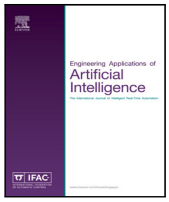
Engineering Applications of Artificial Intelligence 118 (2023) 105626

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Forecasting stock market for an efficient portfolio by combining XGBoost and Hilbert–Huang transform

Arsalan Dezhkam, Mohammad Taghi Manzuri *

Computer Engineering Department, Sharif University of Technology, Azadi Ave., Tehran, Iran

ARTICLE INFO

Dataset link: finance.yahoo.com

Keywords:
Hilbert–Huang transform
Machine learning
XGBoost
Trend prediction
Portfolio formation
Algorithmic trading

ABSTRACT

Portfolio formation in financial markets is the task of not taking non-necessary risks. Quantitative investment powered by machine learning has opened many new opportunities for more insight generation from financial data resulting in the explosion of ideas to enhance the performance of investments in the stock markets. In this research, we propose a first-introduced model called HHT-XGB to predict the changing trends in the next close price of stocks under the study. The proposed model combines Hilbert–Huang Transform (HHT) as the feature engineering part and the extreme gradient boost (XGBoost) as the Close price trend classifier. The classification output is a sequence of ups and downs used to optimize the stocks' portfolio weights with the best trading performance The performance of the portfolios optimized under this study proves that our novel combination of HHT with classification performs 99.8% better than forming the portfolio using raw financial data. The back-testing process suggests that the HHT-XGB strategy outperforms the benchmark strategies even with the poor-performing markets.

1. Introduction

Due to the intrinsic non-linearity and non-stationarity properties of stock market data, it is hard to predict future prices or trends over different stock markets. According to the classic efficient-market hypothesis (EMH), "beating the market" by forecasting future financial returns and making an over-average profit is impossible. However, in recent years the evolution of machine learning and deep learning techniques has opened new opportunities to earn more profits by better forecasting the future behavior of stocks. Most studies are split into two main categories, predicting the next value for a stock price, return rate, or market index as a regression procedure or forecasting how a specific stock or market will fluctuate according to historical data. The latter is a classification task mainly used to predict the market index and asset price trend, including stocks, bonds, cryptocurrencies, etc. However, the predicted direction is a significant informative signal that can be used by an algorithmic trading strategy or a portfolio selection schema. For any particular classification task, we need to know how to combine feature selection, feature extraction, and learning models to achieve maximum classification performance (Holder et al., 2005).

Deep learning techniques in recent years have been extensively exploited to learn market behavior by high-dimension input features. Advanced neural networks such as CNN, LSTM, DNN, RNN, and Reinforcement Learning are among the most applied models for predicting future movements (Carapuço et al., 2018; Chen et al., 2019; Hoseinzade and Haratizadeh, 2019; Kim and Won, 2018; Long et al., 2020, 2019; Ostad-Ali-Askari and Shayan, 2021; Pang et al., 2020; Singh and Srivastava, 2017; Song et al., 2019). However, using deep learning techniques needs access to high-performance computing infrastructures, whereas exploiting machine learning algorithms is still of interest across the industry and firm since the models can be quickly adopted and maintained with fewer computational resources and experts. Adaptive ensemble of Random Forest, the ensemble of Support Vector Machines, SVM, combined SVM with the generative probabilistic models, extreme gradient boosting, XGBoost, and its combination with various parameter optimization techniques are among many attempts that researchers carried out to benefit from the simplicity of ML implementation techniques to the task of stock market prediction (Chen et al., 2021; Derakhshan and Beigy, 2019; Dezhkam et al., 2022; Gomes et al., 2017; Li and Ping, 2015; Li et al., 2015; Min et al., 2021; Nobre and Neves, 2019; Ostad-Ali-Askari et al., 2017a; Pirnazar et al., 2018; Suárez-Cetrulo et al., 2019; Valencia et al., 2019; Yazdani et al., 2017).

In this paper, we propose a new approach for the task of portfolio formation based on forecasting the stock trend using the XGBoost classifier and the instantaneous components extracted from close price data by applying the HHT. There are two main hypotheses that we chase to verify them. The first hypothesis is to claim that "The HHT-XGB strategy outperforms the XGB and B&H strategies in terms of trading profitability and portfolio formation". The second hypothesis tries to verify that "The HHT-XGB portfolio formation avoids short positions".

* Corresponding author.
E-mail addresses: a.dejkam96@sharif.edu (A. Dezhkam), manzuri@sharif.edu (M.T. Manzuri).

https://doi.org/10.1016/j.engappai.2022.105626
Received 27 July 2022; Received in revised form 6 November 2022; Accepted 8 November 2022
Available online 5 December 2022

Our contribution to this study is two-fold. First, we apply a nonlinear non-stationary transform to extract more informative features from the close price data. Second, we simulate the performance of our assets under the study using a trading module. The trading module feeds the portfolio formation module with the stocks of the highest performance. Our proposed approach is further discussed in the following sections. Section 2 explains related works in recent years. Then, we provide the reader with the required materials in Sections 3.1 to 3.3. The outline of our proposed approach is thoroughly explained in 3.4. The reader can find the results and the discussions in Section 4. The paper ends with the conclusion in Section 5.

2. Related work

According to EMH, no investor can consistently "beat the market" since all information required for the best approximation of a company's intrinsic value can be captured within its stock prices (Burton, 2017, pp. 9–13). There are many contributing factors to stock prices, including macroeconomic variables, firm-specific information, market sentiment, financial and political news, and unexpected catastrophic events such as wars and natural disasters. Investors in financial markets are interested in advisory tools to know how these factors will impact their under-management assets. Therefore, researchers have extensively exploited statistical and computational models to study the open problems of this field, including next price prediction, market trend change, co-movement between financial markets, asset pricing, portfolio optimization, and many other topics under the study (Derakhshan and Beigy, 2019; Liagkouras and Metaxiotis, 2018; Liu and Wang, 2019; Mendonça et al., 2020; Pitkäjärvi et al., 2020; Winkler, 2020; Yang et al., 2018; Zhou et al., 2018). Portfolio optimization, however, is at the heart of most studies to find the best possible predictable next returns of stocks or how the market will fluctuate in the coming days. Markowitz's mean–variance asset allocation (also known as Markowitz Portfolio Theory, MPT) forms the optimum portfolio by solving a quadratic programming problem minimizing the risk constraint to a given expected return. The answer is the weight vector of assets forming the optimum portfolio (Markowitz, 1952). While the MV model has been extensively used as the first choice for portfolio formation in the literature, asset allocation is highly sensitive to prediction errors, and a few assets with a high excess return ratio will be overweighted within a portfolio. Hence, an experimental study was carried out by Martínez-Nieto et al. (2021) to review and compare 11 diversification-based strategies using four evaluation metrics the Calmer Ratio (CR), Sharpe Ratio (SR), Mean Return (MR), and Stability Index (SI). For the metrics considered, the global maximum return (GMR) strategy obtained competitive results in most datasets. The authors (Ślepaczuk and Zenkova, 2019) studied the investment strategies for the cryptocurrency market to verify the research hypothesis that the strategy based on the SVM algorithm can outperform the benchmark strategies in terms of return–risk relation. The experimentation was carried out on a set of six technical features, and the highest-ranked coins were used for long positions. The results showed that the EqW portfolio outperformed all the benchmark strategies including the SVM strategy. Therefore, the main hypothesis of their study was rejected.

To extend the existing literature on portfolio formation, Ma et al. (2021) incorporated the return prediction of traditional time series models into two machine learning and three deep learning models. They conducted the stock return prediction using Random Forest (RF), Support Vector Regression (SVR), Long Short-Term Memory (LSTM), Deep Multilayer Perceptron (DMLP), and Convolutional Neural Network (CNN). The authors then applied the Mean–Variance with forecasting (MVF) and Omega with forecasting (OF) portfolio formation models introduced by Yu et al. (2020) and Keating and Shadwick (2002), respectively. Their experimentation on 49 selected stocks from the Chinese Securities 100 Index concluded that for daily trading investment, the combination of RF and MVF is superior in terms of excess return both with and without transaction fees. Chen et al. (2021) proposed a two-phase portfolio selection model first to predict the next stock prices using extreme gradient boosting (XGBoost) and then allocate the investment proportion of the portfolio according to Markowitz's mean–variance (MV) model. Their research focused on improving the prediction performance of XGBoost using an improved version of the firefly optimization algorithm (IFA). Inspired by Particle Swarm Optimization (PSO), the authors introduced the global best solution into the firefly algorithm (FA) to incorporate the global optimal solution into the movement of fireflies besides the intrinsic brightness criteria in the standard FA. They applied their IFA algorithm to find the best hyperparameters for XGBoost resulting in an enhanced cumulative return of the combination of IFAXGBoost and MV to 94% compared with 31% for the XGBoost + MV model. The Fuzzy logic is suitable for expressing vague and uncertain situations, this is exactly what happens for the prediction of future returns in stock markets. Therefore, Authors (Liagkouras and Metaxiotis, 2018) represented the return of the examined assets as trapezoidal fuzzy members. Their experimentation on historical data from FTSE-100 demonstrated that the proposed Multi-period Fuzzy Portfolio Optimization Algorithm (MFPOA) outperforms the benchmark algorithms (NSGAII and MOEA/D) both in terms of portfolio performance and mean computational time. Portfolio construction has also been investigated by Kosc et al. (2019) to study the efficiency of momentum and contrarian strategies compared with EqW and McW reference portfolios. The authors have also calculated the B&H strategies for the S&P500 index as a benchmark for investment strategies comparison. The authors concluded that the short-term contrarian effect is present and quite strong in the cryptocurrency market. However, there is no direct proof of a similar momentum effect. They have also shown that the momentum and contrarian investment strategies result in an abnormal risk-weighted rate of return compared with the S&P500 B&H strategy. LSTM networks have also been extensively used for investment strategy generation in stock and cryptocurrency markets. The authors (Michańków et al., 2022) explored forecasting BTC and S&P500 index by applying LSTM networks on the simple returns with the daily, 1-h, and 15-min frequency to construct B&S strategy. Having tested the five suggested hypotheses, the authors concluded that the efficiency of LSTM in AIS strongly depends on the hyperparameter optimization process, the way the model is constructed, and how the estimation process is carried out. The results showed that while the outcome depends on asset types and the corresponding frequencies, it is not robust to initial assumptions. Furthermore, they concluded that the proper loss function influences the model estimation process.

