Expert Systems With Applications 235 (2024) 121102

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

An in-depth investigation of five machine learning algorithms for optimizing mixed-asset portfolios including REITs

Fatim Z. Habbab *, Michael Kampouridis

Centre for Computational Finance and Economics Agents, School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, United Kingdom

ARTICLE INFO

Keywords:
Machine learning
REITs
Financial time series
ARIMA

ABSTRACT

Real estate is a favored investment option as it allows investors to diversify their portfolios and minimize risk. Investors can invest in real estate directly by purchasing a property, or through real estate investment funds (REITs) where they can purchase shares in companies that own and manage real estate. Investing in REITs has become increasingly popular because it eliminates some of the disadvantages associated with direct real estate investment, such as the need for a large upfront payment. When investing in mixed asset portfolios, it is crucial to predict future prices accurately to ensure profitable and less risky asset allocation. However, literature on price prediction often focuses on only one or two algorithms, and there is no research that explores REITs' price prediction in the context of portfolio optimization. To address this gap, we conducted a thorough evaluation of 5 machine learning algorithms (ML), including Ordinary Least Squares Linear Regression (LR), Support Vector Regression (SVR), k-Nearest Neighbors Regression (KNN), Extreme Gradient Boosting (XGBoost), and Long/Short-Term Memory Neural Networks (LSTM), as well as other financial benchmarks like Holt's Exponential Smoothing (HES), Trigonometric Seasonality, Box–Cox Transformation, ARMA Errors, Trend, and Seasonal Components (TBATS), and Auto-Regression Integrated Moving Average (ARIMA). We applied these algorithms to predict future prices for 30 REITs from the US, UK, and Australia, as well as 30 stocks and 30 bonds. The assets were then used as part of a portfolio, which we optimized using a genetic algorithm. Our results showed that using ML algorithms for price prediction provided at least three times the return over benchmark models and reduced risk by almost two-fold. For REITs, we observed that the use of ML algorithms led to a higher allocation to REITs diversified by country. In particular, our results showed that SVR was the best-performing algorithm in terms of risk-adjusted returns across different time horizons, as confirmed by our Friedman test results (Sharpe ratio). Overall, our study highlights the effectiveness of ML algorithms in predicting asset prices and optimizing portfolio allocation.

1. Introduction

To optimize a portfolio, the ideal weights of investments must be determined to reduce risk and/or increase returns (Brabazon, Kampouridis, & O'Neill, 2020). One way to decrease investment risk is by investing in real estate (Akinsomi, 2020; Jain, 2017; Jayaraman, 2021). Institutional investors have found that a significant allocation to real estate protects their wealth during difficult times, such as the Covid-19 pandemic (Akinsomi, 2020). However, direct investment in real estate assets can be expensive, so many investors choose indirect investment through real estate investment trusts (REITs), which are companies that own and manage real estate. REITs offer individual investors the opportunity to invest in real estate without the hassle of owning or managing properties. The low entry cost of REITs makes them an attractive option, with shares available for as little as $500.[1] Additionally, REITs are highly liquid, like stocks, making them easier to buy and sell quickly compared to real estate properties that can take months to complete.

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 (Anderson, Anderson, Guirguis, Proppe, & Seiler, 2021). 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

* Corresponding author.
E-mail addresses: fh20175@essex.ac.uk (F.Z. Habbab), mkampo@essex.ac.uk (M. Kampouridis).

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

https://doi.org/10.1016/j.eswa.2023.121102
Received 16 April 2023; Received in revised form 30 July 2023; Accepted 31 July 2023
Available online 16 August 2023

portfolio has been associated with potential improvements in risk-adjusted returns (Marzuki & Newell, 2019). Several studies (Günther, Wills, & Piazolo, 2022; Pilusa, Niesing, & Zulch, 2022) 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 (Pilusa et al., 2022; Razak, 2023).

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, as noted by Habbab and Kampouridis (2022b).

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 Delfim and Hoesli (2019), Geiger, Cajias, and Fuerst (2016) and Wiklund, Flood, and Lunde (2020). Our study, on the other hand, concentrates on the first step of this approach, which is to predict REIT prices accurately. This step is crucial since the prices are utilized as input in the portfolio optimization step.

Although a few researchers have made attempts to predict REIT prices using machine learning algorithms, the number of such studies remains limited. For example, Li, Fong, and Chong (2017) 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, Chen, Chang, Ho, and Diaz (2014) 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 Jain, Mandal, Singh, Kulkarni, and Sayed (2021), Lian, Li, and Wei (2021) and Loo (2019). 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 Jain et al. (2021), Lian et al. (2021) and Loo (2019). 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.

Prior research has indicated that machine learning algorithms can outperform traditional models such as autoregressive integrated moving average (ARIMA) when it comes to predicting REIT prices, as noted by Habbab and Kampouridis (2022a), Jain et al. (2021), Lian et al. (2021) and Loo (2019). Nevertheless, most of these studies have concentrated on using artificial neural networks with multiple variables.

Our study seeks to overcome the existing limitations by exploring five different machine learning algorithms for REIT price prediction. Previous research in the context of portfolio optimization has generally used only one or two machine learning algorithms to predict prices of non-REIT assets (Butler & Kwon, 2021; Chen, Zhang, Mehlawat, & Jia, 2021; Freitas, De Souza, & De Almeida, 2009; Ma, Han, & Wang, 2020, 2021; Pawar, Jalem, & Tiwari, 2019; Sen, Dutta, & Mehtab, 2021; Sen et al., 2021). The five algorithms used in our study are Ordinary Least Squares Linear Regression (LR), Support Vector Regression (SVR), k-Nearest Neighbors Regression (KNN), Extreme Gradient Boosting (XGBoost), and Long/Short-Term Memory Neural Networks (LSTM). We will compare the prediction errors generated by these algorithms with commonly used financial benchmarks such as Holt's Exponential Smoothing, Trigonometric seasonality, Box-cox transformation, ARMA errors, Trends and Seasonal components, and ARIMA. Our objective is to demonstrate that machine learning techniques can provide accurate REIT price predictions and outperform popular financial time series benchmarks.

Once we have obtained price predictions, we use a genetic algorithm (GA) to optimize the portfolio (Goldberg, 1989; Habbab & Kampouridis, 2022b; Kalayci, Polat, & Akbay, 2020; Yaman & Dalkılıç, 2021). GA has been widely used for portfolio optimization, and it has proven to perform better than the Global Minimum Variance (GMV) portfolio, which is one of the most advanced methods available (Habbab, Kampouridis, & Voudouris, 2022). Moreover, Li and Wu (2021) 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, Adebiyi, Ogunbiyi, and Amole (2022) 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 (Chen & Smith, 2019).

In this study, we adopt a GA to find the optimal weights for a given set of assets based on the return and risk parameters derived from the Modern Portfolio Theory (MPT) concepts. This will be explained in more detail in Section 3.4. Our goal is to demonstrate that using ML price predictions results in better portfolio performance. We evaluate financial metrics such as Sharpe ratio, returns, and risk and compare the results with two benchmarks. The first benchmark is a portfolio optimized on the training set, which is a standard approach in financial literature (Jones & Trevillion, 2022; Lee & Moss, 2018; Parikh & Zhang, 2019). However, weights calculated using the training set might result in inferior portfolio performance if there are significant differences between prices in the training and test sets. Therefore, we aim to evaluate the added value of accurate price predictions in portfolio allocation. The second benchmark is a theoretical portfolio that assumes perfect price predictions in the test set (referred to as *perfect foresight*), which we use to compare portfolio performance with ML price predictions.

In conclusion, our study assesses the performance of REITs in mixed-asset portfolios and examines the differences among the US, UK, and Australian markets. Prior research on the role of real estate in mixed-asset portfolios has mainly focused on real estate investments within a single country. In contrast, we make a novel contribution to the literature by discussing the similarities and differences between these three markets. We investigate the correlation between stocks, bonds, and REITs in each market's portfolio and discuss the advantages of managing international portfolios.

In summary, this study contributes in three main ways: (i) it conducts a comprehensive exploration of price prediction for REITs and other asset classes in portfolios using five different machine learning techniques and three financial benchmarks; (ii) it optimizes the weights of a mixed-asset international portfolio that includes REITs using the aforementioned price predictions; and (iii) it provides a detailed analysis and discussion of the performance of REITs within mixed asset portfolios.

The rest of this paper is organized as follows. Section 2 provides a brief overview of REITs and the Modern Portfolio Theory. Section 3 explains the methodology used in this study. Our experimental setup is presented in Section 4. The results of our experiments are presented in Section 5, where we provide a detailed discussion of the results obtained by applying machine learning and other financial models to our data. Finally, Section 6 summarizes the conclusions of the study and offers suggestions for future research.

2. Background

2.1. The real estate asset class

Real estate is one of the available investment options, along with publicly listed stocks, bonds, and cash. Investing in real estate provides opportunities for diversification by utilizing correlations between real estate and other asset classes (Baker & Chinloy, 2014).

To gain exposure to the real estate market, an investor can either invest directly in a property or indirectly by purchasing shares in companies that hold real estate investments in their portfolios. Direct investment in real estate allows control and management of the property, but requires a significant initial investment and may be difficult to sell due to the low liquidity in real estate markets. Indirect investment in real estate offers lower unit prices and greater liquidity, but may be harder to manage and control the property. By investing in listed real estate, investors can enjoy the benefits of diversification without the need for a significant financial commitment or the responsibility of property management and control.

The remainder of this section is structured as follows. Section 2.1.1 outlines the current real estate sectors, while Section 2.1.2 furnishes details about a specific kind of publicly traded real estate investments called Real Estate Investment Trusts (REITs), which is the focal point of this article.

2.1.1. Real estate markets

The real estate market is diverse and can be categorized into several types of products, including (a) residential real estate, which includes any property used for residential purposes; (b) commercial real estate, which refers to any property used for business purposes; (c) industrial real estate, which includes any property used for manufacturing, storage, and distribution of goods; and (d) raw land.

The residential real estate sector comprises various types of houses, such as standalone houses that host one family or multi-family houses that accommodate several families. It also includes townhouses, which are individually-owned dwellings, and condominiums, privately-owned properties located together with other units. The residential real estate sector is essential for the well-being of any economy since it addresses the fundamental need of people to find a home. Investors typically invest in this sector to obtain a stable form of income through rent.

The commercial real estate sector is made of properties used for business purposes, such as shopping malls, hotels, and offices. It also includes flat buildings used for business. Investing in commercial real estate can be profitable since it offers the opportunity for capital appreciation, as the value of a commercial property tends to increase over time due to inflation. This makes it possible to sell the property at a higher price in the future.

On the other hand, industrial real estate includes properties used for producing, storing, and transporting goods and services, such as warehouses and farms. This sector plays a critical role in the economy by maintaining the supply chain and improving the efficiency of distribution. High-quality industrial real estate that is well-located can ensure the efficient movement of goods from producers to markets, which satisfies the needs of both customers and producers.

2.1.2. Real estate investment trusts

REITs are entities that manage, fund, or possess income-producing real estate assets. Well-known REITs are Realty Income Corporation (O), Digital Realty Trust, Inc (DLR), and Simon Property Group, Inc (SPG), among others. Through investing in REITs, regular investors can participate in real estate investments and reap the benefits of competitive returns and dividend-based income without the large capital expenditure that direct real estate investment requires (Block, 2011).

Investing in REITs is similar to investing in other financial markets, and there are various ways investors can do so. Some options include purchasing individual company stocks, mutual funds, or exchange-traded funds (ETFs). To identify suitable REIT investments, investors may consult with a broker, financial advisor, or planner to establish their financial objectives. A 2020 study conducted in the US by Chatham Partners[2] showed that approximately 80% of financial advisors recommend REITs to their clients. Additionally, investors can consider investing in private REITs or public non-listed REITs.

The ownership of some properties is transferred to investors who hold shares in REITs, allowing them to earn a share of the income generated without needing to purchase, manage, or finance the property. Optimal portfolio allocation for REITs has been studied extensively, with studies such as those conducted by Bhuyan, Kuhle, Ikromov, and Chiemeke (2014), Hocht, Ng, Wolf, and Zagst (2008), and Jalil, Ali, Razali, and Yim (2015) suggesting that REIT investment should typically make up between 5% and 15% of an investment portfolio. This weighting may vary depending on the investment horizon, with research by Rehring (2012) and Stephen and Simon (2005) highlighting that the diversification potential of REITs increases over longer holding periods.

