Improving REITs Time Series Prediction Using ML and Technical Analysis Indicators

Fatim Z. Habbab, Michael Kampouridis, Tasos Papastylianou
School of Computer Science and Electronic Engineering
University of Essex
Wivenhoe Park, United Kingdom
{fh20175, mkampo, tasos.papastylianou}@essex.ac.uk

***Abstract*—One of the most popular ways to reduce the risk of an investment portfolio is by holding shares of Real Estate Investment Trusts (REITs), which own and manage real estate. An important aspect of this process is to be able to forecast future REITs prices, as this allows investors to achieve higher returns at lower risk. This paper examines the performance of five different machine learning algorithms in the task of REITs price forecasting: Ordinary Least Squares Linear Regression, Support Vector Regression, k-Nearest Neighbours Regression, Extreme Gradient Boosting, and Long/Short-Term Memory Neural Networks. In addition to past REITs prices, we also use Technical Analysis indicators to assist the algorithms in the task of price prediction. While such indicators are very popular in stocks forecasting, they have never been used to forecast REITs. Our experiments show that (i) all ML algorithms produce low error and standard deviation, and are able to outperform the well-known statistical benchmark of AutoRegressive Integrated Moving Average (ARIMA), and (ii) the introduction of Technical Analysis (TA) indicators into the feature set leads to an error reduction of up to 50%.**

***Index Terms*—mixed-asset portfolio, minimum variance, portfolio optimization, risk-adjusted return**

I. Introduction

The main reason why investors in financial markets select their investment weights in a portfolio is to maximize their return and/or to reduce the risk associated to their portfolio [1]. One of the most popular ways to optimize a portfolio, i.e., minimize the risk and/or maximize the return associated to that portfolio, is by investing in real estate [2, 3, 4]. An alternative to direct real estate investments (i.e. purchasing real estate assets in exchange for a given price) is investing in real estate indirectly, i.e. by purchasing shares in listed or non-listed companies that own and manage real estate. Some of these companies are known as REITs (real estate investment trusts) and are listed on major exchanges. Thanks to the existence of REITs, individual investors do not need to directly own or manage properties. Purchasing REITs shares can cost as little as $500,[1] which is much lower than the entry point of purchasing a property. This allows investors to access the benefits of owning real estate (e.g. steady income, diversification, etc.) without needing to spend a large amount of money.

Much work in the asset allocation literature has focused on the problem of optimizing mixed-asset portfolios including real estate [5, 6, 7, 8]. However, typically the focus lies on optimising asset weights, using prices from a training set. The main limitation of this approach is that prices in the test set might differ significantly from those in the training set, thus leading to poor portfolio performance [9]. This motivates us to include price predictions in the portfolio optimization problem rather than using historical data. This work thus attempts to predict the prices of REITs through five Machine Learning (ML) algorithms: Linear Regression, Support Vector Regression, k-Nearest Neighbours Regression, Extreme Gradient Boosting Regression, and a Long Short-Term Memory Recurrent Neural Network. We compare the predictive power of such algorithms against a well-known statistical benchmark, the AutoRegressive Integrated Moving Average (ARIMA) class of models.

Being able to predict REITs prices with low error is crucial because it might affect the future performance of a mixed-asset portfolio including REITs. In addition to past REITs prices, we also use technical analysis (TA) indicators — i.e., Moving Average (MA), Moving Average Convergence/Divergence (MACD), Bollinger Bands, Exponential Moving Average (EMA), and Momentum — to assist the algorithms in the task of price prediction. While such indicators are commonly used to predict stock prices [10, 11, 12], they have never been used to predict REIT prices. In this work, we compare the performance of ML algorithms that use past REITs prices as the main features in the forecasting problem, to the performance of ML algorithms that use both past REITs prices and TA indicators.

In addition to REITs, we also conduct experiments for other asset classes, namely stocks and bonds. As previously mentioned, while price prediction for stocks and bonds has previously been explored in the literature, it tends to be with one or two algorithms only. We thus believe that our investigation will offer a better understanding of the strengths and weaknesses that different ML algorithms can bring into this domain, and how they compare against financial benchmarks. In total, we experiment with 27 datasets, 9 from each class (REITs, stocks, bonds). Collectively, the datasets cover three different markets, namely the US, the UK, and Australia.

In summary, this paper has the following main contribu-

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

tions: (i) apply five popular ML algorithms to the problem of predicting REITs prices from historical data, and compare their performance against each other, as well as a conventional gold-standard (ARIMA); (ii) improve the predictive power of the ML algorithms by incorporating TA indicators as features.

The rest of this paper is organized as follows: Section II presents a brief background on REITs, the Modern Portfolio Theory, and briefly discusses related works; Section III outlines the methodology of this paper; Section IV presents our experimental setup; Section V provides a detailed discussion of the experimental results we obtained by applying machine learning and ARIMA to our data; finally, Section VI concludes the paper.

