University of Essex
School of Computer Science and Electronic Engineering

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

Fatim Zahra Habbab

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computational Finance of the University of Essex,
May 8, 2024

Dedication

To my father, who left us early.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Dr Michael Kampouridis, for their invaluable help, support, and guidance throughout the entire journey of my PhD. Their expertise, patience, and encouragement have been crucial in shaping my research and academic growth. I would also like to extend my appreciation to my colleagues at University of Essex, whose insights and discussions enriched my research experience. I am also grateful to my mother who has supported me in all possible ways during my PhD journey, and believed in me even when I could not. Finally, I thank God for letting me through all the difficulties. I will keep trusting you for my future.

Abstract

Investing in real estate presents an attractive option for investors seeking to enhance their returns and mitigate the inherent risks associated with other asset classes such as stocks and bonds. The advantages of real estate investments can be significant, offering diversification benefits and potential for long-term appreciation. However, gaining access to real estate markets typically requires substantial capital, which may not be feasible for all investors, particularly individuals.

To address this barrier to entry, investors often turn to alternative investment vehicles such as Real Estate Investment Trusts (REITs). These investment vehicles allow individuals to indirectly invest in real estate by purchasing shares in companies that own and manage a portfolio of income-generating properties. By doing so, investors can benefit from the income potential and capital appreciation associated with real estate without the need for direct ownership.

In the context of a diversified investment portfolio that already includes stocks and bonds, the potential benefits of incorporating real estate investments are of particular interest. This study focuses on assessing the added value of real estate within a multi-asset portfolio and explores innovative methodologies to optimize asset allocation.

Rather than relying solely on historical data, the study adopts a forward-looking approach by utilizing future price predictions. By employing machine learning algorithms such as Ordinary Least Squares Linear Regression, Support Vector Regression, k-Nearest Neighbors Regression, Extreme Gradient Boosting, and Long/Short-Term Memory Neural Networks, the study aims to generate accurate predictions of asset prices.

To further enhance the predictive accuracy of the machine learning models, the study incorporates additional features represented by Technical Analysis Indicators (TAIs). These indicators provide valuable insights into market trends and help refine the price prediction process.

In addition to predicting asset prices, the study employs a Genetic Algorithm (GA) to determine optimal portfolio weightings. By considering the expected returns and risks associated with each asset class, the GA identifies the most efficient allocation strategy for a portfolio comprising REITs, stocks, and bonds.

Ultimately, the study evaluates the performance of portfolios constructed using price predictions versus those based on historical data. By comparing the diversification benefits and risk-adjusted returns of these portfolios, the study provides valuable insights into the role of real estate in a mixed-asset investment strategy.

Through a comprehensive analysis that integrates several machine learning techniques, technical analysis indicators, and optimization algorithms, the study aims to explore the potential advantages of real estate investments in a diversified portfolio context. By leveraging future price predictions, investors can make more informed decisions and enhance their overall investment outcomes.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Objectives

The motivation behind this research lies in the crucial role of optimizing portfolios that incorporate real estate within the finance domain [1]. Achieving an optimal asset allocation is fundamental for minimizing risk and maximizing 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, optimizing 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 optimizing 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 optimizing 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 utilizing 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, emphasizing 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 optimizing 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 optimizing 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 Neighbors 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 optimize 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 optimized 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 optimization, 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 machine learning algorithms used in this thesis, both for optimization – 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 Neighbors Regression (KNN). Chapter 3 presents a literature review about different portfolio optimization 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 optimization 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. This will potentially lead to better portfolio performance results as compared to the historical data approach in optimizing the weights of a mixed-asset portfolio including 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 in thesis in Peer-Reviewed Journals are 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 in thesis in Conference Proceedings are 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 optimize 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 optimization), 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 categorized 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, urbanization, 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:

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 centers, 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 accommodations and lodging purposes, such as hotels, resorts, and vacation rentals[31]. This market caters to the hospitality industry, providing spaces for travelers and tourists seeking temporary stays[31].

Agricultural Real Estate Market. The agricultural market involves properties utilized 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].

Specialized Real Estate Markets. Specialized real estate markets are tailored to specific niche segments within the broader real estate industry, such as healthcare real estate (hospitals, medical centers), 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 customized 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 customization, 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 investments. 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 analyzing 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 specialized 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 centralized platforms where the buying and selling of shares of publicly traded companies occurs. 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 analyze financial statements, company performance metrics, and macroeconomic data to make informed investment decisions[43].

Trading in stock markets takes place on organized 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 revolutionized 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 behavior, 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 minimize 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 minimize 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 behaviors 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 optimize 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 behavior 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].

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 serves as an essential metric in assessing the potential profitability of an investment. It represents the average return that an investor anticipates receiving from that asset over a specific period of time. Expected returns can be estimated using historical data, fundamental analysis, or various other quantitative and qualitative techniques.

When constructing a portfolio, MPT takes into account the expected returns of individual assets. 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]. This process considers the asset allocation, which determines the proportion of the portfolio allocated to different assets or asset classes.

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; w_i = the weight of the ith asset in the portfolio; and $E[R_i]$ = the expected return of the ith asset.

By diversifying across different assets with varying expected returns, investors can potentially enhance the overall return of their portfolio. MPT recognizes that combining assets with different expected returns can help balance the portfolio's risk and return profile. For instance, while higher expected returns may be associated with greater risk, combining assets with lower expected returns and lower risk can help achieve a more stable and well-rounded portfolio.

In addition to expected return, MPT also considers the expected portfolio risk. Risk is a fundamental aspect of investing, representing the uncertainty or potential variability of investment returns. In MPT, risk is commonly measured by the standard deviation of asset or portfolio returns, which provides an indication of 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_i is the weight of the ith asset in the portfolio; and σ_i is the standard deviation of the ith asset.

By assessing the expected return and risk of a portfolio, investors can make informed decisions that align with their risk preferences and investment goals. MPT aims to find an optimal combination of assets that maximizes the expected return for a given level of risk or minimizes the risk for a desired level of return. This process involves constructing a portfolio along the efficient frontier, which represents the set of portfolios that offer the highest expected return for a given level of risk or the lowest risk for a given level of return.

By incorporating expected return and risk as key factors in the investment decision-making process, MPT provides a systematic and quantitative framework for constructing portfolios that balance risk and return. This approach has greatly influenced portfolio management practices and has been widely adopted by investors seeking to optimize their investment allocations.

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: $\rho_{i,j}$ is the correlation between assets i and j; $cov(R_i, R_j)$ is the covariance between the returns of assets i and j; σ_i is the standard deviation of the returns of asset i; and σ_j is the standard deviation of the returns of asset j

A high correlation between two assets indicates that their returns tend to move in the same direction, while a low correlation indicates that their returns tend to move independently of each other.

In MPT, diversification is used to reduce risk by investing in assets that are not perfectly correlated with each other. By investing in a diverse portfolio of assets with low correlations, investors can reduce the overall risk of their portfolio.

In conclusion, MPT assumes that an optimal portfolio can be built using information including the expected return, risk, and correlations between assets. 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 optimization 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 optimization problem first, and the regression problem later, we follow the same order in this section.

2.4.1 ML for optimization

To solve our optimization 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 optimization 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 optimization problems (e.g., optimizing 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 optimized. 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*. 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 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. The individuals with best fitness are then included in the next population.

2.4.2 ML for regression

Supervised machine learning, or supervised learning, refers to machine learning algorithms that use labeled 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 recognize some data, and draw conclusions on how those data should be labeled. 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 variable. In the case of one independent variable, the algorithm is called *simple linear regression*; when there are more than one independent variable, the algorithm is called *multiple linear regression*. This is different than *multivariate linear regression*, in which involves multiple correlated dependent variables.

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

Given a dataset $\{y_i,\, x_{i1}, \ldots, x_{ip}\}_{i=1}^{n}$ 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

$$y_i = \sum_{i=1}^{p} \beta_i x_i + \epsilon_i \tag{2.4}$$

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

Support Vector Regression The Support Vector Regression (SVR) algorithm attempts to minimize 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.

As the table above suggests, SVR is made of the following components: (a) hyperplane; (b) kernel; (c) boundary lines; and (d) support vector. The hyperplane is a separation line between two data classes in a higher dimension than the actual dimension. The kernel function helps map the data points into the higher dimension. Boundary lines are drawn around the hyperplane at a distance of ϵ. Support vector is a vector that is used to define the hyperplane. The goal of a SVR is to fit the largest number of data points possible without violating the margin.

The kernel functions that can be used in the SVR algorithm are represented in Table 2.1. Each of

Kernel function	**Formulation**
Gaussian	$K_G(x^i, x^j, \theta) = exp(\sum_k^{N_D} \theta_k \lvert x_k^i - x_k^j \rvert^2)$
Polynomial	$K(x, y) = \tanh{(\gamma, x^T y + c)^d}, \gamma > 0$
Radial basis function (RBF)	$K(x, y) = \exp^{-}(\gamma \lvert x - y \rvert^2)$
Sigmoid	$K(x, y) = \tanh{(\gamma, x^T y + r)}$

Table 2.1: SVR kernel functions.

Distance function	**Formulation**
Euclidean	$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$
Manhattan	$\sum_{i=1}^k \lvert x_i - y_i \rvert$
Minkowski	$(\sum_{i=1}^k (\lvert x_i - y_i \rvert)^q)^{1/q}$

Table 2.2: SVR kernel functions.

those functions has hyperparameters that need to be tuned, such as γ or kernel coefficient. 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 attempts to find a set number k of observations in the training data set that are closest to a given observation x. To do so, it calculates the average of the numerical target of the k-nearest neighbors. Another approach uses an inverse weighted average of the k-nearest neighbors. The functions used to estimate the distance between two points are represented in Table 2.2.

The optimal value for k depends on the training dataset used. In general, a larger value for k leads to more accurate models as it is able to reduce noise; however, a ‘too high‘ value for k will make the feature space blurred.

KNN algorithms are defined as lazy learner algorithms because they do not learn from the training set immediately, but it learns after the training set is stored. This means that the output prediction takes place only at a later stage, following the storage of the training set.

