which enables them to get the funds they need from the public through shares.

2.5 Stock Index

A stock index is basically a number that represents the overall value of a group of chosen stocks. It's used to track how a specific part of the market or sometimes the whole market is doing over time [10]. Indices are constructed using different factors like company size, industry sector, or trading volume. Following are the indexes in BIST:

- **BIST 10 (XU010)**: Comprises the top 10 most liquid stocks.
- **BIST 30 (XU030)**: Includes 30 large-cap and high-liquidity stocks.
- **BIST 50 (XU050)**: Covers the 50 most traded stocks.
- **BIST 100 (XU100)**: The main index representing the top 100 companies.
- **BIST Industrial (XUSIN)**: Tracks the performance of companies in the industrial sector.
- **BIST Services (XUHIZ)**: Comprises companies operating in the services sector.
- **BIST Financial (XUMAL)**: Includes banks, insurance, and financial leasing companies.
- **BIST Technology (XUTEK)**: Represents companies in the technology sector.
- **BIST Construction (XINSA)**: Covers companies engaged in construction activities.
- **BIST Transportation (XULAS)**: Tracks transportation and logistics firms.

- **BIST Tourism (XTRZM)**: Includes firms in the tourism industry.
- **BIST Food & Beverage (XGIDA)**: Comprises companies from the food and beverage sector.
- **BIST Chemicals, Petroleum, Plastic (XKMYA)**: Represents companies in the chemicals, petroleum, and plastic sectors.
- **BIST Electricity (XELKT)**: Tracks firms involved in electricity generation and distribution.
- **BIST Banking (XBANK)**: Includes only banking sector companies.

For example, the **BIST Electricity Index** (XELKT) consists of companies operating in the electrical energy and renewable power sectors listed on Borsa Istanbul. This index enables analysts and investors to track the collective performance of Turkey's electricity market, evaluate sector-specific trends, and compare individual stocks against a broader benchmark.

Indices are crucial tools for portfolio management, passive investing (e.g., ETFs), and market sentiment analysis.

2.6 Portfolio

The term "portfolio" in finance refers to a collection or group of financial assets such as stocks, bonds, and other investments of a person, institution, or entity [11]. Portfolios are strategically constructed to achieve specific financial objectives, such as maximizing returns, minimizing risks through diversification, and matching the investor's comfort with risk and how long they plan to keep their money invested. They serve as a structured approach to managing investments and are regularly monitored and tweaked depending on how the market's doing and what the investor is aiming to achieve.

2.6.1 Importance of Portfolio

Portfolios are essential tools in finance because they allow individuals and organizations to manage their investments effectively. By investing money into a mix of different kinds of investments, portfolios help decrease the risk of losing money if one investment performs poorly. At the same time, they aim to maximize returns by balancing the level of risk taken with the potential for profit. Portfolios are tailored to meet personal goals like saving for retirement or covering the cost

of school, and they can be adjusted over time to adapt to changing economic conditions and market trends. Evaluating a portfolio's performance using metrics like returns and risk measures such as the Sharpe ratio helps investors assess how well their investments are doing and make informed decisions about future investments. Overall, portfolios help bring order to financial planning by offering a structured approach to managing investments while aiming to achieve long-term financial objectives. Following are some of the key points of the portfolio:

2.7 Risk

Risk, in a financial context, refers to the uncertainty or potential for financial loss associated with an investment. It represents the variability of returns and the the likelihood that an investment's actual performance will not match its anticipated return [12]. Here are key aspects of risk:

- **Market Risk**: Refers to the potential for investment losses due to factors that impact the broader financial markets, such as economic downturns, political unrest, or fluctuations in interest rates.
- **Credit Risk**: Arises when a borrower fails to meet their debt obligations, thereby causing financial loss to the lender or investor.
- **Liquidity Risk**: The possibility that an asset cannot be bought or sold quickly enough without causing a significant change in its market price.
- **Operational Risk**: Involves the risk of loss resulting from inadequate or failed internal processes, systems, or from external events like fraud or natural disasters.
- **Interest Rate Risk**: The chance that changes in interest rates will adversely impact the value of an investment, particularly affecting fixed-income instruments such as bonds.
- **Currency Risk**: Pertains to the potential loss caused by shifts in exchange rates, especially for investments that are denominated in foreign currencies.
- **Inflation Risk**: The risk that the purchasing power of investment returns will be diminished due to rising inflation, thereby lowering the real value of those returns.
- **Specific Risk**: The risk tied to a particular company or industry, as opposed to market-wide risks.

Investors assess and manage these risks to achieve their desired balance between risk and return, often using diversification, hedging strategies, and other risk management techniques. These methods will be mentioned later in the paper.

2.8 Return/Profit

The term "Return" describes how much an investment has earned or lost over a certain period, shown as a percentage of what was originally paid for it. It includes income and capital gains (the growth in an investment's worth over time). "Profit" term refers to the absolute amount of money gained from an investment, computed as the selling price minus the initial purchase price (and possibly adjusted for any additional costs or income). While these two terms are related they are not exacly the same. Return or profit can be positive or negative. By strategically allocating assets within a portfolio, investors try to find a middle ground where the potential gains are worth the risks they're taking [13]. Different assets typically have varying levels of risk and potential returns, and portfolio management seeks to optimize this trade-off to maximize overall return potential.

- **Long-Term Financial Goals**: Portfolios are structured to align with investors' financial objectives, whether it's wealth accumulation for retirement, funding education, or achieving specific financial milestones. The composition of a portfolio reflects these goals and is periodically modified to reflect evolving personal circumstances and market status.
- **Flexibility and Adaptability**: A thoughtfully built portfolio gives investors the flexibility to respond to shifts in the market and changes in the economy. It provides the flexibility to rebalance assets, capitalize on emerging opportunities, and protect against market downturns.

2.9 Performance Evaluation

Portfolios are evaluated based on performance metrics such as returns, volatility, and risk-adjusted measures like the Sharpe ratio. These evaluations help investors see if their current approach is working and give them a clearer path for deciding where to move their allocations next. To compare portfolio optimization results, several metrics can be used, depending on the specific goals and risk tolerance.

2.10 Portfolio Risk–Return Framework

2.10.1 Portfolio Return

The expected portfolio return represents the weighted average of individual asset returns in the investment basket. It serves as the primary performance metric, quantifying the mean anticipated gain over a specified period. Investors use this expectation to compare potential investments, where higher expected returns typically correspond to riskier asset allocations. The calculation inherently assumes historical patterns or estimated probabilities will persist, though real-world outcomes invariably deviate due to market uncertainties.

2.10.2 Portfolio Risk (Variance and Std-Dev)

Portfolio risk is statistically captured through two interrelated measures: variance and standard deviation. Variance reflects the squared dispersion of returns around their mean, emphasizing extreme deviations, while standard deviation (its square root) provides an intuitive volatility scale in the original return units. Together, they quantify the uncertainty inherent in achieving the expected return, forming the basis for modern risk-return optimization frameworks. Lower values indicate more predictable performance, though excessively minimized risk may constrain return potential.

2.10.3 Risk-Adjusted Performance Ratios

2.10.3.1 Sharpe Ratio)

The Sharpe Ratio measures risk-adjusted returns by comparing excess portfolio return (over the risk-free rate) to its volatility (standard deviation).

$$S = \frac{R_p - R_f}{\sigma_p} \tag{2.12}$$

S: Sharpe Ratio
R_p: Portfolio return
R_f: Risk-free rate
σ_p: Portfolio standard deviation

2.10.3.2 Sortino Ratio

Measures gains relative to potential losses.

$$\text{Sortino} = \frac{R_p - R_f}{\sigma_d} \tag{2.13}$$

Sortino: Sortino Ratio
R_f: Risk-free rate
σ_d: Downside deviation (standard deviation of negative returns)

2.10.3.3 Beta (β)

The beta coefficient (β) measures an asset's **systematic risk** relative to the overall market. It quantifies the asset's sensitivity to market movements:

- $\beta = 1$: Asset moves with the market
- $\beta > 1$: More volatile than the market (aggressive)
- $\beta < 1$: Less volatile than the market (defensive)

$$\beta = \frac{\text{Cov}(R_i, R_m)}{\text{Var}(R_m)} \tag{2.14}$$

β: Systematic risk coefficient
R_i: Asset returns
R_m: Market returns
$\text{Cov}(R_i, R_m)$: Covariance between asset and market returns
$\text{Var}(R_m)$: Variance of market returns

2.10.3.4 Asymmetric Beta

Upside and downside beta are asymmetric measures that shows how much a stock reacts to changes in the overall market. Unlike traditional beta, which assumes symmetric reaction to market changes, these metrics separately quantify the stock's behavior during positive and negative market returns:

Upside Beta (β^+): Measures sensitivity when the market is going up.

Downside Beta (β^-): Measures sensitivity when the market is falling.

Upside Beta (β_i^+):

$$\beta_i^+ = \frac{\text{Cov}(r_{i,t}, r_{m,t} \mid r_{m,t} > \mu_m)}{\text{Var}(r_{m,t} \mid r_{m,t} > \mu_m)} \tag{2.15}$$

Downside Beta (β_i^-):

$$\beta_i^- = \frac{\text{Cov}(r_{i,t}, r_{m,t} \mid r_{m,t} < \mu_m)}{\text{Var}(r_{m,t} \mid r_{m,t} < \mu_m)} \tag{2.16}$$

$r_{i,t}$: return of stock i at time t
$r_{m,t}$: return of the market index at time t
μ_m: average market return

2.10.3.5 Additional Performance Metrics

Absolute Return: Measures the actual (realized) return over a period.

$$R_p = \sum_{i=1}^{n} w_i R_i \tag{2.17}$$

R_p: Absolute portfolio return
w_i: Proportion of the total portfolio invested in asset i
R_i: Absolute return of asset i

Maximum Drawdown: Represents the biggest recorded drop in value, measured from the highest point to the lowest during a specific period.

Employing a range of these indicators provides a well-rounded understanding of both the portfolio's performance and risk characteristics of different portfolio optimization strategies can be evaluated.

2.10.4 Optimization Models

2.10.4.1 Mean-Variance (MV)

Aims to balance return and risk by selecting asset weights that optimize both objectives.

The following formula is the bi-objective version, which produces the Pareto frontier of non-dominated portfolios:

$$\begin{aligned} &\text{maximize} \quad f_1(\mathbf{w}) = \mathbf{w}^\top \boldsymbol{\mu} \\ &\text{minimize} \quad f_2(\mathbf{w}) = \mathbf{w}^\top \boldsymbol{\Sigma} \mathbf{w} \\ &\text{subject to} \quad \sum_{i=1}^{n} w_i = 1, \quad w_i \geq 0 \quad \text{(no short selling)} \end{aligned} \tag{2.18}$$

$f_1(\mathbf{w})$: Expected portfolio return to be maximized
$f_2(\mathbf{w})$: Portfolio variance to be minimized
$\mathbf{w}$: Vector of portfolio weights
$\boldsymbol{\mu}$: Vector of expected returns for each asset
$\boldsymbol{\Sigma}$: Covariance matrix of asset returns
w_i: Weight of asset i in the portfolio

2.10.4.2 Mean-Variance + Sharpe (MV-SR)

The model synthesizes the bi-objective Mean-Variance (MV) and Sharpe Ratio (SR) approaches through a tri-criteria optimization:

$$\begin{aligned} &\max_{\mathbf{w}}; \mu_p = \mathbf{w}^T \mu && \text{(Maximize expected return)} \\ &\min_{\mathbf{w}}; \sigma_p^2 = \mathbf{w}^T \Sigma \mathbf{w} && \text{(Minimize variance)} \\ &\max_{\mathbf{w}}; S_p = \frac{\mathbf{w}^T \mu - r_f}{\sqrt{\mathbf{w}^T \Sigma \mathbf{w}}} && \text{(Maximize Sharpe ratio)} \end{aligned} \tag{2.19}$$

Constraints:

$$\begin{aligned} &\sum_{i=1}^{n} w_i = 1 && \text{(Full investment)} \\ &w_i \geq 0 \quad \forall i && \text{(No short selling)} \end{aligned} \tag{2.20}$$

$\mathbf{w}$: Vector of portfolio weights $(w_1, w_2, \ldots, w_n)$
μ: Vector of expected asset returns
Σ: Covariance matrix of asset returns
r_f: Risk-free rate

