International Journal on Data Science and Technology
2022; 8(4): 72-86
http://www.sciencepublishinggroup.com/j/ijdst
doi: 10.11648/j.ijdst.20220804.13
ISSN: 2472-2200 (Print); ISSN: 2472-2235 (Online)

Forecasting of Global Stock Market by Two Stage Optimization Model

Junsuke Senoguchi

School of Computer Science, Tokyo University of Technology, Tokyo, Japan

Email address:
senoguchijs@stf.teu.ac.jp

To cite this article:
Junsuke Senoguchi. Forecasting of Global Stock Market by Two Stage Optimization Model. *International Journal on Data Science and Technology*. Vol. 8, No. 4, 2022, pp. 72-86. doi: 10.11648/j.ijdst.20220804.13

Received: November 15, 2022; **Accepted**: December 6, 2022; **Published**: December 15, 2022

Abstract: Currently, about half of the transactions in the US stock market are based on high-frequency algorithmic trading, making it difficult for the investors with the long-term investment horizon, such as pension funds, to obtain stable returns. The development of a market forecast model that could achieve stable returns over the long term is an important issue in supporting not only pensions but also the central bank policy makers or new private businesses. To obtain stable investment performance by a forecast model over the long-term, it is necessary to remove noise from sample data in advance and extract a universal pattern. However, it is difficult to preliminarily distinguish between noise and true patterns and remove noise in advance. In this study, the sample space was divided into 8 sub-spaces using a Two Stage Optimization decision tree, and the versatility of each sub-space was evaluated by a pattern recognition model. Then, the sub-space with a low versatility was defined as the space with relatively large noise, and a forecast model was created by excluding the sub-spaces with large noise. It was found that the forecast model constructed in this way could obtain the prediction accuracy higher than that of the conventional method. Also, when the prediction accuracy of the model was evaluated by the walk-forward method using financial time-series data, investment performance that stably exceeded the return of benchmark assets was obtained over the past 15 years.

Keywords: Evolutionary Computing, Noisy Function Optimization, Financial Market, Forecasting, Regression

1. Introduction

Currently, about half of the transactions in the US stock market are by high-frequency algorithmic trading. A small number of institutional investors with algorithmic trading equipment and know-how make big profits, and the opportunities for other investors to make profits are diminishing. As a result, it is difficult for pension funds to operate stably or for new private businesses to finance from the stock market as in the past. Especially during the corona shock in early 2020, algorithmic trading disrupted stock and oil prices, and even central bank monetary policy and government economic policy could not stabilize the market. Frequent market turmoil would further widen the gap between rich and poor and destabilize society. In order to address these issues, it is now necessary to develop a stock price forecasting model that can obtain stable investment performance over the long term despite frequent short-term market turmoil or regime changes.

So far, various stock price forecasting models have been developed in academia and businesses. From the viewpoint of stock price prediction models using machine learning techniques, Bruno et al. [1] reviewed and classified 101 related works covering specialized literature from 1991 to 2017, and Obthong et al. [2] examined 52 previous research and explored the application of machine learning exclusively focusing on stock price prediction. Among machine learning techniques, Hu et al. [3] and Kumar et al. [4] focused on the neural network and have surveyed 93 and 20 forecasting models respectively specializing and compared them in characteristics and forecasting accuracy. In addition, research on stock market prediction based on sentiment using text mining techniques has been active in recent years [5, 6].

While many stock price forecasting models have been developed so far, few studies have verified the forecasting accuracy of the models over a long period of time (i.e. over 15 years) including different financial regime and successfully achieved the stable forecasting accuracy. One of the reasons is that the model fits into a few stock price drivers for a certain period of time, and the forecast accuracy is high during the period when the drivers are effective, but the forecast accuracy is significantly reduced when they waned in influence. In particular, if a model is created by a method with many parameters such as a deep neural network, it tends to overfit to a few stock price fluctuation factors for a specific time. In such a case, the prediction accuracy of the model is high at a specific time, but the prediction accuracy is low at other times. Also, sentiment analysis generally uses a relatively short period of data set; therefore, it is difficult to verify long-term prediction accuracy.

It has also been reported that when real money is invested using a stock price forecasting model that shows high forecasting accuracy in the paper, the excellent results shown in the paper are often not obtained. [7]. Since the factors that cause stock price fluctuations change as the financial market regime changes, it has been shown that long-term forecasting is difficult even with a stock price forecasting model that shows high predictability over a certain period of time.

As countermeasures against such issues, it is important (1) to ignore stock price drivers peculiar to a certain short period of time (hereinafter referred to as noise) and extract universal stock price fluctuation factors over a long period of time, and (2) to make forecasts using different models in response to different financial market regime.

However, with regard to (1), whether the stock price fluctuation factor recognized by the model is noise or universal over a long period of time will be known later, and it is not known at the time of forecasting. Therefore, it is not possible to create a model by removing noise from the data.

On the other hand, it is possible to create an algorithm that extracts a space containing noise by induction. In such a process, a subsample is extracted from the entire sample space according to a set of several conditions. If a highly versatile pattern is recognized in a subsample, it can be regarded as a space with relatively little noise; however, if pattern recognition is not versatile in another subsample, it can be regarded as a space containing many noise and extraordinary values.

Also, it is possible to create an algorithm that searches for the optimum combination of conditions using the versatility of the prediction model in a subsample as the evaluation value. If the entire sample space can be divided into a space containing a lot of noise and a apace containing little noise, a universal pattern can be recognized over a long period of time by creating a model in a space containing little noise. This can solve (1).

In this research, the above process is implemented using a decision tree. The decision tree divides the space by a set of conditions, and this process is like the financial market regime shift. In addition, since the space that does not contain noise is further subdivided in the decision tree, a unique model can be created in each subspace. This indicates that (2) can be solved as different forecast models can be created for each difference market regime.

On the other hand, constructing a decision tree in a recursive and heuristic way will eventually make a prediction model to overfit to noise. Therefore, when searching for a split of a decision tree, it is essential to find a global solution from all combinations of split criteria.

There are studies that use evolutionary computation to optimize the split of a decision tree [8, 9]. In addition, there are many papers that use evolutionary computation for optimizing other machine learning parameters [10, 11]. However, since the number of combinations of split in a decision tree generally becomes excessively large, its optimization is considered to be NP-hard.

Therefore, in this research, it is proposed that propose Two Stage Optimization in order to optimize a decision tree that divides the space containing noise and the space other than that. Two Stage Optimization performs a search for the optimum threshold value under a specific set of attributes by an evolutionary computation and a search for a combination of the optimum attributes by another metaheuristic computation.

In this research, a model tree that divides the space containing noise and the space other than that is constructed by Two Stage Optimization, and a stock price prediction model is created for each final node of a tree. By doing so, this study is aimed to develop a forecasting model that can stably maintain high forecasting accuracy even in different financial regime and stable investment performance for a long period.

2. Related Research

Research on stock price forecasting has been actively conducted in academia and business. In the olden days, many of them used traditional financial theory, but most of the studies published in recent years used machine learning instead of traditional financial theory. As this research aims to create a stock price forecast model using machine learning methods, the section presents related studies that predict financial instrument prices by machine learning.

2.1. Financial Price Prediction Using Machine Learning

There have been many studies that have created stock price forecasting models using relatively simple neural networks or deep learning [12-14]. Neural networks generally optimize parameters by the back propagation method, but there are also studies in which neural networks of stock price prediction models are optimized by evolutionary computation [11].

There are also many studies that build stock price forecast models using RNNs or LSTMs that add the concept of time series analysis to neural networks [15-18]. Several studies have shown that LSTMs provide high accuracy in forecasting crude oil and gold prices as well as stock prices, demonstrating a wide range of applications of LSTM to financial forecasting models [19, 20].

For machine learning other than neural networks, Basak et al. [21] developed a stock price prediction model using a decision tree, and Xiao [22] and Yang [23] developed a stock price prediction model using SVM, both of which have been recognized the improvement in prediction accuracy.

A model for predicting the financial crisis, which is one of the biggest fluctuation factors of stock market has also been developed using machine learning and is proven useful for predicting stock prices [24, 25]. But the data used are imbalanced and the application to the stock price prediction needs some adjustment.

2.2. Financial Price Prediction Using Text and Sentiment

In addition to a stock price prediction model using numerical data, many related works where a model that recognizes financial market sentiment from text data and predicts financial product prices are available [26-28].

Some studies have shown high predictability not only for stock prices but also for cryptocurrencies whose prices fluctuate significantly in a short period of time. However, as the availability of text data is relatively short, only short-term forecasts are made so far [29].