A recent comprehensive survey has discussed the challenges related to suitable portfolio selection and optimization based on nature-inspired Particle Swarm Optimization (PSO) and concluded that the application of optimization techniques such as PSO might be effectively combined with the models for enhancing the performance of portfolio allocation (Thakkar and Chaudhari, 2021). However, Multi-objective optimization techniques such as those proposed by Chen et al. (2020), Li et al. (2021) and Liang et al. (2022) can be exploited to enhance the portfolio optimization process by simultaneously selecting more informative features and stock forecasting metrics such as Share ratio. Algorithmic investment strategies are of great importance for portfolio formation. In an attempt to study the profitability of quantitative investment based on ML algorithms, authors (Grudniewicz and Ślepaczuk, 2021) exploited technical indicators as input to some ML models generating trading signals. The authors employed their introduced adjusted Information Ratio along with other risk and return measures to compare the strategies on nine different indices, including WIG20, DAX, and S&P500. They concluded that the resulting investment strategies are superior to the B&H benchmark for all studied indices in terms of adjusted Information Ratio.

The literature for financial time series shows that researchers are actively looking to identify more informative features to enhance the prediction tasks for financial time series. Noise removal from price time

series or return data by applying discrete wavelet transform (DWT) has been widely exploited (Arévalo et al., 2018; Kao et al., 2013; Lee et al., 2021; Nobre and Neves, 2019; Wu et al., 2021). Recently, authors (Leung and Zhao, 2021) exploited the Hilbert–Huang Transform, HHT, to generate a collection of features and integrated them into machine learning and deep models to predict the S&P500 index, Gold index, Volatility Index (VIX), and 10-year treasury yield (TNX) through regression task experimentations. They showed how HHT features could enhance the performance of the learning models. Applying HHT in the analysis of stock market data has not received much attention and is limited to a few studies (Bogeh, 2021; Ke et al., 2014; Li and Huang, 2014; Ostad-Ali-Askari et al., 2017b; Valderrama, 2021; Xuan et al., 2015).

3. Materials and proposed approach

3.1. Hilbert-Huang transform

Financial time series are nonlinear and non-stationary. Hence, the Hilbert–Huang transform is most suited to be adopted as a robust two-step signal analysis tool. The method was first introduced by Huang et al. (1998). The first step of the transform is to decompose the input signal into a finite sequence of intrinsic mode functions, IMF, and a residual term. This is done using Empirical Mode Decomposition, EMD.

$$X(t) = \sum_{j=1}^{n} IMF_j(t) + r_n(t) \tag{1}$$

IMFs are non-local oscillating symmetric functions. Since each IMF_j has an instantaneous phase $\theta_j(t)$ which is oscillating by it, and an instantaneous amplitude $A_j(t)$, it is possible to express IMFs as follows:

$$IMF_j(t) = A_j(t)\cos\theta_j(t) \tag{2}$$

The second step is to apply the Hilbert transform over the IMFs and produce an orthogonal pair for each $IMF_j(t)$ that is phase shifted by 90 degrees. The original IMFs and their corresponding orthogonal pairs are used to extract the instantaneous frequency, amplitude, and phase of each $IMF_j(t)$. The Hilbert transform is expressed as follows:

$$Y_j(t) = H[IMF_j(t)] = \frac{1}{\pi}\int_{-\infty}^{+\infty} \frac{IMF_j(\tau)}{t-\tau} d\tau \tag{3}$$

where $Y(t)$ forms the complementary imaginary part of $X(t)$. Having the real function and the corresponding imaginary complement, we can form the function $Z(t)$ as follows:

$$Z(t) = IMF_j(t) + iY_j(t) \tag{4}$$

$Z(t)$ is expressed in polar coordination as follows:

$$Z(t) = A(t)\, e^{i\theta(t)} \tag{5}$$

where $A(t) = \|Z(t)\|$, and $\theta(t) = arg Z(t)$. According to the slow modulation condition, $A(t)$ is the instantaneous amplitude of $X(t)$, and $\theta(t)$ is the corresponding instantaneous phase. The corresponding instantaneous frequency of $X(t)$ is defined as follows:

$$F(t) = \frac{1}{2\pi}\frac{d}{dt}\arg(Z(t)). \tag{6}$$

3.2. Extreme gradient boosting (XGBoost)

In machine learning, a weak learner is a classification model that can perform marginally better than random guessing. Authors (Schapire, 1990) developed boosting in a successful attempt to answer the question "Can a set of weak learners create a single strong learner?" proposed by Kearns et al. (1989). The main idea behind most boosting algorithms is to iteratively apply a weak learner to training data and assign more weights to misclassified observations to find a new decision stump for them. Finally, all learned models are aggregated to form a strong learner able to correctly classify all training samples. Therefore, a decision tree ensemble model with K additive functions is used to predict the target.

$$\hat{y}_i = \phi(\boldsymbol{x_i}) = \sum_{k=1}^{K} f_k(\boldsymbol{x}_i), \quad f_k \in FK(x_i, x_j) = \phi(x_i) \odot \phi(x_j). \tag{7}$$

In Eq. (7), $\boldsymbol{x}$ is the m-dimensional input feature vector, y is one-dimensional target vector forming the n cardinality sample space $D = \{(\boldsymbol{x}_i, y_i); |D| = n, \boldsymbol{x}_i \in \mathbb{R}^m, y_i \in \mathbb{R}\}$. The space of classification and regression trees (CART) with T leaves in each tree, is also indicated by $F = \{f(\boldsymbol{x}) = w_{q(\boldsymbol{x})}; q : \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T\}$ where f_k represents an independent tree structure q whose leaf weights are w. In order to classify the observations, the decision rules in the trees are applied to calculate the predicted target by summing up all w_i, the weights in the corresponding leaves. Eq. (8) shows the objective function used to learn the set of functions used in the model.

$$l(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k),$$
$$where\ \Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2. \tag{8}$$

As can be seen from Eq. (8), the model is trained in an additive manner instead of using traditional optimization methods in Euclidean space. Hence, while the adaptive boosting techniques try to weigh misclassified samples more, in gradient boosting, base learners are generated sequentially so that the current model is always more effective than the previous one by ameliorating a loss function. Therefore, the objective function to be optimized is modified to include greedily adding f_t.

$$l^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\boldsymbol{x}_i)) + \Omega(f_t),$$
$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2. \tag{9}$$

XGBoost (Chen and Guestrin, 2016) is a highly enhanced version of gradient boosting. It mainly aims at increasing computation speed and efficiency since the gradient boosting algorithm analyzes the datasets sequentially, resulting in a very low-performance rate. Furthermore, XGBoost supports parallelization by creating decision trees in a parallel way, like the Random Forest. It also exploits distributed computing methods to evaluate large and complex models and uses Out-of-Core computation to analyze large and varied datasets. Using cache optimization is another technique used in XGBoost to achieve a higher level of optimal resource utilization.

3.3. Portfolio formation

The Modern Portfolio Theory (MPT) is based on the mean–variance (MV) model proposed in Markowitz (1952), a.k.a. Markowitz model. The model provides a solution to the portfolio formation problem by finding the optimum weights of each asset within a portfolio. Markowitz's model is based on the fact that higher-return portfolios with a constant risk level and lower-risk portfolios with a constant return are preferred among most investors. Therefore, the MV model is the solution to the following quadratic program optimization:

$$\min_w w^T \sum w$$
$$subject\ to\ \mu^T w = r\ and\ 1_D^T w = 1 \tag{10}$$

where $\sum$ is the covariance matrix of assets that we want to form the portfolio from, w is the weight vector we are looking to find for the optimized portfolio, μ indicates the mean return of the portfolio, r is desired expected return we want to achieve. $1_D^T w = 1$ constraints the weights of the D-asset portfolio not to exceed 1. To form the optimum portfolio, we first need to find the returns and risks of all assets under the study. Hence, the output of the XGBoost as our classification model is used to evaluate the performance of each stock. The required evaluation is performed using the trading module. The trading module works in this way that an initial balance is considered to be available

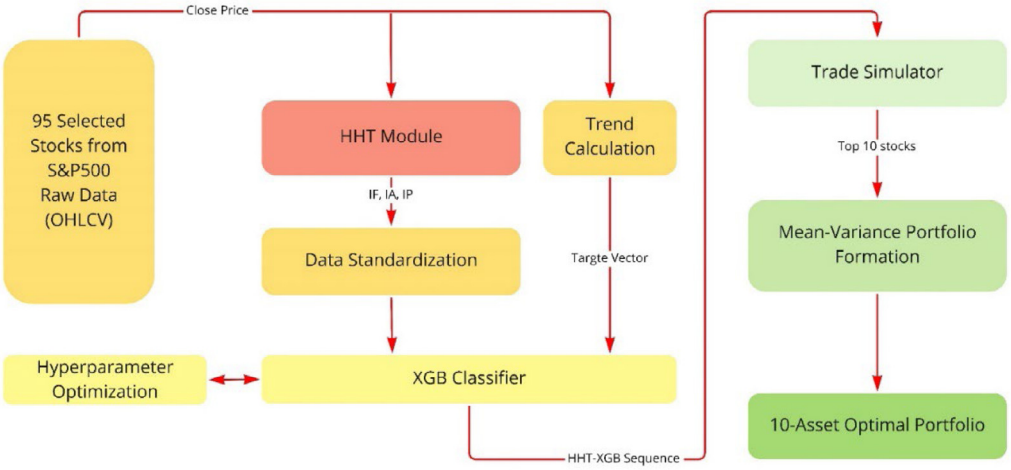


Fig. 1. Proposed approach. HHT Module is at the heart of the model to extract a more informative feature set. The classification output is the next close price direction which is used as a signal for evaluating the performance of the corresponding stock in the Trade Simulator module. The top 10 stocks are selected to form a 10-asset optimally weighted portfolio using Markowitz's model.

at the time t_0 to buy a specific stock. As soon as the system receives the buy signal at a time t_i, the whole available balance is used to purchase as many shares as possible from the target asset. At a later time t_j, the trading module receives a 0 in the signal generated by the classification module; hence it issues the Sell command. Therefore, the whole shares are sold using Open Price at the time $t_j + 1$, and the money received from this trade is summed into the available balance forming the new capital. This process is continued until the end of the study period. The trading module also studies the performance of the B&H strategy to compare the result of the trading with the holding mechanism in terms of trading performance metrics. For the B&H strategy, the first entry point of the B&S strategy is used to buy an asset, and the last exit is accounted for as the time that the B&H strategy ends with selling the shares of the corresponding asset.

