International Journal of Data Science and Analysis
2021; 7(6): 150-160
http://www.sciencepublishinggroup.com/j/ijdsa
doi: 10.11648/j.ijdsa.20210706.13
ISSN: 2575-1883 (Print); ISSN: 2575-1891 (Online)

SciencePG
Science Publishing Group

A Machine Learning Approach for the Short-term Reversal Strategy

Zheng Tan[1, *], Yan Li[2], Chulwoo Han[3, *]

[1]Institute of Information Science and Technology, Chengdu Polytechnic, Chengdu, P. R. China

[2]R&D Department, Xiyuan Quantitative Technology, Chengdu, P. R. China

[3]Durham Business School, Durham University, Durham, UK

Email address:
zhengtan1983@yahoo.com (Zheng Tan), chulwoo.han@durham.ac.uk (Chulwoo Han)

[*]Corresponding author

To cite this article:
Zheng Tan, Yan Li, Chulwoo Han. A Machine Learning Approach for the Short-term Reversal Strategy. *International Journal of Data Science and Analysis*. Vol. 7, No. 6, 2021, pp. 150-160. doi: 10.11648/j.ijdsa.20210706.13

Received: October 25, 2021; **Accepted**: November 12, 2021; **Published**: November 17, 2021

Abstract: The short-term reversal effect is a pervasive and persistent phenomenon in worldwide financial markets that has been found to generate abnormal returns not explainable by traditional asset pricing models. In contrast to the linear model employed in most studies on the short-term reversal, this article aims to establish a nonlinear framework to study the reversal anomaly, by using machine learning approaches. Machine learning methods including Random Forest, Adaptive Boosting, Gradient Boosted Decision Trees and extreme Gradient Boosting, are employed to test the profitability of the short-term strategy in the US and Chinese stock markets. Significant outperformances with extremely high Sharpe ratio, moderate kurtosis, and positive skewness are found, showing remarkable classification efficiency of the machine learning models and their applicability to various markets. Further studies reveal that the strategy returns can be weakened with the extension of the holding period. Notably, by comparing the performances of machine learning with our newly developed linear reversal strategy, the nonlinear methods are proved to be capable of providing a diversified model predictability with improved classification accuracy. Our research indicates the significant potential of machine learning in resolving the stock return and feature relationship, which can be helpful for quantitative traders to make profitable investment decisions.

Keywords: Finance, Artificial Intelligence, Reversal Trading, Stock Market

1. Introduction

The short-term reversal is a well-known anomaly in the financial markets that have existed for a long time. The phenomenon is described as that stocks with relatively low (high) returns over the past month or week earn positive (negative) abnormal returns in the following month or week, thus buying losers (or selling winners) would generate persistent profits. It is claimed that such abnormal effect cannot be explained by the traditional asset pricing models [8]. A large volume of literature has been dedicated to the research of the short-term reversal anomaly, especially in the stock market. Jegadeesh [13] documents that an equally-weighted portfolio that buys losers and sells winners from the past one-month horizon earns an average return of approximately 2% per month for the period from 1934 to 1987. Using weekly stock returns, Lehmann [16] and Cooper [7] report similar findings. Groot et al. [22] construct a daily-rebalanced reversal portfolio based on the past-week returns and show that it yields a gross return of 61.7 basis points per week. The underlying reason of the reversal effect has been attributed to the investor overreaction [16], the transient liquidity shocks [4, 6, 14].

It is notable that most studies on the short-term reversal employ the linear quantile partition scheme, i.e., sorting stocks based on the size of a past return and building a reversal portfolio by buying losers (bottom quantile) and selling winners (top quantile). Although machine learning approaches have become popular in asset pricing over the past few years [1, 15, 20, 23], the study of the short-term reversal based on the machine learning is relatively primitive. Preliminary investigations of AI assisted momentum and reversal trading can be seen in Li and Tam [17], where the market state defined by the returns in the past observation period is learned to predict the possible stock selection policies. In contrast to Li and Tam [17], this paper aims to establish a new nonlinear framework to study the short-term reversal, by predicting the stock classification in the look-ahead time horizon.

Tree-based machine learning algorithms, including the Random Forest (RF), Adaptive Boosting (AdaBoost), Gradient Boosted Decision Trees (GBDT) and eXtreme Gradient Boosting (XGBoost), are employed to serve this objective. These models are trained on the past short-term return features to learn the target classes defined based on the future excess returns over a certain holding period. The trading strategy is implemented in a rolled training and trading scheme, where a reversal portfolio is constructed from the classified stocks with the help of a probability ranking scheme. The models are applied to the US and the Chinese stock markets. Testing in these countries allows us to compare the effectiveness of the models in a developed market and an emerging market. We also develop a novel linear reversal strategy and compare it with the machine learning strategies. The linear reversal strategy utilizes past returns at different periods over the past one month. This approach is different from the conventional short-term reversal strategies that typically rely on a single past return. By comparing the machine learning strategies with the linear strategy, we can identify the role of nonlinear information in predicting future returns.

The modeling framework of this paper follows that of Tan, Yan, and Zhu [21], who use the Random Forest to illustrate the relationship between stock classification and underlying fundamentals. This paper, however, extends their work and include more sophisticated tree-based models such as AdaBoost, GBDT and XGBoost to examine the prediction power of the boosting algorithms.

The contribution of this paper is threefold. Firstly, we employ machine learning methods to exploit the short-term reversal effect and find significant profitability of the portfolios derived from them. Secondly, a novel linear reversal strategy is developed based on multiple look-back periods, and a comparison between the linear and nonlinear frameworks is conducted to evaluate the nonlinear contents that can be informative for performance improvements. Finally, we implement the models in the US and the Chinese markets and assess the significance of this anomaly in both developed and emerging markets.

The paper is organized as follows. Section 2 describes the methodology used in the empirical study, which includes the data and software, the rolled training and trading scheme, the feature space and labelling, the machine learning approaches, and the linear reversal strategy. Section 3 presents empirical results and discusses the main findings of our study. Section 4 concludes.

2. Methodology

In order to exploit the short-term reversal effect, we build a machine learning framework by using the tree-based models including RF, AdaBoost, GBDT and XGBoost. The short-term stock returns are predicted and the strategy profitability is tested on both US and Chinese markets. We also develop a new linear reversal strategy to investigate the discrepancy of the classification efficiency between the linear and nonlinear algorithms. The trading scheme basically follows the work of Tan, Yan, and Zhu [21], the research, however, covers a much wider scenario of the reversal strategy. More details of the methodology are given below.