2.3. Building a Tree While Avoiding Overfitting

This subsection introduces a pioneering research on the search for global solutions of decision trees used in this study.

A decision tree constructed by the greedy method usually has a problem of falling into a locally optimal solution. To avoid this overfitting problem, methods have been developed for determining a solution with multiple decision trees by ensemble learning [30, 31] and for optimizing splits using evolutionary computation [32].

Although both methods have shown higher accuracy than that using the greedy method, interpretability is lost in ensemble learning. Therefore, the research which searches the most suitable decision tree using evolution computation has been often performed [33-38].

The method of constructing a decision tree using evolutionary computation can be divided into two main threads: the evolutionary induction of decision trees and the evolutionary design of decision-tree components. The former method is an approach that optimizes the overall structure of the tree as a whole whereby each individual in evolutionary computation is the decision tree itself. Similar approaches have been used in numerous previous studies [9, 39-41]. In the latter method, however, each individual is a component optimized and combined to search the optimal tree structure. In the case of complex data, the evaluation value of the tree often does not improve when a split that is effective for the data sample created from splits by specific upper nodes is applied to that created from splits by other upper nodes [42, 43].

The attempt which optimizes branch of a decision tree all at once is also performed by a genetic programming extensively from the old days [44-47]. However, when the room where optimized calculation is made efficient is limited. Also, a genetic programming tries to search a numerous decision tree of branch using complicated data; therefore, a convergence to global optimum solution becomes difficult in terms of computation time.

2.4. Evolutionary Computation as a Method for Building Tree

When optimizing the branching criterion of the whole tree collectively by evolutionary calculation, the evaluation value of the tree changes nonlinearly when the attribute of the branching criterion of the upper node is changed, so optimization methods that assume a continuous evaluation function or evolutionary calculations that generate individuals based on distributions cannot be used.

In addition to the steepest gradient, Adam, and Newton methods, optimization methods that assume a continuous evaluation function include Bayesian optimization, which has been applied to optimize hyperparameters in machine learning [48, 49]. However, when dealing with complex data with a large amount of noise, a single point search like these methods may led to a local optimum solution. Therefore, the stochastic search method, which is a black-box optimization method using multipoint search [50] and real-valued evolutionary computation are considered to be suitable for the search of the threshold.

The latter method includes real-valued GA [51], evolution strategy [52], differential evolution [53], and particle swarm optimization [54].

Real-valued GA, through minimal generation gap [55], unimodal distribution crossover (UNDX) [56], and real-coded ensemble crossover star (REXstar) [57], has made it possible to show high performance in evaluation functions with problems such as bad scalability, inter-variable dependence, and global multimodality.

In addition, there are some papers on evolutionary computation as a means of optimizing other machine learning. Chakraborty & Kar [58] used the swarm intelligence for the optimization. However, as with multi-agent simulation, in the swarm intelligence, researchers must make a relatively large number of assumptions, where arbitrariness arises. Slowik & Kwasnicka [59] used the evolutionary algorithms for the optimization of a tree. However, it has not solved the problem that the evaluation function becomes intermittent when the attribute of the branch of the decision tree is changed.

2.5. Human Activities Prediction by Bio-oriented Methods

Although different from the prediction of financial data, there are many studies using text to predict human behavior from sentiment, and many studies show relatively high prediction accuracy. It is difficult to predict the financial market using social media because information is mostly already factored into the stock price when it is available on the social media. However, analysis techniques used in those studies can give important suggestions for stock price prediction of this study in a hybrid bio-inspired computing approach for buzz detection [60].

Anupam & Kar [61] and Batra et al [62] target Phishing websites and spam email respectively, and they are different from the purpose of excluding short-term noise in financial markets. However, it is also a method that gives important suggestions for achieving the purpose of this study.

Kar & Aswani [63] proposed the scheme to differentiate information and misinformation using social media based on bio-inspired computing, where there are some similarities to the purpose of this study in terms of the classification of data into two spaces by bio-inspired computing. In this study, the main purpose is to classify the unsupervised explanatory variables into noise or not; therefore, the classification method is not the similarity or distance of the data, but the versatility of the model. However, it is also a method that gives important suggestions for achieving the purpose of this study.

2.6. Issues of Previous Research and Purpose of This Research

As described above, there are many related works in terms of developing a prediction model using evolutionary computing; however, few methods can handle multidimensional complex data with large noise owing to their difficulty in adjusting the step size, population size, number of offspring, and other factors. Among evolutionary strategies, with which the need to adjust the step size, population size, and number of generator individuals is relatively small, it has been pointed out that the covariance matrix adaptation evolution strategy (CMA-ES) [64] and distance-weighted exponential natural evolution strategies (DX-NES) [65] significantly degrades the search performance when applied to noisy and complex data.

However, it has been reported that when applied to the numerical data including a large number of one-off factors, the DX-NES lost its performance significantly [66]. On the other hand, CMA-ES shows relatively higher resistant to noise and can be considered desirable for the evaluation function that changes significantly by changing the parameter threshold of the split in a decision tree. Among the many variations of CMA-ES, Hansen [67] reported the best performance for data in noisy and uncertain environments, and the Richter [68] model is suitable for searching decision trees.

From these previous studies, it is said that it is important to use a search method that presupposes continuous distribution when searching for tree split criteria collectively. This makes it possible to efficiently search for the global optimum solution even with an evaluation function having a complicated shape. Creating a highly versatile model tree using such a method is a new attempt.

3. Data and Methods

3.1. Data Collection

Time series data of financial markets are used as actual data of complex systems. Financial data is obtained from the Nikkei QUICK news database and Yahoo! Finance.

Financial market data is a typical example with which high prediction accuracy cannot be obtained even by machine learning because it contains many extraordinary factors and noise. This research used as an objective variable the intraday return for the iShares Core S&P 500 exchange traded fund (ETF) from the opening price at 09:30 to the closing price at 16:00. Explanatory variables include financial indexes that represent the economic fundamentals of the United States, such as the change in the closing price of the stock index, exchange rate, and interest rate (Table 1). All financial data is obtained from Nikkei QUICK news database (https://corporate.quick.co.jp/en/).

The financial data used to predict the S&P 500 ETF in next business day (BD) include 3,759 business day (BD) intraday returns between January 03, 2007, and December 30, 2021 as an objective variable.

Table 1. *Financial data used to predict S&P 500 ETF.*

Variables	Name	Term
Objective	iShares Core S&P 500 ETF	NextBD Open~Close
Explanatory1	S&P500	Previous BD ~ Current BD
Explanatory2	Dow Jones/S&P500	Previous BD ~ Current BD
Explanatory3	USD Index	Previous BD ~ Current BD
Explanatory4	US Treasury 10Y Yield	Previous BD ~ Current BD
Explanatory5	US Treasury 5-30Y Yield Gap	Previous BD ~ Current BD
Explanatory6	S&P500	3BDs Ago ~ Currend BD
Explanatory7	Dow Jones/S&P500	3BDs Ago ~ Currend BD
Explanatory8	USD Index	3BDs Ago ~ Currend BD
Explanatory9	US Treasury 10Y Yield	3BDs Ago ~ Currend BD
Explanatory10	US Treasury 5-30Y Yield Gap	3BDs Ago ~ Currend BD
Explanatory11	S&P500	7BDs Ago ~ Currend BD
Explanatory12	Dow Jones/S&P500	7BDs Ago ~ Currend BD
Explanatory13	USD Index	7BDs Ago ~ Currend BD
Explanatory14	US Treasury 10Y Yield	7BDs Ago ~ Currend BD
Explanatory15	US Treasury 5-30Y Yield Gap	7BDs Ago ~ Currend BD

3.2. Data Analysis

Tables 3 and 4 show the results of analyzing the data. Table 2 shows the correlation matrix for the data whose objective variable has a plus sign (when stock price rises), and Table 3 shows the correlation matrix for the data whose objective variable has a minus sign (when stock price falls). The diagonal values are the average in the upper row and the standard deviation in the lower row. There are some explanatory variables with a correlation coefficient relatively higher than others, such as the correlation between the change in one business day and the change in 3 business days, but no correlation coefficient exceeds 0.5; therefore, the explanatory variables can be considered close to independent of each other.

Also, regardless of whether the sign of the objective variable is positive or negative, there is no explanatory variable that shows a high simple correlation with the objective variable. In addition, there are no explanatory variables whose distribution shapes differ greatly depending on whether the sign of the objective variable is positive or negative.