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 over-fitting through the regularization 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.

$$Obj^{(t)} = \sum_{i=1}^{t} L(y_i, \hat{y}_i) + \sum_{i=1}^{t} \Omega(f_i), \tag{2.5}$$

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 regularization term.

Long Short Term Memory Long Short Term Memory (LSTM) is a variation of Recurrent Neural Networks (RNNs) that is able to learn long-term dependencies. LSTM has the ability to process an entire sequence of data using feedback connections. For this reason, it is suitable for sequence prediction problems. It can thus be applied to the prediction of time series, such as financial time series.

An LSTM unit is commonly composed of a cell, an input gate, an output gate, and a forget gate. The input gate determines which of the input values can be used to change the memory. The sigmoid function (Eq. 2.6) determines which values can be used by assigning 0 or 1 to each value, where 0 indicates that the information cannot pass from one layer to the other, while 1 means that all the information can be let through. The tanh function (Eq. 2.7) assigns weight to the data that can be any value between -1 and 1.

$$i_t = \sigma(W_i[h_t - 1, x_t] + b_i), \tag{2.6}$$

$$C_t = \tanh(W_c[h_t - 1, x_t] + b_c) \tag{2.7}$$

The forget gate finds the information that should be removed from the block. Such decision is made by a sigmoid function (Eq. 2.8). For each value in the cell state C_{t-1}, it looks at the preceding state h_{t-1} and the content input X_t to produce a number between 0 (which means ‘omit this‘) and 1 (which means ‘keep this‘).

$$f_t = \sigma(W_f[h_t - 1, x_t] + b_f) \tag{2.8}$$

In the output gate, the sigmoid function (Eq 2.9) determines whether to allow information through by assigning a 0 or 1 value. And the tanh function (Eq 2.10) assigns the weights to the values provided, determining their importance on a scale of -1 to 1, and multiplying each sigmoid output with the weight assigned.

$$O_t = \sigma(W_o[h_t - 1, x_t] + b_o) \tag{2.9}$$

$$h_t = o_t * \tanh(C_t) \tag{2.10}$$

2.5 Summary

In this section we summarize the background information provided in this chapter through the following key points.

Financial markets facilitate global fund transfer through virtual exchanges of diverse instruments, shaping economies by allocating capital, managing risk, and pricing financial assets. Financial markets, virtual arenas for the exchange of diverse financial instruments such as stocks, bonds, currencies, commodities, derivatives, and real estate, serve as crucial conduits for transferring funds among individuals, institutions, and governments. These markets play a crucial role in the global economy, aiding in capital allocation, risk management, and the pricing of financial instruments. Categorically, financial markets include primary markets, where new securities are initially issued, enabling entities to raise capital, and secondary markets, where previously issued securities are traded among investors, offering higher liquidity. The primary asset classes traded in these markets include stocks, bonds, and real estate.

Real estate markets transact diverse properties; the study emphasizes REITs for indirect investment benefits, urging careful evaluation before investing. Real estate markets involve virtual transactions of buying, selling, and leasing various property types, tied to specific geographic areas, influenced by factors such as location, amenities, economic conditions, and government policies, with key categories including residential, commercial, industrial, retail, office, hospitality, agricultural, and specialized markets tailored to niche segments; the study focuses on Real Estate Investment Trusts (REITs) as an indirect investment option providing liquidity, diversification, and regular income through mandatory taxable income distributions, offering investors exposure to diverse real estate assets and sectors, emphasizing careful evaluation of factors like track record, management expertise, portfolio quality, and associated risks before investment.

MPT minimizes risk through diversified investments, considering expected return, risk, and asset correlations, despite criticisms for its assumptions. Modern Portfolio Theory (MPT) emphasizes minimizing risk for a given return by diversifying investments across asset classes and relies on mean-variance analysis. While assuming rational and risk-averse investors, it has faced criticism for assumptions like normal return distribution and rational behavior. Despite critiques, MPT has significantly influenced portfolio management, considering factors like expected portfolio return, risk, and correlations between assets. The theory aims to construct portfolios along the efficient frontier for an optimal risk-return balance, utilizing diversification to reduce risk through low-correlation assets. Ongoing efforts address limitations and enhance risk management practices in the context of correlations.

Genetic algorithms evolve candidate populations, optimizing solutions, and are applied to complex problems like portfolio weight optimization. Genetic algorithms (GAs), inspired by biological evolution, provide a powerful approach to solving optimization problems, such as optimizing investment portfolios. Operating on a population of candidate solutions, GAs utilize genetic operators like crossover and mutation to transform the population iteratively. The stochastic selection process, guided by a fitness function, generates new offspring individuals, allowing GAs to explore diverse solution spaces. The representation of individuals, often binary or numeric, involves genes representing variables to be optimized. Elitism ensures the preservation of high-fitness individuals, while the rest undergo probabilistic selection, often through tournament selection. GAs excel in tackling complex optimization challenges by searching through a set of possible solutions rather than relying on a singular candidate solution.

Supervised machine learning involves labeled datasets for tasks like price prediction, utilizing algorithms such as Linear Regression, Support Vector Regression, K-Nearest Neighbor, Long Short Term Memory, and Extreme Gradient Boosting. Supervised machine learning involves algorithms that use labeled datasets for tasks like classifying or predicting outcomes. The focus of this research is on regression that aims at understanding relationships between variables. Linear Regression models these relationships using linear predictor functions, estimating parameters through the least squares approach. Support Vector Regression minimizes error within a threshold, employing hyperplanes, kernels, boundary lines, and support vectors. K-Nearest Neighbor finds k-closest observations, using distance functions like Euclidean or Manhattan. Extreme Gradient Boosting is a scalable, distributed algorithm known for its efficiency and performance, utilizing decision trees with parameters like maximum depth and learning rate. Long Short Term Memory, a variant of Recurrent Neural Networks, handles sequence prediction with gates determining information flow.

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 optimizing portfolios with real estate, this study employs price predictions for these three asset classes. This chapter provides a review of the literature about portfolio optimization 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 analyze the amount of work that has been done on various portfolio optimization 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. Finally, we summarize our conclusions in Section 3.4.

3.2 Portfolio optimization

In this Section, we explore the techniques that have been utilized for portfolio optimization in the case of stock, bond, and real estate investments. Specifically, we will analyze the mathematical models used in solving portfolio optimization 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 optimization techniques in the case of real estate investments 3.2.2.

3.2.1 Portfolio optimization techniques

The mathematical models used for portfolio optimization 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 optimization technique used in portfolio optimization to determine the optimal allocation of assets within a portfolio. It involves formulating the problem as a linear objective function, which seeks to either minimize or maximize 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, such as the maximum allocation allowed for each asset or asset class [59]. In their study, [59] proposed a robust LP model for portfolio optimization under uncertain conditions, incorporating risk-return analysis. Similarly, [60] employed a linear objective optimization approach to mitigate investment portfolio risk. [61] compared a LP-based portfolio with a benchmark portfolio and demonstrated that the former outperformed the latter in terms of Sharpe ratio. [62] applied LP to an asset allocation problem in the case of real estate investments.

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 minimization, in addition to the mean-variance trade-off. [63] proposed a parallel variable neighborhood search algorithm combined with quadratic programming to solve a cardinality constrained portfolio optimization problem. Similarly, [64] solved a portfolio optimization problem with cardinality constraint using a combination of double roulette wheel selection and QP. Their proposed algorithm achieved better accuracy and greater computational efficiency than other algorithms in the literature, including heuristic algorithms. [65] used QP to model the cardinality constraints of a portfolio optimization problem, and noticed a reduction in the overall computational time and an improvement in the evaluation metrics. [66, 67] used QP in solving a portfolio optimization problem involving real estate.

Nonlinear programming (NLP). Nonlinear programming is a mathematical optimization technique used to solve optimization problems with nonlinear objective functions and/or nonlinear constraints. It involves finding the optimal values of decision variables that satisfy the given constraints while optimizing the objective function. Such technique has been widely applied to portfolio optimization problems, where the objective function and constraints may involve nonlinear relationships between asset weights and portfolio performance measures. For instance, [68] used a nonlinear programming technique to solve a portfolio optimization problem subject to transaction costs. [69] attempted to address the complexities of the investment market by considering additional factors such as transaction costs and diversification in portfolio selection. They utilized a single objective mixed-integer nonlinear programming model to solve a fuzzy portfolio selection problem.

Stochastic programming. Stochastic programming is a mathematical optimization technique used for portfolio optimization that incorporates uncertainty into the decision-making process. In stochastic programming for portfolio optimization, the objective is to find the optimal allocation of assets that maximizes 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. [70] focused on multi-period portfolio optimization, and incorporated the expected return, Conditional Value at Risk (CVaR), and liquidity criteria. They used stochastic programming to model the stochastic nature of market movements. [71] investigated a multi-period, stochastic portfolio optimization model for diversified fund selection. [72] proposed a scenario-based, multi-stage stochastic programming model to deal with multi-period portfolio optimization problem with cardinality constraints and proportional transaction costs. [73] proposed a stochastic programming model to solve a financial optimization problem involving the real estate asset class.

Genetic algorithm (GA). A GA is a computational optimization technique inspired by the principles of natural selection and genetics. It is used for portfolio optimization to search for an optimal asset allocation that maximizes a predefined objective function, such as risk-adjusted return or portfolio diversification. In their study, [74] addressed the problem of portfolio selection in an uncertain economic environment using an improved genetic algorithm. They demonstrated that such approach is effective in dealing with stochastic variables such as stock returns. [75] used a multi-objective genetic algorithm approach to solve a portfolio optimization problem in the case of stock investments. [76] focused on the problem of group stock portfolio optimization and proposed a genetic algorithm for diverse group stock portfolio optimization. In a similar way, [77, 78, 79] proposed a genetic algorithm-based approach to solve a portfolio optimization problem in the case of stock investments.