REITs typically invest in a variety of real estate properties, including but not limited to, offices, apartments, warehouses, retail centers, medical facilities, data centers, cell towers, infrastructures, and hotels. While some REITs focus on a particular type of property, others may have portfolios that comprise multiple property types.

REITs primarily generate income by leasing properties and receiving rent payments, which are then distributed to shareholders in the form of dividends. In the US, REITs are required to pay at least 90% of their taxable income to shareholders, who are then responsible for paying taxes on those dividends.

Investors find REITs an appealing investment choice because of their competitive returns, which come from a mix of steady income and long-term capital appreciation, as well as their low correlation with other asset classes. This attribute provides a chance for portfolio diversification, making portfolios that include REITs less risky than those without, as illustrated in Section 5.

There are several REIT types, including Equity REITs (e-REITs), Mortgage REITs (m-REITs), Public Non-Listed REITs, and Private REITs. The most common type of REITs on the market are Equity REITs, which own or operate income-producing real estate. Mortgage REITs (mREITs) finance income-producing real estate by purchasing or creating mortgages and mortgage-backed securities, earning interest-based income from these investments. Public non-listed REITs are registered with the SEC but do not trade on national exchanges, whereas private REITs are not traded on national exchanges and are exempt from SEC registration.

Like other financial markets, REIT share prices fluctuate throughout the trading day. The value of REIT shares is influenced by various factors such as expected earnings growth, expected total returns, dividend yields compared to other yield-oriented investments like bonds or utility stocks, dividend payout ratios, management quality, corporate structure, and the underlying asset values of the real estate and mortgages. REIT market values are represented by different indices, including the FTSE EPRA/Nareit US Real Estate Index, which contains specific REIT companies operating in the US. This study focuses on

[2] https://www.reit.com/investing/why-invest-reits.

publicly listed equity REITs (e-REITs) like American Tower Corporation (AMT), Prologis (PLD), Crown Castle (CCI), Public Storage (PSA), and Welltower (WELL) that hold various types of real estate properties such as infrastructure, offices, shopping malls, and others.

2.2. Modern portfolio theory

Modern portfolio theory (MPT) is a conceptual framework that helps investors tackle asset allocation issues. MPT is based on the premise that investors are risk-averse and favor portfolios that offer the same anticipated return with lower risk. Investors will only select a more risky portfolio if it is accompanied by a higher expected return. The compromise between maximizing returns and minimizing risk is determined by the degree of individual risk aversion. MPT provides a mathematical approach to resolving this trade-off.

The MPT is a theory that relies on the Efficient Market Hypothesis (EMH), which presumes that a security's price incorporates all the available information and reflects its economic value. An efficient market is one in which a security's price is influenced solely by all available information and not by managerial decisions. This theory is critical in guiding investment decision-making and helps investors predict future market trends that may affect asset allocation.

According to MPT, an effective portfolio is one that maximizes expected return for a given level of risk or minimizes expected risk for a given level of return. To determine the expected return of a portfolio, the past returns of the assets in the portfolio are taken into account, and their weights are determined based on the proportions assigned to each asset class.

The equation for the expected return of a portfolio is typically given as:

$$E(r_p) = \sum_{i=1}^{n} w_i E(r_i), \tag{1}$$

where $E(r_p)$ is the expected return of the portfolio, w_i is the weight of the ith asset in the portfolio, $E(r_i)$ is the expected return of the ith asset, and n is the number of assets in the portfolio.

In the MPT, the expected risk of a portfolio is not solely determined by the individual risks of its constituent assets. Instead, the expected risk of a portfolio depends on the interdependence of the assets in the portfolio, which is captured by their pairwise correlations. The higher the pairwise correlations between assets in the portfolio, the higher the expected risk of the portfolio.

More specifically, the expected risk of a portfolio can be expressed as a function of the variances and pairwise correlations of its constituent assets. This function is commonly known as the portfolio variance equation and is given by:

$$\sigma_p^2 = \sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{i,j} \tag{2}$$

where σ_p^2 is the expected risk (variance) of the portfolio, w_i and w_j are the weights of assets i and j in the portfolio, $\rho_{i,j}$ is the pairwise correlation coefficient between assets i and j, and σ_i and σ_j are the standard deviations of returns for assets i and j, respectively.

Therefore, the expected risk of a portfolio is influenced not only by the individual risks of its constituent assets, but also by their pairwise correlations. In other words, diversification can help reduce the expected risk of a portfolio by combining assets that have low pairwise correlations, as they can offset each other's risk.

The correlation between two assets measures the strength and direction of their relationship. If two assets have a positive correlation, they tend to move in the same direction, while if they have a negative correlation, they tend to move in opposite directions.

In MPT, when the correlation between two assets in a portfolio is low, their movements tend to offset each other, resulting in a lower overall portfolio risk. In contrast, when the correlation between two assets is high, their movements tend to be in the same direction, increasing the overall portfolio risk.

Furthermore, when a portfolio contains many highly correlated assets, the overall portfolio risk can become very high. However, if the portfolio contains a mix of assets with different correlation levels, the overall risk can be reduced. This is because assets with low or negative correlations can help offset the risk of highly correlated assets in the portfolio.

Therefore, in MPT, managing correlation levels between assets in a portfolio is a key factor in determining the expected risk of the portfolio. The goal is to construct a portfolio that achieves the highest expected return for a given level of risk, taking into account the correlations between the assets in the portfolio.

3. 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 above two steps. We will first present the various data pre-processing measures we needed to undertake before using the machine learning algorithms, in Section 3.1. Then, Section 3.2 offers a brief overview of how we utilized the machine learning algorithms. We also present the cost function used by all the algorithms in Section 3.3. Finally, in Section 3.4, we present the genetic algorithm that we used to tackle our portfolio optimization problem.

3.1. Data preprocessing

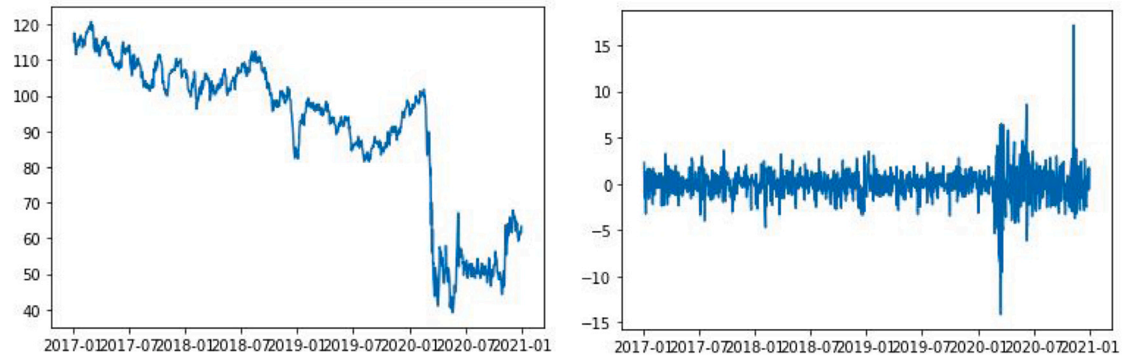
Each time series data is preprocessed by differencing and scaling before being used for price prediction. Differencing makes the time series stationary, removing the upward trend and keeping the average constant over time. Stationarity is crucial in time series analysis because several models, including ARIMA, assume that data are independent of one another. Since market price time series often exhibit time dependence, it is necessary to remove this dependence in order to apply prediction models.

Differencing involves taking the difference between consecutive observations in the time series data, such that $D_t = P_t - P_{t-1}$. For example, the price at time $t1$ is transformed into $D_{t2} = P_{t2} - P_{t1}$. This has the effect of removing the trend component of the time series, making it stationary. The resulting time series data will have a constant mean and variance, and its statistical properties will be consistent over time. The original SL Green Realty Corp REIT (SLG) time series (prior to differencing) is shown in Fig. 1(a), while the differenced time series is shown in Fig. 1(b).

Stationarity is important in time series analysis because many statistical models, such as ARIMA, assume that the data are stationary. By differencing a non-stationary time series, we can make it stationary and use these models to make predictions about future observations in the time series.

Once D_t has been obtained, its values are then scaled to be in the range of 0 and 1, according to the following transformation, presented in Eqs. (3):

$$N_t = \frac{(D - D_{min})}{(D_{max} - D_{min})} \tag{3}$$



(a) SLG original time series (b) SLG differenced time series

Fig. 1. Differencing example.

Table 1
Example of time series differentiation and feature selection.

t	P_t	P_{t-1}	D_t	N_t	N_{t-1}	N_{t-2}
t1	63.88	–	–	–	–	–
t2	61.70	63.88	−2.19	0.07	–	–
t3	59.20	61.70	−2.50	0	0.07	–
t4	59.40	59.20	0.20	0.64	0	0.07
t5	60.21	59.40	0.80	0.79	0.64	0
t6	60.24	60.21	0.03	0.60	0.79	0.64
t7	61.93	60.24	1.69	1	0.60	0.79
t8	61.26	61.93	−0.67	0.44	1	0.60
t9	61.62	61.26	0.36	0.68	0.44	1
t10	63.26	61.62	1.64	0.99	0.68	0.44

where N_t is the standardized value of each variable (in this case the differenced price D), and D_{min} and D_{max} are the minimum and maximum value for D respectively, over all data in each dataset.

Table 1 illustrates the differencing and scaling processes using the SLG time series data between 01 January 2021 and 15 January 2021. The table includes columns for the time steps, the security price P_t, the one-lag value of P_t, the differenced D_t value, the scaled N_t variable, and the lagged values for N_t. To obtain D_t, the one-lag value is subtracted from P_t for each time point. This process creates stationary data by eliminating the upward trend and making the average constant over time. Stationarity is essential for time series analysis as several models require data independence. The fifth column contains the scaled D_t values, which are transformed into N_t. The scaling process normalizes the independent variable to a range between 0 and 1. For example, at time step $t2$, D_t is equal to −2.19, which becomes 0.07 after normalization.[3]

The target variable is N_t, and the predictors consist of past observations of N_t, such as N_{t-1}, N_{t-2}, N_{t-3}, and so on, up to N_{t-n}. The optimal value of n is selected by employing the Akaike Information Criterion (AIC), which is a widely used metric for model selection (Vrieze, 2012; Yamaoka, Nakagawa, & Uno, 1978). The value of n may vary depending on the specific dataset, resulting in a different number of features for each dataset.

After having obtained the predicted time series, we revert the differencing and scaling process in order to calculate the cost function described in Section 3.3.

3.2. Machine learning algorithms for price prediction

Once the data has been pre-processed and the relevant lagged features have been created, they can then be passed to the machine learning algorithms to predict the price of the datasets, which fall into one of three asset classes, namely REITs, stocks and bonds. More information about the datasets will be provided in Section 4. The price thus forms the target variable of this regression task. The objective of this task is to demonstrate that using machine learning algorithms leads to a significantly reduced error when predicting the price of a REIT, stock, or bond.

This study compares the performance of five commonly-used machine learning algorithms, namely linear regression, support vector regression, extreme gradient boosting, long/short-term memory neural network and k-nearest neighbors regression. A brief introduction of these algorithms is presented in Appendix A.

To apply these machine learning algorithms, we used the following python libraries: *sklearn*, *xbgoost*, and *keras*. The functions used to fit the algorithm to the training data include

- `sklearn.linear_model.LinearRegression`,
- `sklearn.svm.SVR`
- `xgboost`
- `Sequential`
- `sklearn.neighbors.KNeighborsRegressor`

The parameters included in such functions were determined using a grid search method, which is described in Section 4. The algorithms were then fitted to the training data and subsequently applied to the test set using the `predict` attribute of the relevant model.

3.3. Cost function

For our regression problem, we use the *root mean square error* as cost function, which is presented in Eq. (4):

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(P_t - \hat{P}_t)^2}{T}}, \tag{4}$$

where P_t refers to the actual value of the price, $\hat{P}_t$ is its predicted value, and T is the number of observations. Please note that as it was explained in Section 3.1, the differenced and scaled values are reverted back to their original price values, so that the cost function can be calculated. That is why in the above equation we use P_t and not D_t or N_t.

