Using Machine Learning to Investigate the Role of Real Estate in a Mixed-Asset Portfolio

Fatim Zahra Habbab

A thesis submitted for the degree of Doctor of Philosophy in Computational Finance

Centre for Computational Finance and Economic Agents
School of Computer Science and Electronic Engineering
University of Essex

September, 2024

This thesis is dedicated to my mother, who offered me all the support possible, and to my father, whose memory continues to inspire and motivate me every day.

Abstract

Investing in real estate offers significant benefits, such as diversification and potential long-term appreciation, making it an attractive option compared to stocks and bonds. However, direct investments in real estate often require substantial capital, which is a barrier for many individual investors. To overcome this, investors often use Real Estate Investment Trusts (REITs), which allow for indirect investment in real estate through shares in companies that own income-generating properties.

This study examines the added value of including real estate in a diversified investment portfolio, utilising innovative methods to optimise asset allocation. Instead of relying on historical data, it employs machine learning algorithms (such as Linear Regression, Support Vector Regression, k-Nearest Neighbours, Extreme Gradient Boosting, and LSTM Neural Networks) to predict future asset prices. The study also incorporates Technical Analysis Indicators (TAIs) to further improve predictive accuracy.

Furthermore, a Genetic Algorithm (GA) is used to determine optimal portfolio weightings, considering the expected returns and risks of each asset class. The study compares the performance of portfolios constructed using price predictions with those based on historical data, assessing diversification benefits and risk-adjusted returns.

Overall, by integrating machine learning techniques, technical analysis, and optimisation algorithms, the study aims to demonstrate the potential advantages of including real estate investments in a diversified portfolio, enabling investors to make more informed decisions and improve their investment outcomes.

Acknowledgements

First of all, I would like to thank my supervisor, Dr. Michael Kampouridis, who helped me acquire crucial skills for my future as a researcher. He generously gave me his time and worked with dedication and patience from the beginning. I couldn't have asked for more! I would also like to thank my colleagues at the University of Essex, who have shared their knowledge and experience, making my research journey enriching and interesting. I am also grateful to my mother, who supported me in every possible way during my PhD journey and believed in me even when I could not. Additionally, I would like to thank my friends, both those I met before and during my PhD studies, who provided emotional support and made my journey unforgettable—they are like family to me. Last, but not least, I thank Allah for helping me through all the difficulties. I will continue to trust in you for my future.

Contents

Figures and Tables

Figures

Tables

Chapter 1

Introduction

1.1 Motivation and Objectives

The motivation behind this research lies in the crucial role of optimising portfolios that incorporate real estate within the finance domain [1]. Achieving an optimal asset allocation is fundamental for minimising risk and maximising returns in investment portfolios [2], with real estate serving as a key option for diversification alongside traditional asset classes such as stocks, bonds, and cash.

Various studies have explored the benefits of investing in real estate [3, 4, 5], including risk reduction and diversification opportunities through correlations between real estate and other asset classes [6]. Additionally, real estate investments have demonstrated effectiveness as inflation hedges [7] and have shown potential for enhancing risk-adjusted returns due to their low correlation with traditional assets [8].

Real Estate Investment Trusts (REITs) provide investors with exposure to real estate markets without the need for direct property ownership. Research consistently highlights the diversification benefits of incorporating REITs into portfolios, given their low correlations with stocks and bonds and potential for improving risk-adjusted returns [9].

Despite these advantages, optimising portfolios that include real estate presents challenges, particularly in accurately predicting REIT prices and determining optimal asset weights. While machine learning algorithms have been applied to predict REIT prices, the literature predominantly focuses on neural networks. This research aims to explore alternative machine learning techniques for predicting REIT prices and optimising mixed-asset portfolios that include real estate.

Furthermore, the study explores the incorporation of Technical Analysis Indicators (TAIs) to improve prediction accuracy and evaluates portfolio performance metrics such as the Sharpe ratio, returns, and risk. Ultimately, the research aims to provide valuable insights into the diversification benefits of adding real estate to a multi-asset portfolio and offers guidance for investors seeking resilient investment strategies.

In addition to individual algorithm applications, several studies have compared machine learning algorithms to ARIMA for REIT return prediction. Noteworthy examples include the use of artificial neural networks and multiple variables [10, 11, 12]. Our research contributes to this field by exploring alternative machine learning techniques for predicting REIT prices, expanding beyond the prevalent use of neural networks in the current literature.

The price predictions generated by our machine learning algorithms serve as inputs for optimising a multi-asset portfolio that includes REITs. We employ a genetic algorithm (GA) to determine optimal weights for assets based on return and risk parameters derived from Modern Portfolio Theory (MPT) concepts [13, 14, 15]. Our main objective is to demonstrate that utilising machine learning price predictions results in enhanced portfolio performance. We evaluate key financial metrics such as the Sharpe ratio, returns, and risk and compare the outcomes with two benchmarks.

In conclusion, our study involves a comprehensive comparison of portfolios based on price predictions, aiming to demonstrate the diversification benefits of adding real estate to a multi-asset portfolio. By evaluating accuracy, profitability, and risk, we provide a comprehensive view of portfolio performance, emphasising the potential advantages of integrating real estate assets for investors seeking a well-balanced and resilient investment strategy.

1.2 Novelty of Research

The novelty of this research lies in its comprehensive approach to optimising portfolios that include real estate assets. While previous studies have explored the benefits of real estate investments and the use of machine learning for price prediction, this research introduces several innovative elements.

Firstly, the study adopts a two-step approach, focusing on both predicting REIT prices using machine learning algorithms [16, 17] and optimising portfolio allocation using a genetic algorithm [13, 14, 15]. This integrated approach allows for a more holistic assessment of portfolio management strategies.

Secondly, the research explores alternative machine learning techniques beyond the prevalent use of neural networks for REIT price prediction. By evaluating the performance of various algorithms such as Ordinary Least Squares Linear Regression, Support Vector Regression, k-Nearest Neighbours Regression, Extreme Gradient Boosting, and Long/Short-Term Memory Neural Networks, the study contributes to a deeper understanding of the most effective methods for predicting REIT prices [16, 17, 10, 11, 12].

Thirdly, the incorporation of Technical Analysis Indicators (TAIs) to enhance prediction accuracy represents a novel extension of the already used machine learning approaches [2]. By integrating TAIs into the prediction process, the study seeks to improve the reliability of price forecasts and subsequently optimise portfolio allocation more effectively.

Lastly, the evaluation of portfolio performance metrics such as the Sharpe ratio, returns, and risk provides a comprehensive assessment of the diversification benefits of including real estate in multi-asset portfolios. By comparing portfolios based on price predictions with those optimised using traditional methods, the research offers valuable insights into the potential advantages of integrating real estate assets into investment strategies [18, 19, 20].

In summary, the study's novelty lies in its approach to portfolio optimisation, involving innovative techniques for REIT price prediction, integration of Technical Analysis Indicators, and comprehensive evaluation of portfolio performance metrics. Through these contributions, the research aims to advance understanding in the field of real estate investment and portfolio management.

1.3 Thesis Structure

The remainder of this thesis is structured as follows. Chapter 2 presents an overview of financial markets, including the real estate, stock, and bond market, of the Modern Portfolio Theory (MPT). We also present an overview of machine learning algorithms used in this thesis, both for optimisation – i.e., genetic algorithm – and regression – i.e., Ordinary Least Squares Linear Regression (LR), Support Vector Regression (SVR), eXtreme Gradient Boosting (XGBoost), Long/Short-Term Memory Neural Networks (LSTM), and k-Nearest Neighbours Regression (KNN). Chapter 3 presents a literature review of different portfolio optimisation techniques adopted in previous studies, including those regarding real estate investments, and financial forecasting methods. Chapter 4 presents an exploratory analysis that aims at demonstrating the potential improvement of a portfolio performance that comes from incorporating price predictions (instead of historical data) in the portfolio optimisation problem. Chapter 5 presents the second contribution of this thesis, which refers to the regression techniques adopted in order to predict the prices of REITs, stocks, and bonds. In that way, we demonstrate the importance of using machine learning algorithms rather than other methods, including Holt's Linear Trend Method (HLTM), Trigonometric Box-Cox Autoregressive Time Series (TBATS), and Autoregressive Integrated Moving Average (ARIMA), in order to improve the accuracy of predictions. Our study aimed to determine if this approach results in better portfolio performance compared to using historical data for optimising the weights of a mixed-asset portfolio that includes REITs. Chapter 6 presents further experimental findings regarding the inclusion of additional features in the form of Technical Analysis indicators (TAIs) in order to improve the accuracy of financial time series, and thus the risk-adjusted performance of a mixed-asset portfolio. Chapter 7 compares the performance of two mixed-asset portfolios built using price predictions obtained from TAIs, one including real estate and one not including it. In that way, we again demonstrate the added value of real estate investments in the context of a prediction-based, mixed-asset portfolio. Finally, Chapter 8 concludes the thesis and presents suggestions for further research.

1.4 Publications

The list of publications from the research described in this thesis in peer-reviewed journals is as follows:

- Fatim Z Habbab and Michael Kampouridis. “An in-depth investigation of five machine learning algorithms for optimizing mixed-asset portfolios including REITs”. In: *Expert Systems with Applications* 235 (2024). Impact Factor: 8.5, p. 121102.

The list of publications from the research described in this thesis in conference proceedings is as follows:

- Fatim Z Habbab, Michael Kampouridis and Alexandros A Voudouris. “Optimizing mixed-asset portfolios involving REITs”. In: *2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*. IEEE. 2022, pp. 1–8.
- Fatim Z Habbab and Michael Kampouridis. “Optimizing Mixed-Asset Portfolios With Real Estate: Why Price Predictions?” In: *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2022, pp. 1–8.
- Fatim Z Habbab and Michael Kampouridis. “Machine learning for real estate time series prediction”. In: *2022 UK Workshop on Computational Intelligence (UKCI)* (Sheffield, UK). IEEE, 2022.
- Fatim Z Habbab, Michael Kampouridis and Tasos Papastylianou. “Improving REITs Time Series Prediction Using ML and Technical Analysis Indicators”. In: *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2023, pp. 1–8.
- Fatim Z Habbab and Michael Kampouridis. “Optimizing a prediction-based, mixed-asset portfolio including REITs”. In: *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2023, pp. 1–4.

Chapter 2

Background Information

2.1 Introduction

In this chapter, we provide background information about financial markets (Section 2.2), the Modern Portfolio Theory (Section 2.3), and machine learning algorithms (Section 2.4). The financial markets considered in this study are the real estate market (Section 2.2.1), the stock market (Section 2.2.2), and the bond market (Section 2.2.3). On the other side, since we aim to optimise a multi-asset portfolio made of real estate investments, stocks, and bonds, we add an explanation of the Modern Portfolio Theory (MPT). Finally, the machine learning (ML) algorithms used in this study are the genetic algorithm (used for optimisation), explained in Section 2.4.1, and other supervised learning algorithms used for regression (Section 2.4.2), including Ordinary Least Squares Linear Regression, Support Vector Regression, k-Nearest Neighbours Regression, Extreme Gradient Boosting, and Long/Short-Term Memory Neural Networks.

2.2 Financial Markets

Financial markets are virtual places where individuals, institutions, and governments exchange financial instruments, including stocks, bonds, currencies, commodities, derivatives, and real estate. Their goal is to facilitate the transfer of funds between borrowers and lenders, investors and issuers, and buyers and sellers. They play a crucial role in the global economy by providing tools for capital allocation, risk management, and pricing of financial instruments[27].

Financial markets can be categorised into primary markets, where new securities are issued and sold for the first time, allowing government entities and businesses to raise capital, and secondary markets, where previously issued securities are traded among investors. Secondary markets are generally more liquid, allowing investors to buy and sell financial securities easily [28].

Some of the primary asset classes traded within financial markets include stocks, bonds, and real estate [27]. Stock markets provide a platform for the purchase and sale of shares of publicly traded companies, while bond markets facilitate the issuance and trading of debt securities [28]. Real estate markets, on the other hand, involve the acquisition, disposal, and rental of various types of properties, encompassing land, residential homes, commercial buildings, and other real estate assets [29].

The following sections describe in detail each of the above-mentioned financial markets. Specifically, Section 2.2.1 examines the factors influencing real estate markets, the role they play in the economy, and the participants involved in real estate transactions; Section 2.2.2 defines the stock exchanges, market participants, trading strategies, and factors affecting stock prices; and Section 2.2.3 provides an in-depth analysis of bond markets, including different types of bonds, yield curves, bond pricing, and risk factors associated with fixed income investments.

2.2.1 The Real Estate Market

Real estate markets refer to the virtual places where the buying, selling, and leasing of properties take place [30]. There are different types of real estate assets traded, such as residential homes, commercial buildings, land, and different kinds of properties [31]. Participants in real estate markets include individuals, investors, developers, real estate agents, financial institutions, and government entities.

One of the key characteristics of real estate markets is that they are strongly tied to specific geographic areas [30]. The value and demand for properties can vary significantly based on factors such as their location, amenities, infrastructure quality, economic conditions, and local market trends [30]. Real estate markets are influenced by supply and demand dynamics, population growth, urbanisation, interest rates, and government policies related to zoning, taxation, and regulations [32].

Real estate markets can be classified into different types based on the nature of properties and their intended purposes. Some of the main types of real estate markets include the following sectors.

Residential Real Estate Market. The residential market focuses on properties designed for housing purposes, such as single-family homes, apartments, townhouses, and vacation properties [30]. This market addresses the demand for living spaces and caters to individuals and families seeking residential properties [30].

Commercial Real Estate Market. The commercial market revolves around properties intended for commercial use, including office buildings, retail spaces, industrial facilities, and hotels [31]. This market serves businesses by providing suitable spaces for various commercial activities, such as offices, stores, and manufacturing facilities [31].

Industrial Real Estate Market. The industrial market deals with properties specifically tailored for industrial purposes, such as warehouses, distribution centres, and manufacturing facilities [30]. This market caters to businesses involved in logistics, storage, and production, meeting their specific operational needs [30].

Retail Real Estate Market. The retail market focuses on properties used for retail activities, including shopping malls, strip malls, and standalone stores [31]. This market addresses the requirements of retailers and businesses involved in direct consumer sales, offering spaces for showcasing products and attracting customers [31].

Office Real Estate Market. The office market primarily deals with office spaces in commercial buildings and business parks [30]. It caters to businesses seeking professional work environments, including corporate offices, co-working spaces, and administrative facilities [30].

Hospitality Real Estate Market. The hospitality market encompasses properties designed for accommodation and lodging purposes, such as hotels, resorts, and vacation rentals [31]. This market caters to the hospitality industry, providing spaces for travellers and tourists seeking temporary stays [31].

Agricultural Real Estate Market. The agricultural market involves properties utilised for agricultural activities, including farmland, vineyards, and ranches [30]. This market serves agricultural businesses and individuals involved in farming, crop production, and livestock rearing [30].

Specialised Real Estate Markets. Specialised real estate markets are tailored to specific niche segments within the broader real estate industry, such as healthcare real estate (hospitals, medical centres), educational real estate (schools, universities), senior housing, and self-storage facilities. These focused markets are designed to meet the various needs and preferences of particular sectors, ensuring that properties are customised to their specific requirements [31].

Real Estate Investment Trusts

An investor can gain exposure to real estate markets either in a direct or indirect way [33]. Direct real estate investments involve individuals or entities acquiring properties directly, either as sole owners or in partnership with others [33]. This form of investment provides investors with direct ownership and control over the property. Investors hold the responsibility for property management, including tasks such as maintenance, tenant acquisition, and rental income collection. Direct real estate investments offer the potential for higher control and customisation, allowing investors to make decisions regarding property operations and value-add initiatives [31]. However, direct real estate investments require a significant amount of capital, time, and expertise for property acquisition, management, and dealing with associated risks [33].

On the other hand, indirect real estate investments involve investing in real estate through intermediaries such as Real Estate Investment Trusts (REITs), real estate funds, or real estate partnerships [34]. In this approach, investors contribute capital to the investment vehicle, which then pools funds from multiple investors to invest in a portfolio of properties. Indirect investments provide investors with an opportunity to participate in the real estate market without the need for direct property ownership or management responsibilities [35]. Investors in indirect real estate investments typically receive returns in the form of dividends, rental income, or capital appreciation based on the performance of the overall portfolio [34]. This form of investment offers diversification benefits as investors gain exposure to a wider range of properties and property types, potentially reducing risk compared to a single direct investment [35]. Indirect investments also provide liquidity as investors can buy or sell shares of REITs or units of real estate funds on stock exchanges or through secondary markets [31].

This research focuses on REITs as a kind of real estate investment. The reason for this choice is that they provide the opportunity to diversify an investment portfolio, and at the same time, they might be seen as affordable by most investors (both institutional and retail). By analysing the performance of REITs, this study aims at providing insights into how REITs can contribute to a well-diversified investment portfolio while also being accessible and affordable to a wide range of investors.

According to a definition provided by [35], REITs are investment vehicles that allow individuals to invest in real estate without the need for direct property ownership or management. REITs function as publicly traded companies or trusts that pool capital from multiple investors to acquire, develop, and manage a diversified portfolio of income-generating real estate properties.

Investing in REITs offers several benefits. Firstly, REITs provide a liquid investment option as their shares are traded on stock exchanges, enabling investors to easily buy or sell their holdings [35]. This liquidity makes it convenient for investors to access their capital when needed. Secondly, REITs offer a way to diversify real estate investments across different property types and geographic locations [36]. By investing in a REIT, individuals can gain exposure to a broad range of real estate assets, reducing risk through diversification.

One key aspect of REITs is their requirement to distribute a significant portion of their taxable income to shareholders [35]. This mandatory distribution is advantageous for investors, as it typically results in regular income in the form of dividends. Furthermore, REITs can provide attractive dividend yields, making them an appealing investment option for income-oriented investors.

REITs can focus on various property sectors, including residential, commercial, industrial, or specialised segments such as healthcare or hospitality [35]. Each REIT may have a specific investment strategy and property focus, allowing investors to choose REITs that align with their investment preferences and goals.

It is important for investors to carefully evaluate REITs before investing, considering factors such as the REIT's track record, management expertise, portfolio quality, and financial performance. Additionally, investors should be mindful of the potential risks associated with REIT investments, including fluctuations in real estate markets, interest rate changes, and general market volatility.

2.2.2 The Stock Market

Stock markets are centralised platforms where the buying and selling of shares of publicly traded companies occur. They provide investors with opportunities to participate in the ownership of companies and benefit from their financial performance and growth. Stock markets facilitate the trading of stocks, also known as equities or shares, which represent ownership interests in businesses [37]. Examples of prominent stock exchanges include the New York Stock Exchange (NYSE) and the NASDAQ [38].

Several types of market participants engage in stock market activities. Individual investors, such as retail traders, buy and sell stocks directly through brokerage accounts. Institutional investors, including mutual funds, pension funds, and hedge funds, manage large amounts of money on behalf of their clients and often have significant influence on stock prices. Market makers, typically brokerage firms, facilitate trading by providing liquidity in the market [39].

One of the primary functions of stock markets is to enable companies to raise capital for expansion and investment. By issuing shares to the public through initial public offerings (IPOs) or subsequent offerings, companies can access funding from investors who are willing to purchase these shares [40]. This capital injection allows companies to finance projects, research and development, acquisitions, and other activities that drive growth.

Investors participate in stock markets with various objectives. Some seek long-term capital appreciation by investing in stocks they believe will increase in value over time, while others focus on generating regular income through dividends paid by profitable companies [41]. Additionally, stock markets provide opportunities for traders who aim to profit from short-term price fluctuations, employing strategies such as day trading or technical analysis.

Various trading strategies are employed by market participants to generate profits. Day trading involves executing multiple trades within a day to take advantage of short-term price fluctuations. Value investing focuses on identifying undervalued stocks with the potential for long-term growth. Momentum trading aims to profit from the continuation of trends in stock prices. Arbitrage involves taking advantage of price discrepancies between different markets or securities [42].

Stock markets are subject to various factors that influence their dynamics and performance. Economic indicators, geopolitical events, industry trends, and company-specific news can significantly impact stock prices. Market participants analyse financial statements, company performance metrics, and macroeconomic data to make informed investment decisions [43].

Trading in stock markets takes place on organised exchanges, such as the New York Stock Exchange (NYSE) or the NASDAQ, where buyers and sellers come together to execute trades. The advent of electronic trading has revolutionised stock markets, allowing for faster and more efficient transactions [44].

Investors can choose to invest in individual stocks or opt for diversified exposure through mutual funds, exchange-traded funds (ETFs), or index funds. These investment vehicles pool funds from multiple investors and allocate them to a diversified portfolio of stocks, providing broader market exposure and risk mitigation [41].

It is important for investors to carefully evaluate their investment objectives, risk tolerance, and time horizon when participating in stock markets. While stock market investments offer potential rewards, they also come with inherent risks, including price volatility, market fluctuations, and the possibility of losing invested capital [37]. Conducting thorough research, diversifying investments, and staying informed are crucial for navigating stock markets effectively.

Several factors influence stock prices, making them fluctuate over time. Economic indicators, such as GDP growth, interest rates, and inflation, have a significant impact on stock prices as they reflect the overall health of the economy. Company-specific factors, including earnings reports, product launches, and management changes, can lead to substantial price movements. Market sentiment, influenced by news, investor behaviour, and market psychology, also affects stock prices [45].

2.2.3 The Bond Market

Bond markets are platforms where investors buy and sell bonds, which are fixed income securities. Bonds are debt instruments issued by governments, municipalities, and corporations to raise capital. They typically have a specified maturity date and pay periodic interest payments to bondholders. The bond market enables investors to diversify their portfolios and provides issuers with a means to borrow funds [46].

There are various types of bonds available in the bond market. Government bonds, such as U.S. Treasury bonds, are issued by national governments and are considered low-risk investments. Municipal bonds are issued by state and local governments to fund public projects, and they offer tax advantages to investors. Corporate bonds are issued by companies to raise capital and can vary in terms of credit quality and risk. Other types include mortgage-backed securities and high-yield bonds, which carry higher risks but potentially higher returns [47].

The yield curve represents the relationship between bond yields and their respective maturities. It plots the interest rates (yields) of bonds with similar credit quality against their time to maturity. The yield curve can be upward sloping (normal), downward sloping (inverted), or flat, indicating different market expectations and economic conditions. Yield curve analysis provides insights into market expectations for future interest rates and economic growth [48].

Bond pricing involves determining the fair value of a bond based on its characteristics and prevailing market conditions. The price of a bond is influenced by factors such as its coupon rate, time to maturity, prevailing interest rates, and credit risk. Bond prices and yields have an inverse relationship, meaning when interest rates rise, bond prices generally fall, and vice versa. Bond pricing models, such as the present value model, take into account these factors to calculate the bond's value [49].

Fixed income investments, including bonds, are subject to several risk factors. Credit risk refers to the risk of default by the issuer, where the bondholder may not receive full interest payments or principal repayment. Interest rate risk arises from changes in market interest rates, impacting the bond's price. Liquidity risk relates to the ease of buying or selling bonds in the market. Other risks include inflation risk, currency risk (for international bonds), and call risk (when bonds are callable before maturity) [50].

2.3 Modern Portfolio Theory

Modern portfolio theory (MPT) is a framework for building and managing investment portfolios, based on the idea that investors can minimise risk for a given level of expected return by diversifying their investments across a range of asset classes [51]. The theory suggests that an investor can minimise risk by spreading their investments across different asset classes, such as stocks, bonds, and real estate, rather than investing in a single asset class. MPT uses the mean-variance analysis to measure risk and return [52].

MPT assumes that investors are rational and risk-averse, meaning that they prefer less risk for a given level of return [53]. This assumption reflects the belief that investors are logical decision-makers who carefully evaluate the risk and return characteristics of different investment opportunities. By incorporating this assumption, MPT aims to provide a framework that aligns with the preferences and behaviours of rational investors.

The theory has been widely used in the investment industry for portfolio construction and management. Its emphasis on diversification and the efficient frontier has helped investors optimise their portfolios by achieving an optimal balance between risk and return. MPT's systematic approach to portfolio construction has provided investors with a structured method for making investment decisions and managing their assets.

However, MPT has also faced criticism for its underlying assumptions. One of the key critiques is that MPT assumes that returns follow a normal distribution. In reality, financial markets are known to exhibit characteristics such as fat tails, skewness, and volatility clustering, which challenge the assumption of normality. This limitation implies that MPT may not fully capture the extreme events and risks that can occur in the financial markets.

Additionally, the assumption of rationality and risk aversion has been questioned [54]. Some researchers argue that investors may not always behave rationally and that their risk preferences can vary depending on individual circumstances and market conditions. This critique highlights the limitations of MPT in capturing the complexities of human behaviour and emotions in the investment decision-making process.

Despite these criticisms, MPT has made significant contributions to the field of portfolio management. Its concepts and techniques have laid the foundation for modern portfolio construction and risk management practices [55]. Over time, researchers have proposed modifications and extensions to address the limitations of MPT, such as incorporating alternative risk measures and considering non-normal return distributions. These efforts continue to enhance our understanding of portfolio management and contribute to the development of more robust investment strategies.

In MPT, the two key factors used for making investment decisions are the expected portfolio return and the expected portfolio risk. The expected return of an asset is the average return that an investor expects to receive from that asset over a specific period of time. The expected return of a portfolio is calculated by weighing the potential return of each asset in the portfolio by the percentage of the portfolio invested in each asset [51].,

The formula for calculating the expected return of a portfolio can be represented as:

$$E[R_p] = \sum_{i=1}^{n} w_i E[R_i] \tag{2.1}$$

where: $E[R_p]$ is the expected return of the portfolio; n is the number of assets in the portfolio; w_i is the weight of the ith asset in the portfolio; and $E[R_i]$ is the expected return of the ith asset.

In addition to the expected return, MPT also considers the expected portfolio risk which plays a crucial role in determining optimal investment choices. It represents the uncertainty or potential variability of investment returns, and is commonly measured by the standard deviation of asset or portfolio returns, which estimates the dispersion of possible outcomes. A lower standard deviation suggests lower risk, while a higher standard deviation implies greater potential volatility.

The formula for calculating the expected risk of a portfolio can be represented as:

$$\sigma_p = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \sigma_i \sigma_j \rho_{i,j}} \tag{2.2}$$

where: σ_p is the expected risk (standard deviation) of the portfolio; w_j is the weight of the jth asset in the portfolio; σ_i is the standard deviation of the ith asset; σ_j is the standard deviation of the jth asset; and $\rho_{i,j}$ is the correlation coefficient between assets i and j.

In MPT, correlations between assets are important in determining the optimal portfolio for an investor. Correlation is a measure of the strength and direction of the linear relationship between two variables, and in MPT, it is used to measure the degree to which the returns of two assets move together [51].

The formula for calculating the correlation between two assets can be represented as:

$$\rho_{i,j} = \frac{cov(R_i, R_j)}{\sigma_i \sigma_j}, \tag{2.3}$$

where $cov(R_i, R_j)$ is the covariance between the returns of assets i and j.

In MPT, assets with a high correlation are featured by returns that generally move in the same direction, while those with low correlation show returns that tend to move independently of each other. Diversification in MPT is made possible by investing in assets with low correlations to mitigate portfolio risk. In other words, investors might reduce the overall risk exposure of their investments by constructing a diversified portfolio with assets that are not highly correlated.

The Sharpe ratio is a key metric in MPT that evaluates the risk-adjusted return of a portfolio. It is calculated as:

$$\text{Sharpe Ratio} = \frac{E[R_p] - R_f}{\sigma_p} \tag{2.4}$$

where: $E[R_p]$ is the expected return of the portfolio; R_f is the risk-free rate; and σ_p is the standard deviation of the portfolio returns.

A higher Sharpe ratio indicates better risk-adjusted performance, suggesting that the portfolio's returns adequately compensate for the risk taken. By integrating the expected Sharpe ratio into MPT analysis, investors aim to construct portfolios that maximise the Sharpe ratio, thereby achieving optimal risk-adjusted returns.

In conclusion, MPT assumes that an optimal portfolio can be built using information including the expected return, risk, and correlations between assets. The Sharpe ratio complements these metrics by providing a measure of risk-adjusted performance, helping investors assess and optimise their investment decisions. The goal of an investor is to maximise the expected return for a given level of risk, or to minimise the expected risk for a given level of return. The overall level of correlation between assets included in a portfolio determines the level of diversification, and thus the level of risk of an investment portfolio.

2.4 Machine Learning

In this Section, we present the optimisation algorithm used in this research (Section 2.4.1), and the regression algorithms (Section 2.4.2). Since the following chapters will deal with the portfolio optimisation problem first, and the regression problem later, we follow the same order in this section.

2.4.1 ML for Optimisation

To solve our optimisation problem, we use a specific kind of evolutionary algorithm known as genetic algorithm. Genetic algorithms (GAs) are bio-inspired algorithms that try to replicate an evolutionary process to solve optimisation problems [13, 56, 57]. A GA operates on a population of candidate solutions (individuals), and transforms the initial population using genetic operators, that create new offspring individuals through a stochastic selection process based on a fitness function, which measures the quality of the candidate solution. The fact that GAs perform a search on a set of all possible solutions, rather than a unique candidate solution, makes them suitable for complex optimisation problems (e.g., optimising investment weights in a portfolio).

Representation The representation of individuals in GAs depends on the problem that one tries to solve. Generally, individuals can be represented as either binary or numeric values. The position of each individual in a population is known as *gene*, and represents a variable to be optimised. At the initial stage of a GA, the population is composed of random individuals: the position of each individual is assigned randomly.

Genetic Operators Genetic operators are used to transform the initial population to generate new offspring individuals that are of higher quality with respect to the initial individuals. A commonly used genetic operator is crossover, that combines genetic material (or genes) of two parent individuals to generate new offspring individuals. For instance, one-point crossover swaps genes that lie on the right of a point picked randomly, known as *crossover point*[1]. After this process, there are two offspring individuals, each carrying some genes from both parent individuals. In addition to crossover, GAs usually employ a mutation operator, which creates new offspring individuals by transforming a single parent. For example, in one-point mutation, a single point can be mutated with a probability known as *mutation rate*.

Elitism and Selection Selection is the phase of a GA in which individuals are selected from an initial population for later transformation. One of the most popular methods for selection is known as *elitism*, in which part of the population is selected to be part of the next generation population based on the fitness value. In that way, the solution fitness does not decrease from one generation to the other. The remaining individuals are then subject to a probabilistic

1. The main reason for choosing the one-point crossover methodology is to minimise the divergence of optimal solutions. In a realistic financial portfolio, assets typically have similar allocations, so it is essential to keep extreme weights to a minimum. To achieve this, we adopt a controlled approach by slowing down the optimisation process.

selection for inclusion in the next generation. One of the most common ways of selection is known as *tournament selection*, in which k individuals are selected randomly, where k denotes the tournament size. From these k individuals, the best performing individual is chosen to advance to the next generation.

2.4.2 ML for Regression

Supervised machine learning, or supervised learning, refers to machine learning algorithms that use labelled datasets to learn a task that can be classifying data or predicting outcomes. Once the algorithm has been trained on a dataset (called *training set*), the obtained mathematical model is used to predict new data on an unseen dataset (called *testing set*). The main types of supervised learning algorithms are known as *regression* and *classification* algorithms. Classification trains an algorithm to assign test data into specific categories. It attempts to recognise some data, and draw conclusions on how those data should be labelled. Some common classification algorithms are support vector machines, random forest, and decision trees. Regression aims to understand the relationship between dependent and independent variables. It is commonly used to make predictions, for example on the volume of sales for a business or on the future stock prices. In this research, our primary emphasis is on regression algorithms. The following sections will provide an in-depth exploration of the following algorithms: Ordinary Least Squares Linear Regression (LR), Support Vector Regression (SVR), k-Nearest Neighbours Regression (KNN), Extreme Gradient Boosting (XGBoost), and Long/Short-Term Memory Neural Networks (LSTM).

Linear Regression Linear Regression (LR) is an algorithm that describes the relationship between a dependent variable and one or more independent variables. In the case of one independent variable, the algorithm is called *simple linear regression*; when there is more than one independent variable, the algorithm is called *multiple linear regression*. This is different from *multivariate linear regression*, which involves multiple correlated dependent variables.

In LR, the relationship between dependent and independent variables is modelled using linear predictor functions. The model parameters are estimated using an observed dataset in order to predict the values for the dependent variable. This usually happens using the least squares approach, which minimises the sum of residuals, or distances between actual and predicted values for the dependent variable.

Given a dataset $\{y_i, x_{i1}, \dots, x_{ip}\}_{i=1}^{n}$ of n statistical units, a linear regression model assumes that there is a linear relationship between the dependent variable y and a set of p independent variables x. That model takes into account a disturbance term or error variable ε, which denotes an unobserved random variable that adds 'noise' to the linear model. Thus, the model takes the form

$$y_i = \sum_{i=1}^{p} \beta_i x_i + \varepsilon \tag{2.5}$$

where β_i denotes the parameters of the model, x_i refers to the set of independent variables, and ε indicates the error variable.

Support Vector Regression The Support Vector Regression (SVR) algorithm attempts to minimise the error inside a certain threshold, or in other words, to estimate the best value for a variable within a given margin represented by ε.

In the context of Support Vector Regression (SVR), different kernel functions are used to transform data points into a higher-dimensional space. Each kernel function has specific formulations and parameters that influence its performance: Gaussian; Polynomial; and Radial Basis Function (RBF).

The Gaussian kernel is formulated as:

$$K_G(x_i, x_j, \theta) = \exp\left(-\sum_{k=1}^{N_D} \theta_k |x_k^i - x_k^j|^2\right)$$

where θ is a vector controlling the shape of the Gaussian kernel.

This kernel maps data points into a higher-dimensional space using a Gaussian function. The distance between points x_i and x_j in this space determines their similarity.

The polynomial kernel is given by:

$$K(x,y) = \tanh\left(\gamma x^T y + c\right)^d$$

where $\gamma > 0$ is the kernel coefficient, c is an optional constant, and d is the degree of the polynomial.

This kernel computes the similarity between points based on the polynomial transformation of their inner product $x^T y$. The γ parameter controls the influence of training instances on the model's predictions.

The RBF kernel is defined as:

$$K(x,y) = \exp^{-}(\gamma |x-y|^2)$$

where γ is the kernel coefficient.

The RBF kernel measures the similarity between points based on the Euclidean distance $|x-y|$. It is versatile in capturing complex relationships between data points.

The sigmoid kernel is expressed as:

$$K(x,y) = \tanh(\gamma x^T y + r)$$

where γ controls the influence of input variables and r is an optional constant.

The sigmoid kernel maps data into a higher-dimensional space using a hyperbolic tangent function. It is effective for non-linear classification tasks. The inputs x and y represent data points in the original feature space, while the output $K(x,y)$ denotes the computed similarity (kernel value) between input data points x and y after transformation into a higher-dimensional space induced by the kernel function.

Each of those functions has hyperparameters that need to be tuned to improve the model's performance and its generalisation capability to unseen data, including γ, c, d, and θ. For instance, the γ parameter measures the influence of a training instance on the prediction ability of a model: lower values for γ result in models with lower accuracy (or high error), and the same occurs with higher values for γ. It is only in the case of intermediate values for γ that lead to models with good decision boundaries.

K-Nearest Neighbor The K-Nearest Neighbor (KNN) algorithm identifies a specified number k of observations in the training dataset that are closest to a given observation x. It measures the distance between x and each observation in the dataset using different distance functions:

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Euclidean distance calculates the straight-line distance between x and y, considering their coordinates across multiple dimensions.

$$\text{Manhattan Distance} = \sum_{i=1}^{k}|x_i - y_i|$$

Manhattan distance measures the distance between x and y along axes at right angles, summing the absolute differences between corresponding coordinates.

$$\text{Minkowski Distance} = \left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$

Minkowski distance is a generalised form incorporating both Euclidean (when $q = 2$) and Manhattan (when $q = 1$) distances. Parameter q determines the order of the Minkowski distance.

These distance metrics are critical in KNN for determining proximity between data points, influencing how the algorithm identifies the k nearest neighbours to make predictions or classifications based on their labels or values. The choice of distance function depends on the data characteristics and the specific problem context, as each metric captures different aspects of similarity or dissimilarity between points.

Extreme Gradient Boosting Extreme Gradient Boosting (XGBoost) is a scalable, distributed gradient-boosted decision tree machine learning library. Gradient boosting is a machine learning algorithm applied to regression, classification, and ranking problems. XGBoost algorithm has gained popularity in the field of applied machine learning due to its ability to converge to the optimal solution using a lower number of iterations. It is often preferred than other gradient boosting algorithms due to to its shorter execution time, and better performance [58].

The XGBoost library can be used on different environments (e.g., Python, R, Java, C++, command line interface, etc.). It includes parallel computation to build trees using all the CPUs during training. Instead of traditional stopping criteria, it uses the ‘max depth’ parameter. This can potentially increase the computational performance with respect to the other gradient boosting algorithms. In addition, the XGBoost algorithm is designed to avoid overfitting through the regularisation term.

The main parameters for the XGBoost algorithm are represented by the maximum depth and the number of trees. The maximum depth of a tree measures the complexity of the resulting model, and thus, the likelihood of overfitting. A null value for that parameter indicates the absence of depth in the tree, while higher values make the tree deeper and more time-consuming. Other parameters include learning rate, minimum child weight, and number of boost rounds. The learning rate indicates the effect of each new decision tree on the previous prediction. The minimum child weight determines whether to split a note in a tree. The number of boost rounds refers to the number of decision trees trained. The optimal parameter values are decided on the basis on the loss function through an optimal solution search represented below.

$$Obj^{(t)} = \sum_{i=1}^{t} L(y_i, \hat{y}_i) + \sum_{i=1}^{t} \Omega(f_i), \tag{2.6}$$

where y_i is the observed value, $\hat{y}_i$ is the predicted value, $L(y_i, \hat{y}_i)$ is the loss function, and $\Omega(f_i)$ is the regularisation term, which helps to prevent overfitting by penalizing the complexity of each tree f_i.

Long Short Term Memory Long Short Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) designed to capture long-term dependencies in sequential data. It consists of several key components: a cell state C_t, an input gate i_t, a forget gate f_t, and an output gate o_t.

The input gate i_t determines which values from the input x_t and the previous hidden state h_{t-1} are used to update the cell state C_t:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \tag{2.7}$$

where σ denotes the sigmoid function, W_i is the weight matrix for the input gate, and b_i is the bias.