II. Background and Related Works

A. Real Estate Investment Trusts

REITs are companies that own, operate, or finance income-producing real estate. Examples of REITs include Realty Income Corporation (O), Digital Realty Trust, Inc (DLR), Simon Property Group, Inc (SPG), and so on. REITs provide the opportunity for everyday investors (not just banks and hedge funds) to benefit from real estate investment [13] by accessing dividend-based income and gaining from competitive returns without having the need to spend a large amount of money (as it happens with direct real estate investments).

Investing in REITs allows anyone to build investment portfolios as for the other financial markets, i.e., through the purchase of an individual company stock or through a mutual fund or exchange traded fund (ETF). An investor could get help from a broker, investment advisor or financial planner to set their financial goals and identify appropriate REIT investments. According to a 2020 Chatham Partners study conducted in the US[2], about 80% of financial advisors recommend REITs to their clients. It is also possible to invest in public non-listed REITs and private REITs.

Holding shares of REITs in a mixed-asset portfolio (i.e., a portfolio already made of other asset classes, such as stocks and bonds) allows an investor to reduce the overall risk level and/or increase the overall return level, thus improving the risk-adjusted performance of that portfolio. This is made possible by the low levels of correlation between REIT shares and other asset classes.

B. Modern Portfolio Theory

Modern portfolio theory (MPT) is a mathematical framework that is largely used to solve asset allocation problems. The main assumption of MPT is that investors are risk averse, in the sense that one would tend to favour portfolios with lower risk among comparable portfolios that provide the same expected return. Consequently, one will choose a riskier portfolio only if compensated by a higher expected return. Different investors have different preferences over such tradeoffs, based on their individual risk aversion levels.

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

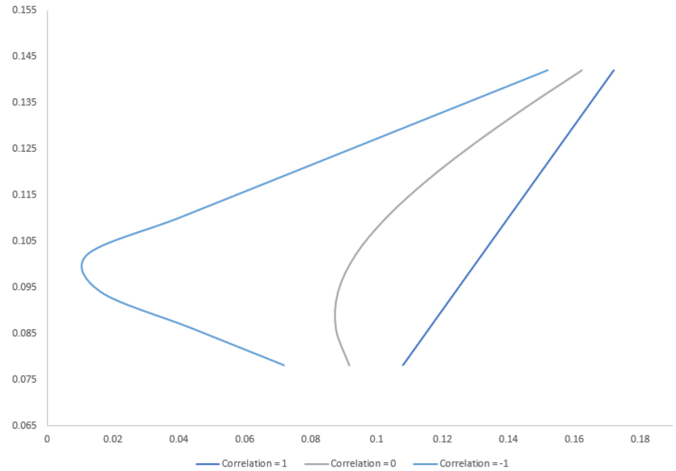

Fig. 1. Portfolio efficient frontier. The x-axis refers to the portfolio standard deviation, and the y-axis refers to the expected portfolio return. The two metrics improve as the correlation coefficient increases.

According to MPT, a portfolio is considered efficient when its expected return is maximized for a given level of risk, or its expected risk is minimized for a given level of return. The expected return of the portfolio is expressed as a weighted average of the historical returns of the assets included in the portfolio, where the weighting factors are the proportions allocated to the different asset classes. The expected risk of the portfolio is expressed as the variance of the historical returns of the asset classes, and is a function of the correlations ρ_{ij}, for all asset pairings (i, j). Given specific combinations of assets and standard deviations of asset returns, the highest possible standard deviation of portfolio returns is obtained when all correlations are equal to 1, which means that all asset pairs are perfectly correlated to each other. It is possible to reduce the portfolio's expected risk by selecting combinations of assets that are not perfectly positively correlated (i.e., $-1 < \rho_{ij} < 1$). This is known as *diversification*. If all asset pairs are perfectly uncorrelated ($\rho_{ij} = 0$ for all i, j), the variance of the portfolio returns is the sum of the squares of all asset weights times the asset's return variance. If all asset pairs are perfectly positively correlated ($\rho_{ij} = 1$ for all i, j), then the standard deviation of the portfolio returns is the sum of the standard deviations of the underlying asset returns, weighted by the proportion allocated to each asset class.

Figure 1 represents three efficient frontiers — i.e., combinations of optimal portfolios —, each one corresponding to a correlation coefficient. As we can observe, the portfolio standard deviation is lower when the correlation coefficient is -1, and tends to increase as the correlation goes to 0 and then 1, with a correlation of 1 determining the highest value of portfolio risk for any given expected return.

C. Related Works

In the current literature about REIT price prediction, there have been some works that applied Neural Network algorithms to predict stock and REIT prices, showing that these algorithms outperformed ARIMA in terms of prediction accuracy. In a similar way, [14] applied multivariate, ML-based regression algorithms (including Neural Networks) to predict REIT returns. Other authors compared ML algorithms to ARIMA for the prediction of REIT returns [15, 16, 17]. Such works focused mainly on artificial neural networks relying on multiple variables. In summary, while there have been a few works on REITs price prediction, the majority of them have focused on neural networks. In a more recent study, [18] used five Machine Learning algorithms to predict REIT prices, in addition to stock and bond prices. However, they adopted a one-step-ahead methodology, which may not be suitable for portfolio optimization purposes because it would imply rebuilding a portfolio on a daily basis according to the changes in price prediction. In addition, none of the above studies, or any other studies (to the best of our knowledge), have used technical analysis indicators as features to predict REITs prices.