3.4. Portfolio optimization via genetic algorithm

Once we have obtained the price predictions, our next step is to use them as input to a portfolio. As already mentioned, we use three different asset classes for the portfolio (REITs, stocks, and bonds). Our objective is to demonstrate that using ML price predictions leads to better portfolio performance. We will assess portfolios in terms of three financial metrics, namely Sharpe ratio, returns, and risk.

[3] To normalize the D_t values, we need to identify the minimum and maximum value for D_t, which are −2.50 and 1.69 respectively. Then, we use Eq. (3) to normalize the D_t values. For instance, to normalize D_2, we calculate the result of $N_t = \frac{D_t-(-2.50)}{1.69-(-2.50)}$ that is 0.07.

Fig. 2. A sample chromosome for a 5-asset portfolio. Each gene (cell) represents a different asset in the portfolio. The value inside each gene represents the allocated weight for that particular asset in the portfolio.

To optimize the weights of the assets in the portfolio we use genetic algorithms (GA) (Goldberg, 1989; Michalewicz, 2013; Mitchell, 1996). GAs have been applied successfully to a wide range of problems, including algorithmic trading (Adegboye, Kampouridis, & Otero, 2023), engineering design (Deb, 2011), financial portfolio optimization (Li, Liang, Li, & Liu, 2015), and image recognition (Liu, Liu, & Xin, 2002).

The following section provides a brief overview of the GA we have utilized.

Representation In GAs, the process of initializing the population involves creating a set of initial solutions to the problem at hand. The population typically consists of chromosomes, each of which represents a potential solution. In the context of portfolio optimization, for example, a chromosome would be comprised of N *genes* that correspond to the weights of N assets in the portfolio. These weights are real numbers between 0 and 1 and must sum to 1. For instance, let us have a look at Fig. 2, where a sample chromosome is presented. This chromosome contains 5 genes (cells), where each gene represents a different asset in the portfolio. The value inside each gene indicates the allocated weight for that particular asset in the portfolio. Hence, Asset 1 has a weight of 31%, Asset 2 a weight of 8% Asset 3 a weight of 2%, Asset 4 a weight of 29%, and Asset 5 a weight of 30%. During population initialization, each gene is assigned an equal weight (i.e., $W_i = 1/N$ for each asset i), and these weights are then updated through a set of operators during the evolution process.

Operators The GA operators utilized in our approach are elitism, one-point crossover, and one-point mutation. Considering the limited number of datasets, one-point crossover and mutation operators are considered to be adequate (additional details can be found in Section 4). To maintain the total weight of 1 for all assets, we perform normalization on each GA individual following the application of the crossover and mutation operators.

Fitness function Several metrics have been employed as fitness functions for solving portfolio optimization problems in the state-of-the-art methods. In this study, we employ the Sharpe ratio, which is computed as the ratio of the difference between the mean return and the risk-free rate to the standard deviation of the returns, that is,

$$S = \frac{r - r_f}{\sigma_r}, \tag{5}$$

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.

The average return of each asset is calculated as the simple average of the returns of that asset, that is,

$$r = \frac{\sum_{t=1}^{T} r_t}{T}, \tag{6}$$

where r_t is the return observed for each time point t and N is the number of observations.

To calculate the return r_t, we need to transform the price time series to returns through the following formula:

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}} \tag{7}$$

The standard deviation of returns is calculated as the square root of the average of the squared differences between the average return and each observed return follows.

$$\sigma_r = \sqrt{\frac{\sum_{t=1}^{T} (r - r_t)^2}{T}}. \tag{8}$$

The above three metrics (Sharpe ratio, average return, risk) are derived from the MPT (see Section 2.2), and are presented for both out-of-sample over a period, and for out-of-sample one-day-ahead forecast (see Section 3.3).

4. Experimental setup

Our experiments' aim is twofold: (a) predict REITs, stocks, and bonds prices through five machine learning algorithms: linear regression (LR), support vector regression (SVR), XGBoost, LSTM, and KNN, and (b) demonstrate the improvements in the performance of a portfolio including stocks, bonds, and REITs that can be achieved by including ML based predictions.

In the remainder of this section, we will first present the data used for our experiments, in Section 4.1. We will then discuss the algorithmic hyperparameter tuning in Section 4.2. Lastly, in Section 4.3 we will discuss the benchmarks used in our experiments.

4.1. Data

We gathered daily prices for financial instruments belonging to three asset classes (stocks, bonds, and real estate) and to three countries (US, UK, and Australia) for the period between January 2019 and July 2021, using the *Eikon Refinitiv* database. We selected ten stocks, ten bonds, and ten REITs for each of the three markets, resulting in a total of 90 datasets (see Table 2). To mitigate the currency risk, we downloaded all datasets in USD.

It is worth noting that a lot of the datasets' price series can fluctuate significantly, particularly stocks and REITs. For example, let us look at Fig. 3, which presents the US REIT close price time series for the period between 1st January 2021 and 1st July 2021. As we can observe, there are high variations in the trend, especially in the lower part of the distribution. This can of course affect the performance of some algorithms, particularly ARIMA's, which is heavily dependent on past observations.

Table 3 presents summary statistics for the return distributions grouped by each of the nine asset classes considered. We summarized those return distributions in terms of mean, median, standard deviation, interquartile range and maximum–minimum range. Each asset is assigned the same weight inside the asset class. We consider the training period in calculating the summary statistics. We noticed that bond rates of return present lower volatility values with respect to other asset classes, and at the same time, lower average values. Stock markets tend to be more volatile, and at the same time more profitable compared to the other asset classes. Real estate returns are positioned in between in terms of expected return and volatility. For instance, for the Australian market, bonds present an average return of 1.97E−04, while REITs have a greater return of 7.35E−04, and stocks show the greatest return average value of 2.00E−03. In terms of volatility, bonds show the lowest value at 5.70E−03, while stocks present the highest value at 2.44E−02, and REITs are positioned in between with a value of 1.44E−02. This explains the higher return, and lower risk that portfolios including real estate present with respect to portfolios including stocks and bonds only (Habbab et al., 2022).

Moreover, in Fig. 4, we present the correlation values between the different asset classes. As we can observe the real estate asset classes generally present low correlation with respect to the other asset classes especially in the case of international investments, explaining the diversification potential, and thus lower risk level, that REITs could bring to a mixed-asset portfolio (Fig. 4). For instance, the value observed for correlation between UK REITs and Australian stocks is −0.23. These values explain why a portfolio including international REIT investments could reduce the risk of a multi-asset portfolio.

To summarize, our findings indicate that, on average, REITs show higher returns compared to bonds and lower returns compared to stocks. This is attributed to a higher risk level when compared to

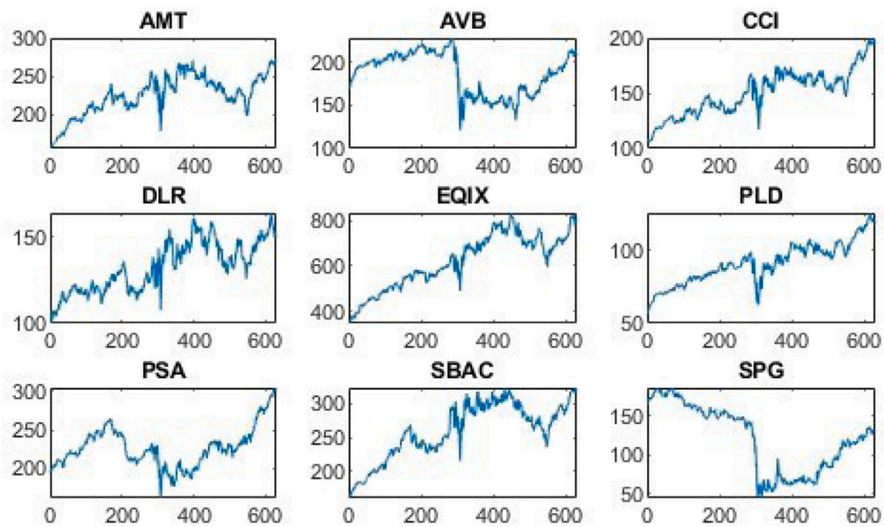


Fig. 3. US REIT time series.

Table 2
Eikon Refinitiv tickers for the data sets 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
REITs	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

Table 3
Summary statistics for different asset classes.

	Average	Median	Std Dev	IQR	Max–Min
AU bonds	1.97E−04	3.15E−04	**5.70E−03**	**3.00E−03**	**9.54E−02**
AU REITs	7.35E−04	1.20E−03	1.44E−02	1.87E−02	2.95E−01
AU stocks	**2.00E−03**	**1.80E−03**	2.44E−02	2.14E−02	2.59E−01
UK bonds	2.38E−04	3.86E−04	7.90E−03	5.70E−03	1.12E−01
UK REITs	7.11E−05	4.35E−04	2.56E−02	2.14E−02	3.51E−01
UK stocks	1.88E−04	3.83E−05	2.14E−02	1.93E−02	2.61E−01
US bonds	3.11E−04	2.74E−04	8.50E−03	7.70E−03	1.07E−01
US REITs	6.99E−04	7.25E−04	2.59E−02	1.95E−02	3.49E−01
US stocks	1.10E−03	1.20E−03	2.25E−02	1.86E−02	2.40E−01

bonds and a lower risk level when compared to stocks. Essentially, the return-to-risk profile of the real estate asset class falls between the other asset classes, allowing for diversification potential in a mixed-asset portfolio. Additionally, we observed that the returns of REITs display a low correlation with other asset classes, which helps explain the reduced risk associated with including REITs in a mixed-asset portfolio. These two findings confirm that the inclusion of REITs in a mixed-asset portfolio could potentially reduce the overall risk of that portfolio, while also increasing its return. Investors look for a greater diversification which is made possible by investing in asset classes that have low correlation.

4.2. Experimental parameters tuning

After differencing and scaling each dataset (see Section 3.1), we split that into three sets: training, which is for the period January 2019–June 2020, and is used to fit a model from each algorithm to the given data; validation, which is for the period July 2020–December 2020, and is used to tune the hyperparameters of each model; and test, which is for the period January 2021–July 2021, and is used to assess the final performance of a previously trained model. During the grid search tuning phase, the validation set was utilized to determine the experimental hyper-parameters.

To address the price prediction problem (using ML algorithms), we customized the experimental parameters for each dataset through tuning. Thus, each dataset has its own unique experimental parameters. The complete range of parameters for each model can be found in Table 4. The Grid Search method in Python was used to select the optimal parameters. The ranges for parameter values were established based on the dataset types used. Furthermore, since LR lacks parameters to be tuned, we did not conduct parameter tuning for this model.

GA parameter values were tuned on the same validation set. The resulted tuned values are presented in Table 5.

4.3. Benchmarks

In addition to investigating the performance of the five ML algorithms for the problems of price prediction and portfolio weights optimization, we also explore the performance of three common financial benchmarks, which are presented next, in Section 4.3.1. 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 4.3.2.