2.1. Data and Software

For the study of the US market, we use all firms listed on NYSE, Amex, and Nasdaq, and employ the SP500 index as a benchmark. The US stock data, which contains price and volume, are obtained on a daily basis from the Center for Research in Security Prices (CRSP). For the study of the Chinese market, we consider all stocks listed on Shenzhen and Shanghai Stock Exchanges and employ the CSI 500 index as the benchmark. The Chinese data are obtained from the Wind financial database in daily frequency. Any stocks that have been traded less than one year are eliminated from the sample.

The preprocessing and data handling are conducted using MATLAB. The RF, AdaBoost and GBDT models are implemented using scikit-learn, a Python library that integrates a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. The XGBoost model is implemented using the XGBoost library.

2.2. Rolled Training and Trading Scheme

The entire sample is divided into a series of training sets and trading sets on a rolling basis, and the out-of-sample trading period is from 1995.01.01 to 2018.12.31 for the US market, and from 2010.06.01 to 2019.03.14 for the Chinese market. The models are trained in each training period to make predictions in the subsequent trading period.

The out-of-sample trading period is split into a series of non-overlapping sub-periods that consist of 60 trading days (approximately three months). The training set corresponding to each trading sub-period contains 250 trading days (approximately one year). In each training dataset, past returns of the stocks are generated to be used as input features and the stocks are divided into N classes based on a future excess return (equity return – benchmark return). These classes define the target variable. We set N equal to 3 for the empirical study. The machine learning models are trained in the training set and used in the subsequent trading period to predict each stock’s probability belonging to each class. Ten stocks with the highest probabilities for the first class, *i.e.*, the class with the largest excess returns are selected on each trading date to form an equally-weighted portfolio, which is rebalanced every two days. All trades are assumed to be subject to a transaction cost of 0.1% for the US market, and 0.16% for the Chinese market.

2.3. Past Return Feature Space and Labelling

For each training period, we generate the input feature space and the target variable (output) as follows.

Input: The input feature space is a u×v matrix, where u is the number of samples and v is the number of features. The past return features are defined as follows:

$$Ret_{t,m}^{s} = \frac{P_t^s}{P_{t-m}^s} - \frac{P_t^I}{P_{t-m}^I} \quad (1)$$

where P_t^s and P_t^I denote the close prices of stock s and the benchmark index at time t, respectively. Past returns up to twenty days, i.e., $m = 1, \ldots, 20$ are considered. The past return features are detrended by subtracting the index return from the equity return. This approach is slightly different from the design in Tan, Yan, and Zhu [21].

Output: At time t, the returns of each stock $R_{t,m}^s$ and the benchmark index $R_{t,m}^I$ over the subsequent m holding days are calculated in Eq. (2) and (3). The holding period m is set to 2 days in the main empirical study and is set to other values (2, 5, 10 and 20 days) in the holding period dependence analysis. The excess return ER_t^s is the difference between the stock return and the index return, as described in Eq. (4). We sort all the stocks on the size of the excess returns in descending order and equally divide them into N classes. The class label is the target variable that we aim to predict.

$$R_{t,m}^s = \frac{P_{t+m}^s}{P_t^s} - 1 \quad (2)$$

$$R_{t,m}^I = \frac{P_{t+m}^I}{P_t^I} - 1 \quad (3)$$

$$ER_{t,m}^s = R_{t,m}^s - R_{t,m}^I \quad (4)$$

2.4. Machine Learning Methods

2.4.1. Random Forest

We construct our random forest model by following the conventional approach given in Breiman [3]. In principle, the random forest consists of many deep but uncorrelated decision trees built upon different samples of the data. The process of constructing a random forest is simple. For each decision tree, we first randomly generate a subset as a sample from the original dataset. Then, we grow a decision tree with this sample to its maximum depth of J_{RF} . In each split, m_{RF} features are selected at random from all the features. This procedure is repeated to generate n_{RF} decision trees. The final output is an ensemble of all decision trees, and the classification is conducted via a majority vote. We realize that the random feature selection might cause bias in the tree construction, due to the feature un-informativeness [18]. A more sophisticated feature selection scheme can be planned in the future work for the possible performance enhancement.

Three parameters must be tuned to ensure the robustness of the random forest, i.e., the number of trees n_{RF}, the maximum depth J_{RF} and the number of features m_{RF} in each split. We perform a shallow tree construction with $J_{RF} = 3$. Regarding the feature subsampling, we typically choose $m_{RF} = \sqrt{p}$, where p is the total number of features [12]. As to the number of trees n_{RF}, we set it to 100. The dependency of strategy performance on the number of trees (as shown in the Appendix) is carefully tested, where little variances on the daily mean returns are found indicating a proper hyper-parametrization and the model robustness. Note that all the presented model hyperparameters (including models given bellow) are similarly tuned to ensure the strategy validity.

2.4.2. Adaptive Boosting

Boosting was first introduced in Schapire [19] as a method to integrate a bundle of weak learning models into one to achieve enhanced prediction accuracy. The algorithm is formalized in Freund and Schapire [10], and works by sequentially applying weak learners to re-weighted versions of the training data [11]. After each boosting iteration, misclassified examples have their weights increased, and correctly classified examples have their weights decreased. After a number of iterations, the predictions of the series of weak classifiers are combined by a weighted majority vote into a final prediction. We select decision trees as the weak classifier, and implement AdaBoost by setting the boosting iterations $N_{AdaBoost} = 70$ and the learning rate $\lambda_{AdaBoost} = 1$. The depth of the tree $J_{AdaBoost}$ is chosen to be 3.

2.4.3. Gradient Boosted Decision Trees

Gradient Boosted Decision Trees (GBDT) is a variation of boosting introduced by Friedman [9]. It iteratively trains the decision tree on the classification residual from the previous decision tree, while simultaneously setting a shrinkage rate to avoid overfitting. It also employs a feature subsampling scheme to increase computational efficiency. To implement GBDT, we have to determine four parameters: the number of trees or boosting iterations N_{GBDT}, the depth of the tree J_{GBDT}, the learning rate λ_{GBDT}, and the subset of features to use at each split M_{GBDT}. We conservatively select the number of trees N_{GBDT} to be 100 to avoid overfitting, as suggested by the standard literature [11], and set the learning rate $\lambda_{GBDT} = 0.1$. Shallow trees are constructed during the iterations with the depth of the tree $J_{GBDT} = 3$; the number of features in the subsampling M_{GBDT} is chosen to be the square root of the total number of features, which is in line with the subsampling scheme in the RF.