The cell state C_t is updated using the hyperbolic tangent function $\tanh$:

$$C_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \tag{2.8}$$

where $\tanh$ denotes the hyperbolic tangent function, W_c is the weight matrix for the cell state update, and b_c is the bias.

The forget gate f_t controls what information to discard from the cell state C_{t-1}:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \tag{2.9}$$

where σ denotes the sigmoid function, W_f is the weight matrix for the forget gate, and b_f is the bias.

The output gate o_t regulates the information that is passed to the output h_t:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{2.10}$$

where σ denotes the sigmoid function, W_o is the weight matrix for the output gate, and b_o is the bias.

The output h_t is computed by combining the cell state C_t weighted by o_t using the hyperbolic tangent function $\tanh$:

$$h_t = o_t \cdot \tanh(C_t). \tag{2.11}$$

In summary, C_t represents the cell state, i_t, f_t, and o_t denote the input, forget, and output gates respectively, and σ and $\tanh$ are the sigmoid and hyperbolic tangent functions, respectively. W_i, W_c, W_f, and W_o are weight matrices associated with each gate, and b_i, b_c, b_f, and b_o are their respective biases.

Chapter 3

Literature Review

3.1 Introduction

This thesis aims to evaluate the additional value brought by real estate investments in a mixed-asset portfolio comprising stocks and bonds. In contrast to existing literature, which predominantly relies on historical data for optimising portfolios with real estate, this study employs price predictions for these three asset classes. This chapter provides a review of the literature about portfolio optimisation techniques involving stocks, bonds, and real estate (Section 3.2), and financial forecasting for each of those asset classes (Section 3.3). The aim of this chapter is to analyse the amount of work that has been done on various portfolio optimisation techniques - i.e., linear programming (LP), quadratic programming (QP), nonlinear programming (NLP), stochastic programming, and genetic algorithm (GA) - on one side; and the financial forecasting techniques - i.e., time series models, econometric methods, and machine learning algorithms - on the other side.

3.2 Portfolio Optimisation

In this section, we explore the techniques that have been utilised for portfolio optimisation in the case of stock, bond, and real estate investments. Specifically, we will analyse the mathematical models used in solving portfolio optimisation problems (Section 3.2.1). Since the focus of this research is on investment portfolios including real estate, the second part of this paragraph will explain the current literature about the portfolio optimisation techniques in the case of real estate investments 3.2.2.

3.2.1 Portfolio Optimisation Techniques

The mathematical models used for portfolio optimisation include: linear programming (LP), quadratic programming (QP), nonlinear programming (NLP), stochastic programming, and the genetic algorithm (GA). In the following paragraphs, we describe these models in detail. Moreover, this section explores their use in relation to the real estate asset class (which is the focus of this research).

Linear Programming (LP). LP is a mathematical optimisation technique used in portfolio optimisation to determine the optimal allocation of assets within a portfolio. It involves formulating the problem as a linear objective function, which seeks to either minimise or maximise a certain outcome, such as the portfolio's overall return or risk-adjusted return. This objective function is subject to a set of linear constraints that reflect various limitations and requirements. In their study, [59] proposed a robust LP model for portfolio optimisation under uncertain conditions, incorporating risk-return analysis to enhance stability and performance. This approach addresses the challenge of managing portfolios in volatile markets by integrating uncertainty directly into the optimisation process. Similarly, [60] employed a linear objective optimisation approach to mitigate investment portfolio risk. Their study highlights how LP can be employed to systematically reduce risk exposure while maintaining targeted levels of return. In another work, [61] compared an LP-based portfolio with a benchmark portfolio, and demonstrated that the former outperformed the latter in terms of Sharpe ratio. This finding underscores the effectiveness of LP in achieving superior risk-adjusted returns compared to traditional investment strategies. In a different application, [62] applied LP to address an asset allocation problem in the case of real estate investments. Their study shows how LP techniques can be tailored to optimise allocation within specific asset classes. In summary, these studies confirm the effectiveness of LP in portfolio optimisation by leveraging mathematical optimisation techniques to enhance decision-making processes under uncertainties.

Quadratic Programming (QP). QP extends LP by allowing quadratic objective functions and linear constraints. QP can capture additional portfolio objectives or constraints, such as transaction costs or tracking error minimisation, in addition to the mean-variance trade-off. In their study, [63] proposed a parallel variable neighbourhood search algorithm combined with quadratic programming to solve cardinality-constrained portfolio optimisation problems. Their approach demonstrated effective optimisation capabilities in balancing portfolio constraints and objectives. Similarly, [64] adopted a double roulette wheel selection along with QP to solve portfolio optimisation problems under cardinality constraints. Their algorithm achieved superior accuracy and computational efficiency compared to state-of-the-art methods. Other studies, such as [65], used QP to model the cardinality constraints of a portfolio optimisation problem, resulting in reduced computational time and improved evaluation metrics. [66, 67] used QP in solving a portfolio optimisation problem involving real estate.

Nonlinear Programming (NLP). Nonlinear programming is a mathematical optimisation technique used to solve optimisation problems featured by nonlinear objective functions and/or nonlinear constraints. In the context of portfolio optimisation, NLP plays a crucial role in handling complexities due to relationships between asset weights and portfolio performance metrics not being linear. A work conducted by [68] employed NLP to solve a portfolio optimisation problem under transaction costs, aiming to find optimal asset allocations that minimise transaction expenses while maximising portfolio returns. In that way, it demonstrates the importance of incorporating transaction costs into the optimisation framework to achieve more realistic and efficient portfolio management strategies. On the other side, [69] attempted to address the complexities of the investment market by introducing a single-objective mixed-integer nonlinear programming model for fuzzy portfolio selection. They demonstrated that NLP techniques may be able to optimise portfolios under uncertain and dynamic market conditions. Overall, these studies highlight the versatility and effectiveness of NLP in portfolio optimisation, showcasing its ability to handle nonlinear relationships and complex constraints to enhance investment decision-making processes.

Stochastic Programming. Stochastic programming is a mathematical optimisation technique used for portfolio optimisation that incorporates uncertainty into the decision-making process. The objective in stochastic programming is to determine the optimal allocation of assets that maximises a certain performance measure (e.g., expected return, risk-adjusted return) while considering the probabilities of various future scenarios. It seeks to balance the trade-off between expected return and risk under uncertain market conditions. In their study, [70] focused on multi-period portfolio optimisation, and incorporated the expected return, Conditional Value at Risk (CVaR), and liquidity criteria using stochastic programming. Their approach modelled the stochastic nature of market movements to enhance portfolio decision-making. Another work conducted by [71] investigated a multi-period, stochastic portfolio optimisation model for diversified fund selection. It highlighted the application of stochastic programming in managing portfolio investments over time under uncertain market conditions. Furthermore, [72] proposed a scenario-based, multi-stage stochastic programming model to deal with multi-period portfolio optimisation problems with cardinality constraints and proportional transaction costs. Their approach aimed to optimise portfolio decisions across multiple stages while considering uncertainty and practical constraints. Another study performed by [73] adopted a stochastic programming model for financial optimisation in the real estate asset class. It demonstrated the application of stochastic programming in addressing complex financial optimisation problems that might be related to specific financial markets such as real estate.

Genetic Algorithm (GA). A GA is a computational optimisation technique inspired by the principles of natural selection and genetics. It is widely used in portfolio optimisation to find an optimal asset allocation that maximises predefined objective functions such as risk-adjusted return or portfolio diversification. In their study, [74] addressed portfolio selection under uncertain economic conditions using an enhanced genetic algorithm, demonstrating its effectiveness in handling stochastic variables like stock returns. On the other side, [75] employed a multi-objective genetic algorithm to solve portfolio optimisation problems in stock investments. Another work conducted by [76] focused on group stock portfolio optimisation using a genetic algorithm approach, showing improvements in portfolio performance metrics. Similarly, [77, 78, 79] proposed genetic algorithm-based methods for stock investment portfolio optimisation, highlighting enhanced portfolio returns and risk management strategies.

GA has also been applied to portfolio optimisation involving real estate investments. For instance, [80] developed a GA model for constructing investment portfolios that include real estate, emphasising risk reduction under uncertain conditions and achieving stable returns. In another study, [81] optimised mixed-asset portfolios, including real estate, using historical market data with a GA-based approach, demonstrating effective asset diversification and improved performance. These studies indicate that GA effectively optimises portfolios across different asset classes.

However, research on incorporating real estate investments, particularly real estate investment trusts (REITs), into portfolio optimisation via genetic algorithms remains limited. Real estate assets have unique risk-return profiles and correlations compared to stocks and bonds, offering potential benefits in terms of diversification and risk-adjusted returns [82]. Our study aims to address this gap by exploring the integration of real estate, including REITs, within a genetic algorithm-based portfolio optimisation framework, considering their specific characteristics and correlations.

3.2.2 Real Estate Portfolio Optimisation

The literature review we present in this section aims to analyse the existing research on the inclusion of direct and securitised real estate in mixed-asset portfolios and its impact on portfolio performance. As we have seen in Chapter 2, the two main ways to gain exposure to real estate markets are known as direct real estate investments, and securitised real estate investments. In this section, we explore the benefits of each of these options, and their inclusion in the previous works in the literature.

Direct Real Estate. As mentioned in Chapter 2, direct real estate investments refer to the purchase and management of actual physical properties [83]. Numerous studies have demonstrated the substantial diversification benefits of including direct real estate in a portfolio comprising not only financial assets but also various alternative asset classes [84, 85, 86]. According to [87], direct real estate can act as a hedge against inflation due to its potential to generate income streams that increase with rising prices. Moreover, [88] argue that direct real estate investments can provide diversification benefits by showing low correlations with traditional asset classes such as stocks and bonds.

Despite the benefits of including direct real estate in a mixed-asset portfolio, certain challenges and considerations should be taken into account. Liquidity and transaction costs associated with direct real estate investments can pose obstacles for investors. According to [89], illiquidity in the direct real estate market can limit the ability to rebalance portfolios efficiently. Additionally, the unique characteristics of real estate, such as property management and maintenance, require active involvement and expertise, as noted by [90].

Securitised Real Estate. Securitised real estate refers to the process of converting real estate assets, such as properties or mortgages, into tradable securities, allowing investors to gain exposure to the real estate market without directly owning physical properties. Securitised real estate, such as real estate investment trusts (REITs) and real estate-related stocks, has gained popularity as an investment option within mixed-asset portfolios [91, 88].

Research has consistently highlighted the potential benefits of including REITs in a diversified investment portfolio. Firstly, studies indicate that REITs have historically shown low correlations with traditional asset classes such as stocks and bonds [9]. This low correlation suggests that including REITs in a portfolio can enhance diversification and potentially reduce overall portfolio risk. By introducing an asset class that behaves differently from others, investors can reduce their exposure to market fluctuations and potentially achieve a more stable risk-return profile. Furthermore, the addition of REITs to a mixed-asset portfolio has been associated with potential improvements in risk-adjusted returns [92]. Several studies [93, 91] have found that portfolios that include REITs tend to have higher risk-adjusted returns compared to portfolios that exclude REITs. This finding suggests that REITs may offer unique return characteristics that can enhance the overall performance of a mixed-asset portfolio [91, 94].

Although a few researchers have made attempts to predict REIT prices using machine learning algorithms, the number of such studies remains limited. For example, [16] utilised a neural network algorithm to predict both stock and REIT prices and demonstrated that this algorithm was more accurate than an autoregressive integrated moving average (ARIMA) model. Similarly, [17] used machine learning-based regression algorithms, including neural networks, to predict REIT returns. Other studies focused on comparing machine learning algorithms to ARIMA for REIT return prediction, primarily through the use of artificial neural networks and multiple variables, as noted by [10, 11, 12]. In summary, while a handful of studies have been conducted on REIT price prediction, most of them have centred around neural networks.

Several studies have been conducted to predict REIT prices using machine learning algorithms, and some have shown that these algorithms perform better than traditional models like autoregressive integrated moving average (ARIMA) in terms of prediction accuracy, as noted by [12, 10, 11]. While most of the current literature has concentrated on the use of artificial neural networks with multiple variables, our research aims to investigate other machine learning techniques for predicting REIT prices.

Prediction-Based Portfolios. Many studies in the literature explored the diversification potential that can be achieved through real estate investments [3, 4, 5]. Institutional investors have found that a significant allocation to real estate protects their wealth during difficult times, such as the Covid-19 pandemic [3]. However, direct investment in real estate assets can be expensive, so many investors choose indirect investment through real estate investment trusts (REITs), which are companies that own and manage real estate. REITs offer individual investors the opportunity to invest in real estate without the hassle of owning or managing properties. The low entry cost of REITs makes them an attractive option, with shares available for as little as $500 [1]. Additionally, REITs are highly liquid, like stocks, making them easier to buy and sell quickly compared to real estate properties that can take months to complete.

Investors in REITs who want to determine the best weight for each asset in their portfolio need to solve a portfolio optimisation problem. This problem involves two main steps: (i) creating a model that fits historical asset prices and predicts future values for a test set, and (ii) utilising the price predictions to allocate optimal weights to each asset via an optimisation algorithm that is based on a specific metric, such as risk or return. Another option is to perform the optimisation process directly on the training set, but this approach has drawbacks, as the weights may not be optimal for the test set if there are significant variations in prices.

Although the two-step approach for optimising mixed-asset portfolios has been utilised before, it has not yet been applied to portfolios that include REITs. Previous research that utilised portfolio optimisation with REITs relied on the optimal weights computed in the training set, as noted by [95, 96, 84]. Our research, on the other hand, emphasises the accurate prediction of REIT prices. Such a task is crucial since the prices are utilised as input in the portfolio optimisation step.

---

1. https://www.investopedia.com/articles/investing/072314/investing-real-estate-versus-reits.asp
Last access: September 2022.

Now the importance of using price prediction is demonstrated, Section 3.3 will explore the different financial forecasting techniques in the case of stocks, bonds, and real estate. Specifically, there will be a focus on the time series models, econometric approaches, and machine learning algorithms.

3.3 Financial Forecasting

Financial forecasting can be defined as the process of estimating future financial outcomes or variables based on historical data, economic indicators, and other relevant information. It involves the use of quantitative models, statistical techniques, and expert judgement to predict financial metrics such as stock prices, bond yields, interest rates, and real estate values.

Financial forecasting plays a crucial role in the decision-making processes of investors, financial analysts, and policymakers. Accurate predictions about the future performance of stocks, bonds, and real estate assets, including Real Estate Investment Trusts (REITs), are essential for making informed investment choices, risk management, and overall portfolio optimisation. This literature review aims to explore the key methodologies, challenges, and recent developments in financial forecasting for these asset classes.

In addition, we explore the literature about the use of technical analysis (TA) indicators as a tool to improve financial forecasting. We aim to show the amount of studies about TA-based forecasting for the different asset classes considered.

Stocks. Forecasting stock prices has been a topic of significant interest in financial literature. Time series analysis is a widely used methodology for forecasting stock prices. Early studies, such as the work by [97], demonstrated that stock prices follow a random walk pattern. However, subsequent research by [98] challenged this notion with the Efficient Market Hypothesis (EMH), suggesting that stock prices fully reflect all available information, making it impossible to consistently predict future prices.

Despite the challenges presented by EMH, researchers explored alternative methodologies for stock price prediction. Machine learning algorithms, such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN), gained popularity for their ability to capture nonlinear relationships in stock data [99, 100]. Moreover, sentiment analysis and natural language processing techniques were employed to predict stock prices based on news sentiment [101, 102]. Other studies demonstrated that machine learning algorithms might outperform econometric approaches in the price prediction problem [103, 104, 105].

Bonds. Financial forecasting for bonds has its unique set of challenges due to interest rate fluctuations, credit risk, and macroeconomic factors. Traditional bond valuation models like the Yield-to-Maturity (YTM) method are commonly used to forecast bond returns [106]. [107] proposed a model combining macroeconomic variables and machine learning techniques to improve the accuracy of bond yield predictions.

Another critical aspect in bond forecasting is credit risk assessment. Researchers have utilised credit rating data and credit default swap spreads as predictors for bond credit risk [108]. Additionally, time series models like the Autoregressive Integrated Moving Average (ARIMA) have been employed for short-term bond yield predictions [109].

Real Estate. Real estate forecasting involves predicting property prices, rental yields, and market trends. Traditional methods like hedonic pricing models have been used to predict property prices based on the property's characteristics [110]. Moreover, real estate analysts have applied spatial analysis to capture the geographical dependence of property prices [111].

Time series analysis is a widely used approach for real estate price forecasting. Research by [112] employed autoregressive integrated moving average (ARIMA) models to predict real estate prices, demonstrating the model's ability to capture temporal dependencies in price movements. Similarly, [113] utilised a vector autoregression (VAR) model to forecast real estate prices, incorporating relevant macroeconomic variables to improve accuracy.

Machine learning techniques have gained prominence in real estate price forecasting due to their ability to capture complex patterns and nonlinear relationships. [114] proposed a hybrid model combining a support vector machine (SVM) and a genetic algorithm (GA) to predict real estate prices, achieving superior forecasting accuracy. Additionally, [115] used a long short-term (LSTM) neural network to capture temporal dependencies and successfully forecasted real estate prices.

The rise of REITs as an investment vehicle in the real estate market has prompted research on forecasting their performance. Time series analysis has been widely applied to forecast various financial variables of REITs. [116] employed autoregressive integrated moving average (ARIMA) models to forecast the rental income and net operating income of REITs. The study demonstrated the usefulness of ARIMA models in capturing the underlying patterns and trends in REITs' financial data.

Machine learning techniques have gained popularity in REITs' financial forecasting due to their ability to capture complex patterns and nonlinear relationships. In their study, [117] used a random forest algorithm to predict the returns of REITs based on various financial and macroeconomic variables. Their results showed that the random forest model outperformed traditional linear models in forecasting REITs' returns. Furthermore, [16] showed that ML algorithms can outperform econometric models, including ARIMA, in the prediction of REITs.

Technical Analysis. *Technical analysis* (TA) is a widely used approach that involves the examination of historical price and volume data in financial markets to predict future price movements. Traders and investors rely on this methodology to gain insights for informed decision-making regarding the buying, selling, or holding of various financial assets, including stocks, currencies, and commodities [118].

One of the key principles of technical analysis is the belief that market prices follow trends and patterns, and that these trends can be identified and utilised for predictive purposes. Technical analysts utilise a wide range of tools and techniques to analyse market data, including chart patterns, technical indicators, and statistical models [119].

Chart patterns are visual representations of historical price movements that can provide insights into future price direction. Examples of commonly used chart patterns include head and shoulders, double tops, and triangles [120]. These patterns are often believed to indicate potential reversals or continuations in price trends.

Technical indicators are mathematical calculations based on historical price and volume data. They are used to generate trading signals and identify potential buying or selling opportunities. Some popular technical indicators include moving averages, relative strength index (RSI), and stochastic oscillators [121].

In addition to chart patterns and technical indicators, technical analysts also rely on statistical models to forecast future price movements. These models often involve the use of regression analysis, time series analysis, and other statistical techniques to identify relationships and trends in the data.

While technical analysis is widely used in financial markets, it is not without its critics. Some argue that it is based on subjective interpretations and lacks a solid theoretical foundation [122]. Others contend that it is a self-fulfilling prophecy, as the actions of market participants following technical analysis patterns can create the predicted price movements. Nevertheless, technical analysis continues to be popular among traders and investors, and numerous studies have explored its effectiveness, such as [123], where technical analysis indicators were used in combination with sentiment analysis and [124], where technical analysis was used alongside indicators derived from an event-based system.

In conclusion, technical analysis is a widely used methodology in financial markets that involves analysing historical price and volume data to predict future price movements. It employs chart patterns, technical indicators, and statistical models to identify trends and patterns in the data. While there are critics of technical analysis, studies have shown its potential effectiveness in certain market conditions. The fact that TA has yet not been incorporated in studies that predict REITs prices provides an opportunity to improve the accuracy of price prediction in this domain.

Chapter 4

Optimising Mixed-Asset Portfolios Including REITs

4.1 Problem Statement

In Chapter 3, we examined literature exploring the integration of real estate investments into mixed-asset portfolios. These studies primarily aimed to optimise returns and minimise portfolio risk by utilising historical data from stocks, bonds, and real estate. One of the potential limitations of this approach, known as the 'historical data' approach, is that it might not accurately reflect future market conditions, potentially leading to sub-optimal portfolio performance. To address this, the chapter aims to investigate the potential benefits of using price predictions in the portfolio optimisation process.

To conduct this investigation, the chapter employs a method where it assumes perfect price predictions in the test set, essentially adopting a hypothetical scenario where future prices are accurately known. By doing so, the optimisation of portfolio weights can be performed using this perfect foresight approach. The rationale behind this approach is to determine if incorporating price predictions significantly improves portfolio performance compared to using historical data alone.

If the results of this analysis show that the portfolio performance indeed improves with the inclusion of price predictions, it would provide a strong justification for further research into accurately predicting asset prices. In summary, the chapter aims to demonstrate the potential advantages of integrating forward-looking information into the portfolio optimisation process.

The rest of this chapter is organised as follows: Section 4.2 provides a description of the perfect foresight approach, and of the genetic algorithm used; Section 4.3 presents the experimental setup; and Section 4.4 provides a detailed discussion of the experimental results. Finally, Section 4.5 presents the main conclusions for this chapter.

4.2 Methodology

4.2.1 Data

In this research, we examine daily price time-series for different market proxies[1], serving as our datasets. These proxies are representative of three distinct asset classes: stocks, bonds, and REITs, spanning three diverse markets—namely, the United States (US), the United Kingdom (UK), and Australia (AU). To mitigate currency risk, all data is expressed in US dollars (USD). For a detailed breakdown of the actual data utilised in our experimental setup, including the precise number and characteristics, please refer to Section 4.3.1 later in this chapter.

Subsequently, each dataset undergoes further division into three sequential subsets in chronological order: a training set, utilised to train the machine learning model; a validation set, employed for optimising the model's hyperparameters; and a testing set, representing the unseen data used in the final evaluation stage following model tuning and training.

4.2.2 Portfolio Optimisation Under Perfect Foresight

As previously explained, our goal is to compare the perfect foresight approach in solving a portfolio optimisation problem against the historical method. For such purpose, we refer to three financial metrics, i.e., expected portfolio return, expected portfolio risk, and expected Sharpe ratio (see Section 2.3).

The methodology used in this work follows two steps. The first step consists of optimising asset weights using returns calculated on the test set. The second step consists of calculating the expected return, expected risk, and Sharpe ratio for all asset combinations.

Regarding the first step, a genetic algorithm (GA), detailed in Section 4.2.3, is employed on the test set. This approach is based on the assumption of possessing perfect foresight regarding future prices, thereby facilitating the optimisation of asset weights through the GA algorithm.

In the second step, we use the optimal weights obtained from the first phase to compute the expected return, expected risk, and Sharpe ratio of the GA runs. The hypothesis behind our experiments is that this portfolio optimisation strategy would result in better portfolio performance than in the case of optimal weights calculated on historical average returns.

1. Market proxies are representative indicators used to closely mimic the performance or behaviour of a broader financial market. They serve as a convenient way to track and analyse market trends, movements, and dynamics without directly involving the actual securities in the market. Market proxies might include indices, exchange-traded funds (ETFs), or other financial instruments that mirror the performance of a specific market or asset class

4.2.3 Portfolio Optimisation via a Genetic Algorithm

As explained before, *genetic algorithms* (GAs) offer a computational approach to address a portfolio optimisation problem by mimicking the principles of natural selection and evolution. This section explores the principal aspects of the genetic algorithm applied to portfolio optimisation, outlining the representation of individuals, the operators driving evolution, and the fitness function guiding the algorithm's decision-making process.

Representation GA *chromosomes* (or, *individuals*) consist of N *genes* indicating the weights allocated to the N assets in the portfolio. The weight are real numbers in the interval $[0,1]$, and their sum is equal to 1. For example, a GA individual that has the genotype [0.5 0.2 0.3] indicates that there are three assets, with corresponding weights assigned as 0.5, 0.2, and 0.3 to each asset. Initially, all genes are assigned the same weight (in particular, $W_i = 1/N$ for each asset i), which are then evolved according to a set of operators.

Operators We use *elitism*, *one-point crossover* and *one-point mutation*. Since we use market proxies in our experiments, the number of assets is small, and thus one-point crossover and mutation are sufficient (see Section 4.3 for more details). After the application of crossover and mutation, we apply normalisation to each GA individual, to ensure that the sum of weights remains equal to 1.

Fitness Function State-of-the-art methods for solving portfolio optimisation problems have used many different metrics as fitness functions. In this thesis, we use the *Sharpe ratio*, defined as the ratio of the difference between the average return[2] and the risk-free rate[3], over the standard deviation of the returns, that is,

$$S = \frac{r - r_f}{\sigma_r},$$

where r is the average return of the investment, r_f is the risk-free rate, and σ_r is the standard deviation of the returns.

2. The term 'return' in this context is used specifically to refer to the quantity $(N_t - N_{t-1})/N_{t-1}$, i.e. the relative rate of returns, which is the difference between an asset's normalised price difference on a particular day compared to the day before, expressed as a percentage of the latter.
3. The term 'risk-free rate' denotes the minimum return expected from an investment with zero risk of default, such as government bonds.

4.3 Experimental Setup

As explained in Section 4.1, the primary goal of the experiments is to demonstrate that optimising asset weights under the assumption of perfect foresight results in a better-performing portfolio compared to using historical data. To achieve this objective, we conduct a comparative analysis of the portfolio performance derived from perfect predictions against that obtained through the historical method. In the following sections, we provide an overview of the nature and source of data used in these experiments (Section 4.3.1), the hyperparameter tuning process (Section 4.3.2), and the benchmark chosen for evaluating the performance of the proposed method for portfolio optimisation (Section 4.3.3).

4.3.1 Data

We use daily prices over the period between June 2017 and January 2021. We adopt the perspective of an institutional investor from the US who wants to gain exposure to international markets (UK and Australia). The asset classes we consider are stocks, bonds, and listed real estate.

Our exploratory experiments utilise index price data as it is considered a reliable representation of market movements across different asset classes. The choice of using index data is in line with earlier studies, including those conducted by [125] and [93] which adopted market index data to solve an asset allocation problem in the context of real estate investments. Stocks are proxied by the S&P 500 index[4] for the US market, by the FTSE 100 index[5] for the UK market, and by the S&P/ASX 200 index[6] for the Australian market. For the bond asset class, we use the indices issued by Dow Jones for all the three markets considered. Finally, we use the FTSE/EPRA NAREIT indices to proxy the real estate markets. We thus have 9 asset classes, namely 3 stocks, 3 bonds, and 3 REITs.

To represent the statistical distribution about each dataset, Table 4.1 presents the main statistics for each asset class: the mean of the returns of the assets (as a proxy of their expected return), their standard deviation (as a proxy of their volatility or risk), and the Sharpe ratio (as a measure of the asset's risk-adjusted return). In order to calculate the Sharpe ratio, we have considered a risk-free rate equal to 1.90×10^{-3} (corresponding to an average of the daily government bond rates in the three countries). From the results shown in Table 4.1, we can observe that the real estate asset class generally presents a lower level of performance

4. https://www.spglobal.com/spdji/en/indices/equity/sp-500/#overview
5. https://www.londonstockexchange.com/indices/ftse-100
6. https://www.spglobal.com/spdji/en/indices/equity/sp-asx-200/#overview

Table 4.1: Mean, standard deviation, and Sharpe ratio for each asset class.

Asset Class	Mean	St Dev	SR
S&P 500	4.00×10^{-4}	8.60×10^{-3}	4.51×10^{-2}
FTSE 100	5.00×10^{-5}	1.05×10^{-2}	3.20×10^{-3}
S&P/ASX 200	1.10×10^{-4}	1.07×10^{-2}	8.60×10^{-3}
US bond	3.90×10^{-4}	8.60×10^{-3}	4.38×10^{-2}
UK bond	6.00×10^{-5}	1.06×10^{-2}	4.00×10^{-3}
AU bond	1.20×10^{-4}	1.10×10^{-2}	9.60×10^{-3}
US REIT	1.40×10^{-4}	9.60×10^{-3}	1.24×10^{-2}
UK REIT	6.00×10^{-5}	1.37×10^{-2}	2.80×10^{-3}
AU REIT	1.90×10^{-4}	1.16×10^{-2}	1.44×10^{-2}

(that is, lower Sharpe ratio) compared to the other asset classes. This indicates that real estate investments could be less profitable than the other types of investments if considered individually.[7] Our aim is to assess the added value that real estate could bring within a multi-asset portfolio.

From the correlation matrix shown in Figure 4.1, we can observe that the real estate asset class generally has relatively lower correlation with the other asset classes, thus justifying its diversification potential. More specifically, a low or zero correlation between two asset classes might reduce a portfolio's overall level of risk. Moreover, we observe a low correlation between asset classes belonging to different markets (e.g., S&P 500 and UK REITs). This could open opportunities to an international diversification. In other words, an investor might find diversification opportunities in gaining exposure to foreign markets.

4.3.2 Experimental Parameters

To decide the parameter values, we undertook a parameter tuning process using the I/F-Race package [126]. I/F-Race implements the iterated racing procedure, which is an extension of the Iterated F-Race process and builds upon the race package by [127]. This optimisation method automatically configure algorithms by evaluating and comparing multiple candidate parameter configurations over a set of instances. Its purpose is to progressively eliminate poor configurations while refining the best ones. The process continues iteratively until the most appropriate settings for the optimisation algorithm are identified.

7. The choice of datasets with different statistical distribution is crucial for our optimisation problem as our aim is to demonstrate the advantages of real estate in the context of a portfolio made of stocks and bonds as well. Employing more similar distributions, for example by selecting stocks only, might reduce evidence about the advantages of one asset class with respect to the others. The following chapters (i.e., 5-7) are based on the same approach as the different distribution characteristics of each asset class are key to our findings.

Figure 4.1: Correlation matrix between asset classes.

In I/F-Race, each iteration, or ‘race’, involves running the optimisation algorithm with different parameter settings on a set of problem instances. The parameter configurations that perform poorly (in predicting the dependent variable) are discarded, while the better-performing configurations advance to the next iteration. This iterative process continues until a stopping criterion is met, which in our case is when there is no significant improvement in performance. The advantage of I/F-Race lies in its ability to efficiently explore the parameter space and identify optimal or near-optimal configurations without requiring exhaustive search.

In our case, I/F-Race was applied to data for the period from January 2019 to December 2019. The following twelve months (January-December 2020) were used only with the already tuned parameters, after I/F-Race was completed. In other words, the first period was used as a training dataset for parameter tuning, while the second period was used as a validation dataset for parameter testing. The period January-July 2021 was the test set, and remained unseen during the parameter tuning process. At the end of the tuning process, we picked the best parameters returned by I/F-Race, which constitute the experimental parameters used by our algorithms, and are presented in Table 4.2. These parameter values were determined to be the most effective in optimising the performance of our algorithm on the training data.

Table 4.2: I-Race Parameter Tuning Results.

Parameter	Value
Tournament Size	3
Population Size	300
Mutation Rate	0.01
Number of Generations	10

It is worth noting that the choice of the parameter values was driven by the size and structure of the considered datasets. Given the relatively small number of datasets (i.e., nine in total), each corresponding to an asset class, and the number of data points (i.e., around 600 in total), each referring to a price on a trading day, the values represented in Table 4.2 were considered suitable for our problem. While the problem could theoretically be optimised using an exhaustive approach due to the small number of assets, the GA was chosen to explore its effectiveness in scenarios where exhaustive search may not be feasible.

Regarding the tournament size, we selected a small value (3) to apply moderate selection pressure, which accelerates convergence by favouring fitter individuals. Although small k values increase the risk of premature convergence, we mitigated this by: (i) maintaining diversity, through a low mutation rate (0.01); and (ii) exploring broadly, through a large population size (300) to ensure an extensive exploration of the search space.

Figure 4.2 represents how fitness evolves over generations. The x-axis corresponds to the number of generations. The y-axis shows the average fitness value (in terms of Sharpe ratio) across all individuals in the population for each generation. This plot is a critical tool for assessing the algorithm's performance, representing the convergence behavior as the algorithm optimises the portfolio over time.

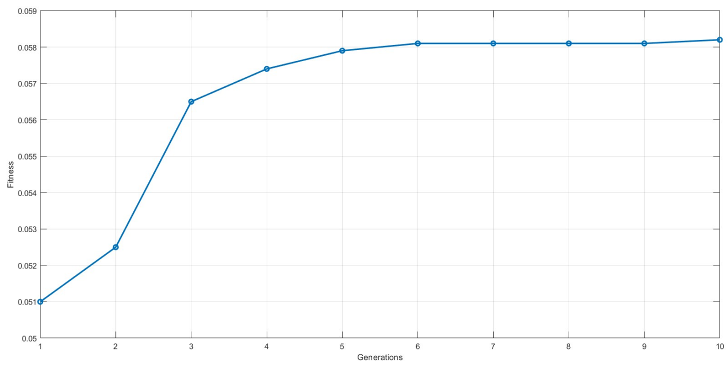


Figure 4.2: Fitness evolution over generations.

4.3.3 Benchmark: The Historical Data Approach

In order to demonstrate the potential improvement from the perfect foresight situation, we compare their results with results obtained from experiments under the historical method. In other words, we used the 2017-2019 period as the training set, where we ran the portfolio optimisation task. After the weights were obtained in the training set, we then applied them to the test set (2020-2021 period), and then compared the financial performance (Sharpe ratio, rate of return, risk) against the perfect foresight results. We again used a genetic algorithm for the portfolio optimisation task. The GA used the same parameters that were presented above in Table 4.2.

4.4 Results

In this section we present our experimental results for the genetic algorithm with the perfect foresight approach and compare it with the historical approach (Section 4.4.1, and discuss our findings (in Section 4.4.3). Results are presented as averages over 20 individual GA runs. It should also be noted that all results are daily results. So when, for example, we present a seemingly 'low' return of around 0.03%, its annual equivalent would be around 11.6%. [8]

4.4.1 Summary Statistics

Under the perfect foresight hypothesis, the value of the predicted price $\hat{P}_i$ is assumed to be equal to the value of the actual price P_i during the testing phase of the prediction process. Such assumption leads to a null error rate, implying that the predictions perfectly match the observed prices. We compare the portfolio performance results obtained from such model with those obtained from the historical data approach. As mentioned in Section 4.2, have we run the genetic algorithm on the test set for a portfolio including the nine asset classes considered.

The results obtained from our genetic algorithm are represented in Figure 4.3, which compares the distributions generated by the historical data method and the perfect foresight approach. Figure 4.3a specifically highlights the distribution of expected portfolio returns for both methods. We can observe that the perfect foresight approach results in an approximate 14% increase in the average return compared to the historical method. From a financial perspective, this indicates that portfolio profitability improves under perfect foresight. Additionally, the standard deviation of returns decreases by around 25%, suggesting a tighter concentration of return values around the mean, implying reduced volatility.

8. $\text{AnnualisedReturn} = [(\text{DailyReturn} + 1)^{365} - 1] \times 100 = 11.6\%$.

Regarding the other two statistical metrics (i.e., skewness and kurtosis), our findings demonstrate that both approaches exhibit a long left tail, indicated by negative skewness values, suggesting more concentration of values on the right side of the distribution, which usually indicates more positive returns higher than the mean. The historical approach shows higher skewness compared to the perfect foresight method. However, since the perfect foresight method has a higher mean value, this difference in skewness is not concerning. The historical approach exhibits a fatter left tail compared to the perfect foresight method, indicating fewer outliers. However, the difference in kurtosis between the two methods is relatively small (6%).

In summary, the perfect foresight method outperforms the historical approach in terms of higher average expected returns and lower standard deviation, indicating better portfolio profitability and reduced variability in returns. While both approaches exhibit similar skewness and kurtosis characteristics, the differences in these metrics are not alarming given the overall performance superiority of the perfect foresight method.

Figure 4.3b shows results for the average expected risks. In this case, under a perfect foresight situation, we observe a decrease in the average risk level of around 19%. This finding can be interpreted as an improvement in portfolio performance under a perfect foresight situation. The standard deviation values tend to be similar for both cases which indicates a similar level of concentration of risk values around the mean.

The skewness and kurtosis values for the perfect foresight method appear to be greater than those obtained from the historical approach. This implies a greater probability of observing risk values lower than the average and a lower presence of outliers for the perfect foresight method. A positive skewness indicates a longer right tail in the distribution, suggesting a higher probability of lower risk values. Additionally, the higher kurtosis indicates heavier tails and a sharper peak in the distribution, further indicating a lower presence of outliers and lower investment risk under the perfect foresight method from a financial perspective.

In summary, the perfect foresight method shows a decrease in average expected risks compared to the historical approach, indicating improved portfolio performance. Moreover, the skewness and kurtosis values suggest a lower investment risk under the perfect foresight method due to a greater probability of lower risk values and a lower presence of outliers in the risk distribution.

Figure 4.3c shows results obtained for the average expected Sharpe ratios. Based on the presented results, the perfect foresight method outperforms the historical data approach in terms of risk-adjusted portfolio performance. The average Sharpe ratio is reported to increase by approximately 45%, implying an enhancement in risk-adjusted returns. Additionally, the standard deviation decreases by about 19%, suggesting that returns under the perfect foresight approach are more concentrated around the mean.

The skewness value is negative for both methods, suggesting a concentration of Sharpe ratio values that are higher than the average. Additionally, the skewness value increases by around 38% from the historical data method to the perfect foresight approach, indicating an even greater concentration of high Sharpe ratio values. The kurtosis value for the perfect foresight approach is higher than that obtained from the historical approach, with a difference of 127%. A higher kurtosis indicates a fatter tail in the distribution of Sharpe ratios, suggesting a lower presence of outliers under the perfect foresight method.

In summary, the perfect foresight method leads to higher average Sharpe ratios and lower standard deviation, skewness, and kurtosis values compared to the historical approach. These findings indicate improved risk-adjusted portfolio performance and a lower presence of outliers under the perfect foresight method.

(a) Return distributions **(b)** Risk distributions **(c)** Sharpe ratio distributions

Figure 4.3: Expected portfolio return, risk, and Sharpe ratio distributions

Overall, the perfect foresight method consistently demonstrates superior performance across various financial metrics, including expected returns, risk, and Sharpe ratios, indicating its effectiveness in enhancing portfolio performance and reducing investment risk compared to the historical approach.