The above limitations thus motivate us in this work to (i) apply five Machine Learning algorithms, (ii) use technical analysis indicators as features, and (iii) to perform period ahead predictions, rather than one-step-ahead.

III. Methodology

Before applying the machine learning algorithms, we first needed to take several data pre-processing steps, which are presented in Section III-A. We then present the features that we included in the price prediction in Section III-B and the loss function, which is the same across all algorithms, in Section III-C. Lastly, we briefly present the Python libraries we used to apply our machine learning algorithms in Section III-D.

A. Data preprocessing

Before being used for price prediction, each time series data is differenced and scaled. Differencing consists of calculating a one-step lag for each time point in such a way that $D_t = P_t - P_{t-1}$. For instance, price at time $t2$ will be transformed into $D_{t2} = P_{t2} - P_{t1}$. The differencing process makes the time series stationary, eliminating its upward trend and making the average constant over time. Stationarity is important in time series analysis as several models (including ARIMA) assume that data are independent of each other. Since market price time series often feature time dependency (i.e., each time point depends on the past ones), it is necessary to remove such a dependency in order to apply our prediction models.

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 Equations 1:

$$N_t = \frac{(D - D_{min})}{(D_{max} - D_{min})} \quad (1)$$

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 I
Example of time series differencing and scaling.

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

We present an example of the differencing and scaling processes in Table I, which presents sample data for the SPG time series reflecting the time period between 01 January 2021 and 30 January 2021. The first column presents the different time steps, the second column the price P_t of the given security, the third column the one-lag value of P_t, the fourth column the differenced D_t value, and the fifth column the scaled N_t variable. As we can observe, at time $t2$, D_t is equal to the difference between P_t and P_{t-1}, which is equal to $3.69 - 3.77 = -0.08$. Similarly, the first-order difference at $t3$ is equal to 0.01, and so on. The fifth column contains the D_t values after scaling, i.e., N_t. As we can observe the independent variable is scaled to the range between 0 and 1. For instance, after normalization at time step $t2$, the independent variable goes from -0.08 to 0.30.

B. Features

For our regression problem, we use two kinds of features: (i) past observations of a given time series N_t; and (ii) technical analysis (TA) indicators. Regarding the first type of features, given a time series N_t we are using past observations of N_t, i.e., N_{t-1}, N_{t-2}, N_{t-3}, ..., N_{t-n}, as features for our regression problem. The n value is decided on the basis of the Akaike Information Criteria (AIC) optimization. In other words, the number of lags corresponds to the optimal value for the p — i.e., autoregression — parameter in the ARIMA model. This process is explained in more detail in Section IV-C2. AIC is a metric widely used for model selection [19, 20, 21]. Each dataset has a different value for n, thus a different number of features.

In addition to past observations, we also use five technical indicators at each timepoint — Simple Moving Average (SMA); Exponential Moving Average (EMA); Moving Average Convergence/Divergence (MACD); Bollinger Bands; and Momentum — as suggested in [22, 23, 21]. These indicators help identify the short- and long-term trends of a time series, and thus can be effectively used for price prediction.

The Simple Moving Average (SMA) gives an estimate of the level of a time series, and thus is commonly used to forecast future observations [24]. The SMA is the weighted average of the past T prices; it can be represented mathematically as follows.

$$\text{SMA}(t) = \frac{\sum_{i=t-(T-1)}^{t} N_i}{T}, \quad (2)$$

where N_t is the normalized price at time i, and T is the number of time periods considered. We use the `rolling` method[3] to calculate the SMA in Python. Note that the period of interest T used for window-averaging purposes is unrelated to the number of lags n which determines the number of historical timepoints to be considered for training purposes.

The Exponential Moving Average (EMA) is similar to the SMA, in that it also represents a weighted average of past observations; however, unlike the SMA which considers a limited range of past observations with all observations given equal weights during the averaging process, the EMA instead considers *all* past observations, but with weights that become exponentially smaller the more distant a timepoint becomes. In other words, more recent observations contribute more than less recent ones during the averaging process. It is typically expressed via the following difference equation:

$$\text{EMA}(t) = \alpha N_t + (1-\alpha)\text{EMA}(t-1) \tag{3}$$

where α is a parameter expressing the amount of weight decay applied at each timestep. This is typically calculated as $\alpha = 2/(T+1)$, where T denotes the period of interest (i.e. such that the total contribution to the weighted average from all observations prior to that timepoint becomes trivial). It can take any real value between 0 and 1, where values closer to 0 ascribe more importance to past information, and 1 indicate that less importance is given to past prices. In Python, we use the `ewm` method[4] to calculate the EMA.