The primary goal of every investment algorithm is to maximize the returns while considering risk and time spent with capital invested in the market. We compute the cumulative rate of return to evaluate the performance of each stock with our trend prediction model. To do this, we first compute the rate of return, RoR_t for some time $t \in [i, j]$ as follows:

$$RoR_t = \frac{Capital_j - Capital_i}{Capital_i} \times 100 \tag{11}$$

Then the cumulative return is calculated easily by summing the last RoR_t values.

The next metric is maximum drawdown (MDD), which is the maximum amount of loss from a peak to a trough in a specific period before a new peak is attained, which indicates the downside risk over the period.

$$DD_t = Max_{k\in(i,t)}\left(RoR_k\right) - RoR_t,$$
$$MDD = Max_{k\in(i,j)}\left(DD_k\right);\ i < k < j. \tag{12}$$

The third metric is the Sharpe ratio which is used as a measure of the performance of the system in maximizing the expected return while minimizing the risk. The Sharpe ratio is defined as the ratio of the excess expected return to the standard deviation of the return.

$$Sharpe\ Ratio = \frac{\mu - R^f}{\sigma} \tag{13}$$

where the R^f is the risk-free rate, μ is the mean of the one-period simple return of an asset between dates $t-1$ and t, and σ is the corresponding standard deviation.

3.4. Proposed approach

The overall architecture that we have designed to form our proposed approach in this study is shown in Fig. 1. HHT Module is used to extract the instantaneous frequency, amplitude, and phase of the close price time series for each of the 95 stocks selected from the S&P500 market. The Data Standardization module then processes the feature set. Turning all input features into the same scale is essential for machine learning models to work properly. For the data that does not follow a bell-curved distribution, the preference is to be normalized; however, our experimentation shows that there is not so much difference between normalization and standardization in our proposed approach. Hence, we standardize data because it outweighs normalization in the sense of the impact of outliers. Therefore, the features are standardized by subtracting the sample mean and dividing by the sample standard deviation, turning the feature space to be of the distribution with zero mean and unit standard deviation.

$$X^* = \frac{X - \mu_X}{\sigma_X} \tag{14}$$

The target vector is generated by the Trend Calculation modules using close price data. If the next close price for the time t_{i+1} is higher than the time t_i then the target vector value tv_i is set to 1; otherwise, 0. The XGB Classifier generates the predicted sequence of 1s and 0s using by Trade Simulator. Trading performance metrics introduced in Section 3.3 are used to select the top 10 stocks for the task of portfolio formation.

4. Results and discussion

All algorithms have been implemented in python 3.7.9 using Scikit Learn for machine learning tasks. The experimentation environment is a 3.80 GHz Intel processor with 32 GB RAM and a Geforce-3080 12 GB GPU. The environment is operated by Ubuntu 20.8.4.

S&P500 is a well-diversified by-sector market which means the systematic risk of the market itself is around one due to the presence of major sectors, including technology, industry, finance, consumer, etc. Hence, we randomly selected a vector of 95 different stocks from the different sectors. The historical data is retrieved from finance.yahoo.com, including 2893 data points from 2011-01-03 to 2022-06-29. The raw financial data consists of 6 columns, Date, Open, Close, High, Low, and Volume, OHLCV. The data set is split into three parts, training, validation, and test set. The training and validation sets contain the first 85 percent of data. The last 15 percent of data points are used for running the back-test procedure that consists of 434 days

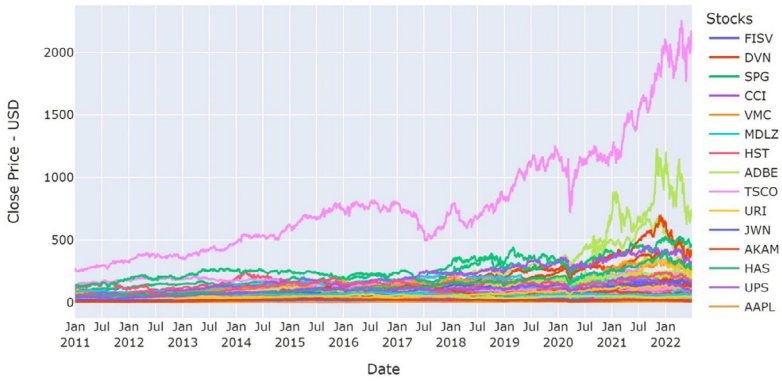


Fig. 2. Close price data for the randomly selected 95 stocks from S&P500.

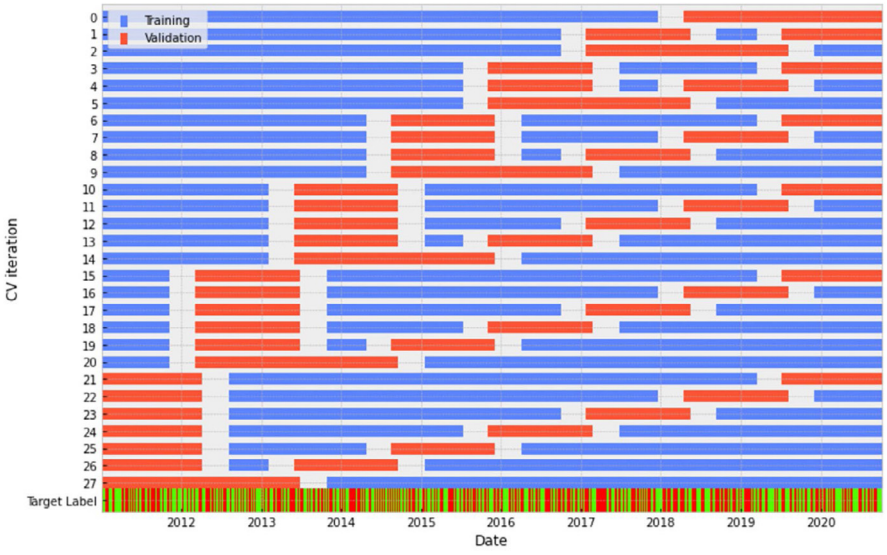


Fig. 3. Combinatorial embargoed purging cross-validation. The combination of 2 validation sets out of 8 splits results in a 28-fold CV in a combinatorial way. Embargoed time delta is set to 0.01 of the length of the training period to prevent extra plausible leakages for test sets that are followed by a training set.

between 2020-10-08 and 2022-06-29. The whole data range for the close price is shown in Fig. 2.

To avoid the look-ahead bias due to the contamination of validation splits by the leakage from the training data, we have applied the combinatorial embargoed purging cross-validation technique introduced in Marcos López De Prado (2018). The 28-fold cross-validation procedure is represented in Fig. 3, where the dataset has been split into six training sets and two validation sets. Using the combinatorial cross-validation technique, we ensure that the different parts of the data sets are used as validation splits, avoiding the multiple test selection bias while validating the training procedure. As can be seen from Fig. 3, while the validation sets can be distinguished into 8-sets groups, there are seven different paths for each group of validation sets. Target labels are also shown in red and green colors for negative and positive classes, respectively.

We have used the extreme gradient boosting algorithm as the base classifier to predict the next change in the close price direction. The model is trained with two different feature vectors. The first feature vector is the raw financial data acquired from finance.yahoo.com. The second model is trained using a feature vector formed by first decomposing the close price using HHT and then taking the first five corresponding frequency, amplitude, and phase components into account. To apply HHT to the Close price data, we first need to decompose the input series using EMD and extract the corresponding IMFs. As can be seen from Fig. 5-a, the IMF components of the selected stock, INCY, from our stock vector, represent the 7 IMFs, IMF-1 to IMF-7, with the decreasing frequency for each mode. The overall trend of the corresponding time series is depicted by the residual term, Residual. The lower frequency modes capture the overall changes better, while the finer details of the series oscillation are better shown within higher frequency components. For example, the sharp decline of the INCY Close price in early 2016 is shown as notable peaks around the period on the IMF-3, IMF-4, IMF-5, and IMF-6 components. The IMF components summed to the same original time series showing that all information in the Close price data is precisely captured in individual IMFs (Fig. 4). The instantaneous frequency, amplitude, and phase of each IMF extracted from HHT are shown in Fig. 5-b, c, and d respectively. IF-x, IA-x, and IP-x represent the instantaneous frequency, amplitude, and phase of the respective IMF-x. These modulation components show the slow modulation properties of the respective intrinsic decomposition mode. Therefore, as we go down from the 1st component of each modulation to the last one, the higher-order modes depict a more transparent long-term trend than the lower orders. Hence, to enhance the quality of trend prediction, we select the first five components of IF, IA, and IP and combine them into a single data frame representing our input feature vector to the classification model.

The classifier used in this study is XGBClassifier, an instance of the XGboost python API. To find the best tuning hyperparameters for the model, we have trained the model under the callback of the Optuna optimizer in 1000 trials. The parameters used for the model are shown

Fig. 4. Close price trend of INCY and the corresponding trend computed by summing the respective IMF components, IMF-Summed. IMF components contain all high and low-frequency data of the original series.

