Research Article

Rojen Erik Sürek and Wee-Yeap Lau*

A refined methodological approach: Long-term stock market forecasting with XGBoost

https://doi.org/10.1515/jisys-2025-0027
received February 15, 2025; accepted May 08, 2025

Abstract: One critical research gap that this study fills is artificial intelligence (AI) and machine learning applications that predict equity market index total returns using long-term prediction horizons, and by experimenting with Extreme Gradient Boosting (XGBoost). The presented models achieved significantly higher accuracy rates than the majority class rate, and they obtained better predictive scores in all metrics than the logistic regression model. The best-performing model had a 100% accuracy rate when negative returns were predicted with a *p*-value of 0.05121. The evidence from this study suggests that XGBoost, a neglected algorithm in the literature, can enable more empirically informed long-term portfolio management decisions regarding overall equity exposure. Moreover, a literature contribution of this study is a refined methodological approach for prospective studies when implementing binary classifiers of prospective stock market returns for enhanced real-life economic utility for investors. The constructed models generated probabilities of whether the S&P 500 will have positive or negative total returns, including dividend payouts, in the subsequent 12 months. The predictive metrics of these models were evaluated against traditional logit models and whether the accuracy rates statistically significantly exceeded the majority class rate.

Keywords: computational finance, quantitative finance, AI and machine learning, stock market return, predictive model

JEL Classification: G17, C53, C49

1 Introduction

In the last decade, the field of statistical analysis and predictive modelling has undergone positive developments in terms of more sophisticated algorithms with improved ability to identify patterns between various data variables, greater data availability, and more computing power that can process large data volumes [1–3]. Thus, these new tools can enable us to acquire a greater understanding of various phenomena and relationships in the financial market compared with the application of more traditional statistical regression modelling.

However, despite these advancements, most machine learning studies in financial forecasting focus on short-term prediction horizons, leaving long-term forecasting largely underexplored. This gap is significant because long-term investment strategies are widely used by institutional, private, and government stakeholders [4]. Additionally, prior research has primarily evaluated predictive models based on price direction [5], rather than total return including dividends, which better reflects real investment outcomes. Another

*** Corresponding author: Wee-Yeap Lau,** Faculty of Business and Economics, Universiti Malaya, 50603 Kuala Lumpur, Malaysia; Malaysia-Japan Research Centre, Universiti Malaya, 50603 Kuala Lumpur, Malaysia, e-mail: wylau@um.edu.my
Rojen Erik Sürek: Faculty of Business and Economics, Universiti Malaya, 50603 Kuala Lumpur, Malaysia, e-mail: s2006688@siswa.um.edu.my, rojen.erik.surek@gmail.com
ORCID: Rojen Erik Sürek 0009-0001-1542-0353; Wee-Yeap Lau 0000-0002-3447-9895

methodological limitation is the common use of a 50% random threshold as a benchmark for accuracy, which is not rigorous as it does not account for the historical upward bias in stock returns [6]. Furthermore, XGBoost has consistently demonstrated strong predictive performance in real-world applications beyond academia [7–9]; meanwhile, it remains significantly underutilized in financial research [5]. Given its potential advantages and limited application in financial predictions, there is a need for further investigation into its effectiveness in this context.

Without robust predictive tools tailored for real-world investment applications and long-term horizons, investors may risk inefficient capital allocation and subpar returns. Given the high stakes of financial decision-making for institutional, private, and government stakeholders, it is crucial to address these deficiencies to enhance long-term forecasting accuracy and inform better investment strategies.

Thus, the aim of this study is to develop and propose a refined and more robust methodological framework when implementing predictive machine learning models of stock returns, with the goal of enhancing real-world value for practitioners. Importantly, the objective of this study is not to perform an exhaustive comparative analysis of the predictability of various algorithms or simulate and assess trading returns; rather, this study serves as a foundational methodological template for future research in financial forecasting.

Kumbure et al. [5] conducted a literature review of machine learning studies that predicted prices or returns on the stock market. They examined 138 journal articles published from 2000 onwards. Nazareth and Reddy [10] also conducted a literature review in the same subject area of 126 research articles published from 2015 onwards. As can be understood from these literature reviews, studies have already demonstrated the added value of various machine learning methods relative to traditional approaches with regard to higher predictability of financial market outcomes or superior returns given an investment strategy that follows machine learning predictions. However, the literature review by Kumbure et al. [5] conveys, as detailed earlier, that a significant majority of existing studies have constructed and tested machine learning models with short-term prediction horizons, such as daily, monthly, or intraday.

Therefore, one neglected area of research that this study will cover is the implementation and evaluation of machine learning models, with a long-term investment horizon of more than days or months, that predict the total return of the stock market index. Thus, the experimental set-up of this study can provide valuable knowledge for various stakeholders who apply more long-term and relatively passive portfolio management instead of short-term trading operations. That is, the study insights could add value by increasing the probability for various industry practitioners to generate higher long-term returns.

Two other aspects, which are common shortcomings in the existing literature [5], as previously mentioned, that this study addresses are (a) the binary classification refers to the future total return instead of price direction, and (b) the out-of-sample accuracy rates of the models are compared against the majority class rate, instead of a random (50%) threshold. The reason why it is suboptimal to predict the price direction is that it does not include the entire real-life monetary profit, which also includes the dividend payout. For example, the price direction can be negative, but the total return is positive. The reason why it is not rigorous to use a random threshold (50%) as a yardstick when evaluating accuracy rates of models that predict the stock market is that the well-known *a priori* distribution of the stock market's excess return, relative to cash, Treasury bills, and bonds, is imbalanced. That is, historically, stocks tend to outperform by a good margin. Thus, for example, if one views a passive buy-and-hold strategy as a prediction, it receives an accuracy rate that corresponds to the majority class rate. Hence, the latter is a better yardstick for evaluating the predictive performance of any given model.

Moreover, empirical evidence indicates that passive buy-and-hold strategies for the stock market index, irrespective of future outcome forecasts, can generate relatively attractive returns over time when compared to more active investment strategies and actively managed funds [11,12]. Thus, since the majority class rate can be viewed as the indirect predictive accuracy rate of a buy-and-hold strategy, this is a competitive and appropriate benchmark.

The reasons for the choice of implementing XGBoost are twofold. First, the most commonly applied algorithm categories in studies that experimented with artificial intelligence (AI) and machine learning models of financial market behaviour are various types of Artificial Neural Networks (ANN) and Support Vector Machines (SVM), while experimentations with gradient boosting techniques, such as XGBoost, are relatively neglected [5]. If one takes the full systematic literature review of Kumbure et al. [5] into account, including the

more recent studies after 2019, the review encompassed a total of 150 study articles. Out of these 150, 4 studies tested XGBoost. Second, XGBoost tends to belong to the top performers in terms of predictive performance metrics in prediction competitions [7–9]. Also, these tree-boosting systems have been empirically shown to result in error rates better than SVM and comparable to deep learning algorithms [13]. Despite this proven competitive predictive ability, XGBoost is a relatively neglected and inadequately tested algorithm in the financial literature, as previously detailed.

This study undertakes the following key tasks. First, XGBoost models employing a binary gradient boosting algorithm are constructed to generate probabilistic binary predictions of whether the Standard & Poor's 500 Index (S&P 500) will have a positive or negative total return over a 12-month horizon. Second, these models are evaluated on out-of-sample test data and benchmarked against a traditional logistic regression model using performance metrics such as accuracy rate, recall, precision, negative predictive value (NPV), balanced accuracy, and specificity. Third, model accuracy is assessed relative to the no-information rate (majority class rate) and tested for statistical significance using the binomial test. Finally, the study discusses the practical implications of its findings for portfolio and investment management; specifically, how predictive AI models can be leveraged to increase the probability of obtaining higher expected and actual returns over the long term.