The Moving Average Convergence/Divergence (MACD) indicator measures the difference between a short-term EMA and a long-term EMA. This indicator can effectively be used to identify bullish — i.e., featured by a general increase in market prices — or bearish moments. Given an F-day and an L-day, where F-day and L-day refer to the first day and last day of the considered period respectively, the MACD is calculated as the difference between the L-day exponential moving average (i.e. the 'short-term' EMA) and the F-day moving average (i.e. the 'long-term' EMA) ([10]), as we can see below.

$$\text{MACD}(t) = \text{EMA}_L(t) - \text{EMA}_F(t) \tag{4}$$

Bollinger Bands (BB) refer to an interval around the SMA at time t, defined as follows: first, the standard deviation of all observations within a period of interest T is computed, where T is the same as that used to obtain the SMA; this is then multiplied by a modifier D which determines how many standard deviations away from the mean we want to define our range as. This is represented mathematically below.

$$\text{BB}(t) = \text{SMA}(t) \pm D\sqrt{\left(\frac{1}{T}\right)\sum_{i=t-(T-1)}^{t}\left[N_i - \text{SMA}(t)\right]^2} \tag{5}$$

[3]https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rolling.html Last access: January 2023.

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

TABLE II
EXAMPLE OF FEATURE SELECTION (LAGGED OBSERVATIONS).

t	N_t	N_{t-1}	N_{t-2}	N_{t-3}	N_{t-4}	N_{t-5}
t2	0.26	-	-	-	-	-
t3	0.59	0.26	-	-	-	-
t4	0.19	0.59	0.26	-	-	-
t5	0.85	0.19	0.59	0.26	-	-
t6	0.00	0.85	0.19	0.59	0.26	-
t7	0.59	0.00	0.85	0.19	0.59	0.26

TABLE III
EXAMPLE OF FEATURE SELECTION (TA INDICATORS).

t	SMA	EMA	MACD	Upper band	Lower band	Momentum
t2	-	0.26	0	-	-	-0.04
t3	-	0.51	-0.03	-	-	0.29
t4	0.35	0.28	0.02	-	-	-0.11
t5	0.54	0.67	-0.07	-	-	0.55
t6	0.35	0.22	0.06	0.52	0.18	-0.30
t7	0.48	0.47	-0.02	0.65	0.31	0.29

This indicator helps identify whether the current price level of a security has deviated considerably (i.e. more than D standard deviations) compared to its recent average, and predict when it might rise or fall back to that level.

Finally, the Momentum [25] is determined by the difference between each price and the initial price for a specific time period, as represented below.

$$\text{Momentum} = N_t - N_{t-T} \tag{6}$$

The Momentum measures the strength of a price trend. For this reason, it can be effectively used to predict the future direction of a time series.

Table II shows an example of lagged observations (with number of lags n = 5) and Table III shows an example of TA indicators for the preprocessed data presented in Table I. For this example, we calculate the 3-day SMA, the EMA with $\alpha = 0.5$, the MACD as the difference between the 6-day EMA and the 3-day EMA, the upper and lower band using the 3-day SMA and the standard deviation of the 3-day SMA multiplied by 0.5, and the Momentum as the rate of change of the N_t series. For our regression problem, we use these features together (i.e. $n+6$ features in total).

C. *Loss function*

The machine learning models used in this paper are evaluated by using out-of-sample predictions, rather than one-day-ahead predictions. The former is when today's N_t value ($t1$) is known and is used to forecast the value of tomorrow ($t2$). However, tomorrow's value is unknown and cannot be used to forecast the value two days ahead. Hence, this method uses the value forecast at time-step 1 to forecast the value at time-step 2, and so on. In the case of one-day-ahead forecasting, the price today (time-step 0) is known, and is used to forecast tomorrow's price (time-step 1). Then tomorrow's *real* price is used to forecast the price at time-step 2, and so on. This second method is expected to be more accurate, because we are using the actual values as features, instead of predictions. However, for portfolio optimization purposes using out-of-sample predictions would be more realistic as using one-day-ahead predictions would require rebalancing a portfolio on a daily basis for a time period of around 150 days which can lead to significant management costs.

For our problem, we use the *root mean square error* (RMSE) as the loss function, which is presented in Equation 7:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{|j|}(P_i - \hat{P}_i)^2}{|j|}}, \tag{7}$$

where P_t refers to the actual value of the price, $\hat{P}_t$ is its predicted value, and $|j|$ denotes the number of observations for each dataset j. Please note that as it was explained in Section III-A, the differenced and scaled values (i.e. D_t and N_t respectively) are reverted back to their original price values (i.e. P_t), so that the loss function can be calculated.

D. Machine learning algorithms

To apply our machine learning algorithms, we used the following python libraries: *sklearn*[5], *keras*[6], and *xbgoost*[7]. The functions used to fit the algorithm to the training data include `sklearn.linear_model.LinearRegression`, `sklearn.svm.SVR`, `xgboost`, `keras.Sequential`, and `sklearn.neighbors.KNeighborsRegressor`. The trainable parameters relating to these functions were determined using a grid search method, as described in Section IV. Once the algorithms were fit to the training data, they were then applied to the test set by using the `predict` attribute of the relevant model.

IV. Experimental setup

Our experiments aim to (i) provide evidence that including TA indicators as features in ML pipelines in addition to price lags can significantly reduce the prediction error rate, and (ii) demonstrate that the above approach can produce price predictions with lower error than ARIMA.