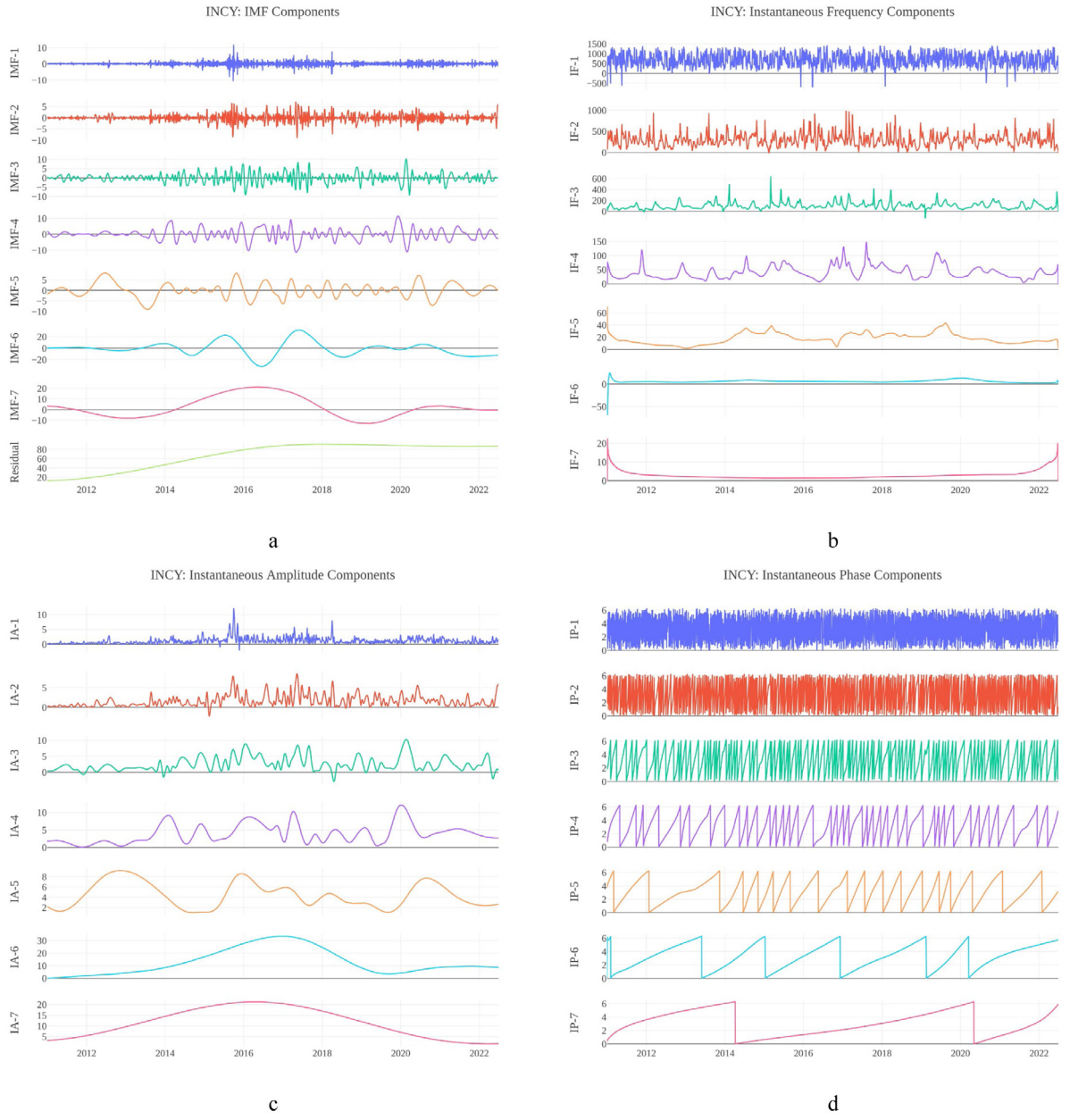


Fig. 5. Empirical mode decomposition components and the corresponding Hilbert–Huang transform frequency, amplitude, and phase components. IF-x, IA-x, and IP-x plots are of the same color as the corresponding IMF-x (x is from 1 to 7). The components are of the same INCY signal as in Fig. 4. The same procedure is repeated for all other stocks to generate the feature vector. Similar charts for all other stocks are available as supplementary materials.

under the Hyperparameter column in Table 1. The XGBClassifier is then used with the best parameters found for the two case studies, first with the raw financial data as input features and second when the input features are instantaneous frequency, amplitude, and phase

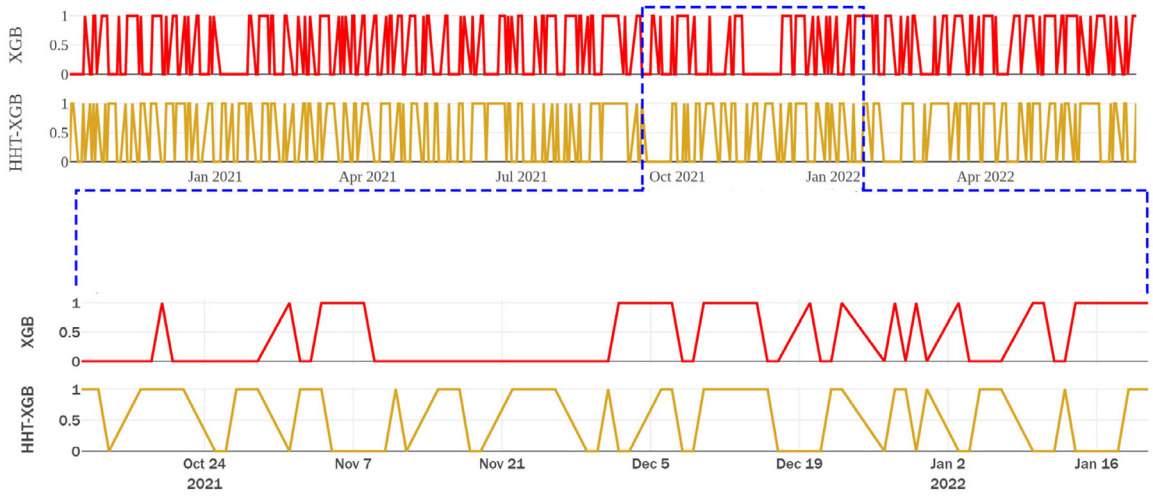


Fig. 6. The sequence of 0s and 1s predicted by XGB and HHT-XGB models is used to compute the corresponding portfolio performance metrics. The excerpt of the two sequences shows that HHT-XGB has predicted more 1s compared to XGB. More 1s means more trading opportunities.

Table 1
Best tuning hyperparameters for the classification model. XGB stands for the XGBClassifier with raw financial data. HHT-XGB stands for the model with the HHT extracted features.

Hyperparameter	Range	Best value XGB	Best value HHT-XGB
n_estimators	50–1000	111	786
max_depth	3–18	8	6
sampling_method	'uniform', 'gradient_based'	gradient_based	gradient_based
colsample_bytree	0.1–0.99	0.97	0.9
learning_rate	0.01–0.2	0.16	0.11
reg_alpha	0.00001–0.01	0.002	0.0014
reg_lambda	0.00001–0.01	0.005	0.0014
gamma	0.1–1.0	0.84	0.9

components. The classifier is trained to learn the sequence of 0s and 1s obtained by downs and ups of the next close price.

The performance of the classification model for a single stock, INCY, is shown in Table 2. The numbers show that the HHT-XGB is better at learning more ups and downs in the next close price. From the XGB column, we find that the model's precision and recall for predicting 0s and 1s from the raw financial data is just the same as a random guess, 50 percent. Nevertheless, the model's performance is increased to 83 percent for the Macro Avg Recall and Precision metrics when the input feature is changed to contain the instantaneous frequency, amplitude, and phase components of Close price. We have then studied the same model for all selected stocks, and the corresponding average performance metrics are shown in Table 3. Again, we see that the overall performance for the HHT-XGB model is 81 percent in terms of accuracy metric, whereas the XGB model remains at the same percentage of 51, just as a random guess. The classification result for the back-test period is shown in Fig. 6. As it can be seen from comparing the plots, the HHT-XGB model has an enhanced power to detect more reliable trading opportunities in terms of better predicting the accurate forthcoming up trends in the Close price series.

The experimentation continues by using the signal generated by the classification models. XGB and HHT-XGB produce two sequences of 0s and 1s to be used as the basis of trading strategy. Trading is simulated to see whether our proposed approach can produce better profits compared to the base classifier and compared to the B&H strategy or not. As discussed in Materials and Methods, the sequence generated by our proposed model is used to buy and sell assets. The first 1 in the sequence is used to buy an asset with the next open price, and the first 0 is used to sell the whole shares of that asset with the next open price.

The consecutive 1s between the first 1 and the first 0 are discarded. The same is done for the consecutive 0s between the first 0 and the next 1. The same scenario is repeated for the first 1 of the second batch and the first 0 of the second batch until reaching the end of the sequence. To simulate the B&H strategy, we assume the asset is bought as the first 1 appears in the sequence, and when the first 0 after the last 1 of the sequence appears, the stock is sold. The cumulative returns of the above-mentioned strategies are shown in Fig. 7. The results are shown for 12 stock codes; however, the rest is available within supplementary materials. One can easily deduce that while the XGB and B&H performance are interchangeably competing not to draw down to negative returns, the cumulative return of HHT-XGB is always superior to the formers by a considerable percentage. Even in the case of a stock such as CLX, whose cumulative return for both XGB and B&H is compounded to a substantial negative percentage, the HHT-XGB shows an outstanding performance of more than 50 percent. To show that the same scenario holds for all the other stock codes not shown in Fig. 7, we have presented the average cumulative return for all stocks under the same strategies in Fig. 8. The average cumulative return for HHT-XGB is compounded to more than 50 percent while XGB and B&H end in 5.08 and 4.7 percent respectively. Annual Sharpe ratio, SR, and maximum drawdown, MDD, are shown for XGB and HHT-XGB in Table 4. The results are shown for all stock codes. The risk-free rate for the back-test period is 0.24, found by taking the average

Table 2
Classification report for HHT-XGB and XGB classifiers for the INCY stock. The support columns show the number of 0s and 1s, respectively. The accuracy of HHT-XGB shows a 64% improvement compared to the XGB model.

	HHT-XGB			XGB			Support
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
0	0.88	0.79	0.83	0.52	0.52	0.52	225
1	0.79	0.89	0.84	0.48	0.48	0.48	209
Accuracy	0	0	0.83	0	0	0.5	434
Macro avg	0.84	0.84	0.83	0.5	0.5	0.5	434
Weighted avg	0.84	0.83	0.83	0.5	0.5	0.5	434