GA has been previously used for portfolio optimization problems involving real estate. For instance, [80] presented a GA model that constructs an investment portfolio including real estate with reduced risk under uncertainty conditions. Their study shows that GA is effective in finding an optimal portfolio composition in the case of real estate investments. In another study, [81] adopted a GA-based model to optimize a mixed-asset portfolio including real estate by using historical market data. Their findings suggested that GA could effectively optimize portfolios including different asset classes. However, the literature has provided limited insights into the inclusion of real estate investments, such as real estate investment trusts (REITs), within the optimization process. Incorporating real estate into portfolio optimization using genetic algorithms presents unique challenges and opportunities. Real estate assets possess distinct risk and return characteristics compared to stocks and bonds, and their integration could potentially enhance diversification and improve risk-adjusted returns. In our study, we aim to fill this gap by investigating the incorporation of real estate investments, including REITs, within the genetic algorithm-based portfolio optimization framework. This would involve considering the specific risk/return characteristics and correlation values that are unique to that asset class [82].

3.2.2 Real estate portfolio optimization

The literature review we present in this Section aims to analyze the existing research on the inclusion of direct and securitized 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 securitized 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].

Securitized real estate. Securitized 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. Securitized 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] utilized 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 centered 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,

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

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 optimization 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) utilizing the price predictions to allocate optimal weights to each asset via an optimization algorithm that is based on a specific metric, such as risk or return. Another option is to perform the optimization 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 optimizing mixed-asset portfolios has been utilized before, it has not yet been applied to portfolios that include REITs. Previous research that utilized portfolio optimization with REITs relied on the optimal weights computed in the training set, as noted by [95, 96, 84]. Our research, on the other hand, emphasizes the accurate prediction of REIT prices. Such task is crucial since the prices are utilized as input in the portfolio optimization step.

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 judgment 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 optimization. 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 utilized 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] utilized 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. [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 utilized for predictive purposes. Technical analysts utilize a wide range of tools and techniques to analyze 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 analyzing 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.

3.4 Summary

From our literature review, we can summarize our findings as follows.

Including real estate in a mixed-asset portfolio can potentially improve its performance. Previous studies demonstrated the superior performance of a mixed-asset portfolio including real estate as compared to a portfolio that does not include that asset class. This is due to a lower correlation and more steady historical returns that typically feature the real estate time series with respect to the other asset classes. However, most of the studies investigated the role of real estate in a mixed-asset portfolio used historical data in solving their portfolio optimization problem. Thus, the focus of this research will be to examine the improvement in the mixed-asset portfolio performance lead by investing in real estate and considering price predictions instead of historical data. In Chapter 4, we solve a porfolio optimization problem involving REITs, and at the same time, we demonstrate the superior performance that can be achieved by adopting price predictions.

There is little contribution regarding the prediction of REITs through ML. We demonstrated that ML techniques can outperform econometric approaches, such as ARIMA, in the prediction of real estate prices. However, the current literature about financial forecasting mainly focused on the prediction of stocks. The aim of this research is to explore the potential of ML techniques in the prediction of REITs (see Chapter 5).

Machine learning models are able to outperform econometric approaches for price prediction. The current literature demonstrated that machine learning techniques, such as ANN and SVM, exhibit superior performance in price prediction compared to traditional time series models like ARIMA. The ML algorithms can effectively capture non-linear and complex relationships featuring financial data, leading to more accurate predictions. Thus, in this research we aim to improve the accuracy of predictions of stocks, bonds, and REITs by using ML algorithms, and we compared them against three benchmarks, i.e., HES, TBATS, and ARIMA. This will be done in Chapter 5.

The accuracy of price predictions can be improved using technical analysis indicators. Several studies demonstrated that the accuracy of stock predictions can be improved using TA indicators. This is due to the ability of these features to capture some relationships in financial data that can help predict future trends in the time series. However, there is little contribution about the use of TA indicators in the prediction of REITs. Therefore, we aim to fill this gap by including

TA indicators as additional features in the prediction of REITs. In Chapter 6, we compare the performance of ML algorithms that incorporate TA indicators against other algorithms that use previous prices as unique features for our regression problem.

Chapter 4

Optimizing 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 optimize returns and minimize portfolio risk by utilizing 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 optimization 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 optimization 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 optimization process.

The rest of this chapter is organized 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 utilized 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, utilized to train the machine learning model; a validation set, employed for optimizing the model's hyperparameters; and a testing set, representing the unseen data used in

[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 analyze 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

the final evaluation stage following model tuning and training.

4.2.2 Portfolio optimization under perfect foresight

As previously explained, our goal is to compare the perfect foresight approach in solving a portfolio optimization 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.

The methodology used in this work follows two steps. The first step consists of optimizing 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 optimization 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 optimization strategy would result in better portfolio performance than in the case of optimal weights calculated on historical average returns.

4.2.3 Portfolio optimization via a Genetic Algorithm

As explained before, *genetic algorithms* (GAs) offer a computational approach to address a portfolio optimization problem by mimicking the principles of natural selection and evolution. This section explores the principal aspects of the genetic algorithm applied to portfolio optimization, 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, and the weight for those asset are 0.5, 0.2, and 0.3, respectively. 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 normalization 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 optimization 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 optimizing 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 optimization (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 utilize 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.

[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}

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 (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. 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.

Figure 4.1: Correlation matrix between asset classes.

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]. Its main purpose is to automatically configure optimization algorithms by finding the most appropriate settings, given a set of instances of a problem.

In our case, I/F-Race was applied to data for the period from June 2017 to December 2018. The following twelve months (January-December 2019) 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-December 2020 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

Table 4.2: I-Race Parameter Tuning Results.

Parameter	**Value**
Tournament size	3
Population size	300
Mutation rate	0.01
Number of generations	10

are presented in Table 4.2.

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 optimization 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 optimization 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%. [7]

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

4.4.1 Summary statistics

As previously explained, under the perfect foresight situation, we assume that the predicted price at t_i is exactly the same as the actual price at t_i. In other words, this hypothetical prediction model leads to a zero error rate. 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.

Table 4.3 shows a comparison in terms of expected returns between the historical approach and perfect foresight method. It is possible to observe an increase in the average level of around 14% following the use of perfect foresight approach with respect to the historical method. Under a financial perspective, an investor sees the portfolio profitability increasing under a perfect foresight situation. At the same time, the standard deviation of return values appears to decrease by about 25% with respect to the historical approach, which indicates a greater concentration of return values around the mean.

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.

Table 4.4 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 (1.37 and 5.26, respectively) are greater than those obtained from the historical approach (-0.22 and 3.38, respectively). 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.

Table 4.5 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.

Table 4.3: 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 4.4: 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 4.5: 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%

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.

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 optimization task is an offline approach, this duration is relatively fast and does not constitute a problem. Besides, speedups can be obtained by parallelizing 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 summarized 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.

Optimizing a portfolio directly in the test set can lead to better risk-adjusted performance results compared to when the optimization 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 optimization 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 optimize a mixed-asset portfolio including REITs by using ML algorithms to predict asset prices and a GA to optimize 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 optimization literature.

The chapter emphasizes the benefits of utilizing 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 Neighbors 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 optimization 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 optimization, 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 optimization 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 organized 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 summarizes 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 optimization. In the first step, the machine learning algorithms employed in this study undergo training on the training set, aiming to minimize the root mean squared error (RMSE) of predicted prices for various assets. Subsequently, these trained models are utilized 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 optimize the allocation of weights assigned to each asset. The performance metric used for this portfolio optimization task is the Sharpe ratio. The portfolio optimization 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 optimization 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.

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.

[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

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 analyze. 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 analyzed; 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

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.

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.

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 (XGBoost), Long/Short-Term Memory Neural Networks (LSTM), and k-Nearest Neighbors 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 to obtain lower error rates from the second technique. However, it is a suitable technique to evaluate the performance, which is meaningful in the context of portfolios that follow a short-term trading strategy that needs to be adjusted according to the current market conditions. Conversely, the first approach is more suitable as an evaluation strategy in the context of investors with long investment horizons, who might thus only to rebalance their portfolios periodically, or adjust their 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 algoritmic 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 utilize 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], 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

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

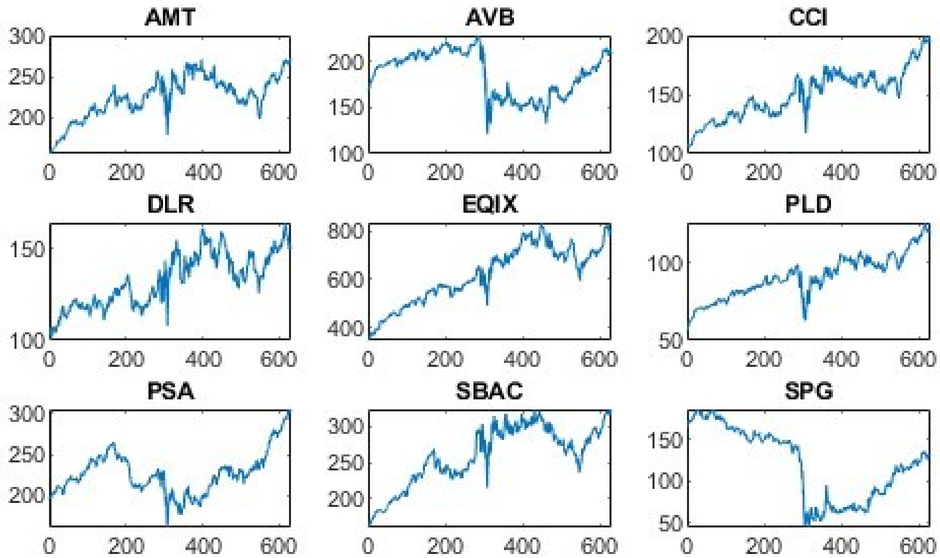

Figure 5.1: US REIT time series. The x-axis represents time in days; the y-axis refers to the price value in USD.

as a percentage of the latter. To be able to explain the prediction capability of our algorithms for the different asset classes throught 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.

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.