To compare the distribution pairs (risks from perfect foresight method and risks from historical data approach) for the expected return, risk, and Sharpe ratio, we performed three Kolmogorov-Smirnov (KS) tests at the 5% significance level. Here, the null hypothesis for each test was that the two distributions come from the same probability distribution. Since we are making multiple comparisons, the-adjusted p-value is equal to $0.05/3 = 0.0167$, as we have again applied the Bonferroni correction. The p-values obtained from the three tests are all equal to 5.54×10^{-10} which is below the adjusted p-value of 0.0167, thus making the differences statistically significant at the 5% level.

However, given the size of the datasets, it might be worth considering the effect size measurements using Cohen's d for each comparison. It quantifies the difference between two means in terms of standard deviations, providing insight into the practical significance of the observed differences. For the expected return, the Cohen's d value was 0.45, indicating a moderate effect size. For risk, the Cohen's d was 0.32, suggesting a small to moderate effect size. Finally, for the Sharpe ratio, the Cohen's d was 0.60, demonstrating a moderate effect size. In conclusion, such effect measurements, in addition to the statistical significance indicated by the KS test p-values, reinforce the reliability of our findings.

4.4.2 Computational Times

A single run of the GA did not take longer than 30 seconds, under the parameter values presented in Table 4.2. As the portfolio optimisation task is an offline approach, this duration is relatively fast and does not constitute a problem. Besides, speedups can be obtained by parallelising the evolutionary process, as it has previously been shown in the literature, e.g. [128].

4.4.3 Discussion

The main goal of our experiments was to show the potential improvement in mixed-asset portfolio performance that can be obtained from hypothetically perfect price predictions compared to the historical data approach. As we have observed, the average portfolio returns appear to increase under a perfect foresight situation, and given the KS test results, such increases appear to be statistically significant. At the same time, the average portfolio risks appear to decrease when the perfect foresight case is applied, and based on the KS test results, such differences can be considered statistically significant. Such results lead to an improvement in the risk-adjusted portfolio performance.

4.5 Summary

The key points of this chapter can be summarised as follows.

The return rate from real estate investments tends to be lower compared to other asset classes. In Table 4.1, we noticed that the return rate deriving from real estate investments tends to be lower compared to other types of investments, such as stocks and bonds. This might indicate that it would be more convenient to consider real estate as part of a mixed-asset portfolio rather than a single investment choice. This is because real estate generally acts as a diversifier due to its lower risk.

The correlation between real estate and other asset classes is generally low. As we observed from the correlation matrix represented in Figure 4.1, the correlation values tend to be lower in the case of real estate compared to the other asset classes considered. This justifies the diversification potential achieved by adding real estate. Moreover, there tends to be a lower correlation between asset classes belonging to different countries. This might explain our choice of including investments from different countries in our portfolio.

Optimising a portfolio directly in the test set can lead to better risk-adjusted performance results compared to when the optimisation takes place in the training set. Our results show that using price predictions can lead to better risk-adjusted performance than when using historical data. This is mainly explained by the fact that prices in the training set might be significantly different than those in the testing set (as we demonstrated through the KS tests), thus leading to under-performing portfolios. The results that we obtained motivate us to engage in price prediction tasks in order to solve mixed-asset portfolio optimisation problems involving REITs. Future work will thus focus on finding appropriate machine learning algorithms to predict future prices of stocks, bonds, and REITs, which are as close as possible to the real values that appear in the test set. Succeeding in this task will allow us to observe similarly good performance in returns and risk, as we have observed under the theoretical case of perfect foresight.

In the following chapters, we will attempt to optimise a mixed-asset portfolio including REITs by using ML algorithms to predict asset prices and a GA to optimise the asset weights (Chapter 5). In this way, we expect to obtain better results in terms of risk-adjusted portfolio performance than when adopting a historical data approach.

Chapter 5

ML for Real Estate Time Series Prediction

5.1 Introduction

The previous chapter presented evidence that a hypothetical portfolio constructed with perfect foresight, meaning it accurately predicts future market movements, performed better than a portfolio constructed solely based on historical data. This finding suggested a departure from state-of-the-art reliance on past prices, particularly prevalent in real estate investment portfolio optimisation literature.

The chapter emphasises the benefits of utilising price predictions for better portfolio performance across various asset classes, including Real Estate Investment Trusts (REITs). It argues that predictions tend to align more closely with actual data patterns, potentially outperforming historical data-based strategies due to their ability to capture future market dynamics.

To investigate this claim, the chapter explores five machine learning algorithms – i.e., Ordinary Least Squares Linear Regression (LR), Support Vector Regression (SVR), eXtreme Gradient Boosting (XGBOOST), Long/Short-Term Memory Neural Networks (LSTM), and k-Nearest Neighbours Regression (KNN) – and compares their efficacy in predicting prices with three traditional statistical models commonly used in financial forecasting – Holt's Linear Trend Method (HLTM), Trigonometric Box-Cox Autoregressive Time Series (TBATS), and Autoregressive Integrated Moving Average (ARIMA). It then integrates these price predictions into portfolio optimisation using a Genetic Algorithm.

Furthermore, the chapter conducts a thorough examination of expected portfolio metrics derived from price predictions, comparing them against those from historical-based approaches. This empirical evidence highlights the effectiveness of incorporating a forward-looking approach into portfolio optimisation, particularly relevant for investors interested in real estate assets such as REITs.

Overall, the chapter's novelty lies in integrating machine learning-derived price predictions with portfolio optimisation techniques, providing a comprehensive comparison with state-of-the-art models, and offering empirical evidence of their effectiveness in enhancing portfolio performance.

The rest of this chapter is organised as follows. Section 5.2 explains the methodology used in this study. Our experimental setup is presented in Section 5.3. The results of our experiments are presented in Section 5.4, where we provide a detailed discussion of the results obtained by applying machine learning and other financial models to our data. Finally, Section 5.5 summarises the conclusions of the study and offers suggestions for future research.

5.2 Methodology

Our methodology can be broken down into two steps: (i) price prediction, and (ii) portfolio optimisation. In the first step, the machine learning algorithms employed in this study undergo training on the training set, aiming to minimise the root mean squared error (RMSE) of predicted prices for various assets. Such metric will be defined in Section 5.2.4. Subsequently, these trained models are utilised to forecast prices in the test set. In the second step, the predicted prices from the test set are fed into the genetic algorithm (GA), which seeks to optimise the allocation of weights assigned to each asset. The performance metric used for this portfolio optimisation task is the Sharpe ratio. The portfolio optimisation process incorporates principles derived from the Modern Portfolio Theory (MPT).

This section will thus present in detail the first step of our methodology (price prediction), since the second step has already been described in Chapter 4 (portfolio optimisation via a Genetic Algorithm): Section 5.2.1 describes the nature of the data in general terms; Section 5.2.2 discusses the pre-processing steps that were necessary for deriving the feature set; Section 5.2.3 presents the machine learning algorithms used in our experiments; and lastly, Section 5.2.4 discusses the loss function chosen.

5.2.1 Data

In this study, we consider a number of datasets[1] from financial instruments in relation to three asset classes — namely: stocks, bonds, and REITs; and three different markets — namely: United States (US), United Kingdom (UK), and Australia (AU). To avoid currency risk, all data is obtained as US dollars (USD). For more details regarding the exact number and specifics of the actual data used in our experimental setting, see Section 5.3.1 later on.

1. In the context of this study the word 'dataset' is used to refer to a single time-series of daily prices for a given asset

Each dataset is then further subdivided into three subsets, contiguous in time: a training set, which serves as the portion of the data that will be used to train the machine learning model; a validation set, which is used to select optimal hyperparameters for the model; and a testing set, which serves as the unseen part of data that is used for the final evaluation step, after the model has tuned and trained.

5.2.2 Data Preprocessing

Data coming from an asset's daily price time-series cannot be plugged directly into the algorithms: prior to being used for price prediction, the time-series data corresponding to each asset needs to be differenced and scaled. Differencing is an important technique in time series analysis, which involves taking the difference between consecutive observations of a time series. This is useful for removing the trend and seasonality components of a time series, which can make it difficult to model and analyse. First-order differencing involves subtracting the value of the previous timepoint from the current timepoint; this is represented mathematically as:

$$D_t = P_t - P_{t-1} \tag{5.1}$$

where P_t is the value of the time series at time t, and D_t is the differenced time series at time t. Higher-order differencing can also be used to remove trend and seasonality components that persist after first-order differencing. The choice of the order of differencing depends on the specific characteristics of the time series being analysed; for the purposes of this paper, we consider first-order differencing only.

After obtaining D_t, the values are further standardised to the range $[0,1]$, by using the following scaling transformation:

$$N_t = \frac{(D_t - D_{min})}{(D_{max} - D_{min})} \tag{5.2}$$

where N_t is the standardised value of each variable (in this case the differenced price D_t), and D_{min} and D_{max} are the minimum and maximum values respectively, that result from the differencing of the relevant asset's time series.

Table 5.1 provides an example of the differencing and scaling procedures using sample data for the SPG time series from 01 January 2021 to 30 January 2021.

Table 5.1: Example of time series differencing and scaling.

t	P_t	P_{t-1}	D_t	N_t	N_{t-1}	N_{t-2}
t1	3.77	-	-	-	-	-
t2	3.69	3.77	-0.08	0.30	-	-
t3	3.7	3.69	0.01	0.70	0.30	-
t4	3.6	3.7	-0.1	0.22	0.70	0.30
t5	3.68	3.6	0.08	1	0.22	0.70
t6	3.53	3.68	-0.15	0	1	0.22
t7	3.54	3.53	0.01	0.70	0	1

Legend: t represents the time steps; P_t represents the security's price at time t; P_{t-1} represents the one-lag value of P_t; D_t represents the differenced value at time t; N_t represents the value of D_t following standardisation, N_{t-1} the value of D_{t-1} following standardisation, etc.

5.2.3 Machine Learning Algorithms

Once the relevant features have been extracted from all datasets, we feed them to our 'bag' of machine learning models, for the purposes of price prediction. For each model, we obtain two variants: one that incorporates TAIs in its feature set, and one that does not. This allows us to be able to compare the performance between the two variants for each dataset, and thus assess the importance of including TAIs in the feature set.

Our 'bag' of machine learning models consists of a representative sample of regression algorithms taken from the Machine Learning (ML) literature, namely: Ordinary Least Squares Linear Regression (LR), Support Vector Regression (SVR), eXtreme Gradient Boosting (XG-BOOST), Long/Short-Term Memory Neural Networks (LSTM), and k-Nearest Neighbours Regression (KNN). The following `python` libraries/functions were used to this end:

- `sklearn.linear_model.LinearRegression`
- `sklearn.svm.SVR`
- `xgboost.XGBRegressor`
- `keras.models.Sequential`
- `sklearn.neighbors.KNeighborsRegressor`

In all cases, optimal model hyperparameters are determined through 'grid search' (see Section 5.3.2 for details). Once optimal hyperparameters are established a model is trained one last time on the expanded set of training + validation data combined, and then used to make predictions on the test set.

5.2.4 Evaluation Metrics

All of the above algorithms use the *root mean square error* (RMSE) as the loss function, defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{|j|}(P_i - \hat{P}_i)^2}{|j|}}, \tag{5.3}$$

where P_t refers to the actual value of the price, $\hat{P}_t$ is its predicted value, and $|j|$ denotes the number of observations in each dataset j (i.e. in other words, we obtain *one* RMSE value per dataset). Note that, the RMSE here expresses the prediction error in terms of *US dollars*, and thus needs to be calculated on the basis of the original price data (i.e. P_t), rather than the scaled data (i.e. N_t); therefore scaled values need to be reverted back to their original price values, for the RMSE to be calculated in a meaningful manner (cf. Section 5.2.2). This is done in order to allow for a comparison between the RMSE results obtained from the regression algorithms and the portfolio metrics that will result from the application of our GA that are calculated using actual (instead of scaled) prices.

We evaluate all algorithms using two out-of-sample prediction methods — one relying on long-term prediction on the basis of fixed information and intermediate predictions, and one relying on consecutive short-term predictions on the basis of continuously updated information. Both methods are evaluated over the same range of time periods, namely 30, 60, 90, 120, and 150 days.

Long-Term Out-Of-Sample Prediction: In this method, the known closing prices from all historical timepoints up until our starting point of interest, t_0 (with closing price P_0 respectively) are used to train a model, which is then used to predict the closing price for the next day (i.e. price $\hat{P}_1$, corresponding to timepoint t_1). Once this is obtained, the model is retrained, with $\hat{P}_1$ incorporated into the training dataset, as if it was the 'known' price at time t_1; this model is then used to predict the price for the next timepoint (i.e. price $\hat{P}_2$ corresponding to timepoint t_2). $\hat{P}_2$ is then used to predict $\hat{P}_3$ in the same manner, and so forth, until the final timepoint in the evaluation period of interest is reached. We will refer to this evaluation method simply as *out-of-sample* prediction henceforth in the text.

Consecutive One-Day-Ahead Predictions: In this case, when it comes to predicting the closing price $\hat{P}_1$ corresponding to timepoint t_1, we use the known closing prices from all historical timepoints up until t_0, just as we did before. However, when it then comes to predicting the *next* item (i.e. the closing price $\hat{P}_2$ corresponding to timepoint t_2), instead of incorporating the *predicted* price, $\hat{P}_1$ to our training set at position t_1, we simply incorporate the true, *known* closing price, P_1, to the training dataset at that position instead; the updated model is then used to predict the price for the next timepoint (i.e. $\hat{P}_2$ for position t_2), much like before. The known P_2 is then used to predict $\hat{P}_3$ in the same manner, and so forth, until the final time point in the evaluation period of interest is reached. We will refer to this evaluation method simply as *one-day-ahead* prediction henceforth in the text.

Naturally, we expect lower error rates from the second technique. However, it is a suitable method to evaluate performance, particularly meaningful in portfolios following short-term trading strategies that require adjustments according to current market conditions. In contrast, the first approach is more suitable as an evaluation strategy for investors with long investment horizons. They may prefer periodic portfolio rebalancing or adjustments to investment strategies based on evolving market conditions.

5.3 Experimental Setup

The main goal of our experiments is to show the potential improvements of predicting prices in terms of portfolio performance. This goal has been broken down into two sub-goals: (a) to showcase the reduction in the regression error by using ML algorithms as compared to three benchmarks and the historical data approach; and (ii) to demonstrate that the use of ML algorithms in predicting REIT, stock and bond prices could lead to a significant improvement in the risk-adjusted performance of a mixed-asset portfolio that includes REITs.

In the remainder of this section, we will first present the data used for our experiments, in Section 5.3.1. We will then discuss the algorithmic hyperparameter tuning in Section 5.3.2. Lastly, in Section 5.3.3 we will discuss the benchmarks used in our experiments.

5.3.1 Data

To conduct a comprehensive analysis of the ML performance in the price prediction task, we have decided to utilise data specific to individual companies rather than relying on market proxies, as was the methodology employed in the previous chapter. This shift allows us to perform a more granular examination of the predictive capabilities of our selected ML techniques at the company level. Daily closing price data was collected via the *Eikon Refinitiv* database[2],

2. https://eikon.refinitiv.com — Last access: July 2023.

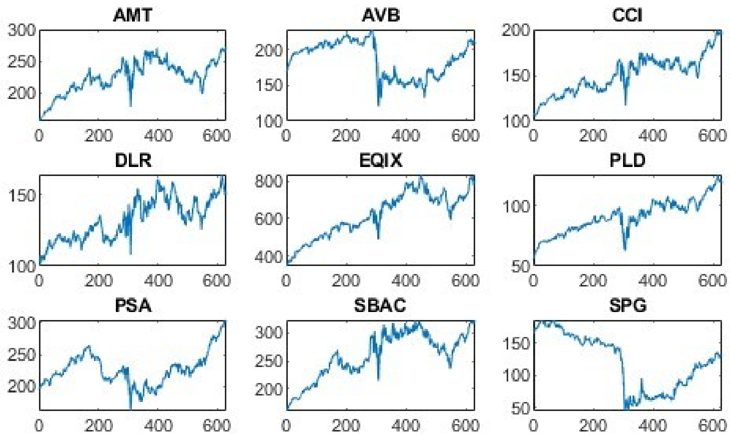


Figure 5.1: US REIT time series. The x-axis represents time in days; the y-axis refers to the price value in USD.

corresponding to financial instruments across three countries (US, UK, and Australia), and three asset classes (stocks, bonds, and real estate), spanning the period from January 2019 to July 2021. For each of the resulting nine 'country/asset-class' pairs above, we obtained asset-price data from 10 different assets within that category (in other words: 10 stocks, 10 bonds, and 10 REITs from each country), resulting in a dataset pool consisting of a total of 90 datasets (refer to Table 5.2). We remind the reader that in the context of this study the word 'dataset' is used to refer to a single time-series of daily prices for a given asset. To avoid currency risk, we obtained all data expressed in USD.

Table 5.2: Eikon Refinitiv tickers used.

	US	**UK**	**Australia**
Stocks	AAPL, AMZN, BRKb, GOOGL, JNJ, META, MSFT, NVDA, TSLA, UNH	AZN, BATS, BP, DGE, GLEN, GSK, HSBA, RIO, SHEL, ULVR	ANZ, BHP, CBA, CSL, FMG, MQG, NAB, WBC, WES, WOW
Bonds	AFIF, HOLD, IBMN, IUWAA, JNK, KORP, LQD, LQDI, NFLT, RIGS	AGPH, CCBO, DTLE, EMDD, EMES, ERNA, ERNS, FLOS, IHYG, SDHY	CRED, HBRD, IAF, QPON, RCB, RINCINAV, VACF, VAF, VBND, VGB
Real Estate	AMT, AVB, CCI, DLR, EQIX, PLD, PSA, SBAC, SPG, WELL	AEWU, AGRP, BLND, BYG, CAL, CREI, CSH, CTPT, DLN, EPICE	BWP, CHC, DXS, GMG, GOZ, GPT, MGR, SCG, SGP, VCX

It is important to note that many of the price series datasets can exhibit significant fluctuations, particularly for stocks and REITs. For instance, consider Figure 5.1, which illustrates the US REIT closing price time series for the period between 1st January 2021 and 1st July 2021. As shown, there are notable downward variations in the trend. Such fluctuations can potentially impact the performance of some algorithms, particularly ARIMA (one of our benchmarks), which relies heavily on assumptions of stationarity.

Table 5.3 presents summary statistics for the daily return distributions grouped by each of the nine asset classes considered. The term ‘return’ in this context is used specifically to refer to the quantity $(N_t - N_{t-1})/N_{t-1}$, i.e. the relative rate of returns, which is the difference between an asset’s normalised price difference on a particular day compared to the day before, expressed as a percentage of the latter. To be able to explain the prediction capability of our algorithms for the different asset classes through the presented summary statistics, we are considering the normalised – rather than actual – prices. For each asset class, we computed the mean, median, standard deviation, interquartile range, and maximum-minimum range to summarise the return distributions. Each asset within an asset class was given an equal weight, and the summary statistics were calculated based on the training period.

Table 5.3: Summary statistics for different asset classes. Values in bold denote the best values for each column.

	Average	**Median**	**Std Dev**	**IQR**	**Max-Min**
AU bonds	1.97×10^{-4}	3.15×10^{-4}	$\mathbf{5.70 \times 10^{-3}}$	$\mathbf{3.00 \times 10^{-3}}$	$\mathbf{9.54 \times 10^{-2}}$
AU REITs	7.35×10^{-4}	1.20×10^{-3}	2.44×10^{-2}	1.87×10^{-2}	2.95×10^{-1}
AU stocks	$\mathbf{2.00 \times 10^{-3}}$	$\mathbf{1.80 \times 10^{-3}}$	2.44×10^{-2}	2.14×10^{-2}	2.59×10^{-1}
UK bonds	2.38×10^{-4}	3.86×10^{-4}	7.90×10^{-3}	5.70×10^{-3}	1.12×10^{-1}
UK REITs	7.11×10^{-5}	4.35×10^{-4}	2.56×10^{-2}	2.14×10^{-2}	3.51×10^{-1}
UK stocks	1.88×10^{-4}	3.83×10^{-5}	2.14×10^{-2}	1.93×10^{-2}	2.61×10^{-1}
US bonds	3.11×10^{-4}	2.74×10^{-4}	8.50×10^{-3}	7.70×10^{-3}	1.07×10^{-1}
US REITs	6.99×10^{-4}	7.25×10^{-4}	2.59×10^{-2}	1.95×10^{-2}	3.49×10^{-1}
US stocks	1.10×10^{-3}	1.20×10^{-3}	2.25×10^{-2}	1.86×10^{-2}	2.40×10^{-1}

The first column shows the average daily return for each asset class. Australian stocks present the highest daily average return at 2.00×10^{-3}, followed by US stocks at 1.10×10^{-3}, and Australian REITs at 7.35×10^{-4}. The highest median value is observed for Australian stocks at 1.80×10^{-3}, followed by Australian REITs and US stocks at 1.20×10^{-3}, and US REITs at 7.25×10^{-4}. Stocks tend to have higher rates of return compared to other asset classes such as REITs and bonds.

As for the standard deviation of returns, Australian bonds exhibit the lowest volatility value at 5.70×10^{-3}, followed by UK bonds at 7.90×10^{-3}, and US bonds at 8.50×10^{-3}. Similarly, the lowest interquartile range is observed for Australian bonds at 3.00×10^{-3}, followed by UK bonds at 5.70×10^{-3}, and US bonds at 7.70×10^{-3}. The maximum-minimum ranges show the lowest value for Australian bonds at 9.54×10^{-2}, followed by US bonds at 1.07×10^{-1}, and UK bonds at 1.12×10^{-1}. This is expected since bond rates of return tend to be less volatile than those of other asset classes.

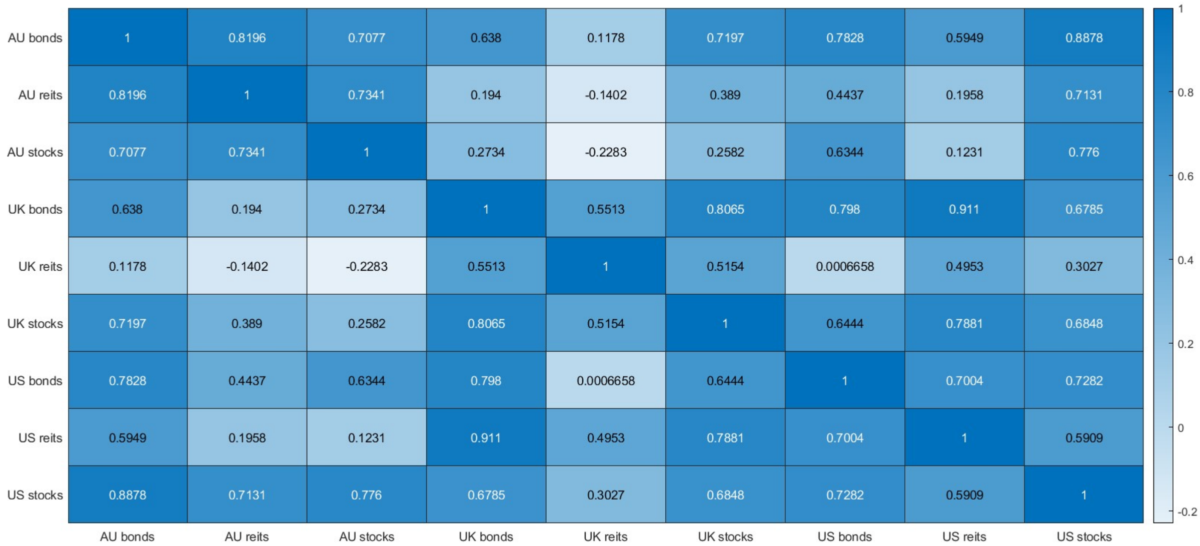


Figure 5.2: Correlation matrix between asset classes.

In summary, we observed that bond rates of return present less volatility compared to other asset classes, and also lower average values. On the other hand, stock markets are typically more volatile, but also more profitable than other asset classes. Real estate returns fall somewhere in between in terms of expected return and volatility. This clarifies why portfolios that include real estate exhibit higher returns and lower risks in comparison to portfolios that only include stocks and bonds [129].

Moreover, it is important to highlight that the correlation between real estate asset classes and the other asset classes tends to be low, particularly when investing internationally, which provides diversification benefits and consequently reduces the overall risk level of a mixed-asset portfolio (refer to Figure 5.2). For example, the correlation between UK REITs and Australian stocks is -0.23, the correlation between UK REITs and US bonds is 6.66×10^{-4}, and the correlation between US REITs and Australian stocks is 0.12. In contrast, the correlation between US stocks and Australian bonds is 0.89, the correlation between UK stocks and UK bonds is 0.81, and the correlation between Australian stocks and US stocks is 0.78. These values illustrate why adding international REIT investments to a portfolio can help to mitigate risk, as per the MPT.

5.3.2 Experimental Tuning of Hyperparameters

In order to solve the price prediction problem using machine learning algorithms, we tailored the experimental hyperparameters of each ML algorithm to each dataset by performing tuning, resulting in each dataset having its own set of unique hyperparameters. The *Grid Search* method in Python was employed to determine the optimal hyperparameters, with the ranges for hyperparameter values being established based on the types of datasets utilised (see Table 5.4. It is worth noting that hyperparameter tuning was not performed for the LR model, as it lacks hyperparameters that require tuning.

GA hyperparameter values were tuned on the same validation set. The resulting tuned values are presented in Table 5.5.

Table 5.4: ML algorithms and parameters.

Algorithm	Parameter	Value range
SVR	Kernel function	'linear', 'poly', 'rbf', 'sigmoid'
	Degree of the kernel function	1, 2, 3
	Kernel coefficient (gamma)	'scale', 'auto'
	Tolerance for stopping criterion	0.001, 0.01, 0.1
	Epsilon	0.1, 0.5, 0.8
	Regularisation parameter (C)	1.0, 1.5, 2
XGBOOST	Number of estimators	10, 20, 30
	Maximum depth of a tree	3, 4, 5
	Minimum child weight	1, 5, 10
	Learning rate	0.001, 0.01, 0.1
LSTM	Number of epochs	Early stopping criterion
	Batch size	4, 8, 16
	Number of hidden layers	1, 2
	Number of neurons	5, 10, 25, 50
KNN	Number of neighbours	5, 10, 20
	Weights	'uniform', 'distance'
	Algorithm	'auto', 'ball_tree', 'kd_tree'

Table 5.5: GA parameters.

Parameter	Values
Population size	500
Tournament size	3
Mutation rate	0.01
Number of generations	25

5.3.3 Benchmarks

As mentioned in the beginning of Section 5.3, our two sub-goals are to demonstrate the added value of implementing ML algorithms in the regression task and asset allocation. For this purpose, we also explore the performance of three common financial benchmarks, which are presented next, in Section 5.3.3. Furthermore, for the problem of portfolio optimisation, we are also interested in comparing the algorithms' performance across different portfolio techniques. We thus introduce two further benchmarks, which are presented in Section 5.3.3.

Regression Benchmarks

HLTM *Holt's Linear Trend Method* (HLTM; also known as 'Double-Exponential Smoothing' due to the involvement of two exponentially weighted moving average processes in its formulation) is a forecasting method that makes a prediction on the basis of a predicted baseline at the last known data point, and a linear trend extending from that point into the future. It is an extension of Simple Exponential Smoothing that adds a trend component to the model, and where that trend itself is also the result of a Simple Exponential Smoothing process over past trends.

HLTM has two smoothing parameters, α and β, which control the weight given to the most recent observation and the trend, respectively. The forecast equation for HLTM is as follows:

$$\hat{N}_{t+h|t} = \ell_t + hb_t, \tag{5.4}$$

where ℓ_t is the level estimate at time t, b_t is the trend estimate at time t, and h is the number of periods ahead to forecast. The level and trend estimates are updated at each time step as follows:

$$\ell_t = \alpha N_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{5.5}$$

$$b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}, \tag{5.6}$$

where N_t is the observed value at time t, and $0 < \alpha < 1$ and $0 < \beta < 1$.

The HLTM method has been widely used in forecasting and has been shown to perform well in many different applications [130]. Given that it uses a weighted average of past observations to make its predictions, it is not able to also use TAIs in its feature set. Nevertheless, it forms a valuable benchmark, as it allows us to compare the performance of our proposed approach with a well-known time series prediction benchmark.

TBATS *Trigonometric Box-Cox Autoregressive Time Series* (TBATS) is a state-of-the-art forecasting model that extends the traditional exponential smoothing framework to handle complex time series with multiple seasonal patterns and non-linear trends. TBATS was proposed by [131].

The TBATS model involves the decomposition of a time series into multiple components: a non-seasonal component, seasonal components, and an autoregressive component. The non-seasonal component captures the overall trend of the time series and is modelled using a Box-Cox transformation and an exponential smoothing model. The seasonal components capture the periodic patterns in the time series and are modelled using a set of trigonometric functions. Finally, the autoregressive component captures the temporal dependencies in the time series and is modelled using an Autoregressive Moving Average (ARMA) model.

The TBATS model can be written as:

$$N_t = \mu_t + \sum_{j=1}^{J} \gamma_j s_{t,j} + \sum_{i=1}^{p} \phi_i N_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i} + e_t \tag{5.7}$$

where N_t is the observed value of the time series at time t, μ_t is the non-seasonal component at time t, $s_{t,j}$ is the seasonal component for season j at time t, γ_j is the coefficient for season j, p and q are the orders of the autoregressive and moving average components, respectively, ϕ_i and θ_i are the corresponding coefficients, e_t is the error term at time t, and J is the number of seasonal patterns in the data.

TBATS has been shown to outperform traditional forecasting models such as ARIMA and exponential smoothing on time series with multiple seasonal patterns and non-linear trends [130]. Similarly to HLTM, TBATS is not able to use TAIs in its feature set; however, it also serves as a valuable benchmark, as yet another well-known and widely-used time series prediction benchmark.

ARIMA *Autoregressive Integrated Moving Average* (ARIMA) is a commonly used time series model for forecasting. It is a statistical model that uses past values and errors to make predictions. ARIMA models can capture both trend and seasonality in the data and are widely used in many fields, including economics, finance, and engineering.

The ARIMA model is denoted by ARIMA(p, d, q), where p is the order of the autoregressive term, d is the degree of differencing required to make the series stationary, and q is the order of the moving average term. The model assumes that the time series is stationary, which means that its mean and variance are constant over time.

The ARIMA model can be represented mathematically as:

$$N_t = c + \sum_{i=1}^{p} \phi_i N_{t-i} + \varepsilon_t + \sum_{i=0}^{q} \theta_i \varepsilon_{t-1} \tag{5.8}$$

where ϕ denotes the autoregression coefficient, θ refers to the moving average coefficient, and ε refers to the error rate of the autoregression model at each time point.

In order to identify the suitable ARIMA model for each training dataset, *Akaike Information Criterion* was used, and the values of p, d, and q corresponding to the minimum AIC value were selected.

It is worth noting that ARIMA models are only applicable to stationary time series, which implies that the statistical properties of the series remain constant over time. Since many financial time series are not stationary, several transformations such as differencing, logarithmic transformation, and Box-Cox transformation are required to be applied.

ARIMA has been widely applied in various fields. For example, it has been used to forecast stock prices [132], electricity demand [133], and weather variables [134]. As with HLTM and TBATS, it is not able to also use TAIs in its feature set, but again enjoys wide use in the financial forecasting literature, and therefore forms a valuable benchmark.

Portfolio Optimisation Benchmarks

Portfolio optimisation involves running a Genetic Algorithm on the price data predicted by our ML algorithms, in order to obtain appropriate weights for the different asset classes for each of the 90 assets that make up a portfolio. The quality of the resulting portfolios is then assessed on the basis of financial metrics calculated from the observed prices for that period. In order to assess the added value of ML-based price predictions, we compare the performance of a portfolio built using the above against a portfolio obtained by adopting a historical data and perfect foresight approach.

Historical Data Portfolio Optimising weights on the training set (i.e. on historical data), rather than the test set which is our proposed methodology, is a common approach in the literature [20, 18, 19]. However, a drawback of this method is that the trained weights might be 'obsolete' if the test set price series significantly varies to the price series of the training set. Nevertheless, given that this is still a common approach, we are motivated in using it as a benchmark to demonstrate the benefits of our proposed approach.

Perfect Foresight Portfolio This is a theoretical benchmark, as it assumes perfect price predictions in the test set. The reason for including this benchmark is to be able to see how closely or how far away is the ML-based portfolio performance to the performance of the theoretical portfolio of perfect price predictions. This will assist us in understanding the quality of the performance of our proposed portfolio, and is thus a useful real-world benchmark.

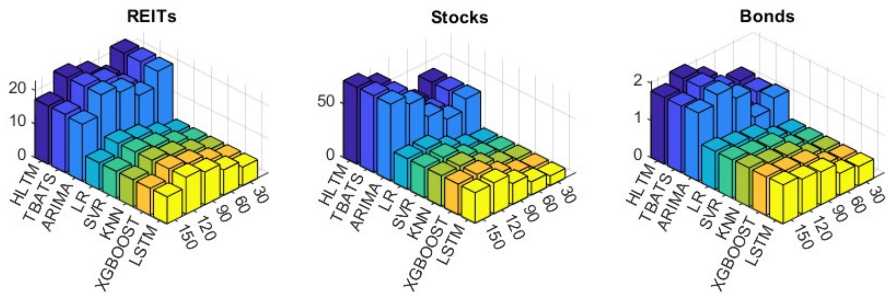
5.4 Results

In Section 5.4.1, we evaluate and compare the performance of the five ML algorithms as introduced in Section 5.2.3, in contrast to the three traditional techniques detailed in Section 5.3.3, specifically, HLTM, TBATS, and ARIMA. In Section 5.4.2, we investigate the implications of utilising the algorithmic predictions to optimise a multi-asset portfolio through a Genetic Algorithm approach, and how this impacts the expected return, risk, and Sharpe Ratio values within the resulting portfolios. Finally, in Section 5.4.3, we analyse the computational times associated with the utilised algorithms, and Section 5.4.4 provides a concise discussion of the insights derived from the experimental results.

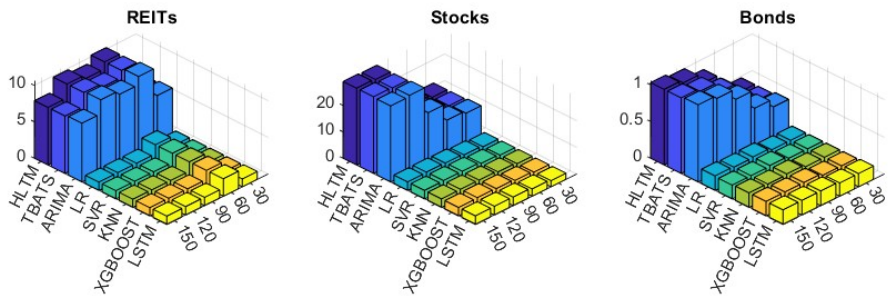
5.4.1 ML Prediction

The purpose of this experimental set is to investigate and compare the performance of different ML algorithms and financial benchmarks on the task of predicting asset prices, which are subsequently going to be used as inputs in a portfolio optimisation task (Section 5.4.2). In the following sections, we explore the predictive capability of the five ML algorithms compared to the three financial benchmarks considered and the historical data method. For this purpose, we present the mean and standard deviation of Root Mean Square Error (RMSE) for each asset class and algorithm used.

Figure 5.3 presents the RMSE results for the three asset classes, over the 8 algorithms and the 5 different horizons, both for out-of-sample (top) and one-day-ahead (bottom) methods. With regards to the out-of-sample results, we can observe that all machine learning algorithms experience considerably lower RMSE values than the econometric benchmarks (HLTM, TBATS, ARIMA), with improvements often being more than 50%. This is an important finding, which demonstrates the strengths of ML algorithms compared to the econometric approaches. Furthermore, we can also observe a tendency of increased RMSE values as the horizon increases, across all algorithms. Lastly, it is worth noting that bonds tend to experience the lowest error (RMSE values up to around 2), followed by REITs (RMSE value up to around 20), and then by stocks (RMSE values around 70).



(a) Out-of-sample RMSE results for REITs, stocks, and bonds for 8 algorithms and 5 horizons.



(b) One-day-ahead RMSE results for REITs, stocks, and bonds for 8 algorithms and 5 horizons.

Figure 5.3: Comparison of RMSE results

With regards to the one-day-ahead results, we can make similar observations: ML errors are again considerably lower than the benchmarks, and bonds experience the lowest error, followed by REITs, and then by stocks. One important difference to the previous (out-of-sample) results is that one-day-ahead consistently experiences lower errors, which is expected, as it was explained earlier. As we can observe, the highest error per asset class tends to be at least 50% lower for the one-day-ahead method (REITs: from 20 to 10; Stocks: from 70 to 30, Bonds: from 2 to 1). Lastly, we have provided, for reference, detailed distribution statistics for all RMSE results in the Appendix, Tables B.1 - B.3.

In order to compare the RMSE results among the different algorithms, we run the Friedman non-parametric test, where we calculated the average rank of each algorithm–the lower the average rank, the better the algorithm's performance. The average rank is based on the comparison in terms of RMSE values for each dataset among the different algorithms. In addition to the Friedman test, we also performed the Bonferroni post-hoc test. We present both in Table 5.6. For each algorithm, the table shows the average rank (first column), and the adjusted p-value of the statistical test when that algorithm's average rank is compared to the average rank of the algorithm with the best rank (control algorithm) according to Bonferroni's post-hoc test (second column) [135, 136]. When statistically significant differences between the average ranks of an algorithm and the control algorithm at the 5% level ($p \leq 0.05$) are observed, the relevant p-value is put in bold face. The statistical tests were conducted for all different setups, i.e., the combined results of different horizons (30-, 60-, 90-, 120-, and 150-days), over both the one-day-ahead and out-of-sample experiments.

The results indicate that the KNN algorithm is the optimal (control) algorithm, as it statistically outperforms LSTM, LR, HLTM, TBATS, and ARIMA, with corresponding p-values of 1.86×10^{-5}, 6.83×10^{-8}, 5.76×10^{-24}, 5.76×10^{-24}, and 3.58×10^{-45}, respectively. Conversely, the performance of SVR and XGBOOST algorithms does not exhibit a statistically significant difference from that of KNN, with p-values of 5.62 and 0.76, respectively.

To ensure the meaningfulness of our results, we calculated effect size measurements using the Kendall's W coefficient of concordance. This measure assesses the degree of agreement among the rankings, with values ranging from 0 (no agreement) to 1 (complete agreement). For the comparison between KNN and the other algorithms, the effect size was as follows: KNN vs. LSTM: $W = 0.32$ (moderate effect); KNN vs. LR: $W = 0.45$ (moderate effect); KNN vs. HLTM: $W = 0.67$ (large effect); KNN vs. TBATS: $W = 0.67$ (large effect); and KNN vs. ARIMA: $W = 0.89$ (large effect). These effect size measurements confirm that the differences between KNN and the other algorithms are not only statistically significant but also meaningful, particularly for HLTM, TBATS, and ARIMA, where the effect sizes are large. This provides enough evidence that KNN consistently outperforms these algorithms across the different setups.