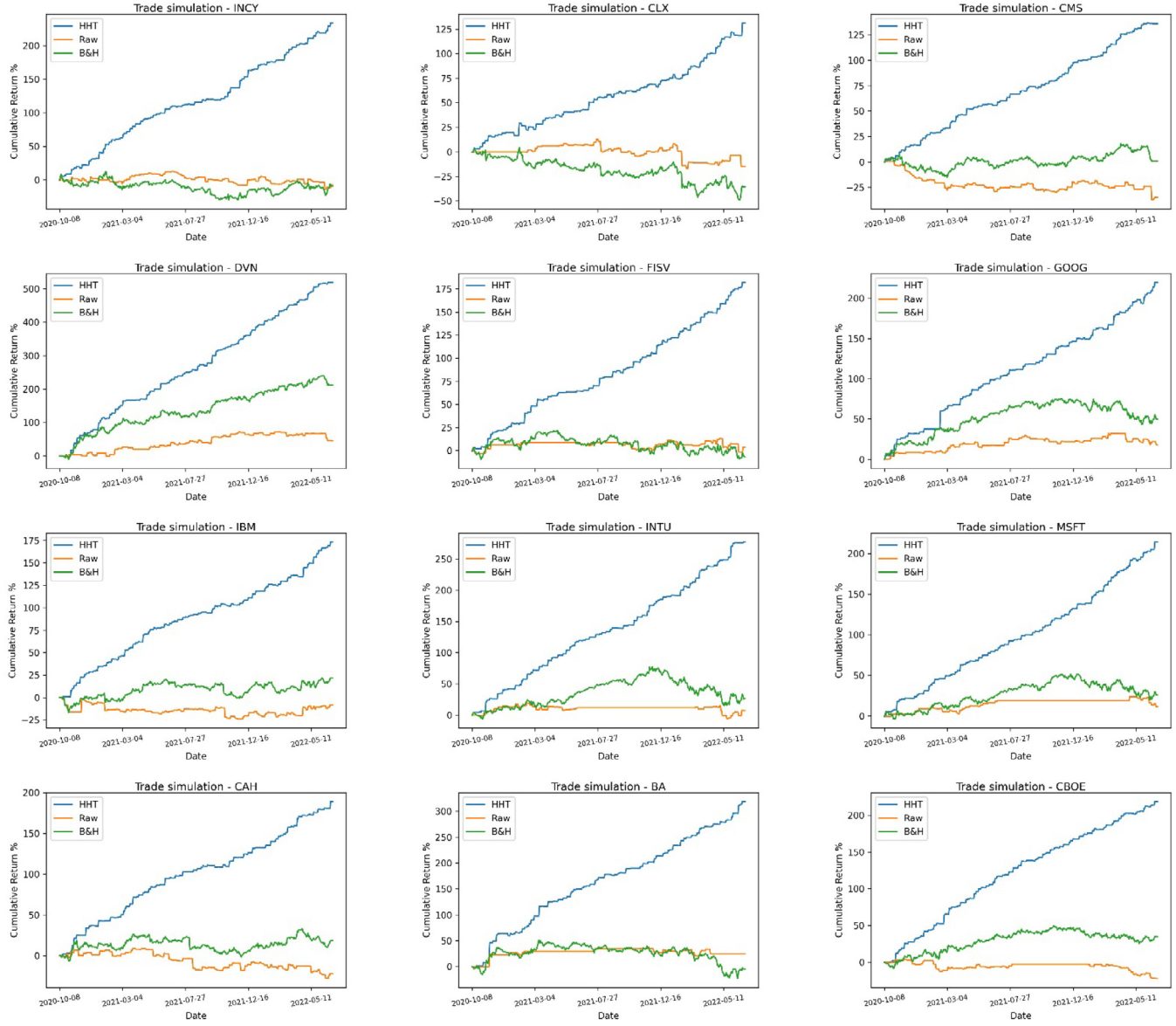


Fig. 7. Cumulative returns for 12 different stocks out of the whole dataset. All plots are available within supplementary materials.

Table 3
Average classification performance metrics for all stocks. The overall performance of HHT-XGB is 58.8% better than the XGB model in terms of accuracy metric.

	HHT-XGB			XGB		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
0	0.81	0.8	0.81	0.5	0.5	0.49
1	0.81	0.82	0.82	0.52	0.52	0.51
Accuracy			0.81			0.51
Macro avg	0.81	0.81	0.81	0.51	0.51	0.5
Weighted avg	0.81	0.81	0.81	0.51	0.51	0.5

of the 3-Month Treasury Bill Secondary Market Rate retrieved from https://fred.stlouisfed.org/series/tb3ms (Fig. 9). Although the risk-free rate is given as a 3-month rate, it is annual. Therefore, we need to turn the rate into a daily rate divided by 252. While the annual SR for XGB is negative for a considerable number of stocks, the metric is of absolutely great performance for the same stock using HHT-XGB. The SR shows the excess return from the excess risk taken by the corresponding strategy. The ratio indicates a good and very good investment if its value equals one or two. A value greater than or equal to 3 proves the

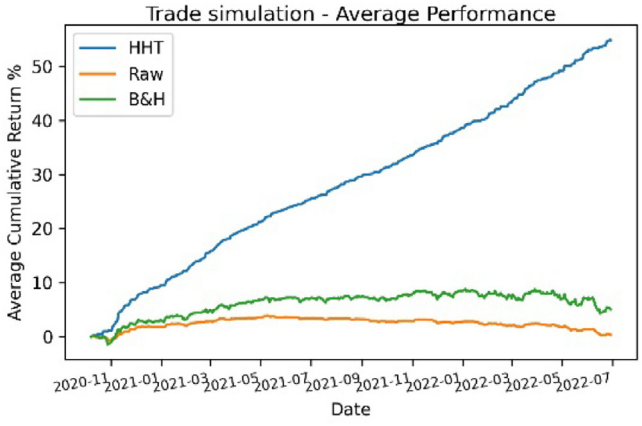


Fig. 8. Average cumulative return. The plots represent the average performance of trading and B&H strategies. The trading is carried out using the signal generated by the output of the classifier. HHT is superior to both XGB and B&H strategies.

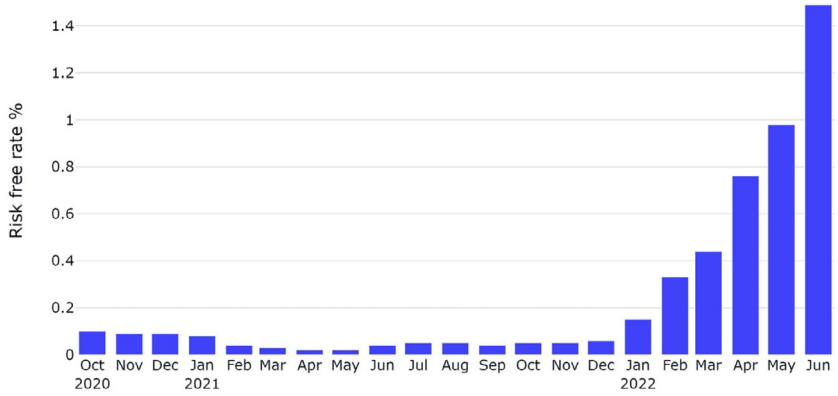


Fig. 9. Risk-free rate. The values are available at https://fred.stlouisfed.org/series/tb3ms.

Table 4
Performance of trading stocks. SR and MDD metrics are computed for the back-test period using signals generated by two classifiers, XGB and HHT-XGB. HHT-XGB is superior in terms of both SR and MDD. SR is shown as an annual rate.

Stock	XGB		HHT-XGB		Stock	XGB		HHT-XGB	
	SR	MDD	SR	MDD		SR	MDD	SR	MDD
INCY	−0.59454	25.01035	4.74415	2.40667	APH	−0.53513	25.35556	4.26526	4.24037
GILD	−1.03647	26.52002	4.05094	2.03078	PDCO	0.15485	25.75856	4.4254	4.47938
DXC	0.39422	37.2121	4.61044	5.88802	FIS	−1.97346	52.55593	3.88453	8.27333
AFL	0.08111	9.84492	4.33501	1.96935	URI	0.00874	32.5669	4.8799	1.88249
ARE	−1.54986	32.80145	4.06012	4.19696	CBOE	−1.56447	23.94137	5.0024	1.54638
AMD	−1.30693	41.98717	5.25439	3.0207	FISV	−0.32113	14.70471	4.19305	2.41418
WYNN	−1.07451	59.42732	4.1816	4.77908	BA	0.27456	13.71541	3.97682	4.55198
IBM	−0.58852	23.47955	4.51287	2.73167	UAA	−0.12786	48.17492	4.39194	5.06781
ATVI	−1.85292	37.65906	3.47161	2.83436	CVS	−0.39517	21.29296	4.44182	3.89278
AES	−0.67486	42.14236	4.95584	5.43748	WU	−1.66097	37.73646	4.6365	4.05745
GPC	−0.27423	15.53979	5.19681	1.16852	SLB	0.56335	32.035	5.16771	3.59383
PRU	−0.73815	36.44755	4.68627	1.93095	AIV	0.29606	39.71986	4.75954	4.0575
HST	0.23432	41.19735	4.60758	2.74768	VLO	0.08444	36.62783	5.43985	1.6855
WMB	0.35502	12.85834	5.21084	3.40906	MU	0.45616	21.55591	5.32146	4.69315
GOOG	0.18113	13.71479	4.07126	4.79201	CAH	−0.87843	33.14561	4.30603	3.04612
SRE	−1.6935	24.64551	4.46404	2.69482	AZO	0.2686	13.2236	4.77998	5.71373
HSIC	−0.30856	19.85764	4.32645	3.74357	CCI	−1.44275	32.11952	4.18053	3.33771
UNP	−1.22607	20.17496	4.44423	1.65071	MCO	−0.63429	16.00621	4.42889	4.90648
UPS	−0.46043	24.88948	3.71032	2.59813	RHI	−0.43228	40.41962	4.36996	2.56063
K	−1.08681	17.25737	3.98677	2.76337	UAL	0.07892	19.89698	4.7067	7.15515
LYB	−1.01591	39.47547	5.34359	2.25691	CB	−0.07762	15.65684	4.97601	2.01926
MDT	−1.96808	34.51117	4.28722	2.35797	VMC	−1.17703	34.87315	4.98069	2.57491
COF	−0.28586	39.0095	5.1163	4.397	GD	0.08065	10.91422	5.03012	1.88993
HAS	−0.77171	17.74389	4.75523	1.83529	AAP	−0.74785	35.68655	5.05383	3.85102
CSX	0.03209	10.10758	4.51889	2.64905	NEE	−0.11034	15.46928	4.36639	4.808
GT	−0.1363	52.05348	5.78372	1.74647	ABT	−0.09734	12.50032	4.51277	2.01791
EBAY	−0.37413	22.24882	4.27909	4.69824	WFC	1.15814	17.31168	5.42447	2.49119
KSS	0.4173	51.6949	4.87474	3.13491	DIS	−0.63336	44.85292	3.92134	3.07295
AJG	0.57602	10.85712	4.55579	1.82503	SNA	−1.04646	34.48148	4.98211	5.81285
JWN	1.19262	31.5285	3.68238	13.39311	MAR	−0.04824	20.6299	4.315	4.55063
SBUX	−0.37296	20.92938	4.37316	3.2079	JPM	−0.05378	21.604	4.17064	2.87386
IT	0.09777	35.25092	3.9939	5.82067	CLX	−0.78637	27.81819	2.98427	7.01791
PH	0.07123	18.87763	4.51539	1.17464	KIM	−0.25748	34.85946	4.01906	1.94473
DVN	0.55416	25.56508	5.3194	5.08162	HBI	−1.4861	60.28773	4.21434	4.66704
MMM	−1.19532	30.87203	4.08996	4.8371	RSG	0.5993	5.56792	3.71448	2.03712
SPG	−0.14825	41.74872	4.49649	2.66773	ETR	−1.12384	20.2504	4.49662	3.93308
AKAM	−0.34505	17.64164	4.04587	4.68169	ADBE	−0.23552	15.16365	5.0195	2.35759
IP	−0.88163	32.48751	4.72884	1.60883	COO	−0.86662	26.97348	4.53691	4.1232
CPB	−2.01408	38.89325	4.38642	1.87103	INTU	−0.19013	23.16099	4.77606	1.92182
AAPL	−1.21438	22.50591	4.6811	2.95118	SLG	−0.29896	43.34036	3.87619	10.15006
NEM	−1.08306	29.87142	4.77975	2.71634	GWW	0.01791	10.09781	5.04653	1.24454
KMX	−1.06944	43.06524	4.21326	4.71383	CMS	−1.64577	33.46386	4.47216	2.11306
TSCO	−0.20131	16.53957	4.42472	3.49575	MDLZ	−1.41575	23.93721	4.15656	1.68111
SYY	0.3769	15.2886	4.73296	1.83339	AMZN	−0.90985	42.27224	3.97112	2.6346
VNO	−0.38604	36.31865	4.03301	3.53466	INTC	0.30145	15.80143	4.82881	2.79946
ACN	0.54055	20.63165	4.51932	4.41465	TSLA	0.75739	24.27399	5.19812	4.6153
V	−0.33153	16.62438	4.01414	5.30369	MSFT	−0.07879	11.9732	5.07381	2.48512
QCOM	0.15102	18.53448	4.67985	2.34962	**Average**	**−0.43325**	**27.55072**	**4.524023**	**3.538909**