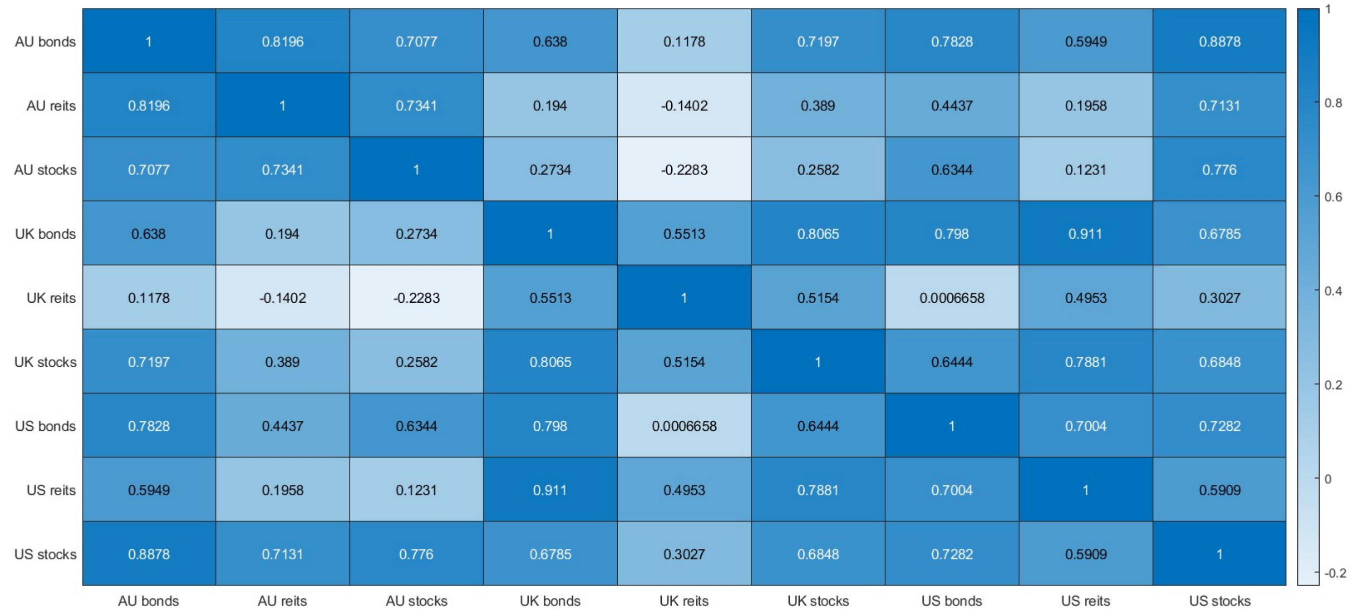

Figure 5.2: Correlation matrix between asset classes.

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
	Regularization 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 neighbors	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 optimization, 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

Holt's Linear Trend Method *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 modeled using a Box-Cox transformation and an exponential smoothing model. The seasonal components capture the periodic patterns in the time series and are modeled using a set of trigonometric functions. Finally, the autoregressive component captures the temporal dependencies in the time series and is modeled 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:

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

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 optimization benchmarks

Portfolio optimization 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 Optimizing 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 utilizing the algorithmic predictions to optimize 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 analyze the computational times associated with the utilized 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.

Table 5.6 presents the RMSE results for the three asset classes, over the 8 algorithms and the 5 different horizons, both for out-of-sample (left) and one-day-ahead (right) methods. With regards to the out-of-sample results, we can observe that all machine learning algorithms experience con-

Table 5.6: 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

siderably lower RMSE values than the econometric benchmarks (HLTM, TBATS, ARIMA), with improvements often being more than 50%. For instance, the RMSE average value observed for

SVR is 5.59 for a 30-day prediction period, in the case of out-of-sample predictions, which is lower than the value observed for HLTM which is 21.77. We also note similar improvements for the other periods: e.g., the 60-day SVR shows an average RMSE value of around 7.46 which is lower than the values observed for HLTM which is 16.87; the 90-day SVR reports a RMSE average value of 9.69 while HLTM shows a greater value of around 20.82; the 120-day XGBoost RMSE value of around 10.95 is lower than the ARIMA value of around 22.88; and finally, the 150-day LR value of around 8 is less than half the value observed for ARIMA of around 16.91. This is an important finding, which demonstrates the strengths of ML algorithms compared to the econometric approaches. Furthermore, we have also compared the RMSE distributions using other statistical properties, e.g., the standard deviation, skewness, and kurtosis. We can observe that the RMSE distributions are featured by less volatile values, a longer right tail and greater concentration around the mean in the case of ML algorithms in most cases. For instance, the 30-day SVR results show a standard deviation of 12.45, a skewness of 3.97, and a kurtosis of 18.38; at the same time, the 30-day HLTM is featured by a standard deviation of 40.15, a skewnss of 2.53, and a kurtosis of 6.94.

With regards to the one-day-ahead results, we can make similar observations: ML errors are again considerably lower than the benchmarks. E.g., the RMSE average values observed for ML algorithms tend to be around 1.02 for a 30-day prediction period, 2.40 for a 60-day period, 1.13 for a 90-day period; 1.14 for a 120-day period, and 1.15 for a 150-day period; on the other side, the average RMSE values observed for the benchmark algorithms tend to be around 6.47 for a 30-day period; 10.28 for a 60-day period, 9.30 for a 90-day period, 9.83 for a 120-day period, and 7.91 for a 150-day period. 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. Furthermore, by observing the other statistical properties of the RMSE distributions, we can observe that the volatility values tend to be lower for the ML algorithms indicating a greater concentration around the average RMSE value. E.g., the 30-day XGBoost SD value is around 2 which is at least 85% lower than the value observed for HLTM of about 14.23. On the other side, the skewness and kurtosis values tend to improve for ML algorithms as the time horizon increases. For example, the RMSE distributions tend to have a longer right tail and a greater concentration around the mean for a 150-day period compared to the benchmark methods. This indicates a lower likelihood of observing lower-than-the-mean values in the case of ML algorithms.

We can observe a similar picture for stock results (Table 5.7). ML algorithms show consistently lower values for RMSE average and SD for both out-of-sample and one-day-ahead predictions, across all periods (30, 60, 90, 120, and 150 days). Regarding the skewness and kurtosis values, the benchmark methods show better values for some prediction periods (e.g., 30 days), but this must be considered in relation to the above mentioned results for mean and SD. Generally speaking, the RMSE values observed for stocks tend to be higher than the values observed for REITs, which is related to the volatile nature of stocks that makes it harder to accurately predict stock prices.

Finally, Table 5.8 shows the RMSE distributions for bonds. It is worth noting that the values observed for the RMSE mean, SD, skewness, and kurtosis tend to be much lower compared to the values observed for the other two asset classes across all algorithms and methodologies. This is related to the lower risk profile of bonds which makes the prediction task easier to complete. Here, we can again observe that the use of ML algorithms leads to a reduction in the RMSE mean and SD for all the considered periods and methodologies, while the skewness and kurtosis values might be lower for the benchmark methods in some cases. However, the lower RMSE average and volatility values might indicate a better performance of the ML algorithms considered in the case of bond prediction.

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 5.9. For each algorithm, the table shows the average rank (first column), and the adjusted p-value of

Table 5.7: 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

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

Table 5.8: 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

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.

We can observe that the best (control) algorithm is KNN which statistically outperforms LSTM, LR, HLTM, TBATS, and ARIMA (given p-values equal to 1.86×10^{-5}; 6.83×10^{-8}; 5.76×10^{-24}; 5.76×10^{-24}; and 3.58×10^{-45}, respectively). The other algorithms (i.e., SVR and XGBoost) can be considered not to be statistically significantly different than KNN (given p-values equal to 5.62 and 0.76, respectively).

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 the Friedman test results, KNN is the best algorithm in predicting the prices of REITs, stocks and bonds both one-day-ahead and out-of-sample.

Table 5.9: 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**	$p_{\mathbf{Bonf}}$
KNN (c)	2.88	-
SVR	2.91	5.62
XGBoost	3.07	0.76
LSTM	3.42	1.86×10^{-5}
LR	3.54	6.83×10^{-8}
HLTM	6.46	5.76×10^{-24}
TBATS	6.46	5.76×10^{-24}
ARIMA	7.26	3.58×10^{-45}

5.4.2 Portfolio optimization

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 optimized portfolios per algorithm considered. For each generated portfolio, the optimized weights were used to calculate the expected return, expected risk, and expected Sharpe Ratio for the portfolio. These were pooled over all generated portfolios, to create and analyze the distributions of expected returns (Section 5.4.2), expected risks (Section 5.4.2), and expected Sharpe Ratios (Section 5.4.2) respectively. The following paragraphs provide a summary of key statistics for these metrics, namely the mean, standard deviation (or SD), skewness, and kurtosis. We compare the performance of our proposed approaches, i.e.i.e. of the five considered ML algorithms (LR, SVR, KNN, XGBOOST, and LSTM), to benchmarks, which consist of portfolios built using the historical data method, as well as HLTM, TBATS, and ARIMA.

Expected portfolio returns.

Table 5.10 shows descriptive statistics for expected return distributions obtained from the GA portfolio optimization for a 30-, 60-, 90-, 120-, and 150-day holding period. In the case of out-of-sample predictions on 30 days, the highest average return is achieved by SVR (1.33×10^{-3}), followed by KNN (1.30×10^{-3}) and LSTM (1.22×10^{-3}). In the case of one-day-ahead predictions for the same time period, the expected return is higher for all the algorithms, with the highest values achieved by SVR and LSTM (1.44×10^{-3}). For reference, the average return for the theoretical benchmark of perfect foresight is 1.76×10^{-3}. The ML algorithm portfolio return values are all higher than those obtained for the econometric benchmarks in the case of out-of-sample prediction (9.06×10^{-4} for HLTM; 1.93×10^{-4} for TBATS; 6.73×10^{-4} for ARIMA) and one-day-ahead prediction (9.62×10^{-4} for HLTM; 9.02×10^{-4} for TBATS; 1.25×10^{-3} for ARIMA). It is worth noting that all algorithms (except TBATS in the case of out-of-sample prediction) outperform the historical method, which showcases 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. As we observed, even the worse-performing ML algorithm (XGBoost) has more than doubled the returns obtained by the historical method. There's, of course, room for even greater improvements, given that the 'ceiling' of perfect foresight is around 188% higher than SVR and LSTM's average return of 1.44×10^{-3} (for the one-day-ahead method), showing that there are significant research potentials in this area.