A. Data

For our experiments, we used daily prices downloaded from *Yahoo!Finance* for stocks and REITs, and *Investing.com* for bonds, referring to the period between January 2017 and January 2021, for financial instruments belonging to three asset classes (i.e., stocks, bonds, and real estate), and to three countries (i.e., US, UK, and Australia). For each of the three markets, we used prices for five stocks, five bonds, and five REITs. Thus, we ran our experiments on a total of 27 datasets. All prices were expressed as USD, so as to account for currency risk.

[5]https://scikit-learn.org/stable/index.html/ Last access: January 2023
[6]https://keras.io/getting_started/ Last access: January 2023
[7]https://xgboost.readthedocs.io/en/stable/ Last access: January 2023

B. Experimental parameters

The data was split into three sets: training (January 2019 - June 2020), validation (July 2020 - December 2020), and test (January 2021 - July 2021). The validation set was used to decide on the experimental parameters through a grid search tuning phase.

We performed grid search tuning for each data set to select the optimal parameters for the TA indicators described in Section III-B. The best value for α in the EMA calculation was selected from $[0.01, 0.05, 0.1]$ [26]. The other parameter values were decided on the basis of previous works [27, 28]. The selected values are shown in Table IV.

TABLE IV
TA PARAMETERS.

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

For the price prediction problem (ML algorithms), we performed tuning for each individual data set, thus each data set has its own tailored experimental parameters. The optimal parameters for the SVR, KNN, LSTM, and XGBoost model are selected using the Grid Search method in Python. Moreover, we did not perform parameter tuning for LR due to the absence of parameters to be tuned.

C. Benchmarks

1) Autoregression with ML: In Section III-B, we explained the different features used for our regression problem. To understand the potential improvement in the predictive accuracy from using TA indicators in addition to lagged values to predict asset prices, we compare the performance of the five ML algorithms that use lagged prices and TA indicators (proposed approach) against the five ML algorithms that only use lagged prices (i.e. without the TA indicators), as is a common practice in the REITs literature. The dependent variable is N_t, while the independent variables are past observations, i.e., $N_{t-1}, N_{t-2}, ..., N_{t-T}$ *without* the TA indicators.

2) ARIMA: The *Autoregressive Integrated Moving Average* (ARIMA) class of models is used to analyze a time series structure. It predicts the value of a variable (e.g., current market price) using the historical values of the same variable and its error distribution. It is used as a benchmark, as it is commonly used in finance with time series prediction.

Given a time series N_t, an ARIMA model of order (p, d, q) contains three components, the autoregression model of order p, differencing of order d, and the moving average model of order q. Equation 8 shows the mathematical form of ARIMA [29].

$$N_t = c + \sum_{i=1}^{p} \phi_i N_{t-i} + \epsilon_t + \sum_{i=0}^{q} \theta_i \epsilon_{t-1} \tag{8}$$

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

In order to fit the ARIMA model for each of the training datasets, we select the p, d, and q order based on the Akaike Information Criterion (or AIC) criterion, as previously mentioned in Section III-A. In other words, the best ARIMA model is found through a search of the minimum AIC value [30]. This indicator measures the quality of a statistical model with respect to a given set of data, by taking the log-likelihood of the maximum likelihood estimate of the model and the number of model parameters into account. It is commonly used to compare different ARIMA models [31, 32].

It is worth noting that being a univariate time series analysis tool, ARIMA does not include TA indicators to predict prices. This is why we will not present the implications of including TA for ARIMA.

V. Results

In this section, we analyze the predictive power of five ML algorithms that use TA indicators as additional features, and compare it with the predictive power of ML algorithms and ARIMA that only use lagged values as features (Section V-A). In addition, we analyze the importance of each feature using Shapley values (Section V-B). In the final part of this section, we examine the computational times for the algorithms used (Section V-C) and offer an overall discussion on the experimental results (Section V-D).

A. RMSE

In this section, we evaluate the prediction accuracy of ML algorithms that use TA indicators against autoregression algorithms. For each algorithm, we reported the mean and standard deviation of the RMSE distributions.

Table V shows the RMSE summary statistics for REITs. In general, we observe a 50% reduction in error when using ML with TA indicators. For instance, we can observe that the average RMSE obtained from the SVR algorithm is around 16.04 when we do not use TA and around 8.05 when we use TA. In terms of volatility, we notice an improvement in the RMSE distributions obtained from ML that use TA compared to the benchmarks. For instance, we can observe that the RMSE standard deviation is around 2.43 for the KNN algorithm when we use TA and 4.87 when we do not use TA. Moreover, we observe a large improvement from ML algorithms to ARIMA: the average RMSE is 66.80 and the standard deviation is 114.73.

Similarly to what observed for REITs, when predicting stocks the average RMSE value for ML algorithms that use TA tends to be around half that obtained from ML algorithms that do not use TA. For instance, we observe that the average RMSE for the KNN algorithm is 9.22 when we use TA and 18.58 when we do not use TA. The volatility values are also favourable for the ML algorithms including TA. For instance, the RMSE standard deviation for LR is 2.63 when including TA and 5.69 when not including TA. Furthermore, the values observed for the ML algorithms show an improvement in their predictive power with respect to ARIMA in the case of stocks. The average RMSE obtained from ARIMA is around 86.49, while its standard deviation is 73.20.

