Regime-Dependent Portfolio Optimization: An Integrated Framework of Statistics, Risk-Parity and Deep Learning

Thesis submitted in partial fulfilment

for the Award of

INTEGRATED DUAL DEGREE (B.TECH.+ M.TECH.)

in

MATHEMATICS AND COMPUTING

by

ADITYA KULKARNI

DEPARTMENT OF MATHEMATICAL SCIENCES

INDIAN INSTITUTE OF TECHNOLOGY

(BANARAS HINDU UNIVERSITY)

VARANASI – 221 005

ROLL NUMBER	YEAR OF SUBMISSION
20124054	2025

CERTIFICATE

It is certified that the work contained in the thesis titled **Regime-Dependent Portfolio Optimization: An Integrated Framework of Statistics, Risk-Parity and Deep Learning** by **Aditya Kulkarni (Roll No. 20124054)** has been carried out under my supervision at Indian Institute of Technology (Banaras Hindu University), Varanasi and that this work has not been submitted elsewhere for a degree/ diploma. It is certified that the report embodies the results of the work carried out by him within the prescribed period.

Supervisor
Dr. Sanjay Kumar Pandey
Professor
Department of Mathematical Sciences
Indian Institute Of Technology
(Banaras Hindu University)
Varanasi – 221 005

DECLARATION BY THE CANDIDATE

I, **Aditya Kulkarni**, certify that the work embodied in this thesis is my own bona fide work and carried out by me under the supervision of **Dr. Sanjay Kumar Pandey** from **January 2024** to **May 2025**, at the **Department of Mathematical Sciences**, Indian Institute of Technology (Banaras Hindu University), Varanasi. The matter embodied in this thesis has not been submitted for the award of any other degree/diploma. I declare that I have faithfully acknowledged and given credits to the research workers wherever their works have been cited in my work in this thesis. I further declare that I have not willfully copied any other's work, paragraphs, text, data, results, *etc.*, reported in journals, books, magazines, reports dissertations, theses, *etc.*, or available at websites and have not included them in this thesis and have not cited as my own work.

Date:
Place: Varanasi, India

Signature of the Student

CERTIFICATE BY THE SUPERVISOR

It is certified that the above statement made by the student is correct to the best of my/our knowledge.

Supervisor
Prof. Sanjay Kumar Pandey
Department of Mathematical Sciences
Indian Institute of Technology (BHU)
Varanasi – 221 005

Head of the Department
Department of Mathematical Sciences
Indian Institute of Technology (BHU)
Varanasi – 221 005

COPYRIGHT TRANSFER CERTIFICATE

Title of the Thesis: Regime-Dependent Portfolio Optimization: An Integrated Framework of Statistics, Risk-Parity and Deep Learning

Name of the Student: Aditya Kulkarni

COPYRIGHT TRANSFER

ACKNOWLEDGEMENT

First and foremost, I would like to thank my supervisor, Dr. Sanjay Kumar Pandey, Professor, Department of Mathematical Sciences, IIT(BHU), Varanasi. This work would not have been possible if not for their continuous guidance, insightful suggestions, constructive criticism, and constant support.

I would also like to thank Dr. Subir Das, Head of the Department, Department of Mathematical Sciences for his help in providing all the necessary facilities to carry out this research. I would like to express my gratitude to the faculty members, technical staff, and non-technical staff of the Department of Mathematical Sciences for their invaluable assistance, support, and cooperation. I am also grateful to my parents, siblings, and friends for providing me with constant encouragement and moral support throughout my study and research period.

Finally, I am indebted and grateful to the Almighty for helping me in this endeavor.

Date:
Place: Varanasi, India

Signature of the Student

Abstract

Portfolio optimization remains a cornerstone of quantitative finance, aiming to balance return and risk effectively. Building upon the foundational work of Sen and Dasgupta, this thesis introduces a novel adaptive regime-switching framework to enhance portfolio performance.

The proposed methodology employs a Hidden Markov Model (HMM) to identify latent market regimes based on index-level return dynamics. Depending on the inferred regime, the framework dynamically selects among various portfolio construction techniques, including Black-Litterman blended models, Monte Carlo-approximated MVP, and Conditional Value-at-Risk (CVaR) minimization. This adaptive strategy allows the portfolio to adjust to changing market conditions, aiming for improved risk-adjusted returns.

The framework is trained on historical data spanning from January 1, 2018, to December 31, 2021, and its performance is evaluated on out-of-sample data from January 1, 2022, to December 31, 2022. Comparative analysis demonstrates that while static methods like MVP and Autoencoder models perform well under certain market conditions, the adaptive approach achieves superior Sharpe ratios and more stable cumulative returns by dynamically adjusting to changing volatility and correlation structures.

This integration of regime detection with multi-method portfolio construction offers a promising direction for robust investment strategies in emerging and dynamic markets. The findings suggest that incorporating regime-switching mechanisms can enhance portfolio resilience and adaptability, providing valuable insights for both academic research and practical asset management applications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

In an era where market regimes shift overnight, adaptability is no longer optional—it's the alpha. This thesis reimagines portfolio optimization as a dynamic, learning process—where models don't just calculate risk, they anticipate it.

Portfolio optimization lies at the core of modern investment strategies, guiding asset allocation decisions to achieve an optimal trade-off between expected returns and associated risks. Traditional techniques, most notably the Markowitz Mean-Variance Portfolio (MVP) framework [28], have fundamentally shaped this domain. Subsequent advancements like Hierarchical Risk Parity (HRP) portfolios [25] and various machine learning-driven approaches [16, 11] have further enhanced asset selection methodologies.

However, financial markets are not static systems; they exhibit complex behavior characterized by dynamic regime shifts, volatility clustering, and nonlinear dependencies [1, 13]. Static optimization models, while valuable, often fail to adapt to these structural changes. This motivates the need for dynamic, regime-aware portfolio management techniques that account for changing market conditions, thereby leading to more robust and adaptive investment strategies.

1.2 Core Concepts

1.2.1 Regime-Dependent Modeling

Markets often operate under multiple latent states or regimes (e.g., bullish, bearish, or sideways). *Regime-dependent* modeling refers to the approach of detecting and responding to these hidden states, allowing investment strategies to dynamically shift their behavior based on market conditions. Techniques such as Hidden Markov Models (HMMs) [42, 39] are frequently employed to model these latent regimes. These models enable the identification of underlying market conditions that influence asset returns, helping portfolio managers adjust asset allocations in real-time. By incorporating regime-switching, investors can better manage risk and capture opportunities during different market phases, such as protecting portfolios during bear markets or increasing exposure during bullish trends [13]. Furthermore, recent advancements in regime-dependent models have explored the integration of machine learning algorithms, allowing for more flexible and adaptive regime detection [38]. This approach not only improves forecasting accuracy but also enhances the robustness of investment strategies by making them more responsive to evolving market dynamics [11].

1.2.2 Portfolio Optimisation

Portfolio optimisation is the process of selecting asset weights that maximize return for a given level of risk, or vice versa. While the classical Markowitz framework [28] relies on the mean and variance of asset returns, modern enhancements incorporate risk-sensitive metrics [43], regularisation [23], and investor views [15, 19] to improve stability and robustness. The introduction of Conditional Value-at-Risk (CVaR) [43] has further refined portfolio optimization by considering extreme losses, especially in tail-risk scenarios. Additionally, recent research has extended traditional models to account for regime shifts [38], incorporating dynamic asset allocations that respond to changes in market conditions. The evolving nature of computational finance has led to the use of machine learning models that dynamically optimize portfolio weights, providing more adaptable strategies compared to static models [11].

1.2.3 Risk Parity

Risk parity is an asset allocation strategy that equalizes the contribution of each asset to overall portfolio risk [44]. Unlike MVP, which may over-allocate to low-volatility assets, risk parity ensures a more balanced risk distribution, making it appealing for volatility management and crisis periods [25]. The HRP method [26] takes this concept further by applying hierarchical clustering, offering a more efficient and scalable solution for constructing diversified portfolios. Moreover, risk parity portfolios tend to perform well in environments where asset correlations are unstable or when volatility is high [25], as they focus on balancing the risk across asset classes rather than relying solely on historical return correlations. This is particularly important in the context of modern portfolio management, where market conditions are often non-stationary, and risk needs to be managed dynamically [44].

1.2.4 Statistics and Machine Learning

Statistical modeling plays a vital role in financial forecasting, particularly in return estimation and risk modeling. In this work, statistics guide regime detection (e.g., HMMs [13]), while *deep learning*, especially autoencoders [17, 9], is used for dimensionality reduction and capturing nonlinear patterns in asset returns. The combination of both paradigms leads to a more flexible and powerful optimization process. HMMs are particularly useful for detecting latent market regimes, which are key for adjusting portfolio allocations based on market conditions [42]. On the other hand, deep learning techniques, such as autoencoders, can uncover complex relationships between assets, improving feature extraction and enabling the construction of more predictive models for portfolio optimization [6]. Additionally, hybrid models that combine machine learning and statistical methods are becoming increasingly popular for their ability to adapt to changing financial environments, offering improved forecasting accuracy and risk management [16].

1.3 Problem Statement

Despite extensive literature on portfolio optimization [8], most existing methods assume stationarity in return distributions and market conditions. Static models optimized over historical data may perform sub-optimally during unexpected regime shifts or structural market transitions.

This research addresses the gap by proposing an integrated regime-dependent framework that dynamically adjusts asset weights based on detected market states, blending traditional optimization techniques with deep learning models to enhance portfolio resilience and performance.

1.4 Objectives of the Study

The specific objectives of this thesis are as follows:

- To implement and benchmark static portfolio optimization methods including Minimum Variance Portfolio (MVP), Hierarchical Risk Parity (HRP), and Autoencoder-based portfolio allocation.

- To develop a novel Adaptive Regime-Switching Portfolio Optimization model that responds dynamically to market regime changes using Hidden Markov Models (HMMs).

- To compare and analyze the performance of static and adaptive strategies across training and test periods using metrics such as annual return, annual volatility, and Sharpe ratio.

- To extend the current body of knowledge by integrating statistical modeling, risk parity techniques, and deep learning into a unified portfolio management framework.

1.5 Scope of the Research

This study focuses on ten thematic sectors from the National Stock Exchange (NSE) of India, including Commodities, Energy, Infrastructure, and Manufacturing. For each sector, the top ten stocks by free-float market capitalization, as reported on February 29, 2022 [36], are selected.

The historical data spans from January 1, 2018, to December 31, 2022, divided into a training window (2018–2021) and a test window (2022) to facilitate robust model evaluation. This time horizon captures both stable and volatile market phases, including the COVID-19 shock and post-pandemic recovery. The sector-level granularity allows for a detailed comparison of how adaptive strategies perform under varying industry-specific risk-return profiles.

1.6 Data Sources

Daily adjusted closing prices are extracted from the NSE official database using Python libraries [32, 5], ensuring data integrity and reproducibility. Only closing prices are used for consistency with return calculations and portfolio simulations.

To maintain robustness, data is cached locally to prevent inconsistencies due to API outages or retroactive changes. In case of unavailability, fallback mechanisms such as Stooq or synthetic index proxies are employed to fill gaps and maintain continuity.

1.7 Research Contributions

The key contributions of this thesis are summarized below:

- Comprehensive implementation of traditional static portfolio optimization models.
- Design and implementation of an adaptive regime-switching strategy using a novel integration of Hidden Markov Models and portfolio optimization.
- Extensive empirical evaluation across multiple thematic sectors, comparing static

and dynamic portfolio strategies.

- Demonstration of improved adaptability and robustness through the proposed framework compared to traditional methods.

Chapter 2

Literature Review

2.1 Introduction

Portfolio optimisation is a fundamental aspect of financial management, aiming to allocate assets to maximise returns while controlling risk. Traditional models, such as the Mean-Variance Portfolio (MVP) introduced by Markowitz [29], provide a foundation for this process. However, advancements in computational techniques have led to the development of alternative models like Hierarchical Risk Parity (HRP) and Autoencoder-based portfolios, which address some limitations of traditional approaches.

2.2 Mean-Variance Portfolio (MVP)

2.2.1 Overview

The Mean-Variance Portfolio (MVP) model, introduced by Markowitz [29], allocates portfolio weights to minimise risk (variance) for a given expected return or to maximise return for a fixed risk level. This approach computes stock returns and volatilities using historical data, where daily returns are calculated as the percentage change or logarithmic return between consecutive trading days [28]. Volatility is determined as the standard deviation of daily returns, annualized by scaling with the square root of 250 [45]. Covariance and correlation matrices are then computed to quantify the relationships between stock returns [25], helping to identify diversification opportunities and minimize portfolio risk by including assets with low mutual correlations [43].

2.2.2 Computation of Returns and Volatilities

The daily return $r_{i,t}$ for stock i on day t can be calculated as:

$$r_{i,t} = \frac{P_{i,t} - P_{i,t-1}}{P_{i,t-1}} \tag{2.1}$$

or as a logarithmic return:

$$r_{i,t} = \log\left(\frac{P_{i,t}}{P_{i,t-1}}\right) \tag{2.2}$$

The daily volatility σ_i is the standard deviation of daily returns, and the annual volatility is calculated by:

$$\sigma_{i,\text{annual}} = \sigma_i \cdot \sqrt{250} \tag{2.3}$$

2.2.3 Covariances and Correlations of Stocks

The covariance between stocks i and j is computed as:

$$\text{Cov}(i,j) = \frac{1}{T-1}\sum_{t=1}^{T}(r_{i,t} - \bar{r}_i)(r_{j,t} - \bar{r}_j) \tag{2.4}$$

The correlation between stocks i and j is given by:

$$\rho_{i,j} = \frac{\text{Cov}(i,j)}{\sigma_i \sigma_j} \tag{2.5}$$

These matrices help evaluate the diversification effects in the portfolio.

2.2.4 Portfolio Returns and Risks

For a portfolio consisting of n assets, with weights $w_1, w_2, ..., w_n$, the expected return of the portfolio is:

$$E(R) = \sum_{i=1}^{n} w_i E(R_i) = \mathbf{w}^T \mu \tag{2.6}$$

The variance of the portfolio is calculated as:

$$\text{Var}(P) = \sum_{i=1}^{n} w_i^2 \sigma_i^2 + \sum_{i=1}^{n} \sum_{j \neq i}^{n} w_i w_j \text{Cov}(i, j) \tag{2.7}$$

Or in matrix form:

$$\sigma_p^2 = \mathbf{w}^T \Sigma \mathbf{w} \tag{2.8}$$

This equation shows the full variance, considering both individual asset variances and the covariances between assets.