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 5.13 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}).

Expected portfolio risks.

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. For instance, in the case of one-day-ahead predictions, the lowest risk average value is observed for XGBoost (around 1.57×10^{-3}), LSTM (around 2.86×10^{-3}), LSTM again (around 4.20×10^{-3}), XGBoost again (around 2.86×10^{-3}), and LR (around 5.02×10^{-3}) for a 30-, 60-, 90-, 120-, and 150-day prediction period respectively. 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. For instance, the lowest average risk value is observed for TBATS (around 1.21×10^{-3}) in the case of out-of-sample, 30-day ahead predictions. Nevertheless, it is worth noting that the

Table 5.10: 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). 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	9.06×10^{-4}	**1.78×10^{-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}	**−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	**1.44×10^{-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	**1.44×10^{-3}**	1.69×10^{-4}	0.22	1.09	**2.72×10^{-3}**	2.37×10^{-4}	**−2.37**	23.91
HistData	6.68×10^{-4}	1.56×10^{-5}	0.26	3.93	6.68×10^{-4}	**1.56×10^{-5}**	0.26	3.93
60 days								
HLTM	3.81×10^{-4}	**2.33×10^{-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	**2.12×10^{-3}**	2.16×10^{-4}	**−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	**1.58×10^{-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}	**−3.09**	13.13	7.00×10^{-4}	**1.46×10^{-5}**	**−3.09**	13.13
90 days								
HLTM	6.49×10^{-4}	**4.14×10^{-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	**1.91×10^{-3}**	1.73×10^{-4}	**−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	**1.91×10^{-3}**	1.77×10^{-4}	**−4.8**	**23.04**
KNN	**1.70×10^{-3}**	2.38×10^{-4}	**−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}	**−1.13**	0.94	1.73×10^{-3}	1.46×10^{-4}	**−3.42**	14.83
HistData	5.75×10^{-4}	3.88×10^{-5}	**−4.07**	20.32	5.75×10^{-4}	**3.88×10^{-5}**	-4.07	20.32
120 days								
HLTM	5.19×10^{-4}	**1.05×10^{-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	**1.49×10^{-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}	**−0.56**	12.53
LSTM	**1.15×10^{-3}**	2.72×10^{-4}	**−0.30**	1.25	1.43×10^{-3}	1.77×10^{-4}	**−0.29**	6.53
HistData	5.67×10^{-4}	2.73×10^{-5}	1.05	23.44	5.67×10^{-4}	**2.73×10^{-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	**1.27×10^{-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	**1.78×10^{-3}**	1.26×10^{-4}	-1.01	19.06
HistData	3.55×10^{-4}	**4.14×10^{-5}**	5.90	43.05	3.55×10^{-4}	**4.14×10^{-5}**	5.9	**43.05**

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 Table 5.11) 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.

Regarding the other statistical metrics, i.e., standard deviation, skewness and kurtosis, we can notice that the concentration of values around the mean tends to be higher for the benchmark algorithms than for the ML algorithms in the case of out-of-sample predictions. For instance, the HLTM algorithms has the lowest standard deviation value (around 6.24×10^{-6}) for a 60-day prediction period. The skewness and kurtosis values are higher for the benchmarks in most cases which indicates a greater concentration of risk values on the left side of the distribution.

With regards to risk, 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).

Expected portfolio Sharpe Ratios.

Lastly, when looking at the Sharpe ratio results of Figure 5.12, we can observe that all ML algorithms outperform the benchmarks for all periods in both cases of out-of-sample and one-day-ahead

[3]This becomes evident when we have a look at the Friedman ranking, which is presented in Table 5.13.

Table 5.11: 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). 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	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

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. Specifically, the highest Sharpe ratio value is observed for SVR (around 2.55×10^{-2}), XGBoost (around 2.01×10^{-2}), SVR (around 1.74×10^{-2}), LSTM (around 1.67×10^{-2}), and XGBoost (around 1.42×10^{-2}) for a 30-, 60-, 90-, 120-, and 150-day prediction period respectively. In the case of one-day-ahead predictions, we observe that the highest average value is obtained through SVR (around 3.43×10^{-2}), XGBoost (around 3.23×10^{-2}), SVR (around 2.80×10^{-2}), LR (around 2.36×10^{-2}), KNN, XGBoost, and LSTM (around 2.32×10^{-2}) for a 30-, 60-, 90-, 120-, and 150-day prediction period respectively. This is an important observation, because it demonstrates the importance of using machine learning for price predictions instead of traditional econometric approaches.

By looking at the other statistical metrics, we can notice that the standard deviation values tend to be lower for the HLTM algorithm for all periods considered in the case of out-of-sample predictions, and for HLTM, ARIMA and the historical based method in the case of one-day-ahead predictions. This indicates that the Sharpe ratio values tend to be less volatile for the benchmark algorithms. The skewness and kurtosis values show that the values tend to be more concentrated around the mean for the benchmark algorithms in most cases.

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.

Table 5.12: 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). 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	1.04×10^{-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

(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	1.40×10^{-4}
LR	4.06	2.14×10^{-18}
ARIMA	4.88	1.98×10^{-54}
HistData	7.35	3.25×10^{-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	4.51×10^{-5}
HLTM	4.89	6.80×10^{-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	9.94×10^{-10}
XGBoost	3.49	2.21×10^{-11}
ARIMA	7.02	1.05×10^{-279}
HLTM	7.05	2.84×10^{-283}
TBATS	7.61	0
HistData	8	0

Table 5.13: 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 the real world, these time differences are not considered significant. Besides, parallelization 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 Neighbors (KNN), Support Vector Regression (SVR), and Extreme Gradient Boosting (XGBoost) 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 utilization 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 favorable Sharpe ratio values compared to the other algorithms.

5.5 Summary

From the above results, we can summarize 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 5.3 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.

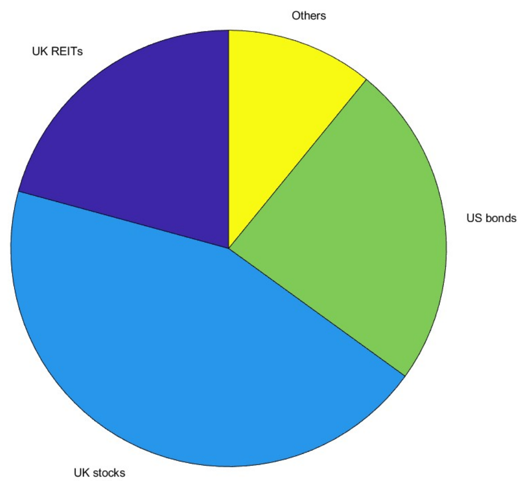

Figure 5.3: SVR-GA portfolio weights

The risk-adjusted performance of a portfolio obtained from ML predictions appear 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 optimization. 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 utilized 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 Neighbors, 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 summarizes 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 optimization, where the predicted prices from the above step are used as input to a portfolio, whose weights are optimized by means of a Genetic Algorithm. The

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

methodology that we follow in this chapter differs than 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).

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 timepoints 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.

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:

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

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

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]:

$$\text{MACD}(t) = \text{EMA}_R(t) - \text{EMA}_H(t) \tag{6.3}$$

[2]https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.ewm.html Last access: June 2023.

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:

$$\mathrm{BB}(t) = \mathrm{SMA}(t) \pm D\sqrt{\left(\frac{1}{T}\right)\sum_{i=t-(T-1)}^{t}\left[N_i - \mathrm{SMA}(t)\right]^2} \tag{6.4}$$

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.

$$\mathrm{Momentum} = N_t - N_{t-T} \tag{6.5}$$

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

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

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 optimization 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 optimization 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 optimization benchmarks

Portfolio optimization 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 optimized 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 Neighbors, 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 optimization 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 analyze 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 Performance

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.

Table 6.4 shows RMSE descriptive statistics for REITs in the case of out-of-sample and one-day-ahead prediction over a 30-, 60-, 90-, 120-, and 150-day period. 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 5.6 (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 9.70 to 5.94 (i.e. an error reduction of ≈39%), and the 120-day LSTM features a reduction from 10.99 to 5.87 (i.e. an error reduction of ≈ 46%). Furthermore, the ML algorithms using TAIs in their feature set also experience lower average standard deviations, when compared to the ML algorithms that do not use TA. In addition, it is worth noting that the performance of the conventional time-series benchmarks (HLTM, TBATS, ARIMA)[3] 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 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.

We can observe a similar picture for stocks results (Table 6.5). Machine learning algorithms that use TAIs in their feature set show consistently better results in terms of mean RMSE and standard deviation 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 both the means and standard deviations of the RMSE values observed here tend to be higher than the respective values in Table 6.4; this can

[3] 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 6.4, and across all remaining tables in this paper, are empty.

be explained by the more volatile nature of stock data, which makes it much harder to predict accurately.

Finally, Table 6.6 shows the RMSE distribution statistics for bonds. One noticeable difference here when compared to the previous two tables is the very low mean and standard deviation values observed across all algorithms and methodologies. This is due to the nature of bonds, which have very low volatility, and are thus 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.

In order to test 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 for all asset classes. The null hypothesis is that the compared RMSE distributions come from the same continuous distribution. Since we are making five comparisons (one for each considered period, i.e, 30-, 60-, 90-, 120-, and 150-day period), we adjust the α value according to the Bonferroni correction, i.e., $0.05/5 = 0.01$. In the case of out-of-sample predictions, we obtained KS test p-values < 0.001 in all cases, much lower than the adjusted α threshold of 0.01; specifically for each of the 30-, 60-, 90-, 120-, and 150-day periods, the obtained 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. This strongly suggests that the introduction of TAIs results in a clear reduction in the RMSE observed Conversely, in the case of one-day-ahead predictions, 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), suggesting that there is no significant difference in RMSE distributions. However, as mentioned earlier, due to the nature of one-day-ahead predictions, RMSE values tend to be very small, and thus much harder to achieve statistically significant results, despite the small reduction observed in the results.