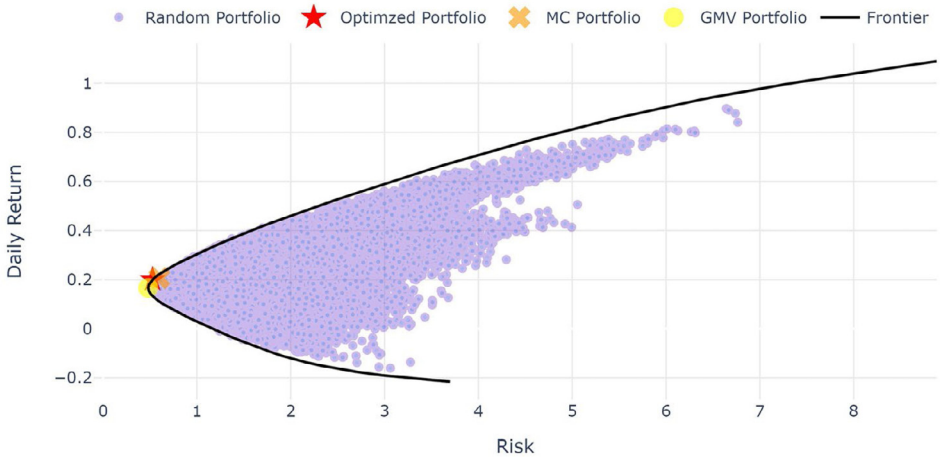


Fig. 10. Markowitz bullet for HHT-XGB. SLSQ optimized portfolio is on the efficient Frontier.

investment performance is excellent. Therefore, our proposed model proves to be superior since the average number of this metric for all stocks is 4.524023. The average MDD of HHT-XGB also shows a huge improvement in terms of decreasing by 87.1 percent compared to XGB.

The result of the trading simulation is compared with the state-of-the-art tri-state labeling algorithm proposed by Dezhkam et al. (2022). The proposed tri-state labeling algorithm has been employed to extract three distinct levels of changing trends, upward, downward, and fluctuating conditions. The authors have implemented their labeling algorithm using LSTM, GRU, XGBoost, and SVM, and the SR value for these implementations are 1.672, 1.675, 2.823, and 2.007, respectively. Therefore, the HHT-XGB strategy proposed in the current proposed approach outperforms the corresponding XGBoost realization of the tri-state labeling algorithm, while the XGBoost classifier for the tri-state labeling algorithm generates the most profitable B&S signals.

The last step in our experimentation pipeline is forming the optimum portfolio using both HHT-XGB and XGB models. The idea for selecting the assets to include in the portfolio is to sort the trading simulation results based on annual SR and then select the top 10 stocks with the highest rate. The outcome of this procedure is shown in Table 5. According to the simulation results, the two models introduce two distinct stock subsets as their portfolio inclusion candidates. We then exploit each group to find the optimum portfolio by maximizing the annual Sharpe ratio of the corresponding portfolio by solving an optimization problem by SLSQP, Sequential Least Squares Programming, method from SciPy. Therefore, we need to obtain daily returns and then compute the corresponding covariance matrix of each group to form the optimization function. In Table 5, the column Mean-Return shows the corresponding daily returns for each stock under two groups of XGB and HHT-XGB. The covariance matrices of two groups of stocks selected for portfolio selection are shown in Tables 6 and 7.

Portfolio optimization is the task of not taking risks than it is necessary. Therefore, the best choice among all portfolios with the same return is one with the minimum risk. In other words, if we need to select a portfolio from a collection of same-risk options, the portfolio with the highest return is the best choice. The portfolios with these properties form the Efficient Frontier. Along with solving a quadratic programming optimization problem to obtain optimum portfolio weights, we also run Monte Carlo, MC, simulation to explore 10 thousand random portfolios for finding the best excess return. In both approaches, we let the portfolio formation include the short-selling weights since short-selling is allowed over the S&P500 stocks through S&P500 ETF. The results of both SLSQ and MC simulation have shown in Table 8. While the most weights under the SLSQ method for both HHT-XGB selected assets are positive, the MC simulation finds some short position opportunities to take into account in portfolio formation. The portfolio

Table 5
Assets selected for portfolio formation. The assets are sorted based on SR and then the top 10 are selected. The Mean Return column shows the average daily return for each stock code. Sharpe ratio values are annualized.

HHT-XGB			XGB		
Stock	Mean-Return	Sharpe ratio	Stock	Mean-Return	Sharpe ratio
GT	0.2611	5.78372	JWN	0.1379	1.19262
VLO	0.292	5.43985	WFC	0.0426	1.15814
WFC	0.1396	5.42447	TSLA	0.054	0.75739
LYB	0.2134	5.34359	RSG	−0.0723	0.5993
MU	0.1672	5.32146	AJG	−0.0693	0.57602
DVN	0.3562	5.3194	SLB	−0.0456	0.56335
AMD	0.2049	5.25439	DVN	−0.0565	0.55416
WMB	0.1514	5.21084	ACN	0.021	0.54055
TSLA	0.2626	5.19812	MU	0.0224	0.45616
GPC	0.146	5.19681	KSS	0.0389	0.4173

for the HHT-XGB found by SLSQ optimization also introduces a single asset with a short position, indicating that our research hypothesis is rejected. However, in terms of SR, the performance of the HHT-XGB strategy is superior to the XGB strategy.

Figs. 10 and 11 show the Markowitz bullet for HHT-XGB and XGB portfolio formation, respectively. The points inside the envelope are random portfolios formed by the MC simulation. The scatterplot's yellow dot indicates the Global Minimum Variance, GMV, portfolio. This is the portfolio at the very tip of the Markowitz bullet. GMV is the portfolio with the minimum risk out of all others. We found GMV by solving a quadratic programming problem to minimize the portfolio variance. The red star and orange cross are portfolios of SLSQ optimization and MC simulation, respectively. According to Figs. 10 and 11, both portfolios for HHT-XGB and XGB found through SLSQ optimization are located on the efficient frontier, implying that these portfolios are the optimums and there is no other portfolio giving us less risk. However, the MC simulation returned portfolios out of the efficient frontier, although those are so close to the SLSQ-optimized portfolios. This means that the MC simulation also produces reasonable results in case the optimization problem may not be possible to be carried out.

5. Conclusion

Machine learning tasks are tightly dependent on the quality of input features. When predicting the stock markets, the model's performance plays a crucial role in more profitable investment and trading strategies. Financial time series are noisy, while the information available in the series is so important to capture future trends. To the best of our knowledge, no research has been conducted to apply Hilbert–Huang transform to the classification task for the next Close price

Table 6
The covariance matrix of XGB selected socks.

	JWN	WFC	TSLA	RSG	AJG	SLB	DVN	ACN	MU	KSS
JWN	9.913	0.71	−0.03	0.037	0.066	0.043	−0.233	−0.07	−0.035	0.163
WFC	0.71	0.718	0.139	0.006	0.016	−0.025	0.045	0.033	0.085	0.079
TSLA	−0.03	0.139	2.991	0.039	0.016	0.007	0.006	0.058	0.093	0.019
RSG	0.037	0.006	0.039	0.114	0.002	0.012	−0.014	0.025	0.022	0.017
AJG	0.066	0.016	0.016	0.002	0.244	0.024	0.051	0.008	0.018	−0.009
SLB	0.043	−0.025	0.007	0.012	0.024	2.949	0.01	−0.013	0.011	0.055
DVN	−0.233	0.045	0.006	−0.014	0.051	0.01	1.379	0.025	−0.002	−0.133
ACN	−0.07	0.033	0.058	0.025	0.008	−0.013	0.025	0.423	0.023	0.056
MU	−0.035	0.085	0.093	0.022	0.018	0.011	−0.002	0.023	0.732	0.063
KSS	0.163	0.079	0.019	0.017	−0.009	0.055	−0.133	0.056	0.063	4.361

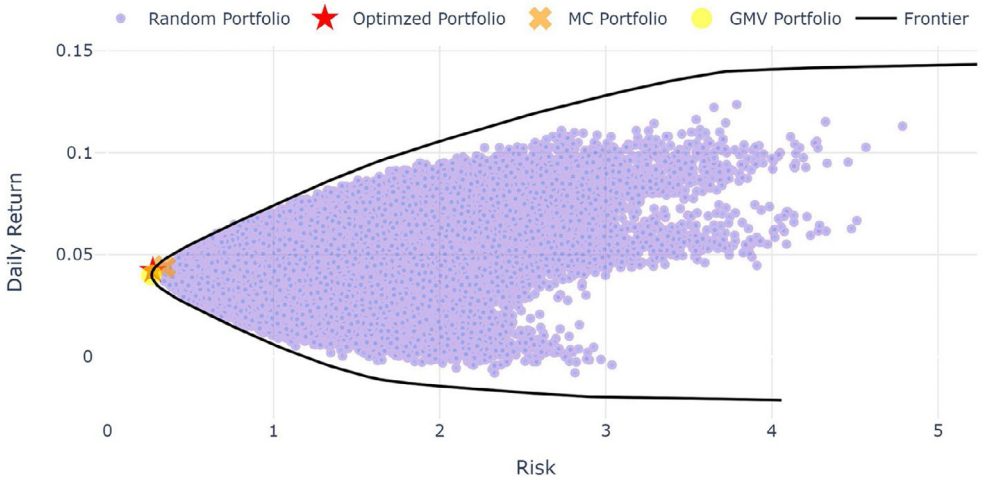


Fig. 11. Markowitz bullet for XGB. SLSQ optimized portfolio is on the efficient Frontier.