This study makes several key contributions. First, it addresses the underexplored area of long-term stock market return prediction by implementing machine learning models with a 1-year prediction horizon. Second, it establishes a more rigorous and practically relevant methodological framework for applying machine learning in financial forecasting. This framework includes four key methodological enhancements: (a) using total return, including dividends, as the target variable rather than price alone; (b) benchmarking model accuracy against the majority class rate through binomial significance testing, rather than the conventional 50% random threshold; (c) demonstrating the predictive power of XGBoost in financial forecasting, an algorithm recognized for its strong performance in various fields but underutilized in financial research; and (d) incorporating controlled experimentation with increased classification threshold for predicting negative return to optimally account for the *a priori* probability distribution.

Thus, the research questions for this study were as follows:

- Can an AI model, which deploys a binary gradient boosting algorithm, result in better predictive ability of the 1-year forward total returns of the US market-capitalized equity index, which will be the S&P 500 Index, relative to a more standard logistic regression model?
- Can the long-term AI model, described in the previous bullet point, generate an accuracy rate that significantly exceeds the majority class rate?

The remainder of this article is structured as follows. Section 2 reviews the literature on forecasting the aggregate US stock market using machine learning, identifying key gaps that this study addresses. Section 3 describes the research design, data, sources, preprocessing steps, sampling method, and model implementation. It details the XGBoost framework and the parameter configurations, along with the out-of-sample testing procedure. Section 4 presents the empirical results, evaluating model performance across multiple evaluation metrics. Section 5 discusses the practical implications of the results for investment decision-making and compares the findings with the state-of-the-art models of existing literature. Finally, Section 6 concludes the study by summarizing key insights, highlighting contributions, and suggesting directions for future research.

2 Literature review

This section reviews related studies that employ machine learning techniques to predict the return or price of aggregate stock market indices, in line with the focus of this study. While some of the examined studies forecast cross-sectional returns, they predict all constituents of specific stock market indices, which they use as benchmarks when evaluating their stock selection strategy returns.

Many studies have leveraged various machine learning algorithms to predict and trade the aggregate US stock market index, specifically the S&P 500, and demonstrated the added predictive and economic value of

machine learning. For example, Gu et al. [14] showed that market timing the S&P 500 with monthly Neural Network predictions gets an annualized Sharpe ratio of 0.77 versus 0.51 of a buy-and-hold strategy. Similarly, Enke and Thawornwong [15] leveraged the power of Neural Networks to predict stock market returns. They employed various Neural Network models, including generalized regression, probabilistic, and multi-layer feed-forward models, to forecast the value and direction of excess stock returns on the S&P 500 stock index portfolio. Their findings indicated that machine learning-based trading strategies produced superior risk-adjusted profits, with Neural Network models outperforming the buy-and-hold strategy in terms of risk-adjusted returns.

Furthermore, Zhong and Enke [16–18] showed that principal component analysis (PCA) combined with ANN outperforms logistic regression in predicting daily S&P 500 returns. They also demonstrated that ANN-based strategies yielded better returns than the buy-and-hold strategy. Lahmiri [19] integrated ANN, intrinsic mode functions from empirical mode decomposition (EMD), and genetic algorithms to forecast S&P 500 intra-day price direction. The system demonstrated superior predictive performance compared to more traditional predictive approaches, achieving better mean absolute deviation, mean absolute error, and root-mean-squared errors.

Krauss et al. [20] conducted a study where they implemented deep neural networks, gradient-boosted trees, Random Forest models, and an ensemble model of these to predict the stocks in the S&P 500 index. The target variable predicted in the study was binary. It was set to one if the subsequent one-day return of the stock exceeded the cross-sectional median return of stocks in the S&P 500 and set to zero if it did not exceed the median. Subsequently, the stocks with the highest probabilities of outperforming the cross-sectional median return were used to rank stocks for long positions and vice versa. Their trading sample results showed that all machine learning-based trading strategies outperformed the aggregate market. Similarly, Wolff and Echterling [21] showed that selecting stocks based on weekly predictions of whether individual stocks within the S&P 500 index would exceed the cross-sectional median return, using deep neural networks, long short-term memory (LSTM), Random Forest, and gradient boosting, resulted in higher risk-adjusted return than the S&P 500 index.

There have also been studies demonstrating the added value of machine learning in terms of the predictability of non-US stock market indices. For instance, Fieberg et al. [22] showed that linear regression underperforms Support Vector Regression, Random Forest, gradient boosted trees, and Neural Networks in the monthly predictability of cross-sectional European equity returns. Kumar et al. [23] applied PCA, particle swarm optimization, and Levenberg–Marquardt algorithm combined with feed-forward Neural Networks, which outperformed PCA-based Auto-Regressive Distributed Lag Model in predicting daily price of Nifty 50, Sensex, and S&P 500. Hussain et al. [24] showed that the ANFIS (Adaptive Neuro-Fuzzy Inference System)-induced OWAWA (Ordered Weighted Averaging Weighted Average) model outperforms traditional time series models in 30-day price predictions of the Australian Securities Exchange. Karathanasoupoulos [25] applied the Gene Expression Programming (GEP) machine learning algorithm to predict and trade futures contracts of FTSE100, DAX30, and S&P 500 daily closing prices. The study results demonstrated that the latter machine learning model outperformed in terms of predictability and trading returns versus more traditional approaches like a random walk model and an autoregressive moving average model.

Furthermore, a study by Grudniewicz and Ślepaczuk [26] applied various machine learning algorithms using technical predictors to predict and trade the following market indices: WIG20 (Poland), DAX (Germany), S&P 500 (United States), BUX (Hungary), PX (Czech Republic), SOFIX (Bulgaria), OMXR (Latvia), OMXT (Estonia), and OMXV (Lithuania). The machine learning algorithms applied in the study were Neural Networks, k-Nearest Neighbour, Regression, Random Forest, Naive Bayes, Bayesian Generalized Linear Model, linear SVM, and polynomial SVM. The target variable was the daily high, low, and close prices of the indices. The research shows that algorithmic strategies based on machine learning models outperformed passive buy-and-hold benchmark strategies in terms of risk-adjusted returns. Nikou et al. [27] implemented LSTM, Support Vector Regression, Random Forest, and ANN to predict the daily close price of an exchange-traded fund tracking the MSCI United Kingdom index. The LSTM model achieved superior prediction accuracy, with lower mean absolute error, mean-squared error, and root-mean-square error compared to the others, with Support Vector Regression being the next best performer. However, there was no inclusion of non-machine learning methods in the latter two studies to optimally evaluate the added value of machine learning.

Kumbure et al. [5] conducted a literature review of studies that used machine learning to predict prices or returns of the stock market. If one takes the full systematic literature review into account, the review included 150 studies. Their review revealed that Neural Networks were the most frequently deployed machine learning method applied for stock market predictions, followed by SVM and fuzzy-theory-based methods. Moreover, Kumbure et al. [5] showed that the vast majority of studies used high-frequency data and deployed daily predictions. Their review showed that 62% of the features in the studies they covered were various technical indicators, while only 7.2% were fundamental indicators. Similarly, Nazareth and Reddy [10] conducted a literature review of 126 articles from 44 reputed journals to examine recent advances in the area of financial machine learning applications. They also observed that ANN (multi-layer perceptron), based on feedforward or backpropagation methods, was most frequently used for predicting stock markets, followed by SVM and Random Forest.