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 optimization

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 optimized portfolios per algorithm considered. For each generated portfolio, the optimized 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 analyze the distributions of expected returns (Section 6.4.2), expected risks (Section 6.4.2), and expected Sharpe Ratios (Section 6.4.2) respectively. The following subsections provide a summary of key statistics for these metrics, namely the mean and standard deviation. 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.

Expected portfolio returns.

Table 6.7 shows descriptive statistics for expected return distributions obtained from the GA portfolio optimization for a 30-, 60-, 90-, 120-, and 150-day holding period. For a 30-day prediction period, we observe an increase in the average expected return obtained from out-of-sample predictions resulting from TAI models. On the other hand, the standard deviation values are not always lower for the proposed approaches. 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 and standard deviation of the expected return distributions also improve when introducing TAIs. For instance, the best result is observed for the KNN algorithm (3.78×10^{-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 can observe similar improvements brought by the use of the TAIs for the remaining periods (60, 90, 120, and 150 days) across both out-of-sample and one-day-ahead methods. Standard deviation results are more mixed, with the best performance alternating between the setups that use TAIs and the ones that do not.

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. 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.

In conclusion, the results obtained from the Genetic Algorithm confirm that the portfolios obtained using TAI-models can lead to improvements of up to 150% in the case of out-of-sample predictions, and of up to 230% in the case of one-day-ahead predictions. According to the KS test results, there is a statistically significant difference in the expected return distributions for the machine learning algorithms when introducing TAIs as additional features. The HLTM, TBATS, and ARIMA methods generally show lower expected return average values with respect to the proposed approaches. Our results show the importance of (i) using TAIs as additional features vs using lagged prices only; and (ii) using ML models vs traditional financial benchmarks.

Expected portfolio risks.

Table 6.8 shows descriptive statistics for the expected portfolio risks on a 30-, 60-, 90-, 120-, and 150-day testing period in the case of out-of-sample and one-day-ahead prediction. In both cases, the average expected risk tends to increase when including TAIs in the regression problem for all periods. 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. The standard deviation values are lower for algorithms that use TAIs in most of cases, which indicates a higher concentration of values around the mean. For instance, the average standard deviation is 5.18×10^{-4} when not using TAIs and 1.44×10^{-4} when using TAIs for a 30-day period. We observe similar values for the other periods. 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 and standard deviation values. For example, the average expected risk is 2.49×10^{-3} obtained from the TBATS algorithm for a 150-day period which is the lowest observed for the considered period. On the other hand, the lowest volatility of expected risk values is observed for HLTM at 6.94×10^{-18}.

In the case of one-day-ahead predictions, the average expected portfolio risk again tends to be lower when not using TAIs as features, while the standard deviation appears to be lower when using TAIs. In other words, the predictions obtained from algorithms that include TAIs lead to higher expected portfolio risk but at the same time, the risk values appear to be more concentrated around the mean. This might indicate a lower presence of outliers. For instance, the addition of TAIs as features results in an average expected risk increase from 3.54×10^{-3} to 1.11×10^{-2} for the KNN algorithm and 60-day prediction period, while its standard deviation appears to be reduced from 7.27×10^{-4} to 1.06×10^{-4}. In the case of a 30-day prediction period, the average expected risk appears to increase at an average rate of 532%, which drops to 219% for a 60-day period, to 145% for a 90-day period, increases to 218% for a 120-day period, and decreases to 90% for a 150-day period. Similarly to what is observed in the case of out-of-sample predictions, the average expected risk values tend to be lower for HLTM, TBATS, and ARIMA than for the algorithms that use TAIs for all periods.

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. Similarly to what we discussed for the expected return distributions, the null hypothesis was that the compared risk distributions came from the same continuous distribution. We again conducted five comparisons, one for each period, and thus, 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. 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.

To conclude, the Genetic Algorithm results show that the introduction of the TAIs in the feature set has led to statistically significant increases in mean risk across all algorithm and periods. This is of course a non-favourable result, but we should keep in mind that risk is just one of the metrics used to evaluate a portfolio's performance. In fact, it should not be considered in isolation, but in conjunction with returns, which as we have already seen are significantly higher when using TAIs. It is thus important to study an aggregate metric, such as the Sharpe Ratio, which adjusts returns for the level of risk taken. By incorporating the standard deviation of returns, it provides a measure of how much return an investment generates per unit of risk. This enables a fair comparison of different investments or portfolios, considering their risk profiles. We present the Sharpe Ratio results next.

Expected portfolio Sharpe Ratios.

In Table 6.9, we reported results for the expected portfolio Sharpe Ratio distributions on a 30-, 60, 90-, 120-, and 150-day testing period. We can observe that the proposed algorithms tend to outperform the benchmarks for all periods in the case of out-of-sample predictions. For instance, the highest average Sharpe Ratio value is observed for KNN (3.15×10^{-2}) for a 30-day period, for XGBoost (3.17×10^{-2}) for a 60-day period, for XGBoost again (2.56×10^{-2}) for a 90-day period, for LSTM (1.98×10^{-2}) for a 120-day period, and for LSTM again (2.24×10^{-2}) for a 150-day period. The standard deviation values appear to be lower for algorithms that use TAIs in some cases. For instance, the volatility of Sharpe Ratio distributions decreases from 2.69×10^{-3} to 1.95×10^{-3} for SVR for a 30-day period. 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. For instance, the average Sharpe Ratio value tends to range between 2.10×10^{-2} and 2.30×10^{-2} for benchmarks, and between 2.38×10^{-2} and 2.80×10^{-2} for the proposed 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 the 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 distributions. For one-day-ahead predictions, the KS test produced p-values of 0.359, 0.364, 0.824, 0.183, and 0.168 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.

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. Moreover, the use of TAIs as additional features for machine learning algorithms causes a statistically significant difference in the expected Sharpe distributions in the case of out-of-sample predictions, as shown by the KS test. Even though we did not observe a statistically significant difference in the Sharpe Ratio distributions when using TAIs in the case of one-day-ahead predictions, we observe improvements across all periods. These results confirm the importance of using TAIs as additional features.

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 analyze 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 partitionining 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[4].

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 6.1 presents the percentage SHAP (on the left side) and SAGE values (on the right side) calculated on the testing set for each

[4]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)

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% 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].

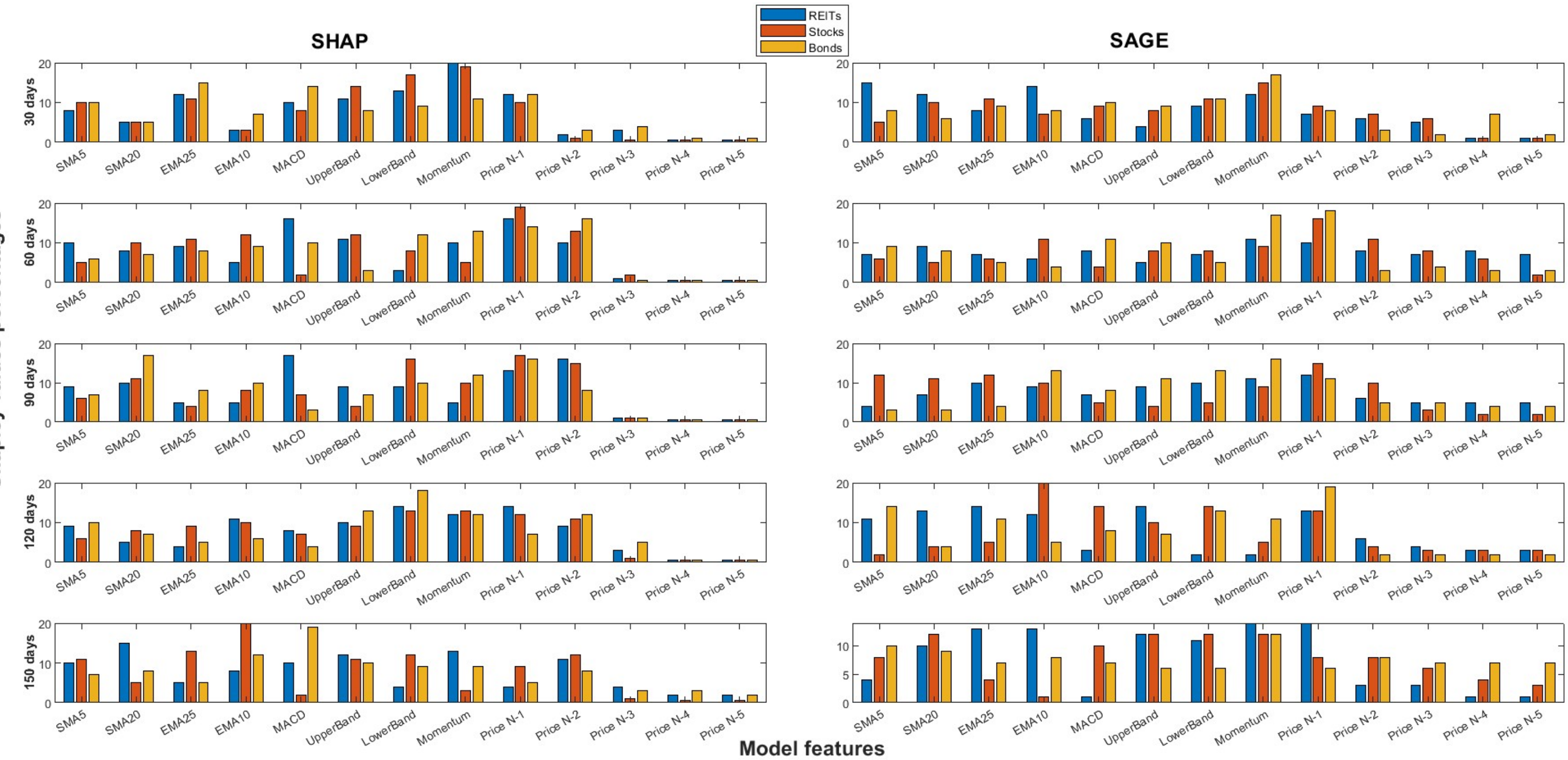

Figure 6.1: Shapley average value for each asset class and feature classified by period considered.