2.10.4.3 Mean-Variance + Asymmetric Beta (MV-AB)

Another model that synthesizes the Mean-Variance (MV) and Asymmetric Beta (AS) approaches through the following multi-objective formulation:

$$\begin{aligned}
&\max_{\mathbf{w}} \mu_p = \mathbf{w}^\top \boldsymbol{\mu} && \text{(Maximize expected return)} \\
&\min_{\mathbf{w}}; \sigma_p^2 = \mathbf{w}^T \Sigma \mathbf{w} && \text{(Minimize variance)} \\
&\max_{\mathbf{w}} \beta^+ p = \mathbf{w}^\top \boldsymbol{\beta}^+ && \text{(Maximize up-market sensitivity)} \\
&\min \mathbf{w} \beta_p^- = \mathbf{w}^\top \boldsymbol{\beta}^- && \text{(Minimize down-market sensitivity)}
\end{aligned} \tag{2.21}$$

Constraints:

$$\mathbf{w}^\top \mathbf{1} = 1, \quad \mathbf{w} \geq 0 \tag{2.22}$$

$\boldsymbol{\beta}^+$: Upside betas (Cov$(R_i, R_m | R_m > 0)$/Var$(R_m | R_m > 0)$
$\boldsymbol{\beta}^-$: Downside betas (Cov$(R_i, R_m | R_m < 0)$/Var$(R_m | R_m < 0)$
Other symbols as previously defined

2.10.5 Mean-Variance + Sharpe + AB (MV-SR-AB)

The model combines the Mean-Variance (MV), Sharpe Ratio (SR), and Asymmetric Beta (AB) models into a unified multi-objective optimization approach.

$$\begin{aligned}
&\max_{\mathbf{w}} \mu_p = \mathbf{w}^\top \boldsymbol{\mu} && \text{(Maximize expected return)} \\
&\min_{\mathbf{w}}; \sigma_p^2 = \mathbf{w}^T \Sigma \mathbf{w} && \text{(Minimize variance)} \\
&\max_{\mathbf{w}} \beta^+ p = \mathbf{w}^\top \boldsymbol{\beta}^+ && \text{(Maximize up-market sensitivity)} \\
&\min \mathbf{w} \beta_p^- = \mathbf{w}^\top \boldsymbol{\beta}^- && \text{(Minimize down-market sensitivity)} \\
&\max_{\mathbf{w}}; S_p = \frac{\mathbf{w}^T \mu - r_f}{\sqrt{\mathbf{w}^T \Sigma \mathbf{w}}} && \text{(Maximize Sharpe ratio)}
\end{aligned} \tag{2.23}$$

Constraints:

$$\mathbf{w}^\top \mathbf{1} = 1, \quad \mathbf{w} \geq 0 \tag{2.24}$$

2.11 Diversification

Diversification plays a key role in managing a portfolio, focusing on lowering risk by spreading investments across different assets and sectors [14]. The core idea behind diversification is that a thoughtfully diversified portfolio—composed of different asset types—typically provides stronger average returns and involves less risk than relying on any single investment alone. [15]. This happens because gains in certain investments can help balance out losses in others, reducing the overall impact of poor performance within the portfolio. In practice, diversification can be achieved by investing in stocks from different sectors, combining equities with bonds, including international assets, or mixing growth and value-oriented strategies [16]. In the context of this study, diversification is particularly critical in managing the inherent volatility found in sectors like renewable energy. By incorporating diverse stocks and optimizing their weights based on predicted returns, investors can construct portfolios that better withstand market fluctuations while aiming for consistent growth.

3
Methodology

3.1 Deep Learning

Deep learning is a subpart of machine learning and uses artificial neural networks made up of several layers to analyze detailed trends and relationships hidden within large sets of information. These models find and pull out features automatically by organizing layers , with each layer focusing on more detailed patterns. Unlike older machine learning models that depend on chosen features deep learning works with raw data [17]. It uncovers hidden structures and connections without assuming anything about how the data is spread out. Building deep learning models requires optimizing a loss function with methods like stochastic gradient descent or other related approaches [18]. The process updates model parameters through backpropagation, which helps the network reduce prediction mistakes and perform better across different tasks. Deep learning has delivered strong results in fields such as image analysis, voice recognition, understanding human language, and predicting patterns in time-based data [19]. Its strength lies in handling huge amounts of data combined with advanced computing power and accessible open-source tools. This combination has made deep learning a key method in artificial intelligence research.

3.2 Artificial Neural Networks (ANN)

ANNs take ideas from how real neurons in the brain work to create computational models [20]. These networks have layers made up of nodes called neurons. Every neuron receives some input, works through it, and then passes the outcome along to the following layer. The structure of the network starts with an input layer, followed by one or more middle layers—called hidden layers—and ends with an output layer that delivers the final result.Neurons connect through weights, and these weights decide how one neuron affects another. Training allows the network to fine-tune its weights so that its predictions come closer to the real outcomes, narrowing the gap between prediction and actual results.

3.2.1 Activation Functions

Activation functions add non-linearity to a neural network, which is what lets it understand more complicated patterns. Without them, no matter how many layers the network has, it would still just act like a basic linear model.In all layers of a neural network, layer is followed by an activation function such as Sigmoid, ReLU, or Tanh. For example, ReLU is known for its computational efficiency and plays a vital role in addressing the vanishing gradient issue, while Tanh scales inputs to $[-1, 1]$ and is useful when zero-centered outputs are desirable [21].

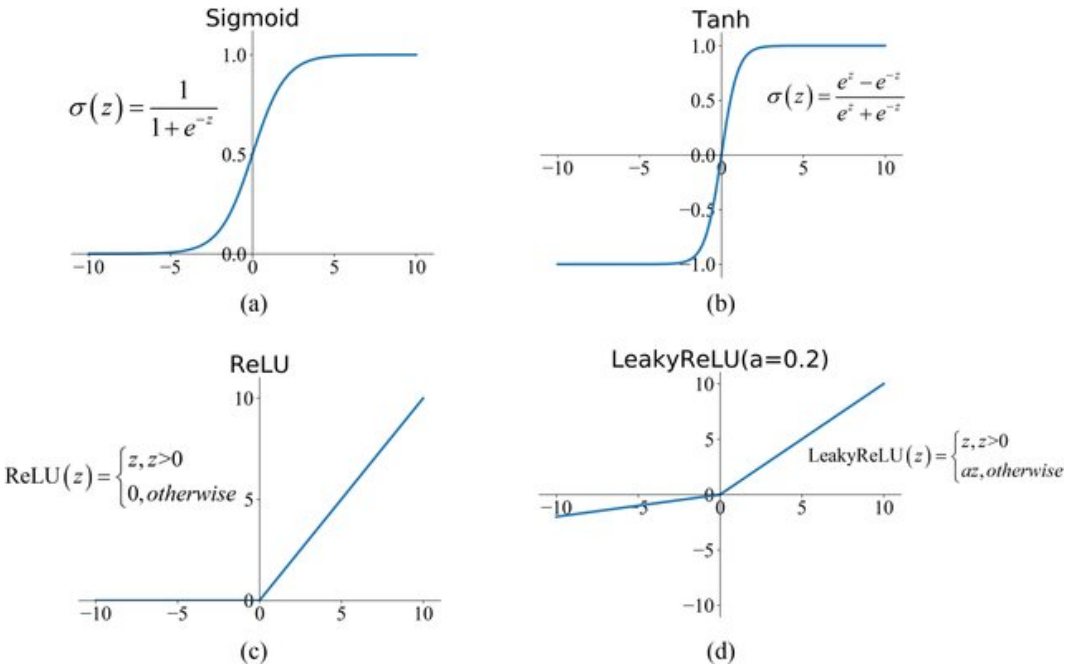

Figure 3.1 Activation function graphs for Sigmoid, Tanh, ReLU, and LReLU

- **Sigmoid**: Maps inputs to a value between 0 and 1.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

- **Hyperbolic Tangent (Tanh)**: Maps inputs to a value between -1 and 1.

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.2}$$

- **ReLU (Rectified Linear Unit)**: Gives an output of zero when the input is negative, and passes positive inputs through as they are.

$$\mathrm{ReLU}(x) = \max(0, x) \tag{3.3}$$

- **Leaky ReLU (LReLU)**: A variation of ReLU that lets a small, non-zero output pass through even when the input is negative (i.e., when $x < 0$). This helps to avoid the "dying ReLU" problem.

$$\text{LReLU}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha x, & \text{if } x < 0 \end{cases} \tag{3.4}$$

x: Input value of activation function
α: Small positive constant (commonly $\alpha = 0.01$)

3.2.2 Loss Function

It measures how far off the network's prediction is from the actual result. It acts as a guide to help improve the model during training by showing what needs to be adjusted. Common loss functions include L1 loss and L2 loss. Average of these functions are MAE and MSE respectively.

$$\mathcal{L}1 = \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{3.5}$$

y_i: Actual value of the i-th sample
$\hat{y}_i$: Prediction
n: Number of samples

$$\mathcal{L}2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3.6}$$

y_i: Actual value of the i-th sample
$\hat{y}_i$: Prediction
n: Number of samples

3.2.3 Gradient Descent Optimization

Gradient Descent is a step-by-step method for improving a model by lowering the loss. The general formula used for this update is:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta) \tag{3.7}$$

θ: Parameter (weight or bias)
η: Learning rate
$\nabla_\theta J(\theta)$: Gradient of the loss function with respect to parameter θ

At each step, it figures out how the loss shifts with changes in the model's parameters, then adjusts those parameters in the opposite direction of that shift, using the learning rate to decide how big each move should be. This adjustment keeps happening until it either settles at a solution or meets a set stopping rule. Training most artifical intelligence models relies on Gradient Descent. Variations like Stochastic Gradient Descent Momentum, and Adam were created to make the method faster and more reliable when handling huge datasets or tricky loss patterns [22]. Methods such as Momentum and ADAM introduce enhancements to improve convergence speed and stability.

3.2.4 Backpropagation Algorithm

Backpropagation allows the calculation of gradient of the loss function with respect to each weight in the network involves using the chain rule from calculus. This process allows the error to be propagated backward from the output layer to the input layer. This helps the network adjust its weights in an organized way. The process has two steps. First, in the forward pass, the network makes predictions. Second, in the backward pass, it calculates gradients and updates weights based on those calculations. Backpropagation plays a key role in training deep neural networks since it automates how gradients are computed, making the process much faster [23].

3.2.5 Recurrent Neural Networks (RNN)

One of the most widely used deep learning models for handling sequential data are RNNs, which are specifically designed to handle time-dependent inputs by keeping a hidden state that changes gradually with each time step. This makes RNNs a good fit for analyzing financial time series, where dependencies across past and future observations is key to spotting trends and dynamics [24].

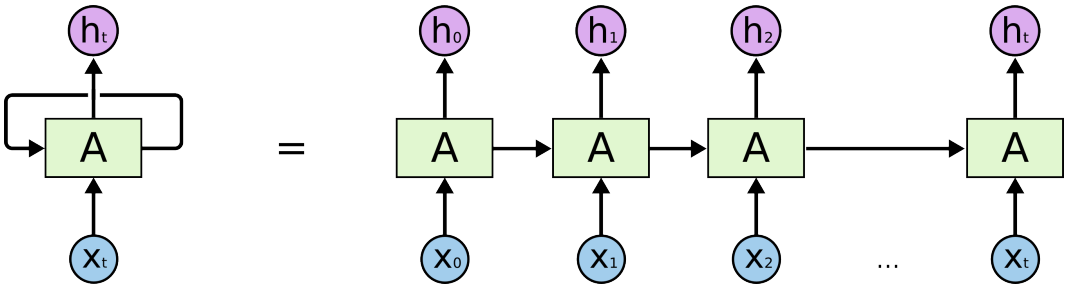

Figure 3.2 Recurrent neural network loop [25]

3.3 Vanilla Recurrent Neural Network

A Vanilla RNN or traditional RNN serves as one of the core structures used to model data that changes over time. It works from feedforward networks, which treat inputs as if they don't depend on each other. RNNs, on the other hand, use a repeating connection to let information carry over across different time steps. At each step, it processes an input vector and updates its hidden state by combining the current input with the hidden state from the previous step. This design helps the network keep a kind of memory, which makes it a good option to use with tasks like predicting sequences, modeling languages, or analyzing time series [26].