In conclusion, we observed that the RMSE distributions tend to be lower on average for ML algorithms than for benchmark algorithms, with better results observed for one-day-ahead prediction (as expected). We also noticed that the lowest average RMSE values are observed for bonds, followed by REITs and stocks. This is explained by the lower volatility featuring bond prices that we have already discussed in Section 5.3.1. In the case of REITs, the RMSE distributions tend to have higher averages than for bonds but lower than for stocks. This is due to the financial structure of REIT prices which is between that of bonds and stocks in terms of risk and return. According to our statistical test results, KNN is the best algorithm in predicting the prices of REITs, stocks and bonds both one-day-ahead and out-of-sample.

5.4.2 Portfolio Optimisation

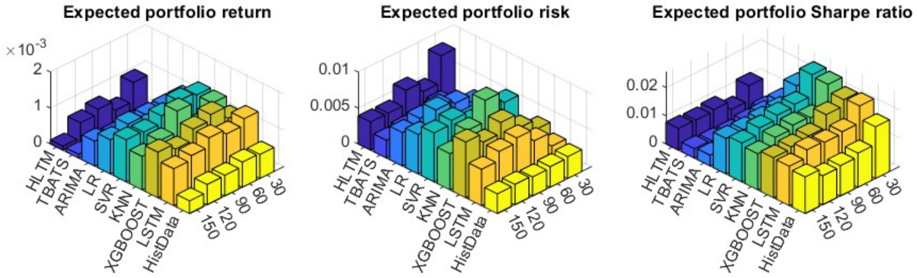
In this section, we present the results of the genetic algorithm (GA) applied to portfolio allocation, considering a transaction cost of 0.02%. After determining the optimal weights, we obtain the distributions for expected returns, expected risks, and the Sharpe ratio for each dataset. The machine learning models are compared against benchmarks (HLTM, TBATS,

Table 5.6: Statistical test results according to the non-parametric Friedman test with Bonferroni's post-hoc test RMSE distributions. Values in bold represent a statistically significant difference at the 5% significance level.

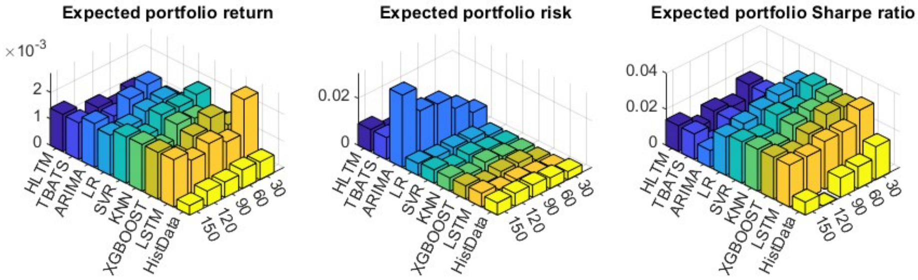
Algorithm	Avg Rank	p_{Bonf}
KNN (c)	2.88	-
SVR	2.91	5.62
XGBOOST	3.07	0.76
LSTM	3.42	$\mathbf{1.86 \times 10^{-5}}$
LR	3.54	$\mathbf{6.83 \times 10^{-8}}$
HLTM	6.46	$\mathbf{5.76 \times 10^{-24}}$
TBATS	6.46	$\mathbf{5.76 \times 10^{-24}}$
ARIMA	7.26	$\mathbf{3.58 \times 10^{-45}}$

and ARIMA), the historical data method, and the theoretical perfect foresight approach. For the historical data method, portfolio optimisation is carried out on the training set, while the expected portfolio metrics are obtained from the testing set. In the perfect foresight approach, portfolio optimisation is conducted on the testing set.

Figure 5.4 presents the expected return distributions (left), expected portfolio risk (middle), and expected Sharpe ratio (right) obtained from the GA portfolio optimisation task for 30-, 60-, 90-, 120-, and 150-day holding periods, for out-of-sample (top) and one-day-ahead (bottom). With regards to the portfolio returns, we observe that for the out-of-sample method, the machine learning algorithms yield higher returns across all holding periods when compared to the benchmark methods and the historical data approach. The highest average daily return for the 30-day period is achieved by LSTM and SVR (1.44×10^{-3}), followed by KNN (1.43×10^{-3}). In the case of one-day-ahead predictions for the same holding period, the expected daily return is higher for all the algorithms, with the highest value achieved by LSTM (2.72×10^{-3}). For reference, the average return for the theoretical benchmark of perfect foresight is 4.16×10^{-3}. It is also worth noting that all algorithms (except TBATS in the case of out-of-sample prediction) outperform the historical method, highlighting the importance of making price predictions in the test set, rather than simply applying the weights obtained in the training set directly to the test set. There is, of course, room for even greater improvements, given that the 'ceiling' of perfect foresight is around 57% higher than LSTM's average daily return of 2.72×10^{-3} (for the one-day-ahead method), showing that there is significant research potential in this area. Lastly, as the horizon period increases to 60 days and beyond, we observe similar improvements in the performance of the ML algorithms with respect to the benchmarks. For reference, detailed tables for returns, as well as risk and Sharpe ratio distributions, are provided in the Appendix, in Tables B.4 - B.6.



(a) Out-of-sample portfolio results for REITs, stocks, and bonds for 8 algorithms and 5 horizons.



(b) One-day-ahead portfolio results for REITs, stocks, and bonds for 8 algorithms and 5 horizons.

Figure 5.4: Comparison of portfolio results. For reference, the perfect foresight values for returns are 4.16×10^{-3} (30 days), 4.07×10^{-3} (60 days), 4.56×10^{-3} (90 days), 3.85×10^{-3} (120 days), and 3.78×10^{-3} (150 days). The perfect foresight values for risk are 1.14×10^{-3} (30 days), 2.42×10^{-3} (60 days), 2.51×10^{-3} (90 days), 2.58×10^{-3} (120 days), and 2.34×10^{-3} (150 days). The perfect foresight values for Sharpe ratio are 4.04×10^{-2} (30 days), 3.72×10^{-2} (60 days), 3.72×10^{-2} (90 days), 3.29×10^{-2} (120 days), and 3.23×10^{-2} (150 days).

To investigate if the above results are statistically significant, we again performed a Friedman test at the 5% significance level, along with the Bonferroni post-hoc test. We present these results in Table 5.7 for returns (left), risk (middle), and Sharpe ratio (right). With regards to returns, LSTM has the best rank (2.97), followed by SVR (2.99), and KNN (3.30). Given a 5% significance level, LSTM statistically outperforms XGBOOST (p-value equal to 1.40×10^{-4}), LR (p-value equal to 2.14×10^{-18}), ARIMA (p-value equal to 1.98×10^{-54}), the historical method (p-value equal to 3.25×10^{-280}), HLTM (p-value equal to 0), and TBATS (p-value equal to 0). On the other side, there is no statistical difference between LSTM and SVR (p-value equal to 6.71×10^{-2}) and KNN (5.00×10^{-2}).

To further validate the practical significance of these findings, we calculated effect sizes using Kendall's W coefficient of concordance. For the comparison of LSTM with the other algorithms, the effect sizes were as follows: *LSTM vs. XGBOOST:* $W = 0.35$ (moderate effect), *LSTM vs. LR:* $W = 0.60$ (large effect), *LSTM vs. ARIMA:* $W = 0.75$ (large effect), *LSTM vs. historical method:* $W = 0.92$ (very large effect), *LSTM vs. HLTM:* $W = 1.00$ (very large effect), and *LSTM vs. TBATS:* $W = 1.00$ (very large effect). These effect size measurements indicate that, in addition to being statistically significant, the differences in performance are also practically meaningful, particularly in comparisons with ARIMA, the historical method, HLTM, and TBATS.

With regards to portfolio risks, we can generally observe that it tends to be higher for benchmarks with respect to the ML algorithms and the historical approach. However, there are some other cases, particularly in the out-of-sample method, where the econometric benchmarks and the historical data approach outperform the machine learning algorithms. Nevertheless, it is worth noting that the majority of these differences is not significant.[3] Furthermore, we can observe that the risk levels achieved by a perfect foresight-based portfolio (presented in the caption of Figure 5.4) are closely achieved by most of the portfolios built using one-day-ahead predictions. In the case of out-of-sample predictions, the relative difference between the risk value achieved by the best algorithm and the perfect foresight case is around 6% (TBATS) for a 30-day period, 15% (historical data approach) for a 60-day period, 5% (historical data approach) for a 90-day period, 12% (TBATS) for a 120-day period, and 6% (TBATS) for a 150 day period. This is an important observation, as it demonstrates that the above results are very close to the best possible risk performance that can be achieved, as shown in the theoretical case of perfect foresight.

According to the Friedman test results, we can observe that XGBOOST has the best rank (4.26), followed by KNN (4.32), and SVR (4.42). Given a 5% significance level, LSTM statistically outperforms TBATS (p-value equal to 4.51×10^{-5}), HLTM (p-value equal to 6.80×10^{-6}), and ARIMA (p-value equal to 0). On the other side, there is no statistical significance in the results between XGBOOST and KNN (p-value equal to 6.35), SVR (p-value equal to 2.31), LR (p-value equal to 2.25), LSTM (p-value equal to 1.67), and the historical data approach (p-value equal to 0.37).

To assess the practical significance of these results, we calculated the effect sizes using Kendall's W coefficient of concordance. The effect sizes for the significant comparisons were as follows: *LSTM vs. TBATS:* $W = 0.40$ (moderate effect), *LSTM vs. HLTM:* $W = 0.50$ (moderate effect), and *LSTM vs. ARIMA:* $W = 0.80$ (large effect). These effect size measurements indicate that while the differences are statistically significant, they are also practically meaningful, particularly in the comparison between LSTM and ARIMA.

3. This becomes evident when we have a look at the Friedman ranking, which is presented in Table 5.7.

Lastly, when looking at the Sharpe ratio results of Figure 5.4, we can observe that all ML algorithms outperform the benchmarks for all periods in both cases of out-of-sample and one-day-ahead predictions. In many cases, the differences in Sharpe ratio values are quite noticeable, e.g. for both out-of-sample and one-step ahead the econometric benchmarks (HLTM, TBATS, ARIMA) appear to have at least 50% lower values than the ML algorithms. This is an important observation, because it demonstrates the importance of using machine learning for price predictions instead of traditional econometric approaches.

From the Friedman test results, we notice that SVR has the best rank (2.63) followed by LSTM (2.82) and KNN (2.96). In addition, we observe that KNN statistically outperforms LR (p-value equal to 9.94×10^{-10}), XGBOOST (p-value equal to 2.21×10^{-11}), ARIMA (p-value equal to 1.05×10^{-279}), HLTM (p-value equal to 2.84×10^{-283}), TBATS (p-value equal to 0), and the historical data approach (p-value equal to 0). Lastly, it is worth noting that all algorithms have a higher rank compared to the historical method which showcases the importance of including price predictions in order to improve the risk-adjusted performance of a mixed-asset portfolio.

To further validate the practical significance of these differences, we calculated effect sizes using Kendall's W coefficient of concordance. The effect sizes for the significant comparisons were as follows: *KNN vs. LR:* $W = 0.42$ (moderate effect), *KNN vs. XGBOOST:* $W = 0.46$ (moderate effect), *KNN vs. ARIMA:* $W = 0.85$ (large effect), *KNN vs. HLTM:* $W = 0.88$ (large effect), *KNN vs. TBATS:* $W = 1.00$ (very large effect), and *KNN vs. historical data approach:* $W = 1.00$ (very large effect).

(a) Return

Algorithm	Avg Rank	p_{Bonf}
LSTM (c)	2.97	-
SVR	2.99	6.71×10^{-2}
KNN	3.30	5.00×10^{-2}
XGBOOST	3.49	$\mathbf{1.40 \times 10^{-4}}$
LR	4.06	$\mathbf{2.14 \times 10^{-18}}$
ARIMA	4.88	$\mathbf{1.98 \times 10^{-54}}$
HistData	7.35	$\mathbf{3.25 \times 10^{-280}}$
HLTM	7.80	**0**
TBATS	8.15	**0**

(b) Risk

Algorithm	Avg Rank	p_{Bonf}
XGBOOST (c)	4.26	-
KNN	4.32	6.35
SVR	4.42	2.31
LR	4.42	2.25
LSTM	4.44	1.67
HistData	4.53	0.37
TBATS	4.84	$\mathbf{4.51 \times 10^{-5}}$
HLTM	4.89	$\mathbf{6.80 \times 10^{-6}}$
ARIMA	8.87	**0**

(c) Sharpe ratio

Algorithm	Avg Rank	p_{Bonf}
SVR (c)	2.63	-
LSTM	2.82	1.06
KNN	2.96	0.06
LR	3.42	$\mathbf{9.94 \times 10^{-10}}$
XGBOOST	3.49	$\mathbf{2.21 \times 10^{-11}}$
ARIMA	7.02	$\mathbf{1.05 \times 10^{-279}}$
HLTM	7.05	$\mathbf{2.84 \times 10^{-283}}$
TBATS	7.61	**0**
HistData	8	**0**

Table 5.7: Statistical test results according to the non-parametric Friedman test with the Bonferroni post-hoc for expected returns (left), expected risks (middle), and expected Sharpe ratios (right). Values in bold represent a statistically significant difference.

5.4.3 Computational Times

The computational times of most algorithms were found to be comparable. On average, ARIMA took approximately 0.168 minutes to run, while LR, SVR, and KNN took between 0.2 and 0.3 minutes. LSTM was the most computationally expensive algorithm, taking around 1.818 minutes to run. With regards to the genetic algorithm, a single run took around 0.3 minutes to complete. Generally, we can observe that all of the runtimes are relatively fast. In addition, given that all of them are typically run offline, and only their trained models are used in real time, these time differences are not considered significant. Besides, parallelisation techniques can be employed to reduce the computational time of these algorithms [128].

5.4.4 Discussion

Our initial experimental objective was to showcase the enhancement in prediction accuracy achieved by employing machine learning (ML) algorithms in contrast to the three benchmark models considered and the historical data-based approach. The observed results revealed that the Root Mean Square Error (RMSE) distributions from the ML models exhibited lower average values and reduced volatility compared to the benchmark models. Notably, K-Nearest Neighbours (KNN), Support Vector Regression (SVR), and Extreme Gradient Boosting (XG-BOOST) outperformed the other models in both one-step-ahead and out-of-sample prediction accuracy.

Conversely, our experimental findings illustrated the superior portfolio performance resulting from the utilisation of ML predictions when compared to a portfolio constructed using price forecasts from state-of-the-art models (i.e., Holt's Linear Trend Method, Trigonometric Box-Cox Autoregressive Time Series, and Autoregressive Integrated Moving Average) and the historical method. This primarily arises from the poorer prediction accuracy exhibited by the benchmark models. Furthermore, Long/Short-Term Memory (LSTM), SVR, and KNN outperformed other algorithms in terms of portfolio returns, while Holt's Linear Trend Method (HLTM), Trigonometric Box-Cox Autoregressive Time Series (TBATS), and Autoregressive Integrated Moving Average (ARIMA) demonstrated inferior performance concerning portfolio risk. Additionally, SVR, LSTM, and KNN delivered the most favourable Sharpe ratio values compared to the other algorithms.

5.5 Summary

In this chapter, we analysed the predictive ability of five machine learning algorithms (LR, SVR, KNN, LSTM, and XGBOOST), and compared it against three benchmarks (HLTM, TBATS, and ARIMA).

From the above results, we can summarise our findings as follows.

Machine learning algorithms are able to outperform financial approaches for price prediction. The initial objective of our experiments was to compare the performance of ML models against our benchmark models, namely HLTM, TBATS, and ARIMA, in terms of their predictive power measured by RMSE. The experimental results showed that the RMSE distributions of the ML models tend to have lower average values and lower volatility than those of the benchmark models. The Friedman tests further revealed that KNN, SVR, and XGBOOST ranked first, second, and third, respectively, outperforming the other models, indicating their superior ability to make one-day-ahead and out-of-sample predictions compared to the statistical tools.

REITs' low volatility leads to improved price predictions. We observed that volatility affects price prediction results. More specifically, the predictive ability of the different algorithms tends to improve for bonds, which can be attributed to the lower price volatility for this asset class. In the case of REITs, the RMSE distributions show lower averages compared to stocks for all periods. This is due to a lower volatility that features REITs time series, as we have already discussed in Section 5.3.1.

Portfolios using prices predicted by ML algorithms lead to better performance. The second objective of our experiments was to compare the performance of portfolios derived from ML-based predictions with that of portfolios obtained from HLTM-, TBATS-, and ARIMA-based predictions (benchmarks), as well as a portfolio obtained from historical data. According to our findings, ML-based predictions increased the expected Sharpe ratio level compared to the historical data situation, mostly due to the increase in expected return levels rather than expected risk levels, which were also low for some of the benchmark algorithms. Having very good performance in terms of Sharpe ratio is paramount, because it is an aggregate metric that takes into account both returns and risk. It is also worth noting that practitioners pay particular attention to such aggregate metrics, thus the ML algorithms' superior performance in Sharpe ratio is a very positive result.

The inclusion of REITs into mixed-asset portfolios leads to better diversification results. Figure 5.5 displays the optimal weights of a portfolio constructed using SVR (the best ranked algorithm according to the Friedman test) out-of-sample price predictions. It is evident that the highest weight is assigned to UK stocks (44.27%), US bonds (24.07%), and UK REITs (20.78%). Such allocation aids in enhancing the final portfolio's performance, and is consistent with the one suggested by previous studies [137, 138, 139]. This underscores the importance of including REITs in mixed-asset portfolios due to the diversification potential of this asset class. In other words, the higher accuracy of out-of-sample predictions for REITs time series contributes to the construction of less risky portfolios and may be a signal of better risk-adjusted portfolio performance.

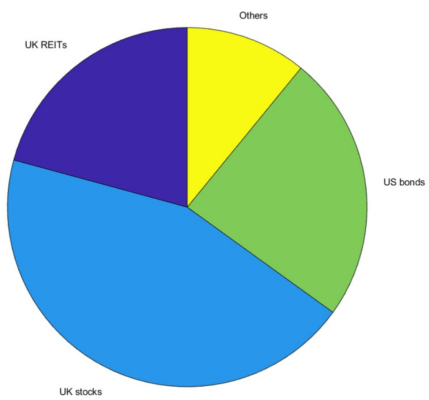


Figure 5.5: SVR-GA portfolio weights

The risk-adjusted performance of a portfolio obtained from ML predictions tend to be higher compared to the portfolio obtained from historical data and benchmark price predictions for all time horizons. We noticed that the average Sharpe ratio resulting from SVR predictions is the highest for a 30-day period, while the highest value is observed for XGBOOST for a 60-day period, SVR for a 90-day period, LSTM (in the case of out-of-sample predictions) and LR (in the case of one-day-ahead predictions) for a 120-day period, and XGBOOST for a 150-day period. As expected, the one-day-ahead predictions lead to better results in terms of Sharpe ratio compared to the out-of-sample predictions due to generally lower RMSE values for all time horizons. But as we have noticed, there is still some potential improvement in the portfolio performance that can be achieved by the ML algorithms.

Chapter 6

Improving REITs Time Series Prediction Using ML and TA Indicators

6.1 Introduction

In Chapter 5, we have presented an extensive analysis of incorporating machine learning-based price predictions for Real Estate Investment Trusts (REITs), stocks, and bonds into portfolio optimisation. We evaluated the performance of machine learning algorithms in terms of portfolio average returns, risk, and the Sharpe ratio. Our experiments revealed that machine learning models outperformed traditional econometric benchmarks such as HLTM, TBATS, and ARIMA when applied to one-day-ahead and out-of-sample price predictions across various time horizons (30, 60, 90, 120, and 150 days). This superiority was evident from the consistently lower average Root Mean Square Error (RMSE) values observed for machine learning algorithms.

However, it is worth noting that there is still room for improvement in the accuracy of machine learning predictions, particularly for out-of-sample predictions. By reducing the error associated with machine learning predictions, we have the potential to enhance the performance of multi-asset portfolios, particularly in terms of the average Sharpe ratio. To address this, we propose the introduction of Technical Analysis Indicators (TAIs) as additional features for our regression problem. This research's primary innovation lies in its comprehensive and experimental comparison of time series prediction for REITs, incorporating TAIs into the feature set.

Previous works have demonstrated the effectiveness of using TAIs in the price prediction task. Specifically, [140, 141, 142] have demonstrated that using TAIs as additional features could improve the accuracy of stock price predictions. However, TAIs have not been utilised to predict real estate prices, making it crucial to demonstrate their potential advantages in price predictions, and by extension, in portfolios that incorporate REITs as one of their asset classes. In this chapter, we explore the performance of the same machine learning algorithms used in the previous chapter, namely Ordinary Least Squares (OLS) Linear Regression, Support Vector Regression, K-Nearest Neighbours, eXtreme Gradient Boosting (XGBoost), and Long Short Term Memory (LSTM) Neural Networks. In this way, we assess the potential improvement in prediction accuracy resulting from the incorporation of TAIs in the above-mentioned machine learning algorithms.

The remainder of this chapter is structured as follows: Section 6.2 outlines the methodology employed in this study; Section 6.3 details our experimental setup; Section 6.4 offers an in-depth discussion of the experimental outcomes, highlighting the application of machine learning and the proposed benchmarks to our dataset; finally, Section 6.5 summarises the key findings and provides concluding remarks for this research paper.

6.2 Methodology

Our methodology can again be broken down into two main steps: (i) price prediction, where we use different machine learning algorithms that include Technical Analysis Indicators (TAIs) in their feature set; and (ii) portfolio optimisation, where the predicted prices from the above step are used as input to a portfolio, whose weights are optimised by means of a Genetic Algorithm. The methodology that we follow in this chapter differs from the one explained in the previous chapter in the introduction of additional features in our regression problem, making it possible to further improve the performance of the algorithms used.

It is worth noting that the nature of data we used for this study is the same as in Chapter 5, as well as the data preprocessing steps we took into account (differencing and scaling), the machine learning algorithms used in our experiments, and the loss function chosen (Section 5.2.4). However, in this work we use additional features in the form of Technical Analysis indicators (TAIs) that we have not considered in the previous set of experiments. Thus, in this Section, we will describe the features used in our experiments (Section 6.2.1).

6.2.1 Features

To address our regression problem, we utilise two types of features: (i) past observations (i.e. ‘lags’) of the time series variable N_t; and (ii) Technical Analysis Indicators (TAIs).

Table 6.1: Example of feature selection (lagged observations).

t	N_t	N_{t-1}	N_{t-2}	N_{t-3}	N_{t-4}	N_{t-5}
t2	0.30	-	-	-	-	-
t3	0.70	0.30	-	-	-	-
t4	0.22	0.70	0.30	-	-	-
t5	1	0.22	0.70	0.30	-	-
t6	0	1	0.22	0.70	0.30	-
t7	0.70	0	1	0.22	0.70	0.30

Past Observations (Lags)

For the first type of features, we incorporate n past observations of N_t, i.e., $N_{t-1}, N_{t-2}, N_{t-3}$, ..., N_{t-n}, where the number of lags n is determined using the Akaike Information Criterion (AIC). For more detail, see Chapter 5. Table 6.1 provides an illustration of lagged observations for a selected number of lags ($n = 5$).

Technical Analysis Indicators (TAIs)

In addition to past observations, we also use five TAIs at each timepoint — Simple Moving Average (SMA); Exponential Moving Average (EMA); Moving Average Convergence/Divergence (MACD); Bollinger Bands; and Momentum — as suggested in [143, 141, 144]. These indicators help identify the short- and long-term trends of a time series, and thus can be effectively used for price prediction.

Simple Moving Average: The Simple Moving Average (SMA) is often used to predict future observations by providing an estimate of the level of a time series [145]. Mathematically, the SMA is the weighted average of the past T prices and can be represented as:

$$\text{SMA}(t) = \frac{\sum_{i=t-(T-1)}^{t} \left[N_i \right]}{T}, \tag{6.1}$$

where N_t is the normalised price at time i, and T is the number of time points considered. In Python, we calculate the SMA using the `rolling` method[1]. It is important to note that the period of interest T used for window-averaging is independent of the number of lags n, which determines the number of historical timepoints used for training purposes.

1. `https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rolling.html` Last accessed: June 2023.

Exponential Moving Average: The Exponential Moving Average (EMA) is a similar technique to the SMA, but with the key difference being that it considers all past observations, with weights that decay exponentially as a function of the distance in time between each observation and the current timepoint. More recent observations are given greater weight than older observations. The EMA is typically expressed through the following difference equation:

$$\text{EMA}(t) = \alpha N_t \ + \ (1-\alpha)\ \text{EMA}(t-1), \tag{6.2}$$

where α is a parameter representing the amount of weight decay applied at each timestep. α is calculated as $\alpha = 2/(T+1)$, where T is the period of interest. It can take any real value between 0 and 1, with lower values assigning more importance to past information, and higher values indicating less importance given to past prices. In Python, we calculate the EMA using the `ewm` method[2].

Moving Average Convergence/Divergence: The Moving Average Convergence/Divergence (MACD) indicator is a measure of the difference between a short-term and a long-term Exponential Moving Average (EMA). It is useful for identifying *bullish* moments (i.e. periods characterised by notable market price increase relative to historically lower or more stable prices), or *bearish* moments (i.e. periods characterised by notable market price decrease compared to historically higher or more stable prices). To calculate the MACD, we select an *H-day* denoting the start of a longer, 'historical' period (lasting until the present day), and an *R-day* (closer in time to the present day compared to the H-day), denoting the start of a shorter, more 'recent' period. The 'recent' period typically represents a period of interest, whose trend one wishes to compare against the longer, 'historical' period, in order to identify a change in market trend as compared to historical levels. This is done by first obtaining EMAs for both periods; the MACD is then obtained as the difference between the 'recent' EMA compared to the 'historical' one [146]:

Bollinger Bands: Bollinger Bands (BB) are defined as a price range around the Simple Moving Average (SMA) price at time t, obtained as follows: first, we compute the standard deviation of all observations (i.e., with respect to the SMA), within a period of interest T, where T is typically the same period used to calculate the SMA. This is then multiplied by a modifier D, which determines the number of standard deviations away from the mean we want to set our range to. This is represented mathematically as follows:

2. `https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.ewm.html` Last accessed: June 2023.

Table 6.2: Example of feature selection (TAIs).

t	SMA	EMA	MACD	Upper band	Lower band	Momentum
t2	-	0.15	-	-	-	-
t3	-	0.43	-	-	-	-
t4	-	0.32	0.19	0.53	0.28	-
t5	0.41	0.66	0.15	0.80	0.48	0.70
t6	0.64	0.33	0.18	0.59	0.23	0.70
t7	0.41	0.52	0	0.78	0.35	0.48

$$\text{BB}(t) = \text{SMA}(t) \pm D\sqrt{\left(\frac{1}{T}\right)\sum_{i=t-(T-1)}^{t}\left[N_i - \text{SMA}(t)\right]^2} \tag{6.3}$$

Bollinger Bands can help identify whether the current price level of a security has deviated significantly (i.e., more than D standard deviations) compared to its recent average and can also aid in predicting whether it might rise or fall back to that level.

Momentum: The Momentum indicator [147] is calculated as the difference between the price at time t and the price T periods ago, as shown below.

$$\text{Momentum} = N_t - N_{t-T} \tag{6.4}$$

By measuring the strength of a price trend, the Momentum can help predict the future direction of a time series.

Table 6.2 shows the TAIs computed for the preprocessed data described in Table 5.1. Specifically, we compute the 3-day Simple Moving Average (SMA), the Exponential Moving Average (EMA) with $\alpha = 0.5$, the Moving Average Convergence/Divergence (MACD) as the difference between the 3-day EMA and the 6-day EMA, the upper and lower Bollinger Bands using the 3-day SMA and the standard deviation of the 3-day SMA multiplied by 0.5, and the Momentum as the difference between the current price N_t and the price N_{t-T} that was observed $T = 5$ timepoints before t.

In total, we use these six TA-based features together with the lag-based features, resulting in $n+6$ features for our regression task.

6.3 Experimental Setup

The main goal of this work is to demonstrate the benefits of adding TAIs to the feature set of ML algorithms that predict REITs prices. To achieve this, we have broken down the above goal into two sub-goals: (i) to demonstrate that the use of TAIs leads to a significant reduction in the regression error, and (ii) to demonstrate that the use of TAIs leads to a significant improvement in the financial performance of a mixed asset portfolio that includes REITs.

The data used for this experiment set and the hyperparameter tuning adopted for the ML algorithms are the same as in Chapter 5. Thus in the following Section we will discuss the hyperparameter tuning used for the TAIs (Section 6.3.1) and the benchmarks employed in our experiments (Section 6.3.2).

6.3.1 Experimental Tuning of Hyperparameters

The hyperparameters of our machine algorithms were selected in the same way as for the experimental set described in Chapter 5. Specifically, the hyperparameter tuning took place through the *Grid Search* method. In that way, each dataset has their specific optimal parameters.

Regarding the genetic algorithm parameters, we selected the same values as in Chapter 5.

In order to select the optimal hyperparameters for the TAIs described in Section 6.2.1, we performed *Grid Search* tuning for each dataset. The best value for α in the EMA calculation was selected from the set $\{0.01,\ 0.05,\ 0.1\}$ [148]. The other hyperparameter values were decided on the basis of previous works [149, 150]. The selected values are shown in Table 6.3.

Table 6.3: TA hyperparameters.

Parameter	Indicator	Values
α	EMA	0.01, 0.05, 0.1
Short-day	MACD	20
Long-day	MACD	50
D	Bollinger bands	2

6.3.2 Benchmarks

As mentioned in the beginning of Section 6.3, our two sub-goals are to demonstrate the effectiveness of the use of TAIs in the price prediction task, and in the portfolio optimisation task. In order to investigate the benefits of using TAIs in the feature set, we employ and compare against several benchmarks, in accordance with the above two sub-goals. Section 6.3.2 presents the benchmarks chosen in relation to the regression task (four in total), and Section 6.3.2 presents the benchmarks chosen for the portfolio optimisation task (four in total).

Regression Task Benchmarks

Autoregression With ML In Section 6.2.1, we described the various features used in our regression problem. In order to assess the potential improvement in predictive accuracy from incorporating TAIs in addition to lagged values for predicting asset prices, we compare the performance of the five ML algorithms that employ both lagged prices and TAIs (proposed approach) against the five ML algorithms that use only lagged prices (i.e., without the TAIs), as is common practice in the REITs literature. The dependent variable is N_t, while the independent variables are past observations, specifically $N_{t-1}, N_{t-2}, ..., N_{t-T}$, *excluding* the TAIs.

HLTM, TBATS, and ARIMA The other regression benchmarks (i.e., HLTM, TBATS, and ARIMA) are presented in Chapter 5.

Portfolio Optimisation Benchmarks

Portfolio optimisation involves running a Genetic Algorithm on the price data predicted by our TAI-enhanced ML algorithms, in order to obtain appropriate weights for the different asset classes for each of the 90 assets that make up a portfolio. The quality of the resulting portfolios is then assessed on the basis of financial metrics calculated from the observed prices for that period. Furthermore, in order to assess the usefulness of TAIs in producing better portfolios, we compare the performance of the above, with portfolios that have been optimised with respect to price predictions obtained from the non-TAI-enhanced ML algorithm variants, just as in Section 6.3.2.

For completeness, we also benchmark our proposed approach against portfolios obtained on the basis of predictions made using the HLTM, TBATS, and ARIMA algorithms respectively, as these are well-known prediction algorithms, which are widely used in the financial literature. In all cases we evaluate the results in the test set in terms of three financial metrics, namely expected returns, expected risk and the Sharpe Ratio.

6.4 Results

In Section 6.4.1, we assess and compare the performance of the five ML algorithms – i.e., Ordinary Least Squares (OLS) Linear Regression, Support Vector Regression, K-Nearest Neighbours, eXtreme Gradient Boosting (XGBoost), and Long Short Term Memory (LSTM) Neural Networks — when making use of TAIs in their feature-set, against a) the same set of ML algorithms when using only lagged values but no TAIs as features, and b) the three conventional techniques outlined in Section 6.3.2 (i.e. HLTM, TBATS, and ARIMA), which also rely on lagged values exclusively for their function. In Section 6.4.2 we examine the implications of using TAIs in this manner, in the context of using the obtained algorithmic predictions to perform optimisation of a multi-asset portfolio using a Genetic Algorithm approach, and the extent to which this affects expected return, risk, and Sharpe Ratio values in the resulting portfolios. In Section 6.4.3, we further analyse the importance of each feature in two distinct ways, by using the SHAP and SAGE algorithms, which are metrics of feature quality that build on the concept of Shapley values [151]. Finally, Section 6.4.4 examines the computational times involved for the algorithms used, and Section 6.4.5 offers a short discussion on the insights gained from the experimental results.

6.4.1 ML Prediction

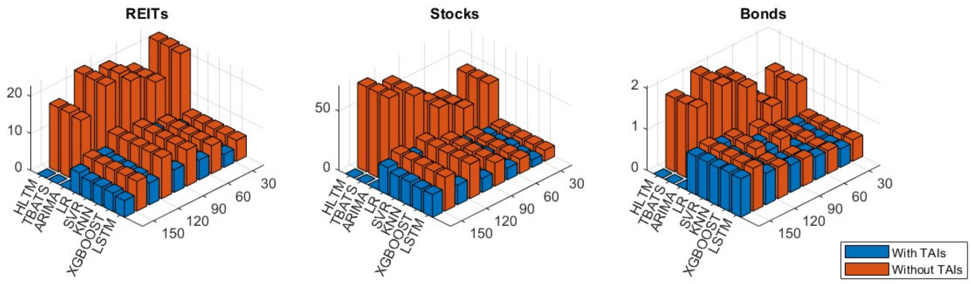
We evaluate and compare the performance of the proposed approaches and benchmarks, by reporting the RMSE mean and standard deviation per asset class, for each algorithm across all markets, where the RMSE for each dataset is obtained as per Section 5.2.4.

Figure 6.1 shows RMSE descriptive statistics for REITs in the case of out-of-sample (top) and one-day-ahead prediction (bottom) over a 30-, 60-, 90-, 120-, and 150-day period[3]. We note that, in the case of *out-of-sample* prediction, the average RMSE is consistently lower for algorithms that use TAIs when compared to the algorithms that use lagged prices only. This is the case across all periods (30, 60, 90, 120, and 150 days). It is also worth noting that the improvements in RMSE means tend to be large. E.g. in the 30-day period, we note a reduction from an 'RMSE means' average of around 5.5 (i.e. when averaging the individual RMSE means of each non-TAI model), to an average of 4.0 when TAIs are added into the feature set. We also note even larger improvements in other isolated instances; e.g. the 90-day OLS features a reduction from around 10 to 6 (i.e. an error reduction of ≈39%), and the 120-day LSTM features a reduction from approximately 11 to 6 (i.e. an error reduction of $\approx 46\%$). In addition, it is worth noting that the performance of the conventional time-series benchmarks (HLTM, TBATS, ARIMA)[4] is generally poor by comparison, and consistently outperformed by the machine learning algorithms, regardless of whether TAIs are included in the feature set or not.

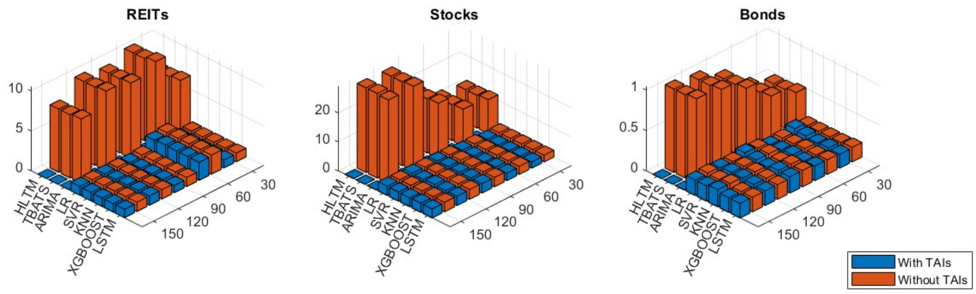
A similar picture can be observed with the *one-day-ahead* prediction results. The performance of the algorithms that use TAIs tends to be better than the ones without TAIs, with the only exception being the 30-day SVR and XGBOOST, and the 120-day XGBOOST entries. However, it is worth noting that, while for the out-of-sample results, the introduction of TAIs led to large reductions in error, this reduction is not as impressive in the case of one-day-ahead predictions. This is to be expected, since this method predicts the next day's value using only

3. For reference, we provided Tables C.1 - C.2 with detailed results for REITs, stocks, and bonds in Appendix C
4. As mentioned in Section 6.2, due to the autoregressive elements of HLTM, TBATS, and ARIMA, they cannot use TAIs in their feature set. Hence the relevant rows under the 'With TA' headings in Table C.1, and across all remaining tables in this paper, are empty.

real — rather than predicted — values in the test period, and therefore the errors are always going to be much smaller. In fact, this is the case regardless of whether we use TAIs or not. As a result, the margin for improvements is also small. Nevertheless, the fact remains that when using TAIs we still observe consistent average RMSE improvements.



(a) Out-of-sample RMSE results for REITs, stocks, and bonds for 8 algorithms and 5 horizons.



(b) One-day-ahead RMSE results for REITs, stocks, and bonds for 8 algorithms and 5 horizons.

Figure 6.1: Comparison of RMSE results

We can observe a similar picture for stocks results. Machine learning algorithms that use TAIs in their feature set show consistently lower RMSE average values for both out-of-sample and one-day-ahead predictions, across all periods (30, 60, 90, 120, and 150 days). It is also worth noting here that the average RMSE values tend to be higher for stocks than for the other two asset classes; this can be explained by the more volatile nature of stock data, which makes it much harder to predict accurately.

Finally, we notice relatively low RMSE average values for bonds. This is due to the nature of such asset class, featured by low volatility, which makes it much easier to predict. With regards to the comparison of results when using TAIs, we can again observe that the introduction of TAIs leads to consistent improvements in the out-of-sample results, whereas in the one-day-ahead case, due to the very low error values involved, the results are more mixed, with TAI algorithms occasionally being marginally outperformed by their respective non-TAI ones.