Finally, we also analyze the predictive power of ML algorithms and ARIMA for bonds. As we can observe, there is a large improvement from the ML algorithms that do not use TA to ML algorithms that use TA. For example, the average RMSE value for the KNN algorithm is 5.36 when using TA and 10.42 when not using TA. In addition, the RMSE distributions appear to be less volatile when using TA. For instance, the RMSE standard deviation for LSTM is 3.17 when using TA and 5.83 when not using TA. The results for ARIMA show a higher average error rate with respect to all ML algorithms (18.72) and a higher volatility (26.97).

In order to determine the statistical difference between the RMSE distributions obtained from the ML algorithms using TA indicators as additional features and the RMSE distributions obtained from algorithms using lagged values only, we performed a Kolmogorov-Smirnov (KS) test at the 5% significance level. The null hypothesis is that the compared RMSE distributions come from the same continuous distribution. Since we are making three comparisons (one for each asset class), we adjust the alpha value according to a Bonferroni's correction, i.e., $0.05/3 = 0.0167$. The KS test p-value for each one of the three tests for REITs, stocks, and bonds is 7.08E-16, 2.08E-13, 3.41E-14, respectively. As we can observe, all of the above p-values are lower than the 5% significance level (adjusted α value: 0.0167), which strongly suggests that the introduction of TA indicators results in a clear reduction in the RMSE observed.

In conclusion, we observed that the use of TA can lead to a significant improvement in the predictive power of ML algorithms as demonstrated by KS test results. In particular, there was a reduction in the average RMSE and its volatility for ML algorithms that use TA compared to those that do not use TA and ARIMA in the magnitude of 50%. Moreover, we noticed the lowest RMSE values for bonds, followed by REITs and stocks. This can be explained by the lower risk that characterizes bond prices (see Section IV-A).

B. Shapley values

In the previous section, we have seen that introducing TA indicators as additional features for our regression problem can significantly reduce the error rate, and thus improve the portfolio performance. In this part, we will analyze the relative importance of the various features by using Shapley values, which is a commonly used tool for model explainability [33, 34]. Table VI reports the simple average of the Shapley values calculated on the training set for each feature, shown for each asset class. In the case of both REITs and stocks, Momentum is the most relevant feature, followed by the 5-day SMA. In the case of REITs, the Momentum's average Shapley value, 3.99E-04, is almost 100% higher than the 5-day SMA's Shapley value, 2.01E-04. In the case of bonds, the momentum's Shapley value, 5.15E-04, is around 200%

TABLE V
RMSE SUMMARY STATISTICS FOR THE THREE ASSET CLASSES.

Algorithm	REITs				Stocks				Bonds			
	Without TA		With TA		Without TA		With TA		Without TA		With TA	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
LR	16.07	4.87	**8.41**	**2.22**	18.45	5.69	**10.38**	**2.63**	10.39	6.40	**7.70**	**6.26**
SVR	16.04	4.87	**8.05**	**2.44**	18.45	5.67	**9.24**	**2.86**	10.42	6.39	**5.36**	**3.14**
KNN	16.35	4.91	**8.03**	**2.43**	18.58	5.65	**9.22**	**2.86**	10.75	6.17	**5.19**	**3.19**
XGB	16.06	4.88	**8.04**	**2.44**	18.45	5.69	**9.50**	**3.04**	10.40	6.40	**5.19**	**3.20**
LSTM	16.16	4.94	**8.22**	**2.34**	18.48	5.67	**9.42**	**2.88**	12.21	5.83	**5.40**	**3.17**
ARIMA	66.80	114.73	-	-	86.49	73.20	-	-	18.72	26.97	-	-

higher than the 5-day SMA, 1.64E-04. In the case of bonds, the most important feature is the 5-day SMA, followed by the momentum. In particular, the Shapley value for the 5-day SMA, 6.06E-04, is almost 40% higher than the Shapley value for the momentum, 4.38E-04. In general, the momentum and 5-day moving average seem to have higher relevance with respect to the other features. This can be related to their ability to explain the future trend of a security in a better way than the prices of the previous days. The lagged prices ($N_{t-1}...N_{t-T}$) seem to have the lowest relevance among the different variables. This might explain the large improvement achieved by ML with TA in terms of RMSE (see Section V-A). In the current literature, lagged observations are commonly used for financial forecasting [35, 36].