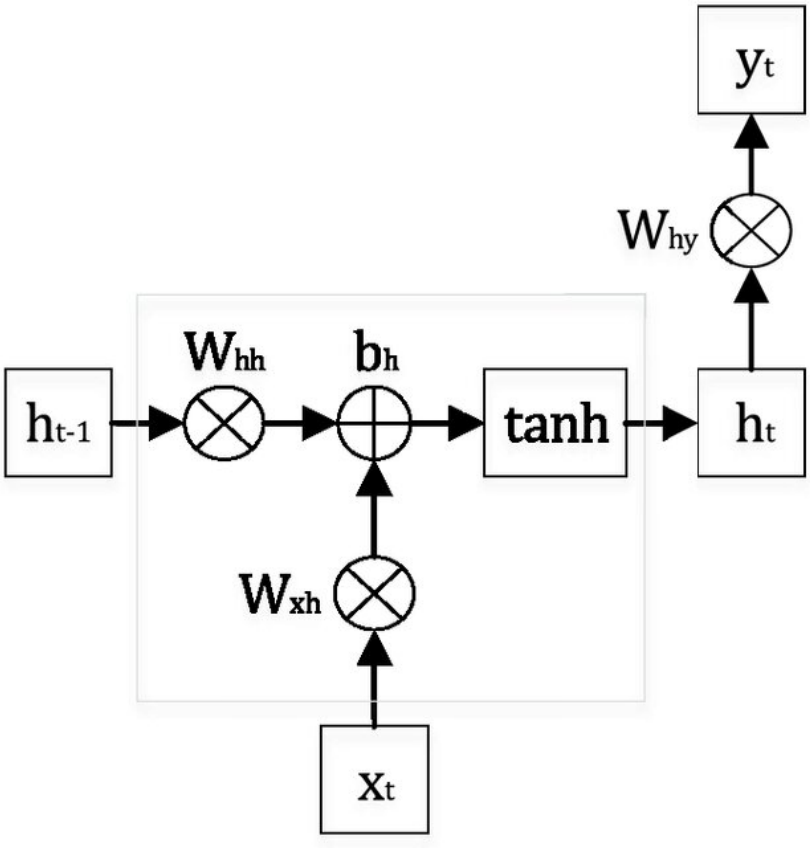

Figure 3.3 Vanilla RNN cell [27]

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{3.8}$$

$$y_t = W_{hy}h_t \tag{3.9}$$

h_t: Hidden state at time t
x_t: Input vector at time t
y_t: Output vector at time t
W_{xh}: Weights from input to hidden layer
W_{hh}: Weight for hidden-to-hidden recurrence
W_{hy}: Weight from hidden to output
b_h: Bias for hidden layer

3.3.1 Vanishing and Exploding Gradient

These problems, which are major challenges in training deep neural networks, was first identified by Hochreiter [28] and further analyzed by Bengio et al. [29]. Its theoretical underpinnings were later refined by Pascanu et al. [30].When an error signal propagates through multiple layers of non-linear transformations, its gradient tends to either grow uncontrollably or diminish rapidly. This leads to two major challenges in training: either the learning process becomes unstable due to excessively large updates, or it stagnates because the gradients become too small to drive meaningful parameter changes.

$$\frac{\partial h_{t'}}{\partial h_t} = \prod_{k=1}^{t'-t} w\sigma'(wh_{t'-k}) = \underbrace{w^{t'-t}}_{!!!} \prod_{k=1}^{t'-t} \sigma'(wh_{t'-k}) \qquad \frac{\partial s_{t'}}{\partial s_t} = \prod_{k=1}^{t'-t} \sigma(\nu_{t+k})$$

Figure 3.4 Gradient formulas. Left: RNN, Right: LSTM [31]

To solve this, researchers made better designs like LSTM and GRU networks. These improved models use gates to manage how data moves through the system. These gates let the network keep useful information for a long time and remove what it doesn't need. This ability helps the network handle patterns over tens or even hundreds of time steps. The derivative of the sigmoid function approaches 0 as its input values move towards either 0 or 1. If the weight isn't exactly 1, it will either shrink quickly toward zero as t'- t increases, or it'll grow rapidly over time. In LSTM, derivative of the Cell State does not include any components that leads to vanishing or exploding gradients. LSTMs include several parts like input, output, and forget gates as well as a memory cell. GRUs simplify things by combining the input and forget gates into a single update gate. Even though they differ in design, both models are much better at learning from time-based data than regular RNNs.

3.4 Long Short-Term Memory (LSTM)

LSTM is a type of neural network architecture specifically developed to maintain information across extended sequences and prevent gradient degradation during training. Its core component is a memory cell that preserves state over time, controlled by three multiplicative gates: the input gate, forget gate, and output gate [32]. These allow the model to decide which information should be written to the memory, retained over time, or exposed as output. By regulating gradient flow through these learned gates, LSTM prevents the vanishing gradient issue that usually hinders learning in deep or temporally extended models. The design enables effective training of models on tasks that rely on long-term temporal patterns, like time series forecasting, language modeling, and sequential classification.

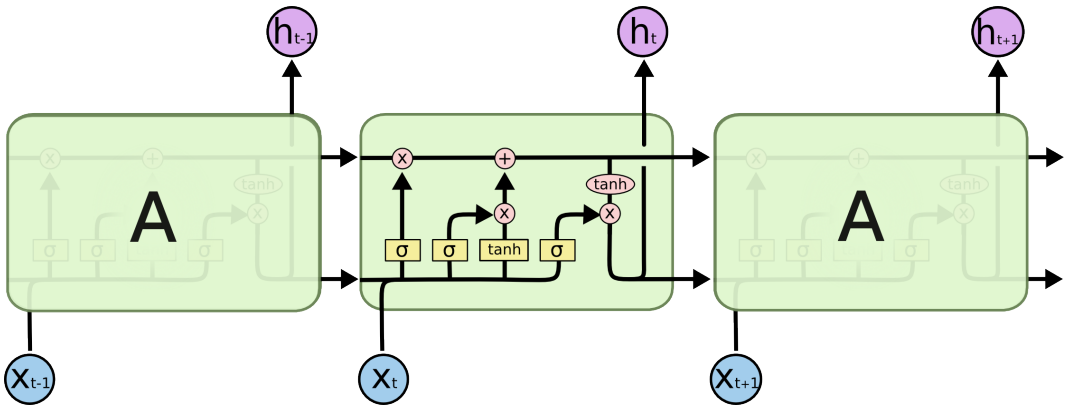

Figure 3.5 LSTM cell structure [25]

3.5 Gated Recurrent Unit (GRU)

GRU is a simplified yet powerful RNN architecture that addresses the limitations of traditional RNNs while using a more compact structure compared to LSTM. GRU merges the input and forget gates into a single update gate and adds a reset gate, which helps cut down on both the number of parameters and computational complexity [33]. The update gate determines the extent to which previous information should be carried forward, while the reset gate controls the extend of the past state to forget while calculating the new memory content. This setup enables GRUs to handle both short and long-term patterns effectively. Empirical studies have shown that GRUs deliver performance on par with LSTMs in many sequence modeling tasks while offering faster training and inference due to their streamlined architecture.

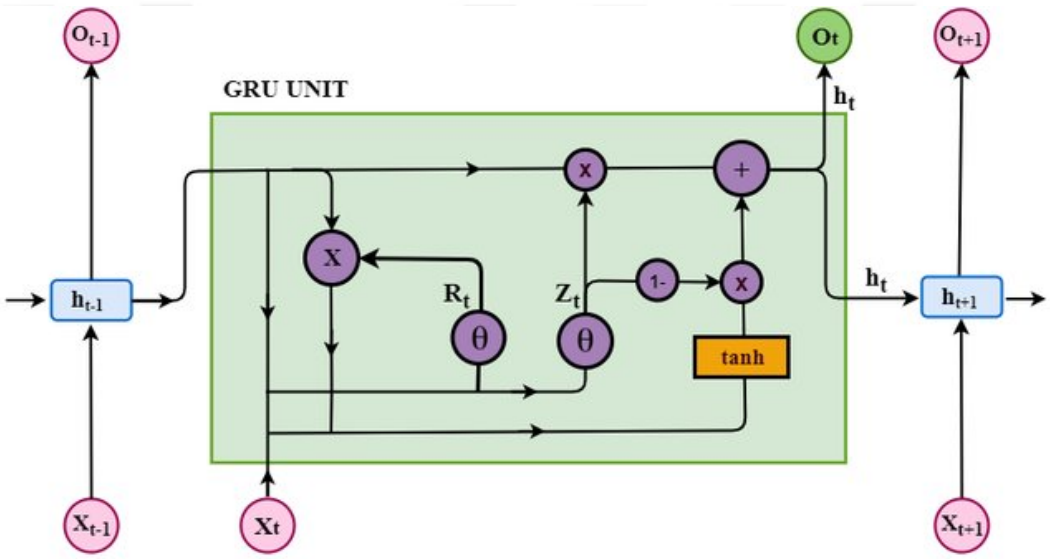

Figure 3.6 GRU cell structure [34]

3.6 Amazon Chronos

Chronos is a recently proposed framework for pretrained probabilistic time series forecasting that leverages transformer-based language models to learn from diverse temporal data. Instead of using raw time series values directly, Chronos applies scaling and quantization to discretize the data into a fixed vocabulary, enabling the use of standard transformer architectures trained via cross-entropy loss. The authors pretrain Chronos was trained using a large set of real-world public datasets, with extra support from added synthetic data generated through Gaussian processes to enhance generalization capabilities [35]. Chronos models are built on the T5 framework and span sizes from 20 million to 710 million parameters, are evaluated on 42 datasets in a benchmark that includes both classical statistical and modern

deep learning approaches. The findings show that Chronos regularly performs better than other models on in-domain datasets and demonstrates strong zero-shot forecasting performance on previously unseen datasets. This positions Chronos as a promising direction in time series forecasting, offering scalable and generalizable solutions through pretrained language models.

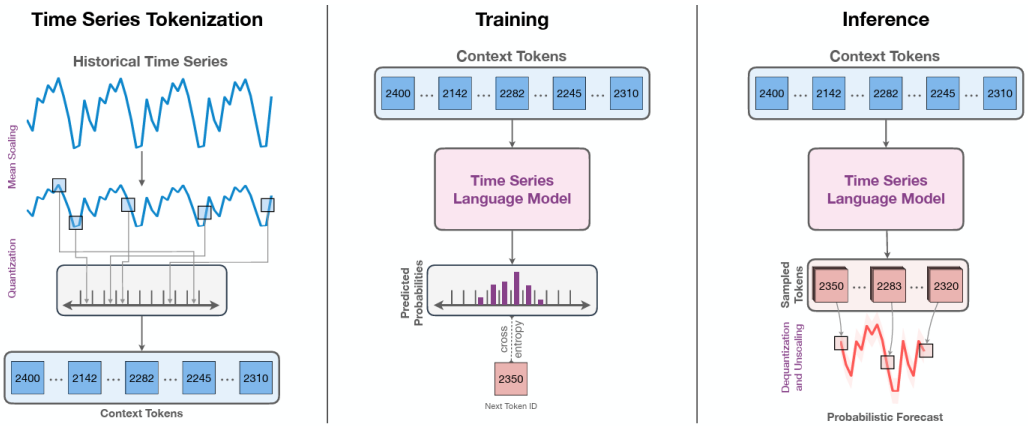

Figure 3.7 Chronos model depiction. [35]

3.7 Hyperparameters

The success of deep learning models relies on choosing hyperparameters with care. These settings have an impact on how the model learns how flexible it is, and its ability to handle unseen data effectively. This part explains the main hyperparameters used in training and the reasons behind picking them. Following are the common hyperparameters: **Learning Rate**: Sets how big a step the model

takes when updating its parameters each time. A small learning rate can result in slow convergence, conversely large one might cause the model to diverge. Proper tuning is essential for effective training [22].

Batch Size: This sets how many training samples are processed at once during a forward and backward pass. Using smaller batch sizes can help with regularization, while larger ones provide computational efficiency.

Number of Epochs: Refers to how many full cycles the model completes over the entire training dataset. Insufficient epochs may cause underfitting, where the model fails to capture key patterns. Conversely, excessive epochs can result in overfitting, where the model performs well on training data but poorly on new, unseen data.

Hidden Units: This refers to how many neurons are placed in the network's hidden layers. They determine the model's capacity to learn complex patterns. More hidden units may increase representational power, but also the risk of overfitting.

Optimizers: Optimizers modify the network's weights to reduce the loss function. Frequently selected options are Adam and RMSProp. Adam is particularly popular because of its adaptive learning rate and momentum properties [36].

3.7.1 Hyperparameter Tuning

When tuning hyperparameters, it requires selecting the most suitable combination of parameters that control how a model learns. Unlike weights and biases, which the model while in training, parameters like epoch or hidden units, and optimizer type are decided prior to the start of training. These decisions have a major impact on how well the model performs and how it trains. Tuning methods aim to balance underfitting and overfitting to enable the model to perform more reliably on data it hasn't seen before. Some common approaches include trying out values, using Grid Search, Random Search, or more advanced methods like Bayesian optimization [37]. This process becomes even more important in deep learning because the number of possible parameter combinations is huge and often hard to sort through.