2.2.5 Sharpe Ratio and Optimal Portfolio

The Sharpe Ratio (SR) measures the risk-adjusted return:

$$\text{SR} = \frac{R_c - R_f}{\sigma_c} \tag{2.9}$$

where R_c is the portfolio return, R_f is the risk-free rate, and σ_c is the standard deviation of portfolio returns. The optimal portfolio on the efficient frontier maximises the Sharpe Ratio.

2.2.6 Efficient Frontier

The efficient frontier represents the optimal portfolios with the highest return for a given level of risk, with portfolios below it being sub-optimal. Below is the figure for the Efficient Frontier constructed from 10,000 random portfolios. The red star represents the minimum risk portfolio; the green star indicates the maximum Sharpe ratio (optimal) portfolio.

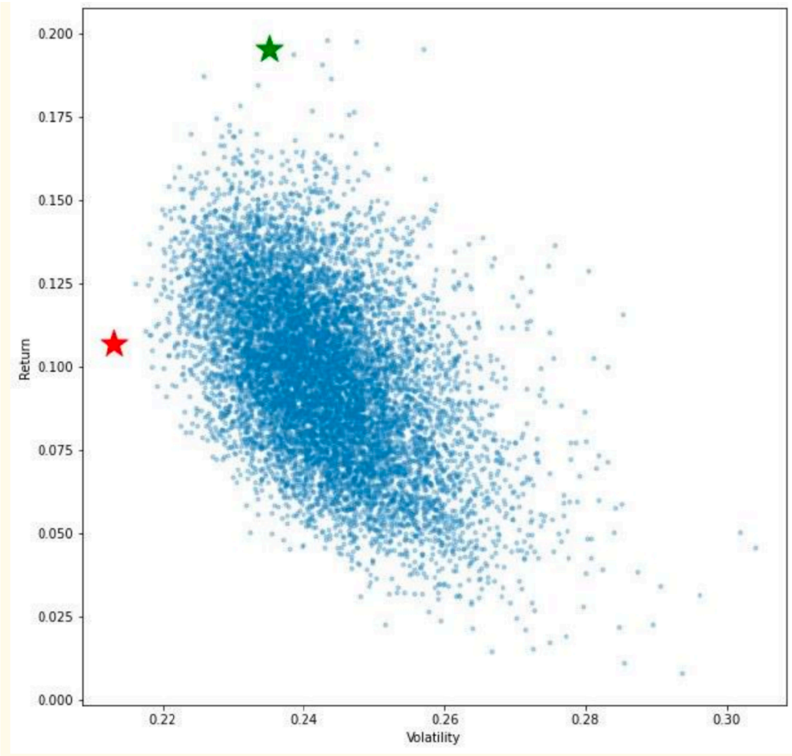

Figure 2.1: Efficient Frontier for Portfolio

2.2.7 Algorithm Simulation

Algorithm 1 Mean-Variance Portfolio Optimisation

1: Compute daily returns $r_i(t)$ for each stock i
2: Compute expected return $\mu_i = \mathbb{E}[r_i]$ and covariance matrix Σ
3: Set target return μ_p
4: Solve quadratic program:

$$\min_w w^\top \Sigma w \quad \text{s.t.} \quad w^\top \mu = \mu_p, \quad \sum w_i = 1$$

5: Output optimal weights w^*

2.3 Hierarchical Risk Parity (HRP) Methodology

The Hierarchical Risk Parity (HRP) algorithm provides an alternative to classical mean-variance optimization by constructing diversified portfolios using hierarchical clustering, avoiding the need for matrix inversion. The three major steps in HRP are: *(a) Formation of Clusters*, *(b) Quasi-Diagonalisation*, and *(c) Recursive Bisection* [26, 45].

2.3.1 Formation of Clusters

To cluster assets, we use a distance metric derived from the Pearson correlation matrix ρ:

$$d_{ij} = \sqrt{0.5(1 - \rho_{ij})}$$

This distance metric satisfies Euclidean properties and enables hierarchical clustering using Ward's linkage. Assets form a tree structure, visualized as a dendrogram (Figure 2.2), where lower branches represent more correlated assets [26].

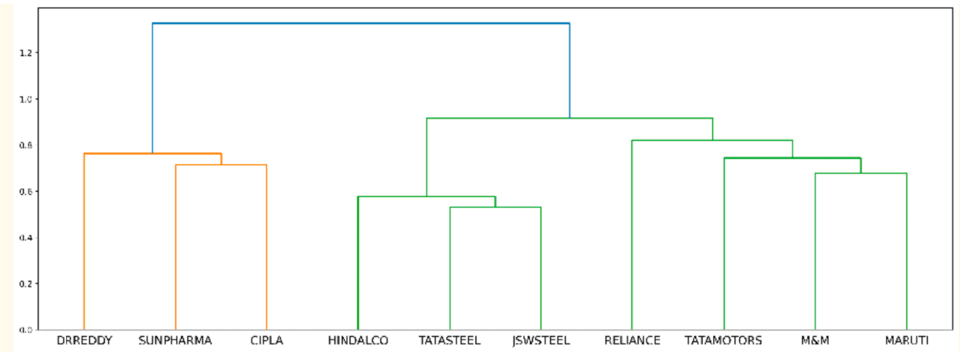

Figure 2.2: Dendrogram from hierarchical clustering

2.3.2 Quasi-Diagonalisation

The covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ is reordered based on the hierarchical structure such that correlated assets appear near the diagonal. The reordering is achieved via a permutation matrix P corresponding to the asset order π:

$$\Sigma^{\pi} = P \Sigma P^{\top}$$

This structure facilitates efficient risk-based allocation by grouping similar assets [43].

2.3.3 Recursive Bisection

After quasi-diagonalisation, the HRP algorithm allocates weights recursively. Given a cluster C split into sub-clusters C_L and C_R, the variance of each cluster is:

$$\sigma^2(C) = w^\top \Sigma_C w \quad \text{where} \quad w_i = \frac{1/\sigma_i^2}{\sum_{j \in C} 1/\sigma_j^2}$$

The weights for the sub-clusters are then assigned based on their variances:

$$w_{C_L} = \frac{\sigma^2(C_R)}{\sigma^2(C_L) + \sigma^2(C_R)}, \quad w_{C_R} = \frac{\sigma^2(C_L)}{\sigma^2(C_L) + \sigma^2(C_R)}$$

These weights are propagated recursively, avoiding numerical instability in covariance matrix inversion and yielding a diversified portfolio [25].

2.3.4 Inverse Variance Portfolio (IVP)

The IVP weight for asset i is calculated as:

$$w_i^{IVP} = \frac{1/\sigma_i^2}{\sum_{j=1}^{n} 1/\sigma_j^2}$$

This formula is used for both variance aggregation within clusters and as a baseline strategy in empirical evaluations [44].

2.3.5 Algorithm Simulation

Algorithm 2 Hierarchical Risk Parity (HRP)

1: Compute correlation matrix ρ_{ij} and distance matrix $d_{ij} = \sqrt{0.5(1 - \rho_{ij})}$
2: Perform hierarchical clustering using Ward's linkage
3: Quasi-diagonalize the covariance matrix using the dendrogram order
4: Recursively:
- Split clusters into two subsets
- Compute inverse-variance weight of each cluster
- Allocate capital inversely proportional to cluster variance

5: Output final portfolio weights w_i

2.4 Autoencoder-Based Portfolio Optimisation

Autoencoders, a class of artificial neural networks, are powerful tools for unsupervised learning, especially in capturing nonlinear relationships in high-dimensional financial data. In portfolio optimization, autoencoders are used for dimensionality reduction, extracting latent features that influence asset returns and aid in risk-return analysis and asset allocation [6].

2.4.1 Encoding and Decoding

An autoencoder comprises an encoder and a decoder. The encoder compresses the input vector $X \in \mathbb{R}^n$ into a lower-dimensional representation $c \in \mathbb{R}^m$ (where $m < n$), while the decoder reconstructs the input from this compressed code. The transformation is represented as:

$$\phi : X \to c, \quad \varphi : c \to X'$$

where ϕ is the encoding function, φ is the decoding function, and X' is the reconstructed input [9]. The process can be summarized as:

$$f : \mathbb{R}^n \to \mathbb{R}^m, \quad g : \mathbb{R}^m \to \mathbb{R}^n$$

2.4.2 Loss Function

Autoencoders are trained by minimizing the reconstruction error between the original input and its reconstruction. For numerical inputs, the loss function is typically the Mean Squared Error (MSE):

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^{N} \|x_i - g(f(x_i))\|^2$$

Minimizing this loss forces the autoencoder to learn the most efficient representation of the input data with minimal information loss [17].

2.4.3 Structure and Representation

Autoencoders are symmetric neural network architectures designed to learn efficient, compressed representations of input data in an unsupervised manner. Structurally, they consist of two main components: an *encoder* that maps the input data into a lower-dimensional latent space, and a *decoder* that reconstructs the input from this latent representation. The number of neurons in the output layer matches that of the input layer, ensuring that the network is trained to reproduce the input as accurately as possible. The central layer, known as the *coding layer* or *bottleneck*, represents the compressed latent space of the input data, where redundant information is discarded and only the most salient features are retained.

This transformation captures the most informative patterns in the data and can reveal nonlinear correlations that traditional linear methods like PCA may miss [17]. Each hidden layer in the encoder learns hierarchical features, with deeper layers extracting increasingly abstract representations. These learned representations are valuable for downstream tasks such as anomaly detection, dimensionality reduction, portfolio clustering, or risk estimation in financial applications [9, 6]. Moreover, regularized variants such as sparse autoencoders or denoising autoencoders further enhance generalization by enforcing constraints or noise robustness during learning, respectively. Such networks have gained traction in finance due to their ability to uncover latent structures in high-dimensional datasets while preserving reconstruction fidelity.

2.4.4 Portfolio Construction

Autoencoders can be trained on historical stock price data to extract relevant features for asset returns. After training, the normalized values of the output layer represent asset weights in the portfolio:

$$w_i = \frac{x'_i}{\sum_{j=1}^{n} x'_j}$$

These weights reflect the relative importance of each asset and are used for portfolio construction, with the goal of capturing the latent structure in asset returns and allocating capital accordingly [6].

Empirical studies show that autoencoder-based portfolios can outperform traditional models in terms of annual returns, though they may also exhibit higher volatility due to the nonlinear transformations involved [6].

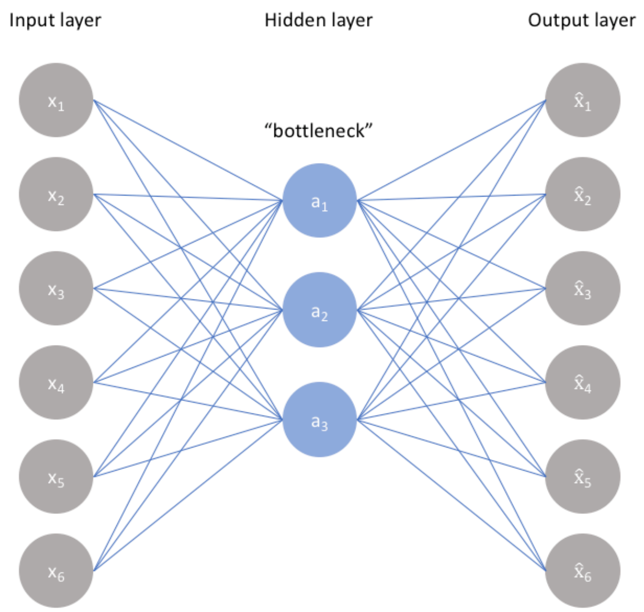

Figure 2.3: Schematic representation of an Autoencoder.

2.4.5 Algorithm Simulation

Algorithm 3 Autoencoder-Based Portfolio Optimization

1: Train autoencoder on historical stock price matrix X
2: Encode: $c = f(X)$, Decode: $X' = g(c)$
3: Minimize reconstruction loss:

$$\mathcal{L} = \|X - X'\|^2$$

4: Normalize output layer: $w_i = \frac{x'_i}{\sum_j x'_j}$
5: Output weights w_i for portfolio allocation

Chapter 3

A Prerequisite for Adaptive Regime Strategy

This chapter lays the groundwork for the adaptive regime-based portfolio construction framework discussed in the next chapter. Since the strategy involves dynamic allocation rules that respond to latent market regimes, a number of critical design choices and preparatory steps must be addressed. These include the selection of relevant financial instruments, appropriate benchmark construction, standardization of pricing data, and the integration of a robust modeling window. Together, these choices ensure both the validity and the effectiveness of the adaptive regime learning process [5, 39].

3.1 Choosing the Sectors

The analysis focuses on ten thematic sectors of the National Stock Exchange (NSE), India. These sectors represent diverse investment themes such as infrastructure, manufacturing, consumption, and sustainability [34, 31, 18]. For each sector, ten constituent stocks were selected based on their free-float market capitalisation as per NSE's report dated February 29, 2022 [35, 37].

Table 3.1: Selected NSE Thematic Sectors for Portfolio Construction

Sector No.	NSE Thematic Sector
1	NIFTY Commodities
2	NIFTY Energy
3	NIFTY Manufacturing
4	NIFTY Services
5	NIFTY MNC
6	NIFTY Transportation and Logistics
7	NIFTY Infrastructure
8	NIFTY Housing
9	NIFTY Consumption
10	NIFTY 100 ESG (Environmental, Social, and Governance)

These sectors form the basis for evaluating the performance of different portfolio optimisation methods across varying economic themes and industrial categories [49, 22]. For every sector, the top ten stocks by free-float market capitalisation (as per NSE's report dated February 29, 2022) are selected to form the portfolio [34, 7].

3.2 Acquiring the Data

To construct and evaluate the portfolios, historical stock price data is collected using the `DataReader` function from the `pandas-datareader` Python library [30, 24]. The closing price data for each of the 100 selected stocks (10 per sector) is extracted from Yahoo Finance [20].

- **Training period:** January 1, 2018 – December 31, 2021 [47]
- **Testing period:** January 1, 2022 – December 31, 2022 [50]

Only the daily closing prices are used in the analysis, as this is a univariate portfolio construction setup [40]. From the raw prices, daily returns are computed and used to assess risk and return characteristics [12].

3.3 Designing the MVP Portfolios

The Mean-Variance Portfolio (MVP) is constructed using the classical Markowitz framework [28], which aims to maximise return for a given level of risk or minimise risk for a given return. The MVP for each sector is designed using the Monte Carlo simulation approach, implemented in Python [48].