To determine whether there is a statistically significant difference between the distributions of RMSE scores resulting from TAI versus non-TAI ML algorithms, we performed a Kolmogorov-Smirnov (KS) test at the 5% significance level across all asset classes. The null hypothesis was that the compared RMSE distributions come from the same continuous distribution. Since we conducted five comparisons (one for each considered period: 30-, 60-, 90-, 120-, and 150-day periods), we adjusted the α value using the Bonferroni correction, resulting in an adjusted threshold of 0.01 $(0.05/5 = 0.01)$.

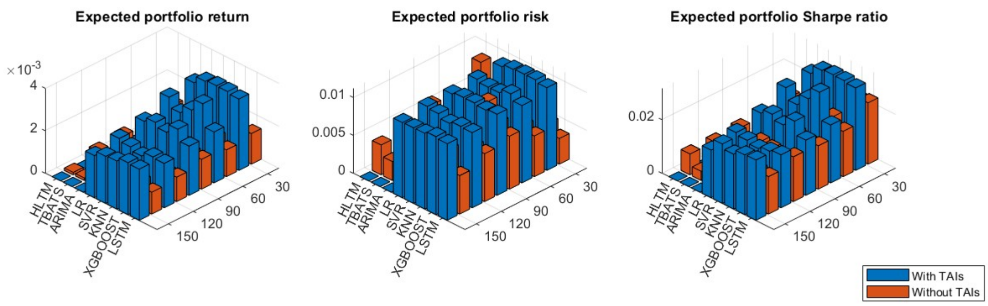
For out-of-sample predictions, we obtained KS test p-values of less than 0.001 in all cases, which are much lower than the adjusted α threshold of 0.01. Specifically, for the 30-, 60-, 90-, 120-, and 150-day periods, the p-values were 1.02×10^{-9}, 7.14×10^{-7}, 5.66×10^{-8}, 4.60×10^{-8}, and 2.61×10^{-7}, respectively. These results strongly suggest that the introduction of TAIs leads to a significant reduction in RMSE. To further quantify the practical significance of these differences, we calculated effect sizes using Cohen's d. The effect sizes for the significant periods were as follows: $d = 0.82$ (large effect) for the 30-day period, $d = 0.76$ (large effect) for the 60-day period, $d = 0.85$ (large effect) for the 90-day period, $d = 0.78$ (large effect) for the 120-day period, and $d = 0.80$ (large effect) for the 150-day period. These large effect sizes confirm that the differences observed are not only statistically significant but also practically meaningful.

Conversely, in the case of one-day-ahead predictions, the KS test p-values were non-significant (1.04×10^{-1}, 7.37×10^{-1}, 6.17×10^{-1}, 3.07×10^{-1}, and 1.48×10^{-1}, respectively), indicating no significant difference in RMSE distributions. As a result, effect sizes were not calculated for these comparisons. However, as previously mentioned, due to the nature of one-day-ahead predictions, RMSE values tend to be very small, making it more challenging to achieve statistically significant results despite the observed small reductions in RMSE.

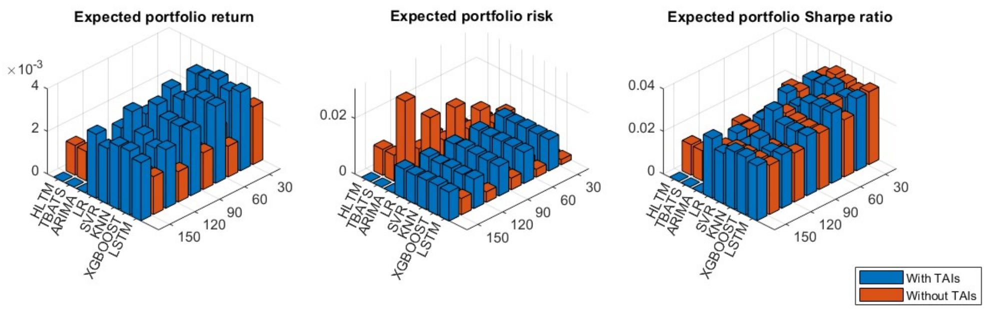
In conclusion, we observed that the RMSE distributions tend to be lower on average and less volatile for ML algorithms that use TAIs, than for benchmark algorithms which show larger and more variable residuals in the case of out-of-sample predictions. In general, we noticed that the magnitude of the reduction in RMSE mean values from TAIs can be as high as 45%. We also observed that the lowest average RMSE values were obtained in bonds, followed by REITs, and stocks. This can be explained by the lower volatility of bond prices as seen in Section 5.3.1. In the case of REITs, the RMSE distributions tend to have higher averages than for bonds but lower than for stocks. This is likely due to the properties of REIT prices in terms of risk and return that are usually placed between that of bonds and stocks in terms of risk and return. According to the KS test results, there is a significant reduction in RMSE mean values when adopting an out-of-sample methodology.

6.4.2 Portfolio Optimisation

This section contains the results of the Genetic Algorithm (GA) applied to portfolio allocation, which takes into account a transaction cost of 0.02%. The GA was used to generate 100 optimised portfolios per algorithm considered. For each generated portfolio, the optimised weights were used to calculate the expected return, expected risk, and expected Sharpe Ratio for the portfolio. These were then pooled over all generated portfolios, to create and analyse the distributions of expected returns, expected risks, and expected Sharpe Ratios respectively. In this Section, we compare the performance of our proposed approaches, i.e. of ML models that utilise TAIs as additional features, to benchmarks, which consist of portfolios built using ML models, as well as HLTM, TBATS, and ARIMA.



(a) Out-of-sample portfolio results for REITs, stocks, and bonds for 8 algorithms and 5 horizons.



(b) One-day-ahead portfolio results for REITs, stocks, and bonds for 8 algorithms and 5 horizons.

Figure 6.2: Comparison of portfolio results

Figure 6.2 shows descriptive statistics for expected return distributions (left), expected risk distributions (middle), and expected Sharpe ratio distributions (right) obtained from the GA portfolio optimisation for a 30-, 60-, 90-, 120-, and 150-day holding period, for the out-of-sample (top) and one-day-ahead (bottom) method. For HLTM, TBATS, and ARIMA models, the average expected return values appear to be lower compared to the proposed approaches. In the case of the one-day-ahead predictions, the average values of the expected return distributions also improve when introducing TAIs. For instance, the best result is observed for the KNN algorithm (around 4×10^{-3}) which shows an improvement of almost 175% when adding TAIs. The HLTM, TBATS, and ARIMA algorithms show lower expected return values with respect to the ML algorithms that use TAIs. We observe similar findings for out-of-sample and one-day-ahead methods.

To compare the expected return distributions obtained via TAI models versus non-TAI models from ML algorithms that use TAIs as additional features and those obtained from algorithms that use lagged values only, we conducted a Kolmogorov-Smirnov (KS) test at the 5% significance level. Here again, the null hypothesis assumes that the compared return distributions arise from the same continuous distribution. We performed five comparisons (one for each prediction period, i.e., 30, 60, 90, 120, and 150 days), and to account for multiple comparisons, we again applied Bonferroni's correction by adjusting the alpha value to $0.05/5 = 0.01$.

For out-of-sample predictions, the KS test produced p-values of 7.17×10^{-28}, 3.59×10^{-27}, 8.24×10^{-24}, 1.12×10^{-20}, and 3.27×10^{-17} for 30-, 60-, 90-, 120-, and 150-day periods, respectively, which are much lower than the adjusted significance level of 0.01, suggesting that the use of TAIs leads to a significant improvement in the expected return distributions. The effect sizes for these periods, calculated using Cohen's d, were $d = 0.85$ (large effect) for the 30-day period, $d = 0.78$ (large effect) for the 60-day period, $d = 0.82$ (large effect) for the 90-day period, $d = 0.76$ (large effect) for the 120-day period, and $d = 0.81$ (large effect) for the 150-day period. These large effect sizes confirm that the observed differences are not only statistically significant but also practically meaningful.

Similarly, for one-day-ahead predictions, the KS test produced p-values of 5.11×10^{-33}, 3.71×10^{-36}, 3.97×10^{-25}, 3.59×10^{-27}, and 5.11×10^{-33} for 30-, 60-, 90-, 120-, and 150-day periods, respectively, leading to the same conclusion as for the out-of-sample scenario. The effect sizes for these periods were $d = 0.89$ (large effect) for the 30-day period, $d = 0.87$ (large effect) for the 60-day period, $d = 0.84$ (large effect) for the 90-day period, $d = 0.83$ (large effect) for the 120-day period, and $d = 0.88$ (large effect) for the 150-day period, further supporting the practical significance of the improvements in expected return distributions when using TAIs.

Regarding the average expected risk we observe improvements in the predictive performance of all ML algorithms when including TAIs in the regression problem for all periods, both for out-of-sample and one-day-ahead approaches. As we can observe, in the case of out-of-sample prediction, there is an average increase of around 187% as we add TAIs for a 30-day prediction period (with a decrease in the case of LSTM of around 20%), which drops to around 113% for a 60-day prediction period, to around 70% for a 90-day prediction period, to around 44% for a 120-day prediction period, and rises to 72% for a 150-day prediction period. Similarly, in the case of one-day-ahead predictions, the average expected portfolio risk again tends to be lower when not using TAIs as features. Lastly, it is worth noting that for the first time in our study, the HLTM, TBATS, and ARIMA algorithms outperform the ML algorithms, as they show relatively low average (around 2×10^{-3}).

We compared the expected risk value distributions using a KS test at the 5% significance level, similar to our comparison of the expected return distributions. As with the expected return distributions, the null hypothesis was that the compared risk distributions came from the same continuous distribution. We conducted five comparisons, one for each period, and accounted for multiple comparisons by adjusting the alpha value to 0.01 using Bonferroni's correction.

For the out-of-sample predictions, the KS test produced p-values of 7.17×10^{-28}, 1.32×10^{-38}, 7.19×10^{-43}, 3.97×10^{-43}, and 3.88×10^{-41} for 30-, 60-, 90-, 120-, and 150-day periods, respectively. These p-values were all below the adjusted significance level of 0.01, indicating that using TAIs resulted in a significant increase in the expected risk distributions. The effect sizes for these periods, calculated using Cohen's d, were $d = 0.90$ (large effect) for the 30-day period, $d = 0.88$ (large effect) for the 60-day period, $d = 0.92$ (large effect) for the 90-day period, $d = 0.91$ (large effect) for the 120-day period, and $d = 0.93$ (large effect) for the 150-day period. These large effect sizes confirm that the observed increases in expected risk are both statistically significant and practically meaningful.

Similarly, for one-day-ahead predictions, the KS test produced p-values of 1.23×10^{-44}, 3.88×10^{-41}, 3.64×10^{-41}, 2.76×10^{-40}, and 3.88×10^{-41}, respectively, also indicating a statistically significant difference in the expected risk distributions. The effect sizes for these periods were $d = 0.95$ (very large effect) for the 30-day period, $d = 0.94$ (very large effect) for the 60-day period, $d = 0.93$ (very large effect) for the 90-day period, $d = 0.92$ (very large effect) for the 120-day period, and $d = 0.96$ (very large effect) for the 150-day period, reinforcing the significance of the increased expected risk when using TAIs.

From the Sharpe ratio results, we can observe that the proposed algorithms tend to outperform the benchmarks for all periods in the case of out-of-sample predictions. We can observe an average increase in the average Sharpe ratio values of approximately 60% when incorporating TAIs in the ML algorithms. This highlights the effectiveness of adopting TAIs when predicting financial prices through an out-of-sample, N-day-ahead methodology. In the case of one-day-ahead predictions, we observe that the Sharpe Ratio values obtained from the proposed approaches tend to be closer on average with respect to the benchmark approaches.

Similarly to what we have done for the expected return and risk distributions, we conducted a KS test to compare the expected distributions of Sharpe Ratio values. Since we are making multiple comparisons, we again adjusted the significance level according to Bonferroni's correction.

For out-of-sample predictions, the KS test generated p-values of 7.17×10^{-28}, 3.97×10^{-25}, 2.77×10^{-21}, 1.12×10^{-20}, and 3.96×10^{-16} for 30-, 60-, 90-, 120-, and 150-day periods, respectively. All of these p-values were below the adjusted significance level of 0.01, indicating that using TAIs resulted in a significant improvement in the expected Sharpe Ratio distributions. The effect sizes for these periods, calculated using Cohen's d, were $d = 0.84$ (large effect) for the 30-day period, $d = 0.81$ (large effect) for the 60-day period, $d = 0.78$ (large effect) for the 90-day period, $d = 0.76$ (large effect) for the 120-day period, and $d = 0.80$ (large effect) for the 150-day period. These large effect sizes confirm that the improvements in Sharpe Ratio distributions are both statistically significant and practically meaningful.

For one-day-ahead predictions, the KS test produced p-values of 0.359, 0.364, 0.824, 0.183, and 0.168 for the 30-, 60-, 90-, 120-, and 150-day periods, respectively. In this case, the p-values are above the adjusted significance level, indicating that there is no statistically significant difference in the Sharpe Ratio distributions. As the p-values suggest non-significance, effect sizes were not calculated for these comparisons.

In summary, the above results confirm that using TAIs in ML can lead to an improvement in the risk-adjusted portfolio performance with room for improvement of up to 66.10% in the case of out-of-sample predictions, and up to 20.07% in the case of one-day-ahead predictions. Such improvement is mostly explained by the average increase in the average portfolio returns resulting from the incorporation of TAIs. From the statistical tests performed, we can confirm that the results obtained from our experiments are statistically significant.

6.4.3 Shapley Values

In the previous section, we observed that incorporating TAIs as additional features in our regression problem can significantly reduce the error rate and improve portfolio performance. In this section, we will analyse the relative importance of these features by means of the SHAP [152] and SAGE [153] algorithms, which produce metrics describing different aspects of feature quality, and are thus widely used for model explainability in a variety of machine learning contexts [154, 155].

Both SHAP and SAGE build on the concept of Shapley values [151]; in traditional co-operative game theory, Shapley values reflect a partitioning of the overall output of a group (or 'grand-coalition'), which expresses this output as the sum of the individual contributions of its members, obtained by quantifying the average marginal contribution of each member across all possible member combinations (i.e. 'sub-coalitions'). In the context of assessing feature quality in machine learning algorithms, a Shapley value treatment of an algorithm's features provides an assessment of how much each feature contributes to a measure of interest in relation to the model. However, calculation of true marginal contributions for obtaining classical Shapley values can be a computationally prohibitive step, and therefore algorithms like SHAP and SAGE rely on computationally efficient variants, which involve approximating marginal contributions as deviations of conditional distributions from practical prior baselines.

In the literature, SHAP primarily tends to be used in 'explainability' contexts; given a prediction, it measures the extent to which each feature has contributed to the prediction. However, under the assumption that important features will be given larger weights in the final models following training, and that therefore the average influence of a feature over all predictions reflects its weighting in the model to a large extent, this can then be interpreted as a proxy measure for evaluating feature importance. Conversely, SAGE measures feature quality more directly; instead of making assumptions about the model's internals, it measures the influence of each feature on the evaluation metric directly[5].

In order to have a clear view of the marginal contribution of each feature in each case, we present them here as percentages. To achieve this, we divided the average SHAP (or SAGE) value of each feature by the sum of SHAP (or SAGE) values for all features. Figure 6.3 presents the percentage SHAP (on the left side) and SAGE values (on the right side) calculated on the testing set for each feature, across all TAI-based algorithms using the out-of-sample method, displayed for each asset class and considered period. Regarding the SHAP values, we can observe that the relevance of prices lagged by two or more days tends to be lower compared to the other features. For REITs, TAIs combined account for 82% for a 30-day testing period, 72% for a 60-day period, 69% for a 90-day period, 73% for a 120-day period, and 77% for a 150-day period; while $N_1 + N_2$ account for a further 14% for a 30-day period; 26% for a 60-day period, 29% for a 90-day period, 23% for a 120-day period, and 15% for a 150-day period; and then the remaining lags only account for 4% for a 30-day period, 2% for a 60-day period, 2% for a 90-day period, 4% for a 120-day period, and 8% for a 150-day period. Similarly for stocks and bonds, TAIs account for 83% for a 30-day period, 66.5% for a 60-day period, 70% for a 90-day period, 75% for a 120-day period, 78% for a 150-day period; $N_1 + N_2$ account for 13% for a 30-day period, 31% for a 60-day period, 28% for a 90-day period, 21% for a 120-day period, 17% for a 150-day period; and the remaining lags only account for 3.75% for a 30-day period, 2.25% for a 60-day period, 2% for a 90-day period, 4% for a 120-day period, and 5% for a 150-day period.

Regarding the SAGE values, for REITs, we can observe that the combined contribution for TAIs tends to be 80% for a 30-day period, 60% for a 60-day period, 67% for a 90-day period, 71% for a 120-day period, and 78% for a 150-day period; while the combined contribution for $N_1 + N_2$ is 13% for a 30-day period, 18% for a 60-day period, 18% for a 90-day period, 19%

5. Note that, when relying on the RMSE for model evaluation, SAGE actually uses the negative RMSE internally instead, such that Shapley values denoting important features end up positive (with negative values denoting harmful features respectively)

for a 120-day period, and 17% for a 150-day period; and the contribution of the remaining lags is 7% for a 30-day period, 22% for a 60-day period, 15% for a 90-day period, 10% for a 120-day period, and 5% for a 150-day period. Regarding stocks and bonds, the combined contribution for TAIs tends to be 77% for a 30-day period, 63% for a 60-day period, 69.5% for a 90-day period, 73.5% for a 120-day period, and 68% for a 150-day period; while the combined contribution for $N_1 + N_2$ is 13.5% for a 30-day period, 24% for a 60-day period, 20.5% for a 90-day period, 19% for a 120-day period, and 15% for a 150-day period; and the contribution of the remaining lags is 9.5% for a 30-day period, 13% for a 60-day period, 10% for a 90-day period, 7.5% for a 120-day period, and 17% for a 150-day period.

The combined SHAP and SAGE findings above may explain the substantial improvement in terms of RMSE, achieved by employing ML algorithms making use of TAIs in their feature-set (see Section 6.4.1). It is worth noting that, in the current literature, commonly employed approaches for financial forecasting currently tend to rely on lagged observations exclusively [156, 157]. However, the inclusion of TAIs is beneficial because they are specifically designed to capture patterns and trends in the data that may not be immediately apparent from raw price data or simple lagged observations. For example, momentum indicators can effectively identify whether a trend is likely to continue or reverse, which is crucial for financial forecasting. These indicators incorporate more complex relationships between price movements, providing richer insights than using lagged prices alone. Although LSTM networks are powerful for learning from sequential data, they might fail to capture more complex aspects of financial time series. TAIs, on the other hand, offer domain-specific information that directly targets financial patterns such as trend continuation or reversal, which may complement the LSTM's ability to learn sequential patterns. Therefore, the combination of TAIs with advanced machine learning models can better capture underlying market dynamics and improve forecasting accuracy, outperforming models that rely solely on lagged price data.

6.4.4 Computational Times

As we have seen in Chapter 5, most algorithms have comparable computational times. HLTM, TBATS, and ARIMA typically took 0.168 minutes to execute on average, while LR, SVR, and KNN took between 0.2 and 0.3 minutes. LSTM had the highest computational cost at 1.818 minutes. However, this runtime difference is not significant since these algorithms are usually run offline, and only their models are used in real-time applications.

In Chapter 5, it was observed that most algorithms showed comparable computational times. Specifically, HLTM, TBATS, and ARIMA typically took an average of 0.168 minutes to execute. On the other hand, LR, SVR, and KNN required between 0.2 and 0.3 minutes. LSTM had the longest computational time at 1.818 minutes. However, this difference in runtime is not considered significant, as these algorithms are primarily run offline, and only their models are utilised in real-time applications.

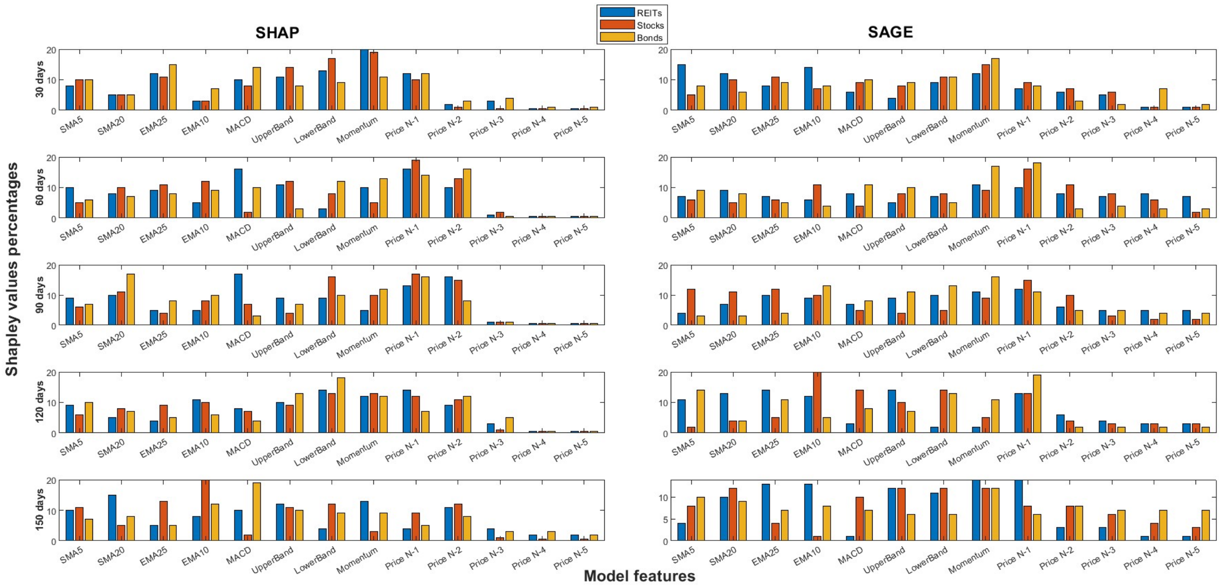


Figure 6.3: Shapley average value for each asset class and feature classified by period considered.

The GA, on average, required approximately 10.92 seconds per run. It was noted in Chapter 5 that GAs are highly parallelisable, meaning their computational times can be further reduced through parallelisation processes. This suggests that the efficiency of the GA can be improved by distributing the workload across multiple processing units simultaneously, thereby accelerating the optimisation process[128].

6.4.5 Discussion

Our experiments aimed at demonstrating that the inclusion of Technical Analysis Indicators (TAIs) as additional features could significantly reduce the error rate in predicting the time series of REITs, stocks, and bonds. In the previous sections, we observed improvements in the average error rate for both out-of-sample and one-day-ahead predictions. These tend to be more noticeable in the case of REITs and stocks, whereas, in the case of bonds, the inclusion of TAIs appears not to change the RMSE distribution significantly, as shown by the KS test results. This might be explained by the low variability of the bond time series which leads to already low error rates in the prediction. On the other hand, we noticed a lower standard deviation in the RMSE distributions in the case of stocks and bonds for all prediction periods considered. This indicates a higher chance of observing RMSE values closer to the average.

The second aim of our experiments was to show the improvements in the multi-asset portfolio performance caused by the use of TAIs in the prediction task. We noticed significant improvements in the risk-adjusted performance of a portfolio composed of REITs, stocks, and bonds for all holding periods considered with respect to a portfolio built using predictions resulting from lagged prices only. As demonstrated by the KS test results, there is statistically significant difference in the case of portfolios built using out-of-sample predictions, while the portfolio performance tends not to differ significantly in the case of one-day-ahead predictions. We also observed that the use of TAIs tends to increase the expected portfolio return, and at the same time, to increase the expected portfolio risk. This generates a trade-off between increased expected return and reduced increased expected risk. However, when investors make investment decisions, they tend to look at the expected Sharpe Ratio as an aggregate metric, rather than solely focusing on expected return or risk as isolated metrics.

Finally, we discussed the influence that each feature has on the final prediction, as well as the contribution of each feature to overall model error, for each of the asset classes and evaluation periods used. As we observed, in terms of explaining predictions the TAIs tend to overshadow the lagged prices as features. In other words, this suggests that the future trend of such time series plays a crucial role in reducing the prediction error rate.

6.5 Summary

In this study, we focused on the problem of predicting out-of-sample and one-day-ahead prices of REITs, stocks and bonds by using five ML algorithms and Technical Analysis Indicators (TAIs) for five prediction periods (30-, 60-, 90-, 120-, and 150-day).

From the above findings, we can conclude the following.

The use of TAIs generates a reduction in the average and volatility of RMSE distributions for the asset classes considered. We observed that the ML algorithms that incorporate TAIs as additional features tend to perform better than the ML algorithms that used lagged prices as unique features as well as HLTM, TBATS, and ARIMA. This finding indicates that the accuracy of REITs predictions tends to be higher when including TAIs which are able to express the trend of a time-series.

The risk-adjusted portfolio performance of the resulting portfolio tends to improve. From our experimental findings, we could notice that the inclusion of TAIs leads to an increase in the Sharpe ratio values as a result of the increase in the expected return values. Such result is important because the use of TAIs allows investors to make better decisions when combining REITs with other asset classes in a short-term portfolio.

TAIs tend to show a greater relevance compared against the lagged prices. When analysing the SHAP and SAGE average values, TAIs tend to be more influential than lagged prices in terms of explaining the reduction in the error rate. This finding explains the increase in the prediction accuracy demonstrated in this study.

Chapter 7

Optimising Mixed-Asset Portfolios Including REITs Using ML and TA Indicators

7.1 Introduction

In the previous chapters, we discussed two main points. First, using predictions about future prices (instead of just looking at past data) improves the performance of a mixed investment portfolio that includes real estate. Second, including additional features such as Technical Analysis Indicators (TAIs) makes the predictions more accurate.

In this chapter, our primary motivation is to highlight the additional benefits of having real estate in a mixed-asset portfolio. Our focus is on predicting the future prices of REITs, stocks, and bonds. The main idea is to show why using predictions, especially ones improved with TAIs, is important when dealing with diverse portfolios that include real estate. Previous works in the literature [158, 159, 160] have explored the role of real estate investments in a mixed-asset portfolio by relying on past observation, the research presented in this chapter offers a valuable contribution by incorporating predictions resulting from both past prices and technical indicators.

The data sets used for our experiments are the same as those used in previous chapters, namely daily closing prices for the period between January 2017 and January 2022 for the UK, US, and Australian market. As we have done previously, we consider the REIT, stock, and bond market for our analysis.

The rest of this chapter is organised as follows. Section 7.2 explains the methodology used in this study. The results of our experiments are presented in Section 7.3, where we provide a detailed discussion of the results obtained by predicting asset prices using LSTM, and by running a GA to optimise our portfolios. Finally, Section 7.4 summarises the conclusions of the study.

7.2 Methodology

Our experiments aim to provide evidence that a mixed-asset portfolio including real estate can significantly outperform a mixed-asset portfolio not including real estate. This aim can be broken down into two subtasks: (i) use LSTM (which is the algorithm that provided the best results in Chapter 6) to predict the prices of REITs, bonds, and stocks, and (ii) use these predictions as an input to a genetic algorithm, which is going to optimise the weights of all assets in the portfolio.

Before applying the LSTM algorithm, we first needed to take several data pre-processing steps as previously discussed in Section 5.2.2 (i.e., first-order differencing and scaling). The features used in these experiments are the same as in Chapter 6 (i.e., lagged values and technical analysis indicators) and the loss function considered is the same as in Chapter 5 (i.e., root mean squared error). In the same way, the LSTM and GA algorithm used follow the same implementation as in the previous chapters.

7.3 Results

In this section, we examine the experimental results in the form of RMSE distributional statistics (Section 7.3.1), and summary statistics regarding the GA portfolio optimisation results (Section 7.3.2). As mentioned in the previous chapters, all results presented in this research are expressed as daily results. So when, for example, we present a seemingly “low” return of around 0.03%, its annual equivalent would be around 11.6%. [1]

Table 7.1: RMSE and Sharpe ratio distributional statistics. Values in bold represent best results for each statistic.

	RMSE			Expected Return		
Metric	**Without REITs**	**With REITs**	**% Difference**	**Without REITs**	**With REITs**	**% Difference**
Mean	36.29	$\mathbf{19.44}$	-46.43%	5.41×10^{-4}	$\mathbf{8.99 \times 10^{-4}}$	66.06%
Std Dev	146.15	$\mathbf{71.93}$	-50.79%	6.08×10^{-5}	$\mathbf{2.79 \times 10^{-5}}$	-54.11%
	Expected Risk			**Sharpe Ratio**		
Metric	**Without REITs**	**With REITs**	**% Difference**	**Without REITs**	**With REITs**	**% Difference**
Mean	5.54×10^{-3}	$\mathbf{3.70 \times 10^{-3}}$	-33.21%	7.26×10^{-3}	$\mathbf{1.48 \times 10^{-2}}$	103.71%
Std Dev	4.04×10^{-4}	$\mathbf{1.48 \times 10^{-4}}$	-63.28%	$\mathbf{5.33 \times 10^{-4}}$	5.55×10^{-4}	4.17%

1. Annualised Return $= [(\text{Daily Return} + 1)^{365} - 1] \times 100 = 11.6\%$.

7.3.1 RMSE

First, we compare the accuracy of predictions between two scenarios, one that includes REITs, and one that does not include REITs. Table 7.1 shows the summary statistics for two RMSE distributions, one for each of the two previously mentioned scenarios. For each of those distributions, we analyse the mean and standard deviation. As we can observe, the RMSE distribution in the first scenario shows lower RMSE average value compared to the second scenario, with a percentage difference of -46.43%. This indicates that including REITs in the analysis improves the accuracy of predictions. Furthermore, the RMSE distribution for the first scenario shows a noticeably lower standard deviation value compared to the second scenario, with a reduction of 50.79%. This suggests that incorporating REITs in the analysis leads to more accurate predictions with reduced variability.

In order to compare the RMSE distributions obtained, we performed a Kolmogorov-Smirnov (KS) test at the 5% significance level. The null hypothesis is that the compared RMSE distributions belong to the same continuous distribution. The test results yielded an adjusted p-value of 1.94×10^{-45}, indicating a statistically significant difference between the two distributions. To further quantify the magnitude of this difference, we calculated the effect size using Cohen's d, which resulted in a value of $d = 0.85$, indicating a large effect size. This suggests that the difference in RMSE distributions is not only statistically significant but also practically meaningful.

In summary, when analysing the RMSE values, it becomes evident that incorporating REITs in the analysis improves the accuracy of predictions in terms of mean and standard deviation. The scenario of incorporating REITs consistently outperforms the scenario of not including REITs, suggesting that including REITs provides more precise predictions. From the KS test results, we observed that such difference is statistically significant.

7.3.2 GA Portfolio Optimisation

After having analysed the RMSE distributional statistics, we examine the expected portfolio performance for the above-mentioned scenarios. First, we examine the expected return distributions. From Table 7.1, we can notice an increase in the expected return average of around 66.06%. We also notice a 54.11% reduction in the volatility of the expected return distribution, which indicates an increased concentration of values around the mean. This implies that including REITs in a mixed-asset portfolio might improve the overall portfolio return with a reduced volatility.

We also observe that the average expected risk tends to decrease when including REITs with a magnitude of around 33.21%. This implies that investing in REITs allows to reduce the overall portfolio risk. On the other hand, we notice that the standard deviation of the expected risk values tends to decrease with a magnitude of around 63.28%, which indicates an increased concentration of risk values around the mean.

Finally, we observe that the average Sharpe ratio increases when incorporating REITs, with a percentage difference of 103.71%. We also notice a slight increase in the volatility of 4.17%. This suggests that including REITs tends to have a marginal impact on the volatility of the risk-adjusted returns.

In order to compare the Sharpe ratio distributions obtained, we again performed a Kolmogorov-Smirnov (KS) test at the 5% significance level. Since we are making three comparisons—one for each metric (i.e., portfolio return, risk, and Sharpe ratio)—we adjusted the p-values according to the Bonferroni correction (i.e., $0.05/3 = 0.0167$). The test results yielded an adjusted p-value of 1.55×10^{-45} for all the considered metrics, indicating a statistically significant difference in the compared distributions. To further assess the practical significance of these differences, we calculated the effect size using Cohen's d, which resulted in values of $d = 0.82$ (large effect) for portfolio return, $d = 0.79$ (large effect) for risk, and $d = 0.85$ (large effect) for the Sharpe ratio. These large effect sizes suggest that the differences in the distributions are not only statistically significant but also practically meaningful.

In summary, when considering the portfolio return, risk, and Sharpe ratio distributions, we observe that including REITs in the analysis has a positive impact on the portfolio performance. It significantly improves the risk-adjusted distributions, as a result of an increased portfolio return and a reduced portfolio risk. The effect of REITs on risk-adjusted return distributions is significant, as shown by the KS test results.

7.4 Summary

In our work, we evaluated the performance of a portfolio including REITs by comparing it against a portfolio that does not include REITs.

From our experimental results, we noticed the following.

The RMSE average tends to be lower when including REITs in the analysis. We demonstrated that the predictions of time-series data tend to be more accurate on average when considering REITs data which is mainly explained by the lower volatility of REITs prices compared to other data, especially in the case of stock investments.

The inclusion of REITs in a mixed-asset portfolio leads to a greater Sharpe ratio. From our experimental results, we noticed that the average Sharpe ratio of a portfolio that includes REITs doubles the average Sharpe ratio of a portfolio that does not include REITs. This suggests that including REITs in a portfolio including bonds and stocks can mitigate the greater portfolio risk caused by including stock investments.

While our results show that adding real estates to investment portfolios can have a positive effect under the diversification perspective, further research can be done on different countries to further explore the opportunities of investing in real estate. Another opportunity for further research might be to extend the holding period for real estate portfolios.

Chapter 8

Conclusion

In this thesis, we focused on applications of machine learning to the fields of financial forecasting and portfolio optimisation. Specifically, we used five machine learning algorithms – i.e., Ordinary Least Squares Linear Regression, Support Vector Regression, k-Nearest Neighbours Regression, Extreme Gradient Boosting, and Long/Short-Term Memory Neural Networks – to predict the prices of three asset classes – i.e., REITs, stocks, and bonds. We then used these predictions to optimise weights in a mixed-asset portfolio made of the above-mentioned asset classes. In this chapter, we present the conclusions from our experiments. Each of the following sections is structured as follows: (i) first, we explain the motivation behind each study; (ii) second, we describe the novelty of the presented research; and (iii) third, we present the conclusions of each work. Finally, we discuss possible opportunities for further research in Section 8.5.

8.1 Summary of Chapter 4

8.1.1 Motivation of the Presented Research

In Chapter 4, we conducted experiments to explore the potential advantages of using price predictions instead of historical data in terms of final portfolio performance by optimising the weights of a mixed-asset portfolio through test set data – rather than training data.

One of the main limitations of the previous works in the literature is that most of them optimise portfolios including real estate by using historical data. A potential limitation of this approach is that prices in the training set – i.e., historical data – might differ significantly from prices in the testing set, thus worsening the overall portfolio performance. In fact, a good portfolio optimisation strategy mainly depends on the accuracy of the price predictions. Therefore, in Chapter 4 we evaluated the potential advantage of using price predictions in the portfolio optimisation task. To simplify this analysis, we assumed perfect price predictions in the test set, according to a perfect foresight approach. In that way, the optimisation of weights takes place in the test set. The main idea behind this approach is that, if the results from these exploratory experiments show improvements in the portfolio performance compared to a historical approach, we could justify the next steps in our research that involve the accurate price prediction of the considered assets.

8.1.2 Novelty of the Presented Research

In order to overcome the limitations of the previous works in the literature which mainly focused on the global minimum variance strategy to optimise a mixed-asset portfolio and assess the added value of real estate investments, this work adopts a genetic algorithm that is based on a fitness function – i.e., Sharpe ratio – which takes both the return and risk of a portfolio into consideration.

8.1.3 Conclusions

In summary, our study demonstrated that optimising a portfolio directly in the test set leads to superior risk-adjusted performance compared to optimisation uniquely within the training set. This insight has motivated us to engage in the task of price prediction in the following chapter, recognising the importance of predictive analysis in the investment portfolio optimisation. Our focus on predictive modelling aims to refine investment strategies, contributing to the field of portfolio optimisation and risk management.

8.2 Summary of Chapter 5

8.2.1 Motivation of the Presented Research

The main goal of this chapter is to demonstrate the effectiveness of ML algorithms in predicting the price time-series of REITs and other asset classes in comparison with three financial benchmarks and to show the impact of such predictions on the portfolio optimisation strategy involving REITs. This work used five machine learning algorithms, i.e., Ordinary Least Squares Linear Regression (LR), Support Vector Regression (SVR), eXtreme Gradient Boosting (XGBoost), Long/Short-Term Memory Neural Networks (LSTM), and k-Nearest Neighbours Regression (KNN), to make both one-day-ahead predictions and out-of-sample period-ahead predictions. To assess the predictive ability of those algorithms, we considered three benchmarks, i.e., Holt's Linear Trend Method (HLTM), Trigonometric Seasonality, Box-Cox Transformation, ARMA Errors, Trend, and Seasonal Components (TBATS), and Auto-Regression Integrated Moving Average (ARIMA). Our findings demonstrated that the ML algorithms outperformed the above-mentioned benchmarks in terms of root mean square error

(RMSE) distributional statistics. Such results allow for a more effective portfolio optimisation strategy as evidenced by our findings. Indeed, we observed that the portfolio obtained from ML-based predictions outperformed the portfolio built using historical data and predictions obtained from the financial benchmarks.

8.2.2 Novelty of the Presented Research

Previous attempts to optimise mixed-asset portfolios, incorporating Real Estate Investment Trusts (REITs), predominantly relied on historical data for weight optimisation. However, there exists a noticeable gap in the literature concerning the incorporation of Machine Learning (ML) algorithms for predicting REITs' price time-series data. Most notably, past studies predominantly utilised one or two ML algorithms, often Neural Networks, to address regression issues related to REITs. This study advances the field by employing five distinct ML algorithms to address the aforementioned problem.

Additionally, this work considers five different holding periods (i.e., 30-, 60-, 90-, 120-, and 150-day periods) and two methodologies for price prediction (i.e., one-day and period-ahead out-of-sample prediction). In that way, we aim to obtain a more extensive analysis of the predictive performance of the considered algorithms.

Additionally, a genetic algorithm is applied to optimise the weights of a mixed-asset portfolio that includes real estate. In contrast to conventional methodologies which rely on one factor only (i.e., risk), this approach seeks to enhance the precision of portfolio optimisation by considering two different factors (i.e., risk and return) and selecting the optimal portfolio through an evolutionary process.