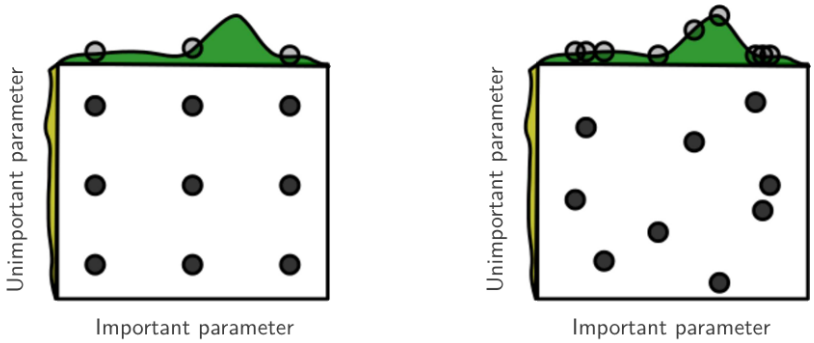

Figure 3.8 Left:Grid search. Right: Random search [38]

3.8 Data Preprocessing

Missing Data Handling Approaches

In time series modeling, missing data can negatively affect model performance and learning stability. Various imputation techniques are commonly applied to handle such gaps. Below are several widely used methods:

- **Forward Fill (Propagation)**: Fills missing values with the last known non-missing value.

$$x_t = x_{t-1} \tag{3.10}$$

  x_t: Imputed value at time t
  x_{t-1}: Last known value before t

- **Backward Fill**: Fills missing values using the next available value in the future.

$$x_t = x_{t+1} \tag{3.11}$$

  x_t: Imputed value at time t
  x_{t+1}: Next known value after t

- **Mean/Median Imputation**: Fills in missing values using either the column's average or its middle value.

$$x_t = \bar{x}, \quad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.12}$$

  x_t: Imputed value at time t
  $\bar{x}$: Mean of all known values
  n: Number of known data points
  x_i: Value of the ith observation

- **Linear Interpolation**: Estimates missing values by interpolating between known values on both sides.

$$y = y_1 + (x - x_1) \cdot \frac{y_2 - y_1}{x_2 - x_1} \tag{3.13}$$

x: Interpolation point
x_1, x_2: Neighboring known x-values
y_1, y_2: Corresponding known y-values
y: Interpolated value at x

Train-Test Split Approaches

In time series forecasting, the way a dataset is split into training and testing sets matters a lot when judging how well a model performs. In contrast to typical machine learning tasks that assume the data points are independent and follow the same distribution, time series data comes with built-in time-based patterns and dependencies that must be preserved. Therefore, specialized validation strategies are required. Below are several commonly used approaches:

- **Holdout Split**: The data is split into two parts — the earlier portion for train process, the later portion for testing. This simple method simulates real-world forecasting scenarios and is suitable for one-shot evaluations [39].
- **Rolling Forecast Origin (Walk-Forward Validation)**: The training set expands over time while the test set rolls forward. At each step, the model is retrained with the growing history and tested on the immediate next time point(s). This approach captures model adaptability [40].
- **Sliding Window Validation**: Both the training and testing windows have fixed sizes and move forward through time. It allows for multiple evaluations across the dataset while keeping the model memory limited to a recent window of observations [41].
- **K-Fold Cross-Validation**: Data is randomly split into k equal folds in this technique, which breaks time dependencies. As such, it is typically avoided in time series analysis due to risk of data leakage and unrealistic evaluation[42].
- **Time Series K-Fold Cross-Validation**: In this variation of K-Fold, data is split into k folds in a way that each fold respects the chronological order. Training always occurs on earlier data than the corresponding test fold, ensuring temporal integrity [42].

Every method comes with its own advantages and is better suited for specific situations depending on the data volume, stationarity, and the forecasting horizon. The splitting strategy adopted in this study is detailed in the implementation section.

Feature Scaling

It is an essential part of prepping data for time series forecasting and deep learning. It helps make sure that each input feature impacts the training process fairly. In financial time series—like prices and trading volumes—values can differ wildly in scale, which can hinder model convergence and performance if left unnormalized. Several common scaling techniques are used depending on the data distribution and robustness requirements.

Min-Max Scaling transforms each value to a fixed range, usually $[0, 1]$. It maintains the shape of the original distribution but is sensitive to outliers:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3.14}$$

x: Original value
$x_{\min}$: Minimum value
$x_{\max}$: Maximum value
x': Scaled value in $[0, 1]$

Standardization (Z-score Normalization)adjusts the data so it has a mean of zero and a standard deviation of one. It is commonly used when the model assumes normally distributed inputs:

$$z = \frac{x - \mu}{\sigma} \tag{3.15}$$

x: Original value
z: Standardized value

Robust Scaling relies on the median and interquartile range (IQR) rather than the mean and standard deviation, making it a better choice when the data has outliers:

$$x' = \frac{x - \text{median}(x)}{\text{IQR}(x)} \tag{3.16}$$

x: Original value
x': Robustly scaled value

Log Transformation is applied to reduce skewness in positively skewed data, often used for volume or price series. It compresses the range of high values:

$$x' = \log(x + 1) \tag{3.17}$$

x: Original value (must be non-negative)
x': Log-transformed value

The choice of method depends on how the data is distributed and what the model needs to perform well. Improper scaling can lead to slower convergence or biased learning, particularly in neural networks [43]. Proper feature scaling is essential for stable and efficient model training, particularly in deep learning, where unnormalized inputs can lead to gradient instability, poor convergence, and reduced model performance [44, 45].

3.9 Evolutionary Portfolio Optimization Procedure

3.9.1 Evolutionary Algorithm

Evolutionary algorithms are a group of optimization methods based on natural evolution ideas like selection, mutation survival of the fittest, and crossover. These methods work by simulating biological evolution to adjust candidate solutions until finding a good or ideal one. The process starts with a created population. It measures each individual's fitness with the objective function and then uses selection and variation operators to form new generations. As time passes and generations progress, the population moves closer to improved solutions. Evolutionary algorithms find many uses in solving tough optimization problems. They work well where gradient-based methods fail to perform in areas like non-convex challenges, problems without clear derivatives, or tasks with multiple goals. Their adaptability and strength make them a solid choice.

3.9.1.1 Non-dominated Sorting Genetic Algorithm II

The Non-dominated Sorting Genetic Algorithm II, or NSGA-II, widely used choice among researchers to solve multi-objective optimization problems. It aims to discover a range of diverse Pareto-optimal solutions. This version of the algorithm improves on the earlier one by using fast non-dominated sorting, an elitist selection process, and a method called the crowding distance operator to keep the solutions varied. To do this, it evolves a group of possible answers through operations like selection, crossover, and mutation. Individuals in the population get ranked based on Pareto dominance while ensuring variety in the objective space [46].

Pareto efficiency refers to a state in which no individual objective can be improved without worsening at least one other objective. These elements make NSGA-II useful to tackle difficult problems with conflicting goals in engineering, finance, and other areas like logistics.

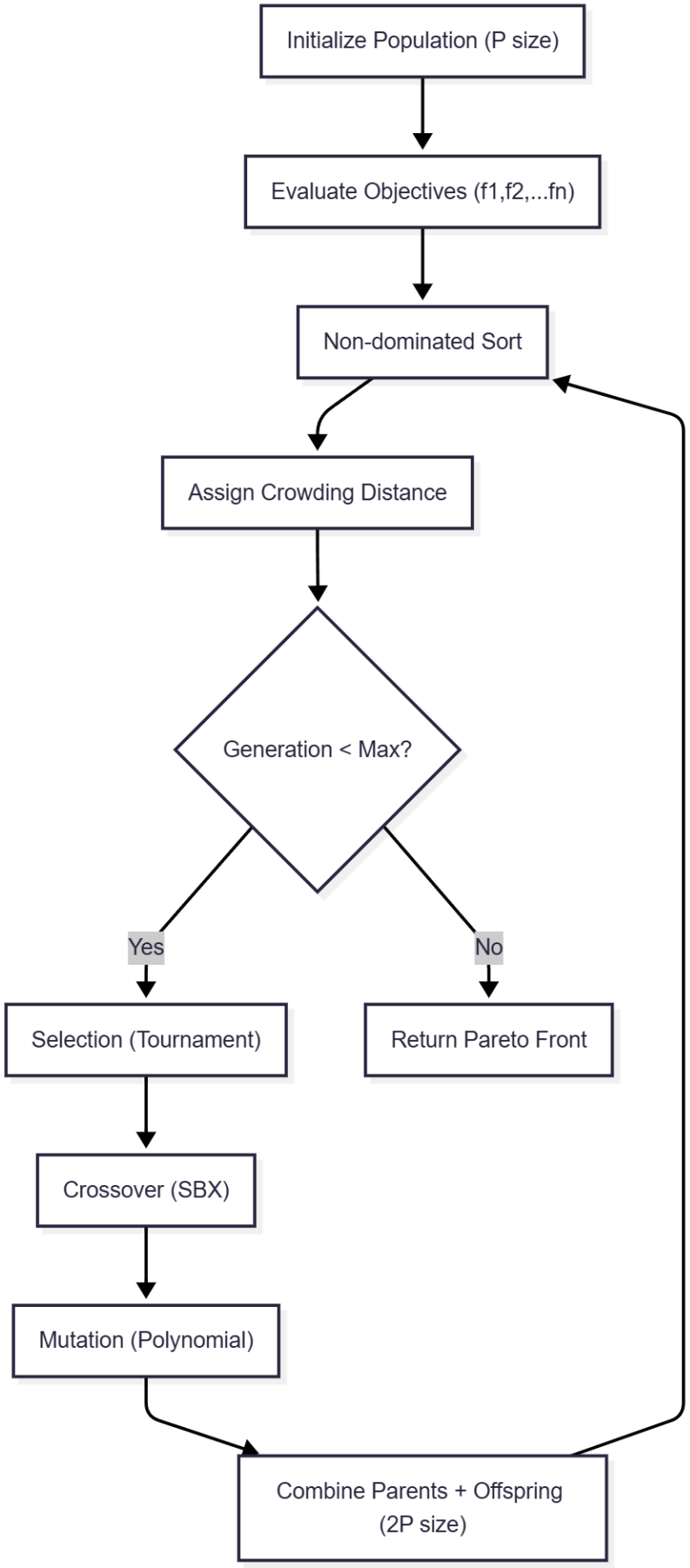

Figure 3.9 Algorithm flowcharts of NSGA-II.

3.9.1.2 Elbow/Knee Point Selection

Elbow/Knee Point Selection: In multi-objective portfolio optimization, the Pareto frontier often exhibits an "elbow" or "knee" point where marginal improvements in one objective (e.g., return) require disproportionate sacrifices in others (e.g., risk). This point represents an optimal trade-off balance, as moving beyond it yields diminishing returns. For a Mean-Variance-Sharpe frontier, the knee can be identified numerically by maximizing the curvature:

$$\kappa = \frac{|\mu_p'' \sigma_p' - \mu_p' \sigma_p''|}{(\mu_p'^2 + \sigma_p'^2)^{3/2}}, \tag{3.18}$$

or visually as the point of maximum inflection in 2D/3D Pareto plots. This method avoids arbitrary weight selection in scalarization approaches.

4
Data Preparation, Model Training, and Portfolio Implementation

This section details the full experimental pipeline—from cleaning the raw price series to deriving the final Pareto–efficient portfolios.

4.1 Environment

All study was conducted under the following technical specifications:

- **Hardware Configuration**:
  - Operating System: Windows 10 Pro
  - GPU: NVIDIA GeForce RTX 3080
  - CPU: [Intel i9-12900K]
  - RAM: [16GB DDR4]
- **Software Stack**:
  - Python 3.10.4
  - PyTorch 2.1.2
  - Additional Libraries: NumPy, pandas, scikit-learn, Matplotlib
  - Development Environment: [Jupyter Notebook, VSCode]

4.2 Dataset Description

Daily *Open*, *High*, *Low*, *Close*, and *Volume* observations for the twelve renewable-energy constituents of the BIST Electricity Index (`XELKT`) between November 16, 2022 and March 19, 2025 (590 trading days) were programmatically retrieved using the `yfinance` Python library. These twelve firms constitute the full free-float-adjusted investable universe of publicly listed Turkish companies whose core activity is generation, distribution, or EPC services in renewable power.