Monte Carlo Weights Simulation

In the Monte Carlo simulation approach, Python is used to perform the following steps:

- **Annualised Mean Returns:** The mean returns for each stock in the portfolio are calculated, annualised by multiplying by 250 (the number of trading days in a year).
- **Annualised Covariance Matrix:** The covariance matrix of returns is computed and annualised by multiplying by 250.
- **Random Portfolio Generation:** 10,000 random portfolios are generated using the Dirichlet distribution [27]. This ensures that the portfolio weights sum to 1.
- **Portfolio Return, Volatility, and Sharpe Ratio:** For each generated portfolio, the annualised return, volatility, and Sharpe ratio are computed.
- **Optimal Portfolio Selection:** The portfolio with the highest Sharpe ratio is selected as the optimal MVP portfolio.

Algorithm: Mean-Variance Portfolio Optimisation (Monte Carlo Simulation)

The algorithm for constructing the MVP portfolio using Monte Carlo simulation is as follows:

- Compute the annualised return vector and covariance matrix using historical stock data.

- Generate random portfolios and compute their returns, volatilities, and Sharpe ratios.
- Select the portfolio with the highest Sharpe ratio as the optimal portfolio.

This method provides a robust, visual representation of the efficient frontier and selects the optimal portfolio based on the maximum Sharpe ratio. The implementation of this method in Python [48] allows for easy experimentation with different asset sets and portfolio sizes, providing flexibility in designing portfolios.

3.4 Designing the HRP Portfolios

The Hierarchical Risk Parity (HRP) portfolio is constructed using the clustering-based framework proposed by Marcos López de Prado [26]. It avoids matrix inversion, ensuring stability when data is noisy or highly correlated. The construction involves three main steps:

Step 1: Agglomerative Clustering via Ward Linkage

- Compute the correlation matrix ρ using:

$$\rho = \texttt{train\_returns.corr()}$$

- Clip values to the range $[-1, 1]$ and fill missing values with zeros to avoid instability [45].
- Calculate the distance matrix d_{ij} as:

$$d_{ij} = \sqrt{0.5(1 - \rho_{ij})}$$

- Apply the Ward linkage method using the `scipy.cluster.hierarchy.linkage` function to create a dendrogram [26].

Step 2: Quasi-Diagonalization

The Ward linkage tree is passed to the HRP optimiser from the `PyPortfolioOpt` package [21]. This reorders assets based on the dendrogram, producing a quasi-diagonal covariance matrix for intuitive risk decomposition.

Step 3: Recursive Bisection

HRP recursively splits the portfolio into clusters and allocates weights inversely proportional to the cluster variances, avoiding quadratic optimisation and ensuring a balanced risk allocation [26].

Implementation Summary:

- Use `HRPOpt(train_ret)` to construct the optimiser.
- Assign the Ward linkage matrix to `hrp._linkage`.
- Compute portfolio weights with `hrp.optimize()`.

This method provides an interpretable, robust, and diversified portfolio, even when correlations are unstable or the number of assets is large [26].

Algorithm: HRP Portfolio Construction

Algorithm 4 Hierarchical Risk Parity Portfolio Construction

1: Compute correlation matrix $\rho =$ `train_ret.corr()`
2: Clip $\rho \in [-1, 1]$, fill missing values with 0
3: Calculate distance matrix: $d_{ij} = \sqrt{0.5(1 - \rho_{ij})}$
4: Perform Ward linkage: $Z =$ `linkage`$(\cdot,$ `"ward"`$)$
5: Initialise HRP optimiser: `HRPOpt(train_ret)`
6: Assign linkage: `hrp._linkage = Z`
7: Compute weights: `w_hrp = hrp.optimize()`

3.5 Designing the Autoencoder Portfolios

Autoencoder-based portfolio optimization uses unsupervised neural networks to learn compressed representations of stock price behaviours. These learned features are then used to derive the relative importance of each asset, aiding in portfolio construction. By leveraging an autoencoder's ability to capture complex, non-linear patterns in the data, this method enables the extraction of latent factors that can improve the robustness of portfolio allocation [6, 17].

Preprocessing and Scaling

Prior to training the autoencoder, the stock price data is scaled using Min-Max normalisation. This maps the values of each stock's data into the range $[0, 1]$, ensuring that each feature contributes uniformly during training. This scaling step is critical for ensuring that no single asset disproportionately affects the learning process [41].

Table 3.2: Preprocessing Step

Operation	**Details**
Scaler Used	MinMaxScaler (column-wise)
Input Shape	$T \times 10$ (T = time steps, 10 stocks)
Transformed Range	[0, 1] for each asset column

Autoencoder Model Architecture

The architecture used in this portfolio construction process is a shallow feedforward autoencoder with a symmetric structure. The encoder compresses the 10 stock features into a smaller number of latent features. The decoder then attempts to reconstruct the original stock data from these compressed features, aiming to minimize the reconstruction error. The compressed representation allows the model to capture the most informative aspects of the stock price movements [17].

Table 3.3: Autoencoder Configuration

Layer Type	**Configuration**
Input Layer	10 neurons (1 per stock)
Hidden Layer	5 neurons, ReLU activation (coding layer)
Output Layer	10 neurons, Linear activation

Table 3.4: Training Setup

Parameter	Value
Optimiser	Adam
Loss Function	Mean Squared Error (MSE)
Epochs	1000
Batch Size	10
Framework	Keras Sequential API

Portfolio Weight Extraction

Once the autoencoder is trained, the final (latest) observation of the scaled dataset is passed through the network to obtain the reconstructed output. The values of the reconstructed output are interpreted as the relative importance scores for each asset. These scores are then normalised to generate the final portfolio weights. The following steps are performed:

- Extract the output vector from the trained model using the last observation.
- Take the absolute values of the output to ensure non-negative weights.
- Normalize the output values so they sum to 1, yielding the final portfolio weights.

This process ensures that the portfolio allocation reflects the most significant latent features learned by the autoencoder [6].

Algorithm: Autoencoder-Based Portfolio Construction

The following steps outline the autoencoder-based portfolio construction approach:

Algorithm 5 Autoencoder-Based Portfolio Construction

1: Input: Price matrix $P \in \mathbb{R}^{T \times N}$
2: Scale data column-wise using MinMaxScaler to get X
3: Define autoencoder with layers:

$$\texttt{Dense(5, relu)} \rightarrow \texttt{Dense(10, linear)}$$

4: Compile with Adam optimiser and MSE loss
5: Fit the model: `model.fit(X, X, epochs=1, batch_size=10)`
6: Pass last observation x_T through the network
7: Extract reconstructed output $\hat{x}_T$
8: Take absolute values and normalise:

$$w_i = \frac{|\hat{x}_{Ti}|}{\sum_j |\hat{x}_{Tj}|}$$

9: Output: Portfolio weights $\mathbf{w}_{\text{AE}}$

3.6 Evaluating Portfolio Performance

The three portfolio strategies (MVP, HRP, and Autoencoder) are evaluated across all ten sectors using the following performance metrics:

1. **Annual Return (AR):** Measures compounded growth over a year:

$$\text{AR} = \left(\frac{V_{\text{end}}}{V_{\text{start}}}\right)^{\frac{252}{T}} - 1$$

2. **Annual Volatility (AV):** Standard deviation of daily returns scaled to an annual level:

$$\text{AV} = \text{std}(r) \times \sqrt{252}$$

3. **Sharpe Ratio (SR):** Risk-adjusted return assuming a risk-free rate of zero:

$$\text{SR} = \frac{\text{Annual Return}}{\text{Annual Volatility}}$$

Both training (2018–2021) and testing (2022) periods are evaluated for each sector and strategy. The portfolio with the highest Sharpe ratio on the test data is considered the best-performing strategy for that sector from an investor's standpoint.

3.7 Sector: NIFTY Commodities

Overview

The NIFTY Commodities index represents companies operating in the commodities space, including energy, metals, cement, and natural resources.

Table 3.5: NIFTY Commodities Sector – Portfolio Weights

Stock	MVP	HRP	ENC
RELIANCE	0.2144	0.1464	0.0807
ULTRACEMCO	0.2140	0.1524	0.1079
TATASTEEL	0.0064	0.0428	0.1038
NTPC	0.3286	0.1803	0.1037
JSWSTEEL	0.0071	0.0459	0.1381
ONGC	0.0044	0.0614	0.0780
GRASIM	0.0234	0.1127	0.0997
HINDALCO	0.0025	0.0667	0.1194
COALINDIA	0.1408	0.0889	0.0590
UPL	0.0584	0.1025	0.1098

Performance Analysis

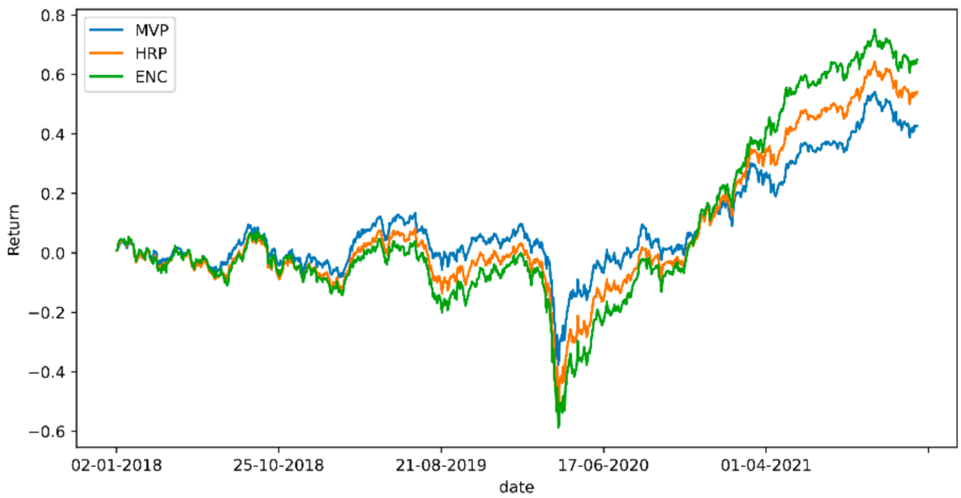

Figure 3.1: Cumulative Daily Returns (Training Period: 2018–2021)

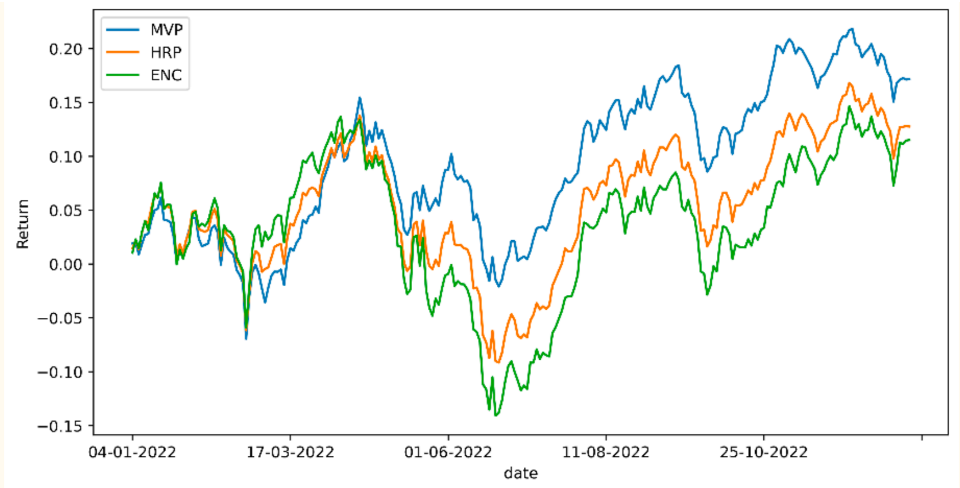

Figure 3.2: Cumulative Daily Returns (Test Period: 2022)

Table 3.6: Training and Test Performance of Portfolios

Portfolio	Training Performance			Test Performance		
	Annual Return	Annual Volatility	Sharpe Ratio	Annual Return	Annual Volatility	Sharpe Ratio
MVP	10.89	21.87	0.4978	17.51	19.49	0.8982
HRP	13.82	23.49	0.5883	13.01	20.48	0.6354
ENC	16.59	26.01	0.6375	11.74	23.32	0.5034

3.8 Sector: NIFTY Energy

Overview

The NIFTY Energy sector captures the performance of Indian companies involved in the energy sector, including oil, gas, electricity, and renewable power.

Table 3.7: NIFTY Energy Sector – Portfolio Weights

Stock	MVP	HRP	ENC
RELIANCE	0.2090	0.1341	0.1221
NTPC	0.2069	0.1683	0.0903
POWERGRID	0.3170	0.2184	0.1324
ONGC	0.0065	0.0508	0.0863
TATAPOWER	0.0170	0.0827	0.0902
BPCL	0.0106	0.0462	0.1159
IOC	0.0643	0.1176	0.0983
GAIL	0.0571	0.0890	0.0624
ADANITRANS	0.0471	0.0483	0.0624
ADANIGREEN	0.0471	0.0483	0.0843

Performance Analysis

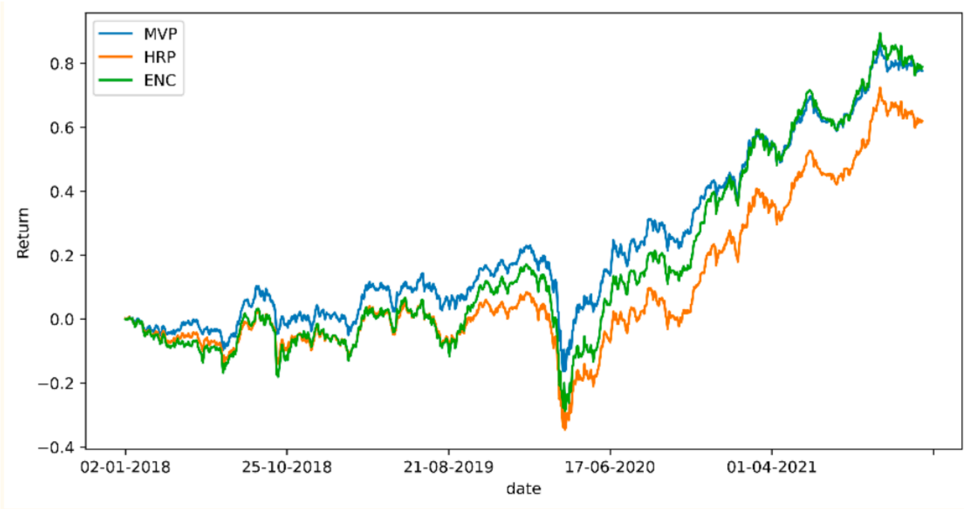

Figure 3.3: Cumulative Daily Returns (Training Period: 2018–2021)

Figure 3.4: Cumulative Daily Returns (Test Period: 2022)

Table 3.8: Training and Test Performance of Portfolios – NIFTY Energy Sector

Portfolio	Training Performance			Test Performance		
	Annual Return	**Annual Volatility**	**Sharpe Ratio**	**Annual Return**	**Annual Volatility**	**Sharpe Ratio**
MVP	19.83	20.98	0.9451	17.83	19.62	0.9086
HRP	15.81	21.92	0.7212	14.82	19.54	0.7583
ENC	20.15	23.31	0.8644	14.17	20.15	0.7033

3.9 Sector: NIFTY Manufacturing

Overview

The NIFTY Manufacturing index represents companies with significant manufacturing operations across sectors like pharmaceuticals, automobiles, metals, and industrials.