Table 7
The covariance matrix of HHT-XGB selected stocks.

	GT	VLO	WFC	LYB	MU	DVN	AMD	WMB	TSLA	GPC
GT	3.278	0.614	0.698	0.664	0.486	1.194	0.724	0.407	0.918	0.48
VLO	0.614	2.488	0.567	0.665	0.297	0.879	0.235	0.516	0.216	0.258
WFC	0.698	0.567	1.544	0.405	0.554	0.764	0.468	0.364	0.489	0.238
LYB	0.664	0.665	0.405	1.344	0.317	0.713	0.236	0.316	0.279	0.354
MU	0.486	0.297	0.554	0.317	2.016	0.558	0.619	0.236	0.519	0.16
DVN	1.194	0.879	0.764	0.713	0.558	3.898	0.65	0.605	0.552	0.404
AMD	0.724	0.235	0.468	0.236	0.619	0.65	2.727	0.269	1.018	0.136
WMB	0.407	0.516	0.364	0.316	0.236	0.605	0.269	0.685	0.226	0.174
TSLA	0.918	0.216	0.489	0.279	0.519	0.552	1.018	0.226	4.006	0.194
GPC	0.48	0.258	0.238	0.354	0.16	0.404	0.136	0.174	0.194	0.637

trend prediction; hence we framed a model based on the extracted instantaneous frequency (IF), amplitude (IA), and phase (IP) of the Close price data. The information extracted from higher-order modes of IF, IA, and IP enabled the classifier to better find true ups and downs within the Close price time series. The proposed approach was tested with a randomly selected dataset from S&P500, including 95 stocks. The simulation was done to compute the performance of the portfolio formation task using the trend changes sequence generated by the proposed model. The results showed that the proposed approach could achieve excellent performance in terms of portfolio performance metrics. It is worth mentioning that the back-testing was carried out in the poor-performing period of the S&P500, i.e. from January 8, 2021, to July 29, 2022. However, according to the Sharpe ratio, MDD, and cumulative returns values, the first hypothesis is maintained, suggesting that our HHT-XGB strategy is always superior to the benchmark strategies employed in this study. On the other hand, the second hypothesis of our research is rejected since the optimized portfolio for the HHT-XGB strategy also introduces a single short position to maximize the excess return to excess risk ratio.

To the best of our knowledge, this study is the first research work on exploiting Hilbert–Huang transform for the classification task to propose a model for portfolio formation. One possible direction of future studies is to apply the HHT on momentum and volatility technical indicators and extract these informative time series' IF, IA, and IP components. The more informative sub-series from the engineered feature set can be selected with an appropriate dimension reduction technique to feed machine learning or deep learning classifiers. The result can be used for the same task of portfolio formation to see how the proposed models will perform, receiving more information from the underlying patterns.

Abbreviations:

The following abbreviations are used in this manuscript:

AIS	Algorithmic Investment Strategy
B&H	Buy&Hold strategy
B&S	Buy&Sell strategy (Trading)
CNN	Convolutional Neural Network
CR	Calmer Ratio
DD	Draw Down
DMLP	Deep Multilayer Perceptron
DNN	Deep Neural Network
DWT	Discrete Wavelet Transform
EMD	Empirical Mode Decomposition
EMH	Efficient-Market Hypothesis
EqW	Equally-Weighted
GMR	Global Maximum Return
HHT	Hilbert Huang Transform
IFA	Improved Firefly Algorithm
IMF	Intrinsic Mode Function

Table 8
Portfolio optimization results. SR values are annualized.

HHT-XGB					XGB				
Stock	SLSQ		Monte Carlo		Stock	SLSQ		Monte Carlo	
	Weights	SR	Weights	SR		Weights	SR	Weights	SR
GT	0.01	4.6376	0.02	4.4086	JWN	0.09	2.3212	0.19	1.9874
VLO	0.15		0.28		WFC	0.54		0.48	
WFC	−0.07		−0.15		TSLA	0.16		0.32	
LYB	0.12		0.12		RSG	−0.5		−0.45	
MU	0.07		0.01		AJG	−0.5		−0.43	
DVN	0.09		0.2		SLB	−0.03		−0.18	
AMD	0.07		0.09		DVN	−0.13		−0.03	
WMB	0.17		−0.07		ACN	0.84		0.39	
TSLA	0.09		0.11		MU	0.44		0.48	
GPC	0.29		0.4		KSS	0.08		0.22	

LSTM	Long Short-Term Memory
McW	Market-cap Weighted
MDD	Maximum Draw Down
ML	Machine Learning
MPT	Modern Portfolio Theory
MR	Mean Return
MV	Mean–Variance
MVF	Mean–Variance with forecasting
OF	Omega with forecasting
OHLCV	Open High Low Close Volume
PSO	Particle Swarm Optimization
RNN	Recurrent Neural Network
RoR	Rate of Return
SI	Stability Index
SR	Sharpe Ratio
SVM	Support Vector Machines
TNX	10-year treasury yield
VIX	Volatility Index
XGB	Extreme Gradient Boosting

CRediT authorship contribution statement

Arsalan Dezhkam: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Mohammad Taghi Manzuri:** Conceptualization, Formal analysis, Methodology, Writing – review & editing.

Declaration of competing interest

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

Funding

This research received no external funding.

Data availability

The authors confirm that the data analyzed in this study are openly available on yahoo finance at finance.yahoo.com and are included in supplementary information files. The authors also confirm that the data generated during the study can be available upon reasonable request from the authors.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.engappai.2022.105626.

References

Arévalo, A., Nino, J., León, D., Hernandez, G., Sandoval, J., 2018. Deep learning and wavelets for high-frequency price forecasting. In: Shi, Y., Fu, H., Tian, Y., Krzhizhanovskaya, V.V., Lees, M.H., Dongarra, J., Sloot, P.M.A. (Eds.), Computational Science – ICCS 2018. In: Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 385–399. http://dx.doi.org/10.1007/978-3-319-93701-4_29.

Bogeh, C.K., 2021. Impact of oil price shocks on stock returns in Turkey: A sectoral analysis based on Hilbert-Huang transform and event study. Jafas 7, 138–154. http://dx.doi.org/10.32602/jafas.2021.007.

Burton, N., 2017. An Analysis of Burton Malkiel's a Random Walk Down Wall Street, the Macat Library. Routledge, London.

Carapuço, J., Neves, R., Horta, N., 2018. Reinforcement learning applied to Forex trading. Appl. Soft Comput. 73, 783–794. http://dx.doi.org/10.1016/j.asoc.2018.09.017.

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Presented at the KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. vol. 78, ACM, San Francisco California USA, pp. 5–794. http://dx.doi.org/10.1145/2939672.2939785.

Chen, Y., He, F., Li, H., Zhang, D., Wu, Y., 2020. A full migration BBO algorithm with enhanced population quality bounds for multimodal biomedical image registration. Appl. Soft Comput. 93, 106335. http://dx.doi.org/10.1016/j.asoc.2020.106335.

Chen, M.-Y., Liao, C.-H., Hsieh, R.-P., 2019. Modeling public mood and emotion: Stock market trend prediction with anticipatory computing approach. Comput. Hum. Behav. 101, 402–408. http://dx.doi.org/10.1016/j.chb.2019.03.021.

Chen, W., Zhang, H., Mehlawat, M.K., Jia, L., 2021. Mean–variance portfolio optimization using machine learning-based stock price prediction. Appl. Soft Comput. 100, 106943. http://dx.doi.org/10.1016/j.asoc.2020.106943.

Derakhshan, A., Beigy, H., 2019. Sentiment analysis on stock social media for stock price movement prediction. Eng. Appl. Artif. Intell. 85, 569–578. http://dx.doi.org/10.1016/j.engappai.2019.07.002.

Dezhkam, A., Manzuri, M.T., Aghapour, A., Karimi, A., Rabiee, A., Shalmani, S.M., 2022. A Bayesian-based classification framework for financial time series trend prediction. J. Supercomput. http://dx.doi.org/10.1007/s11227-022-04834-4.

Gomes, H.M., Bifet, A., Read, J., Barddal, J.P., Enembreck, F., Pfharinger, B., Holmes, G., Abdessalem, T., 2017. Adaptive random forests for evolving data stream classification. Mach. Learn. 106, 1469–1495. http://dx.doi.org/10.1007/s10994-017-5642-8.

Grudniewicz, J., Ślepaczuk, R., 2021. Application of machine learning in quantitative investment strategies on global stock markets. In: Working Papers of Faculty of Economic Sciences. University of Warsaw, WP 23 (371). https://www.wne.uw.edu.pl/files/6216/3603/4435/WNE_WP371.pdf.

Holder, L.B., Russell, I., Markov, Z., Pipe, A.G., Carse, B., 2005. CURRENT AND FUTURE trends IN FEATURE SELECTION AND extraction FOR classification PROBLEMS. Int. J. Pattern Recognit. Artif. Intell. 19, 133–142. http://dx.doi.org/10.1142/S0218001405004010.

Hoseinzade, E., Haratizadeh, S., 2019. CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Syst. Appl. 129, 273–285. http://dx.doi.org/10.1016/j.eswa.2019.03.029.

Huang, N.E., Shen, Z., Long, S.R., Qu, M.C., Shih, H.H., Zheng, Q., Yen, N.-C., Tung, C.C., Liu, H.H., 1998. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. In: Proceedings of the Royal Society of London, A. Presented at the Mathematical, Physical and Engineering Sciences. The Royal Society, pp. 903–995. http://dx.doi.org/10.1098/rspa.1998.0193.

Kao, L.-J., Chiu, C.-C., Lu, C.-J., Chang, C.-H., 2013. A hybrid approach by integrating wavelet-based feature extraction with MARS and SVR for stock index forecasting. Decis. Support Syst. 54, 1228–1244. http://dx.doi.org/10.1016/j.dss.2012.11.012.

Ke, H.-R., Wang, K.-C., Yang, C.-I., Chang, K.-F., 2014. Wavelet and Hilbert-Huang transform based on predicting stock forecasting in second-order autoregressive mode. IJAPM 4, 9–14. http://dx.doi.org/10.7763/IJAPM.2014.V4.246.

Kearns, M., Laboratories, T.B., Hill, M., Valiant, L., 1989. Cryptographic limitations on learning boolean formulae and finite automata. In: Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing. pp. 433–444. http://dx.doi.org/10.1145/73007.73049.