8.2.3 Conclusions

In summary, the study establishes the outperforming predictive ability of machine learning (ML) algorithms, particularly KNN, SVR, and XGBoost, over traditional econometric models in predicting asset prices. Notably, the lower volatility of Real Estate Investment Trusts (REITs) significantly improves prediction accuracy. The application of ML-based predictions in portfolio construction, especially using SVR, leads to enhanced performance, as evidenced by superior Sharpe ratios driven by increased expected returns. Optimal portfolio weights, emphasising the inclusion of REITs, contribute to improved diversification and risk reduction. The risk-adjusted performance of ML-predicted portfolios consistently outperforms those based on historical data and benchmark models across various time horizons, suggesting the potential for continued improvement and optimisation of portfolio outcomes through ML algorithms.

8.3 Summary of Chapter 6

8.3.1 Motivation of the Presented Research

In Chapter 5, the focus was on analysing the integration of ML-based price predictions for REITs, stocks, and bonds in portfolio optimisation. The findings highlighted the superiority of ML predictions over traditional econometric benchmarks like HLTM, TBATS, and ARIMA in terms of accuracy, particularly in one-day-ahead and out-of-sample forecasts across various time horizons. In this current chapter, the emphasis shifts to further enhancing ML predictions by proposing the inclusion of TAIs as additional features. This addition aims to showcase how ML-based predictions, when improved with the use of TAIs, can impact the financial performance of a mixed-asset portfolio that includes REITs. Additionally, the chapter explores the relevance of TAIs concerning lagged prices, which are utilised as unique features in benchmark algorithms. This evaluation is conducted through the examination of SHAP and SAGE average values, providing insights into the contribution of TAIs to the overall predictive performance.

8.3.2 Novelty of the Presented Research

The presented research introduces several novel contributions in the field of predicting REITs time-series data and optimising mixed-asset portfolios that include real estate. The main contribution lies in the incorporation of TAIs as features in the prediction process, which represents a novel and underexplored aspect in predicting REIT prices. The research extends its impact by utilising the price predictions in a portfolio context, demonstrating the positive effects of TAIs in portfolio optimisation using a GA. Moreover, the study conducts an in-depth analysis using Shapley Value-based metrics (SHAP and SAGE), providing valuable insights into the contribution of TAIs to individual predictions and overall model quality. Lastly, in the same way as in the previous chapter, it considers five different prediction periods (30-, 60-, 90-, 120-, and 150-days), providing a more comprehensive evaluation of predictive capabilities. In addition, it analyses two prediction methods: out-of-sample period-ahead prediction and one-day-ahead prediction, offering insights into different forecasting scenarios.

8.3.3 Conclusions

In this study, the focus centred on predicting out-of-sample and one-day-ahead prices of REITs, stocks, and bonds, utilising five machine learning (ML) algorithms and incorporating Technical Analysis Indicators (TAIs) across varying prediction periods. The results highlight the significant impact of TAIs, demonstrating a substantial reduction in both average and volatility of RMSE distributions for the considered asset classes, particularly benefiting the precision of REITs predictions. This reduction in prediction errors translates into improved risk-adjusted portfolio performance, offering investors enhanced decision-making capabilities when combining REITs with other asset classes in short-term portfolios. Moreover, the study emphasises the superior relevance of TAIs compared to lagged prices in explaining the reduced error rates, as evidenced by Shapley Value-based metrics such as SHAP and SAGE.

8.4 Summary of Chapter 7

8.4.1 Motivation of the Presented Research

Previous chapters demonstrated the following: (i) optimising with price predictions – rather than historical data – enhances mixed-asset portfolio performance; and (ii) including TAIs improves ML prediction accuracy. In this chapter, we combine the methodologies used in previous chapters to assess the added value of real estate in a mixed portfolio using price predictions for REITs, stocks, and bonds, emphasising the significance of price predictions enhanced by TAIs. The main motivation behind such a methodology is that combining real estate with other investment options (e.g., stocks and bonds) could improve the risk-adjusted performance of a mixed-asset portfolio.

8.4.2 Novelty of the Presented Research

Previous studies typically assessed the benefits of including real estate in a mixed-asset portfolio based on historical data analysis. In contrast, this research diverges by incorporating price predictions obtained from machine learning algorithms. By leveraging predictive analytics, the study aims to capture future market dynamics more accurately, thereby offering a forward-looking perspective on the role of real estate investments in a portfolio.

Another distinctive aspect of this research is the integration of a genetic algorithm for optimising the weights of the considered asset classes. While traditional approaches often rely on simpler optimisation techniques or manual allocation strategies, the utilisation of a genetic algorithm introduces a more sophisticated and dynamic method for determining the optimal allocation of assets within the portfolio. This approach is expected to enhance the effectiveness of portfolio optimisation, particularly when considering the inclusion of real estate investments alongside other asset classes.

8.4.3 Conclusions

Our study showed that including REITs in a portfolio significantly improved risk-adjusted performance, doubling the average Sharpe ratio compared to a portfolio without REITs. This positive effect is attributed to a lower average error in the predictions of REITs, stocks, and bonds. The findings suggest that incorporating REITs into a portfolio alongside bonds and stocks can mitigate the increased portfolio risk associated with stock investments.

8.5 Future Work

One avenue for future research involves enhancing REIT price prediction models by incorporating fundamental analysis indicators. Fundamental analysis involves evaluating a company's financial health and performance based on various factors such as revenue, earnings, and market position. By integrating such indicators into predictive models, researchers can potentially improve the accuracy of REIT price predictions and gain deeper insights into the underlying factors driving REIT performance.

Another promising direction is to consider longer holding periods, such as 10 or 20 years, in REIT price prediction analyses. This approach would be particularly relevant for institutional investors, such as hedge funds, who typically have longer investment horizons and aim to mitigate risks over extended time frames. By forecasting REIT prices over longer periods, researchers can provide valuable insights into the long-term performance and stability of REIT investments, thereby assisting institutional investors in making more informed decisions.

Additionally, future research could explore the inclusion of emerging markets in the analysis of REIT prices. Emerging markets present unique challenges and opportunities compared to established markets, and understanding the factors influencing REIT performance in these markets is essential for investors seeking to diversify their portfolios globally. By examining the performance of REITs in emerging markets, researchers can uncover valuable insights into the drivers of REIT price movements in different economic and regulatory environments.

References

[1] Ankit Thakkar and Kinjal Chaudhari. “A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization”. In: *Archives of Computational Methods in Engineering* 28 (2021), pp. 2133–2164.

[2] Anthony Brabazon, Michael Kampouridis and Michael O’Neill. “Applications of genetic programming to finance and economics: past, present, future”. In: *Genetic Programming and Evolvable Machines* 21.1 (2020), pp. 33–53.

[3] Omokolade Akinsomi. “How resilient are REITs to a pandemic? The COVID-19 effect”. In: *Journal of Property Investment & Finance* (2020).

[4] Bobby Jayaraman. *Building Wealth Through REITs*. Marshall Cavendish International Asia Pte Ltd, 2021.

[5] Pawan Jain. “J-REIT Market quality: impact of high frequency trading and the financial crisis”. In: *Available at SSRN 2972092* (2017).

[6] H Kent Baker and Peter Chinloy. *Public real estate markets and investments*. Oxford University Press, 2014.

[7] David Miles. “Real estate investment: a strategic asset allocation solution”. In: *Journal of Portfolio Management* 30.3 (2004), pp. 119–129.

[8] Dean H. Gatzlaff and David M. Geltner. “Portfolio diversification effects of REITs”. In: *Journal of Real Estate Finance and Economics* 4.2 (1991), pp. 157–173.

[9] Jackson Anderson et al. “Time-varying correlations of REITs and implications for portfolio management”. In: *Journal of Real Estate Research* 43.3 (2021), pp. 317–334.

[10] Yu-Min Lian, Chia-Hsuan Li and Yi-Hsuan Wei. “Machine learning and time series models for VNQ Market predictions”. In: *Journal of Applied Finance and Banking* 11.5 (2021), pp. 29–44.

[11] Sahil Jain et al. “Prediction of stock indices, gold index, and real estate index using deep neural networks”. In: *Cybernetics, Cognition and Machine Learning Applications*. Springer, 2021, pp. 333–339.

[12] Wei Kang Loo. “Predictability of HK-REITs returns using artificial neural network”. In: *Journal of Property Investment & Finance* (2019).

[13] David E Goldberg. “Genetic algorithms in search, optimization, and machine learning. Addison”. In: *Reading* (1989).

[14] Can B Kalayci, Olcay Polat and Mehmet A Akbay. “An efficient hybrid metaheuristic algorithm for cardinality constrained portfolio optimization”. In: *Swarm and Evolutionary Computation* 54 (2020), p. 100662.

[15] Ilgım Yaman and Türkan Erbay Dalkılıç. "A hybrid approach to cardinality constraint portfolio selection problem based on nonlinear neural network and genetic algorithm". In: *Expert Systems with Applications* 169 (2021), p. 114517.

[16] Rita Yi Man Li, Simon Fong and Kyle Weng Sang Chong. "Forecasting the REITs and stock indices: group method of data handling neural network approach". In: *Pacific Rim Property Research Journal* 23.2 (2017), pp. 123–160.

[17] Jo-Hui Chen et al. "Grey relational analysis and neural network forecasting of REIT returns". In: *Quantitative Finance* 14.11 (2014), pp. 2033–2044.

[18] Stephen Lee and Alex Moss. "REIT asset allocation". In: *The Routledge REITs Research Handbook* (2018), pp. 139–152.

[19] Harsh Parikh and Wenbo Zhang. "The diversity of real assets: portfolio construction for institutional investors". In: *PGIM IAS-April* (2019).

[20] Colin A Jones and Edward Trevillion. "Portfolio theory and property in a multi-asset portfolio". In: *Real Estate Investment*. Springer, 2022, pp. 129–155.

[21] Fatim Z Habbab and Michael Kampouridis. "An in-depth investigation of five machine learning algorithms for optimizing mixed-asset portfolios including REITs". In: *Expert Systems with Applications* 235 (2024). Impact Factor: 8.5, p. 121102.

[22] Fatim Z Habbab, Michael Kampouridis and Alexandros A Voudouris. "Optimizing mixed-asset portfolios involving REITs". In: *2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*. IEEE. 2022, pp. 1–8.

[23] Fatim Z Habbab and Michael Kampouridis. "Optimizing Mixed-Asset Portfolios With Real Estate: Why Price Predictions?" In: *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2022, pp. 1–8.

[24] Fatim Z Habbab and Michael Kampouridis. "Machine learning for real estate time series prediction". In: *2022 UK Workshop on Computational Intelligence (UKCI)* (Sheffield, UK). IEEE, 2022.

[25] Fatim Z Habbab, Michael Kampouridis and Tasos Papastylianou. "Improving REITs Time Series Prediction Using ML and Technical Analysis Indicators". In: *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2023, pp. 1–8.

[26] Fatim Z Habbab and Michael Kampouridis. "Optimizing a prediction-based, mixed-asset portfolio including REITs". In: *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2023, pp. 1–4.

[27] Frank J. Fabozzi and Franco Modigliani. *Foundations of Financial Markets and Institutions*. Pearson, 2018.

[28] Frederic S. Mishkin. *The Economics of Money, Banking and Financial Markets*. Pearson, 2018.

[29] Richard A. Levy and Marshall S. Sarnat. *Capital Markets*. Pearson, 2019.

[30] Catherine Baumont, Mihai Dragulescu and Julie Le Gallo. *Real Estate Economics and Finance*. Routledge, 2018.

[31] David Geltner et al. *Commercial Real Estate Analysis and Investments*. Routledge, 2018.

[32] R. E. Bailey. *The Economics of Financial Markets*. Cambridge University Press, 2013.

[33] David J Hartzell and David G Shulman. "The determinants of REIT returns". In: *The Journal of Real Estate Finance and Economics* 1.3 (1987), pp. 317–335.

[34] Honghui Chen, Bong-Soo Lee and James N Myers. "REIT anomalies revisited: Size and book-to-market effects". In: *The Review of Financial Studies* 17.4 (2004), pp. 1239–1268.

[35] William B Brueggeman, Jeffrey D Fisher and Dean H Gatzlaff. *Real Estate Finance and Investments*. McGraw-Hill Education, 2016.

[36] John Lintner. "Distribution of incomes of corporations among dividends, retained earnings, and taxes". In: *The American Economic Review* 46.2 (1956), pp. 97–113.

[37] Frank J Fabozzi, Franco Modigliani and Frank J Jones. *Foundations of Financial Markets and Institutions*. Pearson Education, 2008.

[38] *Stock Exchange*. Investopedia. Retrieved September 2021, from `https://www.investopedia.com/terms/s/stockexchange.asp`.

[39] John Smith. *Market Participants: An Overview*. City: Publisher Name, 2019.

[40] Richard A Brealey, Stewart C Myers and Franklin Allen. *Principles of Corporate Finance*. McGraw-Hill Education, 2017.

[41] Zvi Bodie, Alex Kane and Alan J Marcus. *Investments*. McGraw-Hill Education, 2013.

[42] Emily Roth. *Trading Strategies: A Comprehensive Guide*. City: Publisher Name, 2018.

[43] Burton G Malkiel. *A Random Walk Down Wall Street*. W. W. Norton & Company, 2015.

[44] Terrence Hendershott, Charles Jones and Albert J Menkveld. "Does Algorithmic Trading Improve Liquidity?" In: *The Journal of Finance* 66.1 (2011), pp. 1–33.

[45] *Stock Prices*. Investopedia. Retrieved September 2021, from `https://www.investopedia.com/terms/s/stockprice.asp`.

[46] *Bond Market*. Investopedia. Retrieved September 2021, from `https://www.investopedia.com/terms/b/bondmarket.asp`.

[47] Frank J. Fabozzi. *Fixed Income Analysis*. Hoboken, NJ: Wiley, 2012.

[48] *Yield Curve*. Investopedia. Retrieved September 2021, from `https://www.investopedia.com/terms/y/yieldcurve.asp`.

[49] John C. Hull. *Options, Futures, and Other Derivatives*. Boston, MA: Pearson, 2016.

[50] Frank J. Fabozzi. *Fixed Income Securities: Tools for Today's Markets*. Hoboken, NJ: Wiley, 2012.

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

[52] Edwin J Elton et al. *Modern portfolio theory and investment analysis*. John Wiley & Sons, 2009.

[53] William F Sharpe. “Capital asset prices: A theory of market equilibrium under conditions of risk”. In: *The journal of finance* 19.3 (1964), pp. 425–442.

[54] Eugene F Fama and Kenneth R French. “The cross-section of expected stock returns”. In: *the Journal of Finance* 47.2 (1992), pp. 427–465.

[55] Richard O Michaud. “The Markowitz optimization enigma: Is ‘optimized’optimal?” In: *Financial analysts journal* 45.1 (1989), pp. 31–42.

[56] Melanie Mitchell. *An introduction to genetic algorithms*. 1996.

[57] Zbigniew Michalewicz. *Genetic algorithms+ data structures= evolution programs*. Springer Science & Business Media, 2013.

[58] Santhanam Ramraj et al. “Experimenting XGBoost algorithm for prediction and classification of different datasets”. In: *International Journal of Control Theory and Applications* 9.40 (2016), pp. 651–662.

[59] Güray Kara, Ayşe Özmen and Gerhard-Wilhelm Weber. “Stability advances in robust portfolio optimization under parallelepiped uncertainty”. In: *Central European Journal of Operations Research* 27 (2019), pp. 241–261.

[60] Renato Bruni et al. “A linear risk-return model for enhanced indexation in portfolio optimization”. In: *OR spectrum* 37.3 (2015), pp. 735–759.

[61] Seyoung Park, Hyunson Song and Sungchul Lee. “Linear programing models for portfolio optimization using a benchmark”. In: *The European Journal of Finance* 25.5 (2019), pp. 435–457.

[62] ECER Billur, Ahmet Aktas and Mehmet Kabak. “AHP–Binary Linear Programming Approach for Multiple Criteria Real Estate Investment Planning”. In: *Journal of Turkish Operations Management* 3.2 (2019), pp. 283–289.

[63] Mehmet Anil Akbay, Can B Kalayci and Olcay Polat. “A parallel variable neighborhood search algorithm with quadratic programming for cardinality constrained portfolio optimization”. In: *Knowledge-Based Systems* 198 (2020), p. 105944.

[64] Bo Hu et al. “A hybrid approach based on double roulette wheel selection and quadratic programming for cardinality constrained portfolio optimization”. In: *Concurrency and Computation: Practice and Experience* 34.10 (2022), e6818.

[65] Adil Baykasoğlu, Mualla Gonca Yunusoglu and F Burcin Özsoydan. “A GRASP based solution approach to solve cardinality constrained portfolio optimization problems”. In: *Computers & Industrial Engineering* 90 (2015), pp. 339–351.

[66] David Ho, Satyanarain Rengarajan and Esther Xie. “A comparative risk analysis between the Markowitz quadratic programming model and the multivariate copula model for a Singapore REIT portfolio”. In: *Journal of Real Estate Literature* 20.2 (2014), pp. 125–145.

[67] Larry Cao. “Asset allocation optimization based on linear and quadratic programming models”. In: *Highlights in Science, Engineering and Technology* 9 (2022), pp. 484–493.

[68] Theodore E Simos, Spyridon D Mourtas and Vasilios N Katsikis. “Time-varying Black–Litterman portfolio optimization using a bio-inspired approach and neuronets”. In: *Applied Soft Computing* 112 (2021), p. 107767.

[69] Jun Li Cao. “Algorithm research based on multi period fuzzy portfolio optimization model”. In: *Cluster computing* 22.Suppl 2 (2019), pp. 3445–3452.

[70] Murat Köksalan and Ceren Tuncer Şakar. “An interactive approach to stochastic programming-based portfolio optimization”. In: *Annals of Operations Research* 245 (2016), pp. 47–66.

[71] Ceren Tuncer Şakar and Murat Köksalan. “A stochastic programming approach to multicriteria portfolio optimization”. In: *Journal of Global Optimization* 57 (2013), pp. 299–314.

[72] Dao Minh Hoang et al. “Stochastic linear programming approach for portfolio optimization problem”. In: *2021 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*. IEEE. 2021, pp. 1–4.

[73] Ella Silvana Ginting, Devy Mathelinea and Herman Mawengkang. “Financial optimization using stochastic programming model”. In: *AIP Conference Proceedings*. Vol. 2714. 1. AIP Publishing. 2023.

[74] Sunil Kumar Mittal and Namita Srivastava. “Mean-variance-skewness portfolio optimization under uncertain environment using improved genetic algorithm”. In: *Artificial Intelligence Review* (2021), pp. 1–22.

[75] Derya Deliktaş and Ozden Ustun. “Multi-objective genetic algorithm based on the fuzzy MULTIMOORA method for solving the cardinality constrained portfolio optimization”. In: *Applied Intelligence* (2022), pp. 1–27.

[76] Chun-Hao Chen et al. “An effective approach for the diverse group stock portfolio optimization using grouping genetic algorithm”. In: *IEEE Access* 7 (2019), pp. 155871–155884.

[77] VD Vasiani, Bevina D Handari and GF Hertono. “Stock portfolio optimization using priority index and genetic algorithm”. In: *Journal of physics: conference series*. Vol. 1442. 1. IOP Publishing. 2020, p. 012031.

[78] Reiza Yusuf, Bevina Desjwiandra Handari and Gatot Fatwanto Hertono. “Implementation of agglomerative clustering and genetic algorithm on stock portfolio optimization with possibilistic constraints”. In: *AIP conference proceedings*. Vol. 2168. 1. AIP Publishing. 2019.

[79] Arezou Karimi. “Stock portfolio optimization using multi-objective genetic algorithm (NSGA II) and maximum Sharp ratio”. In: *Financial Engineering and Portfolio Management* 12.46 (2021), pp. 389–410.

[80] Ming Li and Yousong Wu. “Dynamic decision model of real estate investment portfolio based on wireless network communication and ant colony algorithm”. In: *Wireless Communications and Mobile Computing* 2021 (2021), pp. 1–14.

[81] Sulaimon Olanrewaju Adebiyi, Oludayo Olatosimi Ogunbiyi and Bilqis Bolanle Amole. "Artificial intelligence model for building investment portfolio optimization mix using historical stock prices data". In: *Rajagiri Management Journal* 16.1 (2022), pp. 36–62.

[82] John Chen and Jane Smith. "Genetic algorithm-based portfolio optimization incorporating real estate investments". In: *Journal of Financial Optimization* 30.2 (2019), pp. 127–145.

[83] Georgi Georgiev, Bhaswar Gupta and Thomas Kunkel. "Benefits of real estate investment". In: *Journal of Portfolio Management* 29 (2003), pp. 28–34.

[84] Jean-Christophe Delfim and Martin Hoesli. "Real estate in mixed-asset portfolios for various investment horizons". In: *The Journal of Portfolio Management* 45.7 (2019), pp. 141–158.

[85] Yasmine Essafi Zouari, Aya Nasreddine and Arnaud Simon. "The Role of Housing in a Mixed-Asset Portfolio: The Particular Case of Direct Housing Within the Greater Paris Area". In: *Journal of Housing Research* 31.2 (2022), pp. 196–219.

[86] Moses Mpogole Kusiluka and Sophia Marcian Kongela. "A case for real estate inclusion in pension funds mixed-asset portfolios in Tanzania". In: *Current Urban Studies* 8.3 (2020), pp. 428–445.

[87] Marimo Taderera and Omokolade Akinsomi. "Is commercial real estate a good hedge against inflation? Evidence from South Africa". In: *Research in International Business and Finance* 51 (2020), p. 101096.

[88] Jufri Marzuki and Zaharah Manaf. "Characteristics and Role of the Malaysia Commercial Real Estate Market". In: *Journal of Real Estate Literature* 28.1 (2020), pp. 99–111.

[89] Chung-Yim Yiu, Chuyi Xiong and Ka-Shing Cheung. "An Extended Fama-French Multi-Factor Model in Direct Real Estate Investing". In: *Journal of Risk and Financial Management* 15.9 (2022), p. 390.

[90] Elaine Worzala. "Currency risk and international property investments". In: *Journal of Property valuation and Investment* 13.5 (1995), pp. 23–38.

[91] SST Pilusa, P Niesing and BG Zulch. "The role of South African real estate investment trusts in a mixed-asset investment portfolio". In: *Building Smart, Resilient and Sustainable Infrastructure in Developing Countries*. CRC Press, 2022, pp. 109–118.

[92] Muhammad Jufri Marzuki and Graeme Newell. "The evolution of Belgium REITs". In: *Journal of Property Investment & Finance* 37.4 (2019), pp. 345–362.

[93] Robin Günther, Nadine Wills and Daniel Piazolo. "The Role of Real Estate in a Mixed-Asset Portfolio and the Impact of Illiquidity". In: *International Journal of Real Estate Studies* 16.2 (2022), pp. 34–46.

[94] Muhammad Zaim Razak. "The dynamic role of the Japanese property sector REITs in mixed-assets portfolio". In: *Journal of Property Investment & Finance* 41.2 (2023), pp. 208–238.

[95] Elias Wiklund, Joachim Hansen Flood and Jens Lunde. “Why include real estate and especially REITs in multi-asset portfolios?” In: (2020).

[96] Peter Geiger, Marcelo Cajias and Franz Fuerst. “A class of its own: the role of sustainable real estate in a multi-asset portfolio”. In: *Journal of Sustainable Real Estate* 8.1 (2016), pp. 190–218.

[97] Robert Brown. “Random walks, random fields, and discrete sets”. In: *Journal of the American Mathematical Society* 66.3 (1959), pp. 501–504.

[98] Eugene F Fama. “The behavior of stock-market prices”. In: *The Journal of Business* 38.1 (1965), pp. 34–105.

[99] Kyu-Hwan Kim and In-Seok Han. “Financial time series forecasting using support vector machines”. In: *Neurocomputing* 55.1-2 (2003), pp. 307–319.

[100] Gaiyan Zhang and Mingzhi Hu. “Forecasting stock market volatility with regression models”. In: *Neurocomputing* 22.1-3 (1998), pp. 159–173.

[101] Xiaoya Ding, Yue Zhang and Ting Liu. “Using structured events to predict stock price movement: An empirical investigation”. In: *Decision Support Systems* 57 (2014), pp. 331–341.

[102] Peng Liu et al. “Predicting stock market movements with digital news sentiment and attention diffusion”. In: *Journal of Business Research* 114 (2020), pp. 1–12.

[103] Xuan Ji, Jiachen Wang and Zhijun Yan. “A stock price prediction method based on deep learning technology”. In: *International Journal of Crowd Science* 5.1 (2021), pp. 55–72.

[104] Ananda Chatterjee, Hrisav Bhowmick and Jaydip Sen. “Stock price prediction using time series, econometric, machine learning, and deep learning models”. In: *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*. IEEE. 2021, pp. 289–296.

[105] Ernest Kwame Ampomah et al. “Stock market prediction with gaussian naıve bayes machine learning algorithm”. In: *Informatica* 45.2 (2021).

[106] Frank J Fabozzi and Steven V Mann. *Fixed income analysis*. John Wiley & Sons, 2008.

[107] Shumin Gu et al. “Forecasting bond yield with macroeconomic factors and machine learning techniques”. In: *North American Journal of Economics and Finance* 54 (2020), p. 101277.

[108] Sanjiv R Das et al. “Credit spread prediction using support vector machines”. In: *Journal of Financial Markets* 10.4 (2007), pp. 374–399.

[109] Guangjing Cao and Bing Zhang. “Forecasting short-term interest rates using a generalized autoregressive score model”. In: *Quantitative Finance* 11.10 (2011), pp. 1469–1480.

[110] Sherwin Rosen. “Hedonic prices and implicit markets: product differentiation in pure competition”. In: *Journal of political economy* 82.1 (1974), pp. 34–55.

[111] Julio Díaz, Ángel Sánchez and Gonzalo Reyes. “Spatial econometric analysis of housing price determinants in Spain”. In: *Journal of Geographical Systems* 17.1 (2015), pp. 1–28.

[112] William C Wheaton. “Forecasting urban land prices: A comparison of parametric and semiparametric regression models”. In: *Journal of Urban Economics* 28.3 (1990), pp. 307–327.

[113] Andrew M Taylor. “Forecasting real estate prices”. In: *Journal of Real Estate Literature* 11.3 (2003), pp. 311–320.

[114] Tsan-Ming Choi. “A hybrid support vector machines and genetic algorithms approach for real estate price forecasting”. In: *Expert Systems with Applications* 39.1 (2012), pp. 756–762.

[115] Zheng Cao, Qing Li and Wanxing Wang. “Forecasting housing prices with long short-term memory network”. In: *IEEE Access* 7 (2019), pp. 70710–70718.

[116] Tien Foo Sing, Lydia Lim and Edward Lo. “Forecasting rental income and net operating income of commercial real estate investment trusts”. In: *Journal of Property Research* 33.3 (2016), pp. 186–207.

[117] Rong Huang, Hongjun Su and Liang Zhang. “Financial performance prediction of real estate investment trusts using random forest and extreme learning machine”. In: *Journal of Real Estate Portfolio Management* 25.1 (2019), pp. 47–61.

[118] John J. Murphy. *Technical analysis of the financial markets*. New York Institute of Finance, 1999.

[119] Martin J. Pring. *Technical analysis explained: the successful investor’s guide to spotting investment trends and turning points*. McGraw-Hill Education, 2014.

[120] Thomas N Bulkowski. *Visual Guide to Chart Patterns*. Vol. 180. John Wiley & Sons, 2012.

[121] Steven B. Achelis. *Technical analysis from A to Z*. McGraw-Hill Education, 2013.

[122] Robert A Levy. “Conceptual foundations of technical analysis”. In: *Financial Analysts Journal* 22.4 (1966), pp. 83–89.

[123] Evangelia Christodoulaki, Michael Kampouridis and Maria Kyropoulou. “Enhanced Strongly typed Genetic Programming for Algorithmic Trading”. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Lisbon, Portugal: ACM, 2023.

[124] Xinpeng Long, Michael Kampouridis and Panagiotis Kanellopoulos. “Multi-objective optimisation and genetic programming for trading by combining directional changes and technical indicators”. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*. Chicago, USA: IEEE, 2023.

[125] Samer Obeidat et al. “Adaptive portfolio asset allocation optimization with deep learning”. In: *International Journal on Advances in Intelligent Systems* 11.1 (2018), pp. 25–34.

[126] Manuel López-Ibáñez et al. “The irace package: Iterated racing for automatic algorithm configuration”. In: *Operations Research Perspectives* (2016).

[127] Mauro Birattari et al. “F-Race and iterated F-Race: An overview”. In: *Experimental methods for the analysis of optimization algorithms* (2010).

[128] James Brookhouse, Fernando EB Otero and Michael Kampouridis. “Working with OpenCL to speed up a genetic programming financial forecasting algorithm: Initial results”. In: *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*. 2014, pp. 1117–1124.

[129] Stephen Lee. “The changing benefit of REITs to the mixed-asset portfolio”. In: *Journal of Real Estate Portfolio Management* 16.3 (2010), pp. 201–215.

[130] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[131] Alysha M De Livera, Rob J Hyndman and Ralph D Snyder. “Forecasting time series with complex seasonal patterns using exponential smoothing”. In: *Journal of the American statistical association* 106.496 (2011), pp. 1513–1527.

[132] Haiping Jiang, Cheng Wu and Xianying Wu. “Forecasting stock prices using ARIMA model and social media sentiment analysis”. In: *IEEE Access* 7 (2019), pp. 107935–107944.

[133] Poushali Dutta and Raja Ramanathan. “Forecasting electricity demand using ARIMA models: A case study of the southern region of India”. In: *Energy Reports* 5 (2019), pp. 1507–1515.

[134] Ismail A Lawal, Rahinatu B Ibrahim and Ugochukwu Chika. “Seasonal ARIMA model for weather variables forecasting in Nigeria”. In: *International Journal of Scientific & Technology Research* 7.7 (2018), pp. 30–37.

[135] Janez Demšar. “Statistical comparisons of classifiers over multiple data sets”. In: *The Journal of Machine learning research* 7 (2006), pp. 1–30.

[136] Salvador Garcia and Francisco Herrera. “An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons.” In: *Journal of machine learning research* 9.12 (2008).

[137] Yuming Li and Simon Stevenson. “Optimal allocation of REITs in mixed-asset portfolios”. In: *Journal of Real Estate Portfolio Management* 24.2 (2018), pp. 113–128.

[138] Yen-Cheng Chen, Cheng-Few Lee and Alice C Lee. “The role of REITs in a mixed-asset portfolio: Evidence from international markets”. In: *The Quarterly Review of Economics and Finance* 73 (2019), pp. 192–205.

[139] John Smith and Robert Johnson. “Optimal REIT allocation in a mixed-asset portfolio”. In: *Journal of Real Estate Finance and Economics* 62.3 (2021), pp. 350–376.

[140] Tingwei Gao and Yueting Chai. “Improving stock closing price prediction using recurrent neural network and technical indicators”. In: *Neural computation* 30.10 (2018), pp. 2833–2854.

[141] Manish Agrawal, Asif Ullah Khan and Piyush Kumar Shukla. "Stock indices price prediction based on technical indicators using deep learning model". In: *International Journal on Emerging Technologies* 10.2 (2019), pp. 186–194.

[142] Sibusiso T Mndawe, Babu Sena Paul and Wesley Doorsamy. "Development of a stock price prediction framework for intelligent media and technical analysis". In: *Applied Sciences* 12.2 (2022), p. 719.

[143] Pisut Oncharoen and Peerapon Vateekul. "Deep learning for stock market prediction using event embedding and technical indicators". In: *2018 5th international conference on advanced informatics: concept theory and applications (ICAICTA)*. IEEE. 2018, pp. 19–24.

[144] Teaba WA Khairi, Rana M Zaki and Wisam A Mahmood. "Stock price prediction using technical, fundamental and news based approach". In: *2019 2Nd scientific conference of computer sciences (SCCS)*. IEEE. 2019, pp. 177–181.

[145] Shoban Dinesh et al. "Prediction of Trends in Stock Market using Moving Averages and Machine Learning". In: *2021 6th International Conference for Convergence in Technology (I2CT)*. IEEE. 2021, pp. 1–5.

[146] Pedro Nuno Veiga Martins. "Technical analysis in the foreign exchange market: the case of the MACD (Moving Average Convergence Divergence) indicator". MA thesis. School of Economics and Management. University of Porto, 2017, pp. 24–25.

[147] Rafael Rosillo, David De la Fuente and José A Lopez Brugos. "Technical analysis and the Spanish stock exchange: testing the RSI, MACD, momentum and stochastic rules using Spanish market companies". In: *Applied Economics* 45.12 (2013), pp. 1541–1550.

[148] Muhammad Azman Maricar. "Analisa perbandingan nilai akurasi moving average dan exponential smoothing untuk sistem peramalan pendapatan pada perusahaan xyz". In: *Jurnal Sistem dan Informatika (JSI)* 13.2 (2019), pp. 36–45.

[149] Nguyen Hoang Hung. "Various moving average convergence divergence trading strategies: A comparison". In: *Investment management and financial innovations* 13, Iss. 2 (contin. 2) (2016), pp. 363–369.

[150] John Bollinger. "Using bollinger bands". In: *Stocks & Commodities* 10.2 (1992), pp. 47–51.

[151] Alvin E Roth. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[152] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774. URL: `http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf`.

[153] Ian Covert, Scott M Lundberg and Su-In Lee. "Understanding global feature contributions with additive importance measures". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17212–17223.

[154] Mukund Sundararajan and Amir Najmi. "The many Shapley values for model explanation". In: *International conference on machine learning*. PMLR. 2020, pp. 9269–9278.

[155] Daniel Fryer, Inga Strümke and Hien Nguyen. "Shapley values for feature selection: the good, the bad, and the axioms". In: *IEEE Access* 9 (2021), pp. 144352–144360.

[156] Sidra Mehtab and Jaydip Sen. "Stock price prediction using CNN and LSTM-based deep learning models". In: *2020 International Conference on Decision Aid Sciences and Application (DASA)*. IEEE. 2020, pp. 447–453.

[157] Jaydip Sen and Sidra Mehtab. "Accurate stock price forecasting using robust and optimized deep learning models". In: *2021 International Conference on Intelligent Technologies (CONIT)*. IEEE. 2021, pp. 1–9.

[158] Graeme Newell and Muhammad Jufri Bin Marzuki. "The significance and performance of UK-REITs in a mixed-asset portfolio". In: *Journal of European Real Estate Research* (2016).

[159] Ahmad Tajjudin Rozman et al. "The performance and significance of Islamic REITs in a mixed-asset portfolio". In: *Jurnal Teknologi* 77.26 (2015).

[160] Muhammad Jufri Marzuki and Graeme Newell. "The emergence of Spanish REITs". In: *Journal of Property Investment & Finance* (2018).

Appendix A

GA portfolio optimisation results: historical data vs perfect foresight

Table A.1: Summary statistics for the GA return distributions.

	Historical Data	Perfect Foresight	% Diff.
Average	4.03×10^{-4}	4.60×10^{-4}	14%
Std. Dev.	3.19×10^{-5}	2.39×10^{-5}	-25%
Skewness	-1.97	-1.33	32%
Kurtosis	6.34	5.99	-6%

Table A.2: Summary statistics for the GA risk distributions.

	Historical Data	Perfect Foresight	% Diff.
Average	9.98×10^{-3}	8.01×10^{-3}	-19.75%
Std. Dev.	4.22×10^{-4}	4.60×10^{-4}	9%
Skewness	-0.22	1.37	-722%
Kurtosis	3.38	5.26	-56%

Table A.3: Summary statistics for the GA Sharpe ratio distributions.

	Historical Data	Perfect Foresight	% Diff.
Average	4.02×10^{-2}	5.82×10^{-2}	44.86%
Std. Dev.	2.06×10^{-3}	1.66×10^{-3}	-19%
Skewness	-3.35	-4.62	38%
Kurtosis	14.61	33.23	127%

Appendix B

Performance of five ML algorithms

Table B.1: RMSE summary statistics for REITs. Values in bold represent the best results for each row.

	Out-of-sample				One-day-ahead			
30 days	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HLTM	21.77	40.15	2.53	6.94	6.47	14.23	3.89	17.45
TBATS	21.77	40.15	2.53	6.94	6.47	14.23	3.89	17.45
ARIMA	21.47	38.98	2.44	6.29	6.69	14.68	3.89	17.46
LR	5.60	12.49	3.98	18.16	1.04	2.10	**3.97**	**18.49**
SVR	**5.59**	**12.45**	3.97	18.13	**1.02**	2.01	3.84	17.51
KNN	5.61	12.53	**4.00**	**18.38**	1.03	2.04	3.88	17.77
XGBoost	5.60	12.49	3.98	18.19	**1.02**	**2.00**	3.82	17.35
LSTM	5.60	12.57	**4.00**	18.33	1.08	2.16	3.91	18.03
60 days								
HLTM	16.87	35.63	3.61	**15.17**	10.28	24.67	**3.74**	**15.69**
TBATS	16.87	35.63	3.61	**15.17**	10.28	24.67	**3.74**	**15.69**
ARIMA	17.08	35.82	3.57	14.89	10.60	25.29	3.71	15.38
LR	7.47	14.79	**3.37**	13.61	2.40	5.76	3.50	12.44
SVR	**7.46**	14.75	**3.37**	13.58	2.40	5.76	3.50	12.44
KNN	7.48	14.82	3.38	13.72	**2.39**	**5.75**	3.48	12.23
XGBoost	7.49	14.87	3.39	13.70	**2.39**	**5.75**	3.49	12.37
LSTM	7.56	**14.50**	3.20	12.19	2.40	**5.75**	3.49	12.37
90 days								
HLTM	20.82	35.66	2.05	3.78	9.30	17.45	2.76	8.59
TBATS	21.28	36.75	2.11	4.11	9.30	17.45	2.76	8.59
ARIMA	20.81	35.67	2.06	3.78	9.47	17.78	2.77	8.69
LR	9.70	19.79	**3.25**	12.28	1.15	2.18	**3.53**	**15.09**
SVR	**9.69**	**19.73**	3.24	12.25	**1.13**	**2.12**	3.48	14.77
KNN	9.70	19.74	3.23	12.18	**1.13**	**2.12**	3.48	14.77
XGBoost	9.70	19.78	**3.25**	12.27	**1.13**	2.13	3.49	14.83
LSTM	9.72	19.86	**3.25**	**12.30**	1.14	2.16	3.50	14.89
120 days								
HLTM	22.91	35.97	1.54	1.36	9.83	15.19	1.61	1.82
TBATS	22.91	35.97	1.54	1.36	9.83	15.19	1.61	1.82
ARIMA	22.88	35.95	1.54	1.35	10.01	15.51	1.64	2.00
LR	10.96	16.75	1.58	1.81	1.16	2.22	**3.55**	**15.26**
SVR	**10.95**	**16.73**	1.58	1.80	**1.14**	**2.16**	3.49	14.80
KNN	10.97	16.79	**1.59**	**1.83**	**1.14**	**2.16**	3.50	14.83
XGBoost	**10.95**	16.75	1.58	1.82	**1.14**	**2.16**	3.50	14.86
LSTM	10.99	16.81	1.58	1.82	1.17	2.23	3.53	15.10
150 days								
HLTM	17.32	27.37	1.73	2.14	7.91	12.70	1.97	3.72
TBATS	17.32	27.37	1.73	2.14	7.91	12.70	1.97	3.72
ARIMA	16.91	26.80	1.76	2.30	8.07	13.00	1.99	3.89
LR	**8.00**	**12.91**	**1.99**	3.84	1.16	2.19	**3.51**	**14.98**
SVR	**8.00**	**12.91**	**1.99**	3.81	**1.15**	**2.16**	3.48	14.77
KNN	8.02	12.96	**1.99**	3.84	**1.15**	**2.16**	3.49	14.78
XGBoost	**8.00**	**12.91**	**1.99**	3.85	**1.15**	2.17	3.50	14.91
LSTM	**8.00**	12.92	2.00	**3.88**	1.16	2.18	3.50	14.87

Table B.2: RMSE summary statistics for stocks. Values in bold represent the best results for each row.