2.4.4. Extreme Gradient Boosting

XGBoost is proposed as a scalable machine for tree boosting [5], which can be viewed as an upgraded version of the gradient boosted tree model by enabling an approximate tree learning paradigm with parallel computing. The model performs a second-order Taylor expansion on a regularized loss function and proposes a quantile sketching scheme for the best split finding as an alternative to the exact greedy algorithm, which is usually employed by the conventional tree learning. Since this method avoids enumerating over all the possible splits on all the features, the tree boosting can be implemented in a more computationally efficient way. We set the number of trees or boosting iterations $N_{XGBoost} = 100$ and the learning rate $\lambda_{XGBoost} = 0.1$ as done in the GBDT. We choose slightly deeper trees by setting the depth of the tree $J_{XGBoost} = 6$, as the faster approximate learning scheme allows us to save the computational cost. The threshold $G_{XGBoost}$ for the reduction of the loss function when splitting a leaf node is chosen to be 0.1. No subsampling procedure is used in the current XGBoost study.

2.5. Linear Reversal Strategy

The linear reversal model is an ensemble model of short-term reversal strategies with different look-back periods. The portfolio is constructed by sorting all available stocks based on the Z-scores of past returns with various lags. The Z-score is a normalized score of a sample and is calculated by subtracting the population mean from an individual raw score and then dividing the difference by the population standard deviation. The Z-score is calculated for the past returns defined in Eq. (1) with m = 1, …, 20, and the overall Z-score is computed as the average of the Z-scores of all the individual past return features. We construct an equally-weighted portfolio using the ten stocks with the lowest overall Z-scores and rebalance it every two days.

Table 1. *Daily return characteristics and annualized risk matrices of the RF, AdaBoost, GBDT, XGBoost, and linear strategies in the US market. A transaction cost of 0.1% is assumed.*

	1995.01.01-2018.12.31					
	XGBoost	**GBDT**	**AdaBoost**	**RF**	**Linear**	**SP500**
daily mean return	0.0101	0.0095	0.0087	0.009	0.0081	0.0003
t-statistics	22.2568	21.6390	19.1956	21.0787	17.6257	2.2692
standard deviation	0.0351	0.0343	0.0354	0.0331	0.0359	0.0117
skewness	1.4442	1.1365	1.5488	1.0981	1.5728	-0.0855
kurtosis	13.3680	10.2756	17.2263	10.2654	14.9445	11.4960
5-percent VaR	0.0387	0.0383	0.0397	0.0391	0.0402	0.0182
maximum drawdown	0.5306	0.6352	0.5105	0.587	0.6757	0.5713
annual return	2.5359	2.4042	2.2033	2.2621	2.0524	0.0858
Sharpe	4.4924	4.3648	3.8681	4.2484	3.5476	0.3014
Sortino	5.6925	5.4088	4.8065	5.1727	4.4294	0.2912
Calmar	4.7792	3.7848	4.3159	3.8535	3.0373	0.1502

3. Results and Discussions

3.1. Case of US Market (1995.01.01 ~ 2018.12.31)

3.1.1. Machine Learning And Linear Reversal Strategy Performances

As shown in Table 1, for the US stock market, all the machine learning strategies and the linear reversal strategy significantly outperform the market during the test period from 1995.01.01 to 2018.12.31. The portfolios earn mean daily returns around 1%, which is equivalent to a tremendous annualized mean return of 200%. The associated *t*-statistic indicates that the mean returns are significantly different from 0. Note that all the strategies exhibit positive skewness, which is a desirable property for investors. The 5-percent VaR is 3.87%, 3.83%, 3.97%, 3.91%, and 4.02% for the XGBoost, GBDT, AdaBoost, RF, and linear strategy, respectively, which are about twice the level of the market. The greater VaR can be attributed to the enlarged volatility, as evidenced by the standard deviation.

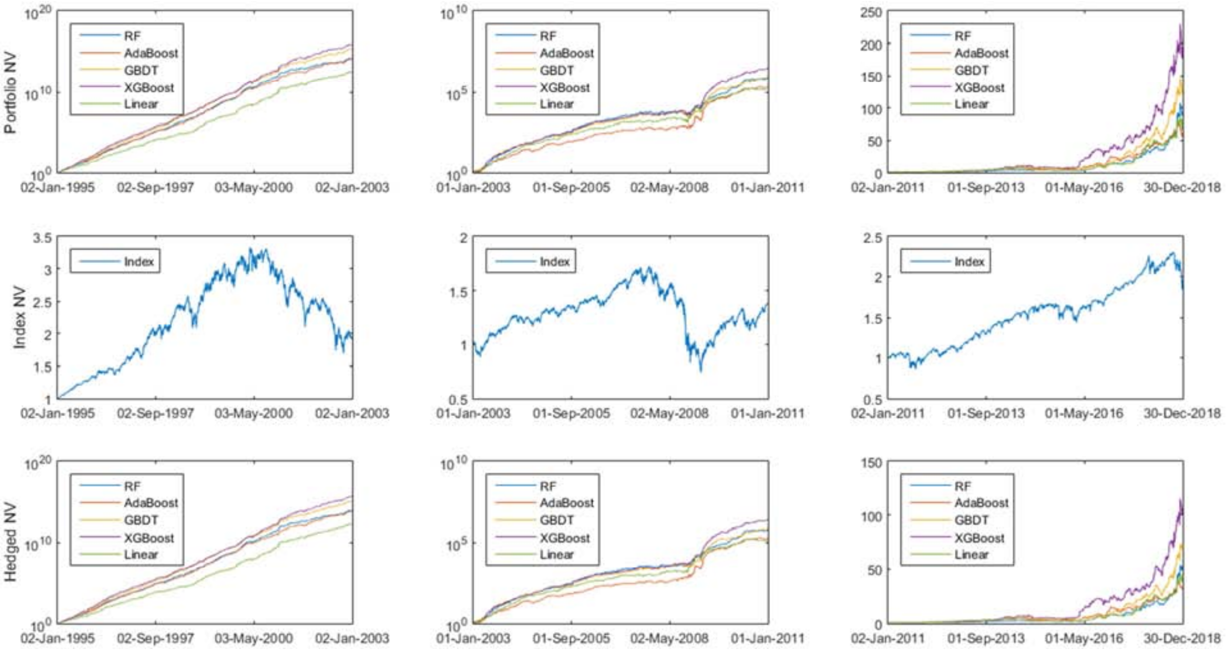

Figure 1. *Net asset values of the RF, AdaBoost, GBDT, XGBoost, and linear strategies in the US market in the sub-periods. The first row displays the net values of the strategies, the second row the net value of the benchmark, and the third row the net values of the strategies hedged by the index. A transaction cost of 0.1% is assumed.*