TABLE VI
SHAPLEY AVERAGE FOR EACH FEATURE AND ASSET CLASS

	REITs	Stocks	Bonds
Price N-1	8.50E-06	2.31E-05	1.11E-05
Price N-2	4.06E-06	4.68E-06	-7.36E-06
Price N-3	6.01E-06	-1.49E-06	-2.19E-06
Price N-4	4.66E-07	3.30E-07	2.62E-06
Price N-5	6.08E-07	4.45E-07	6.64E-07
ma5	2.01E-04	1.64E-04	**6.06E-04**
ma20	9.62E-06	1.08E-04	2.54E-05
25ema	4.96E-05	5.51E-05	-4.37E-06
10ema	7.02E-06	8.78E-06	2.30E-05
MACD	1.35E-04	3.91E-05	8.05E-05
20sd	5.42E-05	5.76E-05	7.95E-05
upper band	-2.49E-06	6.75E-06	-2.77E-05
lower band	2.74E-06	3.90E-05	3.21E-05
ema	1.59E-04	-5.84E-06	1.30E-04
momentum	**3.99E-04**	**5.15E-04**	4.38E-04

C. Computational times

The computational times for the majority of algorithms are similar. On average, ARIMA took 0.168 minutes to run, while LR, SVR and KNN took between 0.2 and 0.3 minutes. LSTM was the most computationally expensive at 1.818 minutes. But this difference in runtime is not of concern, as usually such algorithms are ran offline, and only their models are run in real time. Besides, such algorithms' computational times can be reduced by parallelization processes [37].

D. Discussion

Our experiments aimed at demonstrating that adding TA indicators as features in the regression algorithms can increase the predictive power of our ML algorithms. We compared the RMSE results for ML algorithms using TA against the results obtained from ML algorithms using lagged prices only; we also compared the ML algorithms with ARIMA. We observed an improvement in the ability to predict out-of-sample asset prices when adding TA indicators, which is close to a 50% reduction in error. The lowest RMSE values were observed for the bond asset class given its lower volatility compared to the other asset classes. Moreover, we noticed that KNN and XGB were the algorithms with the highest predictive power among the others in the set.

Finally, we analysed the Shapley values for the different features used in our regression problem. The most relevant features were the Momentum, and 5-day Simple Moving Average for all the asset classes considered. On the other hand, we noticed that the lagged prices (used in the benchmark algorithms as unique features) had the lowest relevance in predicting the final prices.

VI. CONCLUSION

In this study, we focused on the problem of predicting out-of-sample prices of REITs, stocks and bonds by using five ML algorithms and Technical Analysis indicators. Our experimental analysis indicates that adding TA indicators in predicting prices increases the predictive power of ML algorithms in this setting. This might be explained by the lower feature importance observed for the lagged prices compared to the other features. On the other side, the most important features appear to be the Momentum and the 5-day Simple Moving Average.

Further research can be done on adding other features (e.g., fundamental analysis) to increase the accuracy of the predictive models, and thus improve the risk-adjusted portfolio performance of a multi-asset portfolio. Another opportunity for future research can be to use more algorithms to predict real estate prices.

REFERENCES

[1] A. Brabazon, M. Kampouridis, and M. O'Neill, "Applications of genetic programming to finance and economics: past, present, future," *Genetic Programming and Evolvable Machines*, vol. 21, no. 1, pp. 33–53, 2020.

[2] O. Akinsomi, "How resilient are reits to a pandemic? the covid-19 effect," *Journal of Property Investment & Finance*, 2020.

[3] B. Jayaraman, *Building Wealth Through REITS*. Marshall Cavendish International Asia Pte Ltd, 2021.

[4] P. Jain, "J-reit market quality: impact of high frequency trading and the financial crisis," *Available at SSRN 2972092*, 2017.