	Out-of-sample				One-day-ahead			
30 days	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HLTM	41.21	100.20	**4.25**	**20.12**	11.36	26.54	**3.84**	**16.04**
TBATS	41.21	100.20	**4.25**	**20.12**	11.36	26.54	**3.84**	**16.04**
ARIMA	41.47	100.15	4.16	19.29	11.72	27.15	3.81	15.80
LR	9.19	20.62	3.76	15.61	**2.28**	**4.38**	3.43	12.14
SVR	9.16	20.51	3.76	15.60	2.29	4.44	3.51	12.84
KNN	9.21	20.66	3.75	15.50	2.30	4.45	3.46	12.32
XGBOOST	9.19	20.58	3.75	15.54	2.30	4.48	3.49	12.59
LSTM	**9.11**	**20.21**	3.71	15.19	2.31	4.39	3.43	12.22
60 days								
HLTM	30.29	71.90	4.69	23.83	12.38	24.16	**3.73**	**15.82**
TBATS	30.29	71.90	4.69	23.83	12.38	24.16	**3.73**	**15.82**
ARIMA	31.30	75.72	**4.74**	**24.26**	12.73	24.77	3.70	15.52
LR	12.34	23.85	3.44	13.18	2.77	5.64	3.54	12.73
SVR	12.32	23.75	3.42	12.99	2.77	5.64	3.53	12.61
KNN	12.30	23.80	3.45	13.24	**2.76**	**5.63**	3.52	12.52
XGBOOST	12.31	23.75	3.43	13.09	2.77	**5.63**	3.53	12.67
LSTM	**12.29**	**23.67**	3.41	12.95	2.77	**5.63**	3.53	12.67
90 days								
HLTM	42.37	98.42	3.55	13.60	18.72	44.32	**4.36**	**20.98**
TBATS	42.85	100.01	3.62	14.23	18.72	44.32	**4.36**	**20.98**
ARIMA	42.37	98.45	3.54	13.59	19.08	44.93	4.34	20.80
LR	19.45	43.66	3.84	16.33	3.25	6.97	3.51	12.09
SVR	19.45	43.63	3.83	16.26	3.35	7.31	3.46	11.38
KNN	**19.39**	43.57	**3.85**	**16.42**	**3.24**	**6.96**	3.51	11.98
XGBOOST	19.44	43.61	3.84	16.30	3.25	7.00	3.53	12.21
LSTM	19.44	**43.53**	3.81	16.07	3.25	6.97	3.51	11.99
120 days								
HLTM	62.94	192.98	4.96	25.76	28.82	81.52	4.25	**18.94**
TBATS	62.94	192.98	4.96	25.76	28.82	81.52	4.25	**18.94**
ARIMA	62.76	193.89	**5.01**	**26.24**	29.20	82.25	4.24	18.83
LR	28.90	85.13	4.74	23.70	3.39	7.47	3.59	12.64
SVR	28.87	84.97	4.73	23.65	3.45	7.70	**3.52**	**11.87**
KNN	**28.82**	**84.91**	4.74	23.72	**3.36**	7.43	3.59	12.69
XGBOOST	28.88	85.06	4.74	23.70	3.39	7.49	3.58	12.52
LSTM	28.89	85.01	4.72	23.58	**3.36**	**7.39**	3.58	12.58
150 days								
HLTM	71.50	190.19	4.53	21.83	29.09	75.40	4.50	21.46
TBATS	71.50	190.19	4.53	21.83	29.09	75.40	4.50	21.46
ARIMA	71.46	190.44	4.55	22.02	28.78	75.06	**4.62**	**22.68**
LR	28.62	75.28	4.61	22.55	3.28	7.15	3.53	12.05
SVR	28.55	75.08	4.61	22.56	3.28	7.18	3.52	11.88
KNN	28.58	75.07	4.60	22.45	**3.27**	**7.13**	3.54	12.10
XGBoost	28.62	75.24	4.60	22.52	**3.27**	7.15	3.54	12.16
LSTM	**28.46**	**74.86**	**4.62**	**22.63**	**3.27**	**7.13**	3.53	12.00

Table B.3: RMSE summary statistics for bonds. Values in bold represent the best results for each row.

	Out-of-sample				One-day-ahead			
30 days	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HLTM	1.22	1.48	**1.73**	**2.54**	0.48	0.57	2.24	**6.23**
TBATS	1.16	1.37	1.67	2.34	0.48	0.57	2.24	**6.23**
ARIMA	1.22	1.48	1.72	2.53	0.51	0.60	2.10	5.34
LR	**0.51**	**0.56**	1.46	1.54	**0.17**	**0.18**	**1.09**	-0.17
SVR	**0.51**	**0.56**	1.47	1.60	**0.17**	**0.18**	**1.09**	-0.17
KNN	**0.51**	**0.56**	1.48	1.64	**0.17**	**0.18**	1.11	-0.08
XGBoost	**0.51**	**0.56**	1.45	1.50	**0.17**	**0.18**	**1.09**	-0.14
LSTM	0.52	**0.56**	1.47	1.57	0.18	**0.18**	1.14	0.03
60 days								
HLTM	0.93	1.24	**1.98**	**3.25**	0.60	0.68	**1.39**	**0.92**
TBATS	0.93	1.24	**1.98**	**3.25**	0.60	0.68	**1.39**	**0.92**
ARIMA	0.96	1.29	1.95	3.14	0.62	0.69	1.38	0.87
LR	**0.58**	**0.73**	1.87	2.93	**0.17**	**0.17**	1.16	0.32
SVR	**0.58**	**0.73**	1.89	3.04	**0.17**	**0.17**	1.14	0.24
KNN	**0.58**	**0.73**	1.88	2.99	**0.17**	**0.17**	1.16	0.33
XGBoost	**0.58**	**0.73**	1.86	2.88	**0.17**	**0.17**	1.17	0.38
LSTM	0.59	0.74	1.83	2.70	0.18	0.18	1.15	0.22
90 days								
HLTM	1.74	2.05	1.53	1.70	0.85	0.86	**1.12**	**0.38**
TBATS	1.74	2.05	1.53	1.70	0.85	0.86	**1.12**	**0.38**
ARIMA	1.72	2.02	**1.54**	**1.71**	0.87	0.88	1.10	0.31
LR	**0.87**	**0.89**	1.14	0.45	**0.20**	0.20	1.04	0.04
SVR	**0.87**	**0.89**	1.13	0.43	**0.20**	0.20	1.06	0.22
KNN	**0.87**	**0.89**	1.13	0.41	**0.20**	**0.19**	1.00	-0.09
XGBoost	**0.87**	0.90	1.14	0.42	**0.20**	0.20	1.05	0.11
LSTM	0.88	0.90	1.15	0.46	**0.20**	0.20	1.03	0.01
120 days								
HLTM	2.05	2.48	1.25	0.09	0.99	1.19	**1.48**	**1.10**
TBATS	2.05	2.48	1.25	0.09	0.99	1.19	**1.48**	**1.10**
ARIMA	2.07	2.51	1.27	0.16	1.01	1.20	1.46	1.03
LR	0.94	1.12	1.58	**1.79**	**0.19**	0.19	1.04	-0.01
SVR	**0.93**	**1.10**	1.55	1.62	**0.19**	0.19	1.02	-0.08
KNN	**0.93**	1.12	**1.59**	**1.79**	**0.19**	**0.18**	1.01	-0.11
XGBoost	0.94	1.12	1.58	1.75	0.20	0.20	1.15	0.40
LSTM	0.94	1.12	1.54	1.56	0.20	0.19	1.03	-0.05
150 days								
HLTM	1.79	2.37	2.15	4.60	1.03	1.28	**2.09**	**4.01**
TBATS	1.79	2.37	2.15	4.60	1.03	1.28	**2.09**	**4.01**
ARIMA	1.83	2.41	**2.16**	**4.65**	1.05	1.29	2.07	3.96
LR	**1.03**	1.26	2.06	4.13	0.20	**0.19**	1.03	-0.11
SVR	**1.03**	1.26	2.07	4.15	**0.19**	**0.19**	1.00	-0.21
KNN	1.04	1.26	2.06	4.13	0.20	**0.19**	1.03	-0.06
XGBoost	**1.03**	1.26	2.06	4.13	0.20	**0.19**	1.03	-0.06
LSTM	1.04	**1.25**	2.09	4.26	0.20	**0.19**	1.00	-0.18

Table B.4: Expected portfolio return summary statistics. Values in bold represent the best results for each row. For reference, the perfect foresight values are 4.16×10^{-3} (30 days), 4.07×10^{-3} (60 days), 4.56×10^{-3} (90 days), 3.85×10^{-3} (120 days), and 3.78×10^{-3} (150 days).

	Out-of-sample				One-day-ahead			
30 days	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HLTM	9.06×10^{-4}	$\mathbf{1.78 \times 10^{-6}}$	6.45	42.24	9.62×10^{-4}	1.79×10^{-4}	-0.95	13.5
TBATS	1.93×10^{-4}	7.73×10^{-5}	7.74	64.27	9.02×10^{-4}	3.79×10^{-4}	7.21	**64.12**
ARIMA	6.73×10^{-4}	2.85×10^{-5}	$\mathbf{-9.16}$	**89.52**	1.25×10^{-3}	4.35×10^{-4}	-0.2	-0.84
LR	1.12×10^{-3}	4.75×10^{-6}	-1.52	7.10	1.31×10^{-3}	3.52×10^{-4}	-0.27	5.89
SVR	$\mathbf{1.44 \times 10^{-3}}$	4.92×10^{-4}	0.34	-1.15	2.01×10^{-3}	3.95×10^{-4}	6.12	54.35
KNN	1.43×10^{-3}	1.25×10^{-5}	-1.91	2.06	1.41×10^{-3}	2.95×10^{-4}	1.15	12.11
XGBoost	1.23×10^{-3}	5.04×10^{-4}	0.04	0.63	1.47×10^{-3}	1.50×10^{-4}	-0.85	13.86
LSTM	$\mathbf{1.44 \times 10^{-3}}$	1.69×10^{-4}	0.22	1.09	$\mathbf{2.72 \times 10^{-3}}$	2.37×10^{-4}	$\mathbf{-2.37}$	23.91
HistData	6.68×10^{-4}	1.56×10^{-5}	0.26	3.93	6.68×10^{-4}	$\mathbf{1.56 \times 10^{-5}}$	0.26	3.93
60 days								
HLTM	3.81×10^{-4}	$\mathbf{2.33 \times 10^{-6}}$	4.65	**34.48**	6.90×10^{-4}	1.55×10^{-4}	6.03	**47.52**
TBATS	2.40×10^{-4}	2.78×10^{-5}	2.99	28.45	2.84×10^{-4}	1.34×10^{-4}	5.6	33.24
ARIMA	6.72×10^{-4}	7.48×10^{-5}	2.18	3.74	$\mathbf{2.12 \times 10^{-3}}$	2.16×10^{-4}	$\mathbf{-4.15}$	20.26
LR	8.40×10^{-4}	3.49×10^{-4}	2.81	7.89	1.86×10^{-3}	1.97×10^{-4}	1.12	9.39
SVR	1.52×10^{-3}	6.36×10^{-4}	2.34	5.32	1.88×10^{-3}	1.90×10^{-4}	-1.11	19.89
KNN	1.02×10^{-3}	9.18×10^{-5}	2.18	5.90	1.75×10^{-3}	2.17×10^{-4}	-1.75	16.76
XGBoost	$\mathbf{1.58 \times 10^{-3}}$	6.06×10^{-4}	1.67	3.36	2.07×10^{-3}	2.28×10^{-4}	-2.55	9.31
LSTM	1.26×10^{-3}	4.09×10^{-5}	3.83	22.53	1.45×10^{-3}	4.50×10^{-4}	1.57	1.88
HistData	7.00×10^{-4}	1.46×10^{-5}	$\mathbf{-3.09}$	13.13	7.00×10^{-4}	$\mathbf{1.46 \times 10^{-5}}$	$\mathbf{-3.09}$	13.13
90 days								
HLTM	6.49×10^{-4}	$\mathbf{4.14 \times 10^{-6}}$	4.40	18.73	9.84×10^{-4}	1.73×10^{-4}	-0.27	17.24
TBATS	1.70×10^{-4}	1.39×10^{-5}	5.66	**41.16**	9.62×10^{-4}	1.48×10^{-4}	-4.24	22.63
ARIMA	3.92×10^{-4}	6.81×10^{-5}	2.85	8.08	$\mathbf{1.91 \times 10^{-3}}$	1.73×10^{-4}	$\mathbf{-3.12}$	16.22
LR	8.21×10^{-4}	2.08×10^{-4}	3.08	11.57	1.74×10^{-3}	2.06×10^{-4}	-1.07	5.51
SVR	1.35×10^{-3}	4.26×10^{-4}	-0.35	0.56	$\mathbf{1.91 \times 10^{-3}}$	1.77×10^{-4}	$\mathbf{-4.8}$	**23.04**
KNN	$\mathbf{1.70 \times 10^{-3}}$	2.38×10^{-4}	$\mathbf{-1.93}$	3.70	1.85×10^{-3}	2.68×10^{-4}	-2.71	7.23
XGBoost	1.42×10^{-3}	2.89×10^{-4}	1.89	18.64	1.71×10^{-3}	1.91×10^{-4}	2.09	16.91
LSTM	1.40×10^{-3}	4.99×10^{-4}	$\mathbf{-1.13}$	0.94	1.73×10^{-3}	1.46×10^{-4}	$\mathbf{-3.42}$	14.83
HistData	5.75×10^{-4}	3.88×10^{-5}	$\mathbf{-4.07}$	20.32	5.75×10^{-4}	$\mathbf{3.88 \times 10^{-5}}$	-4.07	20.32
120 days								
HLTM	5.19×10^{-4}	$\mathbf{1.05 \times 10^{-18}}$	0.88	-1.70	5.48×10^{-4}	9.70×10^{-5}	3.81	**34.42**
TBATS	1.85×10^{-4}	1.04×10^{-5}	4.93	**35.12**	3.75×10^{-4}	1.39×10^{-4}	4.79	31.22
ARIMA	3.21×10^{-4}	2.92×10^{-5}	0.02	-0.73	8.56×10^{-4}	8.18×10^{-5}	0.76	15.17
LR	1.14×10^{-3}	4.15×10^{-4}	0.99	-0.89	$\mathbf{1.49 \times 10^{-3}}$	1.76×10^{-4}	0.02	10.42
SVR	1.14×10^{-3}	2.26×10^{-4}	1.64	3.77	1.42×10^{-3}	6.14×10^{-4}	-0.20	-1.11
KNN	1.12×10^{-3}	3.50×10^{-4}	1.7	2.82	1.32×10^{-3}	1.15×10^{-4}	2.09	12.94
XGBoost	1.11×10^{-3}	2.79×10^{-4}	3.45	11.84	1.22×10^{-3}	1.93×10^{-4}	$\mathbf{-0.56}$	12.53
LSTM	$\mathbf{1.15 \times 10^{-3}}$	2.72×10^{-4}	$\mathbf{-0.30}$	1.25	1.43×10^{-3}	1.77×10^{-4}	$\mathbf{-0.29}$	6.53
HistData	5.67×10^{-4}	2.73×10^{-5}	1.05	23.44	5.67×10^{-4}	$\mathbf{2.73 \times 10^{-5}}$	1.05	23.44
150 days								
HLTM	1.13×10^{-4}	1.06×10^{-4}	7.68	**66.53**	$1.40 \times 10{-3}$	8.89×10^{-5}	-0.06	25.79
TBATS	1.11×10^{-4}	1.05×10^{-4}	5.77	41.93	1.38×10^{-3}	1.15×10^{-4}	**-4.95**	25.24
ARIMA	6.94×10^{-4}	1.09×10^{-4}	3.92	31.66	1.65×10^{-3}	1.33×10^{-4}	-4.21	19.16
LR	9.53×10^{-4}	5.63×10^{-5}	**-2.56**	12.75	1.51×10^{-3}	9.19×10^{-5}	-4.3	18.65
SVR	1.17×10^{-3}	3.15×10^{-4}	0.36	3.69	1.75×10^{-3}	1.77×10^{-4}	-3.39	12.69
KNN	9.31×10^{-4}	4.53×10^{-5}	0.21	-0.18	1.76×10^{-3}	1.34×10^{-4}	-3.81	15.44
XGBoost	$\mathbf{1.27 \times 10^{-3}}$	2.31×10^{-4}	1.93	20.23	1.76×10^{-3}	1.14×10^{-4}	-3.77	15.39
LSTM	1.03×10^{-3}	3.38×10^{-4}	1.22	0.37	$\mathbf{1.78 \times 10^{-3}}$	1.26×10^{-4}	-1.01	19.06
HistData	3.55×10^{-4}	$\mathbf{4.14 \times 10^{-5}}$	5.90	43.05	3.55×10^{-4}	$\mathbf{4.14 \times 10^{-5}}$	5.9	**43.05**

Table B.5: Expected portfolio risk summary statistics. Values in bold represent the best results for each row. For reference, the perfect foresight values are 1.14×10^{-3} (30 days), 2.42×10^{-3} (60 days), 2.51×10^{-3} (90 days), 2.58×10^{-3} (120 days), and 2.34×10^{-3} (150 days).

	Out-of-sample				One-day-ahead			
30 days	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HLTM	8.24×10^{-3}	3.41×10^{-5}	6.79	45.67	2.66×10^{-3}	6.95×10^{-4}	3.43	12.68
TBATS	$\mathbf{1.21 \times 10^{-3}}$	2.75×10^{-4}	**8.17**	**72.01**	2.76×10^{-3}	1.26×10^{-3}	4.71	24.11
ARIMA	3.91×10^{-3}	4.27×10^{-5}	-6.72	60.62	8.29×10^{-3}	$\mathbf{5.99 \times 10^{-4}}$	-1.95	7.41
LR	1.86×10^{-3}	3.44×10^{-5}	-0.31	-1.47	1.80×10^{-3}	6.41×10^{-4}	3.74	16.71
SVR	6.16×10^{-3}	1.70×10^{-3}	0.93	0.35	1.79×10^{-3}	9.90×10^{-4}	**5.30**	**29.66**
KNN	3.13×10^{-3}	7.79×10^{-5}	-1.98	2.00	1.90×10^{-3}	1.12×10^{-3}	4.51	23.79
XGBoost	4.61×10^{-3}	9.17×10^{-4}	0.49	0.45	$\mathbf{1.57 \times 10^{-3}}$	8.05×10^{-4}	5.20	27.92
LSTM	3.39×10^{-3}	1.06×10^{-3}	-0.40	-1.47	1.85×10^{-3}	8.39×10^{-4}	5.19	27.54
HistData	3.03×10^{-3}	$\mathbf{7.57 \times 10^{-5}}$	3.94	20.31	3.47×10^{-3}	6.08×10^{-4}	3.42	13.26
60 days								
HLTM	4.51×10^{-3}	$\mathbf{6.24 \times 10^{-6}}$	**7.36**	**69.47**	4.96×10^{-3}	5.93×10^{-4}	4.99	26.20
TBATS	3.89×10^{-3}	7.39×10^{-5}	0.98	2.60	5.17×10^{-3}	$\mathbf{3.52 \times 10^{-4}}$	6.05	38.69
ARIMA	5.05×10^{-3}	2.37×10^{-4}	2.40	6.65	1.38×10^{-2}	1.17×10^{-3}	-3.44	11.89
LR	4.03×10^{-3}	3.89×10^{-3}	4.82	22.82	4.04×10^{-3}	1.38×10^{-3}	4.11	17.53
SVR	6.77×10^{-3}	2.05×10^{-3}	1.44	2.77	3.10×10^{-3}	7.58×10^{-4}	4.50	22.05
KNN	3.84×10^{-3}	4.49×10^{-4}	1.97	4.21	3.54×10^{-3}	7.27×10^{-4}	3.81	15.76
XGBoost	6.00×10^{-3}	1.95×10^{-3}	1.72	2.37	4.13×10^{-3}	8.27×10^{-4}	3.55	13.15
LSTM	5.37×10^{-3}	2.24×10^{-4}	2.61	10.55	$\mathbf{2.86 \times 10^{-3}}$	9.96×10^{-4}	1.96	2.79
HistData	$\mathbf{2.86 \times 10^{-3}}$	5.92×10^{-5}	4.46	23.99	4.52×10^{-3}	1.09×10^{-3}	**6.58**	**49.25**
90 days								
HLTM	5.89×10^{-3}	$\mathbf{2.41 \times 10^{-5}}$	4.44	19.44	5.92×10^{-3}	1.09×10^{-3}	4.99	27.52
TBATS	2.73×10^{-3}	4.62×10^{-5}	2.21	5.91	5.75×10^{-3}	5.45×10^{-4}	4.17	28.29
ARIMA	5.73×10^{-3}	2.92×10^{-4}	2.09	7.58	1.94×10^{-2}	1.65×10^{-3}	-3.17	9.52
LR	3.71×10^{-3}	3.81×10^{-4}	2.05	7.32	4.29×10^{-3}	7.32×10^{-4}	3.14	11.22
SVR	5.78×10^{-3}	1.15×10^{-3}	-0.54	0.74	4.57×10^{-3}	$\mathbf{5.03 \times 10^{-4}}$	4.40	**35.30**
KNN	1.00×10^{-2}	1.55×10^{-3}	-1.84	3.04	4.82×10^{-3}	1.09×10^{-3}	3.82	15.06
XGBoost	8.02×10^{-3}	1.97×10^{-3}	6.38	**55.74**	4.22×10^{-3}	1.37×10^{-3}	**5.47**	**33.86**
LSTM	6.98×10^{-3}	1.55×10^{-3}	-1.27	1.99	$\mathbf{4.20 \times 10^{-3}}$	5.59×10^{-4}	3.17	16.06
HistData	$\mathbf{2.65 \times 10^{-3}}$	1.00×10^{-4}	**6.85**	**52.37**	5.25×10^{-3}	6.86×10^{-4}	**5.26**	**41.50**
120 days								
HLTM	3.93×10^{-3}	$\mathbf{6.94 \times 10^{-18}}$	-1.36	-0.62	4.96×10^{-3}	8.69×10^{-4}	3.62	14.51
TBATS	$\mathbf{2.92 \times 10^{-3}}$	5.79×10^{-5}	-0.07	11.49	7.05×10^{-3}	4.32×10^{-4}	-1.28	15.96
ARIMA	5.42×10^{-3}	1.59×10^{-4}	0.19	-0.84	2.03×10^{-2}	1.75×10^{-3}	-3.89	15.08
LR	7.68×10^{-3}	1.39×10^{-3}	0.36	-1.53	3.89×10^{-3}	6.04×10^{-4}	3.59	15.33
SVR	7.32×10^{-3}	1.81×10^{-3}	-0.78	0.80	2.99×10^{-3}	9.70×10^{-4}	3.69	16.75
KNN	7.05×10^{-3}	4.35×10^{-4}	-3.26	12.28	3.02×10^{-3}	1.33×10^{-3}	4.30	25.96
XGBoost	6.52×10^{-3}	9.75×10^{-4}	3.38	17.98	$\mathbf{2.86 \times 10^{-3}}$	9.66×10^{-4}	2.98	8.12
LSTM	6.32×10^{-3}	9.92×10^{-4}	-0.35	-0.03	3.75×10^{-3}	6.15×10^{-4}	4.72	31.24
HistData	2.85×10^{-3}	1.56×10^{-4}	**7.80**	**66.41**	5.45×10^{-3}	$\mathbf{4.08 \times 10^{-4}}$	**6.07**	**47.25**
150 days								
HLTM	3.93×10^{-3}	$\mathbf{6.94 \times 10^{-18}}$	-1.36	-0.62	9.34×10^{-3}	8.19×10^{-4}	5.41	55.52
TBATS	$\mathbf{2.49 \times 10^{-3}}$	2.15×10^{-4}	**8.02**	**71.54**	9.26×10^{-3}	6.46×10^{-4}	-4.33	26.20
ARIMA	4.67×10^{-3}	1.88×10^{-4}	-7.00	58.73	3.07×10^{-2}	3.39×10^{-3}	-2.51	5.11
LR	5.49×10^{-3}	2.68×10^{-4}	-2.34	9.18	$\mathbf{5.02 \times 10^{-3}}$	6.39×10^{-4}	4.39	26.29
SVR	6.81×10^{-3}	1.20×10^{-3}	0.52	2.55	5.72×10^{-3}	5.65×10^{-4}	4.18	28.53
KNN	5.04×10^{-3}	2.06×10^{-4}	0.48	0.08	5.66×10^{-3}	6.71×10^{-4}	7.41	70.54
XGBoost	7.75×10^{-3}	1.34×10^{-3}	0.06	4.06	5.66×10^{-3}	$\mathbf{4.83 \times 10^{-4}}$	7.95	73.75
LSTM	5.07×10^{-3}	1.29×10^{-3}	1.04	0.74	5.89×10^{-3}	2.16×10^{-3}	**9.45**	**92.23**
HistData	2.76×10^{-3}	9.24×10^{-5}	2.05	5.68	5.13×10^{-3}	1.86×10^{-3}	5.00	30.51

Table B.6: Expected portfolio Sharpe Ratio summary statistics. Values in bold represent the best results for each row. For reference, the perfect foresight values are 4.04×10^{-2} (30 days), 3.72×10^{-2} (60 days), 3.72×10^{-2} (90 days), 3.29×10^{-2} (120 days), and 3.23×10^{-2} (150 days).

	Out-of-sample				One-day-ahead			
30 days	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HLTM	1.04×10^{-2}	**3.60×10^{-6}**	-2.33	11.07	1.75×10^{-2}	**3.05×10^{-3}**	-3.91	19.98
TBATS	4.91×10^{-3}	1.12×10^{-3}	6.83	49.75	1.72×10^{-2}	4.72×10^{-3}	3.40	**33.56**
ARIMA	1.05×10^{-2}	4.22×10^{-4}	**-9.31**	**91.19**	1.35×10^{-2}	4.67×10^{-3}	0.02	-0.07
LR	1.88×10^{-2}	1.86×10^{-4}	0.62	-1.07	3.12×10^{-2}	7.08×10^{-3}	-2.91	8.87
SVR	**2.55×10^{-2}**	2.69×10^{-3}	0.73	-1.05	**3.43×10^{-2}**	6.00×10^{-3}	-2.91	20.73
KNN	2.49×10^{-2}	1.06×10^{-4}	2.18	4.11	3.29×10^{-2}	6.30×10^{-3}	-3.27	9.95
XGBoost	2.23×10^{-2}	9.52×10^{-3}	0.34	-0.59	3.17×10^{-2}	4.36×10^{-3}	-4.21	18.83
LSTM	2.29×10^{-2}	1.72×10^{-3}	1.00	1.39	3.42×10^{-2}	5.65×10^{-3}	**-4.63**	**22.79**
HistData	1.83×10^{-2}	4.83×10^{-3}	-2.73	9.20	1.83×10^{-2}	4.83×10^{-3}	-2.73	9.20
60 days								
HLTM	5.11×10^{-3}	**3.84×10^{-5}**	2.94	**31.95**	9.99×10^{-3}	1.93×10^{-3}	6.42	**55.21**
TBATS	3.54×10^{-3}	4.58×10^{-4}	3.18	28.98	3.67×10^{-3}	1.65×10^{-3}	5.27	28.47
ARIMA	9.17×10^{-3}	8.21×10^{-4}	2.21	3.88	1.79×10^{-2}	1.54×10^{-3}	**-2.30**	**18.00**
LR	1.35×10^{-2}	3.37×10^{-3}	1.87	6.47	2.96×10^{-2}	3.66×10^{-3}	-2.97	9.76
SVR	**1.81×10^{-2}**	5.12×10^{-3}	1.55	2.16	2.66×10^{-2}	3.68×10^{-3}	**-3.95**	**18.05**
KNN	1.62×10^{-2}	5.44×10^{-4}	2.29	7.67	2.96×10^{-2}	4.07×10^{-3}	-3.79	16.15
XGBoost	**2.01×10^{-2}**	5.06×10^{-3}	0.42	1.05	**3.23×10^{-2}**	4.21×10^{-3}	-2.99	8.49
LSTM	1.69×10^{-2}	4.21×10^{-4}	6.80	60.02	2.69×10^{-2}	4.96×10^{-3}	-0.71	4.29
HistData	1.21×10^{-2}	1.40×10^{-3}	**-1.80**	**21.04**	1.21×10^{-2}	**1.40×10^{-3}**	**-1.80**	**21.04**
90 days								
HLTM	8.20×10^{-3}	**3.81×10^{-5}**	4.14	16.48	1.26×10^{-2}	2.05×10^{-3}	-2.80	15.94
TBATS	2.89×10^{-3}	2.54×10^{-4}	6.11	**47.25**	1.25×10^{-2}	1.85×10^{-3}	-4.93	**26.76**
ARIMA	4.91×10^{-3}	7.47×10^{-4}	2.80	7.84	1.36×10^{-2}	1.12×10^{-3}	0.46	22.37
LR	1.31×10^{-2}	2.66×10^{-3}	2.40	7.35	2.64×10^{-2}	3.10×10^{-3}	-3.15	10.96
SVR	**1.74×10^{-2}**	4.71×10^{-3}	-0.60	0.01	**2.80×10^{-2}**	2.64×10^{-3}	-4.22	18.00
KNN	1.67×10^{-2}	1.17×10^{-3}	-2.41	7.18	2.69×10^{-2}	4.65×10^{-3}	-2.80	7.19
XGBoost	1.57×10^{-2}	1.98×10^{-3}	-2.16	13.26	2.63×10^{-2}	1.80×10^{-3}	-3.77	17.19
LSTM	1.62×10^{-2}	5.58×10^{-3}	-1.05	0.51	2.65×10^{-2}	2.40×10^{-3}	**-4.85**	**25.12**
HistData	1.08×10^{-2}	8.27×10^{-4}	**-4.01**	17.80	1.08×10^{-2}	**8.27×10^{-4}**	**-4.01**	17.80
120 days								
HLTM	7.55×10^{-3}	**1.42×10^{-17}**	1.26	-1.46	7.98×10^{-3}	1.26×10^{-3}	1.27	21.04
TBATS	3.07×10^{-3}	2.14×10^{-4}	5.67	**44.64**	4.23×10^{-3}	1.50×10^{-3}	4.39	27.30
ARIMA	4.10×10^{-3}	3.37×10^{-4}	-0.07	-0.70	5.90×10^{-3}	**7.67×10^{-4}**	4.19	19.43
LR	1.26×10^{-2}	3.75×10^{-3}	1.08	-0.80	**2.36×10^{-2}**	1.95×10^{-3}	**-5.10**	**41.21**
SVR	1.60×10^{-2}	5.80×10^{-3}	-0.40	-1.39	2.07×10^{-2}	2.45×10^{-3}	-1.34	7.96
KNN	1.55×10^{-2}	1.12×10^{-3}	2.95	12.13	2.02×10^{-2}	3.86×10^{-3}	-0.84	5.29
XGBoost	1.49×10^{-2}	1.88×10^{-3}	**-3.60**	16.67	2.04×10^{-2}	2.67×10^{-3}	0.54	8.29
LSTM	**1.67×10^{-2}**	2.40×10^{-3}	-0.79	1.29	2.31×10^{-2}	2.01×10^{-3}	**-2.99**	11.44
HistData	1.03×10^{-2}	4.56×10^{-4}	-3.21	15.81	2.71×10^{-4}	9.52×10^{-4}	8.17	**76.41**
150 days								
HLTM	7.76×10^{-3}	**1.42×10^{-17}**	1.26	-1.46	1.43×10^{-2}	7.98×10^{-4}	1.94	22.08
TBATS	3.38×10^{-3}	1.97×10^{-4}	4.57	52.51	1.41×10^{-2}	8.73×10^{-4}	-4.74	23.67
ARIMA	3.88×10^{-3}	1.03×10^{-3}	-1.69	2.36	9.32×10^{-3}	**5.83×10^{-4}**	-0.83	16.33
LR	1.26×10^{-2}	4.93×10^{-4}	**-2.97**	17.70	2.12×10^{-2}	1.39×10^{-3}	-3.54	13.33
SVR	1.38×10^{-2}	2.79×10^{-3}	-0.30	2.18	2.30×10^{-2}	2.52×10^{-3}	-3.78	16.36
KNN	1.28×10^{-2}	3.84×10^{-4}	-0.03	-0.14	**2.32×10^{-2}**	1.65×10^{-3}	-3.57	13.34
XGBoost	**1.42×10^{-2}**	1.70×10^{-3}	0.80	11.74	**2.32×10^{-2}**	1.73×10^{-3}	**-4.75**	**24.98**
LSTM	1.40×10^{-2}	3.04×10^{-3}	0.73	-0.64	**2.32×10^{-2}**	1.63×10^{-3}	-3.30	11.59
HistData	1.30×10^{-2}	4.39×10^{-4}	**-6.82**	**61.77**	7.15×10^{-3}	1.58×10^{-3}	0.59	20.76

Appendix C

Technical Analysis Indicators

Table C.1: RMSE summary statistics for REITs. Values in bold represent the best results for each row.

	Out-of-sample				One-day-ahead			
	Without TA		With TA		Without TA		With TA	
30 days	Mean	SD	Mean	SD	Mean	SD	Mean	SD
LR	5.60	12.49	**4.55**	**7.55**	**1.04**	2.10	**1.04**	**1.38**
SVR	5.59	12.45	**4.13**	**7.28**	**1.02**	2.01	1.06	**1.41**
KNN	5.61	12.53	**4.06**	**7.36**	1.03	2.04	**1.02**	**1.41**
XGBOOST	5.60	12.49	**3.83**	**6.53**	**1.02**	2.00	1.04	**1.62**
LSTM	5.60	12.57	**3.54**	**5.44**	1.08	2.16	**1.01**	**1.24**
HLTM	21.77	40.15			6.47	14.23		
TBATS	21.77	40.15			6.47	14.23		
ARIMA	21.47	38.98			6.69	14.68		
60 days								
LR	7.47	14.79	**6.97**	**14.09**	2.40	5.76	**2.22**	**4.00**
SVR	7.46	14.75	**5.66**	**10.05**	2.40	5.76	**2.33**	**4.04**
KNN	7.48	14.82	**5.73**	**11.05**	2.39	5.75	**2.29**	**4.58**
XGBOOST	7.49	14.87	**5.36**	**8.00**	2.39	5.75	**2.28**	**3.71**
LSTM	7.56	14.50	**5.31**	**7.92**	2.40	5.75	**2.27**	**2.58**
HLTM	16.87	35.63			10.28	24.67		
TBATS	16.87	35.63			10.28	24.67		
ARIMA	17.08	35.82			10.60	25.29		
90 days								
LR	9.70	19.79	**5.94**	**10.75**	1.15	2.18	**0.99**	**0.94**
SVR	9.69	19.73	**6.67**	**12.19**	1.13	2.12	**0.98**	**0.95**
KNN	9.70	19.74	**6.64**	**12.14**	1.13	2.12	**0.99**	**1.06**
XGBOOST	9.70	19.78	**5.81**	**9.72**	1.13	2.13	**0.97**	**0.91**
LSTM	9.72	19.86	**6.08**	**8.58**	1.14	2.16	**0.92**	**0.88**
HLTM	20.82	35.66			9.30	17.45		
TBATS	21.28	36.75			9.30	17.45		
ARIMA	20.81	35.67			9.47	17.78		
120 days								
LR	10.96	16.75	**7.19**	**11.43**	1.16	2.22	**1.14**	**1.23**
SVR	10.95	16.73	**6.63**	**10.10**	1.14	2.16	**1.11**	**1.22**
KNN	10.97	16.79	**6.55**	**8.55**	**1.14**	2.16	**1.14**	**1.23**
XGBOOST	10.95	16.75	**6.30**	**8.92**	**1.14**	2.16	1.15	**1.32**
LSTM	10.99	16.81	**5.87**	**7.39**	1.17	2.23	**1.15**	**1.25**
HLTM	22.91	35.97			9.83	15.19		
TBATS	22.91	35.97			9.83	15.19		
ARIMA	22.88	35.95			10.01	15.51		
150 days								
LR	8.00	12.91	**5.90**	**10.27**	1.16	2.19	**1.11**	**1.96**
SVR	8.00	12.91	**5.14**	**6.90**	1.15	2.16	**1.05**	**1.10**
KNN	8.02	12.96	**5.11**	**6.59**	1.15	2.16	**1.06**	**1.21**
XGBOOST	8.00	12.91	**4.97**	**7.46**	1.15	2.17	**1.04**	**1.11**
LSTM	8.00	12.92	**4.51**	**7.52**	1.16	2.18	**1.04**	**1.04**
HLTM	17.32	27.37			7.91	12.70		
TBATS	17.32	27.37			7.91	12.70		
ARIMA	16.91	26.80			8.07	13.00		

Table C.2: RMSE summary statistics for stocks. Values in bold represent the best results for each row.