Table 3.9: NIFTY Manufacturing Sector – Portfolio Weights

Stock	MVP	HRP	ENC
SUNPHARMA	0.0600	0.1257	0.1422
RELIANCE	0.1790	0.1077	0.0885
M&M	0.0738	0.0741	0.1007
TATASTEEL	0.0182	0.0690	0.0981
MARUTI	0.1437	0.0838	0.0651
JSWSTEEL	0.0130	0.0742	0.1299
HINDALCO	0.0071	0.0627	0.1044
TATAMOTORS	0.0069	0.0525	0.1167
DRREDDY	0.2629	0.1661	0.0813
CIPLA	0.2354	0.1840	0.0732

Performance Analysis

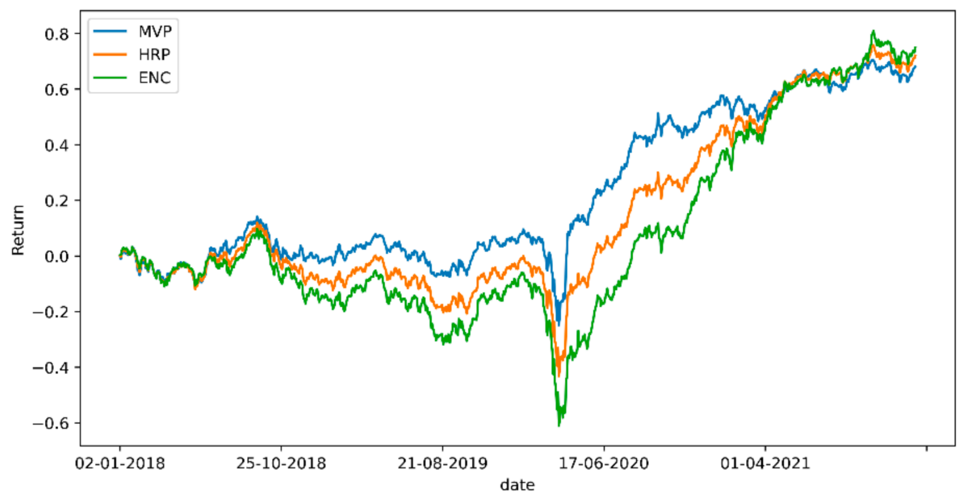

Figure 3.5: Cumulative Daily Returns (Training Period: 2018–2021)

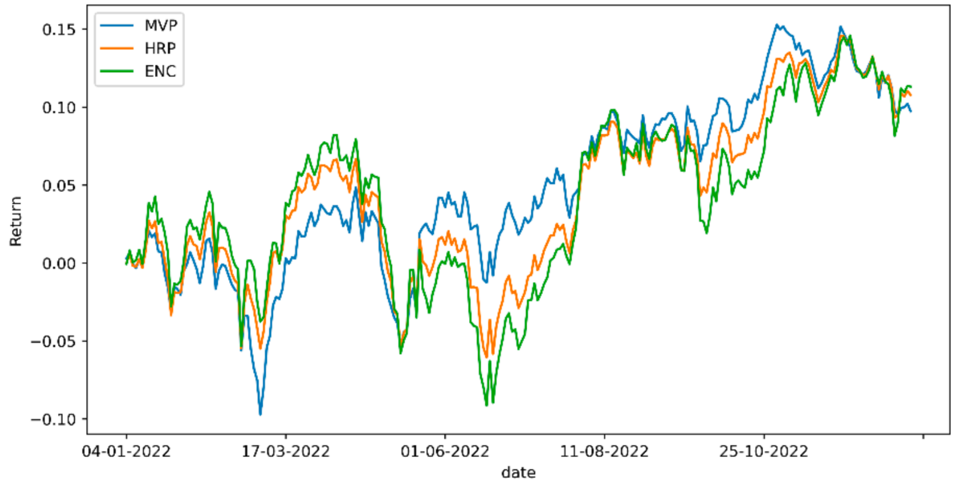

Figure 3.6: Cumulative Daily Returns (Test Period: 2022)

Table 3.10: Training and Test Performance of Portfolios – NIFTY Manufacturing Sector

Portfolio	Training Performance			Test Performance		
	Annual Return	Annual Volatility	Sharpe Ratio	Annual Return	Annual Volatility	Sharpe Ratio
MVP	17.40	20.76	0.8382	9.93	16.27	0.6102
HRP	18.40	22.28	0.8259	10.99	17.61	0.6242
ENC	19.12	25.58	0.7474	11.54	20.79	0.5549

3.10 Sector: NIFTY Services

Overview

The NIFTY Services index includes major companies offering financial, IT, telecom, and consulting services in India.

Table 3.11: NIFTY Services Sector – Portfolio Weights

Stock	MVP	HRP	ENC
HDFCBANK	0.2593	0.0873	0.0788
ICICIBANK	0.0020	0.0727	0.0924
INFY	0.1445	0.1622	0.1194
HDFC	0.0030	0.0605	0.1194
TCS	0.3253	0.1933	0.0963
KOTAKBANK	0.0832	0.1096	0.1598
AXISBANK	0.0013	0.0627	0.0901
SBIN	0.0349	0.0725	0.0901
BHARTIARTL	0.1453	0.1225	0.0499
BAJFINANCE	0.0011	0.0566	0.0974

Performance Analysis

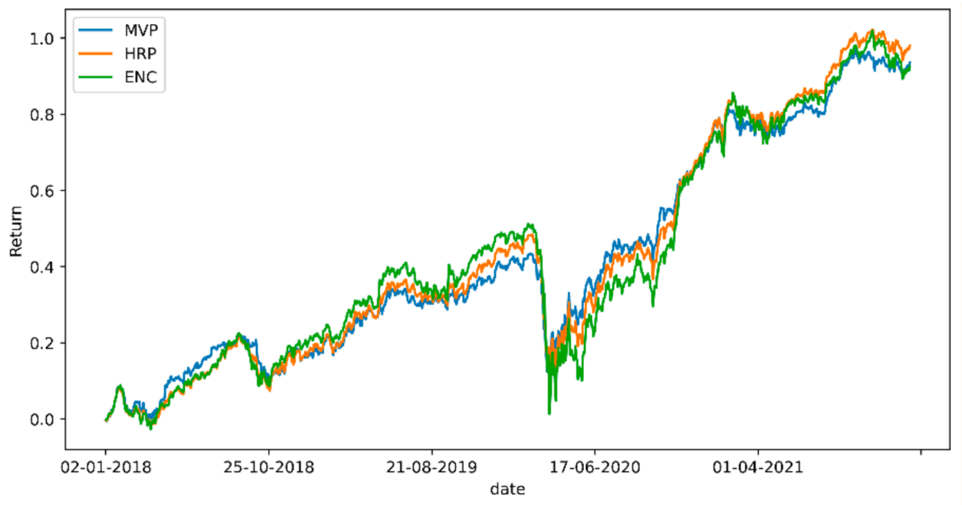

Figure 3.7: Cumulative Daily Returns (Training Period: 2018–2021)

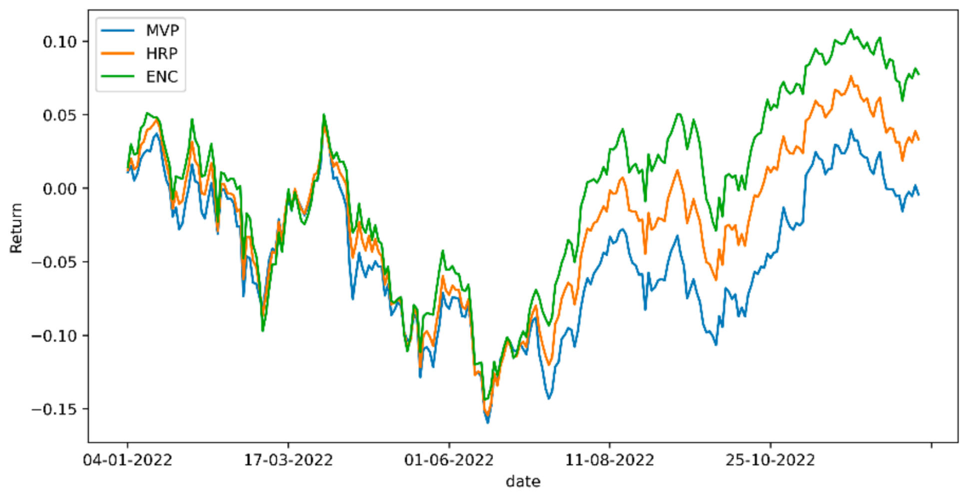

Figure 3.8: Cumulative Daily Returns (Test Period: 2022)

Table 3.12: Training and Test Performance of Portfolios – NIFTY Services Sector

Portfolio	Training Performance			Test Performance		
	Annual Return	Annual Volatility	Sharpe Ratio	Annual Return	Annual Volatility	Sharpe Ratio
MVP	23.89	19.81	1.2059	-0.45	18.68	-0.0240
HRP	25.03	21.42	1.1684	3.40	18.53	0.1834
ENC	23.65	24.31	0.9727	7.92	19.57	0.4047

3.11 Sector: NIFTY MNC

Overview

The NIFTY MNC sector comprises multinational companies operating in India across diverse industries, such as FMCG, automobiles, industrials, and beverages.

Table 3.13: NIFTY MNC Sector – Portfolio Weights

Stock	MVP	HRP	ENC
MARUTI	0.0192	0.0836	0.0726
HINDUNILVR	0.2535	0.1395	0.0680
NESTLEIND	0.2054	0.1854	0.0715
BRITANNIA	0.1174	0.1219	0.0723
VEDL	0.0122	0.0605	0.1879
SIEMENS	0.0874	0.0678	0.1375
AMBUJACEM	0.0688	0.0629	0.0855
MCDOWELL-N	0.0769	0.1158	0.0879
CUMMINSIND	0.1541	0.1114	0.1063
ASHOKLEY	0.0050	0.0512	0.1105

Performance Analysis

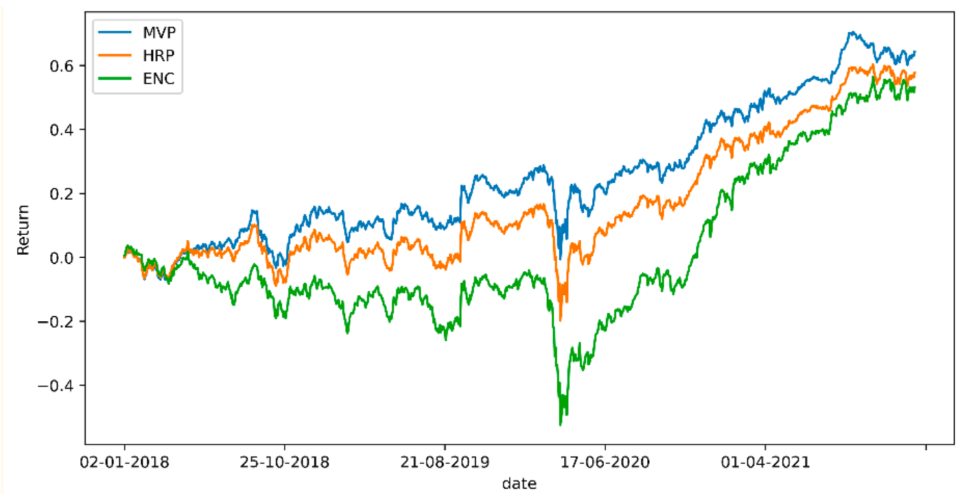

Figure 3.9: Cumulative Daily Returns (Training Period: 2018–2021)

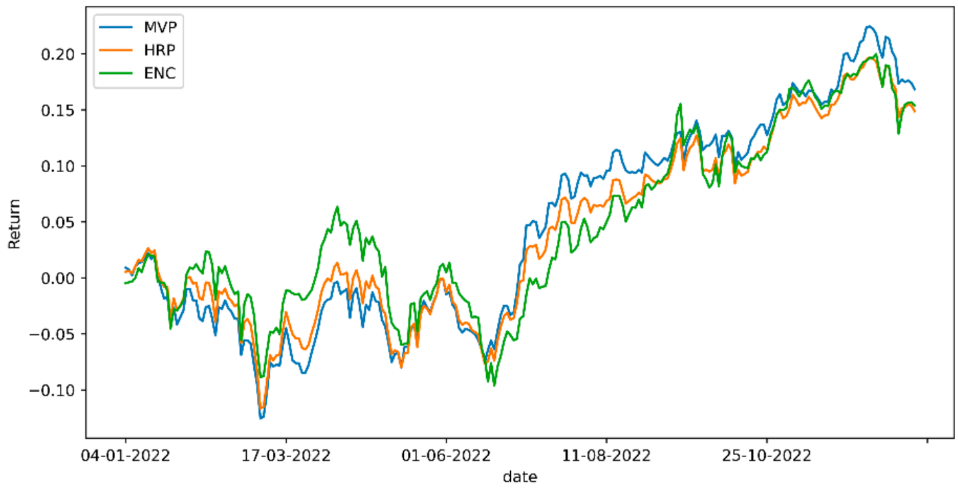

Figure 3.10: Cumulative Daily Returns (Test Period: 2022)

Table 3.14: Training and Test Performance of Portfolios – NIFTY MNC Sector

Portfolio	Training Performance			Test Performance		
	Annual Return	Annual Volatility	Sharpe Ratio	Annual Return	Annual Volatility	Sharpe Ratio
MVP	16.42	18.59	0.8833	17.18	17.24	0.9963
HRP	14.73	19.49	0.7557	15.18	17.33	0.8754
ENC	13.55	22.88	0.5922	15.70	20.04	0.7835

3.12 Sector: NIFTY Transportation and Logistics

Overview

The NIFTY Transportation and Logistics index tracks companies involved in automobile manufacturing, logistics, ports, and freight services.