Table 2. *Data Analysis (when stock price rises).*

	Obj.	**Exp. 1**	**Exp. 2**	**Exp. 3**	**Exp. 4**	**Exp. 5**	**Exp. 6**	**Exp. 7**	**Exp. 8**	**Exp. 9**	**Exp. 10**	**Exp. 11**	**Exp. 12**	**Exp. 13**	**Exp. 14**	**Exp. 15**
Obj.	0.01 0.01	-0.19	0.07	-0.05	-0.08	0.07	-0.23	0.08	-0.09	-0.11	0.05	-0.28	0.10	-0.12	-0.12	0.05
Exp. 1		0.05 1.02	-0.17	0.06	0.30	-0.08	0.41	-0.06	0.02	0.08	-0.04	0.21	0.01	0.01	0.04	-0.03
Exp. 2			-0.00 1.00	0.01	-0.06	0.01	-0.08	0.45	-0.00	0.01	-0.02	-0.00	0.14	0.01	0.02	0.02
Exp. 3				0.00 0.99	0.02	-0.00	0.15	-0.02	0.40	0.18	-0.10	0.03	0.02	0.15	0.10	-0.04
Exp. 4					-0.02 1.01	-0.21	0.10	-0.02	-0.02	0.42	-0.11	0.06	-0.00	0.01	0.14	-0.05
Exp. 5						-0.00 0.98	-0.04	0.01	-0.01	-0.07	0.42	-0.06	-0.00	-0.03	0.01	0.11
Exp. 6							-0.02 0.99	-0.17	0.12	0.24	-0.04	0.47	-0.05	0.03	0.09	-0.03
Exp. 7								-0.00 0.99	-0.03	-0.01	-0.05	-0.04	0.46	-0.00	0.04	-0.03
Exp. 8									0.01 1.00	0.15	-0.10	0.08	0.01	0.46	0.19	-0.12
Exp. 9										-0.01 1.00	-0.20	0.10	0.00	0.05	0.45	-0.14
Exp. 10											0.00 1.00	-0.06	-0.04	-0.05	-0.04	0.45
Exp. 11												-0.02 0.99	-0.15	0.13	0.19	-0.08
Exp. 12													0.01 1.00	-0.03	0.03	-0.07
Exp. 13														-0.00 1.01	0.20	-0.15
Exp. 14															0.01 1.03	-0.21
Exp. 15																-0.00 1.01

Table 3. *Data Analysis (when stock price falls).*

	Obj.	**Exp. 1**	**Exp. 2**	**Exp. 3**	**Exp. 4**	**Exp. 5**	**Exp. 6**	**Exp. 7**	**Exp. 8**	**Exp. 9**	**Exp. 10**	**Exp. 11**	**Exp. 12**	**Exp. 13**	**Exp. 14**	**Exp. 15**
Obj.	-0.01 0.01	0.06	-0.03	0.01	0.00	-0.01	0.16	-0.03	0.03	0.07	-0.05	0.21	-0.03	0.04	0.08	-0.09
Exp. 1		0.05 0.97	-0.19	0.03	0.25	-0.04	0.26	-0.06	-0.01	0.06	-0.00	0.02	-0.02	-0.03	0.01	-0.01
Exp. 2			0.01 1.00	-0.00	-0.02	-0.02	-0.03	0.40	-0.02	0.04	-0.03	-0.01	0.13	-0.02	0.01	-0.02
Exp. 3				-0.00 1.01	-0.01	-0.03	0.11	-0.04	0.41	0.19	-0.13	0.02	-0.01	0.15	0.09	-0.07
Exp. 4					0.02 0.98	-0.20	0.02	0.03	-0.02	0.36	-0.10	-0.03	0.00	-0.03	0.09	-0.06
Exp. 5						0.00 1.03	-0.01	-0.08	-0.01	-0.07	0.41	-0.03	-0.03	0.01	-0.02	0.11
Exp. 6							0.02 1.01	-0.17	0.15	0.20	-0.08	0.48	-0.05	0.06	0.11	-0.07
Exp. 7								0.00 1.02	-0.03	0.03	-0.04	-0.05	0.51	-0.04	0.03	-0.01
Exp. 8									-0.02 1.00	0.13	-0.11	0.11	-0.04	0.52	0.21	-0.13
Exp. 9										0.01 0.99	-0.24	0.05	0.01	0.02	0.46	-0.14
Exp. 10											-0.00 1.00	-0.08	-0.04	-0.04	-0.09	0.46
Exp. 11												0.02 1.01	-0.10	0.15	0.20	-0.12
Exp. 12													-0.01 1.00	-0.04	0.04	-0.05

	Obj.	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8	Exp. 9	Exp. 10	Exp. 11	Exp. 12	Exp. 13	Exp. 14	Exp. 15
Exp. 13														0.00 0.99	0.21	-0.15
Exp. 14															-0.01 0.97	-0.23
Exp. 15																0.01 0.99

3.3. Method

3.3.1. Two Stage Optimization

In this paper, an improved version of the method used in [69, 70] is used.

Since the attributes used for the splits of a certain node change, the evaluation value of a decision tree usually changes intermittently, when its overall structure is optimized all at once. With this reason, the continuous functions cannot be used as a search method. Therefore, in this research, Two Stage Optimization that consist of the inner-level search that optimizes the threshold for each feature and the outer-level search that searches the best features and their positions in a tree is adopted. Figure 1 outlines the mechanism of a Two Stage Optimization.

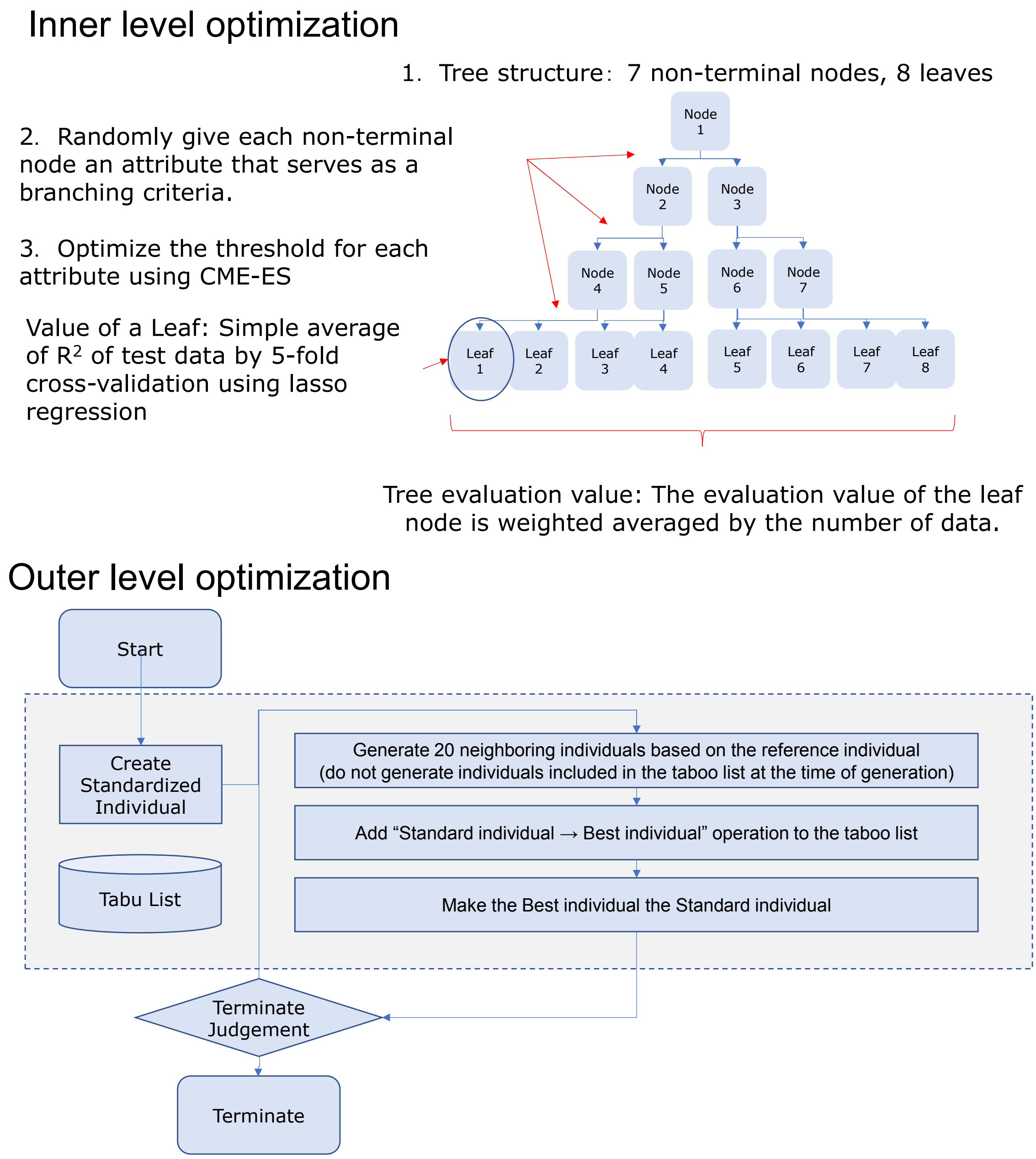

Figure 1. *Outline of Two Stage Optimization used in this research.*

3.3.2. Inner-Level Optimization

Inner level optimization is performed by the following mechanism, and the outline is shown in Figure 1 (upper figure).