4.3.1. Machine learning benchmarks

Holt's exponential smoothing The *Holt's Exponential Smoothing* (HES) technique uses the exponential window function to forecast time series data. The use of an exponential window function was first introduced by Poisson (Oppenheim & Schafer, 1975) and recommended in the statistical literature by Brown (1956). Holt (2004) further developed this technique. Unlike the simple moving average (SMA) technique, which assigns the same weight to all past observations, HES assigns exponentially decreasing weights over time. HES employs a window function to eliminate noise in time series data, that is, observations

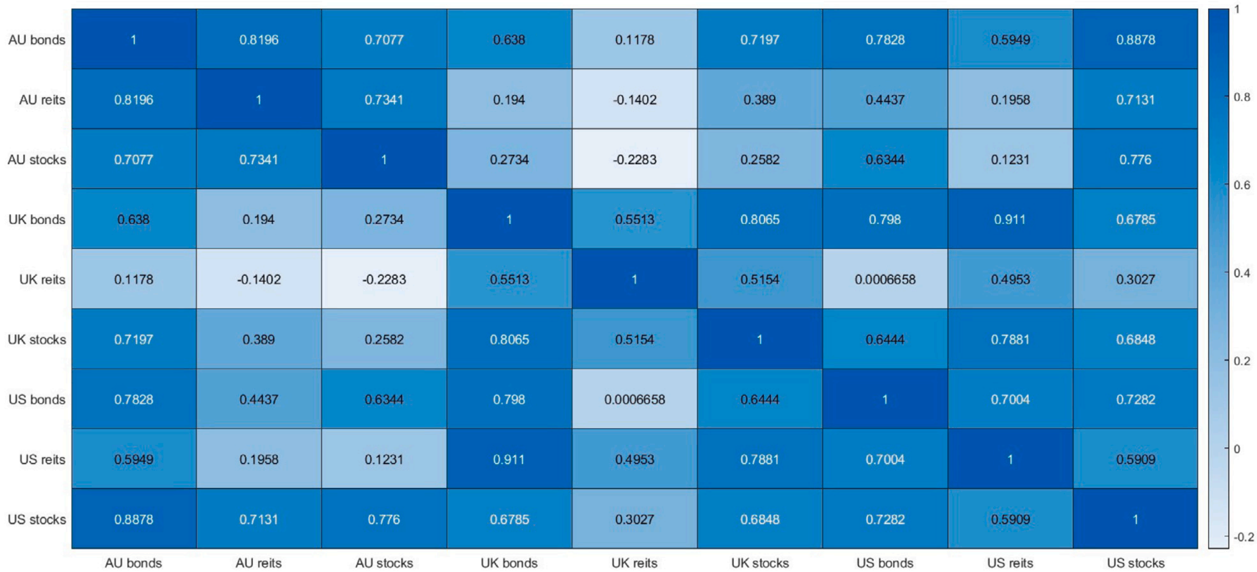


Fig. 4. Correlation matrix between asset classes.

Table 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
GA parameters.

Parameter	Values
Population size	500
Tournament size	3
Mutation rate	0.1
Number of generations	25

that deviate greatly from the average. If the time series is composed of $N_0, N_1, N_2, \dots, N_t$ points, the function can be expressed as follows.

$$\hat{N}_0 = N_0 \tag{9}$$

$$\hat{N}_t = \alpha N_t + (1-\alpha) n_{t-1}, \tag{10}$$

where α is the smoothing parameter, and $0 < \alpha < 1$. The value for α results from a parameter tuning process. The $\hat{N}_t$ value is a simple weighted average of the current observation N_t and the previous smoothed values N_{t-1}, where α is the weighting factor. Larger values of α reduce the level of smoothing, and in the case of $\alpha = 1$ the resulting time series contains the actual values. Higher values for α give more weight to recent observations, while lower values for α have a greater smoothing effect, and thus assigns less weight to recent observations. The optimal value for alpha was found within the range $[0.1, 0.2, 0.3]$ on the training set, and verified on the validation set.

TBATS TBATS is an abbreviation for Trigonometric seasonality, Box–Cox transformation, ARMA errors, Trend, and Seasonal components. It refers to a set of models created for fitting seasonal periods. The basic premise behind these models is that high-frequency data requires a transformation to be used effectively in fitting a model.

To accomplish this transformation, the Box–Cox method is applied using an exponential parameter, lambda λ, which can take on any value between −5 and 5. The optimal value of λ is determined by the best approximation of a normal distribution curve. In this study, we fine-tuned the λ parameter by identifying the optimal value for λ based on the training–validation split. This transforms the time series into a stationary state, where statistical properties such as the mean and standard deviation remain constant over time.

The Box–Cox transformation of N_t is represented as below:

$$N_t(\lambda) = \begin{cases} \frac{N_t^{\lambda}-1}{\lambda}, & \text{if } \lambda \neq 0, \\ \log N_t, & \text{if } \lambda = 0. \end{cases} \tag{11}$$

Regarding the other components, the ARMA (which stands for AutoRegressive Moving Average) errors process attempts to capture information in the residuals. On the other side, the trend component explains the long-term change in the average values of a time series. Finally, the seasonal component explains the periodical variation in the series (e.g., daily or monthly).

ARIMA The *Autoregressive Integrated Moving Average* (ARIMA) models are used to study the structure of a time series. These models forecast the value of a variable (e.g., current market price) by utilizing its historical values and the distribution of its errors. ARIMA is frequently used in finance for time series prediction and is used as a benchmark in this study.

An ARIMA model of order (p, d, q) consists of three components: the autoregression model of order p, differencing of order d, and the moving average model of order q. The mathematical form of ARIMA is given by Eq. (12), where N_t represents the time series.

$$N_t = c + \sum_{i=1}^{p} \phi_i N_{t-i} + \epsilon_t + \sum_{j=0}^{q} \theta_j \epsilon_{t-1} \tag{12}$$

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.

To determine the appropriate ARIMA model for each training dataset, we utilized the AIC criterion as mentioned in Section 3.1, and selected the p, d, and q order that corresponded to the minimum AIC value.

It is important to note that for ARIMA models to be applied, the time series being analyzed must be stationary, meaning that its statistical properties remain constant over time. Since many financial time series are not stationary, various transformations such as differencing, logarithmic transformation, and Box–Cox transformation are required.

4.3.2. Additional benchmarks for portfolios

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 (Jones & Trevillion, 2022; Lee & Moss, 2018; Parikh & Zhang, 2019). 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. Results

In this section, we analyze the predictive power of our five machine learning algorithms in terms of RMSE distributional statistics (Section 5.1), and the implications for the expected returns, risks, and Sharpe ratio of a multi-asset portfolio (Section 5.2).

5.1. RMSE

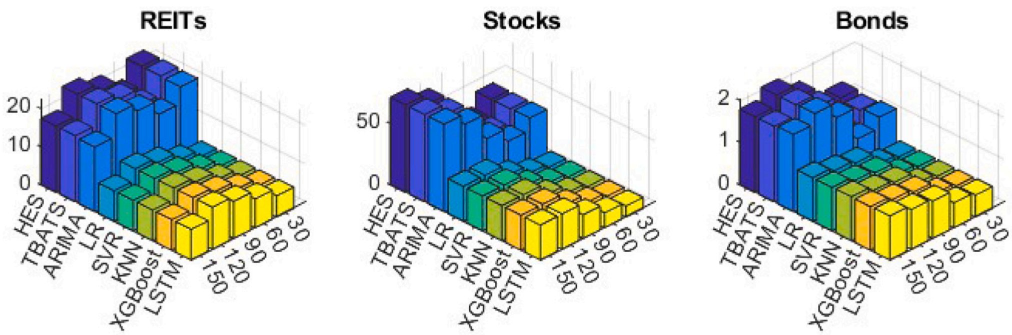
In this section, we examine the predictive power of the eight algorithms (five ML algorithms and three benchmarks) used against two out-of-sample prediction methods. First, we evaluate out-of-sample predictions over different time periods (30, 60, 90, 120, and 150 days); second, we make one-day-ahead predictions over a specific period. In the first case, the out-of-sample predictions, today's price (at time-step 0) is known and used to predict the price of tomorrow (at time-step 1). However, tomorrow's price is unknown and cannot be used to predict the price two days ahead (at time-step 2). Hence, we use the price at time-step 1 to predict the price at time-step 2, and so on. We call this method the out-of-sample over a period forecast.[4] In the second case, the one-day-ahead prediction, we operate as follows. The price of today (at time-step 0) is known, and is used to predict tomorrow's price (at time-step 1). Then, tomorrow's price is used to predict the price at time-step 2, and so on. We refer to this method as the one-day-ahead over a period forecast.[5] Naturally, we expect to obtain lower error rates from the second technique.

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

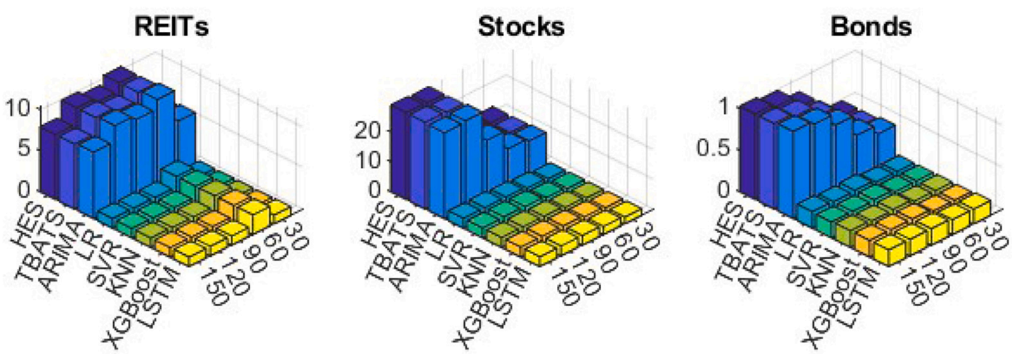
With regards to the one-day-ahead results, we can make similar observations: ML errors are again considerably lower than the benchmarks, and bonds experience the lowest error, followed by REITs, and then by stocks. One important difference to the previous (out-of-sample) results is that one-day-ahead consistently experiences lower errors, which is expected, as it was explained earlier. As we can

[4] To provide a clearer explanation, let us consider a numerical example. Suppose we are using linear regression with 5 lagged values, meaning we use the previous five price values as features for prediction. These five data points, denoted as P_1, P_2, P_3, P_4, and P_5, have the respective values of 0.07, 0, 0.64, 0.79, and 0.60. Let us assume the next two prices, at times t_6 and t_7 (i.e. prices P_6 and P_7), represent the test set and have values of 1 and 0.44 respectively. To calculate the out-of-sample prediction at time t_6, we use the linear regression equation: $\hat{P}_6 = \beta_0 + \beta_1 \cdot P_1 + \beta_2 \cdot P_2 + \beta_3 \cdot P_3 + \beta_4 \cdot P_4 + \beta_5 \cdot P_5 = \beta_0 + \beta_1 \cdot 0.07 + \beta_2 \cdot 0 + \beta_3 \cdot 0.64 + \beta_4 \cdot 0.79 + \beta_5 \cdot 0.60$, where β_0 to β_5 are the coefficients estimated by the algorithm. Let us assume the algorithm's prediction $\hat{P}_6$ is 0.75. Now, to calculate the prediction at time t_7, we consider the prices from t_2 to t_5 (i.e. P_2 to P_7), along with the prediction for time t_6 (i.e. $\hat{P}_6$, which was equal to 0.75). Therefore, the price at time t_7 is predicted as: $P_7 = \beta_0 + \beta_1 \cdot P_2 + \beta_2 \cdot P_3 + \beta_3 \cdot P_4 + \beta_4 \cdot P_5 + \beta_5 \cdot \hat{\mathbf{P}}_6 = \beta_0 + \beta_1 \cdot 0 + \beta_2 \cdot 0.64 + \beta_3 \cdot 0.79 + \beta_4 \cdot 0.60 + \beta_5 \cdot 0.75$. As we can observe, the last feature was the predicted price $\hat{P}_6$ (equal to 0.75), rather than the actual price P_6 (equal to 1). This is because under the out-of-sample approach, the actual values are unknown in the test set. Hence, to compute our predictions, we use the predicted values from the previous time steps.

[5] Let us again consider a numerical example by using the same data points as in the previous footnote. The prediction at time t_6 (i.e. $\hat{P}_6$) is done in the same way as for the out-of-sample prediction method. However, to calculate the prediction at time step t_7 (i.e. $\hat{P}_7$) under the one-day-ahead paradigm, we use the actual value (i.e. P_6) at time step t_6. Hence, the predicted value is computed as follows: P $= \beta_0 + \beta_1 \cdot P_2 + \beta_2 \cdot P_3 + \beta_3 \cdot P_4 + \beta_4 \cdot P_5 + \beta_5 \cdot P_6 = \beta_0 + \beta_1 \cdot 0 + \beta_2 \cdot 0.64 + \beta_3 \cdot 0.79 + \beta_4 \cdot 0.60 + \beta_5 \cdot 1$. As we can observe, the last feature was the actual price P_6, and not the predicted $\hat{P}_6$, as it happened with the out-of-sample approach.



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



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

Fig. 5. Comparison of RMSE results.

observe, the highest error per asset class tends to be at least 50% lower for the one-day-ahead method (REITs: from 20 to 10; Stocks: from 70 to 30, Bonds: from 2 to 1). Lastly, we have provided, for reference, detailed distribution statistics for all RMSE results in the Appendix, Tables B.8–B.10.