The companies analyzed are as follows:

- **ALFAS.IS (Alfa Solar Enerji)** – Provides solar panel manufacturing and project development services.
- **SMRTG.IS (Smart Güneş Teknolojileri)** – Produces photovoltaic solar panels and develops turnkey solar power plant projects.
- **HUNER.IS (Hun Yenilenebilir Enerji)** – Operates as a renewable energy producer through hydroelectric and wind power plants.
- **MAGEN.IS (Margün Enerji)** – A renewable energy company with a portfolio of wind and solar power plants.
- **KARYE.IS (Kare Enerji)** – A growing company focusing on investments in the solar energy sector.
- **BIOEN.IS (Biotrend Enerji)** – A pioneering firm producing renewable energy based on biomass and waste sources.
- **AYDEM.IS (Aydem Yenilenebilir Enerji)** – A major renewable energy producer operating hydro, wind, and solar power plants across Turkey.
- **GWIND.IS (Galata Wind Enerji)** – A sustainability-focused company producing energy solely from wind and solar sources.
- **NATEN.IS (Naturel Yenilenebilir Enerji)** – Provides EPC (engineering, procurement, construction) services and has its own generation investments.
- **PAMEL.IS (Pamukova Elektrik)** – A company primarily engaged in hydroelectric power production and expanding its renewable investments.
- **ZOREN.IS (Zorlu Enerji)** – One of Turkey's leading energy firms investing in solar, wind, and geothermal sources.
- **ENJSA.IS (Enerjisa Üretim)** – The generation subsidiary of Enerjisa, operating hydro, wind, solar, and natural gas power plants.

4.2.1 Data Cleaning and Interpolation of Stock Price Series

To keep the input data consistent and complete for time series modeling, daily stock prices from twelve companies listed on BIST were preprocessed using a structured cleaning routine. The raw CSV files were read using `pandas`, with special care taken to standardize column names and enforce a consistent order: Date, Open, High, Low, Close, Adj Close, and Volume.

Missing or zero-valued data points in numeric columns were treated as `NaN`, 3% of data was missing. Missing values were interpolated using linear interpolation. Linear interpolation respects the time-dependent structure of financial data by estimating missing values based on adjacent observations, unlike forward-fill which artificially flatlines values. To handle remaining gaps at the ends, forward-fill and backward-fill techniques were applied. These methods help maintain the temporal integrity of stock price movements while preserving underlying trends. Each cleaned dataset was then saved back to disk for use in modeling and training phases.

4.2.2 Time Series Preparation and Data Loader Construction

To prepare the selected BIST renewable energy stocks for deep learning models, all values were converted to float type to ensure compatibility with numerical processing.

To normalize input values across varying stock price ranges, a method that uses `MinMaxScaler` was created to transform all features into the range $[0, 1]$, which is critical for stable neural network training. The scaled data was then reshaped into sequences using a sliding window approach with given sequence length parameter. Rolling window stategy is used for train/test/validation split. For a given sequence length L, each training instance was formed by taking L consecutive observations as input, and the close value immediately following the sequence as the target. This transformation results in a 3D array suitable for Recurrent Neural Networks.

The features and labels are then turned into PyTorch tensors and divided into training and test sets. For smoother batch processing during training, `DataLoader` objects were constructed using `TensorDataset`, setting a batch size of 32 and preserving the temporal order by disabling shuffling.

4.3 Hyperparameter Optimization

To identify the most effective architecture and training configuration for the recurrent neural network, a grid search strategy was employed over a predefined hyperparameter space for all four models: RNN, LSTM,GRU and custom model MULTI which has stacked one lstm and one gru layers. The search space included combinations of hidden layer sizes, learning rates, number of recurrent layers, and training epochs. The model was trained with the given parameters for every combination and evaluated on the validation loss which is calculated by real close value and predicted close value of stocks for given windows. Chronos model is a large framework which is trained in industry level hardware so it is evaluated without gridsearch nor training.

- **Hidden Size**: Number of units in each RNN layer, tested with values {32, 64, 128}, affecting the model's learning skills of temporal dependencies.
- **Sequence Length**: The count of consecutive time steps used as input for forecasting, tested with values {30, 50}. This controls how far into the past the model looks when making a prediction. A properly tuned sequence length helps the network capture temporal dependencies without overwhelming it with noise or irrelevant history.
- **Number of Layers**: The depth of the RNN architecture, with configurations of {1, 2, 3} stacked layers. More depth can capture more complicated patterns but it can be vulnerable to vanishing gradients.
- **Learning Rate**: Tested with {0.01, 0.001, 0.0001}. Smaller values typically yield more stable convergence.
- **Epochs**: The number of full passes over the training data, with options {50, 200, 400}, allowing examination of underfitting and overfitting tendencies.

The process was executed using `sklearn.model_selection.ParameterGrid`, which systematically iterates through every possible parameter combination. A custom training function, `train_model_with_params`, was called for each configuration, and the resulting loss was tracked. The configuration yielding the lowest validation loss was selected as the optimal setting. This exhaustive search approach, while computationally expensive, ensures that the selected model structure is well-suited to the dynamics of the financial time series under consideration.

The grid search was applied separately to each stock in the dataset to capture stock-specific dynamics and model behavior. After training, the configuration that yielded the lowest validation loss for each stock-model pair was recorded. These best-performing hyperparameter sets were saved and later used for final evaluation and forecasting. This architecture-wise approach ensured a fair and tailored comparison between models, while also enabling analysis of performance consistency across different stocks.

Model	Hidden Size	Layers	Epoch	Learning rate	İnput Length
RNN	32	1	200	0.001	30
GRU	64	1	100	0.001	30
LSTM	32	1	200	0.001	30
Multi	64	-	200	0.001	50

Figure 4.1 Best parameters for each model

4.4 Model Selection and Forecast Evaluation

To evaluate and contrast how well different models predict, the actual stock closing prices were first merged with the corresponding forecasts produced by GRU, LSTM, RNN, MULTI and Amazon Chronos. This alignment was performed based on the common `Date` column to ensure temporal consistency across all predictions. The evaluation window was set between 11-11-2024 and 28-02-2025 to reflect the most recent and relevant forecast period.

For each of the twelve renewable energy stocks, five commonly used error metrics were calculated: MSE, RMSE, MAE, MAPE and MDA. MSE and RMSE capture the magnitude of errors, MAE evaluates average absolute deviation, MAPE assesses relative percentage errors, and MDA measures the directional consistency of predicted trends with actual movements.

After computing these metrics for each model-stock pair, the model with the lowest MAPE was selected as the best-performing predictor for that stock. These results were then compiled into a summary table, enabling both stock-wise and method-wise performance analysis.

Across the twelve stocks, performance varied across models: LSTM and GRU each won 3 stock series, while RNN, Multi-head, and Chronos won 2 series each.

STOCK	BEST MODEL	MSE	RMSE	MAE	MAPE	MDA
ALFAS.IS	GRU	4.75	2.18	1.44	2.11	53.49
SMRTG.IS	MULTI	1.06	1.03	0.64	1.54	55.81
HUNER.IS	RNN	0.01	0.1	0.08	2.13	51.16
MAGEN.IS	MULTI	0.34	0.58	0.38	1.81	37.21
KARYE.IS	AMAZON	0.27	0.52	0.36	1.27	39.53
BIOEN.IS	LSTM	0.08	0.28	0.23	1.39	48.84
AYDEM.IS	LSTM	0.14	0.38	0.3	1.24	65.12
GWIND.IS	LSTM	0.53	0.73	0.56	1.83	58.14
NATEN.IS	RNN3	6.75	2.6	2.09	3.1	58.14
PAMEL.IS	AMAZON	3.04	1.74	1.39	1.44	53.49
ZOREN.IS	RNN	0.01	0.07	0.06	1.36	53.49
ENJSA.IS	RNN3	1.28	1.13	0.85	1.4	55.81

Figure 4.2 Best forecasting models per stock based on MSE and RMSE

STOCK	BEST MODEL	MSE	RMSE	MAE	MAPE	MDA
ALFAS.IS	GRU	4.75	2.18	1.44	2.11	53.49
SMRTG.IS	MULTI	1.06	1.03	0.64	1.54	55.81
HUNER.IS	GRU	0.01	0.11	0.07	1.98	48.84
MAGEN.IS	MULTI	0.34	0.58	0.38	1.81	37.21
KARYE.IS	AMAZON	0.27	0.52	0.36	1.27	39.53
BIOEN.IS	RNN	0.09	0.3	0.23	1.38	53.49
AYDEM.IS	LSTM	0.14	0.38	0.3	1.24	65.12
GWIND.IS	LSTM	0.53	0.73	0.56	1.83	58.14
NATEN.IS	RNN3	6.75	2.6	2.09	3.1	58.14
PAMEL.IS	GRU	3.06	1.75	1.3	1.34	53.49
ZOREN.IS	RNN	0.01	0.07	0.06	1.36	53.49
ENJSA.IS	LSTM	1.3	1.14	0.84	1.37	53.49

Figure 4.3 Best forecasting models per stock based on MAE

STOCK	BEST MODEL	MSE	RMSE	MAE	MAPE	MDA
ALFAS.IS	GRU	4.75	2.18	1.44	2.11	53.49
SMRTG.IS	MULTI	1.06	1.03	0.64	1.54	55.81
HUNER.IS	GRU	0.01	0.11	0.07	1.98	48.84
MAGEN.IS	MULTI	0.34	0.58	0.38	1.81	37.21
KARYE.IS	AMAZON	0.27	0.52	0.36	1.27	39.53
BIOEN.IS	RNN	0.09	0.3	0.23	1.38	53.49
AYDEM.IS	LSTM	0.14	0.38	0.3	1.24	65.12
GWIND.IS	LSTM	0.53	0.73	0.56	1.83	58.14
NATEN.IS	AMAZON	9.31	3.05	2.1	3.01	53.49
PAMEL.IS	GRU	3.06	1.75	1.3	1.34	53.49
ZOREN.IS	RNN	0.01	0.07	0.06	1.36	53.49
ENJSA.IS	LSTM	1.3	1.14	0.84	1.37	53.49

Figure 4.4 Best forecasting models per stock based on MAPE

STOCK	BEST MODEL	MSE	RMSE	MAE	MAPE	MDA
ALFAS.IS	LSTM	4.97	2.23	1.46	2.13	51.16
SMRTG.IS	RNN	1.1	1.05	0.66	1.6	44.19
HUNER.IS	AMAZON	0.01	0.11	0.08	2.05	46.51
MAGEN.IS	RNN	0.4	0.63	0.42	1.99	37.21
KARYE.IS	MULTI	0.38	0.61	0.49	1.71	34.88
BIOEN.IS	MULTI	2142.75	46.29	45.26	278.48	41.86
AYDEM.IS	AMAZON	0.17	0.41	0.33	1.4	60.47
GWIND.IS	LSTM	0.53	0.73	0.56	1.83	58.14
NATEN.IS	AMAZON	9.31	3.05	2.1	3.01	53.49
PAMEL.IS	LSTM	3.05	1.75	1.39	1.44	48.84
ZOREN.IS	AMAZON	0.01	0.08	0.07	1.54	44.19
ENJSA.IS	AMAZON	1.67	1.29	0.97	1.59	41.86

Figure 4.5 Best forecasting models per stock based on MDA

4.4.1 Forecast Visualization of Best Performing Models

To get a visual sense of how well the top model performed for each stock, forecast plots were created showing 50 days of predicted closing prices alongside the real ones during the evaluation period. Each graph highlights the ground truth data in contrast with the model's predictions, showcasing how well the selected model captures the direction and magnitude of price changes. This visual validation supports the quantitative evaluation and helps reveal model behavior.