1) The structure of the tree is 4 depths, 7 non-terminal nodes, and 8 leaf nodes.
2) Randomly give each non-terminal node an attribute that serves as a branching reference and fix it.
3) Optimize the threshold of each attribute by CME-ES.
   (a) Set the number of generations of CME-ES to 500 times.
   (b) The evaluation value is the weighted average of the predicted versatility of each leaf node by the number of elements of each leaf node.
   (c) The predictive versatility of each leaf node is the simple average of the coefficients of determination of the validation data by the 5-fold cross-validation method using lasso regression (the average of the coefficients of determination obtained from the five test data).
   (d) The regularization parameter of the lasso regression is 0.1.
   (e) If the number of data classified as a non-terminal node is less than 15 times the number of attributes of training data, it becomes a leaf node without further branching.

In the inner level optimization, the attributes that serve as branching criteria for the seven branches are randomly extracted from the attribute set prepared in advance, and then the threshold value of each attribute is searched. The depth is set to 4 in order to suppress the solution space of CMA-ES and the calculation time. The search range of the threshold value for each attribute is from 25% tile point to 75% tile point.

In general CMA-ES applications, the degree of freedom is $n+\frac{n^2-n}{2}$; the time complexity is $O(n^2)$; and the spatial complexity is $O(n^3)$ for the number of the dimension n of the evaluation function. By limiting the variance–covariance matrix $C^{(t+1)}$ used for individual generation to diagonal components, the degree of freedom becomes n, and the amount of time and spatial complexity is reduced to $O(n)$:

$$c_{jj}^{(t+1)} = (1-c_{cov})c_{jj}^{(t)} + \frac{1}{\mu_{cov}} c_{cov}\left(p_c^{(t+1)}\right)_j^2 + c_{cov}\left(1-\frac{1}{\mu_{cov}}\right)\sum_{i=1}^{\mu} w_i c_{jj}^{(t)}\left(z_{i:\lambda}^{(t+1)}\right)_j^2, j=1,..., \tag{1}$$

where $c_{cov}\in[0,1]$ is the learning rate of diagonal element updates; $\frac{1}{\mu_{cov}}\in[0,1]$ is the weighting coefficient of the evolution path $p_c^{(t+1)}$; $z_{i:\lambda}^{(t+1)}$ is the i-th most rated of the $z^{(t+1)}$; and $\left(z_{i:\lambda}^{(t+1)}\right)_j$ is the i-th component of $\left(z_{i:\lambda}^{(t+1)}\right)_j$.

3.3.3. Outer-Level Optimization

In the outer level search, the attributes used for each branch and their positions are optimized. The structure of a decision tree changes significantly if the attributes of one of the splits in a tree used is changed. With this reason, the outer-level search needs to use the search method assuming non-continuous distribution. Therefore, Tabu Search, which is one of the metaheuristic methods that does not consider the shape of the evaluation function is used. In [69, 70], Naïve GA is used as a method in the outer level search, however, since Tabu Search uses the previous search results for an efficient optimization, the time to converge is considered to be shorter than that of Naïve GA, and the solution is close to the optimum solution with a small number of generations.

Outer level optimization is performed by the following algorithm, and the outline is shown in Figure 1 (lower figure).

1. Give each non-terminal node of an initial tree an attribute that serves as a branching reference and optimize the threshold of each attribute by CME-ES (by Inter level optimization). This initial tree is referred to as S_0.
2. Create the best state S_b and the current state S, and record S_0 in both for the time being.
3. Select multiple (M) neighborhoods of S and set the neighborhood with the best grade as S'.
4. Judge the state transition (either of the following).
   (a) If S' is better than S_b, then S_b = S = S'. At this time, if the taboo list contains an operation that changes from S to S', that part is moved to the newest description in the taboo list.
   (b) If S' is worse than Sb, check if the S → S' operation is listed in the taboo list. If it is not described, enter the operation that makes S → S' in the taboo list and set S = S'. At this time, if the size of the taboo list exceeds the upper limit, the oldest description is deleted.

3.3.4. Model Evaluation Method Without Removing the Space Containing Noise

The weighted prediction accuracy by linear regression analysis at the final node is used as an evaluation value of a tree. As the prediction accuracy, the average R^2 obtained by the five-fold cross-validation method is used. Also, the Lasso regression model is used as a linear regression model in this study with a regularization parameter of 0.1.

The Oakbridge-CX supercomputer system at the Information Technology Center, University of Tokyo, is used to calculate the largescale parallelization calculation.

3.3.5. Model Evaluation Method Including Removal of the Space Containing Noise

The model evaluation method discussed in Section 3.3.4 targets all final nodes for evaluation. Even if many final nodes show high evaluation values, the evaluation as a tree is not high when one final node shows extremely low evaluation values. However, because the evaluation value is low for only some of the final nodes, the subsets classified in that location could have relatively large numbers of one-off factors and extraordinary values. If the evaluation values of the other final nodes are high except for those of the subsets, this tree extracts the space including noise, and a true model reflecting the core structure of the population can be built in the area excluding the space containing noise. Therefore, in this research, the model for only the final nodes of the tree having high evaluation values is evaluated. The evaluation value of the decision tree is the weighted average of the R^2 of the final node that was weighted by the number of elements classified into the final node.

3.3.6. Accuracy Comparison and Robustness Check

To evaluate the prediction accuracy of each method using financial time series data, the following methods were used: multiple regression analysis, lasso regression analysis, partial minimum error, neural network, XGBoost, Two Stage Optimization including noise, and Two Stage Optimization excluding noise. For the prediction accuracy used for comparison, the average R^2 according to the five-fold cross-validation method was used.

The results may show excellent accuracy only when a particular random number is generated. In order to avoid drawing conclusions from such accidental results, the seeds of random numbers is changed to see if there is any change in the results and verify the robustness of the model.

4. Results

4.1. Two Stage Optimization Prediction Accuracy Excluding Noise

Two Stage Optimization (referred to as TSO in Table 4) excluding noise showed the better results in the prediction of financial data (Table 1) than that of general machine learning methods.

Table 4. *Accuracy comparison using financial time series data (Average R^2 for Training and Test Data).*

	Train	Test
Linear Regression	0.43	0.01
Lasso Regression	0.44	0.02
Partial Minimum Error	0.43	0.02
Neural Network	0.64	0.03
XGBoost	0.62	0.05
TSO Including Noise	0.55	0.19
TSO Excluding Noise	0.56	0.36

In the Two Stage Optimization, many hyperparameters are generated randomly such as the solution of the initial group of CMA-ES at the inner level, including the features and thresholds; individuals for initial generation at the outer level, and cross-validation of data samples in the performance evaluation. When the calculation was repeated 10 times with different random number seeds, strongly similar results were obtained (Table 5).

Table 5. *Prediction accuracy using different random number seeds.*

	Train	Test
Seed1	0.56	0.36
Seed2	0.59	0.36
Seed3	0.56	0.33
Seed4	0.60	0.34
Seed5	0.55	0.34
Seed6	0.58	0.35
Seed7	0.58	0.35
Seed8	0.60	0.35
Seed9	0.58	0.35
Seed10	0.55	0.36

4.2. Inner-Level Evolutionary Computation

Inner-level searches require numerous iterative calculations and long processing time for converging to a global optimal solution when using complex information such as financial time series data. To confirm that the CMA-ES used in this research is superior to general CMA-ES and natural evolutionary computation in terms of processing time, the processing time needed for evaluation value convergence using the same features and their positions as the best tree is calculated, as shown in the first row in Table 5. The average convergence time T_{conv} following Zhang [71] was used as the processing time.

The processing time of CMA-ES used in this research was shorter than that of NES, which is a similar evolution strategy (Table 6).