[5] M. Andrew and M. Glenn, "Public and private real estate in a mixed-asset portfolio," *Journal of Real Estate Portfolio Management*, vol. 9, no. 3, pp. 193–203, 2003.
[6] J. R. Webb, R. J. Curcio, and J. H. Rubens, "Diversification gains from including real estate in mixed-asset portfolios," *Decision Sciences*, vol. 19, no. 2, pp. 434–452, 1988.
[7] L. Stephen, "The return due to diversification of real estate to the us mixed-asset portfolio," *Journal of Real Estate Portfolio Management*, vol. 11, no. 1, pp. 19–28, 2005.
[8] F. Habbab, M. Kampouridis, and A. Voudouris, "Optimizing mixed-asset portfolios involving reits," in *IEEE Symposium on Computational Intelligence for Financial Engineering & Risk (CIFEr)*, 2022.
[9] F. Z. Habbab and M. Kampouridis, "Optimizing mixed-asset portfolio with real estate: Why price prediction?" in *IEEE Congress on Evolutionary Computation (CEC)*, 2022.
[10] P. N. V. Martins, "Technical analysis in the foreign exchange market: the case of the macd (moving average convergence divergence) indicator," Master's thesis, School of Economics and Management. University of Porto, 2017.
[11] Y. Hari and L. P. Dewi, "Forecasting system approach for stock trading with relative strength index and moving average indicator," Ph.D. dissertation, Petra Christian University, 2018.
[12] J. M.-J. Leung and T. T.-L. Chong, "An empirical comparison of moving average envelopes and bollinger bands," *Applied Economics Letters*, vol. 10, no. 6, pp. 339–341, 2003.
[13] R. L. Block, *Investing in REITs: real estate investment trusts*. John Wiley & Sons, 2011.
[14] J.-H. Chen, T.-T. Chang, C.-R. Ho, and J. F. Diaz, "Grey relational analysis and neural network forecasting of reit returns," *Quantitative Finance*, vol. 14, no. 11, pp. 2033–2044, 2014.
[15] Y.-M. Lian, C.-H. Li, and Y.-H. Wei, "Machine learning and time series models for vnq market predictions," *Journal of Applied Finance and Banking*, vol. 11, no. 5, pp. 29–44, 2021.
[16] S. Jain, P. Mandal, B. Singh, P. V. Kulkarni, and M. Sayed, "Prediction of stock indices, gold index, and real estate index using deep neural networks," in *Cybernetics, Cognition and Machine Learning Applications*. Springer, 2021, pp. 333–339.
[17] W. K. Loo, "Predictability of hk-reits returns using artificial neural network," *Journal of Property Investment & Finance*, 2019.
[18] F. Z. Habbab and M. Kampouridis, "Machine learning for real estate time series prediction," in *The 21st UK Workshop on Computational Intelligence (UKCI)*, 2022.
[19] S. I. Vrieze, "Model selection and psychological theory: a discussion of the differences between the akaike information criterion (aic) and the bayesian information criterion (bic)." *Psychological methods*, vol. 17, no. 2, p. 228, 2012.
[20] K. Yamaoka, T. Nakagawa, and T. Uno, "Application of akaike's information criterion (aic) in the evaluation of linear pharmacokinetic equations," *Journal of pharmacokinetics and biopharmaceutics*, vol. 6, no. 2, pp. 165–175, 1978.
[21] T. W. Khairi, R. M. Zaki, and W. A. Mahmood, "Stock price prediction using technical, fundamental and news based approach," in *2019 2Nd scientific conference of computer sciences (SCCS)*. IEEE, 2019, pp. 177–181.
[22] P. Oncharoen and P. Vateekul, "Deep learning for stock market prediction using event embedding and technical indicators," in *2018 5th international conference on advanced informatics: concept theory and applications (ICAICTA)*. IEEE, 2018, pp. 19–24.
[23] M. Agrawal, A. U. Khan, and P. K. Shukla, "Stock indices price prediction based on technical indicators using deep learning model," *International Journal on Emerging Technologies*, vol. 10, no. 2, pp. 186–194, 2019.
[24] S. Dinesh, N. Rao, S. Anusha, and R. Samhitha, "Prediction of trends in stock market using moving averages and machine learning," in *2021 6th International Conference for Convergence in Technology (I2CT)*. IEEE, 2021, pp. 1–5.
[25] R. Rosillo, D. De la Fuente, and J. A. L. Brugos, "Technical analysis and the spanish stock exchange: testing the rsi, macd, momentum and stochastic rules using spanish market companies," *Applied Economics*, vol. 45, no. 12, pp. 1541–1550, 2013.
[26] M. A. Maricar, "Analisa perbandingan nilai akurasi moving average dan exponential smoothing untuk sistem peramalan pendapatan pada perusahaan xyz," *Jurnal Sistem dan Informatika (JSI)*, vol. 13, no. 2, pp. 36–45, 2019.
[27] N. H. Hung, "Various moving average convergence divergence trading strategies: A comparison," *Investment management and financial innovations*, no. 13, Iss. 2 (contin. 2), pp. 363–369, 2016.
[28] J. Bollinger, "Using bollinger bands," *Stocks & Commodities*, vol. 10, no. 2, pp. 47–51, 1992.
[29] B. K. Meher, I. T. Hawaldar, C. M. Spulbar, and F. R. Birau, "Forecasting stock market prices using mixed arima model: A case study of indian pharmaceutical companies," *Investment Management and Financial Innovations*, vol. 18, no. 1, pp. 42–54, 2021.
[30] J. Mohamed, "Time series modeling and forecasting of somaliland consumer price index: a comparison of arima and regression with arima errors," *American Journal of Theoretical and Applied Statistics*, vol. 9, no. 4, pp. 143–53, 2020.
[31] R. Biswas, B. Bhattacharyya *et al.*, "Arima modeling to forecast area and production of rice in west bengal," *Journal of crop and weed*, vol. 9, no. 2, pp. 26–31, 2013.
[32] E. Dhamo and L. Puka, "Using the r-package to forecast time series: Arima models and application," in *INTERNATIONAL CONFERENCE Economic & Social Challenges and Problems*, 2010.
[33] M. Sundararajan and A. Najmi, "The many shapley values for model explanation," in *International conference on machine learning*. PMLR, 2020, pp. 9269–9278.
[34] D. Fryer, I. Strümke, and H. Nguyen, "Shapley values for feature selection: the good, the bad, and the axioms," *IEEE Access*, vol. 9, pp. 144 352–144 360, 2021.
[35] S. Mehtab and J. Sen, "Stock price prediction using cnn and lstm-based deep learning models," in *2020 International Conference on Decision Aid Sciences and Application (DASA)*. IEEE, 2020, pp. 447–453.
[36] J. Sen and S. Mehtab, "Accurate stock price forecasting using robust and optimized deep learning models," in *2021 International Conference on Intelligent Technologies (CONIT)*. IEEE, 2021, pp. 1–9.
[37] J. Brookhouse, F. E. Otero, and M. Kampouridis, "Working with OpenCL to speed up a genetic programming financial forecasting algorithm: Initial results," in *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 2014, pp. 1117–1124.