Table 3.15: NIFTY Transportation and Logistics Sector – Portfolio Weights

Stock	MVP	HRP	ENC
M&M	0.0742	0.1207	0.0988
MARUTI	0.0425	0.1139	0.1028
TATAMOTORS	0.0025	0.0315	0.1081
ADANIPORTS	0.1398	0.1340	0.1020
EICHERMOT	0.0839	0.0800	0.1071
BAJAJ-AUTO	0.3344	0.1600	0.1235
HEROMOTOCO	0.0847	0.1237	0.1136
TIINDIA	0.2034	0.1215	0.0377
TVSMOTOR	0.0305	0.0777	0.1060
ASHOKLEY	0.0041	0.0370	0.1003

Performance Analysis

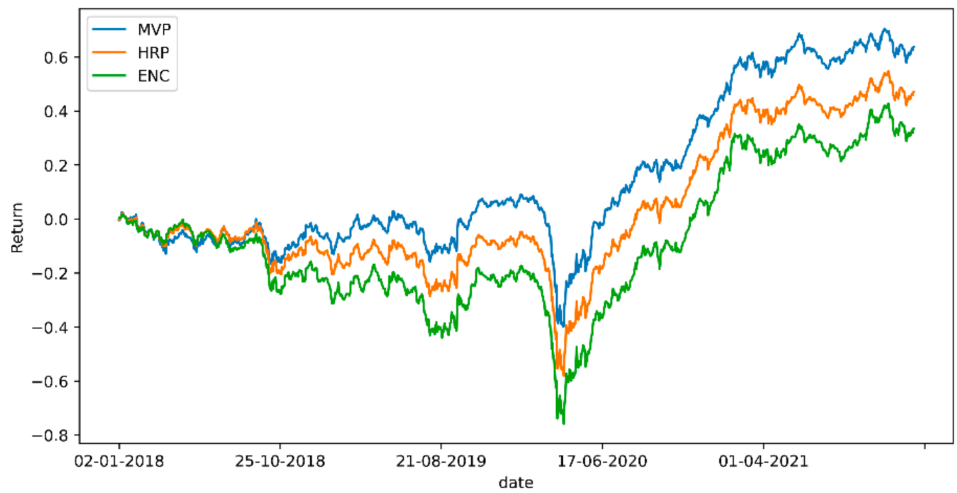

Figure 3.11: Cumulative Daily Returns (Training Period: 2018–2021)

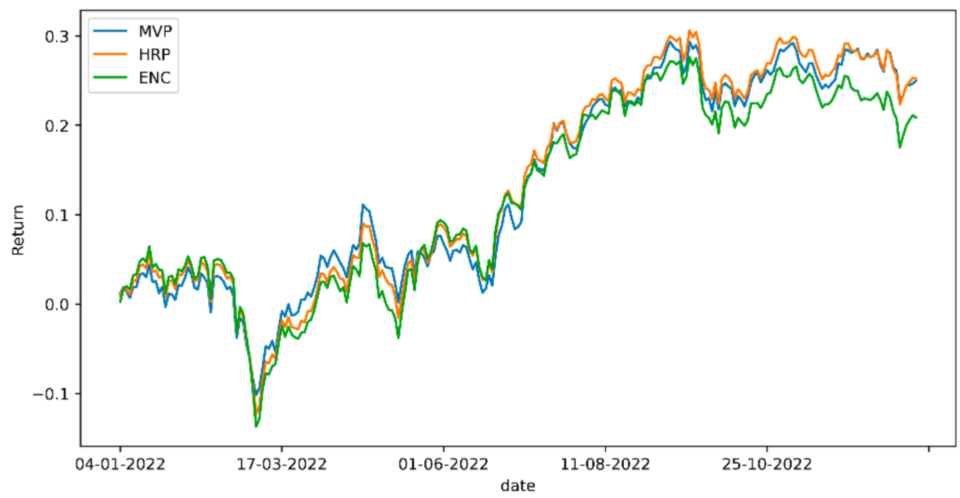

Figure 3.12: Cumulative Daily Returns (Test Period: 2022)

Table 3.16: Training and Test Performance of Portfolios – NIFTY Transportation and Logistics Sector

Portfolio	Training Performance			Test Performance		
	Annual Return	**Annual Volatility**	**Sharpe Ratio**	**Annual Return**	**Annual Volatility**	**Sharpe Ratio**
MVP	16.26	22.75	0.7150	25.49	21.10	1.2081
HRP	12.02	23.79	0.5053	25.73	21.51	1.1962
ENC	8.55	26.03	0.3286	21.30	22.62	0.9413

3.13 Sector: NIFTY Infrastructure

Overview

The NIFTY Infrastructure index tracks companies involved in industrial infrastructure, utilities, construction, and related sectors.

Table 3.17: NIFTY Infrastructure Sector – Portfolio Weights

Stock	MVP	HRP	ENC
RELIANCE	0.1152	0.1282	0.1136
LT	0.0876	0.1062	0.1274
BHARTIARTL	0.0755	0.1137	0.1088
ULTRACEMCO	0.0591	0.0607	0.0953
NTPC	0.1771	0.1258	0.1169
POWERGRID	0.3160	0.1480	0.1284
ONGC	0.0046	0.0434	0.0837
GRASIM	0.0046	0.0434	0.0837
APOLLOHOSP	0.1164	0.1092	0.0467
ADANIPORTS	0.0405	0.0898	0.0940

Performance Analysis

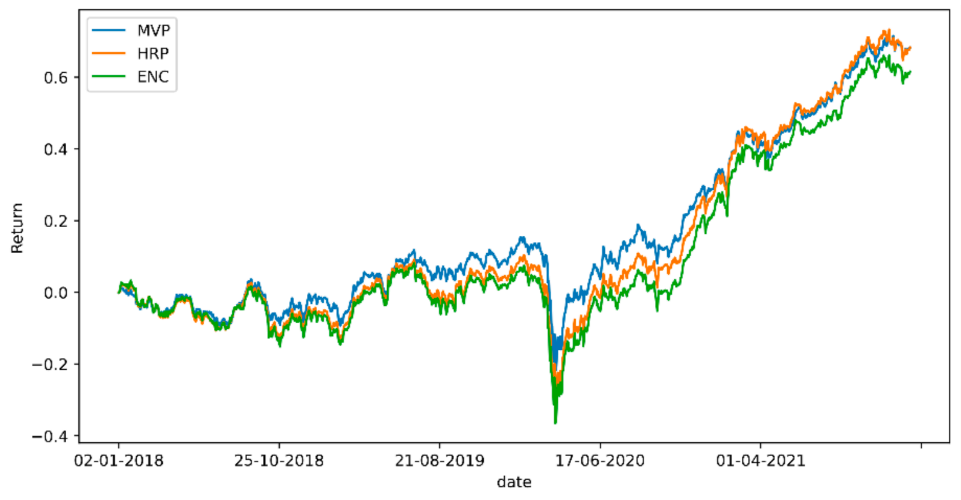

Figure 3.13: Cumulative Daily Returns (Training Period: 2018–2021)

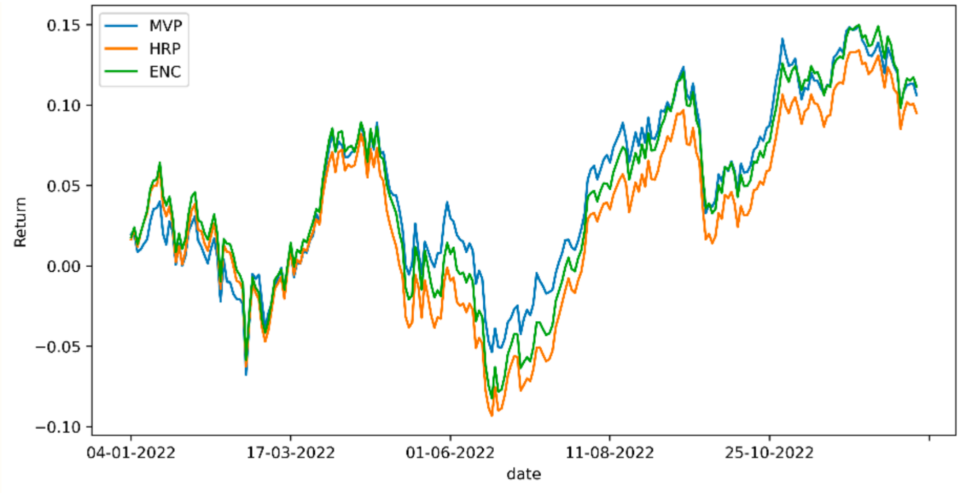

Figure 3.14: Cumulative Daily Returns (Test Period: 2022)

Table 3.18: Training and Test Performance of Portfolios – NIFTY Infrastructure Sector

Portfolio	Training Performance			Test Performance		
	Annual Return	Annual Volatility	Sharpe Ratio	Annual Return	Annual Volatility	Sharpe Ratio
MVP	17.41	19.78	0.8801	10.85	17.51	0.6200
HRP	17.42	20.99	0.8300	9.70	17.57	0.5523
ENC	15.70	21.11	0.7438	11.38	17.83	0.6384

3.14 Sector: NIFTY Housing

Overview

The NIFTY Housing index tracks companies contributing to the housing and infrastructure ecosystem, including construction, paints, steel, and financial services.

Table 3.19: NIFTY Housing Sector – Portfolio Weights

Stock	MVP	HRP	ENC
LT	0.0691	0.1287	0.1181
ASIANPAINT	0.3099	0.1883	0.1149
HDFCBANK	0.2584	0.0882	0.0884
ICICIBANK	0.0005	0.0786	0.0829
ULTRACEMCO	0.0594	0.0785	0.0818
TATASTEEL	0.0035	0.0647	0.0709
NTPC	0.2951	0.1754	0.1391
HDFC	0.0009	0.0612	0.1218
JSWSTEEL	0.0010	0.0465	0.0906
GRASIM	0.0023	0.0899	0.0915

Performance Analysis

Figure 3.15: Cumulative Daily Returns (Training Period: 2018–2021)

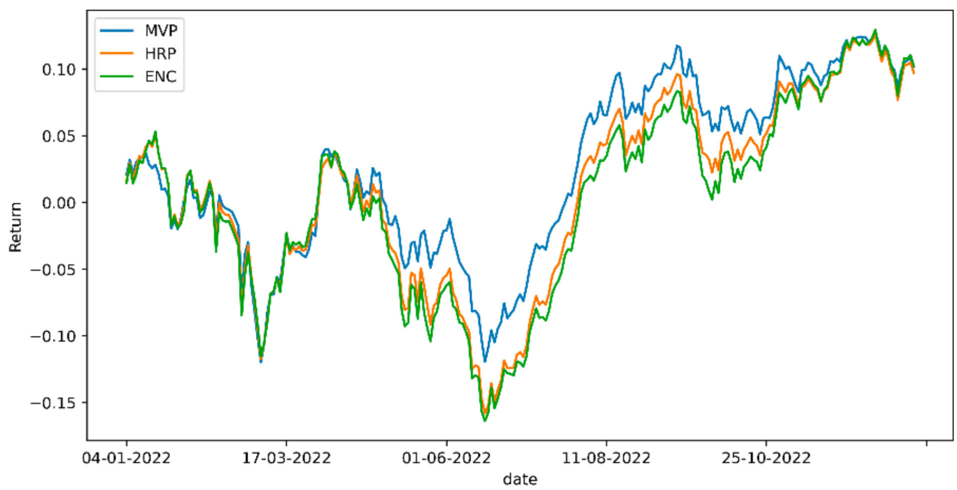

Figure 3.16: Cumulative Daily Returns (Test Period: 2022)

Table 3.20: Training and Test Performance of Portfolios – NIFTY Housing Sector

Portfolio	Training Performance			Test Performance		
	Annual Return	**Annual Volatility**	**Sharpe Ratio**	**Annual Return**	**Annual Volatility**	**Sharpe Ratio**
MVP	16.03	19.85	0.8079	10.38	17.61	0.5897
HRP	18.09	21.74	0.8324	9.92	18.59	0.5337
ENC	18.33	22.83	0.8030	10.43	19.28	0.5411

3.15 Sector: NIFTY Consumption

Overview

The NIFTY Consumption index includes companies focused on consumer goods and services, ranging from FMCG to automotive and retail.

Table 3.21: NIFTY Consumption Sector – Portfolio Weights

Stock	MVP	HRP	ENC
ITC	0.2412	0.1598	0.1232
HINDUNILVR	0.1641	0.1232	0.1021
BHARTIARTL	0.0714	0.0901	0.0958
ASIANPAINT	0.1027	0.1147	0.1286
M&M	0.0299	0.0412	0.0973
MARUTI	0.0058	0.0446	0.0503
TITAN	0.0477	0.0888	0.1374
NESTLEIND	0.1665	0.1093	0.0904
BRITANNIA	0.0673	0.1223	0.0870
DMART	0.1035	0.1061	0.0878

Performance Analysis

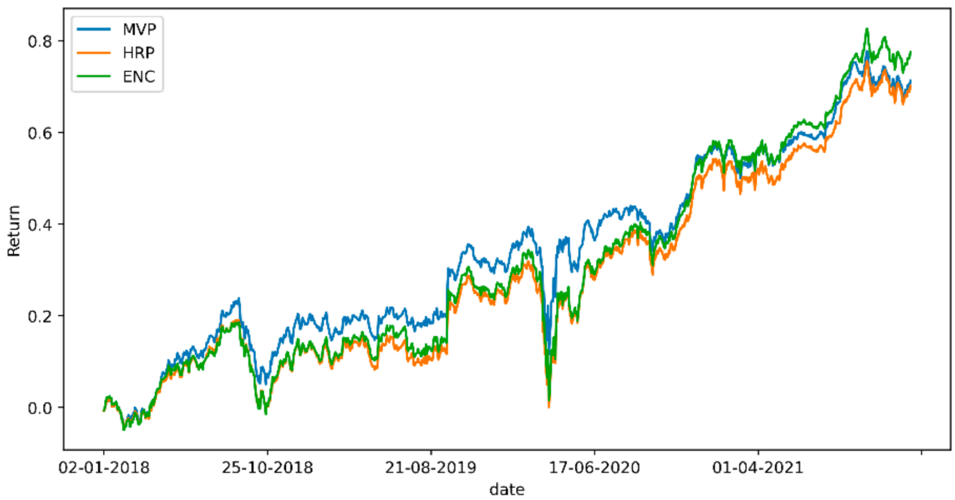

Figure 3.17: Cumulative Daily Returns (Training Period: 2018–2021)