Table 6. *Time in seconds needed for convergence in each evolutionary computation.*

NES	13.587
CMA-ES	11.948
CMA-ES (Hyperparameters optimized)	10.037

4.3. Evaluation of the Model Assuming Actual Operation (S&P 500, Binary Tree)

Few previous studies on financial market forecasts obtained the same results when the proposed methods were faithfully reproduced and put into actual operation. Therefore, to confirm the prediction accuracy of Two Stage Optimization excluding noise under the situation as if it actually operates, the operational performance using the walk-forward method is calculated as follows.

By using daily time series data for 1,500 business days prior to December 29, 2006, the best tree was created by applying Two Stage Optimization excluding noise. By using the data for December 29, 2006, the best tree created above was able to predict the intraday return from the opening price at 09:30 to the closing price at 16:00 on January 03, 2007 (next business day). If the prediction by the tree has a positive value, the investment return of the tree equals the intraday return of the iShares Core S&P 500 ETF; a negative value represents the intraday return with the opposite sign. If the data of December 29, 2006 is classified as the final node, which was excluded from the tree owing to the low evaluation value, it is likely to contain noise; thus, no investment is made (the investment return is zero).

This process was repeated from January 03, 2007, to December 30, 2021, from which 3,760 business day investment returns were added to the initial asset value. As a result, the asset value showed a 4.7-fold increase, which is a large return compared with the 3.4-fold increase in the S&P 500 over the same period. The prediction accuracy of the sign in the predicted value was 59.7%. Investments were not made on 916 of the 3,760 business days owing to the likely inclusion of noise. In addition, the prediction accuracy of the sign in the predicted value was 42.2% if the forecast was made for business days for which the forecast was not actually made based on the final node, which was excluded from the tree (Table 7 and Figure 2).

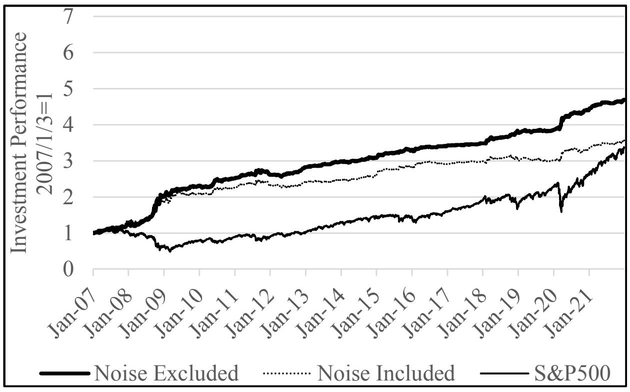

Figure 2. *Transition of the asset value by Two Stage Optimization (S&P 500 ETF, 2007/1/3 = 1).*

4.4. Evaluation of the Model Assuming Actual Operation (S&P 500, Ternary Tree)

A space containing noise in which a pattern is not easily recognized is not necessarily one side of the two subsamples generated from the division of the entire data sample. When dividing a space by a decision tree, it is desirable to search the space in more detail using a multiway classification.

The same analysis discussed in section 4.3 was performed using ternary trees instead of binary trees. As a result, the asset value increased 6.2 times, and the sign accuracy of the predicted value was 64.6%. Of the 3,760 business days, no investments were made on 941 days owing to a strong possibility of noise. The sign accuracy of the predicted value was 43.3% (Table 7 and Figure 3). The performance in terms of both the asst value and the prediction accuracy was improved by using ternary trees.

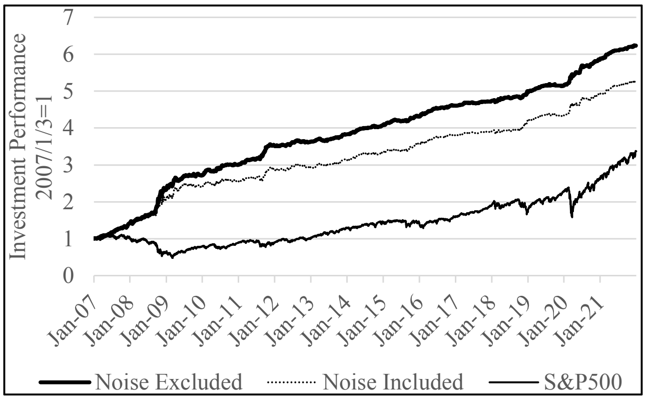

Figure 3. *Transition of the asset value by Two Stage Optimization (S&P 500 ETF, 2007/1/3 = 1).*

Table 7. *Accuracy comparison using S&P 500 ETF.*

S&P500							
		# of Cases	**Asset Value**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
	Noise Included	3,760	3.6	55.5%	60.0%	53.6%	56.6%
Binary Tree	Noise	916	n.m.	42.2%	45.3%	38.4%	41.5%
	Noise Excluded	2,844	4.7	59.7%	64.3%	58.5%	61.3%
	Noise Included	3,760	5.3	59.3%	63.5%	58.6%	61.0%
Ternary Tree	Noise	941	n.m.	43.3%	47.9%	43.4%	45.5%
	Noise Excluded	2,819	6.2	64.6%	68.6%	63.8%	66.1%

4.5. Evaluation of the Model Assuming Actual Operation (TOPIX Futures and iShares MSCI Germany ETF)

The iShares Core S&P 500 ETF is a representative indicator of the global financial markets but has been on a consistent upward trend since 2009. Therefore, the same analysis discussed in section 4.4 and 4.5 was performed using the TOPIX Futures nearby month (Table 8) and iShares MSCI Germany ETF (Table 9). Although they have a smaller trading volume than the iShares Core S&P 500 ETF, they have no long-term trends and are a better example of complex system data.

Table 8. *Financial data used to predict TOPIX Futures.*

Variables	**Name**	**Change**
Objective	TOPX Futures Nearby Month	NextBD Open~Close
Explanatory1	TOPIX	Previous BD ~ Current BD
Explanatory2	Nikkei 225/TOPIX	Previous BD ~ Current BD
Explanatory3	JPY/USD	Previous BD ~ Current BD
Explanatory4	JGB 10Y Yield	Previous BD ~ Current BD
Explanatory5	JGB 5-30Y Yield Gap	Previous BD ~ Current BD
Explanatory6	TOPIX	3BDs Ago ~ Currend BD
Explanatory7	Nikkei 225/TOPIX	3BDs Ago ~ Currend BD
Explanatory8	JPY/USD	3BDs Ago ~ Currend BD
Explanatory9	JGB 10Y Yield	3BDs Ago ~ Currend BD
Explanatory10	JGB 5-30Y Yield Gap	3BDs Ago ~ Currend BD
Explanatory11	TOPIX	7BDs Ago ~ Currend BD
Explanatory12	Nikkei 225/TOPIX	7BDs Ago ~ Currend BD
Explanatory13	JPY/USD	7BDs Ago ~ Currend BD
Explanatory14	JGB 10Y Yield	7BDs Ago ~ Currend BD
Explanatory15	JGB 5-30Y Yield Gap	7BDs Ago ~ Currend BD

Table 9. *Financial data used to predict iShares MSCI Germany ETF.*

Variables	Name	Term
Objective	iShares MSCI Germany ETF	NextBD Open~Close
Explanatory1	MSCI Germany Index	Previous BD ~ Current BD
Explanatory2	DAX/MSCI Germany Index	Previous BD ~ Current BD
Explanatory3	EUR/USD	Previous BD ~ Current BD
Explanatory4	US Treasury 10Y Yield	Previous BD ~ Current BD
Explanatory5	US Treasury 5-30Y Yield Gap	Previous BD ~ Current BD
Explanatory6	MSCI Germany Index	3BDs Ago ~ Currend BD
Explanatory7	DAX/MSCI Germany Index	3BDs Ago ~ Currend BD
Explanatory8	EUR/USD	3BDs Ago ~ Currend BD
Explanatory9	US Treasury 10Y Yield	3BDs Ago ~ Currend BD
Explanatory10	US Treasury 5-30Y Yield Gap	3BDs Ago ~ Currend BD
Explanatory11	MSCI Germany Index	7BDs Ago ~ Currend BD
Explanatory12	DAX/MSCI Germany Index	7BDs Ago ~ Currend BD
Explanatory13	EUR/USD	7BDs Ago ~ Currend BD
Explanatory14	US Treasury 10Y Yield	7BDs Ago ~ Currend BD
Explanatory15	US Treasury 5-30Y Yield Gap	7BDs Ago ~ Currend BD

The financial data used to predict the TOPIX Futures in next BD include 3,670 business day (BD) intraday returns between January 04, 2007, and December 30, 2021 as an objective variable. Also, the financial data used to predict the iShares MSCI Germany ETF in next BD include 3,776 business day (BD) intraday returns between January 03, 2007, and December 30, 2021 as an objective variable.