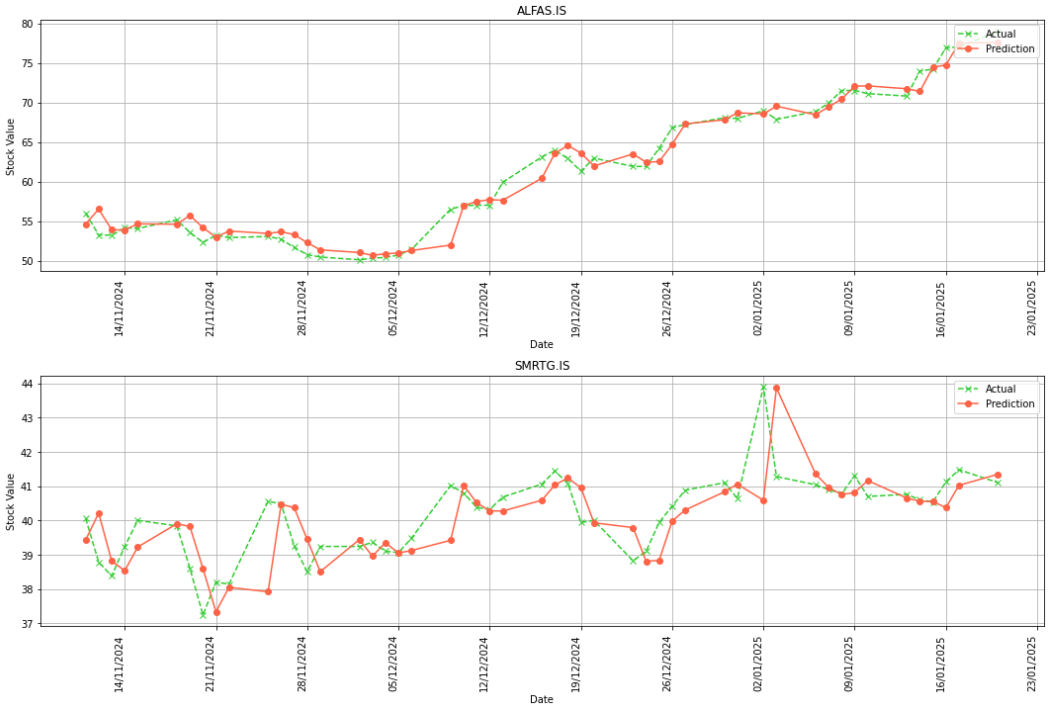

Figure 4.6 Forecast & ground truth for ALFAS.IS and SMRTG.IS

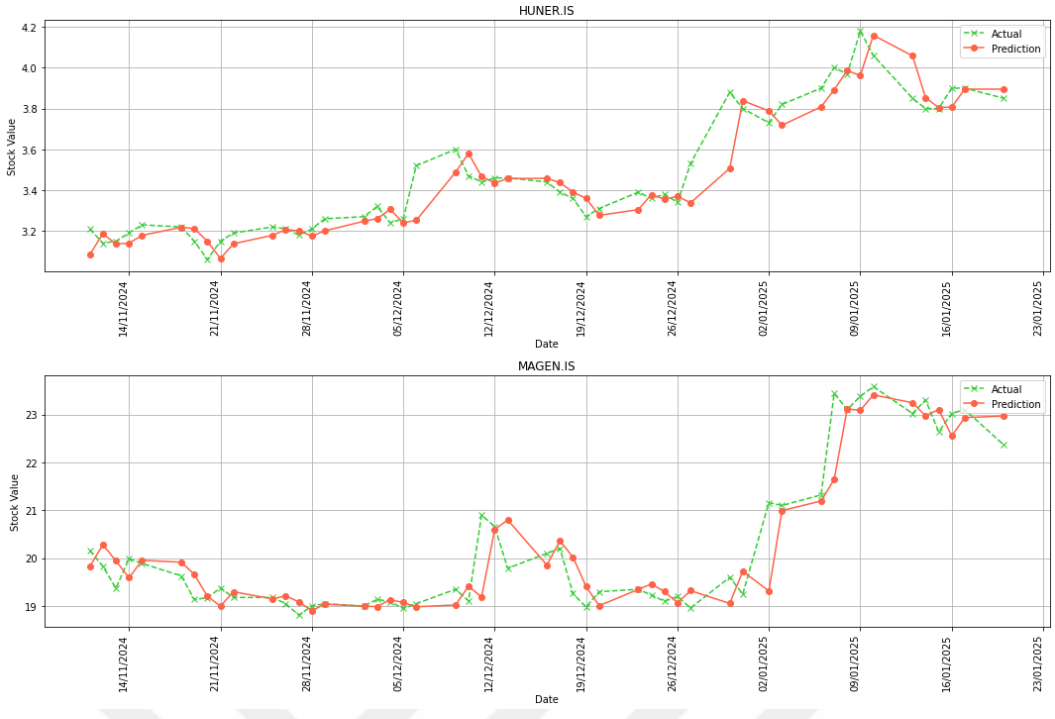

Figure 4.7 Forecast & ground truth for HUNER.IS and MAGEN.IS



Figure 4.8 Forecast & ground truth for KARYE.IS and BIOEN.IS

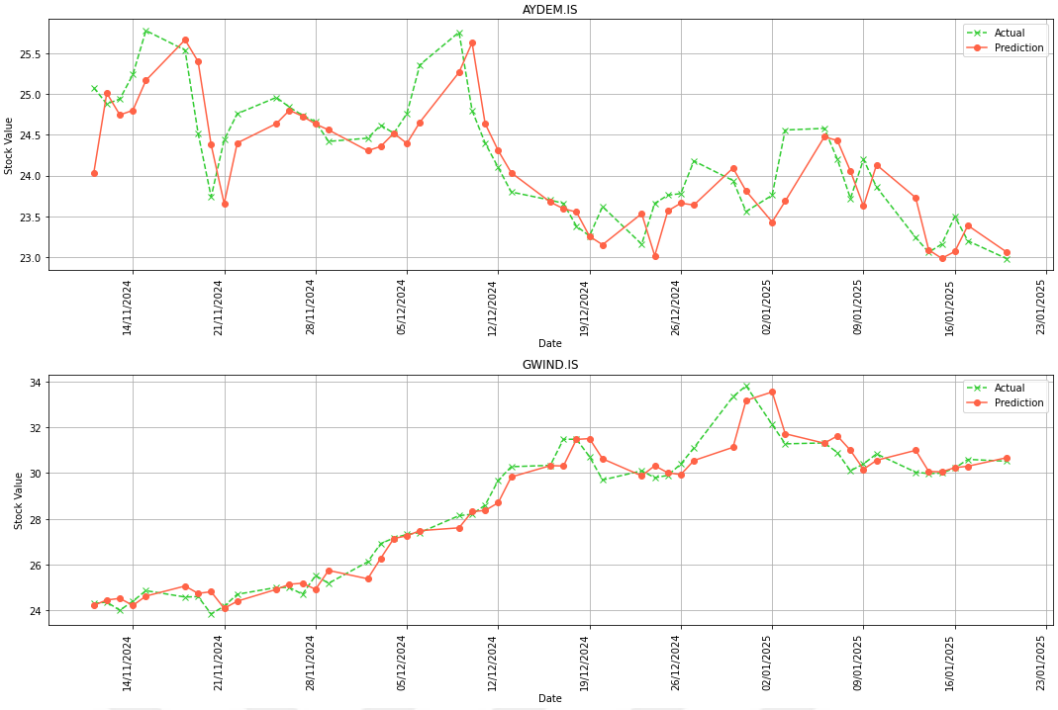

Figure 4.9 Forecast & ground truth for AYDEM.IS and GWIND.IS

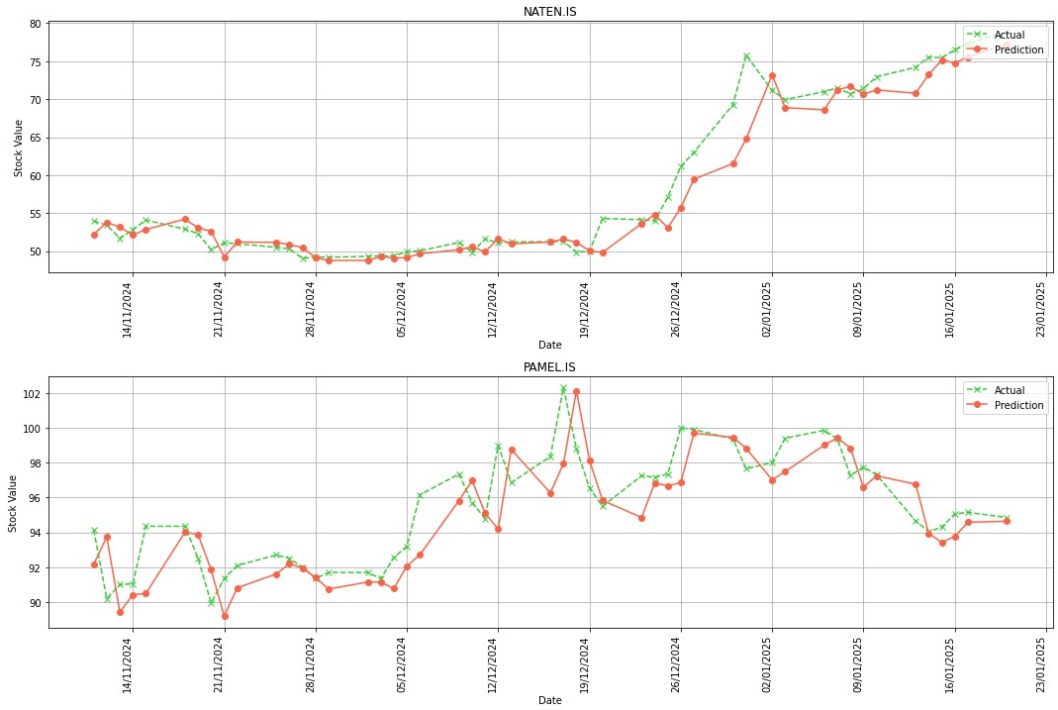

Figure 4.10 Forecast & ground truth for NATEN.IS and PAMEL.IS

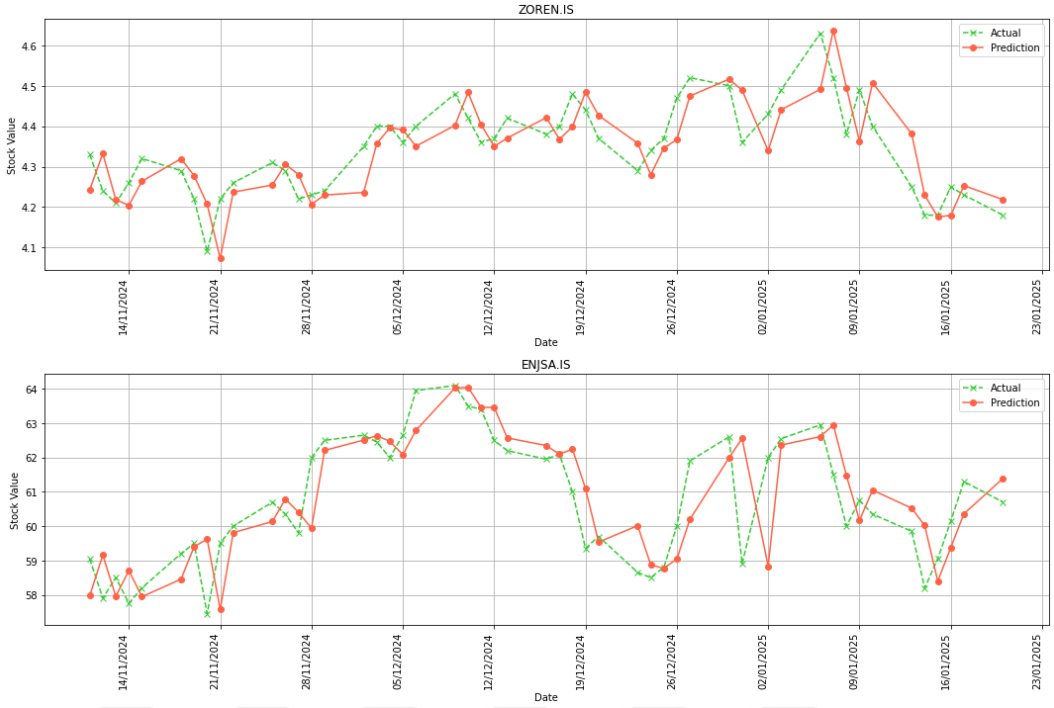

Figure 4.11 Forecast & ground truth for ZOREN.IS and ENJSA.IS

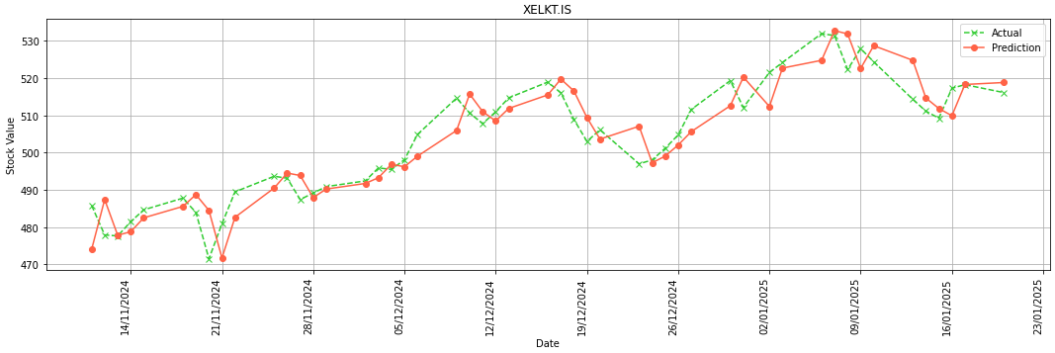

Figure 4.12 Forecast & ground truth for XELKT.IS

4.5 Portfolio Construction Using Risk-Adjusted Optimization Strategies

Four distinct portfolio optimization strategies were implemented using the annualized expected return vector and covariance matrix of the stocks using a period which has equal up and down beta dates. 05-12-2024 and 30-01-2025 period has 20 up and 20 down beta days. First, the **MV Portfolio** was constructed to minimize overall risk without considering return, subject to full investment and no short-selling constraints. Second, the **MV-SR Portfolio** was derived by maximizing the extra return earned for each unit of risk taken, adding the Sharpe ratio as a objective function. Risk-free rate was set to 0% to isolate portfolio performance from external yield curve effects.