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 6. For each algorithm, the table shows the average rank (first column), and the adjusted p-value of the statistical test when that algorithm's average rank is compared to the average rank of the algorithm with the best rank (control algorithm) according to Bonferroni's post-hoc test (second column) (Demšar, 2006; Garcia & Herrera, 2008). When statistically significant differences between the average ranks of an algorithm and the control algorithm at the 5% level ($p \leq 0.05$) are observed, the relevant p-value is put in bold face. The statistical tests were conducted for all different setups, i.e., the combined results of different horizons (30-, 60- 90-, 120-, and 150-days), over both the one-day-ahead and out-of-sample experiments.

We can observe that the best (control) algorithm is KNN which statistically outperforms LSTM, LR, HES, TBATS, and ARIMA (given p-values equal to 1.86E−05; 6.83E−08; 5.76E−24; 5.76E−24; and 3.58E−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).

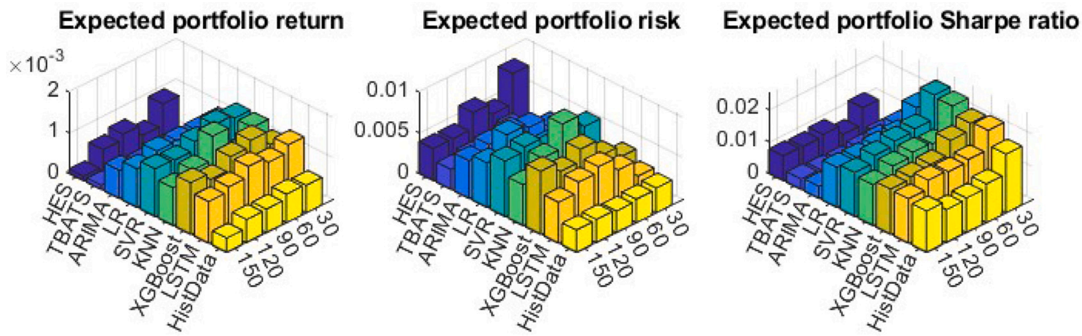
In conclusion, we observed that the RMSE distributions tend to be lower on average for ML algorithms than for benchmark algorithms, with better results observed for one-day-ahead prediction (as expected). We also noticed that the lowest average RMSE values are observed for bonds, followed by REITs and stocks. This is explained by the lower volatility featuring bond prices that we have already discussed in Section 4.1. 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 6
Statistical test results according to the non-parametric Friedman test with the Bonferroni's post-hoc test RMSE distributions. Values in bold represent a statistically significant difference at the 5% significance level.

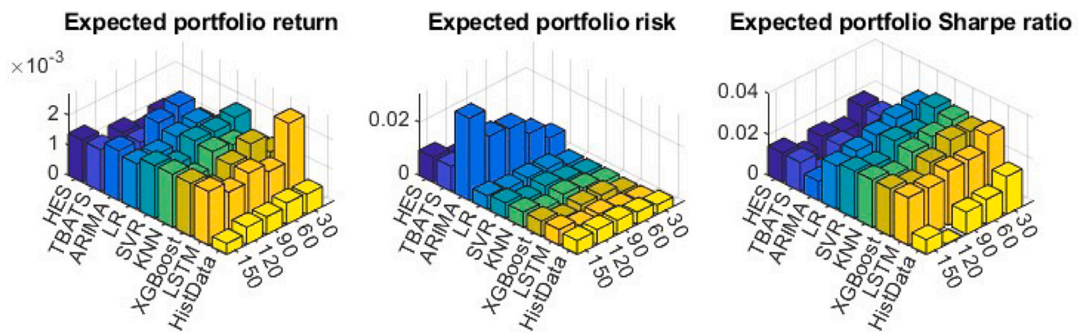
Algorithm	Average rank	p_{Bonf}
KNN (c)	2.88	–
SVR	2.91	5.62
XGBoost	3.07	0.76
LSTM	3.42	**1.86E−05**
LR	3.54	**6.83E−08**
HES	6.46	**5.76E−24**
TBATS	6.46	**5.76E−24**
ARIMA	7.26	**3.58E−45**

5.2. Portfolio optimization

In this section, we present results for the genetic algorithm (GA) applied to portfolio allocation considering a transaction cost of 0.02%. After calculating the optimal weights, we obtain the distribution for expected returns, expected risks and Sharpe ratio for each of the datasets.



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



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

Fig. 6. Comparison of portfolio results. For reference, the perfect foresight values for returns are 4.16E−03 (30 days), 4.07E−03 (60 days), 4.56E−03 (90 days), 3.85E−03 (120 days), and 3.78E−03 (150 days). The perfect foresight values for risk are 1.14E−03 (30 days), 2.42E−03 (60 days), 2.51E−03 (90 days), 2.58E−03 (120 days), and 2.34E−03 (150 days). The perfect foresight values for Sharpe ratio are 4.04E−02 (30 days), 3.72E−02 (60 days), 3.72E−02 (90 days), 3.29E−02 (120 days), and 3.23E−02 (150 days).

The ML models are compared to the benchmarks (HES, TBATS, and ARIMA), to the historical data method, and to the theoretical perfect foresight approach. In the case of historical data, portfolio optimization is performed on the training set, while the expected portfolio metrics are obtained using data from the testing set. Under the perfect foresight approach, the portfolio optimization takes place on the testing set.

Fig. 6 presents the expected return distributions (left), expected portfolio risk (middle), and expected Sharpe ratio (right) obtained from the GA portfolio optimization task for a 30-, 60-, 90-, 120-, and 150-day holding period, for out-of-sample (top) and one-day-ahead (bottom). With regards to the portfolio returns, we can observe that for the out-of-sample method the machine learning algorithms yield higher returns across all holding periods when compared to the benchmark methods and the historical data approach. The highest average daily return for the 30 days is achieved by LSTM and SVR (1.44E−03), followed by KNN (1.43E−03). In the case of one-day-ahead predictions for the same holding period, the expected daily return is higher for all the algorithms, with the highest values achieve by LSTM (2.72E−03). For reference, the average return for the theoretical benchmark of the perfect foresight is 4.16E−03. It is also 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. There is of course room for even greater improvements, given that the 'ceiling' of the perfect foresight is around 57% higher than LSTM's average daily return of 2.72E−03 (for the one-day-ahead method), showing that there are significant research potentials in this area. Lastly, as the horizon period increases to 60 days and higher, we observe similar improvements in the performance of the ML algorithms with respect to the benchmarks. For reference, we have provided detailed tables for returns, as well as risk and Sharpe ratio distributions, in the Appendix, in Tables B.11–B.13.

With regards to portfolio risks, we can generally observe that it tends to be higher for benchmarks with respect to the ML algorithms and the historical approach. However, there are some other cases, particularly in the out-of-sample method, where the econometric benchmarks and the historical data approach outperform the machine learning algorithms. Nevertheless, it is worth noting that the majority of these differences is not significant.[6] Furthermore, we can observe that the risk levels achieved by a perfect foresight-based portfolio (presented in the caption of Fig. 6) 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.

Lastly, when looking at the Sharpe ratio results of Fig. 6, we can observe that all ML algorithms outperform the benchmarks for all periods in both cases of out-of-sample and one-day-ahead predictions. In many cases, the differences in Sharpe ratio values are quite noticeable, e.g. for both out-of-sample and one-step ahead the econometric benchmarks (HES, TBATS, ARIMA) appear to have at least 50% lower values than the ML algorithms. This is an important observation, because it demonstrates the importance of using machine learning for price predictions instead of traditional econometric approaches.

To investigate if the above results are statistically significant, we again performed a Friedman test at the 5% significance level, along

[6] This becomes evident when we have a look at the Friedman ranking, which is presented in Table 7.

Table 7
Statistical test results according to the non-parametric Friedman test with the Bonferroni post-hoc for expected returns (left), expected risks (middle) and expected Sharpe ratios (right). Values in bold represent a statistically significant difference.

(a) Return			(b) Risk			(c) Sharpe ratio		
Algorithm	Average rank	p_{Bonf}	Algorithm	Average rank	p_{Bonf}	Algorithm	Average rank	p_{Bonf}
LSTM (c)	2.97	–	XGBoost (c)	4.26	–	SVR (c)	2.63	–
SVR	2.99	6.71	KNN	4.32	6.35	LSTM	2.82	1.06
KNN	3.30	0.05	SVR	4.42	2.31	KNN	2.96	0.06
XGBoost	3.49	**1.40E−04**	LR	4.42	2.25	LR	3.42	**9.94E−10**
LR	4.06	**2.14E−18**	LSTM	4.44	1.67	XGBoost	3.49	**2.21E−11**
ARIMA	4.88	**1.98E−54**	HistData	4.53	0.37	ARIMA	7.02	**1.05E−279**
HistData	7.35	**3.25E−280**	TBATS	4.84	**4.51E−05**	HES	7.05	**2.84E−283**
HES	7.80	**0**	HES	4.89	**6.80E−06**	TBATS	7.61	**0**
TBATS	8.15	**0**	ARIMA	8.87	**0**	HistData	8	**0**

with the Bonferroni post-hoc test. We present these results in Table 7 for returns (left), risk (middle), and Sharpe ratio (right). With regards to returns, LSTM has the best rank (2.97), followed by SVR (2.99), and KNN (3.30). Given a 5% significance level, LSTM statistically outperforms XGBoost (p-value equal to 1.40E−04), LR (p-value equal to 2.14E−18), ARIMA (p-value equal to 1.98E−54), the historical method (p-value equal to 3.25E−280), HES (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) and KNN (0.05).

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 4.51E−05), HES (p-value equal to 6.80E−06), 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).

With regards to Sharpe ratio, 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 9.94E−10), XGBoost (p-value equal to 2.21E−11), ARIMA (p-value equal to 1.05E−279), HES (p-value equal to 2.84E−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.

5.3. Computational times

The computational times of most algorithms were found to be comparable. On average, ARIMA took approximately 0.168 min to run, while LR, SVR, and KNN took between 0.2 and 0.3 min. LSTM was the most computationally expensive algorithm, taking around 1.818 min to run. With regards to the genetic algorithm, a single run took around 0.3 min 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 (Brookhouse, Otero, & Kampouridis, 2014).

5.4. Discussion

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

Machine learning algorithms are able to outperform econometric approaches for price prediction. The initial objective of our experiments was to compare the performance of ML models against our benchmark models, namely HES, 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 4.1.

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 HES-, 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. Fig. 7 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 (Chen, Lee, & Lee, 2019; Li & Stevenson, 2018; Smith & Johnson, 2021). 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.

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

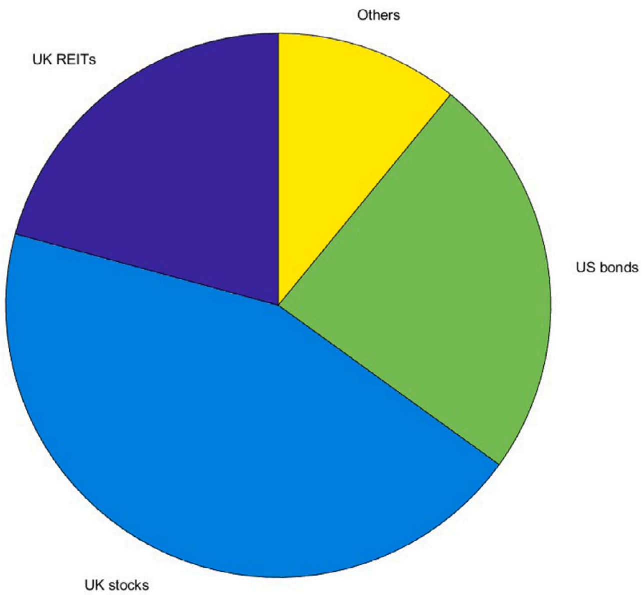


Fig. 7. SVR-GA portfolio weights.

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.

6. Conclusion

To conclude, this work contributed to the current literature about multi-asset portfolio optimization including REITs by incorporating price predictions through the use of machine learning algorithms, rather than relying on historical prices, which is the usual practice. Our experimental findings suggest that ML models outperform the commonly used econometric benchmarks (HES, TBATS, ARIMA) when applied to one-day-ahead and out-of-sample price predictions over 30-, 60-, 90-, 120-, and 150-day periods, due to the lower average RMSE values observed for ML algorithms. Among the ML algorithms, KNN, SVR, and XGBoost performed the best and showed a statistically significant difference compared to LSTM and LR. Furthermore, our results demonstrate that ML-based predictions yield better results in terms of mixed-asset portfolio performance, considering all three metrics of return, risk, and Sharpe ratio. Finally, our analysis highlights the significance of incorporating REITs in such portfolios, as it aids in diversification, maximizes returns, and reduces risk.