By using daily time series data for 1,500 business days ending on December 28, 2006, the best tree was created by applying Two Stage Optimization excluding noise. By using the data on December 29, 2006, which represented 1,501 business days later, the best tree created above was able to predict the intraday return from the opening price to the closing price on the next business day (the beginning of January 2007). If the prediction by the tree has a positive value, the investment return of the tree equals the intraday return of the underlying index; a negative value represents the intraday return with the opposite sign. If the data is classified as the final node, which was excluded from the tree owing to the low evaluation value, it is likely to contain noise; thus, no investment is made (the investment return is zero).

This process was repeated by using both binary trees and ternary trees from the beginning of January to December 30, 2021. The results are shown on Table 10 and Table 11. Although the performance with both TOPIX Futures and iShares MSCI Germany ETF in terms of both the asst value and the prediction accuracy are not as good as that with S&P 500 ETF, the asset value grew much more than that of the underlying indices over the same period. Also, the performance in terms of both the asst value and the prediction accuracy was improved by using ternary trees.

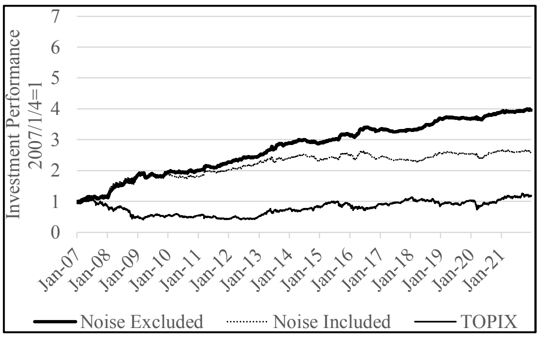

Figure 4. *Transition of the asset value by Two Stage Optimization (TOPIX Futures, 2007/1/4 = 1, Binary Tree).*

Table 10. *Accuracy comparison using TOPIX Futures.*

TOPIX							
		# of Cases	**Asset Value**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Binary Tree	Noise Included	3,670	2.6	53.2%	51.7%	54.2%	52.9%
	Noise	881	n.m.	41.2%	40.8%	42.9%	41.8%
	Noise Excluded	2,789	4.0	57.0%	55.2%	57.8%	56.5%
Ternary Tree	Noise Included	3,670	5.0	58.2%	56.5%	59.9%	58.2%
	Noise	995	n.m.	44.9%	44.2%	46.7%	45.4%
	Noise Excluded	2,675	5.8	63.1%	61.2%	64.9%	63.0%

Table 11. *Accuracy comparison using iShares MSCI Germany ETF.*

DAX							
		# of Cases	**Asset Value**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Binary Tree	Noise Included	3,776	3.1	53.0%	56.0%	52.2%	54.0%
	Noise	978	n.m.	43.9%	46.9%	43.2%	44.9%
	Noise Excluded	2,798	3.9	56.2%	59.2%	55.3%	57.2%
Ternary Tree	Noise Included	3,776	4.5	55.1%	58.2%	53.6%	55.8%
	Noise	945	n.m.	42.8%	45.6%	40.4%	42.9%
	Noise Excluded	2,831	5.3	59.2%	62.2%	58.0%	60.0%

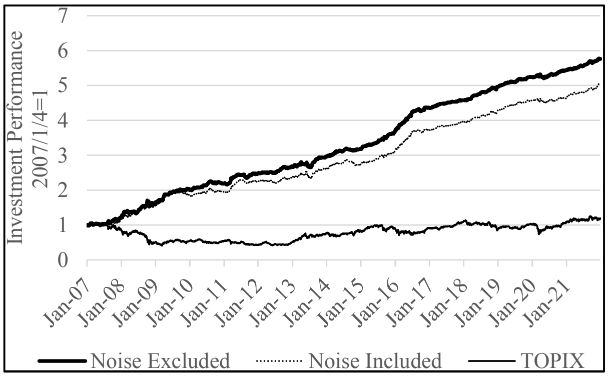

Figure 5. *Transition of the asset value by Two Stage Optimization (TOPIX Futures, 2001/1/4 = 1, Ternary Tree).*

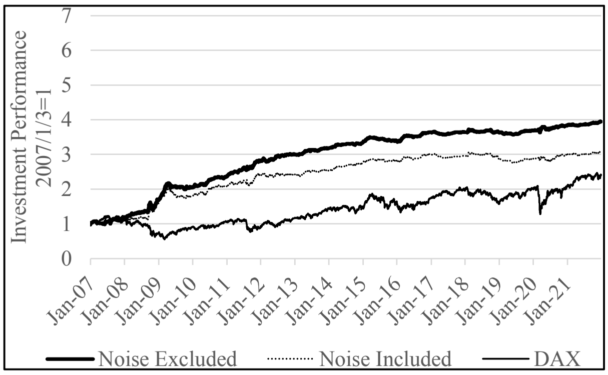

Figure 6. *Transition of the asset value by Two Stage Optimization (iShares MSCI Germany ETF, 2007/1/4 = 1, Binary Tree).*

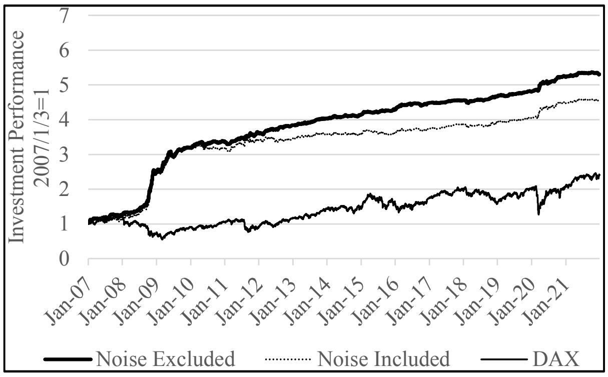

Figure 7. *Transition of the asset value by Two Stage Optimization (iShares MSCI Germany ETF, 2007/1/3 = 1, Ternary Tree).*

5. Discussion

5.1. Contributions to Literature

5.1.1. Evolutionary Computation as a Method for Building Tree

Model trees have been widely used for data analysis because of their ease of interpretation. However, if the greedy method that searches the splitting criterion recursively from the upper node to the lower node is used, overfitting is likely to occur. On the other hand, attempts have been made to collectively optimize the structure of the model tree using evolutionary computation, but in many studies, while searching for the optimum tree structure, it is divided by a specific upper node. The splitting criteria that are valid for the area is applied to the area divided by other higher-level nodes.

As an efficient method for collectively optimizing the overall structure of a tree, Two Stage Optimization, which performs the attributes of splitting criteria and their thresholds by separate evolutionary computation, has been proposed in recent previous research. This research proposed a new Two Stage Optimization that improved the problems in the previous research and surpassed the conventional methods in terms of performance from relatively simple problems to complicated problems.

5.1.2. Stock Price Prediction Using Machine Learning

There are many previous studies on stock price forecast by machine learning. On the other hand, as shown on 2.1 and 2.2, previous studies have also shown that it is difficult for a stock price forecast model to maintain stable prediction accuracy over a long period of time. One of the main reasons is that if machine learning recognizes a relatively short-term stock price driver, the model shows high prediction accuracy in a short period of time; however, if that driver wanes in influence, the prediction accuracy will decrease. In order to eliminate short-term stock price drivers as noise, it is necessary to determine in advance whether or not the data contains noise. Previous studies have actively classified supervised data by machine learning and have shown excellent results. However, studies on classification of unsupervised data, such as whether the data are noise or not, have not produced significant results. In this study, a model tree was used to divide the space containing noise and the space not containing noise, and a stock price forecast model was created in each space. The prediction accuracy in this study exceeded the existing methods, and more importantly, the long-term prediction performance of the model in this study was stable over about 20 years horizon, which was not performed by any other previous works.

5.2. Implication for Practice

Many previous studies on stock price forecast show an excellent prediction accuracy; however, when real money is actually invested using a stock price forecasting model, the excellent results shown in the paper are often not obtained. One of the main reasons is a bit-ask spread. When the trading volume is small, such as individual stocks and crypto assets, the difference between the selling price and the buying price is large, and it may not be possible to trade at the desired price. Many of the previous studies do not consider this bit ask spread.

In addition, some previous studies are predicting non-tradable price such as the S&P500, TOPIX or MSCI GERMANY INDEX. In contrast, this study predicts the prices of tradeable ETFs or futures of S&P500, TOPIX and MSCI GERMANY INDEX; therefore, there is no difference between theory and practice. In addition, since the opening and closing prices during the business day are determined in an auction format, there are no bit ask spreads such as individual stocks and crypto assets. The big problem of this study in practical use is that if you deal with a large amount of money, you will not be able to trade at the opening and closing auctions, and the results may differ from the data used in the analysis. However, with S&P500, TOPIX and MSCI GERMANY INDEX, if the amount is about USD10millioln, it will be possible to trade at auction almost every day. Therefore, it is considered that the results obtained in this study can be obtained even if a model is actually used in practice.