2.1 Limitations and gaps in previous studies

Based on the literature review, this section details seven critical gaps and limitations in previous studies that this study addresses. First, most studies have focused on short-term prediction horizons, typically daily or intraday, rather than long-term forecasting. This limits their practical applicability for institutional investors and portfolio managers who operate on longer timeframes.

Second, none of the studies utilizes XGBoost, despite its proven predictive power and documented competitiveness in data scientific predictive competitions outside academia [7–9]. This neglect is also apparent in the literature review of Kumbure et al. [5], which encompassed a total of 150 study articles. Out of these 150, only 4 studies tested XGBoost.

Third, most studies predict stock prices rather than total return, excluding the effect of dividends, which is a crucial component of the overall investment outcome in terms of monetary value. Fourth, many studies that apply binary predictions follow a non-rigorous approach to assess the added value of the machine learning models by evaluating the predictive accuracy rates against a random 50% threshold. This is not optimal, as a simple passive buy-and-hold strategy independent of any machine learning models, which can be viewed as an implicit continuous prediction that stock returns will be positive next period, would achieve a predictive accuracy significantly above the random threshold. This is because monthly returns of stocks are mostly positive.

Fifth, there has been limited experimentation with adjusting classification thresholds to better reflect the prior probability distribution of stock market returns, which tend to exhibit a positive bias. Thus, financial professionals would assumably require higher predictive confidence before taking short positions, for example. Hence, implementing higher thresholds for predicting negative returns could potentially enhance the real-world applicability.

Sixth, most studies use technical predictors, and there is often a neglect of the potential predictive power of various fundamental predictive variables. There is empirical evidence from the financial literature that suggests that various fundamental variables can exhibit explanatory power of variations in future aggregate stock market returns or cross-sectional equity returns. For example, dividend yield [28], equity risk premium [29], and the cyclically adjusted price-to-earnings ratio (CAPE) [30]. Moreover, valuation multiples such as the book-to-market ratio [31] and average analyst earnings estimates [32] have demonstrated predictive power of cross-sectional stock returns. For these reasons, it may be considered sub-optimal to exclude various variations of these fundamental predictors.

Finally, while the following limitation is not consistent across all studies, many employ regression predictions of the stock market as a continuous target variable, instead of directional binary predictions. Regression outputs may not be optimal if the aim ultimately is to maximize the predictability of the prospective stock market directional outcome, in terms of positive or negative returns, as suggested by Campisi et al. [33]. This study compared classification models with regression models that instead used a continuous target variable by transforming the continuous output into a binary variable depending on whether the regression

output implied positive or negative returns. The study demonstrated that the classification models, all else equal, outperform in terms of predictive accuracy.

3 Method

This study develops a model that facilitates data-driven portfolio management and asset allocation decisions. The model is intended to increase the likelihood of achieving positive abnormal risk premiums relative to benchmark portfolios. More specifically, the objective is to determine whether machine learning and the XGBoost algorithm can generate better out-of-sample prediction accuracy of the total return in the broad market index from time t to 12 months forward relative to the traditional logistic regression model. Given the competitive predictive accuracy of machine learning algorithms, it is interesting to explore whether these algorithms can achieve competitive predictive accuracy and add value in the context of portfolio management. The literature shows that there is a plethora of research studies that apply machine learning algorithms to asset pricing. However, to distinguish between exaggerated and valuable insights, it is imperative to determine whether the research results and conclusions can be implemented and generate practical value [34]. Therefore, this study focuses on assessing whether applied machine learning algorithms generate added value in the form of a practically implementable portfolio strategy.

3.1 Conceptual framework

Figure 1 shows the conceptual framework of the constructed models in this study. This conceptual framework also applies to the traditional logistic regression model, which is explained in more detail below and will be used as a benchmark to evaluate the added value of the XGBboost models.

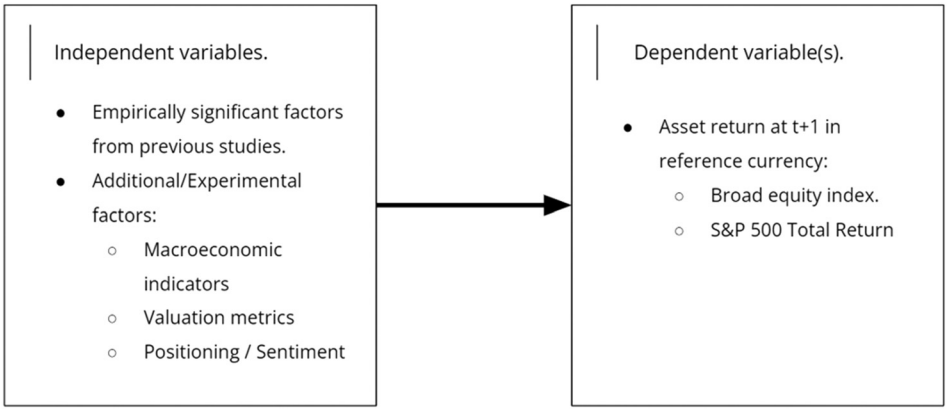

Figure 1: Conceptual framework of the model. Note. An illustration of how the x and y variables, in the intended model creation, hypothetically can correlate or have a causal relationship. Figure created by authors.

3.2 Design

A gradient boosting algorithm was employed, which is a machine learning method that involves an ensemble of weaker prediction models. The decision tree boosting algorithm XGBoost [35] was utilized. The choice of this algorithm was motivated in the introduction. Equation (1) for the predicted value y, given a set of independent variables X, is cited from the study of Chen and Guestrin [35]:

$$\hat{y} = \phi(x_i) = \sum_{i=1}^{n} w_i f_i(x_i), f_i \in \mathcal{F}. \quad (1)$$

In equation (1), f_i represents every ith and unique decision tree, from $i = 1$ to n, and $f_i \in F$ indicates that each decision tree is an element of the total set of F regression trees [35]. Every sequentially added independent regression tree f_i aims to correct the prediction errors of the ensemble of the preceding trees. Therefore, given a set of X variables, the final prediction $\hat{y}$ is the weighted sum of all independent scores from each decision tree f_i.

In more detail, binary and logistic XGBoost models will be constructed, which, on the basis of the independent variables, at time t, intend to predict a binary y-variable at time $t + 1$, where 1 means that the total return in S&P 500 Index from time t to 12 months forward is predicted to be negative. Conversely, 0 corresponds to the positive total return in the S&P 500 Index from time t to 12 months forward. The frequency of the time-series data frame is 1 month, and at each row/observation, at time t, the total percentage rolling return from that point in time to 12 months ahead (i.e. 12 rows/observation downwards) was calculated, which was subsequently converted into the binary dependent variable.