Table 2. *Sub-period daily return characteristics and annualized risk matrices of the RF, AdaBoost, GBDT, XGBoost, and linear strategies in the US market. A transaction cost of 0.1% is assumed.*

	1995.01.01-2002.12.31					
	XGBoost	**GBDT**	**AdaBoost**	**RF**	**Linear**	**SP500**
daily mean return	0.0191	0.0184	0.0171	0.0171	0.0153	0.0004
t-statistics	21.6824	21.2031	19.0303	20.2297	16.269	1.4772
standard deviation	0.0396	0.039	0.0403	0.0379	0.0421	0.012
skewness	0.8935	0.9703	0.8978	0.8717	1.349	-0.036
kurtosis	7.0732	7.9119	10.3651	8.1442	11.2006	5.9348
5-percent VaR	0.0383	0.039	0.0426	0.0399	0.0435	0.0193
maximum drawdown	0.4254	0.4252	0.3284	0.3644	0.4036	0.4915
annual return	4.8236	4.6461	4.3021	4.3031	3.8442	0.0995
Sharpe	7.622	7.4517	6.6846	7.1059	5.7099	0.365
Sortino	10.2564	9.7783	8.4924	9.1345	7.4688	0.3515
Calmar	11.3377	10.9261	13.1001	11.8088	9.5241	0.2025

	2003.01.01-2010.12.31					
	XGBoost	**GBDT**	**AdaBoost**	**RF**	**Linear**	**SP500**
daily mean return	0.008	0.0074	0.0068	0.0072	0.0066	0.0003
t-statistics	10.6637	9.9527	8.3142	10.2128	8.6404	0.8406
standard deviation	0.0337	0.0332	0.0365	0.0318	0.0341	0.0134
skewness	1.6121	1.2226	2.3931	1.1246	1.7959	0.0065
kurtosis	15.4511	13.5733	27.1145	12.1105	20.9315	14.1975
5-percent VaR	0.0413	0.0388	0.0387	0.0394	0.0387	0.0191
maximum drawdown	0.5306	0.6352	0.5105	0.587	0.5502	0.5678
annual return	2.0154	1.8548	1.7023	1.8217	1.6523	0.0633
Sharpe	3.7159	3.4636	2.8891	3.5531	3.0009	0.1563
Sortino	4.475	4.0084	3.5581	4.0504	3.5732	0.1522
Calmar	3.7983	2.9199	3.3345	3.1033	3.003	0.1114

	2011.01.01-2018.12.31					
	XGBoost	**GBDT**	**AdaBoost**	**RF**	**Linear**	**SP500**
daily mean return	0.003	0.0028	0.0024	0.0026	0.0026	0.0004
t-statistics	4.6372	4.5554	4.0671	4.3365	4.0114	1.8338
standard deviation	0.0295	0.0278	0.0263	0.0271	0.0293	0.0092
skewness	2.0814	0.7732	0.6405	1.0141	1.1524	-0.4452
kurtosis	30.3766	9.2902	8.9973	11.7468	12.0549	7.9361
5-percent VaR	0.0381	0.0378	0.0372	0.0377	0.0393	0.0153
maximum drawdown	0.4732	0.5793	0.4715	0.4458	0.6757	0.1978
annual return	0.7677	0.7107	0.6023	0.6606	0.6598	0.0947
Sharpe	1.5778	1.5449	1.3684	1.4657	1.3558	0.4436
Sortino	1.7649	1.6961	1.4519	1.5973	1.5159	0.4186
Calmar	1.6225	1.2268	1.2775	1.482	0.9764	0.4787

To investigate the performance and the risk profile of the strategies in different periods, we decompose the test period into three sub-periods, as shown in Figure 1 and Table 2. The first sub-period is from 1995.01.01 to 2002.12.31, during which the strategies significantly outperform the market. Astonishingly, the portfolios earn mean daily returns more than 1.5% (380% when annualized) and the net values reach around 10^{15}. The Sharpe ratios are even larger than 7 for the machine learning strategies. To the best of our knowledge, such performance is unprecedented and superior compared to the remarkable short-term strategy performance by Krauss et al. [15], who implement a momentum driven statistical arbitrage. Still, the strategies' maximum drawdowns are comparable with that of SP500, indicating that they do not bear a significant risk. The maximum drawdowns occur during the dot-com bubble crisis. One potential explanation for the tremendous performance is the absence of machine learning techniques until early 2000s in the US market.

The second sub-period ranges from 2003.01.01 to 2010.12.31. The strategies still perform superbly earning mean daily returns around 0.7% (annualized mean returns around 200%). It is however notable that the performances are seriously weakened compared to the former sub-period. We conjecture that the performance weakening is related to the attenuated reversal effect in the market. The strategy returns still exhibit a positive skewness, and the standard deviation and 5-percent VaR are similar to those in the first sub-period, which are about twice the level of the market. The maximum drawdown reaches 53.06%, 63.52%, 51.05%, 58.7%, and 55.02% for the XGBoost, GBDT, AdaBoost, RF, and the linear strategy, respectively. These values are comparable to the market's maximum drawdown which occurs during the global financial crisis. The Sharpe ratios of the strategies remain above 3, exhibiting outstanding return-to-risk performances.

The third sub-period ranges from 2011.01.01 to 2018.12.31. We observe further weakening of the strategies' performances in this post-crisis era. The portfolios show mean daily returns around 0.3% (annualized returns around 70%), which are substantially lower compared to former sub-periods. Nevertheless, the strategies still outperform the market. The skewness is positive, but the standard deviation and 5-percent VaR are larger than those of the benchmark. In particular, the maximum drawdown of the strategy is significantly higher than that of the market. A year-by-year performance analysis reveals that the strategies perform poorly in 2011, 2014, and 2015, with the annualized Sharpe ratios falling below 0.7 while the maximum drawdowns reaching 40%. The poor performance in 2011 can be associated with the European debt crisis, but there are no widely-known economic events that can be linked to the poor performances in 2014 and 2015. We conjecture that the performance deterioration in this sub-period is associated with the attenuated anomalous effects in recent years.