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 utilized in real-time applications.

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

Table 6.4: 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 6.5: 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 6.6: 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 6.7: 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	1.12E-03	**4.75E-06**	**3.07E-03**	2.16E-04	1.31E-03	3.52E-04	**3.51E-03**	**1.93E-04**
SVR	1.44E-03	3.95E-04	**3.40E-03**	**2.02E-04**	1.44E-03	4.92E-04	**3.55E-03**	**1.67E-04**
KNN	1.43E-03	**1.25E-05**	**3.46E-03**	2.05E-04	1.38E-03	2.95E-04	**3.78E-03**	**1.69E-04**
XGBOOST	1.23E-03	5.04E-04	**3.45E-03**	**2.06E-04**	1.47E-03	**1.50E-04**	**3.49E-03**	1.99E-04
LSTM	1.44E-03	**1.69E-04**	**3.36E-03**	1.86E-04	2.72E-03	2.37E-04	**3.70E-03**	**1.57E-04**
HLTM	9.06E-04	1.78E-06			9.62E-04	1.79E-04		
TBATS	1.93E-04	7.73E-05			9.02E-04	3.79E-04		
ARIMA	6.73E-04	2.85E-05			1.25E-03	4.35E-04		
60 days								
LR	8.40E-04	3.49E-04	**2.98E-03**	**2.13E-04**	1.86E-03	1.97E-04	**3.40E-03**	**1.45E-04**
SVR	1.52E-03	6.36E-04	**2.51E-03**	**2.34E-04**	1.48E-03	**1.90E-04**	**3.13E-03**	2.16E-04
KNN	1.02E-03	**9.18E-05**	**2.90E-03**	1.66E-04	1.75E-03	2.17E-04	**3.49E-03**	**1.72E-04**
XGBOOST	1.58E-03	6.06E-04	**3.46E-03**	**1.86E-04**	2.07E-03	2.28E-04	**3.66E-03**	**1.76E-04**
LSTM	1.26E-03	**4.09E-05**	**2.40E-03**	2.45E-04	1.45E-03	4.50E-04	**3.62E-03**	**1.80E-04**
HLTM	3.81E-04	2.33E-06			6.90E-04	1.55E-04		
TBATS	2.40E-04	2.78E-05			2.84E-04	1.34E-04		
ARIMA	6.72E-04	7.48E-05			2.12E-03	2.16E-04		
90 days								
LR	8.21E-04	**2.08E-04**	**2.43E-03**	2.35E-04	1.74E-03	**2.06E-04**	**2.79E-03**	2.37E-04
SVR	1.35E-03	4.26E-04	**2.64E-03**	**2.58E-04**	1.91E-03	**1.77E-04**	**3.48E-03**	1.85E-04
KNN	1.70E-03	**2.38E-04**	**2.44E-03**	2.60E-04	1.85E-03	2.68E-04	**3.03E-03**	**2.07E-04**
XGBOOST	1.42E-03	2.89E-04	**2.85E-03**	**2.31E-04**	1.71E-03	**1.91E-04**	**3.00E-03**	1.99E-04
LSTM	1.40E-03	4.99E-04	**2.32E-03**	**2.90E-04**	1.73E-03	**1.46E-04**	**3.06E-03**	2.03E-04
HLTM	6.49E-04	4.14E-06			9.84E-04	1.73E-04		
TBATS	1.70E-04	1.39E-05			9.62E-04	1.48E-04		
ARIMA	3.92E-04	6.81E-05			1.91E-03	1.73E-04		
120 days								
LR	1.14E-03	4.15E-04	**2.21E-03**	**2.73E-04**	1.49E-03	**1.76E-04**	**2.72E-03**	2.33E-04
SVR	1.14E-03	2.26E-04	**2.08E-03**	**2.47E-04**	1.42E-03	6.14E-04	**3.75E-03**	**1.99E-04**
KNN	1.12E-03	3.50E-04	**1.91E-03**	1.97E-04	1.32E-03	**1.15E-04**	**2.92E-03**	1.77E-04
XGBOOST	1.11E-03	2.79E-04	**2.13E-03**	2.55E-04	1.22E-03	**1.93E-04**	**2.57E-03**	1.97E-04
LSTM	1.15E-03	2.72E-04	**2.09E-03**	**1.98E-04**	1.43E-03	**1.77E-04**	**2.76E-03**	2.26E-04
HLTM	5.19E-04	1.05E-18			5.48E-04	9.70E-05		
TBATS	1.85E-04	1.04E-05			3.75E-04	1.39E-04		
ARIMA	3.21E-04	2.92E-05			8.56E-04	8.18E-05		
150 days								
LR	9.53E-04	**5.63E-05**	**1.99E-03**	2.43E-04	1.51E-03	**9.19E-05**	**3.01E-03**	1.87E-04
SVR	1.17E-03	3.15E-04	**2.22E-03**	**2.09E-04**	1.75E-03	**1.77E-04**	**2.61E-03**	2.29E-04
KNN	9.31E-04	**4.53E-05**	**2.26E-03**	2.15E-04	1.76E-03	**1.34E-04**	**2.98E-03**	2.20E-04
XGBOOST	1.27E-03	**2.31E-04**	**2.45E-03**	2.37E-04	1.76E-03	**1.14E-04**	**3.01E-03**	2.39E-04
LSTM	1.03E-03	3.38E-04	**2.41E-03**	**1.94E-04**	1.78E-03	**1.26E-04**	**2.73E-03**	1.72E-04
HLTM	1.13E-04	1.06E-04			1.40E-03	8.89E-05		
TBATS	1.11E-04	1.05E-04			1.38E-03	1.15E-04		
ARIMA	6.94E-04	1.09E-04			1.65E-03	1.33E-04		

Table 6.8: 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	**1.86E-03**	**3.44E-05**	1.07E-02	1.85E-04	**1.80E-03**	6.41E-04	1.12E-02	**1.31E-04**
SVR	**6.16E-03**	1.70E-03	1.11E-02	**1.40E-04**	**1.79E-03**	9.90E-04	1.11E-02	**1.27E-04**
KNN	**3.13E-03**	**7.79E-05**	1.12E-02	1.17E-04	**1.90E-03**	1.12E-03	1.13E-02	**1.28E-04**
XGBOOST	**4.61E-03**	9.17E-04	1.12E-02	**1.31E-04**	**1.57E-03**	8.05E-04	1.12E-02	**1.32E-04**
LSTM	3.39E-03	1.06E-03	**1.09E-02**	**1.48E-04**	**1.85E-03**	8.39E-04	1.13E-02	**1.13E-04**
HLTM	8.24E-03	3.41E-05			2.66E-03	6.95E-04		
TBATS	1.21E-03	2.75E-04			2.76E-03	1.26E-03		
ARIMA	3.91E-03	4.27E-05			8.29E-03	5.99E-04		
60 days								
LR	**4.03E-03**	3.89E-03	1.07E-02	**1.65E-04**	**4.04E-03**	1.38E-03	1.09E-02	**1.18E-04**
SVR	**6.77E-03**	2.05E-03	1.04E-02	**1.73E-04**	**3.10E-03**	7.58E-04	1.08E-02	**1.64E-04**
KNN	**3.84E-03**	4.49E-04	1.04E-02	**1.47E-04**	**3.54E-03**	7.27E-04	1.11E-02	**1.06E-04**
XGBOOST	**6.00E-03**	1.95E-03	1.10E-02	**1.46E-04**	**4.13E-03**	8.27E-04	1.12E-02	**1.30E-04**
LSTM	**5.37E-03**	2.24E-04	1.02E-02	**1.93E-04**	**2.86E-03**	9.96E-04	1.12E-02	**1.30E-04**
HLTM	4.51E-03	6.24E-06			4.96E-03	5.93E-04		
TBATS	3.89E-03	7.39E-05			5.17E-03	3.52E-04		
ARIMA	5.05E-03	2.37E-04			1.38E-02	1.17E-03		
90 days								
LR	**3.71E-03**	3.81E-04	1.04E-02	**1.54E-04**	**4.29E-03**	7.32E-04	1.06E-02	**1.72E-04**
SVR	**5.78E-03**	1.15E-03	1.07E-02	**1.81E-04**	**4.57E-03**	5.03E-04	1.11E-02	**1.35E-04**
KNN	**1.00E-02**	1.55E-03	1.07E-02	**1.61E-04**	**4.82E-03**	1.09E-03	1.08E-02	**1.42E-04**
XGBOOST	**8.02E-03**	1.97E-03	1.05E-02	**2.14E-04**	**4.22E-03**	1.37E-03	1.07E-02	**1.50E-04**
LSTM	**6.98E-03**	**1.55E-03**	1.07E-02	1.71E-04	**4.20E-03**	5.59E-04	1.07E-02	**1.49E-04**
HLTM	5.89E-03	2.41E-05			5.92E-03	1.09E-03		
TBATS	2.73E-03	4.62E-05			5.75E-03	5.45E-04		
ARIMA	5.73E-03	2.92E-04			1.94E-02	1.65E-03		
120 days								
LR	**7.68E-03**	1.39E-03	1.04E-02	**1.94E-04**	**3.89E-03**	6.04E-04	1.05E-02	**1.74E-04**
SVR	**7.32E-03**	1.81E-03	1.02E-02	**1.65E-04**	**2.99E-03**	9.70E-04	1.00E-02	**1.55E-04**
KNN	**7.05E-03**	4.35E-04	9.65E-03	**1.57E-04**	**3.02E-03**	1.33E-03	1.05E-02	**1.44E-04**
XGBOOST	**6.52E-03**	9.75E-04	1.00E-02	**2.19E-04**	**2.86E-03**	9.66E-04	1.02E-02	**1.76E-04**
LSTM	**6.32E-03**	9.92E-04	9.91E-03	**1.41E-04**	**3.75E-03**	6.15E-04	1.06E-02	**1.70E-04**
HLTM	3.93E-03	6.94E-18			4.96E-03	8.69E-04		
TBATS	2.92E-03	5.79E-05			7.05E-03	4.32E-04		
ARIMA	5.42E-03	1.59E-04			2.03E-02	1.75E-03		
150 days								
LR	**5.49E-03**	2.68E-04	9.92E-03	**1.97E-04**	**5.02E-03**	6.39E-04	1.06E-02	**1.44E-04**
SVR	**6.81E-03**	1.20E-03	9.97E-03	**1.67E-04**	**5.72E-03**	5.65E-04	1.04E-02	**1.79E-04**
KNN	**5.04E-03**	2.06E-04	9.99E-03	**1.78E-04**	**5.66E-03**	6.71E-04	1.06E-02	**1.80E-04**
XGBOOST	**7.75E-03**	**1.34E-03**	1.03E-02	1.81E-04	**5.66E-03**	4.83E-04	1.08E-02	**1.73E-04**
LSTM	**5.07E-03**	1.29E-03	1.01E-02	**1.56E-04**	**5.89E-03**	2.16E-03	1.04E-02	**1.11E-04**
HLTM	3.93E-03	6.94E-18			9.34E-03	8.19E-04		
TBATS	2.49E-03	2.15E-04			9.26E-03	6.46E-04		
ARIMA	4.67E-03	1.88E-04			3.07E-02	3.39E-03		