The focus of future work will be to improve the predictive capabilities of the machine learning models by incorporating more algorithms and additional features. Our findings suggest that there is potential to enhance portfolio performance beyond the hypothetical scenario of perfect foresight. Since our study relied only on lagged features, such as normalized prices at time step $t-1$, it would be valuable to investigate the impact of incorporating other variables, including technical analysis indicators.

CRediT authorship contribution statement

Fatim Z. Habbab: Conceptualization, Methodology, Implementation, Formal analysis, Visualization, Writing – original draft. **Michael Kampouridis:** Conceptualization, Implementation, Writing – review & editing, Supervision, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Brief introduction on the machine learning algorithms used in this article

The algorithms considered in this study are: Ordinary Least Squares Linear Regression (or LR), Support Vector Regression (or SVR), k-Nearest Neighbor (or KNN), Extreme Gradient Boosting (or XGBoost), and Long-Short Term Memory Neural Networks (or LSTM).

LR is a modeling technique that aims to find a linear relationship between the dependent variable (price) and independent variables (features).

The LR equation for price prediction is given by:

$$\text{Price} = \beta_0 + \beta_1 \cdot \text{Feature}_1 + \beta_2 \cdot \text{Feature}_2 + \cdots + \beta_T \cdot \text{Feature}_T \tag{13}$$

Here, we are predicting the price using a linear combination of the features. Let us break down the components:

- Price: This represents the dependent variable, which is the variable we want to predict. In the context of price prediction, it could represent the price of a security, such as a stock, bond, or REIT share.
- β_0: This is the y-intercept or the constant term. It represents the baseline value of the dependent variable when all the independent variables are zero.
- $\beta_1, \beta_2, \dots, \beta_T$: These are the coefficients corresponding to the independent variables $\text{Feature}_1, \text{Feature}_2, \dots, \text{Feature}_T$, respectively. Each coefficient represents the change in the dependent variable for each unit change in the respective independent variable.

The LR equation allows us to estimate the relationship between the features and the price. By estimating the coefficients (β values), we can determine the impact of each feature on the predicted price.

SVR is a machine learning technique that can handle both linear and non-linear relationships between the dependent variable (price) and independent variables (features).

The SVR equation for price prediction is given by:

$$\text{Price} = \sum_{t=1}^{T} \alpha_t K(\mathbf{N}_t, \mathbf{N}) + b \tag{14}$$

Here, we are predicting the price using a linear combination of the kernel evaluations between the training instances and the new instance **N**. Let us break down the components:

- Price: This represents the dependent variable, which is the variable we want to predict. In the context of price prediction, it could represent the price of a house, a product, or any other relevant variable.
- α_t: These are the Lagrange multipliers, determined during the training process of the SVR model. They control the contribution of each training instance to the prediction.
- $K(\mathbf{N}_t, \mathbf{N})$: This is the kernel function that measures the similarity between two instances $\mathbf{N}_t$ (training instance) and $\mathbf{N}$ (new instance). The choice of kernel depends on the problem at hand and can be linear, polynomial, Gaussian (RBF), or other types.
- b: This is the bias term, which accounts for any translation in the predicted values. It allows the regression line to fit the data more flexibly.

Table B.8
RMSE summary statistics for REITs. Best value per column (for each algorithm) is shown in boldface.

30 days	Out-of-sample				One-day-ahead			
	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HES	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								
HES	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								
HES	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								
HES	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								
HES	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

The SVR equation enables us to estimate the relationship between the features and the price by constructing a hyperplane that maximizes the margin around the predicted values. It finds the optimal weights α_t and bias b to minimize the prediction error while satisfying the specified tolerance.

KNN is a non-parametric machine learning algorithm that relies on the similarity between instances to make predictions.

The KNN regression equation for price prediction is given by:

$$\text{Price} = \frac{1}{T}\sum_{t=1}^{T} \text{Price}_t \tag{15}$$

Here, we are predicting the price by taking the average of the prices of the t nearest neighbors. Let us break down the components:

- Price: This represents the dependent variable, which is the variable we want to predict.
- Price_t: These are the prices of the t nearest neighbors to the instance we want to predict. The t nearest neighbors are chosen based on a distance metric, such as Euclidean distance.
- t: This is the number of nearest neighbors to consider in the prediction. It is a hyperparameter that needs to be specified in advance.

The KNN regression equation allows us to estimate the relationship between the features and the price by finding the most similar instances in the training dataset. By averaging the prices of the k nearest neighbors, we obtain the predicted price for the new instance.

XGBoost is a powerful gradient boosting algorithm that can handle complex relationships and has gained popularity in various machine learning competitions.

The XGBoost regression equation for price prediction is given by:

$$\text{Price} = \sum_{t=1}^{T} f_t(\mathbf{N}) \tag{16}$$

Here, we are predicting the price by summing the outputs of multiple individual regression trees $f_t(\mathbf{N})$. Each tree represents a weak learner that captures a different aspect of the relationship between the features and the price.

XGBoost builds the regression model in a stage-wise manner, where each new tree is trained to correct the residuals of the previous trees.

Table B.9
RMSE summary statistics for stocks. Best value per column (for each algorithm) is shown in boldface.

30 days	Out-of-sample				One-day-ahead			
	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HES	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								
HES	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								
HES	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								
HES	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								
HES	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

This allows XGBoost to handle non-linear relationships, interactions between features, and perform feature selection.

The final prediction is obtained by summing the outputs of all the individual trees. XGBoost employs regularization techniques such as shrinkage, tree depth control, and column subsampling to prevent overfitting and improve generalization performance.

LSTM is a type of recurrent neural network (RNN) that can capture long-term dependencies and patterns in sequential data.

The LSTM regression equation for price prediction is given by:

$$\text{Price}_t = f(\text{Price}_{t-1}, \text{Price}_{t-2}, \dots, \text{Price}_{t-n}, \text{Feature}_{t-1}, \text{Feature}_{t-2}, \dots, \text{Feature}_{t-T}) \tag{17}$$

Here, we are predicting the price at time step t based on the previous T prices and corresponding features. Let us break down the components:

- Price_t: This represents the price at time step t, which is the variable we want to predict.
- $\text{Price}_{t-1}, \text{Price}_{t-2}, \dots, \text{Price}_{t-T}$: These are the previous T prices leading up to time step t.
- $\text{Feature}_{t-1}, \text{Feature}_{t-2}, \dots, \text{Feature}_{t-T}$: These are the corresponding features at each time step, if available.
- $f(\cdot)$: This function represents the mapping from the input sequence of prices and features to the predicted price at time step t. It is modeled by the LSTM network, which consists of LSTM cells and trainable parameters.

LSTM networks are well-suited for capturing complex temporal patterns in price data. The network learns to recognize relevant patterns in the historical prices and features, allowing it to make predictions for future time steps. The training process involves optimizing the network's parameters to minimize the prediction error.

Appendix B. Supplementary tables for results for REITs, stocks, and bonds

In this section, we present summary statistics, namely mean, standard deviation, skewness and kurtosis for all RMSE and portfolio results.

Table B.10
RMSE summary statistics for bonds. Best value per column (for each algorithm) is shown in boldface.

30 days	Out-of-sample				One-day-ahead			
	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HES	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								
HES	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								
HES	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								
HES	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								
HES	1.79	2.37	2.15	4.60	1.03	1.28	**2.09**	**4.01**
TBATS	1.79	2.37	2.15	4.60	1.03	1.28	**2.09**	**4.01**
ARIMA	1.83	2.41	**2.16**	**4.65**	1.05	1.29	2.07	3.96
LR	**1.03**	1.26	2.06	4.13	0.20	**0.19**	1.03	−0.11
SVR	**1.03**	1.26	2.07	4.15	**0.19**	**0.19**	1.00	−0.21
KNN	1.04	1.26	2.06	4.13	0.20	**0.19**	1.03	−0.06
XGBoost	**1.03**	1.26	2.06	4.13	0.20	**0.19**	1.03	−0.06
LSTM	1.04	**1.25**	2.09	4.26	0.20	**0.19**	1.00	−0.18

Table B.11
Expected portfolio return summary statistics. For reference, the perfect foresight values are 4.16E−03 (30 days), 4.07E−03 (60 days), 4.56E−03 (90 days), 3.85E−03 (120 days), and 3.78E−03 (150 days). Best value per column (for each algorithm) is shown in boldface.

30 days	Out-of-sample				One-day-ahead			
	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HES	9.06E−04	**1.78E−06**	6.45	42.24	9.62E−04	1.79E−04	−0.95	13.5
TBATS	1.93E−04	7.73E−05	7.74	64.27	9.02E−04	3.79E−04	7.21	**64.12**
ARIMA	6.73E−04	2.85E−05	**−9.16**	**89.52**	1.25E−03	4.35E−04	−0.2	−0.84
LR	1.12E−03	4.75E−06	−1.52	7.10	1.31E−03	3.52E−04	−0.27	5.89
SVR	**1.44E−03**	4.92E−04	0.34	−1.15	2.01E−03	3.95E−04	6.12	54.35
KNN	1.43E−03	1.25E−05	−1.91	2.06	1.41E−03	2.95E−04	1.15	12.11
XGBoost	1.23E−03	5.04E−04	0.04	0.63	1.47E−03	1.50E−04	−0.85	13.86
LSTM	**1.44E−03**	1.69E−04	0.22	1.09	**2.72E−03**	2.37E−04	**−2.37**	23.91
HistData	6.68E−04	1.56E−05	0.26	3.93	6.68E−04	**1.56E−05**	0.26	3.93
60 days								
HES	3.81E−04	**2.33E−06**	4.65	**34.48**	6.90E−04	1.55E−04	6.03	**47.52**
TBATS	2.40E−04	2.78E−05	2.99	28.45	2.84E−04	1.34E−04	5.6	33.24
ARIMA	6.72E−04	7.48E−05	2.18	3.74	**2.12E−03**	2.16E−04	**−4.15**	20.26
LR	8.40E−04	3.49E−04	2.81	7.89	1.86E−03	1.97E−04	1.12	9.39
SVR	1.52E−03	6.36E−04	2.34	5.32	1.88E−03	1.90E−04	−1.11	19.89
KNN	1.02E−03	9.18E−05	2.18	5.90	1.75E−03	2.17E−04	−1.75	16.76
XGBoost	**1.58E−03**	6.06E−04	1.67	3.36	2.07E−03	2.28E−04	−2.55	9.31
LSTM	1.26E−03	4.09E−05	3.83	22.53	1.45E−03	4.50E−04	1.57	1.88
HistData	7.00E−04	1.46E−05	**−3.09**	13.13	7.00E−04	**1.46E−05**	−3.09	13.13
90 days								
HES	6.49E−04	**4.14E−06**	4.40	18.73	9.84E−04	1.73E−04	−0.27	17.24
TBATS	1.70E−04	1.39E−05	5.66	**41.16**	9.62E−04	1.48E−04	−4.24	22.63
ARIMA	3.92E−04	6.81E−05	2.85	8.08	**1.91E−03**	1.73E−04	−3.12	16.22
LR	8.21E−04	2.08E−04	3.08	11.57	1.74E−03	2.06E−04	−1.07	5.51
SVR	1.35E−03	4.26E−04	−0.35	0.56	**1.91E−03**	1.77E−04	**−4.8**	**23.04**
KNN	**1.70E−03**	2.38E−04	−1.93	3.70	1.85E−03	2.68E−04	−2.71	7.23
XGBoost	1.42E−03	2.89E−04	1.89	18.64	1.71E−03	1.91E−04	2.09	16.91
LSTM	1.40E−03	4.99E−04	−1.13	0.94	1.73E−03	1.46E−04	−3.42	14.83
HistData	5.75E−04	3.88E−05	**−4.07**	20.32	5.75E−04	**3.88E−05**	−4.07	20.32
120 days								
HES	5.19E−04	**1.05E−18**	0.88	−1.70	5.48E−04	9.70E−05	3.81	**34.42**
TBATS	1.85E−04	1.04E−05	4.93	**35.12**	3.75E−04	1.39E−04	4.79	31.22
ARIMA	3.21E−04	2.92E−05	0.02	−0.73	8.56E−04	8.18E−05	0.76	15.17
LR	1.14E−03	4.15E−04	0.99	−0.89	**1.49E−03**	1.76E−04	0.02	10.42
SVR	1.14E−03	2.26E−04	1.64	3.77	1.42E−03	6.14E−04	−0.20	−1.11
KNN	1.12E−03	3.50E−04	1.7	2.82	1.32E−03	1.15E−04	2.09	12.94
XGBoost	1.11E−03	2.79E−04	3.45	11.84	1.22E−03	1.93E−04	**−0.56**	12.53
LSTM	**1.15E−03**	2.72E−04	**−0.30**	1.25	1.43E−03	1.77E−04	−0.29	6.53
HistData	5.67E−04	2.73E−05	1.05	23.44	5.67E−04	**2.73E−05**	1.05	23.44
150 days								
HES	1.13E−04	1.06E−04	7.68	**66.53**	1.40E−03	8.89E−05	−0.06	25.79
TBATS	1.11E−04	1.05E−04	5.77	41.93	1.38E−03	1.15E−04	**−4.95**	25.24
ARIMA	6.94E−04	1.09E−04	3.92	31.66	1.65E−03	1.33E−04	−4.21	19.16
LR	9.53E−04	5.63E−05	**−2.56**	12.75	1.51E−03	9.19E−05	−4.3	18.65
SVR	1.17E−03	3.15E−04	0.36	3.69	1.75E−03	1.77E−04	−3.39	12.69
KNN	9.31E−04	4.53E−05	0.21	−0.18	1.76E−03	1.34E−04	−3.81	15.44
XGBoost	**1.27E−03**	2.31E−04	1.93	20.23	1.76E−03	1.14E−04	−3.77	15.39
LSTM	1.03E−03	3.38E−04	1.22	0.37	**1.78E−03**	1.26E−04	−1.01	19.06
HistData	3.55E−04	**4.14E−05**	5.90	43.05	3.55E−04	**4.14E−05**	5.9	**43.05**