6. Conclusions

To avoid overfitting in data analysis, it is desirable to remove in advance extraordinary values that deviate significantly from the true pattern of the population. However, because the true pattern is unknown, the extraordinary values to be excluded are also usually unknown. In this research, the sample space was divided using a decision tree, and the versatility was evaluated by using a pattern recognition model at each final node. The spaces were divided into those with high and low versatility. Those with a low versatility were defined having relatively large noise, and a model excluding such space was created to reflect the true pattern of the group. To efficiently search conditions for excluding the space containing noise, the Two Stage Optimization was used for efficiently optimizing the splits to divide the sample space rather than using a recursive method.

It is confirmed that the globally optimal decision tree can be searched using the Two Stage Optimization, and the best decision tree obtained in this manner can be used to extract a space containing relatively large noise in a complicated problem. Moreover, the pattern recognition model constructed in the space reflecting the true pattern of the population showed higher prediction accuracy than that using the conventional method. In particular, when the prediction accuracy of the model was confirmed by the walk-forward method using financial time series data under the same conditions as those of actual operation, an investment performance that stably exceeded the return of the benchmark assets over the past 12 years was obtained.

A space containing noise in which a pattern is not easily recognized is not necessarily one side of the two subsamples generated from the division of the entire data sample. Therefore, when dividing a space by a decision tree, it is desirable to subdivide the space by a polyadic tree. In this research, it was confirmed that the prediction accuracy improved when the binary tree was changed to a ternary tree. On the contrary, if the number of branches is further increased, the number of elements included in the final node decreases when the number of training data are limited, and the versatility of the pattern recognition model is lost. In this research, analysis using daily data is conducted. In the future, a similar study will be conducted on multiway trees using high-frequency data to obtain additional data.

Highlight

1) The accuracy of stock price prediction can be greatly improved if pattern recognition excluding spaces containing noise is used.
2) A globally optimized model tree can effectively extract a space containing noise.
3) A model tree can be globally optimized by Two Stage Optimization.
4) CMA-ES is suitable for threshold optimization when attributes are given to each split in a model tree.
5) A ternary tree is better than a binary tree for removing noise.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

[1] Bruno, M. H., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, 124, 226-251. https://doi.org/10.1016/j.eswa.2019.01.012

[2] Obthong, M., Tantisantiwong, N., Jeamwatthanachai, W., & Wills, G. (2020). A survey on machine learning for stock price prediction: algorithms and techniques. *2nd International Conference on Finance, Economics, Management and IT Business*, 63-71. https://doi.org/10.5220/0009340700630071

[3] Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, 4 (9). https://doi.org/10.3390/asi4010009

[4] Kumar, S., Gupta, R., Kumar, P., & Aggarwal, N. (2021). A Survey on Artificial Neural Network based Stock Price Prediction Using Various Methods. *5th International Conference on Intelligent Computing and Control Systems*. https://doi.org/10.1109/ICICCS51141.2021.9432329

[5] Maqsood, H., Mehmood, I., Maqsood, M., Yasir, M., Afzal, S., Aadil, F., Selim, M. M., & Muhammad, K. (2020). A local and global event sentiment based efficient stock exchange forecasting using deep learning. *International Journal of Information Management*, 50, 432-451. https://doi.org/10.1016/j.ijinfomgt.2019.07.011

[6] Jin, Z., Yang, Y., & Liu, Y. (2020). Stock closing price prediction based on sentiment analysis and LSTM. *Neural Computing and Applications*, 32 (13), 9713–9729. https://doi.org/10.1007/s00521-019-04504-2

[7] Shen, J., & Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. Journal of Big Data, 7 (66). https://doi.org/10.1186/s40537-020-00333-6

[8] Rivera-Lopeza, R., Canul-Reich, J., Mezura-Montes, E., & Cruz-Chávezd, M. A. (2021). Induction of decision trees as classification models through metaheuristics. *Swarm and Evolutionary Computation*, https://doi.org/10.1016/j.swevo.2021.101006

[9] Tanigawa, T., & Zhao, Q. (2000). A study on efficient generation of decision trees using genetic programming, *Genetic and Evolutionary Computation Conference 2000*, 1047–1052. https://dl.acm.org/doi/pdf/10.5555/2933718.2933915

[10] Falcón-Cardona, J. G., & Coello, C. A. C. (2020). Indicator-based multi-objective evolutionary algorithms: A comprehensive survey. *ACM Computing Surveys*, 53 (2), 1-35. https://dl.acm.org/doi/pdf/10.1145/3376916

[11] Chung, H., & Shin, K. (2020). Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction. *Neural Computing and Applications*, 32 (12), 7897–7914. https://doi.org/10.1007/s00521-019-04236-3

[12] Kamalov, F. (2020). Forecasting significant stock price changes using neural networks. *Neural Computing and Applications*, 32 (23), 17655–17667. https://doi.org/10.1007/s00521-020-04942-3

[13] Rezaei, H., Faaljou, H., & Mansourfar, G. (2021). Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications*, 169, 114332. https://doi.org/10.1016/j.eswa.2020.114332

[14] Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks. Neural Computing and Applications, 32 (6), 1609–1628. https://doi.org/10.1007/s00521-019-04212-x

[15] Harikrishnan, H., & Urolagin, S. (2020). Prediction of Stock Market Prices of Using Recurrent Neural Network—Long Short-Term Memory. https://link.springer.com/chapter/10.1007/978-981-15-5243-4_33

[16] Lu, W., Li, J., Wang, J. & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications*, 33 (10), 4741–4753. https://doi.org/10.1007/s00521-020-05532-z

[17] Ta, V., Liu, C., & Tadesse, D. A. (2020). Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Applied Science*, 10 (2), 437–456. https://doi.org/10.3390/app10020437

[18] Lee, S. W., & Kim, H. Y. (2020). Stock market forecasting with super-high dimensional time-series data using ConvLSTM, trend sampling, and specialized data augmentation. Expert Systems with Applications, 161, 113704. https://doi.org/10.1016/j.eswa.2020.113704

[19] Urolagin, S., Sharma, N., & Datta, T. K. (2021). A combined architecture of multivariate LSTM with Mahalanobis and Z-Score transformations for oil price forecasting. *Energy*, 231, 120963. https://doi.org/10.1016/j.energy.2021.120963

[20] Livieris, I., Pintelas, E., & Pintelas, A. (2020). A CNN–LSTM model for gold price time-series forecasting. Neural Computing and Applications, 32 (23), 17351–17360. https://doi.org/10.1007/s00521-020-04867-x

[21] Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. (2018). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, 47, 552–567. https://doi.org/10.1016/j.najef.2018.06.013

[22] Xiao, C., Xia, W., & Jiang, J. (2020). Stock price forecast based on combined model of ARI-MA-LS-SVM. *Neural Computing and Applications*, 32 (10), 5379–5388. https://doi.org/10.1007/s00521-019-04698-5

[23] Yang, R., Yu, L., Zhao, Y., Yu, H., Xu, G., Wu, Y., Liu, Z. (2020). Big data analytics for financial Market volatility forecast based on support vector machine, *International Journal of Information Management*, 50, 452-462. https://doi.org/10.1016/j.ijinfomgt.2019.05.027

[24] Pandey, M. K., Mittal, M., & Subbiah, K. (2021). Optimal balancing & efficient feature ranking approach to minimize credit risk. *International Journal of Information Management Data Insights*, 1 (2), 100037. https://doi.org/10.1016/j.jjimei.2021.100037

[25] Ma, X., Yang, R., Zou, D., & Liu, R. (2020). Measuring extreme risk of sustainable financial system using GJR-GARCH model trading data-based, International Journal of Information Management, 50, 526-537. https://doi.org/10.1016/j.ijinfomgt.2018.12.013

[26] Urolagin, S. (2017). Text Mining of Tweet for Sentiment Classification and Association with Stock Prices, *International Conference on Computer and Applications*. https://doi.org/10.1109/COMAPP.2017.8079788

[27] Seong, N., & Nam, K. (2021). Predicting stock movements based on financial news with segmentation. *Expert Systems with Applications*, 164, 113988. https://doi.org/10.1016/j.eswa.2020.113988

[28] Xu, Q., Wang, L., Jiang, C., & Liu, Y. (2020) A novel (U) MIDAS-SVR model with multi-source market sentiment for forecasting stock returns. *Neural Computing and Applications*, 32 (10), 5875–5888. https://doi.org/10.1007/s00521-019-04063-6

[29] Tandon, C., Revankar, S., & Parihar, S. S. (2021). How can we predict the impact of the social media messages on the value of cryptocurrency? Insights from big data analytics. *International Journal of Information Management Data Insights*, 1 (2), 100035. https://doi.org/10.1016/j.jjimei.2021.100035

[30] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed). Springer Series in Statistics.