In all three sub-periods, the machine learning and linear reversal strategies yield remarkable performances. Among the machine learning methods, XGBoost consistently outperforms GBDT, AdaBoost and RF, while GBDT, AdaBoost and RF perform comparably with each other. It is notable that the linear method renders a similar performance to the AdaBoost and RF strategies, particularly in the third sub-period, the linear model even outperforms AdaBoost. The boosting algorithms with enhanced prediction power, such as GBDT and XGBoost, can still persistently outperform the linear model. The observation implies a diversified predictability of the machine learning framework, and by properly choosing the model system, more nonlinear information might be captured leading to improved classification efficiency.

3.1.2. Prediction Accuracy Analysis

In order to evaluate the prediction quality of the linear and nonlinear models, we conduct a class-level classification accuracy analysis. The classification accuracy is calculated as in Eq. (5),

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN} \tag{5}$$

where TP, TN, FP and FN respectively denote the true positive, true negative, false positive, and false negative. For the first class, true positive represents samples that are in the first class and also predicted to be in the first class; true negative represents those that are not in the first class and predicted to be not in the first class; false positive represents those that are not in the first class but predicted to be in the first class; false negative represents those that are in the first class but predicted to be in another class. By definition, the linear model divides stocks uniformly across classes. For the machine learning models, each class can contain a different number of samples, and therefore, we reclassify stocks based on the probability for the first class so that they are uniformly divided across classes. As the XGBoost performs best among the machine learning models, we compare it with the linear reversal model.

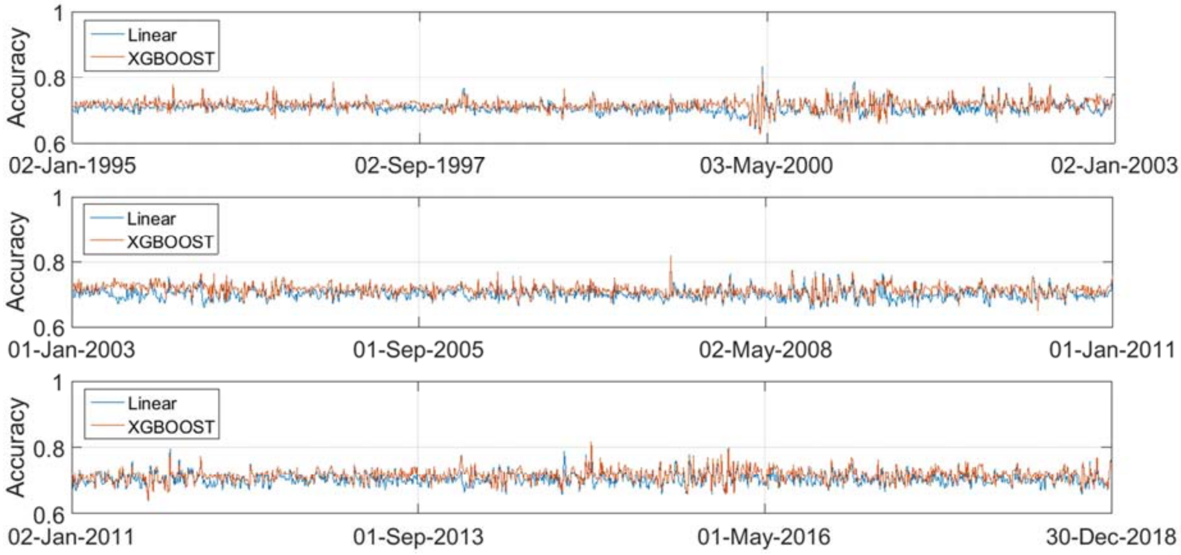

Figure 2. *Classification accuracy of the XGBoost and linear reversal models in the US market in three sub-periods.*

Figure 2 demonstrates the time variation of the classification accuracy for both models. As shown in the figure, the accuracy of the XGBoost is slightly above the linear model in all three sub-periods, which might correspond to the 2 thousandth higher daily mean return compared to the linear reversal strategy. The finding is consistent with the earlier analysis where nonlinear strategies yield a diversified predictability with enhanced performances, suggesting a superiority of the machine learning framework.

Table 3. *Holding period dependence of the daily return characteristics of the XGBoost strategy in the US market.*

	1995.01.01-2018.12.31				
	2 days	**5 days**	**10 days**	**20 days**	**SP500**
daily mean return	0.0101	0.0034	0.0032	0.0013	0.0003
t-statistics	22.2568	7.8314	7.8092	3.2008	2.2692
standard deviation	0.0351	0.0334	0.0321	0.0322	0.0117
skewness	1.4442	1.2280	1.1528	0.9199	-0.0855
kurtosis	13.3680	10.4733	11.5146	10.8843	11.4960
5-percent VaR	0.0387	0.0430	0.0442	0.0457	0.0182
maximum drawdown	0.5306	0.7859	0.6914	0.8370	0.5713
annual return	2.5359	0.8487	0.8126	0.3342	0.0858
Sharpe	4.4924	1.5431	1.5362	0.5951	0.3014
Sortino	5.6925	1.8115	1.7678	0.6468	0.2912
Calmar	4.7792	1.0800	1.1753	0.3993	0.1502

3.1.3. Effects of Holding Period

To test the effects of the holding period on portfolio performance, we re-implement the XGBoost strategy extending the portfolio rebalancing period to 5, 10 and 20 days. As shown in Table 3, the machine learning strategy with the elongated holding periods can still outperform the index benchmark. The performance, however, is significantly deteriorated when the holding period is extended to 5 and 10 days, as revealed in the diminished daily mean returns and Sharpe ratios. When the holding period is set to 20 days, the performance is further deteriorated, suggesting that the past one month return features are not sufficient to provide enough information for the one-month ahead prediction. The *t*-statistic is also reduced with the holding period, but remains above 3 even for the 20-day holding period, indicating that the return is significantly different from 0. The skewness is still positive, but the 5-percent VaR and maximum drawdown increase with the holding period. The Sharpe ratio decreases from 4.49 to 0.59 when the holding period increases from 2 to 20 days, implying a less satisfiable return-to-risk performance.