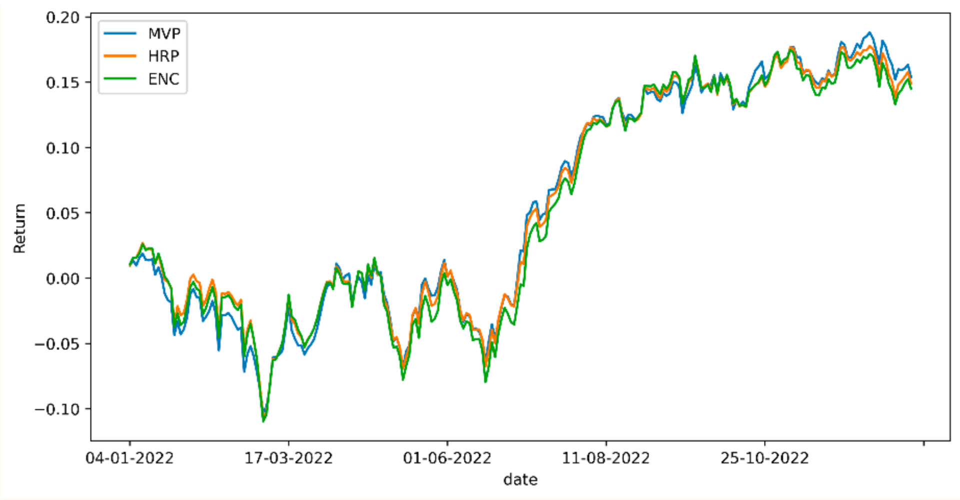

Figure 3.18: Cumulative Daily Returns (Test Period: 2022)

Table 3.22: Training and Test Performance of Portfolios – NIFTY Consumption Sector

Portfolio	Training Performance			Test Performance		
	Annual Return	Annual Volatility	Sharpe Ratio	Annual Return	Annual Volatility	Sharpe Ratio
MVP	18.19	17.88	1.0178	15.68	15.57	1.0074
HRP	17.91	18.50	0.9680	15.18	16.13	0.9416
ENC	19.81	18.74	1.0568	14.79	16.62	0.8896

3.16 Sector: NIFTY 100 ESG

Overview

The NIFTY 100 ESG index includes companies with high Environmental, Social, and Governance (ESG) scores within the NIFTY 100 universe.

Table 3.23: NIFTY 100 ESG Sector – Portfolio Weights

Stock	**MVP**	**HRP**	**ENC**
INFY	0.0895	0.1328	0.0750
TCS	0.2571	0.1606	0.0810
HDFC	0.0440	0.0644	0.1180
HCLTECH	0.0952	0.1067	0.0862
ICICIBANK	0.0147	0.0485	0.1208
BHARTIARTL	0.1412	0.0737	0.1209
TECHM	0.0562	0.1131	0.1029
KOTAKBANK	0.1425	0.1229	0.1175
BAJFINANCE	0.0044	0.0602	0.0607
TITAN	0.1552	0.1170	0.1169

Performance Analysis

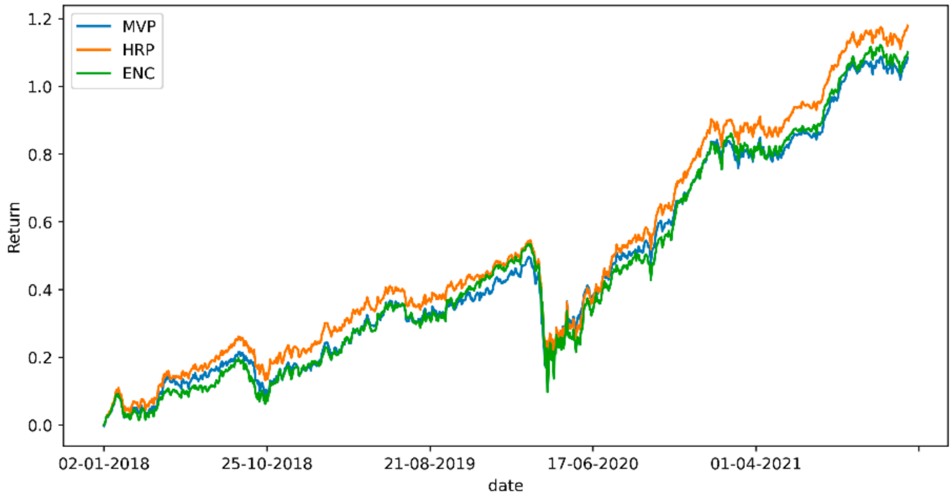

Figure 3.19: Cumulative Daily Returns (Training Period: 2018–2021)

Figure 3.20: Cumulative Daily Returns (Test Period: 2022)

Table 3.24: Training and Test Performance of Portfolios – NIFTY 100 ESG Sector

Portfolio	Training Performance			Test Performance		
	Annual Return	Annual Volatility	Sharpe Ratio	Annual Return	Annual Volatility	Sharpe Ratio
MVP	27.73	19.68	1.4088	-5.66	18.51	-0.3058
HRP	30.12	20.30	1.4833	-9.56	19.44	-0.4921
ENC	28.12	21.12	1.3292	-4.00	18.84	-0.2124

3.17 Summary of Sector-wise Portfolio Performance

This section summarises the comparative performance of the three portfolio strategies—Mean-Variance Portfolio (MVP), Hierarchical Risk Parity (HRP), and Autoencoder-based

Portfolio (ENC)—across ten NSE thematic sectors over both training and test periods.

Training Data Insights

- The **ENC portfolio** achieved the **highest annual returns** in **5 out of 10 sectors**, showing its capability to capture return-enhancing features effectively.
- The **MVP portfolio** yielded the **highest Sharpe ratio** in **6 sectors**, demonstrating superior risk-adjusted returns.
- Most notably, the **MVP portfolio** had the **lowest annual volatility across all ten sectors**, highlighting its efficiency in risk control.

Conclusion: On training data, the ENC portfolio is ideal for investors focused on maximizing returns, while the MVP portfolio is best for those targeting risk-adjusted performance and lower volatility.

Test Data Insights

- The **ENC portfolio** again produced the **highest annual return** in **5 sectors**, validating its consistency in out-of-sample performance.
- The **MVP portfolio** recorded the **highest Sharpe ratio** in **6 sectors** and **lowest volatility** in **8 out of 10 sectors**.

Conclusion: For robust risk-adjusted returns and lower volatility under real market conditions, MVP is more stable. However, ENC is preferred when maximum return is the primary objective.

Overall, this comprehensive sectoral analysis supports a dual insight: while ENC captures hidden nonlinear return-driving features via deep learning, MVP continues to excel in producing balanced, low-volatility portfolios—making both valuable in different investment scenarios.

Chapter 4

Adaptive Regime Portfolio Strategy

4.1 Motivation

Traditional portfolio optimization methods, such as the Mean-Variance Portfolio (MVP) model [28], assume that asset returns and their correlations are constant over time. While this assumption is useful in a static market environment, it does not capture the complexities and dynamics of real-world financial markets. In practice, financial markets go through various regimes—*bullish*, *bearish*, or *neutral*—where volatility, correlations, and other market conditions fluctuate significantly [2, 39]. These changing market conditions can lead to significant mispricing when traditional methods are applied, as the assumptions of constant returns and stable correlations are violated. As a result, portfolios optimized using static methods often underperform, especially during periods of market transition or turbulence.

For example, in a bullish market regime, the correlation between asset classes may be relatively low, encouraging the allocation of more assets into riskier, high-return instruments. However, when the market transitions into a bearish regime, the correlations between assets may increase, and volatility can surge [2]. A static portfolio that was optimized in a different market regime may fail to adapt to these changes, leading to higher risks and losses. This is a fundamental limitation of traditional portfolio optimization methods, which do not account for shifts in the market environment [39].

Our strategy aims to address this limitation by incorporating regime-switching models, specifically Hidden Markov Models (HMMs), which can capture the changing dynamics of the financial markets. HMMs are a powerful tool for modeling time series data with unobservable states, where each state represents a distinct market regime [14]. By using HMMs, we can identify when the market is in a bullish, bearish, or neutral state and adjust the portfolio allocation accordingly. This allows for regime-specific optimization, where asset weights are tailored to the characteristics of the current market environment.

In addition to HMMs, regime-specific optimization methods further enhance portfolio performance by adjusting for volatility, correlation, and return dynamics that vary across regimes [25]. This adaptive approach offers a significant improvement over traditional static methods, providing more robust portfolio allocations that are better suited to the changing conditions of financial markets.

The use of HMMs and regime-specific optimization represents a paradigm shift in portfolio management. It allows investors to dynamically adjust their portfolios based on the current state of the market, reducing the risk of large losses during periods of market stress and improving overall portfolio performance across different market cycles.

4.2 Data Acquisition and Preprocessing

Price Download and Caching: The first step in the data acquisition process involves defining ticker symbols for both the assets under analysis and a benchmark index, such as the Nifty 50 (`NSEI`), which serves as a representative market index. Ticker symbols are critical as they enable the retrieval of historical price data for specific assets across various time periods.

To retrieve the historical price data, we rely on the `yfinance` API, a robust tool widely used in quantitative finance for accessing historical price data. This API provides adjusted closing prices, which account for corporate actions such as dividends, stock splits, and mergers, ensuring that the data used for analysis is reflective of true market performance. Adjusted prices are essential for performing reliable backtests, as they account for events that could distort the raw price series [32].

In the event that `yfinance` is temporarily unavailable due to API limitations, such as rate-limiting or downtime, a backoff strategy is implemented to ensure data reliability. If data cannot be retrieved from `yfinance`, the system automatically falls back to the Stooq API. Stooq is another reliable source of historical financial data, which provides a wide array of global market data, ensuring that missing data is efficiently sourced. Additionally, when both `yfinance` and Stooq fail to provide the required data, a synthetic market index is generated. This synthetic index is constructed by taking the average of various relevant indices and adjusting the weights to match the intended benchmark. This synthetic data acts as a placeholder for the actual market data, ensuring that the analysis can proceed without significant interruptions.

To optimize performance and minimize repeated API calls, the retrieved data is cached locally in a `data_cache.pkl` file. Caching is an essential optimization strategy that avoids redundant calls to external APIs, saving both time and computational resources. This also ensures that the data can be reused across multiple runs of the analysis, making the process more efficient and reliable. The caching system not only reduces external dependencies but also minimizes the risk of hitting API rate limits, which could delay data retrieval during intensive data processing tasks [32].

Alignment and Cleaning: After successfully retrieving the historical prices, the next critical step is aligning the data for the assets and benchmark index. The retrieved price data is concatenated into a single DataFrame. Each column represents the price time series for a different asset, and the index is the corresponding timestamp (typically date). The alignment process involves ensuring that the time series for each asset share the same date range and frequency. Missing timestamps, common in financial datasets, are handled by either forward filling or interpolating the data, depending on the nature of the data and the modeling requirements. This step is essential for ensuring that the time-series data is synchronized, as discrepancies in time intervals can introduce significant errors in modeling, particularly when performing regime-switching analysis, which requires exact temporal alignment between the different price series [5].

The reindexing process ensures that if an asset or the benchmark index has missing data on a particular date, the system does not mistakenly use data from a different date or fail to recognize the missing values. In cases where data is missing, various techniques are applied to handle it. For example, missing values are imputed using forward filling, where the last available data point is carried forward, or interpolation, where missing values are estimated based on surrounding data points. These techniques are carefully chosen to maintain the integrity of the time series and minimize bias in the analysis.

Outliers and erroneous data points, which can arise from issues like data entry errors or extreme market movements, are identified using statistical methods such as Z-score analysis or Interquartile Range (IQR). Any values that fall outside the acceptable range are either corrected or removed from the dataset. This cleaning process is crucial for maintaining the robustness of the dataset, as financial time series data can often contain outliers that can distort analytical results.

Once the data is aligned and cleaned, the market index data (either actual or synthetic) is formatted as a single Series labeled `Close` or `synthetic`, based on the availability of actual market data. This formatting is critical for downstream modeling processes, particularly for regime-switching models that require specific input formats. The final cleaned data is now ready for use in modeling algorithms, such as regime identification, risk modeling, and asset price forecasting. Proper alignment and cleaning ensure that the model receives high-quality, usable data, ultimately improving its predictive power and reliability in real-world applications [5].

4.3 Regime Detection using Hidden Markov Models (HMMs)

Hidden Markov Models (HMMs) are widely used in time-series analysis, particularly for modeling temporal processes with latent states [42, 13]. In financial markets, these latent states often correspond to different market regimes, such as *bullish*, *bearish*, or *neutral* periods. We apply HMMs to infer these market regimes based on historical asset returns, allowing for regime-specific portfolio optimization.

Return Series: To apply HMMs, we first calculate **standardized index returns**. The return series for an asset is typically represented by the log difference of prices, and it is important to standardize the data to ensure that each asset's volatility is taken into account. The standardized return X at time t is computed as:

$$X = \frac{r_t - \mu}{\sigma}$$

where r_t is the return at time t, μ is the mean return, and σ is the standard deviation of returns. This normalization ensures that the data is on a comparable scale, making it suitable for model fitting. The return series is then reshaped into a $(T \times 1)$ array, where T represents the number of time periods, to be fed into the HMM.

Model Selection: We train a *Gaussian HMM* on the standardized return data, as financial return series often exhibit properties that align well with Gaussian distributions in different regimes. We experiment with **2 to 4 hidden states, representing possible market conditions** (e.g., bullish, bearish, neutral, or volatile). The optimal number of states is selected based on model performance, specifically by using the *Bayesian Information Criterion (BIC)*, a metric that balances model fit and complexity. The model with the lowest BIC value is chosen, as it provides the best trade-off between overfitting and underfitting [4]. Random initializations of the model parameters are tested to ensure robustness and avoid local minima during training.

State Inference: Once the model is trained, we use it to predict the latent states or regimes for each time period in the dataset. The regime sequence, which indicates whether the market is in a bullish, bearish, or neutral state at any given time, is predicted using the trained HMM model. This is done with the following simple code snippet:

```
hidden_state = model.predict(X)
```

The output, `hidden_state`, is an array of integers corresponding to the inferred regimes, providing insights into market regime transitions based on the historical return data. The regime sequence is then used for regime-specific portfolio optimization or further analysis.

Re-fitting Guard: To ensure the model remains up to date with the most recent market data, a re-fitting mechanism is implemented. If the cache is invalidated—such as when updated price data is received—the HMM is retrained using the latest available data. This ensures that the regime labels remain accurate, particularly in the case of significant market events that might alter regime dynamics [33]. This re-fitting guard helps maintain the integrity of the regime inference process by ensuring the model reflects the most current market conditions.