The constructed model, in essence, generates an underlying conditional probability distribution of the dependent variable, which represents the total return in the S&P 500 from time t to 12 months forward, given the explanatory variables. However, the predicted probability is assigned to a class in the form of 0/1 through a specified threshold. In other words, all models, including the logit model, are probabilistic classifiers that generate probability distributions over classes given specific values, x_i, that each of the random explanatory variables X_i takes, as detailed in equation (2). This study experimented with two classification thresholds, which are the probability that the positive outcome will occur postulated by the model that must be exceeded for the latter outcome to be predicted. More specifically, each model will be tested on the test data by using 50%, which is the standard procedure in the literature, and 70% as the prediction limit. This is done to observe whether a higher limit value, that is, a probability that is above 70%, to predict that stocks will give a negative return – the less likely outcome according to the *a priori* probability distribution – can provide better prediction measures:

$$P(Y = 1|X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n). \quad (2)$$

Therefore, the potential utility or added value for a hypothetical end user in the proposed model is that it can enable more empirical and data-driven portfolio allocation decisions between equities and bonds. The x-variables of the models are time-lagged predictors of future outcomes of whether the total return will be positive or negative in the next 12 months. The data frame consists of a set of hypothetically or potentially significant explanatory variables at time t and a binary dependent variable column of equity index total percentage return in US dollars from time t to 12 months forward at time $t + 12$, which takes a value of 1 when negative and 0 when positive. Hence, the models constructed in this study were binary. The fact that the values of explanatory variables are at time t and that the total return in the stock index refers to the period from and after time t to 12 months ahead, $t + 12$, will facilitate that the potential and hypothetical end users at a given time could input all independent variables into the model to receive the most probable outcome. The choice to try to predict 12 months of forward return instead of short-term future returns is not arbitrary. Instead, it depends partly on a hypothetical assessment but also to some extent based on observations from the literature review that the potentially predictive signals used in this study are more likely to have long-term predictive value as reversal and contrarian indicators, such as, for example, the various volatility, sentiment, and valuation variables used in this study. The second reason is a hypothetical assessment that high-frequency predictions and trading are likely to be more crowded, relative to longer period predictions on a quarterly or annual basis, among market participants, and hence, it may be more difficult to identify market anomalies.

A logistic regression model, based on the same x and y variables and the same historical period, will be constructed to measure and observe the performance differences between the machine learning algorithm and model and a more traditional statistical modelling approach, which has historically been more common in the financial literature when the research questions concerned various binary dependent variables [36,37]. By applying this comparison method, we can discern whether it is the predictive signal value of the explanatory

variables or the machine learning algorithm that contributes added value in terms of prediction accuracy, that is, those out-of-sample evaluation metrics that will be specified in more detail below.

Missing data observations in the variables are replaced with the mean value before logit modelling and left as an empty data point before XGBoost modelling. This is because the logit algorithm cannot handle empty cell values, whereas the XGBoost algorithm can handle empty cell values. During training, XGBoost determines the optimal path for handling missing values. It decides on whether to place missing values in the left or right node of a decision tree to reduce loss. If no missing values are encountered during training, any new missing values are, by default, directed to the right node. Thus, it can be considered an essential distinction between the abilities and capacities of the XGBoost model and the baseline model. Therefore, relevant to maintaining this difference in how missing cell values are handled as part of the research objective regarding the comparison between the machine learning algorithm's potential ability or inability to obtain better prediction results relative to the traditional logistic regression method. For clarification, there were no missing values in the dependent variable for the XGBoost model.

3.3 Data, preparation, dimensions, splitting, and source

All data in this study were obtained from S&P Global's S&P Capital IQ system. More specifically, the tool named Chart Builder, which is available on the Capital IQ Pro platform, has been used. In terms of accessibility, S&P Capital IQ is a subscription-based service, and therefore, access to the data is dependent on a paid subscription. As for rights and permissions, the use of data from the S&P Capital IQ Pro platform is subject to the terms and conditions of the subscription agreement. Through the chart-building tool, the variables needed to construct the models in this study have been uploaded as time series, which this study intends to use, into one graph. These multiple time series have then been downloaded from the chart builder tool to Excel as a multidimensional data frame. In Excel, the time series have then either been used in their original form or processed in various ways to obtain the desired variable for the model constructions. All explanatory variables underwent stationarity tests, and if they were not stationary, they were replaced with the percentage or absolute change from 12 observations (i.e. months) to eliminate the trend in the time series of explanatory variables. The dependent variable was created by calculating the percentage change in the S&P 500 total return index from a given time to 12 observations forward in time and then converting it to a binary variable based on whether the value is positive or negative. The final time series and data frame as the predictive models, including the benchmark and benchmark models, had a time frequency of 1 month, 24 variable columns, and 410 observations spanning the period from November 1987 to February 2022.

For the XGBoost modelling, random sampling without replacement was performed to obtain training (70%), validation (15%), and test (15%) samples. This approach enables model generalizability across time and samples. Validation samples were used to fine-tune the parameters regarding the learning rate and number of iterations in the training process for XGBoost modelling. More specifically, the validation sample was used as out-of-sample test data to avoid overfitting the training data and, consequently, the risk of poor prediction measures on new data. Next, the actual model test was carried out, where the prediction scores, which are presented in Section 4, were measured on the test sample. For the logistic baseline model, a separate and independent random sampling without replacement, but without a validation sample, was carried out, where 80% of the same original data became training data and 20% became test data.

The proposed model includes some independent variables that are either the same as or similar to those that previous empirical studies have proved to possess predictive power for one or more financial market assets. Moreover, independent variables, which may not have been sufficiently used in the literature for predictive modelling of future returns of broader US equity indices or cross-sectional stock returns at time $t + 1$, are included in the model construction. The independent variables will be either the raw absolute value of the variable, if it is stationary, or the percentage or absolute change in the absolute value up to time t. However, it should be emphasized that the research objective in this study is not to identify new statistically significant market anomalies and predictors or to test whether market anomalies that previous studies have

identified are robust. The research objective is to determine whether machine learning and the XGBoost algorithm can generate better out-of-sample prediction accuracy for the total return in the broad market index from time t to 12 months forward, relative to the traditional logistic regression model. Hence, both the machine learning model and the logistic regression model will include the same set of independent variables to keep as many parameters as possible the same in the comparison model to be able to determine with greater certainty whether it is the machine learning algorithm that leads or does not lead to better prediction metrics.

A list that contains the predictors used in the construction of the machine learning models and the baseline logistic regression model is as follows:

- CBOE Volatility for S&P 500 Index (VIX index) at time t.
- S&P 500 monthly Total Return, from $t - 12$ to t.
- United States Treasury Constant Maturity 10-Year Rate Value at time t.
- The change from $t - 12$ to t, for the United States Treasury Constant Maturity 10-Year Rate Value.
- Copper to gold price ratio at time t.
- Forward earnings at time t.
- Forward earnings monthly % change from $t - 1$.
- Forward earnings quarterly % change from $t - 3$.
- Forward earnings yearly % change from $t - 12$.
- The equity risk premium, % change from $t - 12$ to t.
- The equity risk premium, absolute change from $t - 12$ to t.
- The equity risk premium, calculated with dividend yield instead of earnings yield, absolute change from $t - 12$ to t.
- The dividend yield of the S&P 500 absolute change from $t - 12$ to t.
- Monthly absolute change, from time $t - 12$ to t, of the index value of the S&P 500 Index (SPX).
- Monthly absolute change, from time $t - 12$ to t, of the S&P 500 Total Return index.
- Monthly absolute change in Break-even inflation rate from $t - 12$ to t.
- Gold's last price changed from $t - 12$ to t.
- Copper's last price changed from $t - 12$ to t.
- Forward price to earnings ratio for S&P 500, change from time $t - 12$.
- Equity premium return relative to government bonds, from time $t - 12$ to t.
- The share pricing of the long-term treasury mutual fund (VUSTX) at time t.
- The monthly return of the long-term treasury mutual fund (VUSTX), from $t - 12$ to t.