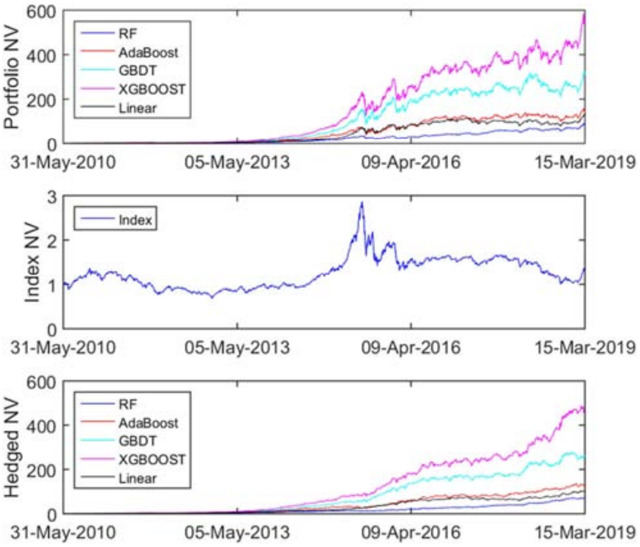

Figure 3. *Net asset values of the RF, AdaBoost, GBDT, XGBoost, and linear strategies in the Chinese market. The first row displays the net values of the strategies, the second row the net value of the benchmark, and the third row the net values of the strategies hedged by the index. A transaction cost of 0.16% is assumed.*

Table 4. *Daily return characteristics and annualized risk matrices of the RF, AdaBoost, GBDT, XGBoost, and linear strategies in the Chinese market. A transaction cost of 0.16% is assumed.*

	2010.06.01-2019.03.14					
	XGBoost	**GBDT**	**AdaBoost**	**RF**	**Linear**	**CSI 500**
daily mean return	0.0032	0.0030	0.0026	0.0023	0.0025	0.0003
t-statistics	6.2953	5.6829	4.9534	4.8004	5.1311	0.7464
standard deviation	0.0238	0.0242	0.0244	0.0225	0.0229	0.0171
skewness	-0.5888	-0.4489	-0.6216	-0.5813	-0.4521	-0.7951
kurtosis	5.6187	5.5033	6.0695	6.4127	7.1845	6.3593
5-percent VaR	0.0359	0.0389	0.0383	0.0358	0.0337	0.0278
maximum drawdown	0.4607	0.4518	0.4695	0.4755	0.4112	0.6520
annual return	0.8157	0.7515	0.6605	0.5893	0.6412	0.0696
Sharpe	2.0838	1.8749	1.6249	1.5656	1.6807	0.1460
Sortino	1.9023	1.7384	1.4773	1.4288	1.5690	0.1298
Calmar	1.7708	1.6634	1.4069	1.2393	1.5592	0.1068

3.2. Case of Chinese Market (2010.06.01-2019.03.14)

3.2.1. Machine Learning and Linear Reversal Strategy Performances

As displayed in Figure 3 and Table 4, both machine learning and linear reversal strategies exhibit remarkable outperformances compared to the benchmark in the Chinese market from 2010.06.01 to 2019.03.14. The portfolios earn mean daily returns around 0.3%, equivalent to annualized mean returns around 80%. The standard deviations of the returns are around 2%, slightly larger than that of the market, whereas the 5-percent VaR is comparable to that of the benchmark. As opposed to the case of the US market, however, the skewness is negative for all the strategies, which is not favored by investors [2]. Nevertheless, the strategies exhibit moderate maximum drawdowns that are smaller than that of the benchmark. The Sharpe, Sortino and Calmar ratios of the strategies are significant, demonstrating their considerable return-to-risk profiles. Note that in Chinese market, XGBoost outperforms GBDT, AdaBoost and RF, which is similar to the case in US market.

The performances in the US and the Chinese markets demonstrate the profit generating capacity of the tree-based machine learning algorithms both in developed and emerging markets. The strategies significantly outperform the benchmark and yield unprecedented return-to-risk ratios in the US market, although the performances are observed to decline in recent years. The diminishing performances can be attributed to the enhanced market efficiency and deteriorated anomalous effect. The strategies also perform remarkably in the Chinese market during the past 10 years, demonstrating the reversal strategies' potential in emerging markets.

Similar to the US market, the linear model again performs comparably with AdaBoost and RF, but underperforms GBDT and XGBoost, indicating that a considerable amount of nonlinear information contained in the short-term past returns can be uncovered by the boosting algorithms.

Furthermore, the return and risk matrices from our model system give rise to superior profiles compared to the previous machine learning assisted momentum/reversal strategies [17], i.e., a Sharpe ratio of around 2 is achieved by XGBoost compared to a Sharpe of 1.68 given by SVM in Li and Tam [17]. Even though the prominent return may come from a much shorter holding period or the contribution of micro-cap stocks within the wider selection range, we reserve our view that models predicting the classification in look-ahead periods, rather than the market state defined from past observation periods, can be more robust for the profit exploitation.

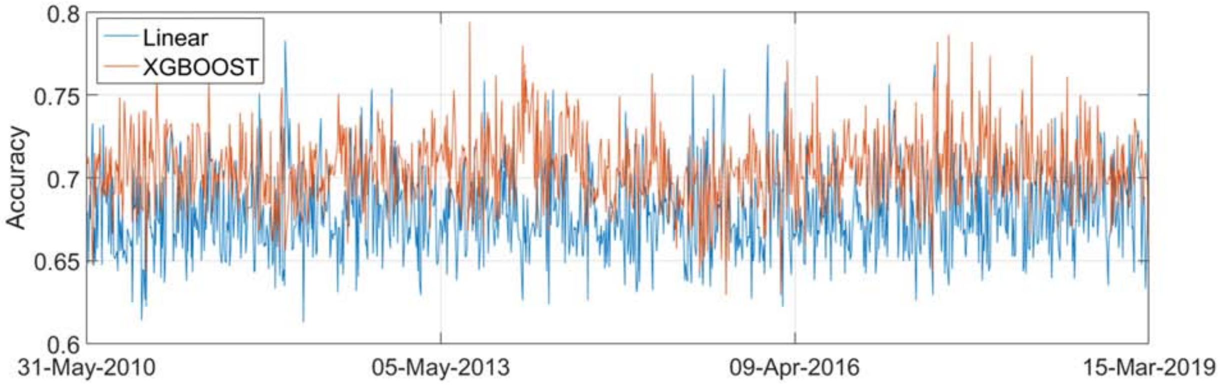

Figure 4. *Classification accuracy of the XGBoost and linear reversal models in the Chinese market.*

3.2.2. Prediction Accuracy Analysis

We conduct the class-level classification accuracy analysis described in 3.1.2 for the Chinese market and compare the prediction power of the XGBoost and linear reversal models. As shown in Figure 4, the first-class classification accuracies of the XGBoost and linear models are around 70%. The accuracy of the linear model lies persistently below XGBoost in the period of testing, which is in line with the previous findings.