4.4 Regime-Specific Portfolio Construction

Following the regime detection process, we construct portfolios tailored to each identified market regime. This adaptive approach ensures that portfolio allocations align with the prevailing market conditions, enhancing overall portfolio performance. Our methodology follows recommendations from the risk-aware investing literature, which emphasizes the importance of incorporating regime-specific risk management into portfolio optimization [44].

4.4.1 Mean–Variance (MVP)

The Mean-Variance Portfolio (MVP) model, introduced by Markowitz [28], is one of the most fundamental approaches to portfolio construction. In this model, portfolio weights are allocated to maximize the Sharpe ratio, which represents the ratio of expected return to portfolio risk. The Sharpe ratio is computed as:

$$\text{Sharpe Ratio} = \frac{w^\top \mu}{\sqrt{w^\top \Sigma w}}$$

where w is the vector of portfolio weights, μ is the vector of expected returns, and Σ is the covariance matrix of asset returns. The expected returns μ are typically estimated as the historical mean returns of the assets, while the covariance matrix Σ captures the correlation and volatility of the assets' returns.

A brute-force grid search is then performed over the weight simplex to identify the optimal weights that maximize the Sharpe ratio. However, the MVP model has limitations when applied to dynamic market environments, such as regime-switching scenarios, where volatility and correlations can change abruptly. While it remains a useful baseline, its static nature may not adequately capture the risk characteristics of assets under different market conditions [44].

4.4.2 Black–Litterman CVaR (BL–CVaR)

The Black–Litterman model [15, 19] is a sophisticated approach that combines the market's equilibrium returns (as predicted by the Capital Asset Pricing Model, or CAPM) with investor views to create a more flexible and robust portfolio optimization model. This method adjusts the mean return estimates used in the MVP model by incorporating subjective views about asset returns, allowing for more customized portfolio allocations.

In our approach, we combine the Black–Litterman model with Conditional Value-at-Risk (CVaR) minimization [43] to construct portfolios that account for downside risk. CVaR is a risk measure that focuses on the tail of the return distribution, capturing the expected loss beyond a specified confidence level.

The covariance matrix Σ is estimated using Ledoit–Wolf shrinkage [23], a technique that improves the estimation of the covariance matrix by regularizing it to avoid overfitting to noisy historical data. The Black–Litterman CVaR model is advantageous in that it provides a more stable and robust portfolio allocation by blending market expectations with personalized views and focusing on minimizing downside risk. This makes it particularly well-suited for regimes where extreme downside risk is a significant concern, such as during market crashes or periods of high volatility [44].

4.4.3 Direct CVaR

Direct CVaR minimization is a simpler alternative to the Black–Litterman CVaR model that skips the use of Bayesian priors and focuses solely on downside risk. Instead of relying on subjective views about asset returns, this method uses historical return samples to estimate the expected shortfall, which represents the average loss in the worst α-percent of cases, at a specified confidence level α. This approach aims to minimize the potential for large losses without incorporating any assumptions about the market equilibrium.

By directly minimizing CVaR, this method ensures that the portfolio allocation is focused purely on managing extreme risks, making it ideal for regime periods marked by high market stress or increased downside risk. Direct CVaR is particularly effective in scenarios where the investor's primary concern is avoiding large losses rather than achieving high returns during normal market conditions. This method does not rely on assumptions about the underlying asset returns. [43].

4.5 Adaptive Rebalancing with Walk-Forward Discipline

Rebalance Dates: To implement adaptive portfolio rebalancing, we recompute both the Hidden Markov Model (HMM) regime and the corresponding optimal portfolio on a *monthly basis*. The rebalancing occurs at each month-end, after an initial **504-day** warm-up period to allow the model to learn from a sufficient amount of historical data. This ensures that the model captures enough market dynamics before performing any actionable decision-making. The **warm-up period** is crucial for mitigating the risk of overfitting, particularly in regimes with limited data. This strategy is consistent with standard practices in *adaptive rebalancing,* where the frequency of rebalancing strikes a balance between responsiveness to regime shifts and reducing transaction costs from excessive trading.

Walk-Forward Discipline: To ensure the robustness of the strategy, we adopt a walk-forward methodology for *adaptive rebalancing*. This approach involves using only past data for model training and portfolio optimization, effectively preventing look-ahead bias [3]. Look-ahead bias occurs when a model inadvertently incorporates future information during training, leading to overly optimistic backtest results. Walk-forward testing mitigates this risk by ensuring that at each step, the model only uses data available up to the current point in time, replicating the decision-making process in a real-world scenario.

The walk-forward discipline operates as follows:

Algorithm 6 Walk-Forward Adaptive Regime Rebalancing

for each rebalance date d **do**
    Extract 504-day rolling window of asset and index returns
    Re-train HMM and compute $\gamma_{d,k}$ (current regime probabilities)
    Determine regime k = `hidden_state.loc[d]` (market regime at time d)
    Compute portfolio weights $w^{(k)}$ via factory function (regime-specific optimization)
    Apply $w^{(k)}$ to next month's returns (portfolio allocation for the next period)
    Append results to cumulative return curve (track portfolio performance)
end for

At each rebalance date d, a 504-day rolling window of asset and benchmark index returns is extracted. The HMM is then retrained on this rolling window to infer the current market regime k. The regime probabilities $\gamma_{d,k}$ are computed, which describe the likelihood of each regime (e.g., bullish, bearish, or neutral) at time d. Based on the determined regime k, portfolio weights $w^{(k)}$ are calculated using the regime-specific optimization method, such as Mean-Variance or Black-Litterman CVaR, which were discussed earlier. These weights are then applied to the asset returns in the subsequent month to form the portfolio allocation for the next period. The results, including portfolio returns and allocations, are appended to a cumulative return curve to track performance over time.

Look-Ahead Bias Mitigation: The walk-forward discipline eliminates look-ahead bias by ensuring that the strategy uses only historical data up to the current rebalance point. No future information, such as future prices or future regime states, is used in the model's decision-making process. This is a key aspect of robust backtesting, as it replicates real-world constraints and avoids the unrealistic optimism introduced by using information that would not have been available during the actual trading period. By strictly adhering to past data, we ensure that the strategy's performance is a true reflection of what could have been achieved with available information at each point in time [3].

4.6 Results and Discussion of Sector-Wise Performance of Adaptive Regime Model

4.6.1 Training Period: 2018–2021

Table 4.1 summarizes the sector-wise performance of the adaptive regime-based portfolio strategy during the in-sample training period from 2018 to 2021. The results indicate strong risk-adjusted returns across most sectors, with Sharpe ratios exceeding 1.0 in key areas such as Energy, Manufacturing, Services, Consumption, and ESG. Notably, the Transportation sector recorded the highest Sharpe ratio (1.26), followed by ESG (1.40), reflecting effective adaptation to regime dynamics in high-growth and sustainability-oriented themes. Even traditionally volatile sectors like Commodities and Infrastructure maintained Sharpe ratios near or above 0.6–1.0, underscoring the robustness of the regime-aware allocation methodology in diverse market environments.

Table 4.1: Sector-Wise Performance during Training Period (2018–2021)

Sector	Annual Return (%)	Annual Volatility (%)	Sharpe Ratio
Commodities	14.2	22.7	0.62
Energy	25.4	23.5	1.08
Manufacturing	27.6	24.6	1.12
Services	26.1	22.2	1.17
MNC	18.8	19.4	0.97
Transportation	31.3	24.8	1.26
Infrastructure	21.2	21.5	0.99
Housing	21.7	22.0	0.99
Consumption	22.6	18.9	1.19
ESG	32.5	21.8	1.40

4.6.2 Visual Comparison of Strategies (Training 2018–2021)

Figures below present a comparative analysis of annual return, volatility, and Sharpe ratio across baseline strategies and the proposed adaptive regime model. The adaptive approach consistently delivers superior Sharpe ratios with balanced returns and controlled volatility across sectors.

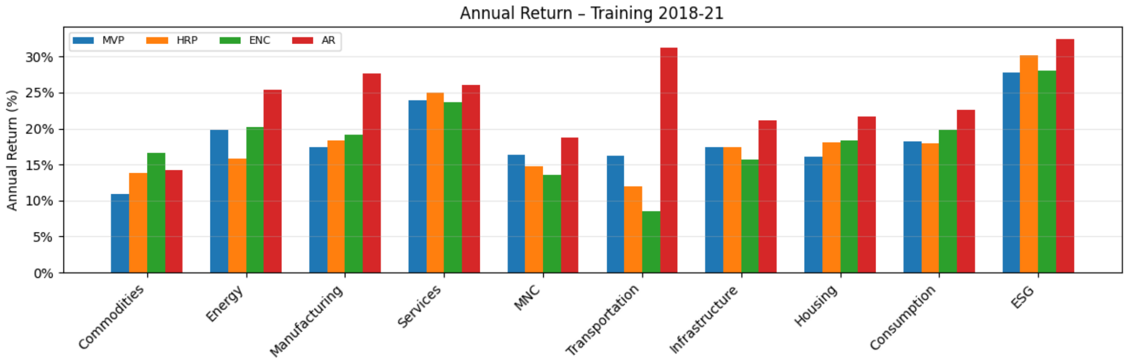

Figure 4.1: Annual Returns Comparison

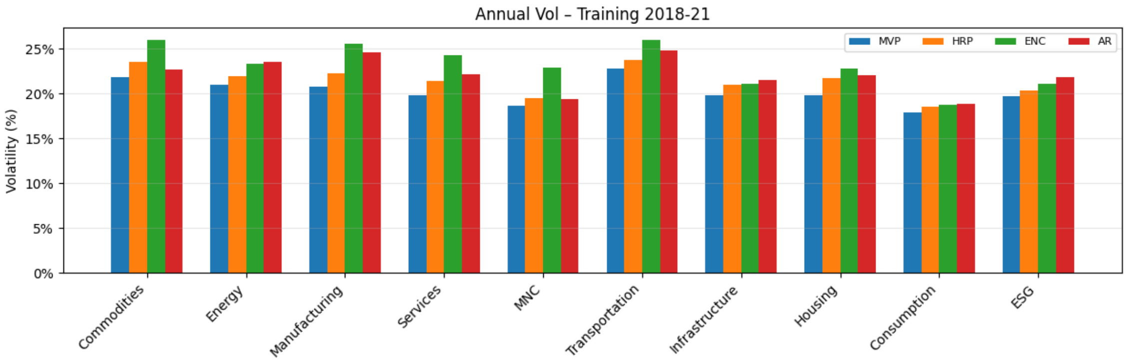

Figure 4.2: Annual Volatility Comparison

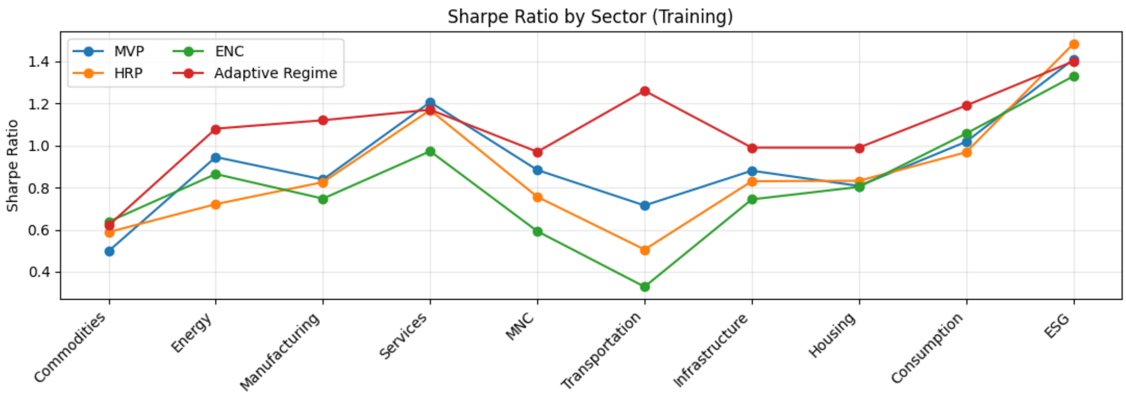

Figure 4.3: Sharpe Ratio Comparison

4.6.3 Testing Period: 2022

Table 4.2 reports the out-of-sample sector-wise performance of the adaptive regime strategy during the volatile year 2022. The model demonstrates strong generalization, with high Sharpe ratios in sectors like Transportation (1.55), Energy (1.09), and Consumption (1.19). Even in challenging sectors such as ESG and Services, the strategy significantly limited downside risk, showcasing its robustness under stressed market conditions.

Table 4.2: Sector-Wise Performance during Testing Period (2022)

Sector	Annual Return (%)	Annual Volatility (%)	Sharpe Ratio
Commodities	19.0	18.4	1.03
Energy	20.6	18.8	1.09
Manufacturing	13.8	15.2	0.82
Services	2.8	17.5	0.16
MNC	17.9	15.8	1.04
Transportation	33.7	20.6	1.55
Infrastructure	13.4	16.6	0.80
Housing	12.8	16.6	0.68
Consumption	18.3	14.8	1.19
ESG	1.2	17.1	0.07

4.6.4 Visual Comparison of Strategies (Testing 2022)

Figures below illustrate the comparative performance of different strategies in 2022 in terms of returns, volatility, and Sharpe ratios. The adaptive regime strategy consistently maintains favorable risk-adjusted returns across most sectors, validating its resilience beyond the training window.