To incorporate asymmetrical market reactions, **MV-AB Portfolio** model which adds asymmmetric beta objectives into MV model, penalizing downside beta while rewarding upside beta. Finally, **MV-AB-SR Portfolio** model is constructed.

These are particularly important in risk-aware portfolio construction, especially for investors who exhibit greater sensitivity to financial losses compared to equivalent gains.

Evolutionary algorithms, the detailed information of which is provided in the methodology section, were used for optimization. All multiobjective portfolio strategies are solved with Non-Dominated Sorting Genetic Algorithm 2 (NSGA-II) and elbow/knee method is used to select best result.

The performance of each portfolio was evaluated in terms of return, volatility, Sharpe ratio, upside and downside portfolio beta, Sortino ratio which is a risk-adjustment metric used to determine the additional return for each unit of downside risk, Max Drawdown which represents the largest observed decline in the portfolio from its peak and CAGR(compound annual growth rate). Results were visualized using scatter plots and beta bar charts to highlight the trade-off between return and risk under different optimization objectives.

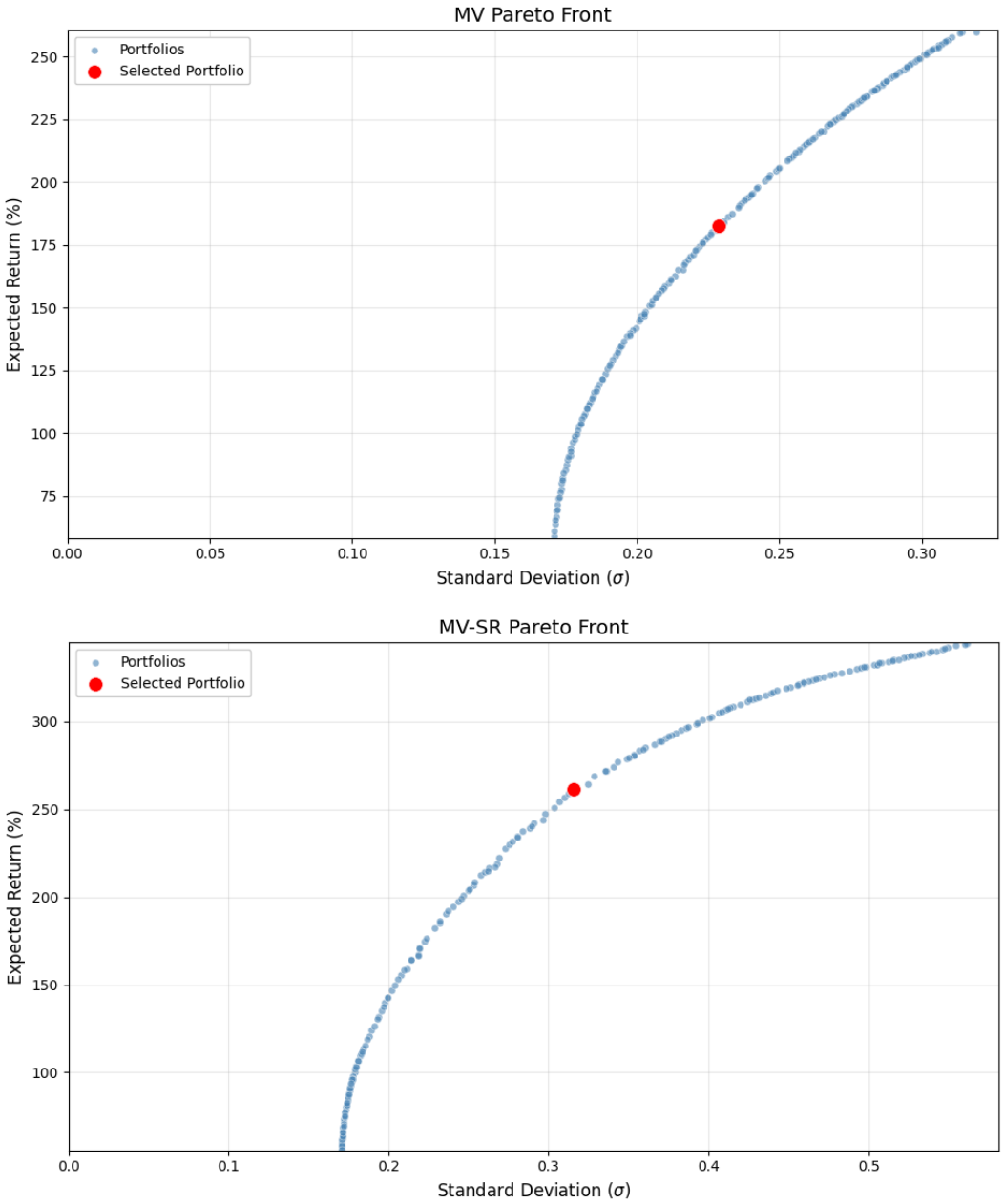

Figure 4.13 2D Pareto front visualization using return and variance objective values of MV and MV-SR model.

It can be seen that Pareto front resembles efficient frontier, because MV and SR model objectives are directly related to return and standard devitation. So optimum points are same as stocks efficient frontier points.

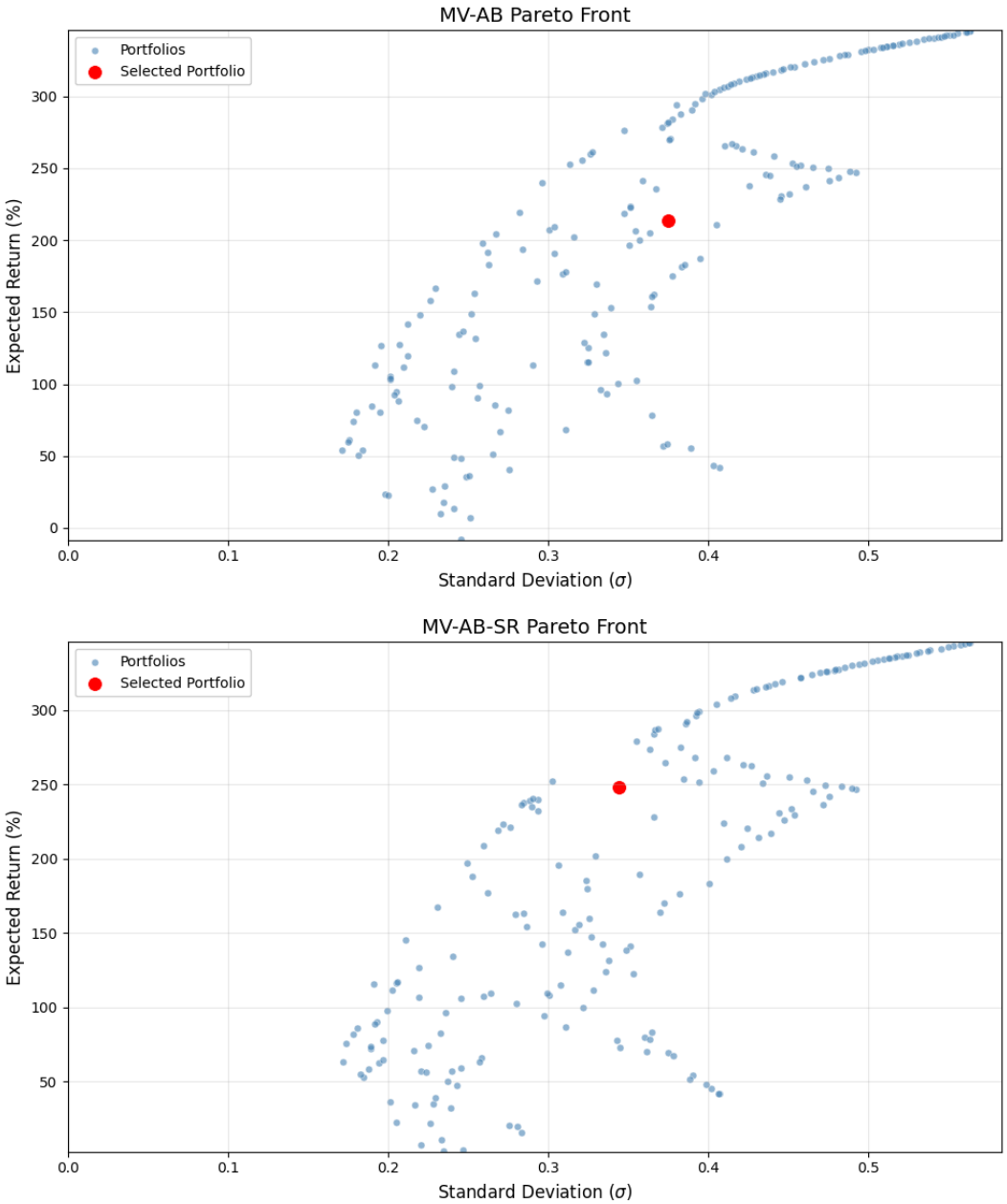

Figure 4.14 2D Pareto front visualization using return and variance objective values of MV-AB and MV-AB-SR model.

On the other hand, MV-AB and MV-AB-SR models incorporate upside and downside beta in their objective functions, so optimal portfolio selection considers these asymmetric beta values, as a result their Pareto front visualizations (using only return and standard deviation) differ from traditional efficient frontier points.

Table 4.1 Calculated portfolio weights using forecast values.

Stock	MV	MV-SR	MV-AB	MV-AB-SR
ALFAS.IS	0.11	0.23	0.71	0.64
SMRTG.IS	0.04	0.00	0.04	0.02
HUNER.IS	0.00	0.00	0.00	0.00
MAGEN.IS	0.17	0.24	0.25	0.02
KARYE.IS	0.17	0.06	0.00	0.07
BIOEN.IS	0.00	0.00	0.00	0.00
AYDEM.IS	0.09	0.00	0.00	0.00
GWIND.IS	0.17	0.09	0.00	0.00
NATEN.IS	0.12	0.26	0.00	0.24
PAMEL.IS	0.02	0.00	0.00	0.00
ZOREN.IS	0.00	0.00	0.00	0.00
ENJSA.IS	0.12	0.12	0.01	0.00

As seen in 4.1, all strategies are mainly focused on few stocks: ALFAS, MAGEN, NATEN. ALFAS has the highest weight in all scenarios with 0.71. HUNER and ZOREN are never used in any portfolios.

Table 4.2 Portfolio comparison of strategies, evaluated with ground truth values.

Strategy	Return	Std	Sharpe	Sortino	Max Drawdown	CAGR
MV	1.16	0.127	1.94	4.12	-0.04	2.61
MV-SR	1.32	0.162	2.72	6.71	-0.04	5.60
MV-AB	1.36	0.247	2.04	4.16	-0.11	6.91
MV-AB-SR	1.43	0.237	2.47	5.61	-0.10	9.57

We can see in 4.2, MV-AB-SR strategy has highest return,sharpe ratio and sortino ratio, whereas MV and MV-SR has the lowest standard deviation. Max drawdown is lowest in MV and MV-SR strategies.