The explanatory variables listed above partially consist of predictors that are mostly neglected in ML applications, despite that they have been empirically shown in the financial literature to significantly add to the explanatory power of variations in future aggregate stock market excess or absolute returns. These are the dividend yield [28], the implied volatility (VIX) [38], the gold return [39], the equity risk premium [29], and lagged returns. Moreover, the inclusion of historical stock market excess and absolute returns was to incorporate potential predictive information from the momentum anomaly [40] or the mean reversion anomaly [41]. Since XGBoost is insensitive to the scale of the input data, and to avoid loss of interpretability and meaningful information inherent in original scales, normalization was not applied.

3.4 Objective function of the binary XGBoost model

The goal of XGBoost model training is to minimize the loss function and regularization term in equation (3) [35]:

$$\left[\sum_{i=1}^{n} L(\hat{y}_i, y_i)\right] + \gamma T + \frac{1}{2}\lambda w^2. \tag{3}$$

Alternatively, the objective of the tth iteration is to minimize the function below, as stated by Chen and Guestrin [35]:

$$L^{(t)} = \sum_{i}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{4}$$

In equation (4), l represents the differentiable convex loss function, $\hat{y}_i^{(t-1)}$ is the prediction at the $t-1$ iteration, and f_t corresponds to the tree at the tth iteration [35,42]. $\Omega(f_t)$ is a regularization term that adds a penalty based on the complexity of the function.

This study applies binary and logistic classification models; thus, l corresponds to the log-likelihood of the Bernoulli distribution [42]. The probability will be logistic $(\hat{y}_i^{(t-1)} + f_t(x_i))$, and the complete formula will be according to equation (5). Equation (6) is an algebraic expression of equation (5), given that $\sigma(-z) = 1 - \sigma(z)$:

$$l = y_i \log(\text{logistic}(\hat{y}_i^{(t-1)} + f_t(x_i)) + (1 - y_i)\log(1 - \text{logistic}(\hat{y}_i) + f_t(x_i))) \tag{5}$$

$$l = y_i(\hat{y}_i^{(t-1)} + f_t(x_i)) - \log(1 + \exp(\hat{y}_i^{(t-1)}) + f_t(x_i))) \tag{6}$$

3.5 Hyperparameters and model configurations

The xgb.train function has been applied, which belongs to the package named xgboost in R [43].

As will be seen in the results, this study created two different final XGBoost models. These models were named XGBoost model 1 and XGBoost model 2. More specifically, XGBoost model 1 had 22,000 iterations (nrounds), a learning rate (ETA) of 0.001, and a maximum tree depth (max_depth) of 6. XGBoost model 2 had 9,500 iterations (nrounds), a learning rate (ETA) of 0.001, and a maximum tree depth (max_depth) of 6. The latter parameter that specifies maximum tree depth (max_depth) is, by default, 6. These parameter values were determined based on the procedure described in Section 3.7.

Given the imbalanced target variable of this study, a parameter that handles this form of class imbalance has been utilized. The xgb.train function enables the user to specify a positive-to-negative ratio in the binary response variable in the parameter named scale_pos_weight. Thus, this ratio was specified as one of the parameters in all XGBoost models constructed and is the same for XGBoost models 1 and 2. Imbalanced binary response variables can cause issues in statistical modelling, such as prediction bias towards the majority class. The latter bias may lead to poor overall model performance due to the inability to correctly predict instances of the minority class.

3.6 Out-of-sample test of predictability

Different metrics were calculated to compare and measure the performance of the XGBoost model relative to the more traditional logistic regression model, that is, the benchmark model. These metrics are the accuracy rate, recall, precision rate, NPV, balanced accuracy, and specificity [44]. These predictive metrics are used to answer the research question regarding whether the machine learning algorithm can result in better predictions relative to the more traditional approach. The results will include confusion matrices so that readers can independently assess the model's performance by calculating other metrics from each matrix that they find suitable. For clarification, the value of the binary variable is positive (1) when the total return of the S&P 500, from time t on the last day of a given month to the last day 12 months into the future, is negative; conversely, the binary variable is negative (0) when the total return is positive.

An additional criterion will be set, as illustrated in equation (7), which the machine learning and XGBoost model should meet in order for it to be considered that it can contribute a predictive benefit and potential economic added value for any market participant who uses the model in their investment positioning. The criterion in equation (7) postulates that the accuracy rate should be higher than the no-information rate, which corresponds to the percentage proportion in the test data, which is the most frequent outcome and, therefore,

most likely. Intuitively, a hypothetical rational investor with access to the *a priori* distribution of the binary dependent variable would achieve an accuracy rate equivalent to the no-information rate (NIR) by always predicting that the total return would be positive without having access to a model. Hence, a model with potential financial benefits should obtain an accuracy rate that exceeds the no-information rate:

$$\text{ACC} > \text{NIR}. \tag{7}$$

The null hypothesis is that the model has no information and predicts only the most common class in the data. The alternative hypothesis is that the model contains information and performs better than NIR. The hypotheses are specified in equations (8) and (9), where p corresponds to the prediction accuracy rate. More specifically, a p-value is obtained through binomial testing, which corresponds to the probability of obtaining at least as many correct predictions as the model if one guessed according to the most probable outcome according to the *a priori* distribution:

$$H_0: \quad p = \text{NIR}, \tag{8}$$

$$H_1: \quad p > \text{NIR}. \tag{9}$$

3.7 Fine-tuning of parameters and overfitting control

To avoid overfitting, which is a common problem when using machine learning models, we must regularize our model and prevent it from fitting too closely to the training data. Thus, we can ensure that our model generalizes well to new data samples and makes accurate predictions. In this study, we used the XGBoost algorithm, which has a built-in regularization term in its objective function, as shown previously. This term helps reduce the complexity of the model and the risk of overfitting. We also used a validation sample to monitor the error rate of the model during the training process. We adjusted the learning rate (ETA) and the number of iterations to find the optimal balance between model complexity and the error rate. We plotted the error rate against the number of iterations for both the training and validation samples (see Section 4) and observed how they change with an increase or decrease in the learning rate and the number of iterations. We aimed to achieve a low and stable error rate for both samples without any sign of divergence. A divergence indicates that our model overfits the training data and loses its predictive power on the validation data.

4 Results

4.1 Out-of-sample prediction measures and comparison

In Table 1, the research results are presented and summarized, which can answer the research objective and the question of whether the XGBoost learning algorithm leads to better prediction ability relative to the logistic regression model. In addition, it is also reported in Table 1 whether each model meets the predetermined criterion that the accuracy rate must be higher than the no-information rate for the model to be considered able to contribute information value. The model's binomial test p-value is also reported, with smaller p-values indicating stronger evidence against the null hypothesis. The null hypothesis is that the model has no informational value and predicts only the most common class in the data.