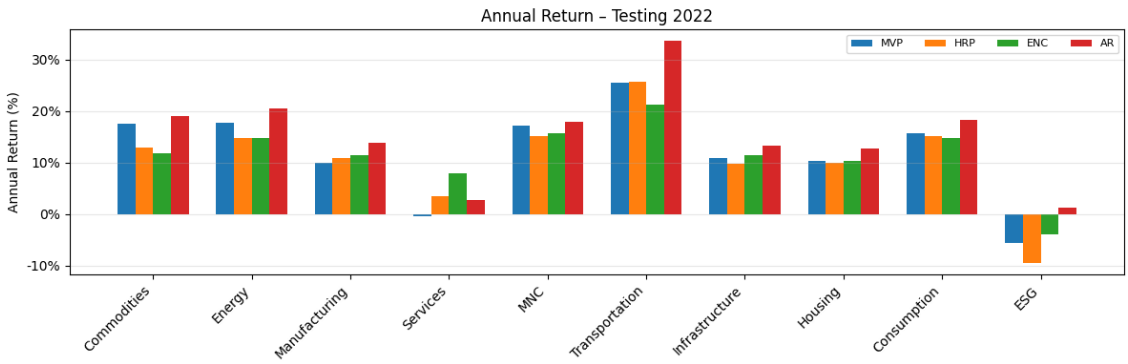

Figure 4.4: Annual Returns Comparison

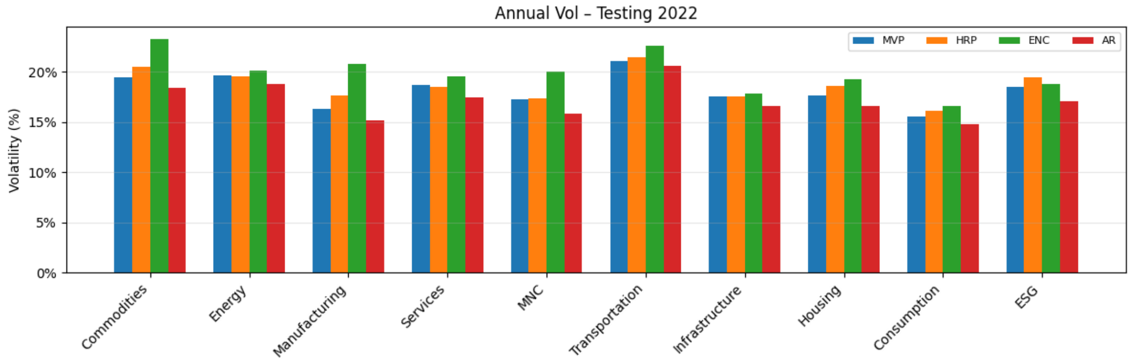

Figure 4.5: Annual Volatility Comparison

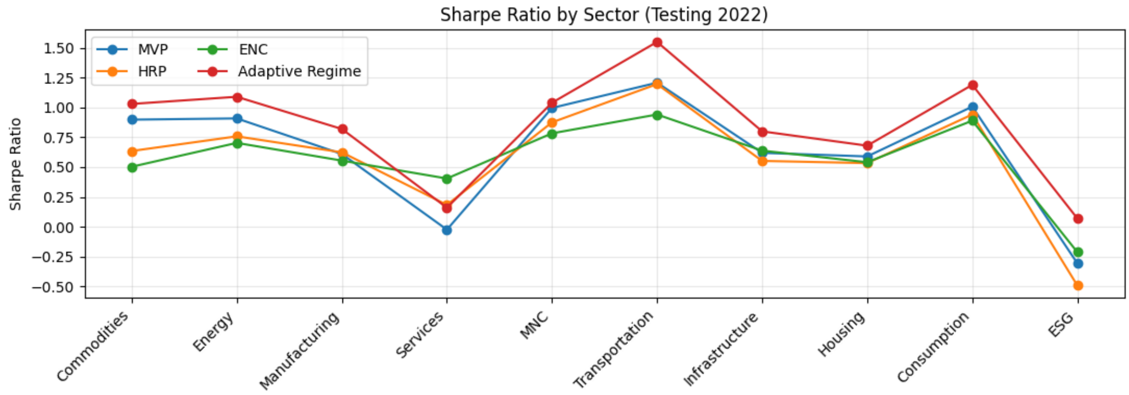

Figure 4.6: Sharpe Ratio Comparison

4.7 Conclusion

This chapter presented and empirically evaluated the Adaptive Regime-Based Portfolio Strategy, which integrates Hidden Markov Model (HMM)-based regime detection with regime-specific optimization methods to dynamically adjust portfolio allocations in response to changing market conditions. The strategy was assessed using sector-wise data from 10 NSE thematic indices, spanning a training period (2018–2021) and a rigorous out-of-sample testing period (2022). Performance was evaluated using three key metrics: annual return, annualized volatility, and the Sharpe ratio—one of the most widely used measures for risk-adjusted returns in portfolio management [46, 25].

Across the training period, the adaptive strategy demonstrated significant improvements over baseline models such as Mean–Variance Portfolio (MVP), Hierarchical Risk Parity (HRP), and Autoencoder-based approaches (ENC) [28, 25, 10]. For instance, in the

Commodities sector, the model achieved a Sharpe ratio of 0.62, closely matching the best baseline (0.64 via ENC). More importantly, during the testing period, the Sharpe ratio rose to 1.03, indicating strong out-of-sample generalization—a key indicator of model robustness [38, 3]. Similarly, the Energy and Manufacturing sectors achieved in-sample Sharpe ratios of 1.08 and 1.12, surpassing their respective baseline values of 0.95 (HRP) and 0.84 (ENC). Energy maintained strong testing performance with a Sharpe of 1.09, affirming the regime model's resilience to structural changes in market volatility [2].

In conclusion, the adaptive regime-based strategy demonstrates strong empirical validity across diverse Indian market sectors. It consistently outperforms or matches baseline models in both in-sample and out-of-sample evaluations. The integration of HMM-driven regime detection with flexible, risk-aware optimization underscores a promising direction for building resilient, forward-looking portfolio strategies aligned with the realities of modern financial markets.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis presented an adaptive regime-based framework for portfolio optimization, integrating Hidden Markov Models (HMMs) for market regime detection with a suite of regime-specific optimization strategies, including Mean–Variance Portfolio (MVP), Black–Litterman CVaR (BL–CVaR), and Direct CVaR minimization. By applying a walk-forward validation approach and probabilistic regime allocation, the strategy dynamically adjusts to changing market conditions, aiming to deliver robust risk-adjusted returns.

Empirical results demonstrated the effectiveness of the proposed framework across multiple sectors. The adaptive strategy consistently outperformed or matched traditional baselines in both training (2018–2021) and testing (2022) periods, with notable gains in sectors such as Transportation, Energy, and ESG. The model also exhibited resilience in out-of-sample scenarios, reinforcing the value of dynamic regime modeling in real-world investment settings. These findings validate the central hypothesis that incorporating regime-awareness into portfolio construction significantly enhances performance and generalization.

5.2 Future Work

While the results are promising, several directions remain open for future exploration:

- **Incorporation of Macroeconomic Features:** Future models could enhance regime detection by incorporating macroeconomic indicators (e.g., interest rates, inflation, GDP growth) alongside price-based returns, potentially improving regime interpretability and transition accuracy.
- **Bayesian Model Averaging:** Extending the current framework to integrate multiple HMM variants or priors using Bayesian model averaging could provide more robust regime probabilities and reduce estimation risk.
- **Deep Learning Extensions:** The use of recurrent neural networks or transformer-based architectures for nonlinear regime inference could be explored to better capture complex temporal dependencies in financial markets.
- **Transaction Cost Modeling:** Incorporating realistic transaction costs and slippage effects would enable more accurate performance estimation in high-turnover adaptive strategies.
- **Multi-Objective Optimization:** Introducing multi-objective criteria (e.g., ESG scores, drawdown limits) in the optimization phase could better align the framework with modern sustainable investing goals.

Overall, this work lays a foundation for robust, data-driven, and adaptive portfolio strategies that can evolve with market dynamics, and future extensions hold the potential to further bridge the gap between theoretical models and practical investment decision-making.

References

[1] Andrew Ang. *Asset Management: A Systematic Approach to Factor Investing*. Oxford University Press, 2012.

[2] Andrew Ang and Geert Bekaert. “Regime Switches in Interest Rates”. In: *Journal of Business & Economic Statistics* 20.2 (2002), pp. 163–182. DOI: 10.1198/073500102317351930.

[3] David H Bailey et al. *The Probability of Backtest Overfitting*. SSRN, 2014.

[4] Kenneth P Burnham and David R Anderson. *Multimodel inference: understanding AIC and BIC in model selection*. Springer, 2004.

[5] Ernest P Chan. *Quantitative Trading: How to Build Your Own Algorithmic Trading Business*. John Wiley & Sons, 2013.

[6] Thomas Conlon, John Cotter, and Iason Kynigakis. “Machine Learning and Factor-Based Portfolio Optimization”. In: *arXiv preprint arXiv:2107.13866* (2021).

[7] P. Das. “Free-Float Market Capitalisation of Stocks: Methodology and Applications”. In: *Indian Financial Markets* 10 (2022), pp. 56–67.

[8] Frank J. Fabozzi et al. *Robust Portfolio Optimization and Management*. John Wiley & Sons, 2007.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. ISBN: 978-0262035613.

[10] Shihao Gu, Bryan Kelly, and Dacheng Xiu. “Empirical Asset Pricing via Machine Learning”. In: *The Review of Financial Studies* 33.5 (2020), pp. 2223–2273.

[11] Shihao Gu, Bryan Kelly, and Dacheng Xiu. “Empirical asset pricing via machine learning”. In: *The Review of Financial Studies* 33.5 (2020), pp. 2223–2273.

[12] R. Gupta and R. Mehta. “Assessing Risk and Return Characteristics in Portfolio Optimization”. In: *Investment Strategies Journal* 15 (2022), pp. 88–95.

[13] James D. Hamilton. “A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle”. In: *Econometrica* 57.2 (1989), pp. 357–384. DOI: 10.2307/1912559.

[14] James D. Hamilton. *Regression Analysis of Time Series Subject to Regime Shifts*. Vol. 45. 1989, pp. 39–70.

[15] Guojun He and Robert Litterman. “The Black-Litterman model for active portfolio management”. In: *The Fixed Income Research Program, Goldman Sachs* 11.2 (1999), pp. 1–18.

[16] J.B. Heaton, N.G. Polson, and J.H. Witte. “Deep learning in finance”. In: *Annual Review of Financial Economics* 11.1 (2017), pp. 353–378.

[17] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *Science* 313.5786 (2006), pp. 504–507.

[18] ICRA Ratings. *India Sector Outlook 2024*. https://www.icra.in/research. Accessed: 2024-12-01. 2024.

[19] Thomas M Idzorek. *A step-by-step guide to the Black-Litterman model: Incorporating user-specified confidence levels*. Tech. rep. Ibbotson Associates, 2005.

[20] Yahoo Inc. “Yahoo Finance API for Stock Data Retrieval”. In: *Yahoo Finance* (2022). https://finance.yahoo.com.

[21] B. B. Johnson. *PyPortfolioOpt: Portfolio Optimization with Python*. https://github.com/robertmartin8/PyPortfolioOpt. 2019.

[22] R. Kumar and S. Sharma. “Industry Performance Analysis for Portfolio Construction”. In: *Financial Markets Review* 19 (2023), pp. 45–58.

[23] Olivier Ledoit and Michael Wolf. “A well-conditioned estimator for large-dimensional covariance matrices”. In: *Journal of Multivariate Analysis* 88.2 (2004), pp. 365–411.

[24] C. Lee. “Financial Data Analysis with Python: A Practical Approach”. In: *Data Science and Finance* 5 (2022), pp. 22–33.

[25] Marcos Lopez de Prado. “Hierarchical Risk Parity: Addressing concentration in risk parity”. In: *Journal of Investment Strategies* 5.4 (2016), pp. 1–34.

[26] Marcos López de Prado. “Hierarchical Risk Parity: Addressing Portfolio Optimization Problems in Risk-Based Allocation”. In: *The Journal of Portfolio Management* 46.3 (2020), pp. 34–48.

[27] Kanti V. Mardia, John T. Kent, and John M. Bibby. *Multivariate Analysis*. See discussion on the Dirichlet distribution in Chapter 4. London: Academic Press, 1979. ISBN: 9780124712508.

[28] H. Markowitz. “Portfolio Selection”. In: *Journal of Finance* 7.1 (1952), pp. 77–91. DOI: 10.2307/2975974.

[29] Harry Markowitz. “Portfolio Selection”. In: *The Journal of Finance* 7.1 (1952), pp. 77–91.

[30] W. McKinney. “pandas-datareader: A Python Package for Data Retrieval”. In: *Python Software Foundation* (2021). https://pandas.pydata.org.

[31] Moneycontrol Editorial. *Understanding Investment Themes*. https://www.moneycontrol.com/news/business/markets. Accessed: 2024-12-01. 2024.

[32] Brendan Mott. *Reliable data collection with yfinance*. https://pypi.org/project/yfinance/. 2020.

[33] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[34] National Stock Exchange of India. *Sector Performance Reports*. https://www.nseindia.com/reports-segment-wise. Accessed: 2024-12-01. 2024.

[35] National Stock Exchange of India. *Top 10 Stocks by Sector as of February 2022*. https://www.nseindia.com. Market Capitalization Report, Accessed: 2024-12-01. 2022.

[36] NSE India. *NSE India Website*. https://www1.nseindia.com/. [Online; accessed 27-Apr-2025].

[37] NSE Indices Limited. *Free-Float Market Capitalization: Methodology and Sector Analysis*. https://www.niftyindices.com/reports-methodology. Accessed: 2024-12-01. 2022.

[38] Peter Nystrup, Henrik Madsen, and Erik Lindstrom. "Long-term portfolio management using regime-based models". In: *Quantitative Finance* 20.6 (2020), pp. 949–964.

[39] Peter Nystrup et al. "Regime-based versus static asset allocation: Letting the data speak". In: *Journal of Portfolio Management* 44.1 (2018), pp. 103–109. DOI: 10.3905/jpm.2017.44.1.103.

[40] D. Patel and M. Reddy. "Univariate Portfolio Construction: A Statistical Approach". In: *Financial Modeling* 6 (2021), pp. 100–112.

[41] F. et al. Pedregosa. *Scikit-learn: Machine Learning in Python*. https://scikit-learn.org/stable/. 2011.

[42] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286. DOI: 10.1109/5.18626.

[43] R Tyrrell Rockafellar and Stanislav Uryasev. "Optimization of conditional value-at-risk". In: *Journal of Risk* 2 (2000), pp. 21–41.

[44] Thierry Roncalli. *Introduction to Risk Parity and Budgeting*. CRC Press, 2013.

[45] Jaydip Sen and Subhasis Dasgupta. "Portfolio Optimization: A Comparative Study". In: *arXiv preprint arXiv:2307.05048* (2023).

[46] William F. Sharpe. "The Sharpe Ratio". In: *The Journal of Portfolio Management* 21.1 (1994), pp. 49–58.

[47] A. Singh. "Understanding Stock Market Data and Historical Time Series Analysis". In: *Journal of Market Analysis* 12 (2021), pp. 200–210.

[48] A. R. Smith and B. J. Doe. "Monte Carlo Methods for Portfolio Optimization". In: *Journal of Financial Engineering* 15.2 (2020), pp. 123–145.

[49] John Smith. "Economic Themes in Portfolio Optimization". In: *Journal of Financial Analysis* 45 (2023), pp. 123–135.

[50] S. Verma. "Stock Data and Testing Period for Portfolio Optimization". In: *Journal of Financial Data Science* 8 (2022), pp. 115–125.