Keating, C., Shadwick, W.F., n.d. A Universal Performance Measure 42.

Kim, H.Y., Won, C.H., 2018. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. Expert Syst. Appl. 103, 25–37. http://dx.doi.org/10.1016/j.eswa.2018.03.002.

Kosc, K., Sakowski, P., Ślepaczuk, R., 2019. Momentum and contrarian effects on the cryptocurrency market. Physica A 523, 691–701. http://dx.doi.org/10.1016/j.physa.2019.02.057.

Lee, J., Koh, H., Choe, H.J., 2021. Learning to trade in financial time series using high-frequency through wavelet transformation and deep reinforcement learning. Appl. Intell. 51, 6202–6223. http://dx.doi.org/10.1007/s10489-021-02218-4.

Leung, T., Zhao, T., 2021. Financial time series analysis and forecasting with Hilbert–Huang transform feature generation and machine learning. Appl. Stoch. Models Bus. Ind. 37, 993–1016. http://dx.doi.org/10.1002/asmb.2625.

Li, H., He, F., Chen, Y., Pan, Y., 2021. MLFS-CCDE: multi-objective large-scale feature selection by cooperative coevolutionary differential evolution. Memetic Comput. 13, 1–18. http://dx.doi.org/10.1007/s12293-021-00328-7.

Li, M., Huang, Y., 2014. Hilbert–Huang transform based multifractal analysis of China stock market. Physica A 406, 222–229. http://dx.doi.org/10.1016/j.physa.2014.03.047.

Li, H., Ping, Y., 2015. Recent advances in support vector clustering: Theory and applications. Int. J. Pattern Recognit. Artif. Intell. 29, 1550002. http://dx.doi.org/10.1142/S0218001415500020.

Li, X., Wang, F., Chen, X., 2015. Support vector machine ensemble based on choquet integral for financial distress prediction. Int. J. Pattern Recognit. Artif. Intell. 29, 1550016. http://dx.doi.org/10.1142/S0218001415500160.

Liagkouras, K., Metaxiotis, K., 2018. Multi-period mean–variance fuzzy portfolio optimization model with transaction costs. Eng. Appl. Artif. Intell. 67, 260–269. http://dx.doi.org/10.1016/j.engappai.2017.10.010.

Liang, Y., He, F., Zeng, X., Luo, J., 2022. An improved loop subdivision to coordinate the smoothness and the number of faces via multi-objective optimization. Integr. Comput.-Aided Eng. 29, 23–41. http://dx.doi.org/10.3233/ICA-210661.

Liu, G., Wang, X., 2019. A new metric for individual stock trend prediction. Eng. Appl. Artif. Intell. 82, 1–12. http://dx.doi.org/10.1016/j.engappai.2019.03.019.

Long, J., Chen, Z., He, W., Wu, T., Ren, J., 2020. An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market. Appl. Soft Comput. 91, 106205. http://dx.doi.org/10.1016/j.asoc.2020.106205.

Long, W., Lu, Z., Cui, L., 2019. Deep learning-based feature engineering for stock price movement prediction. Knowl.-Based Syst. 164, 163–173. http://dx.doi.org/10.1016/j.knosys.2018.10.034.

Ma, Y., Han, R., Wang, W., 2021. Portfolio optimization with return prediction using deep learning and machine learning. Expert Syst. Appl. 165, 113973. http://dx.doi.org/10.1016/j.eswa.2020.113973.

Marcos López De Prado, 2018. Advances in Financial Machine Learning. Wiley.

Markowitz, H., 1952. Portfolio selection. J. Finance 7, 77–91.

Martínez-Nieto, L., Fernández-Navarro, F., Carbonero-Ruz, M., Montero-Romero, T., 2021. An experimental study on diversification in portfolio optimization. Expert Syst. Appl. 181, 115203. http://dx.doi.org/10.1016/j.eswa.2021.115203.

Mendonça, G.H.M., Ferreira, F.G.D.C., Cardoso, R.T.N., Martins, F.V.C., 2020. Multi-attribute decision making applied to financial portfolio optimization problem. Expert Syst. Appl. 158, 113527. http://dx.doi.org/10.1016/j.eswa.2020.113527.

Michańków, J., Sakowski, P., Ślepaczuk, R., 2022. LSTM in algorithmic investment strategies on BTC and S & P500 index. Sensors 22 (917), http://dx.doi.org/10.3390/s22030917.

Min, L., Dong, J., Liu, J., Gong, X., 2021. Robust mean-risk portfolio optimization using machine learning-based trade-off parameter. Appl. Soft Comput. 113, 107948. http://dx.doi.org/10.1016/j.asoc.2021.107948.

Nobre, J., Neves, R.F., 2019. Combining principal component analysis, discrete wavelet transform and XGBoost to trade in the financial markets. Expert Syst. Appl. 125, 181–194. http://dx.doi.org/10.1016/j.eswa.2019.01.083.

Ostad-Ali-Askari, K., Shayan, M., 2021. Subsurface drain spacing in the unsteady conditions by HYDRUS-3D and artificial neural networks. Arab J. Geosci. 14, 1936. http://dx.doi.org/10.1007/s12517-021-08336-0.

Ostad-Ali-Askari, K., Shayannejad, M., Eslamian, S., Zamani, F., Shojaei, N., Navabpour, B., Majidifar, Z., Sadri, A., Ghasemi-Siani, Z., Nourozi, H., Vafaei, O., Homayouni, S.-M.-A., 2017a. Deficit irrigation: Optimization models. In: Handbook of Drought and Water Scarcity. Taylor & Francis, pp. 373–389.

Ostad-Ali-Askari, K., Shayannejad, M., Ghorbanizadeh-Kharazi, H., 2017b. Artificial neural network for modeling nitrate pollution of groundwater in marginal area of Zayandeh-rood River, Isfahan, Iran. KSCE J. Civ. Eng. 21, 134–140. http://dx.doi.org/10.1007/s12205-016-0572-8.

Pang, X., Zhou, Y., Wang, P., Lin, W., Chang, V., 2020. An innovative neural network approach for stock market prediction. J. Supercomput. 76, 2098–2118. http://dx.doi.org/10.1007/s11227-017-2228-y.

Pirnazar, M., Hasheminasab, H., Karimi, A.Z., Ostad-Ali-Askari, K., Ghasemi, Z., Haeri-Hamedani, M., Mohri-Esfahani, E., Eslamian, S., 2018. The evaluation of the usage of the fuzzy algorithms in increasing the accuracy of the extracted land use maps. Int. J. Global Environ. Issues 17, 307–321. http://dx.doi.org/10.1504/IJGENVI.2018.095063.

Pitkäjärvi, A., Suominen, M., Vaittinen, L., 2020. Cross-asset signals and time series momentum. J. Financ. Econ. 136, 63–85. http://dx.doi.org/10.1016/j.jfineco.2019.02.011.

Schapire, R.E., 1990. The strength of weak learnability. Mach. Learn. 5, 197–227. http://dx.doi.org/10.1007/BF00116037.

Singh, R., Srivastava, S., 2017. Stock prediction using deep learning. Multimed. Tools Appl. 76, 18569–18584. http://dx.doi.org/10.1007/s11042-016-4159-7.

Ślepaczuk, R., Zenkova, M., 2019. Robustness of support vector machines in algorithmic trading on cryptocurrency market. Cent. Eur. Econ. J. 5, 186–205. http://dx.doi.org/10.1515/ceej-2018-0022.

Song, Y., Lee, J.W., Lee, J., 2019. A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction. Appl. Intell. 49, 897–911. http://dx.doi.org/10.1007/s10489-018-1308-x.

Suárez-Cetrulo, A., Cervantes, A., Quintana, D., 2019. Incremental market behavior classification in presence of recurring concepts. Entropy 21 (25), http://dx.doi.org/10.3390/e21010025.

Thakkar, A., Chaudhari, K., 2021. A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization. Arch. Comput. Methods Eng. 28, 2133–2164. http://dx.doi.org/10.1007/s11831-020-09448-8.

Valderrama, C.E., 2021. A comparison between the Hilbert-Huang and Discrete Wavelet Transforms to recognize emotions from electroencephalographic signals. In: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). Presented at the 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society. EMBC, IEEE, Mexico, pp. 496–499. http://dx.doi.org/10.1109/EMBC46164.2021.9630188.

Valencia, F., Gómez-Espinosa, A., Valdés-Aguirre, B., 2019. Price movement prediction of cryptocurrencies using sentiment analysis and machine learning. Entropy 21 (589), http://dx.doi.org/10.3390/e21060589.

Winkler, F., 2020. The role of learning for asset prices and business cycles. J. Monetary Econ. 114, 42–58. http://dx.doi.org/10.1016/j.jmoneco.2019.03.002.

Wu, D., Wang, X., Wu, S., 2021. A hybrid method based on extreme learning machine and wavelet transform denoising for stock prediction. Entropy 23 (440), http://dx.doi.org/10.3390/e23040440.

Xuan, L., Guozhong, C., Fulong, L., 2015. Stock data analysis based on Hilbert-Huang transform. In: 2015 IEEE International Conference on Grey Systems and Intelligent Services (GSIS). Presented at the 2015 IEEE International Conference on Grey Systems and Intelligent Services. GSIS, IEEE, Leicester, United Kingdom, pp. 618–621. http://dx.doi.org/10.1109/GSIS.2015.7301816.

Yang, L., Tian, S., Yang, W., Xu, M., Hamori, S., 2018. Dependence structures between Chinese stock markets and the international financial market: Evidence from a wavelet-based quantile regression approach. North Am. J. Econ. Finance 45, 116–137. http://dx.doi.org/10.1016/j.najef.2018.02.005.

Yazdani, S.F., Murad, M.A.A., Sharef, N.M., Singh, Y.P., Latiff, A.R.A., 2017. Sentiment classification of financial news using statistical features. Int. J. Pattern Recognit. Artif. Intell. 31, 1750006. http://dx.doi.org/10.1142/S0218001417500069.

Yu, J.-R., Paul Chiou, W.-J., Lee, W.-Y., Lin, S.-J., 2020. Portfolio models with return forecasting and transaction costs. Int. Rev. Econ. Finance 66, 118–130. http://dx.doi.org/10.1016/j.iref.2019.11.002.

Zhou, Z., Lin, L., Li, S., 2018. International stock market contagion: A CEEMDAN wavelet analysis. Econ. Model. 72, 333–352. http://dx.doi.org/10.1016/j.econmod.2018.02.010.