Expert Systems With Applications 235 (2024) 121102

Table B.12
Expected portfolio risk summary statistics. For reference, the perfect foresight values are 1.14E−03 (30 days), 2.42E−03 (60 days), 2.51E−03 (90 days), 2.58E−03 (120 days), and 2.34E−03 (150 days). Best value per column (for each algorithm) is shown in boldface.

30 days	Out-of-sample				One-day-ahead			
	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HES	8.24E−03	3.41E−05	6.79	45.67	2.66E−03	6.95E−04	3.43	12.68
TBATS	**1.21E−03**	2.75E−04	**8.17**	**72.01**	2.76E−03	1.26E−03	4.71	24.11
ARIMA	3.91E−03	4.27E−05	−6.72	60.62	8.29E−03	**5.99E−04**	−1.95	7.41
LR	1.86E−03	3.44E−05	−0.31	−1.47	1.80E−03	6.41E−04	3.74	16.71
SVR	6.16E−03	1.70E−03	0.93	0.35	1.79E−03	9.90E−04	**5.30**	**29.66**
KNN	3.13E−03	7.79E−05	−1.98	2.00	1.90E−03	1.12E−03	4.51	23.79
XGBoost	4.61E−03	9.17E−04	0.49	0.45	**1.57E−03**	8.05E−04	5.20	27.92
LSTM	3.39E−03	1.06E−03	−0.40	−1.47	1.85E−03	8.39E−04	5.19	27.54
HistData	3.03E−03	**7.57E−05**	3.94	20.31	3.47E−03	6.08E−04	3.42	13.26
60 days								
HES	4.51E−03	**6.24E−06**	**7.36**	**69.47**	4.96E−03	5.93E−04	4.99	26.20
TBATS	3.89E−03	7.39E−05	0.98	2.60	5.17E−03	**3.52E−04**	6.05	38.69
ARIMA	5.05E−03	2.37E−04	2.40	6.65	1.38E−02	1.17E−03	−3.44	11.89
LR	4.03E−03	3.89E−03	4.82	22.82	4.04E−03	1.38E−03	4.11	17.53
SVR	6.77E−03	2.05E−03	1.44	2.77	3.10E−03	7.58E−04	4.50	22.05
KNN	3.84E−03	4.49E−04	1.97	4.21	3.54E−03	7.27E−04	3.81	15.76
XGBoost	6.00E−03	1.95E−03	1.72	2.37	4.13E−03	8.27E−04	3.55	13.15
LSTM	5.37E−03	2.24E−04	2.61	10.55	**2.86E−03**	9.96E−04	1.96	2.79
HistData	**2.86E−03**	5.92E−05	4.46	23.99	4.52E−03	1.09E−03	**6.58**	**49.25**
90 days								
HES	5.89E−03	**2.41E−05**	4.44	19.44	5.92E−03	1.09E−03	4.99	27.52
TBATS	2.73E−03	4.62E−05	2.21	5.91	5.75E−03	5.45E−04	4.17	28.29
ARIMA	5.73E−03	2.92E−04	2.09	7.58	1.94E−02	1.65E−03	−3.17	9.52
LR	3.71E−03	3.81E−04	2.05	7.32	4.29E−03	7.32E−04	3.14	11.22
SVR	5.78E−03	1.15E−03	−0.54	0.74	4.57E−03	**5.03E−04**	4.40	35.30
KNN	1.00E−02	1.55E−03	−1.84	3.04	4.82E−03	1.09E−03	3.82	15.06
XGBoost	8.02E−03	1.97E−03	6.38	**55.74**	4.22E−03	1.37E−03	**5.47**	33.86
LSTM	6.98E−03	1.55E−03	−1.27	1.99	**4.20E−03**	5.59E−04	3.17	16.06
HistData	**2.65E−03**	1.00E−04	**6.85**	52.37	5.25E−03	6.86E−04	5.26	**41.50**
120 days								
HES	3.93E−03	**6.94E−18**	−1.36	−0.62	4.96E−03	8.69E−04	3.62	14.51
TBATS	**2.92E−03**	5.79E−05	−0.07	11.49	7.05E−03	4.32E−04	−1.28	15.96
ARIMA	5.42E−03	1.59E−04	0.19	−0.84	2.03E−02	1.75E−03	−3.89	15.08
LR	7.68E−03	1.39E−03	0.36	−1.53	3.89E−03	6.04E−04	3.59	15.33
SVR	7.32E−03	1.81E−03	−0.78	0.80	2.99E−03	9.70E−04	3.69	16.75
KNN	7.05E−03	4.35E−04	−3.26	12.28	3.02E−03	1.33E−03	4.30	25.96
XGBoost	6.52E−03	9.75E−04	3.38	17.98	**2.86E−03**	9.66E−04	2.98	8.12
LSTM	6.32E−03	9.92E−04	−0.35	−0.03	3.75E−03	6.15E−04	4.72	31.24
HistData	2.85E−03	1.56E−04	**7.80**	**66.41**	5.45E−03	**4.08E−04**	**6.07**	**47.25**
150 days								
HES	3.93E−03	**6.94E−18**	−1.36	−0.62	9.34E−03	8.19E−04	5.41	55.52
TBATS	**2.49E−03**	2.15E−04	**8.02**	**71.54**	9.26E−03	6.46E−04	−4.33	26.20
ARIMA	4.67E−03	1.88E−04	−7.00	58.73	3.07E−02	3.39E−03	−2.51	5.11
LR	5.49E−03	2.68E−04	−2.34	9.18	**5.02E−03**	6.39E−04	4.39	26.29
SVR	6.81E−03	1.20E−03	0.52	2.55	5.72E−03	5.65E−04	4.18	28.53
KNN	5.04E−03	2.06E−04	0.48	0.08	5.66E−03	6.71E−04	7.41	70.54
XGBoost	7.75E−03	1.34E−03	0.06	4.06	5.66E−03	**4.83E−04**	7.95	73.75
LSTM	5.07E−03	1.29E−03	1.04	0.74	5.89E−03	2.16E−03	**9.45**	**92.23**
HistData	2.76E−03	9.24E−05	2.05	5.68	5.13E−03	1.86E−03	5.00	30.51

Table B.13
Expected portfolio Sharpe ratio summary statistics. For reference, the perfect foresight values are 4.04E−02 (30 days), 3.72E−02 (60 days), 3.72E−02 (90 days), 3.29E−02 (120 days), and 3.23E−02 (150 days). Best value per column (for each algorithm) is shown in boldface.

30 days	Out-of-sample				One-day-ahead			
	Mean	SD	Skewness	Kurtosis	Mean	SD	Skewness	Kurtosis
HES	1.04E−02	**3.60E−06**	−2.33	11.07	1.75E−02	**3.05E−03**	−3.91	19.98
TBATS	4.91E−03	1.12E−03	6.83	49.75	1.72E−02	4.72E−03	3.40	**33.56**
ARIMA	1.05E−02	4.22E−04	**−9.31**	**91.19**	1.35E−02	4.67E−03	0.02	−0.07
LR	1.88E−02	1.86E−04	0.62	−1.07	3.12E−02	7.08E−03	−2.91	8.87
SVR	**2.55E−02**	2.69E−03	0.73	−1.05	**3.43E−02**	6.00E−03	−2.91	20.73
KNN	2.49E−02	1.06E−04	2.18	4.11	3.29E−02	6.30E−03	−3.27	9.95
XGBoost	2.23E−02	9.52E−03	0.34	−0.59	3.17E−02	4.36E−03	−4.21	18.83
LSTM	2.29E−02	1.72E−03	1.00	1.39	3.42E−02	5.65E−03	**−4.63**	22.79
HistData	1.83E−02	4.83E−03	−2.73	9.20	1.83E−02	4.83E−03	−2.73	9.20
60 days								
HES	5.11E−03	**3.84E−05**	2.94	**31.95**	9.99E−03	1.93E−03	6.42	**55.21**
TBATS	3.54E−03	4.58E−04	3.18	28.98	3.67E−03	1.65E−03	5.27	28.47
ARIMA	9.17E−03	8.21E−04	2.21	3.88	1.79E−02	1.54E−03	−2.30	18.00
LR	1.35E−02	3.37E−03	1.87	6.47	2.96E−02	3.66E−03	−2.97	9.76
SVR	1.81E−02	5.12E−03	1.55	2.16	2.66E−02	3.68E−03	**−3.95**	18.05
KNN	1.62E−02	5.44E−04	2.29	7.67	2.96E−02	4.07E−03	−3.79	16.15
XGBoost	**2.01E−02**	5.06E−03	0.42	1.05	**3.23E−02**	4.21E−03	−2.99	8.49
LSTM	1.69E−02	4.21E−04	6.80	60.02	2.69E−02	4.96E−03	−0.71	4.29
HistData	1.21E−02	1.40E−03	**−1.80**	21.04	1.21E−02	**1.40E−03**	−1.80	21.04
90 days								
HES	8.20E−03	**3.81E−05**	4.14	16.48	1.26E−02	2.05E−03	−2.80	15.94
TBATS	2.89E−03	2.54E−04	6.11	**47.25**	1.25E−02	1.85E−03	−4.93	**26.76**
ARIMA	4.91E−03	7.47E−04	2.80	7.84	1.36E−02	1.12E−03	0.46	22.37
LR	1.31E−02	2.66E−03	2.40	7.35	2.64E−02	3.10E−03	−3.15	10.96
SVR	**1.74E−02**	4.71E−03	−0.60	0.01	**2.80E−02**	2.64E−03	−4.22	18.00
KNN	1.67E−02	1.17E−03	−2.41	7.18	2.69E−02	4.65E−03	−2.80	7.19
XGBoost	1.57E−02	1.98E−03	−2.16	13.26	2.63E−02	1.80E−03	−3.77	17.19
LSTM	1.62E−02	5.58E−03	−1.05	0.51	2.65E−02	2.40E−03	**−4.85**	25.12
HistData	1.08E−02	8.27E−04	**−4.01**	17.80	1.08E−02	**8.27E−04**	−4.01	17.80
120 days								
HES	7.55E−03	**1.42E−17**	1.26	−1.46	7.98E−03	1.26E−03	1.27	21.04
TBATS	3.07E−03	2.14E−04	5.67	**44.64**	4.23E−03	1.50E−03	4.39	27.30
ARIMA	4.10E−03	3.37E−04	−0.07	−0.70	5.90E−03	**7.67E−04**	4.19	19.43
LR	1.26E−02	3.75E−03	1.08	−0.80	**2.36E−02**	1.95E−03	−5.10	41.21
SVR	1.60E−02	5.80E−03	−0.40	−1.39	2.07E−02	2.45E−03	−1.34	7.96
KNN	1.55E−02	1.12E−03	2.95	12.13	2.02E−02	3.86E−03	−0.84	5.29
XGBoost	1.49E−02	1.88E−03	**−3.60**	16.67	2.04E−02	2.67E−03	0.54	8.29
LSTM	**1.67E−02**	2.40E−03	−0.79	1.29	2.31E−02	2.01E−03	**−2.99**	11.44
HistData	1.03E−02	4.56E−04	−3.21	15.81	2.71E−04	9.52E−04	8.17	**76.41**
150 days								
HES	7.76E−03	**1.42E−17**	1.26	−1.46	1.43E−02	7.98E−04	1.94	22.08
TBATS	3.38E−03	1.97E−04	4.57	52.51	1.41E−02	8.73E−04	−4.74	23.67
ARIMA	3.88E−03	1.03E−03	−1.69	2.36	9.32E−03	**5.83E−04**	−0.83	16.33
LR	1.26E−02	4.93E−04	**−2.97**	17.70	2.12E−02	1.39E−03	−3.54	13.33
SVR	1.38E−02	2.79E−03	−0.30	2.18	2.30E−02	2.52E−03	−3.78	16.36
KNN	1.28E−02	3.84E−04	−0.03	−0.14	**2.32E−02**	1.65E−03	−3.57	13.34
XGBoost	**1.42E−02**	1.70E−03	0.80	11.74	**2.32E−02**	1.73E−03	**−4.75**	**24.98**
LSTM	1.40E−02	3.04E−03	0.73	−0.64	**2.32E−02**	1.63E−03	−3.30	11.59
HistData	1.30E−02	4.39E−04	−6.82	**61.77**	7.15E−03	1.58E−03	0.59	20.76