Table 5. *Holding period dependence of the daily return characteristics of the XGBoost strategy in the Chinese market.*

	2010.06.01-2019.03.14				
	2 days	**5 days**	**10 days**	**20 days**	**CSI 500**
daily mean return	0.0032	0.0021	0.0012	0.0006	0.0003
t-statistics	6.2953	4.2008	2.3970	1.1406	0.7464
standard deviation	0.0238	0.0231	0.0232	0.0227	0.0171
skewness	-0.5888	-0.6341	-0.4696	-0.6410	-0.7951
kurtosis	5.6187	5.3653	5.0761	5.1811	6.3593
5-percent VaR	0.0359	0.0367	0.0401	0.0401	0.0278
maximum drawdown	0.4607	0.5539	0.5097	0.7558	0.6520
annual return	0.8157	0.5296	0.3032	0.1413	0.0696
Sharpe	2.0838	1.3618	0.7422	0.3087	0.1460
Sortino	1.9023	1.2262	0.6747	0.2724	0.1298
Calmar	1.7708	0.9561	0.5948	0.1870	0.1068

3.2.3. Effects of Holding Period

We re-implement the XGBoost strategy in the Chinese market extending the portfolio rebalancing period to 5, 10 and 20 days in order to examine the impact of the holding period on the profitability. As shown in Table 5, increasing the holding period sharply attenuates the daily mean returns. The *t*-statistic is also reduced with the holding period, implying a less statistically significant profitability. Similar to the case in the US market, the 5-percent VaR and the maximum drawdown increase when the holding period increases. The maximum drawdown and the Sharpe ratio are respectively 75.58% and 0.31 for the one-month rebalancing period, indicating a limited predictive power of the short-term past returns for a longer time horizon.

4. Conclusions

In this paper, we establish a new framework for machine learning-assisted reversal strategies by employing a variety of tree-based algorithms. We use short-term past returns up to a month to predict the future return class defined by the two-day ahead return, and develop a portfolio strategy that invests in stocks with a high probability for the high return class. The portfolios yield tremendous performances compared to the general market. They perform particularly well in the US market until early 2000s, showing extremely high Sharpe ratios, moderate kurtosis, and positive skewness. The superior performances in the US and Chinese markets imply that the machine learning based short-term reversal strategy can be successfully implemented in both developed and emerging markets. However, it should be noted that the performances deteriorate in recent years, which can be attributed to the attenuated anomalous effects and wider adoption of machine-learning techniques in finance. We also develop a novel linear reversal strategy based on multiple look-back periods, and investigate the discrepancy of the classification efficiency between the linear and nonlinear algorithms. It is found that the nonlinear framework is capable of providing a diversified predictability, and by properly choosing the model system, an improved classification accuracy can be readily achieved leading to enhanced strategy performances.

Appendix

Table 6. *Hyperparameters employed by the RF, AdaBoost, GBDT, XGBoost models.*

	US				CHINA			
	XGBoost	**GBDT**	**AdaBoost**	**RF**	**XGBoost**	**GBDT**	**AdaBoost**	**RF**
n_estimators	100	100	70	100	70	80	70	100
max_depth	6	3	3	3	9	6	3	3
learning_rate	0.1	0.1	1		0.1	0.1	1	

Table 7. *The dependence of strategy daily mean returns on the number of trees in the RF model. Note that the hyper-parametrization is performed for both US and Chinese markets.*

num tree	**US**						**China**					
	20	**40**	**60**	**80**	**100**	**120**	**20**	**40**	**60**	**80**	**100**	**120**
daily mean return	0.009	0.0089	0.0091	0.0091	0.009	0.0089	0.0017	0.0019	0.002	0.0021	0.0023	0.0022

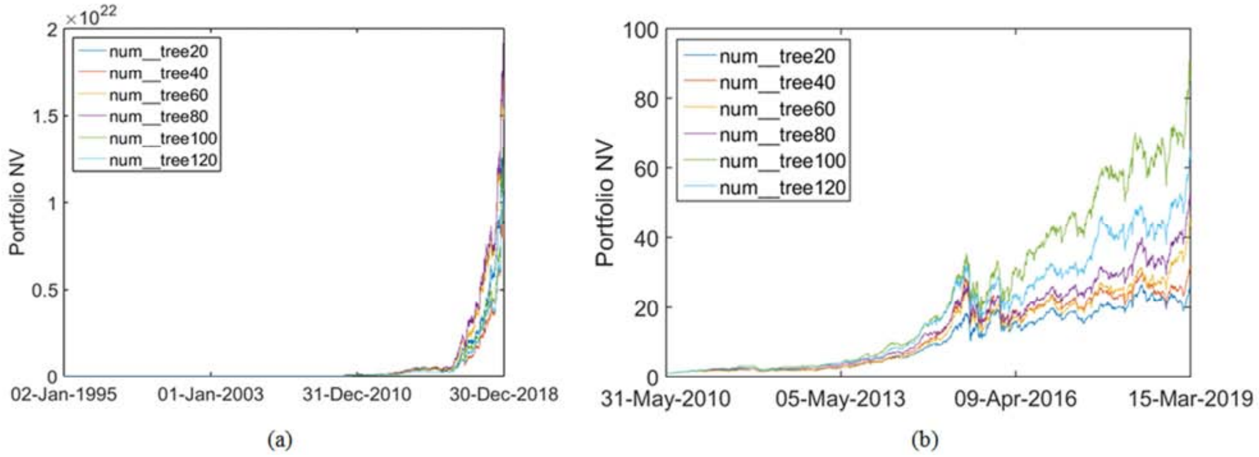

Figure 5. *Portfolio profiles for the RF model with different number of trees in both US (a) and Chinese (b) markets.*