Table 6.9: 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	1.88E-02	**1.86E-04**	**2.78E-02**	2.15E-03	3.12E-02	7.08E-03	**3.21E-02**	**1.86E-03**
SVR	2.55E-02	2.69E-03	**3.10E-02**	**1.95E-03**	**3.43E-02**	6.00E-03	3.28E-02	**1.58E-03**
KNN	2.49E-02	**1.06E-04**	**3.15E-02**	1.96E-03	3.29E-02	6.30E-03	**3.45E-02**	**1.66E-03**
XGBOOST	2.23E-02	9.52E-03	**3.14E-02**	**1.99E-03**	3.17E-02	4.36E-03	**3.19E-02**	**1.91E-03**
LSTM	2.29E-02	**1.72E-03**	**3.08E-02**	1.78E-03	**3.42E-02**	5.65E-03	3.40E-02	**1.50E-03**
HLTM	1.04E-02	3.60E-06			1.75E-02	3.05E-03		
TBATS	4.91E-03	1.12E-03			1.72E-02	4.72E-03		
ARIMA	1.05E-02	4.22E-04			1.35E-02	4.67E-03		
60 days								
LR	1.35E-02	3.37E-03	**2.72E-02**	**2.09E-03**	2.96E-02	3.66E-03	**3.18E-02**	**1.41E-03**
SVR	1.81E-02	5.12E-03	**2.29E-02**	**2.30E-03**	2.66E-02	3.68E-03	**2.84E-02**	**2.13E-03**
KNN	1.62E-02	**5.44E-04**	**2.72E-02**	1.64E-03	2.96E-02	4.07E-03	**3.23E-02**	**1.67E-03**
XGBOOST	2.01E-02	5.06E-03	**3.17E-02**	**1.79E-03**	3.23E-02	4.21E-03	**3.34E-02**	**1.67E-03**
LSTM	1.69E-02	**4.21E-04**	**2.16E-02**	2.43E-03	2.69E-02	4.96E-03	**3.31E-02**	**1.74E-03**
HLTM	5.11E-03	3.84E-05			9.99E-03	1.93E-03		
TBATS	3.54E-03	4.58E-04			3.67E-03	1.65E-03		
ARIMA	9.17E-03	8.21E-04			1.79E-02	1.54E-03		
90 days								
LR	1.31E-02	2.66E-03	**2.23E-02**	**2.29E-03**	**2.64E-02**	3.10E-03	2.53E-02	**2.33E-03**
SVR	1.74E-02	4.71E-03	**2.40E-02**	**2.46E-03**	2.80E-02	2.64E-03	**3.20E-02**	**1.75E-03**
KNN	1.67E-02	**1.17E-03**	**2.21E-02**	2.48E-03	2.69E-02	4.65E-03	**2.79E-02**	**2.01E-03**
XGBOOST	1.57E-02	**1.98E-03**	**2.56E-02**	2.27E-03	2.63E-02	**1.80E-03**	**2.77E-02**	1.92E-03
LSTM	1.62E-02	5.58E-03	**2.06E-02**	**2.82E-03**	**2.65E-02**	2.40E-03	2.82E-02	**1.98E-03**
HLTM	8.20E-03	3.81E-03			1.26E-02	2.05E-03		
TBATS	2.89E-03	2.54E-04			1.25E-02	1.85E-03		
ARIMA	4.91E-03	7.47E-04			1.36E-02	1.12E-03		
120 days								
LR	1.26E-02	3.75E-03	**1.97E-02**	**2.29E-03**	**2.36E-02**	**1.95E-03**	2.47E-02	2.31E-03
SVR	1.60E-02	5.80E-03	**1.89E-02**	**2.46E-03**	2.07E-02	2.45E-03	**2.17E-02**	**2.01E-03**
KNN	1.55E-02	**1.12E-03**	**1.79E-02**	2.48E-03	2.02E-02	3.86E-03	**2.74E-02**	**1.73E-03**
XGBOOST	1.49E-02	**1.88E-03**	**1.90E-02**	2.27E-03	2.04E-02	2.67E-03	**2.39E-02**	**1.95E-03**
LSTM	1.67E-02	**2.40E-03**	**1.98E-02**	2.82E-03	2.31E-02	**2.01E-03**	**2.52E-02**	2.17E-03
HLTM	7.55E-03	1.42E-17			7.98E-03	1.26E-03		
TBATS	3.07E-03	2.14E-04			4.23E-03	1.50E-03		
ARIMA	4.10E-03	3.37E-04			5.90E-03	7.67E-04		
150 days								
LR	1.26E-02	**4.93E-04**	**1.78E-02**	2.39E-03	2.12E-02	**1.39E-03**	**2.80E-02**	1.84E-03
SVR	1.38E-02	2.79E-03	**2.22E-02**	**2.09E-04**	2.30E-02	2.52E-03	**2.38E-02**	**2.22E-03**
KNN	1.28E-02	**3.84E-04**	**2.05E-02**	2.05E-03	2.32E-02	**1.65E-03**	**2.71E-02**	2.15E-03
XGBOOST	1.42E-02	**1.70E-03**	**2.23E-02**	2.36E-03	2.32E-02	**1.73E-03**	**2.72E-02**	2.33E-03
LSTM	1.40E-02	3.04E-03	**2.25E-02**	**1.92E-03**	2.32E-02	**1.63E-03**	**2.58E-02**	1.73E-03
HLTM	7.76E-03	1.42E-17			1.43E-02	7.98E-04		
TBATS	3.38E-03	1.97E-04			1.41E-02	8.73E-04		
ARIMA	3.88E-03	1.03E-03			9.32E-03	5.83E-04		

Chapter 7

Optimizing 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 experiements 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 organized 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 optimize our portfolios. Finally, Section 7.4 summarizes 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 optimize 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 optimization 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	**19.44**	-46.43%	5.41E-04	**8.99E-04**	66.06%
Std Dev	146.15	**71.93**	-50.79%	6.08E-05	**2.79E-05**	-54.11%
	Expected risk			**Sharpe ratio**		
Metric	**Without REITs**	**With REITs**	**% Difference**	**Without REITs**	**With REITs**	**% Difference**
Mean	5.54E-03	**3.70E-03**	-33.21%	7.26E-03	**1.48E-02**	103.71%
Std Dev	4.04E-04	**1.48E-04**	-63.28%	**5.33E-04**	5.55E-04	4.17%

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 analyze 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.

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

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. According to the test results, the adjusted p-value is equal to 1.94E-45, which indicates a statistically significant difference in the two distributions.

In summary, when analyzing 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 optimization

After having analyzed 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 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's correction (e.g., $0.05/3 = 0.0167$). According to the test results, the adjusted p-value is equal to 1.55E-45 for all the considered metrics, which indicates a statistically significant difference in the compared distributions.

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 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 optimization. Specifically, we used five machine learning algorithms – i.e., Ordinary Least Squares Linear Regression, Support Vector Regression, k-Nearest Neighbors 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 optimize 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 optimizing 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 optimize 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 than prices in the testing set, thus worsening the overall portfolio performance. In fact, a good portfolio optimization strategy mainly depends on the accuracy of the price predictions. Therefore, in Chapter 4 we evaluate the potential advantage of using price predictions in the portfolio optimization task. To simplify this analysis, we assume perfect price predictions in the test set, according to a perfect foresight approach. In that way, the optimization 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 optimize 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 optimizing a portfolio directly in the test set leads to superior risk-adjusted performance compared to optimization uniquely within the training set. This insight has motivated us to engage into the task of price prediction in the following chapter, recognizing the importance of predictive analysis in the investment portfolio optimization. Our focus on predictive modeling aims to refine investment strategies, contributing to the field on portfolio optimization 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 optimization 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 Neighbors 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 to obtain a more effective portfolio optimization 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 optimize mixed-asset portfolios, incorporating Real Estate Investment Trusts (REITs), predominantly relied on historical data for weight optimization. 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 utilized 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 period) 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 optimize 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 optimization 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, emphasizing 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 optimization 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 analyzing the integration of ML-based price predictions for REITs, stocks, and bonds in portfolio optimization. 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 utilized 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 optimizing mixed-asset portfolios that include real estate. The main contribution lies in the incorporation of TAIs as features in the prediction process represents a novel and underexplored aspect in predicting REIT prices. The research extends its impact by utilizing the price predictions in a portfolio context, demonstrating the positive effects of TAIs in portfolio optimization 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 analyzes 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 centered on predicting out-of-sample and one-day-ahead prices of REITs, stocks, and bonds, utilizing 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 an 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 emphasizes 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) optimizing with price predictions – rather than historical data – enhances mixed-asset portfolio performance; and (iii) 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, emphasizing the significance of price predictions enhanced by TAIs. The main motivation behind such 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 optimizing the weights of the considered asset classes. While traditional approaches often rely on simpler optimization techniques or manual allocation strategies, the utilization 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 optimization, 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 semi-parametric 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).