[31] Seni, G., & Elder, J. (2010). Ensemble methods in data mining: Improving accuracy through combining predictions. Morgan & Claypool Publishers.

[32] Freitas, A. A. (2002). Data mining and knowledge discovery with evolutionary algorithms. New York, NJ: Springer-Verlag New York, Inc.

[33] Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21 (3), 660–674. https://doi.org/10.1109/21.97458

[34] Murthy, S. K. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, *2* (4), 345–389. https://doi.org/10.1023/A:1009744630224

[35] Freitas, A. A. (2004). A critical review of multi-objective optimization in data mining: A position paper. *ACM SIGKDD Explorations Newsletter*, *6* (2), 77–86. https://doi.org/10.1145/1046456.1046467

[36] Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers—A survey. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, *35* (4), 476–487. https://doi.org/10.1109/TSMCC.2004.843247

[37] Espejo, P. G., Ventura, S., & Herrera, F. (2010). A survey on the application of genetic programming to classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *40* (2), 121–144. https://doi.org/10.1109/TSMCC.2009.2033566

[38] Barros, R. C., Basgalupp, M. P., de Carvalho, A. C. P. L. F., & Freitas, A. A. (2012). A survey of evolutionary algorithms for decision tree induction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *42* (3), 291–312. https://doi.org/10.1109/TSMCC.2011.2157494

[39] Shirasaka, M., Zhao, Q., Hammami, O., Kuroda, K., & Saito, K. (1998). Automatic design of binary decision trees based on genetic programming, *Second Asia-Pacific Conference on Simulated Evolution and Learning*.

[40] Zhao, Q., & Shirasaka, M. (1999). A study on evolutionary design of binary decision trees, *IEEE Congress on Evolutionary Computation* (pp. 1988–1993).

[41] Aitkenhead, M. J. (2008). A co-evolving decision tree classification method. *Expert Systems with Applications*, *34* (1), 18–25. https://doi.org/10.1016/j.eswa.2006.08.008

[42] Papagelis, A., & Kalles, D. (2001). Breeding decision trees using evolutionary techniques, *Eighteenth International Conference on Machine Learning* (pp. 393–400). Morgan Kaufmann Publishers, Inc.

[43] Fu, Z., Golden, B. L., Lele, S., Raghavan, S., & Wasil, E. A. (2003). A genetic algorithm-based approach for building accurate decision trees. *INFORMS Journal on Computing*, *15* (1), 3–22. https://doi.org/10.1287/ijoc.15.1.3.15152

[44] Burgess, C. J., & Lefley, M. (2001). Can genetic programming improve software effort estimation? a comparative evaluation. *Information and Software Technology*, *43* (14), 863–873. https://doi.org/10.1016/S0950-5849(01)00192-6

[45] DeLisle, R. K., & Dixon, S. L. (2004). Induction of decision trees via evolutionary programming. *Journal of Chemical Information and Computer Sciences*, *44* (3), 862–870. https://doi.org/10.1021/ci034188s

[46] Zhao, H. (2007). A multi-objective genetic programming approach to developing pareto optimal decision trees. *Decision Support Systems*, *43* (3), 809–826. https://doi.org/10.1016/j.dss.2006.12.011

[47] To, C., & Pham, T. (2009). Analysis of cardiac imaging data using decision tree based parallel genetic programming, *6th International Symposium on Image and Signal Processing and Analysis* (pp. 317–320).

[48] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, *104* (1), 148–175. https://doi.org/10.1109/JPROC.2015.2494218

[49] Adams, R. P., & Stegle, O. (2008). Gaussian process product models for nonparametric nonstationarity, *International Conference on Machine Learning* (pp. 1–8).

[50] Larraga, R. E., Lozano, J. A., & Pena, J. M. (1999). A review of cooperation between evolutionary computation and probabilistic graphical models, *Second Symposium on Artificial Intelligence CIMAF 99* (pp. 314–324).

[51] Davis, L. (1990). The handbook of genetic algorithms. New York: Van Nostrand Reinhold.

[52] Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies: A comprehensive introduction. *Natural Computing*, *1* (1), 3–52. https://doi.org/10.1023/A:1015059928466

[53] Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11* (4), 341–359 (1197). https://doi.org/10.1023/A:1008202821328

[54] Kennedy, J., & Eberhart, R. C. Particle swarm optimization. (2005). *IEEE international joint conference on neural networks*, pp. 1942–1948 (1995).

[55] Sato, H., Ono, I., & Kobayashi, S. (1997). A new generation alternation model of genetic algorithms and its assessment. *Journal of Japanese Society for Artificial Intelligence*, 12 (5), 734–744.

[56] Ono, I., Kobayashi, S., & Yoshida, K. (2000). Optimal lens design by real-coded genetic algorithms using UNDX. *Computer Methods in Applied Mechanics and Engineering*, *186* (2–4), 483–497. https://doi.org/10.1016/S0045-7825(99)00398-9

[57] Kobayashi, S. (2009). The frontiers of real-coded genetic algorithms. *Journal of Japanese Society for Artificial Intelligence*, 24 (1), 128–143.

[58] Chakraborty, A., & Kar, A. K. (2017). Swarm intelligence: A review of algorithms. *Nature-Inspired Computing and Optimization*, 475-494. https://doi.org/10.1007/978-3-319-50920-4_19

[59] Slowik, A., & Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications*, 32 (16), 12363–12379. https://doi.org/10.1007/s00521-020-04832-8

[60] Jain, R., Batra, J., Kar, A. K., Agrawal, H., & Tikkiwal, V. A. (2021). A hybrid bio-inspired computing approach for buzz detection in social media. *Evolutionary Intelligence*. https://doi.org/10.1007/s12065-020-00512-7

[61] Anupam, S., & Kar, A. K. (2021). Phishing website detection using support vector machines and nature-inspired optimization algorithms. *Telecommunication Systems*, 76 (1), 17-32. https://doi.org/10.1007/s11235-020-00739-w

[62] Batra, J., Jain, R., Tikkiwal, V. A., & Chakraborty, A. (2021). A comprehensive study of spam detection in e-mails using bio-inspired optimization techniques. International Journal of Information Management Data Insights, 1 (1), 100006. https://doi.org/10.1016/j.jjimei.2020.100006

[63] Kar, A. K., & Aswani, R. (2021). How to differentiate propagators of information and misinformation–Insights from social media analytics based on bio-inspired computing. *Journal of Information and Optimization Sciences*, 42 (6), 1307-1335. https://doi.org/10.1080/02522667.2021.1880147

[64] Hansen, N., & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, *9* (2), 159–195. https://doi.org/10.1162/106365601750190398

[65] Fukushima, N., Nagata, Y., Kobayashi, S., & Ono, I. (2011). Proposal of distance-weighted exponential natural evolution strategies, *IEEE congress on evolutionary computing* (pp. 164–170).

[66] Masutomi, K., Nagata, Y., & Ono, I. (2015). A novel evolution strategy for noisy function optimization, *Transaction of the Japanese Society for Evolutionary Computation*, 6 (1), 1–12. https://doi.org/10.11394/tjpnsec.6.1

[67] Hansen, N., Niederberger, A. S. P., Guzzella, L., & Koumoutsakos, P. (2009). A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. *IEEE Transactions on Evolutionary Computation*, *13* (1), 180–197. https://doi.org/10.1109/TEVC.2008.924423

[68] Richter, S. N., Schoen, M. G., & Tauritz, D. R. (2019). Evolving mean-update selection methods for CMA-ES, *Proceedings of the 2019 Genetic and Evolutionary Computation Conference* (pp. 1513–1517).

[69] Senoguchi, J. (2021). Forecast of complex financial big data using model tree optimized by bilevel evolution strategy, *Journal of Big Data*, 8 (116), 1-13.

[70] Senoguchi, J (2021). Extraction of Space Containing Noise and Forecast of Complex Data by Multiway Tree Bi-Level GA, *The Japanese Society for Artificial Intelligence*, 36 (5), 1-12.

[71] Zhang, Y., Ma, Q., Sakamoto, M., & Furutani, H. (2011). Experimental analysis of the first appearing time of optimum solution in genetic algorithm, *Journal of Information Processing*, 4 (1), 82–88.