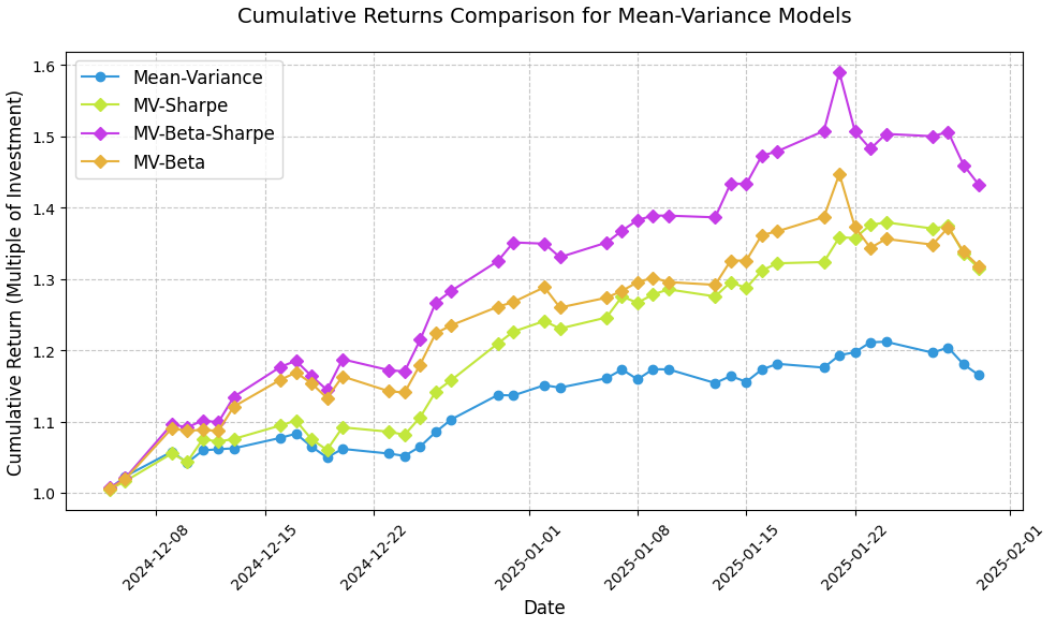

Figure 4.15 Cumulative returns of Mean-Variance included strategies.

The effect of optimization objectives can be clearly seen in 4.15. classic MV model has the lowest returns. When sharpe is added to the objectives, returns increased. Adding beta objectives yielded even better results than adding sharpe. Combination of all three methods gave the highest returns.

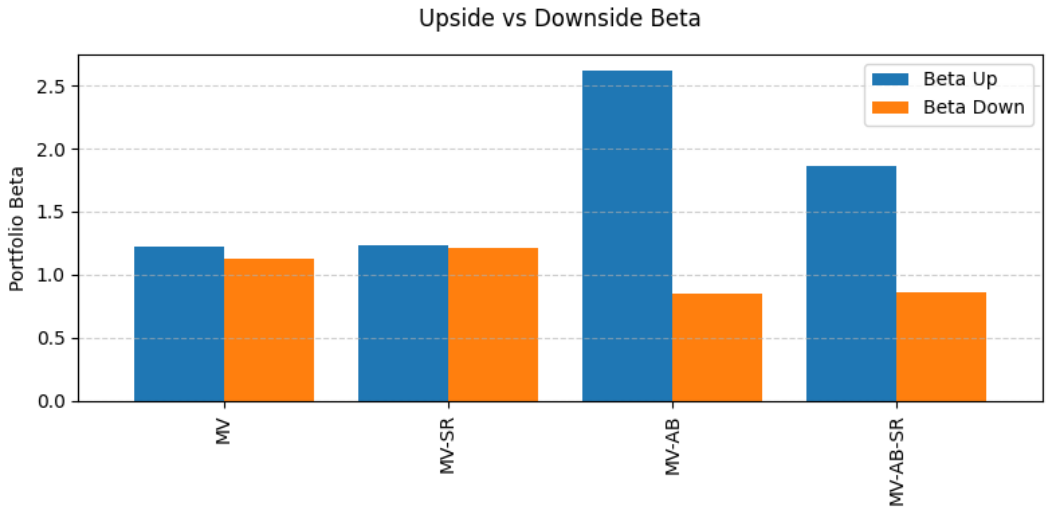

Figure 4.16 Beta Ratios of each strategy portfolios

Beta enchanced strategies has higher upside beta than downside beta as expected. Other strageies has evenly distributed beta values.

5
CONCLUSION

This thesis studied how deep learning techniques can predict stock returns in Borsa Istanbul's renewable energy market. It used different models of Recurrent Neural Networks like RNN, GRU, and LSTM, alongside Amazon's Chronos, to forecast stock prices one day ahead. The focus was on daily stock prices of twelve companies from the BIST Electricity Index. The performance of these models was checked using error metrics such as MSE, MAE, RMSE, MAPE, and MDA. LSTM and GRU had overall better accuracy than other models.

Predicted returns were used in four strategies of portfolio optimization: MV, MV-SR, MV-AB and MV-AB-SR. From the results, it is shown that using beta as enhancement over classic MV yields better returns.

This study shows how deep learning can model financial time series data well, and give useful insights to guide investment. It highlights how using deep learning models in portfolio building can help improve decisions and manage risks in unpredictable sectors like renewable energy. Upside and downside beta parameters are also inspected in this study and proved that beta parameters can improve optimization. The research proves the growing usefulness of deep learning methods in today's financial analysis.

REFERENCES

[1] H. Markowitz, "Portfolio selection*," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952. doi: 10.1111/j.1540-6261.1952.tb01525.x.

[2] L. Trung Hieu, "Deep reinforcement learning for stock portfolio optimization," *International Journal of Modeling and Optimization*, vol. 10, no. 5, pp. 139–144, Oct. 2020, issn: 2010-3697. doi: 10.7763/ijmo.2020.v10.761.

[3] C. Zhang, Z. Zhang, M. Cucuringu, S. Zohren, *A universal end-to-end approach to portfolio optimization via deep learning*, 2021. arXiv: 2111.09170 [q-fin.PM]. [Online]. Available: https://arxiv.org/abs/2111.09170.

[4] Ta, C.-M. Liu, Tadesse, "Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading," *Applied Sciences*, vol. 10, p. 437, Jan. 2020. doi: 10.3390/app10020437.

[5] F. S. Mishkin, S. G. Eakins, *Financial markets*. Pearson Italia, 2019.

[6] M. Company, *What is inflation: The causes and impact*, Accessed via McKinsey Explainers, 2024. [Online]. Available: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-inflation.

[7] P. Krugman, R. Wells, *Macroeconomics*, 5th. Worth Publishers, 2018.

[8] Z. Zulkifli, "Investigation into the fundamental analysis of stock investment instruments on the indonesian stock exchange," *Economy*, 2023. doi: 10.61996/economy.v1i1.7.

[9] M. "Firth, "The role of the stock market," in *The Valuation of Shares and the Efficient-Markets Theory*. London: Macmillan Education UK, 1977, pp. 1–13. doi: 10.1007/978-1-349-15819-5_1.

[10] C. Helmenstein, C. Haefke, "Stock market indices," in *Wiley Encyclopedia of Management*. John Wiley Sons, Ltd, 2015, ch. 11, pp. 1–2, isbn: 9781118785317. doi: 10.1002/9781118785317.weom040072.

[11] A. Yadav, R. Madhavi, O. Bagaria, A. Ambulkar, S. Sharma, "Survey on financial portfolio management's role in investment decision-making strategies.," *Multidisciplinary Reviews*, vol. 6, p. 101, 2023. doi: 10.31893/multirev.2023ss101.

[12] R. Kumar, "2 - risk and return," in *Valuation*, R. Kumar, Ed., San Diego: Academic Press, 2016, pp. 47–72, isbn: 978-0-12-802303-7. doi: 10.1016/B978-0-12-802303-7.00002-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128023037000024.

[13] P. Barroso, P. Maio, "The risk–return tradeoff among equity factors," *Journal of Empirical Finance*, vol. 78, p. 101 518, 2024, issn: 0927-5398. doi: 10.1016/j.jempfin.2024.101518. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927539824000537.

[14] M. Statman, "How many stocks make a diversified portfolio?" *Journal of Financial and Quantitative Analysis*, vol. 22, no. 3, pp. 353–363, 1987. doi: 10.2307/2330969.

[15] D. S. Fransiska, N. Rahmawati, F. G. Saputra, M. Y. R. Pandin, "Stock diversification (portfolio) strategy to increase investment returns and reduce risk," *Finance : International Journal of Management Finance*, vol. 1, no. 2, pp. 33–39, Dec. 2023. doi: 10.62017/finance.v1i2.14. [Online]. Available: https://ejournal.internasional.org/index.php/finance/article/view/14.

[16] E. J. Elton, M. J. Gruber, S. J. Brown, W. N. Goetzmann, *Modern Portfolio Theory and Investment Analysis*, 9th ed. Wiley, 2014.

[17] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. doi: 10.1038/nature14539.

[18] L. Bottou, "Stochastic gradient descent tricks," G. Montavon, G. Orr, K.-R. Müller, Eds., pp. 421–436, 2012.

[19] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2014.

[20] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1998.

[21] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org.

[22] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016. [Online]. Available: https://arxiv.org/abs/1609.04747.

[23] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986. doi: 10.1038/323533a0.

[24] Z. C. Lipton, J. Berkowitz, C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015. [Online]. Available: https://arxiv.org/abs/1506.00019.

[25] C. Olah, *Understanding lstm networks*, 2015. arXiv: 2111.09170 [q-fin.PM]. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[26] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, "Deep learning for time series classification: A review," *Data Mining and Knowledge Discovery*, vol. 33, pp. 917–963, 2019. doi: 10.1007/s10618-019-00619-1.

[27] J.-m. Yang, J. Kim, "An accident diagnosis algorithm using lstm," *Nuclear Engineering and Technology*, vol. 50, Mar. 2018. doi: 10.1016/j.net.2018.03.010.

[28] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen netzen," Cited on page 11, Master's thesis, Technische Universität München, Institut für Informatik, 1991.

[29] Y. Bengio, P. Simard, P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994, Cited on page 11.

[30] R. Pascanu, T. Mikolov, Y. Bengio, "On the difficulty of training recurrent neural networks," University of Montreal, Technical Report, 2012, Cited on pages 11, 12, and 55. arXiv: arXiv:1211.5063.

[31] J. S. Bayer, "Learning sequence representations," München, Technische Universität München, Diss., 2015, Doctoral dissertation, Technische Universität München, 2015.

[32] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[33] K. Cho et al., "Learning phrase representations using rnn encoder–decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014. [Online]. Available: https://arxiv.org/abs/1406.1078.

[34] I. Bibi, A. Akhunzada, J. Malik, J. Iqbal, A. Musaddiq, S. Kim, "A dynamic dl-driven architecture to combat sophisticated android malware," *IEEE Access*, vol. PP, pp. 1–1, Jul. 2020. doi: 10.1109/ACCESS.2020.3009819.

[35] A. F. Ansari et al., *Chronos: Learning the language of time series*, 2024. arXiv: 2403.07815 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2403.07815.

[36] D. P. Kingma, J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1412.6980.

[37] J. Bergstra, Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, 2012. [Online]. Available: http://jmlr.org/papers/v13/bergstra12a.html.

[38] J. Bergstra, Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. null, pp. 281–305, Feb. 2012, issn: 1532-4435.

[39] R. J. Hyndman, G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed. OTexts, 2018. [Online]. Available: https://otexts.com/fpp2/.

[40] L. J. Tashman, “Out-of-sample tests of forecasting accuracy: An analysis and review,” *International Journal of Forecasting*, vol. 16, no. 4, pp. 437–450, 2000.

[41] V. Cerqueira, L. Torgo, C. Soares, “Evaluating time series forecasting models: An empirical study on performance estimation methods,” *Machine Learning*, vol. 109, pp. 1997–2028, 2020.

[42] C. Bergmeir, R. J. Hyndman, B. Benítez, “Bagging exponential smoothing methods using stl decomposition and box–cox transformation,” *International Journal of Forecasting*, vol. 32, no. 2, pp. 303–312, 2016.

[43] J. Han, M. Kamber, J. Pei, *Data Mining: Concepts and Techniques*, 4th ed. Elsevier, 2022.

[44] J. Brownlee, *Deep Learning for Time Series Forecasting*. Machine Learning Mastery, 2017.

[45] S. Ioffe, C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[46] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002. doi: 10.1109/4235.996017.

PUBLICATIONS FROM THE THESIS

Conference Papers

1. O. Alp, G. Kemalbay, *"Deep Learning Models for Forecasting BIST Renewable Energy Stocks"*, International Marmara Science and Social Sciences Congress (IMASCON) 2025 Spring, Sciences Proceedings Book, p. 56, May 16–17, 2025.