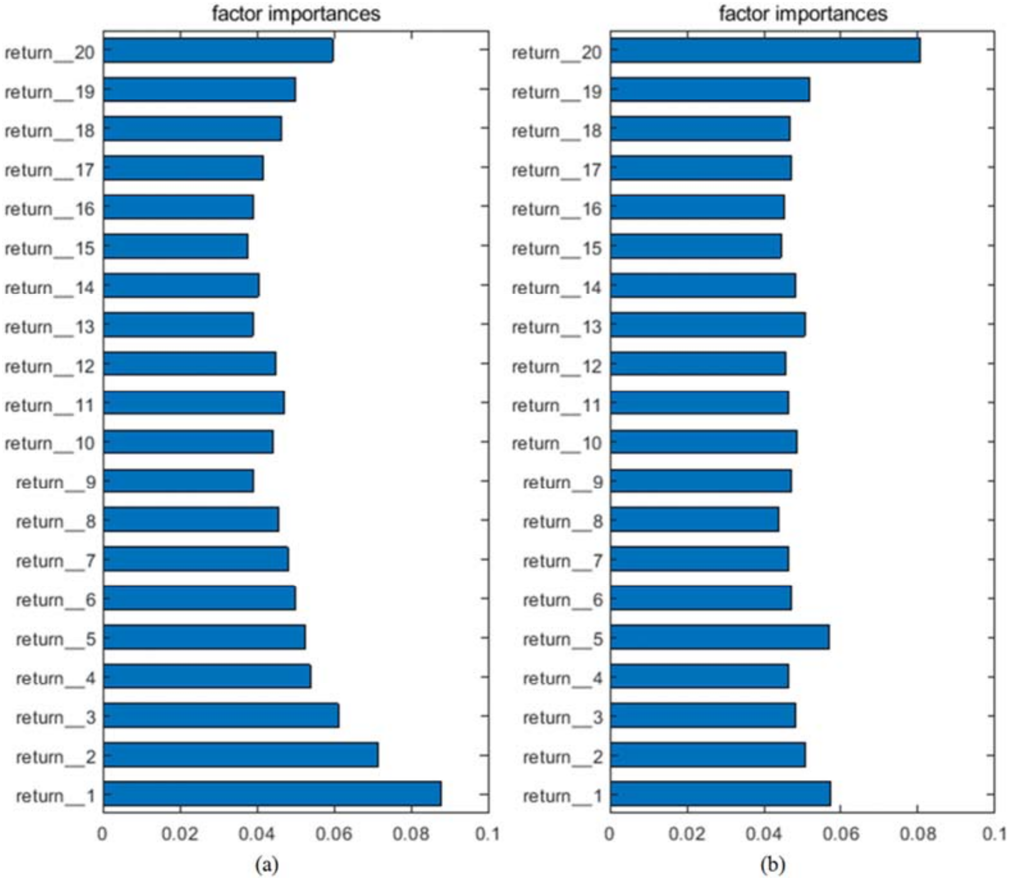

Figure 6. *Factor weight distribution in the XGBoost model for both US (a) and Chinese (b) markets.*

Note that Figure 6 shows the factor weight distribution in the XGBoost model, which is calculated from the relative influence of each variable in trees growing. The presented weight distributions are averages of distributions from each model trained in every training period. It is interesting that in both US and Chinese markets, past returns with 1 day and 20 days lags are given higher importance in determining the stock forward process. Besides, noteworthy explanatory power of return_5 is observed in the Chinese market, probably due to a weekly based reversal investment behavior. Compared to the linear strategy where the factor weight is mostly empirically defined, the machine learning offers a more appropriate framework where a more relevant feature distribution can be captured, so that a higher classification accuracy is achievable leading to enhanced performances.

References

[1] Andriyashin, A., Härdle, W. K., & Timofeev, R. V. (2008). Recursive portfolio selection with decision trees. *SSRN Electronic Journal*.

[2] Barroso, P., & Santa-Clara, P. (2014). Momentum has its moments. *Journal of Financial Economics*, 116 (1), 111–20.

[3] Breiman, L. (2001). Random forests. *Machine Learning*, 45 (1), 5–32.

[4] Cheng, S., Allaudeen, H., Avanidhar, S., & Sheridan T. (2017). Short-Term Reversals: The Effects of Past Returns and Institutional Exits. *Journal of Financial and Quantitative Analysis*, 52 (1), 143–73.

[5] Chen, T., & Guestrin, C. (2016). XGBoost: a scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on knowledge discovery and data mining*, 785-794.

[6] Conrad, J., Mustafa N. G., & Gautam K. (1997). Profitability of Short-Term Contrarian Strategies: Implications for Market Efficiency. *Journal of Business and Economic Statistics*, 15 (3), 379–86.

[7] Cooper, M. (1999). Filter Rules Based on Price and Volume in Individual Security Overreaction. *Review of Financial Studies*, 12 (4), 901–35.

[8] Fama, E. F., & French, K. R. (1996). Multifactor Explanations of Asset Pricing Anomalies. *Journal of Finance*, 51 (1), 55–84.

[9] Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38 (4), 367–378.

[10] Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55 (1), 119–139.

[11] Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). The elements of statistical learning: Data mining, inference, and prediction. *Springer*, New York.

[12] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. *Spriner*, New York.

[13] Jegadeesh, N. (1990). Evidence of Predictability of Security Returns. *Journal of Finance*, 45 (3), 881–98.

[14] Jegadeesh, N., & Sheridan, T. (1995). Overreaction, Delayed Reaction, and Contrarian Profits. *Review of Financial Studies*, 8 (4), 973–93.

[15] Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 259 (2), 689–702.

[16] Lehmann B. N. (1990). Fads, Martingales, and Market Efficiency. *The Quarterly Journal of Economics*, 105 (1), 1–28.

[17] Li, Z., & Tam, V. (2018). A Machine Learning View on Momentum and Reversal Trading. *Algorithms*, 11 (11), 170.

[18] Nguyen, T. T., Huang, J. Z., & Nguyen, T. T. (2015). Unbiased Feature Selection in Learning Random Forests for High-Dimensional Data. *The Scientific World Journal*, 2015, 471371-18.

[19] Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, 5 (2), 197–227.

[20] Sorensen, E. H., Miller, K. L., & Ooi, C. K. (2000). The decision tree approach to stock selection. *Journal of Portfolio Management*, 27 (1), 42–52.

[21] Tan, Z., Yan, Z., & Zhu, G. (2019). Stock selection with random forest: An exploitation of excess return in the Chinese stock market. *Heliyon*, 5 (8).

[22] Groot, W. D., Huij, J., & Zhou, W. (2012). Another Look at Trading Costs and Short-Term Reversal Profits. *Journal of Banking and Finance*, 36 (2), 371–82.

[23] Zhu, M., Philpotts, D., & Stevenson, M. J. (2012). The benefits of tree-based models for stock selection. *Journal of Asset Management*, 13 (6), 437–448.