As shown in Table 1, the XGBoost models achieved superior predictive measures for all metrics. Moreover, all XGBoost models fulfilled the criteria of receiving accuracy rates that exceeded the no-information rate. One of the two comparative logit models, which was the one with an increased classification threshold, achieved an accuracy rate that exceeded the no-information rate, but, as postulated by the p-value shown in Table 1, with a relatively low degree of certainty that the unfavourable null hypothesis can be rejected. Meanwhile, the XGBoost models with the same probability threshold of 70%, in order to predict that stock market returns

Table 1: Models and their predictive metrics on the test sample

Model	Accuracy	Recall	Specificity	Precision	Negative predicted value	Balanced accuracy	NIR	ACC > NIR	*p*-value (ACC > NIR)
Xgb Model 1 (0.5)	0.9	0.98	0.5	0.9074	0.833	0.74	0.833	Yes	0.1081
Xgb Model 1 (0.7)	**0.9167**	**1.0**	0.5	0.9091	**1.0**	0.75	0.833	Yes	**0.05121**
Xgb Model 2 (0.5)	0.9	0.96	**0.6**	**0.9231**	0.75	**0.78**	0.833	Yes	0.1081
Xgb Model 2 (0.7)	**0.9167**	**1.0**	0.5	0.9091	**1.0**	0.75	0.833	Yes	**0.05121**
Logit Model (0.5)	0.8375	0.9254	0.3846	0.8857	0.5	0.6550	0.8375	No	0.5733
Logit Model (0.7)	0.8625	0.9851	0.2308	0.8684	0.75	0.6079	0.8375	Yes	0.33488

Note. Bold values indicate the model that received the best score for the given metric. The value within parentheses (0.5 or 0.7) of a given model indicates the classification threshold. Since negative returns are defined as class 1, the metric calculations assume that 1 is the negative class and 0 is the positive class.

will be negative, as the latter logit model had significantly lower *p*-values of 0.05121. This indicates a relatively low probability that the models are insignificant postulated by the null hypothesis.

4.2 Confusion matrices

Figure 2 shows the confusion matrices for each of the constructed models, including the comparative logistic regression models. It includes the models with classification thresholds of 50 and 70%, which means that the required probability that the total stock market return will be negative (binary class 1) must exceed these thresholds in order to predict this outcome.

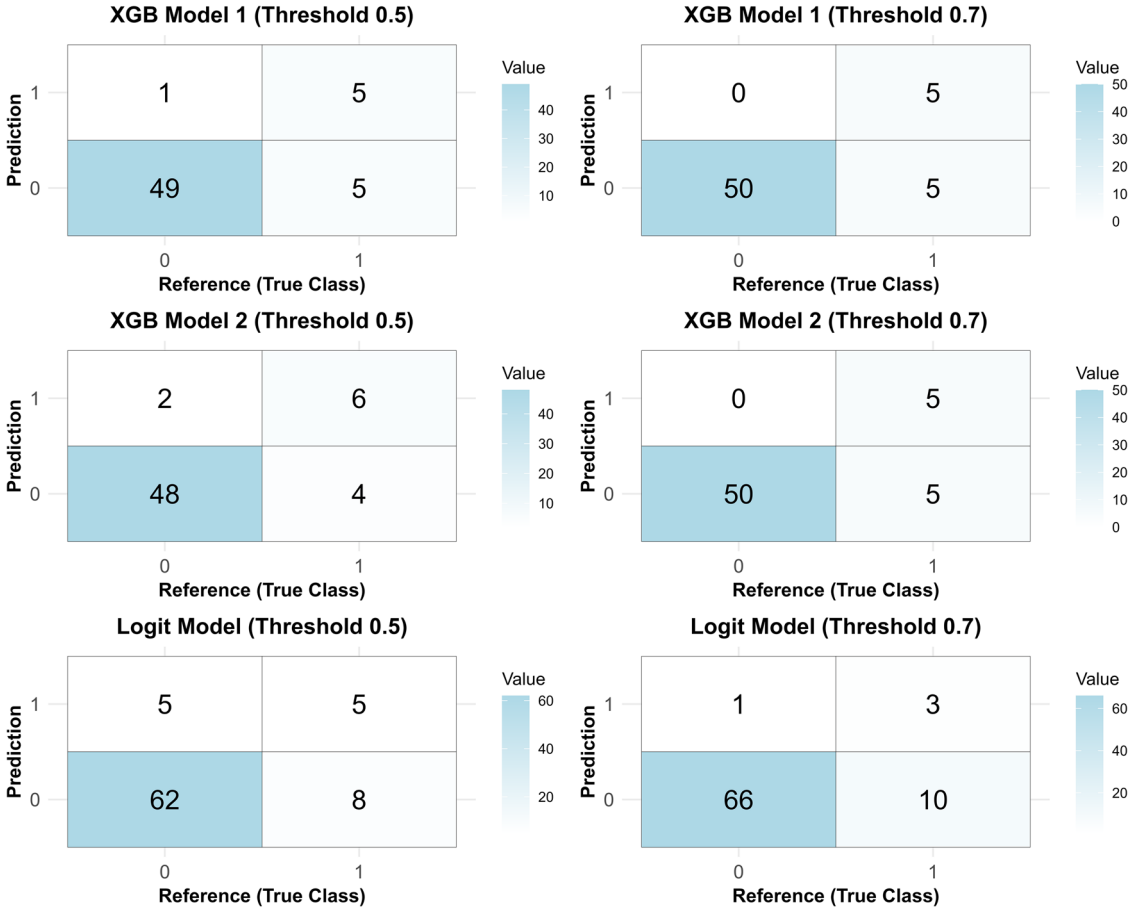

Figure 2: Confusion matrices for each model based on the test sample predictions. Note. Thresholds refer to the classification thresholds to predict class 1. XGB: XGBoost. Class 1 refers to negative returns, and class 0 to positive returns. Figure created by authors.

5 Discussion

One main literature contribution is a refined and more robust methodological framework for prospective studies, which will implement binary machine learning models of stock market index returns for enhanced real-life utility and applicability for investors. These methodological suggestions are fourfold. First, this study has provided empirical evidence that suggests that XGBoost should be incorporated in the set of benchmark models when evaluating any given model. Second, binary classifiers should predict the direction of the total

return, instead of the price. Third, statistical binomial tests should be conducted to ensure that the accuracy rate significantly exceeds the no-information rate. Finally, controlled experimentation with different classification thresholds for a given model to better account for the *a priori* probability distribution.

The following paragraphs explore the potential practical implications of the study outcomes and findings for various industry practitioners within the investment management industry. These implications can be divided into generic and specific. First, from a general perspective, given that the study results demonstrate that machine learning algorithms can be more capable of exceeding other more traditional statistical modelling practices, the implication is that asset managers with the intention to maximize their returns should at least expand their use and exploration of various machine learning algorithms and models that have shown compelling out-of-sample predictive metrics to predict financial market asset returns.

The more specific implication for the same end user as above concerns how the XGBoost models in this study can be used and benefited in practice. A potential investor and market participant who wants to utilize the XGBoost model produced in this study would, in practice, enter the values of the respective explanatory variables into the model on the last day of a given month, which then generates a prediction in the form of 1, meaning that the stock index will give a negative return in the upcoming 12 months, or 0, meaning it will generate a positive return in the next 12 months.

Given that the XGBoost models, in most cases, predict that the stock index will generate positive returns, the implication is that the user who follows the model as an investment strategy would consequently follow a relatively passive investment strategy, wherein the majority of the time, the investor is along the S&P 500, with or without leverage. Then, on occasion when the model predicts negative returns with a high probability of being right, as suggested by the out-of-sample NPV in the results, the investor would take positioning action by, for example, (a) reducing its exposure to the stock index and increasing its cash holdings or (b) reducing its leverage in the S&P 500 to neutral, and conversely, increase its leverage when the prediction is that it will deliver positive return again. The exact positioning actions depend on the investor's preferences in terms of, for example, long-term return targets, as well as risk tolerance regarding short- to medium-term volatility in returns.

One key advantage of the proposed approach of increased classification threshold is its ability to generate an NPV of 100%, meaning that every negative return prediction was correct. This aspect of model performance is particularly relevant for risk-averse investors or portfolio managers seeking to minimize drawdowns, as it enables strategic portfolio adjustments based on reliable downside signals.