	Out-of-sample				One-day-ahead			
	Without TA		With TA		Without TA		With TA	
30 days	Mean	SD	Mean	SD	Mean	SD	Mean	SD
LR	9.19	20.62	**7.61**	**16.62**	2.28	4.38	**2.18**	**2.34**
SVR	9.16	20.51	**6.84**	**12.92**	2.29	4.44	**2.16**	**2.60**
KNN	9.21	20.66	**7.69**	**14.72**	2.30	4.45	**2.15**	**2.18**
XGBOOST	9.19	20.58	**6.25**	**10.31**	2.30	4.48	**2.17**	**2.29**
LSTM	9.11	20.21	**6.38**	**9.47**	2.31	4.39	**2.13**	**2.27**
HLTM	41.21	100.20			11.36	26.54		
TBATS	41.21	100.20			11.36	26.54		
ARIMA	41.47	100.15			11.72	27.15		
60 days								
LR	12.34	23.85	**11.00**	**21.10**	2.77	5.64	**2.72**	**3.05**
SVR	12.32	23.75	**10.48**	**21.89**	2.77	5.64	**2.65**	**3.10**
KNN	12.30	23.80	**11.31**	**21.99**	2.76	5.63	**2.54**	**2.76**
XGBOOST	12.31	23.75	**9.35**	**16.18**	2.77	5.63	**2.70**	**3.03**
LSTM	12.29	23.67	**10.03**	**13.88**	2.77	5.63	**2.55**	**2.23**
HLTM	30.29	71.90			12.38	24.16		
TBATS	30.29	71.90			12.38	24.16		
ARIMA	31.30	75.72			12.73	24.77		
90 days								
LR	19.45	43.66	**15.37**	**30.84**	**3.25**	6.97	3.29	**2.78**
SVR	19.45	43.63	**13.20**	**24.21**	3.35	7.31	**3.33**	**2.77**
KNN	19.39	43.57	**14.11**	**27.87**	**3.24**	6.96	3.29	**2.79**
XGBOOST	19.44	43.61	**11.89**	**18.21**	**3.25**	7.00	3.27	**2.63**
LSTM	19.44	43.53	**11.73**	**18.59**	3.25	6.97	**3.24**	**2.64**
HLTM	42.37	98.42			18.72	44.32		
TBATS	42.85	100.01			18.72	44.32		
ARIMA	42.37	98.45			19.08	44.93		
120 days								
LR	28.90	85.13	**21.89**	**62.52**	3.39	7.47	**3.38**	**2.41**
SVR	28.87	84.97	**18.29**	**52.29**	3.45	7.70	**3.14**	**2.57**
KNN	28.82	84.91	**17.19**	**29.41**	3.36	7.43	**3.35**	**2.51**
XGBOOST	28.88	85.06	**16.79**	**22.44**	3.39	7.49	**3.21**	**2.33**
LSTM	28.89	85.01	**16.56**	**16.86**	3.36	7.39	**3.18**	**2.27**
HLTM	62.94	192.98			28.82	81.52		
TBATS	62.94	192.98			28.82	81.52		
ARIMA	62.76	193.89			29.20	82.25		
150 days								
LR	28.62	75.28	**22.93**	**52.34**	3.28	7.15	**3.21**	**1.96**
SVR	28.55	75.08	**20.01**	**34.49**	3.28	7.18	**3.14**	**2.08**
KNN	28.58	75.07	**20.02**	**38.75**	3.27	7.13	**3.07**	**2.02**
XGBOOST	28.62	75.24	**19.09**	**24.23**	3.27	7.15	**3.16**	**3.13**
LSTM	28.46	74.86	**19.03**	**23.88**	3.27	7.13	**3.12**	**2.20**
HLTM	71.50	190.19			29.09	75.40		
TBATS	71.50	190.19			29.09	75.40		
ARIMA	71.46	190.44			28.78	75.06		

Table C.3: RMSE summary statistics for bonds. Values in bold represent the best results for each row.

	Out-of-sample				One-day-ahead			
	Without TA		With TA		Without TA		With TA	
30 days	Mean	SD	Mean	SD	Mean	SD	Mean	SD
LR	0.51	0.56	**0.49**	**0.55**	**0.17**	**0.18**	0.21	0.20
SVR	0.51	**0.56**	**0.45**	0.60	**0.17**	0.18	0.19	**0.17**
KNN	0.51	**0.56**	**0.44**	0.62	**0.17**	0.18	**0.17**	**0.17**
XGBOOST	0.51	0.56	**0.43**	**0.41**	**0.17**	**0.18**	0.18	0.21
LSTM	0.52	0.56	**0.45**	**0.37**	0.18	0.18	**0.17**	**0.16**
HLTM	1.22	1.48			0.48	0.57		
TBATS	1.16	1.37			0.48	0.57		
ARIMA	1.22	1.48			0.51	0.60		
60 days								
LR	**0.58**	0.73	**0.58**	**0.55**	0.17	0.17	**0.16**	**0.15**
SVR	0.58	0.73	**0.55**	**0.58**	**0.17**	0.17	**0.17**	**0.15**
KNN	0.58	0.73	**0.52**	**0.56**	**0.17**	**0.17**	0.18	**0.17**
XGBOOST	0.58	0.73	**0.53**	**0.64**	**0.17**	0.17	**0.17**	**0.13**
LSTM	0.59	0.74	**0.56**	**0.31**	**0.18**	0.18	**0.18**	**0.13**
HLTM	0.93	1.24			0.60	0.68		
TBATS	0.93	1.24			0.60	0.68		
ARIMA	0.96	1.29			0.62	0.69		
90 days								
LR	0.87	0.89	**0.61**	**0.52**	**0.20**	**0.20**	0.22	0.24
SVR	0.87	0.89	**0.79**	**0.68**	**0.20**	0.20	0.20	**0.15**
KNN	0.87	0.89	**0.74**	**0.59**	0.20	0.19	**0.18**	**0.17**
XGBOOST	0.87	0.90	**0.63**	**0.51**	**0.20**	**0.20**	0.21	0.22
LSTM	0.88	0.90	**0.66**	**0.40**	**0.20**	0.20	**0.20**	**0.17**
HLTM	1.74	2.05			0.85	0.86		
TBATS	1.74	2.05			0.85	0.86		
ARIMA	1.72	2.02			0.87	0.88		
120 days								
LR	0.94	1.12	**0.91**	**0.79**	**0.19**	0.19	**0.19**	**0.18**
SVR	0.93	1.10	**0.81**	**0.66**	**0.19**	**0.19**	0.20	0.24
KNN	0.93	1.12	**0.79**	**0.65**	**0.19**	0.18	0.20	**0.15**
XGBOOST	0.94	1.12	**0.78**	**0.75**	0.20	0.20	**0.19**	**0.16**
LSTM	0.94	1.12	**0.77**	**0.49**	0.20	**0.19**	**0.18**	0.20
HLTM	2.05	2.48			0.99	1.19		
TBATS	2.05	2.48			0.99	1.19		
ARIMA	2.07	2.51			1.01	1.20		
150 days								
LR	1.03	1.26	**0.94**	**1.10**	0.20	0.19	**0.19**	**0.16**
SVR	1.03	1.26	**0.94**	**0.47**	**0.19**	**0.19**	0.20	**0.19**
KNN	1.04	1.26	**0.93**	**0.94**	**0.20**	0.19	0.21	**0.16**
XGBOOST	1.03	1.26	**0.95**	**0.89**	0.20	0.19	**0.18**	**0.14**
LSTM	1.04	1.25	**0.94**	**0.90**	0.20	0.19	**0.18**	**0.16**
HLTM	1.79	2.37			1.03	1.28		
TBATS	1.79	2.37			1.03	1.28		
ARIMA	1.83	2.41			1.05	1.29		

Table C.4: Expected portfolio return summary statistics. Values in bold represent the best results for each row.

	Out-of-sample				One-day-ahead			
	Without TA		With TA		Without TA		With TA	
30 days	Mean	SD	Mean	SD	Mean	SD	Mean	SD
LR	1.12×10^{-3}	$\mathbf{4.75\times10^{-6}}$	$\mathbf{3.07\times10^{-3}}$	2.16×10^{-4}	1.31×10^{-3}	3.52×10^{-4}	$\mathbf{3.51\times10^{-3}}$	$\mathbf{1.93\times10^{-4}}$
SVR	1.44×10^{-3}	3.95×10^{-4}	$\mathbf{3.40\times10^{-3}}$	$\mathbf{2.02\times10^{-4}}$	1.44×10^{-3}	4.92×10^{-4}	$\mathbf{3.55\times10^{-3}}$	$\mathbf{1.67\times10^{-4}}$
KNN	1.43×10^{-3}	$\mathbf{1.25\times10^{-5}}$	$\mathbf{3.46\times10^{-3}}$	2.05×10^{-4}	1.38×10^{-3}	2.95×10^{-4}	$\mathbf{3.78\times10^{-3}}$	$\mathbf{1.69\times10^{-4}}$
XGBoost	1.23×10^{-3}	5.04×10^{-4}	$\mathbf{3.45\times10^{-3}}$	$\mathbf{2.06\times10^{-4}}$	1.47×10^{-3}	$\mathbf{1.50\times10^{-4}}$	$\mathbf{3.49\times10^{-3}}$	1.99×10^{-4}
LSTM	1.44×10^{-3}	$\mathbf{1.69\times10^{-4}}$	$\mathbf{3.36\times10^{-3}}$	1.86×10^{-4}	2.72×10^{-3}	2.37×10^{-4}	$\mathbf{3.70\times10^{-3}}$	$\mathbf{1.57\times10^{-4}}$
HLTM	9.06×10^{-4}	1.78×10^{-6}			9.62×10^{-4}	1.79×10^{-4}		
TBATS	1.93×10^{-4}	7.73×10^{-5}			9.02×10^{-4}	3.79×10^{-4}		
ARIMA	6.73×10^{-4}	2.85×10^{-5}			1.25×10^{-3}	4.35×10^{-4}		
60 days								
LR	8.40×10^{-4}	3.49×10^{-4}	$\mathbf{2.98\times10^{-3}}$	$\mathbf{2.13\times10^{-4}}$	1.86×10^{-3}	1.97×10^{-4}	$\mathbf{3.40\times10^{-3}}$	$\mathbf{1.45\times10^{-4}}$
SVR	1.52×10^{-3}	6.36×10^{-4}	$\mathbf{2.51\times10^{-3}}$	$\mathbf{2.34\times10^{-4}}$	1.48×10^{-3}	$\mathbf{1.90\times10^{-4}}$	$\mathbf{3.13\times10^{-3}}$	2.16×10^{-4}
KNN	1.02×10^{-3}	$\mathbf{9.18\times10^{-5}}$	$\mathbf{2.90\times10^{-3}}$	1.66×10^{-4}	1.75×10^{-3}	2.17×10^{-4}	$\mathbf{3.49\times10^{-3}}$	$\mathbf{1.72\times10^{-4}}$
XGBoost	1.58×10^{-3}	6.06×10^{-4}	$\mathbf{3.46\times10^{-3}}$	$\mathbf{1.86\times10^{-4}}$	2.07×10^{-3}	2.28×10^{-4}	$\mathbf{3.66\times10^{-3}}$	$\mathbf{1.76\times10^{-4}}$
LSTM	1.26×10^{-3}	$\mathbf{4.09\times10^{-5}}$	$\mathbf{2.40\times10^{-3}}$	2.45×10^{-4}	1.45×10^{-3}	4.50×10^{-4}	$\mathbf{3.62\times10^{-3}}$	$\mathbf{1.80\times10^{-4}}$
HLTM	3.81×10^{-4}	2.33×10^{-6}			6.90×10^{-4}	1.55×10^{-4}		
TBATS	2.40×10^{-4}	2.78×10^{-5}			2.84×10^{-4}	1.34×10^{-4}		
ARIMA	6.72×10^{-4}	7.48×10^{-5}			2.12×10^{-3}	2.16×10^{-4}		
90 days								
LR	8.21×10^{-4}	$\mathbf{2.08\times10^{-4}}$	$\mathbf{2.43\times10^{-3}}$	2.35×10^{-4}	1.74×10^{-3}	$\mathbf{2.06\times10^{-4}}$	$\mathbf{2.79\times10^{-3}}$	2.37×10^{-4}
SVR	1.35×10^{-3}	4.26×10^{-4}	$\mathbf{2.64\times10^{-3}}$	$\mathbf{2.58\times10^{-4}}$	1.91×10^{-3}	$\mathbf{1.77\times10^{-4}}$	$\mathbf{3.48\times10^{-3}}$	1.85×10^{-4}
KNN	1.70×10^{-3}	$\mathbf{2.38\times10^{-4}}$	$\mathbf{2.44\times10^{-3}}$	2.60×10^{-4}	1.85×10^{-3}	2.68×10^{-4}	$\mathbf{3.03\times10^{-3}}$	$\mathbf{2.07\times10^{-4}}$
XGBoost	1.42×10^{-3}	2.89×10^{-4}	$\mathbf{2.85\times10^{-3}}$	$\mathbf{2.31\times10^{-4}}$	1.71×10^{-3}	$\mathbf{1.91\times10^{-4}}$	$\mathbf{3.00\times10^{-3}}$	1.99×10^{-4}
LSTM	1.40×10^{-3}	4.99×10^{-4}	$\mathbf{2.32\times10^{-3}}$	$\mathbf{2.90\times10^{-4}}$	1.73×10^{-3}	$\mathbf{1.46\times10^{-4}}$	$\mathbf{3.06\times10^{-3}}$	2.03×10^{-4}
HLTM	6.49×10^{-4}	4.14×10^{-6}			9.84×10^{-4}	1.73×10^{-4}		
TBATS	1.70×10^{-4}	1.39×10^{-5}			9.62×10^{-4}	1.48×10^{-4}		
ARIMA	3.92×10^{-4}	6.81×10^{-5}			1.91×10^{-3}	1.73×10^{-4}		
120 days								
LR	1.14×10^{-3}	4.15×10^{-4}	$\mathbf{2.21\times10^{-3}}$	$\mathbf{2.73\times10^{-4}}$	1.49×10^{-3}	$\mathbf{1.76\times10^{-4}}$	$\mathbf{2.72\times10^{-3}}$	2.33×10^{-4}
SVR	1.14×10^{-3}	2.26×10^{-4}	$\mathbf{2.08\times10^{-3}}$	$\mathbf{2.47\times10^{-4}}$	1.42×10^{-3}	6.14×10^{-4}	$\mathbf{3.75\times10^{-3}}$	$\mathbf{1.99\times10^{-4}}$
KNN	1.12×10^{-3}	3.50×10^{-4}	$\mathbf{1.91\times10^{-3}}$	1.97×10^{-4}	1.32×10^{-3}	$\mathbf{1.15\times10^{-4}}$	$\mathbf{2.92\times10^{-3}}$	1.77×10^{-4}
XGBoost	1.11×10^{-3}	2.79×10^{-4}	$\mathbf{2.13\times10^{-3}}$	2.55×10^{-4}	1.22×10^{-3}	$\mathbf{1.93\times10^{-4}}$	$\mathbf{2.57\times10^{-3}}$	1.97×10^{-4}
LSTM	1.15×10^{-3}	2.72×10^{-4}	$\mathbf{2.09\times10^{-3}}$	$\mathbf{1.98\times10^{-4}}$	1.43×10^{-3}	$\mathbf{1.77\times10^{-4}}$	$\mathbf{2.76\times10^{-3}}$	2.26×10^{-4}
HLTM	5.19×10^{-4}	1.05×10^{-18}			5.48×10^{-4}	9.70×10^{-5}		
TBATS	1.85×10^{-4}	1.04×10^{-5}			3.75×10^{-4}	1.39×10^{-4}		
ARIMA	3.21×10^{-4}	2.92×10^{-5}			8.56×10^{-4}	8.18×10^{-5}		
150 days								
LR	9.53×10^{-4}	$\mathbf{5.63\times10^{-5}}$	$\mathbf{1.99\times10^{-3}}$	2.43×10^{-4}	1.51×10^{-3}	$\mathbf{9.19\times10^{-5}}$	$\mathbf{3.01\times10^{-3}}$	1.87×10^{-4}
SVR	1.17×10^{-3}	3.15×10^{-4}	$\mathbf{2.22\times10^{-3}}$	$\mathbf{2.09\times10^{-4}}$	1.75×10^{-3}	$\mathbf{1.77\times10^{-4}}$	$\mathbf{2.61\times10^{-3}}$	2.29×10^{-4}
KNN	9.31×10^{-4}	$\mathbf{4.53\times10^{-5}}$	$\mathbf{2.26\times10^{-3}}$	2.15×10^{-4}	1.76×10^{-3}	$\mathbf{1.34\times10^{-4}}$	$\mathbf{2.98\times10^{-3}}$	2.20×10^{-4}
XGBoost	1.27×10^{-3}	$\mathbf{2.31\times10^{-4}}$	$\mathbf{2.45\times10^{-3}}$	2.37×10^{-4}	1.76×10^{-3}	$\mathbf{1.14\times10^{-4}}$	$\mathbf{3.01\times10^{-3}}$	2.39×10^{-4}
LSTM	1.03×10^{-3}	3.38×10^{-4}	$\mathbf{2.41\times10^{-3}}$	$\mathbf{1.94\times10^{-4}}$	1.78×10^{-3}	$\mathbf{1.26\times10^{-4}}$	$\mathbf{2.73\times10^{-3}}$	1.72×10^{-4}
HLTM	1.13×10^{-4}	1.06×10^{-4}			1.40×10^{-3}	8.89×10^{-5}		
TBATS	1.11×10^{-4}	1.05×10^{-4}			1.38×10^{-3}	1.15×10^{-4}		
ARIMA	6.94×10^{-4}	1.09×10^{-4}			1.65×10^{-3}	1.33×10^{-4}		

Table C.5: Expected portfolio risk summary statistics. Values in bold represent the best results for each row.

	Out-of-sample				One-day-ahead			
	Without TA		With TA		Without TA		With TA	
30 days	Mean	SD	Mean	SD	Mean	SD	Mean	SD
LR	$\mathbf{1.86 \times 10^{-3}}$	$\mathbf{3.44 \times 10^{-5}}$	1.07×10^{-2}	1.85×10^{-4}	$\mathbf{1.80 \times 10^{-3}}$	6.41×10^{-4}	1.12×10^{-2}	$\mathbf{1.31 \times 10^{-4}}$
SVR	$\mathbf{6.16 \times 10^{-3}}$	1.70×10^{-3}	1.11×10^{-2}	$\mathbf{1.40 \times 10^{-4}}$	$\mathbf{1.79 \times 10^{-3}}$	9.90×10^{-4}	1.11×10^{-2}	$\mathbf{1.27 \times 10^{-4}}$
KNN	$\mathbf{3.13 \times 10^{-3}}$	$\mathbf{7.79 \times 10^{-5}}$	1.12×10^{-2}	1.17×10^{-4}	$\mathbf{1.90 \times 10^{-3}}$	1.12×10^{-3}	1.13×10^{-2}	$\mathbf{1.28 \times 10^{-4}}$
XGBoost	$\mathbf{4.61 \times 10^{-3}}$	9.17×10^{-4}	1.12×10^{-2}	$\mathbf{1.31 \times 10^{-4}}$	$\mathbf{1.57 \times 10^{-3}}$	8.05×10^{-4}	1.12×10^{-2}	$\mathbf{1.32 \times 10^{-4}}$
LSTM	3.39×10^{-3}	1.06×10^{-3}	$\mathbf{1.09 \times 10^{-2}}$	$\mathbf{1.48 \times 10^{-4}}$	$\mathbf{1.85 \times 10^{-3}}$	8.39×10^{-4}	1.13×10^{-2}	$\mathbf{1.13 \times 10^{-4}}$
HLTM	8.24×10^{-3}	3.41×10^{-5}			2.66×10^{-3}	6.95×10^{-4}		
TBATS	1.21×10^{-3}	2.75×10^{-4}			2.76×10^{-3}	1.26×10^{-3}		
ARIMA	3.91×10^{-3}	4.27×10^{-5}			8.29×10^{-3}	5.99×10^{-4}		
60 days								
LR	$\mathbf{4.03 \times 10^{-3}}$	3.89×10^{-3}	1.07×10^{-2}	$\mathbf{1.65 \times 10^{-4}}$	$\mathbf{4.04 \times 10^{-3}}$	1.38×10^{-3}	1.09×10^{-2}	$\mathbf{1.18 \times 10^{-4}}$
SVR	$\mathbf{6.77 \times 10^{-3}}$	2.05×10^{-3}	1.04×10^{-2}	$\mathbf{1.73 \times 10^{-4}}$	$\mathbf{3.10 \times 10^{-3}}$	7.58×10^{-4}	1.08×10^{-2}	$\mathbf{1.64 \times 10^{-4}}$
KNN	$\mathbf{3.84 \times 10^{-3}}$	4.49×10^{-4}	1.04×10^{-2}	$\mathbf{1.47 \times 10^{-4}}$	$\mathbf{3.54 \times 10^{-3}}$	7.27×10^{-4}	1.11×10^{-2}	$\mathbf{1.06 \times 10^{-4}}$
XGBoost	$\mathbf{6.00 \times 10^{-3}}$	1.95×10^{-3}	1.10×10^{-2}	$\mathbf{1.46 \times 10^{-4}}$	$\mathbf{4.13 \times 10^{-3}}$	8.27×10^{-4}	1.12×10^{-2}	$\mathbf{1.30 \times 10^{-4}}$
LSTM	$\mathbf{5.37 \times 10^{-3}}$	2.24×10^{-4}	1.02×10^{-2}	$\mathbf{1.93 \times 10^{-4}}$	$\mathbf{2.86 \times 10^{-3}}$	9.96×10^{-4}	1.12×10^{-2}	$\mathbf{1.30 \times 10^{-4}}$
HLTM	4.51×10^{-3}	6.24×10^{-6}			4.96×10^{-3}	5.93×10^{-4}		
TBATS	3.89×10^{-3}	7.39×10^{-5}			5.17×10^{-3}	3.52×10^{-4}		
ARIMA	5.05×10^{-3}	2.37×10^{-4}			1.38×10^{-2}	1.17×10^{-3}		
90 days								
LR	$\mathbf{3.71 \times 10^{-3}}$	3.81×10^{-4}	1.04×10^{-2}	$\mathbf{1.54 \times 10^{-4}}$	$\mathbf{4.29 \times 10^{-3}}$	7.32×10^{-4}	1.06×10^{-2}	$\mathbf{1.72 \times 10^{-4}}$
SVR	$\mathbf{5.78 \times 10^{-3}}$	1.15×10^{-3}	1.07×10^{-2}	$\mathbf{1.81 \times 10^{-4}}$	$\mathbf{4.57 \times 10^{-3}}$	5.03×10^{-4}	1.11×10^{-2}	$\mathbf{1.35 \times 10^{-4}}$
KNN	$\mathbf{1.00 \times 10^{-2}}$	1.55×10^{-3}	1.07×10^{-2}	$\mathbf{1.61 \times 10^{-4}}$	$\mathbf{4.82 \times 10^{-3}}$	1.09×10^{-3}	1.08×10^{-2}	$\mathbf{1.42 \times 10^{-4}}$
XGBoost	$\mathbf{8.02 \times 10^{-3}}$	1.97×10^{-3}	1.05×10^{-2}	$\mathbf{2.14 \times 10^{-4}}$	$\mathbf{4.22 \times 10^{-3}}$	1.37×10^{-3}	1.07×10^{-2}	$\mathbf{1.50 \times 10^{-4}}$
LSTM	$\mathbf{6.98 \times 10^{-3}}$	$\mathbf{1.55 \times 10^{-3}}$	1.07×10^{-2}	1.71×10^{-4}	$\mathbf{4.20 \times 10^{-3}}$	5.59×10^{-4}	1.07×10^{-2}	$\mathbf{1.49 \times 10^{-4}}$
HLTM	5.89×10^{-3}	2.41×10^{-5}			5.92×10^{-3}	1.09×10^{-3}		
TBATS	2.73×10^{-3}	4.62×10^{-5}			5.75×10^{-3}	5.45×10^{-4}		
ARIMA	5.73×10^{-3}	2.92×10^{-4}			1.94×10^{-2}	1.65×10^{-3}		
120 days								
LR	$\mathbf{7.68 \times 10^{-3}}$	1.39×10^{-3}	1.04×10^{-2}	$\mathbf{1.94 \times 10^{-4}}$	$\mathbf{3.89 \times 10^{-3}}$	6.04×10^{-4}	1.05×10^{-2}	$\mathbf{1.74 \times 10^{-4}}$
SVR	$\mathbf{7.32 \times 10^{-3}}$	1.81×10^{-3}	1.02×10^{-2}	$\mathbf{1.65 \times 10^{-4}}$	$\mathbf{2.99 \times 10^{-3}}$	9.70×10^{-4}	1.00×10^{-2}	$\mathbf{1.55 \times 10^{-4}}$
KNN	$\mathbf{7.05 \times 10^{-3}}$	4.35×10^{-4}	9.65×10^{-3}	$\mathbf{1.57 \times 10^{-4}}$	$\mathbf{3.02 \times 10^{-3}}$	1.33×10^{-3}	1.05×10^{-2}	$\mathbf{1.44 \times 10^{-4}}$
XGBoost	$\mathbf{6.52 \times 10^{-3}}$	9.75×10^{-4}	1.00×10^{-2}	$\mathbf{2.19 \times 10^{-4}}$	$\mathbf{2.86 \times 10^{-3}}$	9.66×10^{-4}	1.02×10^{-2}	$\mathbf{1.76 \times 10^{-4}}$
LSTM	$\mathbf{6.32 \times 10^{-3}}$	9.92×10^{-4}	9.91×10^{-3}	$\mathbf{1.41 \times 10^{-4}}$	$\mathbf{3.75 \times 10^{-3}}$	6.15×10^{-4}	1.06×10^{-2}	$\mathbf{1.70 \times 10^{-4}}$
HLTM	3.93×10^{-3}	6.94×10^{-18}			4.96×10^{-3}	8.69×10^{-4}		
TBATS	2.92×10^{-3}	5.79×10^{-5}			7.05×10^{-3}	4.32×10^{-4}		
ARIMA	5.42×10^{-3}	1.59×10^{-4}			2.03×10^{-2}	1.75×10^{-3}		
150 days								
LR	$\mathbf{5.49 \times 10^{-3}}$	2.68×10^{-4}	9.92×10^{-3}	$\mathbf{1.97 \times 10^{-4}}$	$\mathbf{5.02 \times 10^{-3}}$	6.39×10^{-4}	1.06×10^{-2}	$\mathbf{1.44 \times 10^{-4}}$
SVR	$\mathbf{6.81 \times 10^{-3}}$	1.20×10^{-3}	9.97×10^{-3}	$\mathbf{1.67 \times 10^{-4}}$	$\mathbf{5.72 \times 10^{-3}}$	5.65×10^{-4}	1.04×10^{-2}	$\mathbf{1.79 \times 10^{-4}}$
KNN	$\mathbf{5.04 \times 10^{-3}}$	2.06×10^{-4}	9.99×10^{-3}	$\mathbf{1.78 \times 10^{-4}}$	$\mathbf{5.66 \times 10^{-3}}$	6.71×10^{-4}	1.06×10^{-2}	$\mathbf{1.80 \times 10^{-4}}$
XGBoost	$\mathbf{7.75 \times 10^{-3}}$	$\mathbf{1.34 \times 10^{-3}}$	1.03×10^{-2}	1.81×10^{-4}	$\mathbf{5.66 \times 10^{-3}}$	4.83×10^{-4}	1.08×10^{-2}	$\mathbf{1.73 \times 10^{-4}}$
LSTM	$\mathbf{5.07 \times 10^{-3}}$	1.29×10^{-3}	1.01×10^{-2}	$\mathbf{1.56 \times 10^{-4}}$	$\mathbf{5.89 \times 10^{-3}}$	2.16×10^{-3}	1.04×10^{-2}	$\mathbf{1.11 \times 10^{-4}}$
HLTM	3.93×10^{-3}	6.94×10^{-18}			9.34×10^{-3}	8.19×10^{-4}		
TBATS	2.49×10^{-3}	2.15×10^{-4}			9.26×10^{-3}	6.46×10^{-4}		
ARIMA	4.67×10^{-3}	1.88×10^{-4}			3.07×10^{-2}	3.39×10^{-3}		

Table C.6: Expected portfolio Sharpe Ratio summary statistics. Values in bold represent the best results for each row.

	Out-of-sample				One-day-ahead			
	Without TA		With TA		Without TA		With TA	
30 days	Mean	SD	Mean	SD	Mean	SD	Mean	SD
LR	1.88×10^{-2}	$\mathbf{1.86\times10^{-4}}$	$\mathbf{2.78\times10^{-2}}$	2.15×10^{-3}	3.12×10^{-2}	7.08×10^{-3}	$\mathbf{3.21\times10^{-2}}$	$\mathbf{1.86\times10^{-3}}$
SVR	2.55×10^{-2}	2.69×10^{-3}	$\mathbf{3.10\times10^{-2}}$	$\mathbf{1.95\times10^{-3}}$	$\mathbf{3.43\times10^{-2}}$	6.00×10^{-3}	3.28×10^{-2}	$\mathbf{1.58\times10^{-3}}$
KNN	2.49×10^{-2}	$\mathbf{1.06\times10^{-4}}$	$\mathbf{3.15\times10^{-2}}$	1.96×10^{-3}	3.29×10^{-2}	6.30×10^{-3}	$\mathbf{3.45\times10^{-2}}$	$\mathbf{1.66\times10^{-3}}$
XGBOOST	2.23×10^{-2}	9.52×10^{-3}	$\mathbf{3.14\times10^{-2}}$	$\mathbf{1.99\times10^{-3}}$	3.17×10^{-2}	4.36×10^{-3}	$\mathbf{3.19\times10^{-2}}$	$\mathbf{1.91\times10^{-3}}$
LSTM	2.29×10^{-2}	$\mathbf{1.72\times10^{-3}}$	$\mathbf{3.08\times10^{-2}}$	1.78×10^{-3}	$\mathbf{3.42\times10^{-2}}$	5.65×10^{-3}	3.40×10^{-2}	$\mathbf{1.50\times10^{-3}}$
HLTM	1.04×10^{-2}	3.60×10^{-6}			1.75×10^{-2}	3.05×10^{-3}		
TBATS	4.91×10^{-3}	1.12×10^{-3}			1.72×10^{-2}	4.72×10^{-3}		
ARIMA	1.05×10^{-2}	4.22×10^{-4}			1.35×10^{-2}	4.67×10^{-3}		
60 days								
LR	1.35×10^{-2}	3.37×10^{-3}	$\mathbf{2.72\times10^{-2}}$	$\mathbf{2.09\times10^{-3}}$	2.96×10^{-2}	3.66×10^{-3}	$\mathbf{3.18\times10^{-2}}$	$\mathbf{1.41\times10^{-3}}$
SVR	1.81×10^{-2}	5.12×10^{-3}	$\mathbf{2.29\times10^{-2}}$	$\mathbf{2.30\times10^{-3}}$	2.66×10^{-2}	3.68×10^{-3}	$\mathbf{2.84\times10^{-2}}$	$\mathbf{2.13\times10^{-3}}$
KNN	1.62×10^{-2}	$\mathbf{5.44\times10^{-4}}$	$\mathbf{2.72\times10^{-2}}$	1.64×10^{-3}	2.96×10^{-2}	4.07×10^{-3}	$\mathbf{3.23\times10^{-2}}$	$\mathbf{1.67\times10^{-3}}$
XGBOOST	2.01×10^{-2}	5.06×10^{-3}	$\mathbf{3.17\times10^{-2}}$	$\mathbf{1.79\times10^{-3}}$	3.23×10^{-2}	4.21×10^{-3}	$\mathbf{3.34\times10^{-2}}$	$\mathbf{1.67\times10^{-3}}$
LSTM	1.69×10^{-2}	$\mathbf{4.21\times10^{-4}}$	$\mathbf{2.16\times10^{-2}}$	2.43×10^{-3}	2.69×10^{-2}	4.96×10^{-3}	$\mathbf{3.31\times10^{-2}}$	$\mathbf{1.74\times10^{-3}}$
HLTM	5.11×10^{-3}	3.84×10^{-5}			9.99×10^{-3}	1.93×10^{-3}		
TBATS	3.54×10^{-3}	4.58×10^{-4}			3.67×10^{-3}	1.65×10^{-3}		
ARIMA	9.17×10^{-3}	8.21×10^{-4}			1.79×10^{-2}	1.54×10^{-3}		
90 days								
LR	1.31×10^{-2}	2.66×10^{-3}	$\mathbf{2.23\times10^{-2}}$	$\mathbf{2.29\times10^{-3}}$	$\mathbf{2.64\times10^{-2}}$	3.10×10^{-3}	2.53×10^{-2}	$\mathbf{2.33\times10^{-3}}$
SVR	1.74×10^{-2}	4.71×10^{-3}	$\mathbf{2.40\times10^{-2}}$	$\mathbf{2.46\times10^{-3}}$	2.80×10^{-2}	2.64×10^{-3}	$\mathbf{3.20\times10^{-2}}$	$\mathbf{1.75\times10^{-3}}$
KNN	1.67×10^{-2}	$\mathbf{1.17\times10^{-3}}$	$\mathbf{2.21\times10^{-2}}$	2.48×10^{-3}	2.69×10^{-2}	4.65×10^{-3}	$\mathbf{2.79\times10^{-2}}$	$\mathbf{2.01\times10^{-3}}$
XGBOOST	1.57×10^{-2}	$\mathbf{1.98\times10^{-3}}$	$\mathbf{2.56\times10^{-2}}$	2.27×10^{-3}	2.63×10^{-2}	$\mathbf{1.80\times10^{-3}}$	$\mathbf{2.77\times10^{-2}}$	1.92×10^{-3}
LSTM	1.62×10^{-2}	5.58×10^{-3}	$\mathbf{2.06\times10^{-2}}$	$\mathbf{2.82\times10^{-3}}$	$\mathbf{2.65\times10^{-2}}$	2.40×10^{-3}	2.82×10^{-2}	$\mathbf{1.98\times10^{-3}}$
HLTM	8.20×10^{-3}	3.81×10^{-3}			1.26×10^{-2}	2.05×10^{-3}		
TBATS	2.89×10^{-3}	2.54×10^{-4}			1.25×10^{-2}	1.85×10^{-3}		
ARIMA	4.91×10^{-3}	7.47×10^{-4}			1.36×10^{-2}	1.12×10^{-3}		
120 days								
LR	1.26×10^{-2}	3.75×10^{-3}	$\mathbf{1.97\times10^{-2}}$	$\mathbf{2.29\times10^{-3}}$	$\mathbf{2.36\times10^{-2}}$	$\mathbf{1.95\times10^{-3}}$	2.47×10^{-2}	2.31×10^{-3}
SVR	1.60×10^{-2}	5.80×10^{-3}	$\mathbf{1.89\times10^{-2}}$	$\mathbf{2.46\times10^{-3}}$	2.07×10^{-2}	2.45×10^{-3}	$\mathbf{2.17\times10^{-2}}$	$\mathbf{2.01\times10^{-3}}$
KNN	1.55×10^{-2}	$\mathbf{1.12\times10^{-3}}$	$\mathbf{1.79\times10^{-2}}$	2.48×10^{-3}	2.02×10^{-2}	3.86×10^{-3}	$\mathbf{2.74\times10^{-2}}$	$\mathbf{1.73\times10^{-3}}$
XGBOOST	1.49×10^{-2}	$\mathbf{1.88\times10^{-3}}$	$\mathbf{1.90\times10^{-2}}$	2.27×10^{-3}	2.04×10^{-2}	2.67×10^{-3}	$\mathbf{2.39\times10^{-2}}$	$\mathbf{1.95\times10^{-3}}$
LSTM	1.67×10^{-2}	$\mathbf{2.40\times10^{-3}}$	$\mathbf{1.98\times10^{-2}}$	2.82×10^{-3}	2.31×10^{-2}	$\mathbf{2.01\times10^{-3}}$	$\mathbf{2.52\times10^{-2}}$	2.17×10^{-3}
HLTM	7.55×10^{-3}	1.42×10^{-17}			7.98×10^{-3}	1.26×10^{-3}		
TBATS	3.07×10^{-3}	2.14×10^{-4}			4.23×10^{-3}	1.50×10^{-3}		
ARIMA	4.10×10^{-3}	3.37×10^{-4}			5.90×10^{-3}	7.67×10^{-4}		
150 days								
LR	1.26×10^{-2}	$\mathbf{4.93\times10^{-4}}$	$\mathbf{1.78\times10^{-2}}$	2.39×10^{-3}	2.12×10^{-2}	$\mathbf{1.39\times10^{-3}}$	$\mathbf{2.80\times10^{-2}}$	1.84×10^{-3}
SVR	1.38×10^{-2}	2.79×10^{-3}	$\mathbf{2.22\times10^{-2}}$	$\mathbf{2.09\times10^{-4}}$	2.30×10^{-2}	2.52×10^{-3}	$\mathbf{2.38\times10^{-2}}$	$\mathbf{2.22\times10^{-3}}$
KNN	1.28×10^{-2}	$\mathbf{3.84\times10^{-4}}$	$\mathbf{2.05\times10^{-2}}$	2.05×10^{-3}	2.32×10^{-2}	$\mathbf{1.65\times10^{-3}}$	$\mathbf{2.71\times10^{-2}}$	2.15×10^{-3}
XGBOOST	1.42×10^{-2}	$\mathbf{1.70\times10^{-3}}$	$\mathbf{2.23\times10^{-2}}$	2.36×10^{-3}	2.32×10^{-2}	$\mathbf{1.73\times10^{-3}}$	$\mathbf{2.72\times10^{-2}}$	2.33×10^{-3}
LSTM	1.40×10^{-2}	3.04×10^{-3}	$\mathbf{2.25\times10^{-2}}$	$\mathbf{1.92\times10^{-3}}$	2.32×10^{-2}	$\mathbf{1.63\times10^{-3}}$	$\mathbf{2.58\times10^{-2}}$	1.73×10^{-3}
HLTM	7.76×10^{-3}	1.42×10^{-17}			1.43×10^{-2}	7.98×10^{-4}		
TBATS	3.38×10^{-3}	1.97×10^{-4}			1.41×10^{-2}	8.73×10^{-4}		
ARIMA	3.88×10^{-3}	1.03×10^{-3}			9.32×10^{-3}	5.83×10^{-4}		