References

Adebiyi, S. O., Ogunbiyi, O. O., & Amole, B. B. (2022). Artificial intelligence model for building investment portfolio optimization mix using historical stock prices data. *Rajagiri Management Journal*, *16*(1), 36–62.

Adegboye, A., Kampouridis, M., & Otero, F. (2023). Algorithmic trading with directional changes. *Artificial Intelligence Review*, *56*(6), 5619–5644.

Akinsomi, O. (2020). How resilient are REITs to a pandemic? The COVID-19 effect. *Journal of Property Investment & Finance*.

Anderson, J., Anderson, R., Guirguis, H. S., Proppe, S., & Seiler, M. J. (2021). Time-varying correlations of REITs and implications for portfolio management. *Journal of Real Estate Research*, *43*(3), 317–334.

Baker, H. K., & Chinloy, P. (2014). *Public real estate markets and investments*. Oxford University Press.

Bhuyan, R., Kuhle, J., Ikromov, N., & Chiemeke, C. (2014). Optimal portfolio allocation among REITs, stocks, and long-term bonds: An empirical analysis of US financial markets. *Journal of Mathematical Finance*, *2014*.

Block, R. L. (2011). *Investing in REITs: Real estate investment trusts*. John Wiley & Sons.

Brabazon, A., Kampouridis, M., & O'Neill, M. (2020). Applications of genetic programming to finance and economics: past, present, future. *Genetic Programming and Evolvable Machines*, *21*(1), 33–53.

Brookhouse, J., Otero, F. E., & Kampouridis, M. (2014). Working with opencl to speed up a genetic programming financial forecasting algorithm: Initial results. (pp. 1117–1124).

Brown, R. G. (1956). Exponential smoothing for predicting demand. cambridge, mass., arthur d. little. In *Book exponential smoothing for predicting demand*. NBER Cambridge, MA, USA.

Butler, A., & Kwon, R. H. (2021). Integrating prediction in mean-variance portfolio optimization. arXiv preprint arXiv:2102.09287.

Chen, J.-H., Chang, T.-T., Ho, C.-R., & Diaz, J. F. (2014). Grey relational analysis and neural network forecasting of REIT returns. *Quantitative Finance*, *14*(11), 2033–2044.

Chen, Y.-C., Lee, C.-F., & Lee, A. C. (2019). The role of REITs in a mixed-asset portfolio: Evidence from international markets. *The Quarterly Review of Economics and Finance*, *73*, 192–205.

Chen, J., & Smith, J. (2019). Genetic algorithm-based portfolio optimization incorporating real estate investments. *Journal of Financial Optimization*, *30*(2), 127–145.

Chen, W., Zhang, H., Mehlawat, M. K., & Jia, L. (2021). Mean–variance portfolio optimization using machine learning-based stock price prediction. *Applied Soft Computing*, *100*, Article 106943.

Deb, K. (2011). *Multi-objective optimisation using evolutionary algorithms: An introduction*. Springer.

Delfim, J.-C., & Hoesli, M. (2019). Real estate in mixed-asset portfolios for various investment horizons. *The Journal of Portfolio Management*, *45*(7), 141–158.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, *7*, 1–30.

Freitas, F. D., De Souza, A. F., & De Almeida, A. R. (2009). Prediction-based portfolio optimization model using neural networks. *Neurocomputing*, *72*(10–12), 2155–2170.

Garcia, S., & Herrera, F. (2008). An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research*, *9*(12).

Geiger, P., Cajias, M., & Fuerst, F. (2016). A class of its own: the role of sustainable real estate in a multi-asset portfolio. *Journal of Sustainable Real Estate*, *8*(1), 190–218.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading: Addison.

Günther, R., Wills, N., & Piazolo, D. (2022). The role of real estate in a mixed-asset portfolio and the impact of illiquidity. *International Journal of Real Estate Studies*, *16*(2), 34–46.

Habbab, F. Z., & Kampouridis, M. (2022a). Machine learning for real estate time series prediction. In *2022 UK workshop on computational intelligence (UKCI)*. IEEE.

Habbab, F. Z., & Kampouridis, M. (2022b). Optimizing mixed-asset portfolios with real estate: why price predictions? In *2022 IEEE congress on evolutionary computation (CEC)* (pp. 1–8). IEEE.

Habbab, F. Z., Kampouridis, M., & Voudouris, A. A. (2022). Optimizing mixed-asset portfolios involving REITs. In *2022 IEEE symposium on computational intelligence for financial engineering and economics (CIFEr)* (pp. 1–8). IEEE.

Hocht, S., Ng, K. H., Wolf, J., & Zagst, R. (2008). Optimal portfolio allocation with Asian hedge funds and Asian REITs. *International Journal of Services Sciences*, *1*(1), 36–68.

Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, *20*(1), 5–10.

Jain, P. (2017). J-REIT market quality: impact of high frequency trading and the financial crisis. Available at SSRN 2972092.

Jain, S., Mandal, P., Singh, B., Kulkarni, P. V., & Sayed, M. (2021). Prediction of stock indices, gold index, and real estate index using deep neural networks. In *Cybernetics, cognition and machine learning applications* (pp. 333–339). Springer.

Jalil, R. A., Ali, H. M., Razali, N., & Yim, J. L. M. (2015). Optimal portfolio allocation of Malaysian real estate investment trusts during economic downturn. *International Journal Real Estate Studies*, *9*(2), 1–15.

Jayaraman, B. (2021). *Building wealth through REITs*. Marshall Cavendish International Asia Pte Ltd.

Jones, C. A., & Trevillion, E. (2022). Portfolio theory and property in a multi-asset portfolio. In *Real estate investment* (pp. 129–155). Springer.

Kalayci, C. B., Polat, O., & Akbay, M. A. (2020). An efficient hybrid metaheuristic algorithm for cardinality constrained portfolio optimization. *Swarm and Evolutionary Computation*, *54*, Article 100662.

Lee, S., & Moss, A. (2018). REIT asset allocation. In *The Routledge REITs research handbook* (pp. 139–152). Routledge.

Li, R. Y. M., Fong, S., & Chong, K. W. S. (2017). Forecasting the REITs and stock indices: group method of data handling neural network approach. *Pacific Rim Property Research Journal*, *23*(2), 123–160.

Li, X., Liang, J., Li, Y., & Liu, X. (2015). A survey of multi-objective portfolio optimization using evolutionary algorithms. *Journal of Computer Science*, *9*, 135–146.

Li, Y., & Stevenson, S. (2018). Optimal allocation of REITs in mixed-asset portfolios. *Journal of Real Estate Portfolio Management*, *24*(2), 113–128.

Li, M., & Wu, Y. (2021). Dynamic decision model of real estate investment portfolio based on wireless network communication and ant colony algorithm. *Wireless Communications and Mobile Computing*, *2021*, 1–14.

Lian, Y.-M., Li, C.-H., & Wei, Y.-H. (2021). Machine learning and time series models for VNQ market predictions. *Journal of Applied Finance and Banking*, *11*(5), 29–44.

Liu, H., Liu, D., & Xin, J. (2002). Real-time recognition of road traffic sign in motion image based on genetic algorithm. In *Proceedings. international conference on machine learning and cybernetics, Vol. 1* (pp. 83–86). IEEE.

Loo, W. K. (2019). Predictability of HK-REITs returns using artificial neural network. *Journal of Property Investment & Finance*.

Ma, Y., Han, R., & Wang, W. (2020). Prediction-based portfolio optimization models using deep neural networks. *IEEE Access*, *8*, 115393–115405.

Ma, Y., Han, R., & Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, *165*, Article 113973.

Marzuki, M. J., & Newell, G. (2019). The evolution of Belgium REITs. *Journal of Property Investment & Finance*, *37*(4), 345–362.

Michalewicz, Z. (2013). *Genetic algorithms+ data structures=evolution programs*. Springer Science & Business Media.

Mitchell, M. (1996). *An introduction to genetic algorithms*. MA: MIT Press.

Oppenheim, A. V., & Schafer, R. W. (1975). *Digital signal processing:(By) Alan V. Oppenheim (and) Ronald W. Schafer*. Prentice-Hall.

Parikh, H., & Zhang, W. (2019). The diversity of real assets: portfolio construction for institutional investors. *PGIM Institutional Advisory & Solutions-April*.

Pawar, K., Jalem, R. S., & Tiwari, V. (2019). Stock market price prediction using LSTM RNN. In *Emerging trends in expert applications and security* (pp. 493–503). Springer.

Pilusa, S., Niesing, P., & Zulch, B. (2022). 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* (pp. 109–118). CRC Press.

Razak, M. Z. (2023). The dynamic role of the Japanese property sector REITs in mixed-assets portfolio. *Journal of Property Investment & Finance*, *41*(2), 208–238.

Rehring, C. (2012). Real estate in a mixed-asset portfolio: the role of the investment horizon. *Real Estate Economics*, *40*(1), 65–95.

Sen, J., Dutta, A., & Mehtab, S. (2021). Stock portfolio optimization using a deep learning LSTM model. In *2021 IEEE Mysore sub section international conference (MysuruCon)* (pp. 263–271). IEEE.

Smith, J., & Johnson, R. (2021). Optimal REIT allocation in a mixed-asset portfolio. *Journal of Real Estate Finance and Economics*, *62*(3), 350–376.

Stephen, L., & Simon, S. (2005). The case for REITs in the mixed-asset portfolio in the short and long run. *Journal of Real Estate Portfolio Management*, *11*(1), 55–80.

Vrieze, S. I. (2012). Model selection and psychological theory: a discussion of the differences between the akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychological methods*, *17*(2), 228.

Wiklund, E., Flood, J. H., & Lunde, J. (2020). Why include real estate and especially REITs in multi-asset portfolios?

Yaman, I., & Dalkılıç, T. E. (2021). A hybrid approach to cardinality constraint portfolio selection problem based on nonlinear neural network and genetic algorithm. *Expert Systems with Applications*, *169*, Article 114517.

Yamaoka, K., Nakagawa, T., & Uno, T. (1978). Application of akaike's information criterion (AIC) in the evaluation of linear pharmacokinetic equations. *Journal of pharmacokinetics and biopharmaceutics*, *6*(2), 165–175.