Recall measures how well the model detects instances where the actual returns were positive. The recall and NPVs of 1 in the models with a higher classification threshold (70%) result from the absence of false negatives, as shown in Figure 2, meaning that whenever the model predicted a negative return, the actual return was indeed negative. This occurs because the model only predicts negative returns when it is highly confident (above the 70% threshold), reducing the likelihood of false negatives. While a metric of 1 can sometimes indicate overfitting to the given sample, this is not necessarily the case here. Thus, the high recall and NPV are a natural consequence of the stricter classification threshold rather than an overfitting issue. Moreover, the model's performances were evaluated using out-of-sample test data, further mitigating concerns about overfitting.

All XGBoost models outperformed the no-information rate; in other words, the majority class rate. This latter benchmark corresponds to the accuracy rate that a buy-and-hold strategy would implicitly obtain in the same test sample, as it indirectly predicts each time that the stock will go up. This latter benchmark is relevant since it has been shown that passive buy-and-hold strategies for the stock market index, regardless of future outcome predictions, can generate relatively attractive returns over time compared to more active investment strategies and actively managed funds [11,12].

Given the full literature review, Kumbure et al. [5] summarized the best prediction performances across all studies for each of the following machine learning categories: ANN, SVM, fuzzy theory, deep learning, feature selection, and others. Below we summarize the ones of these best performances that used binary classification performance measures and had US stock market or individual US stocks as their target variable, like this study, for comparison with the state of the art in the literature. For deep learning approaches, the best reported accuracy rate was 66.32% [45], which also predicted the S&P 500 like this study. Thus, the accuracy

rate of 91.67% of the proposed models of this study significantly outperforms the latter. For ANN approaches, the best-performing study by Hu et al. [46] used hit ratio as the performance measure, and they achieved a hit ratio of 86.81% in predicting the S&P 500. In the context of stock market predictions, the hit ratio refers to the proportion of true predictions of positive returns out of the total number of predictions [5]. As shown in Figure 2 (50 out of 60 total predictions), this study achieved a hit ratio of 83.33%, which is somewhat lower but competitively close. For feature selection approaches, the best reported accuracy rate was 85.8% [47], and for the other category, which includes various machine learning algorithms, the best accuracy rate was 86.67% [48]. Thus, the XGBoost models in this study, which achieved out-of-sample test accuracies of 91.67 and 90%, are considerably higher than most reported accuracy rates of state-of-the-art models from the literature.

6 Conclusion

This study evaluated the predictive power of XGBoost, a machine learning algorithm, relative to the logistic regression algorithm, a more standard statistical model, for the binary classification of the S&P 500 index's total return in the next 12 months. The results demonstrated that XGBoost outperformed logit significantly in all six measures of prediction accuracy, as detailed in Table 1, using out-of-sample test data, which were randomly sampled without replacement. In addition, it can also be concluded that all XGBoost models obtained an accuracy rate that was notably higher than the no-information rate, as postulated by the predetermined criterion specified in equation (7), which means that the models have and can contribute with predictive added value relative to a prediction strategy that only predicts the most probable outcome according to the prior distribution of the y-variable in the training sample.

It is noteworthy that the XGBoost models with a higher classification threshold of 70%, both for XGBoost models 1 and 2, were more statistically significant than the models with a standard classification threshold of 50%, as they obtained p-values of 0.05121, which can be considered statistically significant. Even the XGBoost models with a classification threshold of 50% obtained relatively low p-values of 0.1081, that is, a probability of only 10.91% that the null hypothesis that the model lacks information, as shown in equation (8), is true.

Moreover, it is also noteworthy that the latter higher classification threshold resulted in an NPV of 100%, which corresponds to an accuracy rate of 100% when the models predict that the stock index will give a negative return in the coming year. This result is intuitively understandable, as a higher classification limit of 70% means that greater probabilistic certainty is required when a less likely outcome according to the *a priori* distribution is to be predicted. Thus, this decreases the number of predictions that postulate that stocks will generate negative returns compared with when the classification threshold is 50%. More specifically, it can be observed from the confusion matrices that the number of negative-return predictions decreased from eight to five for XGBoost Model 2 and from six to five for XGBoost Model 1. However, this leads to a higher accuracy rate when the negative returns are predicted.

Thus, one can intuitively derive that the gross returns from the out-of-sample test data, which were randomly sampled without replacement, of an investment strategy that would follow these statistically significant XGBoost models with a higher classification threshold by underweighting the position in the S&P 500 when negative returns are predicted, and being fully invested in the S&P 500 when positive returns are predicted, would result in higher gross returns relative to the passive index gross returns of S&P 500.

6.1 Future work

Future research could aim to enhance performance in terms of out-of-sample test metrics by experimenting with (a) various classification thresholds; (b) testing with multi-year, semi-annually, and quarterly data frequency and predictions; (c) adding other categories of potentially significant predictors to the model; and (d)

adding data rows after February 2022, where this study's data frame ends, to include a period of inflation crisis, that is, a period when risk-off sentiment prevails due to rising inflation rates.

In addition, it would be interesting to simulate an investment strategy that follows the predictions of the XGBoost model in this study or an approximately similar binary XGBoost model relative to, for example, a passive strategy that constantly owns the S&P 500 to be able to determine with certainty whether it would generate a net excess return after transaction costs. It would also be interesting to test the robustness of the predictability of the XGBoost model in this study, or approximately similar binary XGBoost models, across different data samples.

Funding information: The authors state no funding involved.

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and consented to its submission to the journal, reviewed all the results, and approved the final version of the manuscript. The contributor roles taxonomy according to CRediT: Rojen Erik Sürek: methodology, writing – original draft, formal analysis, software, investigation, and conceptualization; Wee-Yeap Lau: supervision, project administration, and writing – review and editing.

Conflict of interest: The authors state no conflict of interest.

Data availability statement: We, as authors, agree to make data and materials supporting the results or analyses presented in this paper available upon reasonable request.

References

[1] Friedrich S, Antes G, Behr S, Binder H, Brannath W, Dumpert F, et al. Is there a role for statistics in artificial intelligence? Adv Data Anal Classif. 2022;16(3):823–46.
[2] Sghir N, Adadi A, Lahmer M. Recent advances in predictive learning analytics: A decade systematic review (2012–2022). Educ Inf Technol. 2022;28(1):1–381.
[3] Zhang C, Ma Y. Ensemble machine learning. Vol. 144, New York: Springer; 2012.
[4] Ibbotson RG, Kaplan PD. Does asset allocation policy explain 40, 90, or 100 percent of performance? Financ Anal J. 2000;56(1):26–33.
[5] Kumbure MM, Lohrmann C, Luukka P, Porras J. Machine learning techniques and data for stock market forecasting: A literature review. Expert Syst Appl. 2022;197:116659. doi: 10.1016/j.eswa.2022.116659.
[6] Fawcett T. An introduction to ROC analysis. Pattern Recognit Lett. 2006;27(8):861–74. doi: 10.1016/j.patrec.2005.10.010.
[7] Brownlee J, "How to use XGBoost for time series forecasting." Machine Learning Mastery. Accessed 4 April 2024. [Online]. Available at: https://machinelearningmastery.com/xgboost-for-time-series-forecasting/.
[8] Martinez H, "Scaling Kaggle competitions using XGBoost: Part 1." PyImageSearch. Accessed 4 April 2024. [Online]. Available at: https://pyimagesearch.com/2022/11/21/scaling-kaggle-competitions-using-xgboost-part-1/.
[9] Ye A., "XGBoost, LightGBM, and other Kaggle competition favorites." Analytics Vidhya. Accessed 4 April 2024. [Online]. Available at: https://medium.com/analytics-vidhya/xgboost-lightgbm-and-other-kaggle-competition-favorites-6212e8b0e835.
[10] Nazareth N, Reddy YVR. Financial applications of machine learning: A literature review. Expert Syst Appl. 2023;219:119640. doi: 10.1016/j.eswa.2023.119640.
[11] Dichtl H. Investing in the S&P 500 index: Can anything beat the buy-and-hold strategy? Rev Financ Econ. 2019;56:78–94.
[12] Siegel JJ. Evaluating a buy and hold strategy for the S&P 500 index. J Portf Manag. 2002;28(2):110–9.
[13] Li P. An empirical evaluation of four algorithms for multi-class classification: Mart, abc-mart, robust logitboost, and abc-logitboost. arXiv preprint arXiv:1001.1020; 2010.
[14] Gu S, Kelly B, Xiu D. Empirical asset pricing via machine learning. Rev Financ Stud. 2020;33(5):2223–73. doi: 10.1093/rfs/hhaa009.
[15] Enke D, Thawornwong S. The use of data mining and neural networks for forecasting stock market returns. Expert Syst Appl. 2005;29(4):927–40. doi: 10.1016/j.eswa.2005.06.024.
[16] Zhong X, Enke D. Forecasting daily stock market return using dimensionality reduction. Expert Syst Appl. 2017a;67:126–39. doi: 10.1016/j.eswa.2016.09.027.
[17] Zhong X, Enke D. A comprehensive cluster and classification mining procedure for daily stock market return forecasting. Neurocomputing. 2017b;267:152–68. doi: 10.1016/j.neucom.2017.06.010.

[18] Zhong X, Enke D. Predicting the daily return direction of the stock market using hybrid machine learning algorithms. Financ Innov. 2019;5(1):1–20. doi: 10.1186/s40854-019-0138-0.
[19] Lahmiri S. A predictive system integrating intrinsic mode functions, artificial neural networks, and genetic algorithms for forecasting S&P500 intra-day data. Intell Syst Account Financ Manag. 2020;27(2):55–65.
[20] Krauss C, Do XA, Huck N. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. Eur J Oper Res. 2017;259(2):689–702.
[21] Wolff D, Echterling F. Stock picking with machine learning. J Forecast. 2024;43(1):81–102. doi: 10.1002/for.3021.
[22] Fieberg C, Metko D, Poddig T, Loy T. Machine learning techniques for cross-sectional equity returns' prediction. Spectr. 2023;45(1):289–323. doi: 10.1007/s00291-022-00693-w.
[23] Kumar G, Singh UP, Jain S. Hybrid evolutionary intelligent system and hybrid time series econometric model for stock price forecasting. Int J Intell Syst. 2021;36(9):4902–35. doi: 10.1002/int.22495.
[24] Hussain W, Merigó JM, Raza MR. Predictive intelligence using ANFIS-induced OWAWA for complex stock market prediction. Int J Intell Syst. 2022;37(8):4586–611. doi: 10.1002/int.22732.
[25] Karathanasopoulos A. Modelling and trading the London, New York and Frankfurt stock exchanges with a new gene expression programming trader tool. Intell Syst Account Financ Manag. 2017;24(1):3–11. doi: 10.1002/isaf.1401.
[26] Grudniewicz J, Ślepaczuk R. Application of machine learning in algorithmic investment strategies on global stock markets. Res Int Bus Financ. 2023;66:102052. doi: 10.1016/j.ribaf.2023.102052.
[27] Nikou M, Mansourfar G, Bagherzadeh J. Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. Intell Syst Account Financ Manag. 2019;26(4):164–74. doi: 10.1002/isaf.1459.
[28] Fama EF, French KR. Dividend yields and expected stock returns. J Financ Econ. 1988;22(1):3–25.
[29] Lander J, Orphanides A, Douvogiannis M. Earnings forecasts and the predictability of stock returns: Evidence from trading the S&P. J Portf Manag. 1997;23(4):24–35.
[30] Campbell JY, Shiller RJ. Stock prices, earnings, and expected dividends. J Financ. 1988;43(3):661–76. doi: 10.2307/2328190.
[31] Fama EF, French KR. The cross-section of expected stock returns. J Financ. 1992;47(2):427–65. doi: 10.2307/2329112.
[32] Bordalo P, Gennaioli N, La Porta R, Shleifer A. Expectations of fundamentals and stock market puzzles. Cambridge, MA, USA: National Bureau of Economic Research; 2020.
[33] Campisi G, Muzzioli S, De Baets B. A comparison of machine learning methods for predicting the direction of the us stock market on the basis of volatility indices. Int J Forecast. 2024;40(3):869–80. doi: 10.1016/j.ijforecast.2023.07.002.
[34] Coqueret G, Guida T. Machine learning for factor investing: R version. Boca Raton, FL, USA:Chapman and Hall/CRC; 2020.
[35] Chen T, Guestrin C. XGBoost: A scalable tree-boosting system. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016a, August. p. 785–94.
[36] Ge W, Whitmore GA. Binary response and logistic regression in recent accounting research publications: a methodological note. Rev Quant Financ Account. 2010;34(1):81–93. doi: 10.1007/s11156-009-0123-1.
[37] Hosmer Jr DW, Lemeshow S, Sturdivant RX. Applied logistic regression. Vol. 398. Hoboken, NJ, USA:John Wiley & Sons. 2013.
[38] Rubbaniy G, Asmerom R, Rizvi SKA, Naqvi B. Do fear indices help predict stock returns? Quant Financ. 2014;14(5):831–47.
[39] Baek C. How are gold returns related to stock or bond returns in the US market? Evidence from the past 10-year gold market. Appl Econ. 2019;51(50):5490–7.
[40] Moskowitz TJ, Ooi YH, Pedersen LH. Time series momentum. J Financ Econ. 2012;104(2):228–50. doi: 10.1016/j.jfineco.2011.11.003.
[41] Serletis A, Rosenberg AA. Mean reversion in the US stock market. Chaos, Solitons Fractals. 2009;40(4):2007–15.
[42] Hu M., "What is the "binary logistic" objective function in XGBoost?" Cross Validated Stack Exchange. Accessed 4 April 2024. [Online]. Available at: https://stats.stackexchange.com/questions/342552/what-is-the-binarylogistic-objective-function-in-XGBoost.
[43] Chen T, Guestrin C. XGBoost: Extreme gradient boosting. R package version 1.1.1.1. 2016b. Retrieved from https://CRAN.R-project.org/package=xgboost.
[44] Brown JB. Classifiers and their metrics quantified. Mol Inform. 2018;37:1–2. 1700127. doi: 10.1002/minf.201700127.
[45] Lien Minh D, Sadeghi-Niaraki A, Huy HD, Min K, Moon H. Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. IEEE Access. 2018;6:55392–404. doi: 10.1109/ACCESS.2018.2868970.
[46] Hu H, Tang L, Zhang S, Wang H. Predicting the direction of stock markets using optimized neural networks with Google Trends. Neurocomputing. 2018;285:188–95. doi: 10.1016/j.neucom.2018.01.038.
[47] Weng B, Ahmed MA, Megahed FM. Stock market one-day ahead movement prediction using disparate data sources. Expert Syst Appl. 2017;79:153–63. doi: 10.1016/j.eswa.2017.02.041.
[48] Zhou PY, Chan KC, Ou CX. Corporate communication network and stock price movements: Insights from data mining. IEEE Trans Comput Soc Syst. 2018;5(2):391–402. doi: 10.1109/TCSS.2018.2812703.