Quantitative Finance and Economics

http://www.aimspress.com/journal/QFE

QFE, 7(4): 569–594.
DOI: 10.3934/QFE.2023028
Received: 31 July 2023
Revised: 31 October 2023
Accepted: 06 November 2023
Published: 13 November 2023

Research article

Machine learning-based quantitative trading strategies across different time intervals in the American market

Yimeng Wang[1] **and Keyue Yan**[2,*]

[1] Department of Mathematics, University of Macau, Macau, China
[2] Choi Kai Yau College, University of Macau, Macau, China

* **Correspondence:** Email: conneryan@foxmail.com.

Abstract: Stocks are the most common financial investment products and attract many investors around the world. However, stock price volatility is usually uncontrollable and unpredictable for the individual investor. This research aims to apply different machine learning models to capture the stock price trends from the perspective of individual investors. We consider six traditional machine learning models for prediction: decision tree, support vector machine, bootstrap aggregating, random forest, adaptive boosting, and categorical boosting. Moreover, we propose a framework that uses regression models to obtain predicted values of different moving average changes and converts them into classification problems to generate final predictive results. With this method, we achieve the best average accuracy of 0.9031 from the 20-day change of moving average based on the support vector machine model. Furthermore, we conduct simulation trading experiments to evaluate the performance of this predictive framework and obtain the highest average annualized rate of return of 29.57%.

Keywords: stock price prediction; machine learning; quantitative trading; moving average

JEL Codes: C32, C63, E37

1. Introduction

As the most popular traditional financial market products, stocks have always attracted millions of people to invest. Investors and speculators can earn profits from the changes of stock prices in the stock market, and they always follow the volatility from various sources. The S&P 500 Index, as one of the most important and traditional reference indexes, can represent the general trend in the American stock market (Liu et al., 2016). According to the historical data of the S&P 500 Index, there has been an obvious increasing trend over the past 20 years. However, the S&P 500 Index suffered a significant drop when faced with some major events like the financial crisis in 2008. Since then, the American stock market has steadily recovered and reached new highs until the end of 2021. For a regular investor, the most important thing is to earn profits from the financial market by following the trend. As science and technology develop, more and more researchers and professional investment institutions apply advanced technologies like machine learning techniques to make price predictions on the stock market and help investors to earn profits from the trend of stocks (Obthong et al., 2020).

The objective of this research is to forecast the stock price changes from the perspectives of individual investors, based on the public transaction data only, without considering the information behind the stock. This is because the public data can be easily obtained from the website and used for stock price forecasting. We construct some variables related to technical indicators in the financial market from the raw data and train machine learning regression models with them. We employ six popular machine learning models for stock price predictions: decision tree, support vector machine, bootstrap aggregating, random forest, adaptive boosting, and categorical boosting. By building these machine learning prediction models, we obtain the change of moving average predicted value and transform it into a binary classification problem to obtain the corresponding classification results. Based on all the results, we conduct simulation trading experiments to evaluate the performance of the machine learning prediction models on the stock transaction.

The rest of this research is organized as follows: Section 2 reviews some related research papers to show the feasibility and novelty of the research idea. Section 3 describes the methods in detail and explains the core steps in the modelling process. Section 4 presents the regression and classification results for the prediction models provided by the method in this research. Process and analysis are given as well. Section 5 conducts simulation trading experiments to verify whether the method provided by this research can help investors to earn profits. Section 6 concludes this research and suggests several possible directions for future research.

2. Related work

Many professional works seek various methods to predict the changes in stock prices and earn more benefits from the stock market. As the research by Obthong et al. (2020) reviewed, machine learning techniques became powerful tools for predicting stock price changes and providing trading decision information for investors in the stock market with the development of computer science technology. Since the stock price changes are time series data, one of the most common prediction methods is using regression models to forecast the stock prices and capture the trend of stock price changes. Henrique et al. (2018) applied support vector machine regression model to predict the daily and minute stock prices and the prediction errors were smaller for daily data. Vijh et al. (2020) used neural network and random forest to predict future stock close prices directly with some constructed variables such as 7-day moving average and the best error result of 0.42 was obtained in a neural network model for Pfizer Inc. stock. Wang and Yan (2022) tried to employ some famous times series deep learning models such as long short term memory and gate recurrent unit to predict the price of Bitcoin and captured the significant changes successfully.

Instead of predicting the stock price, some studies focus on predicting the trend of the stock market and use machine learning classification models for this task. A related paper applied the random forest classification model to predict the stock close direction and achieved high accuracy, which was more than 0.8 (Khaidem et al., 2016). Ampomah et al. explored the performance of some ensemble learning models such as bootstrap aggregating, random forest, extra trees, and adaptive boosting for predicting the one-day stock price movement by classification. They found that these ensemble algorithms performed well in general and the adaboost of bagging model had the best accuracy among them (Ampomah et al., 2021). Khan et al. (2022) also predicted stock price trends based on some machine learning classification models, such as support vector machine, random forest, and so on. The random forest model obtained the highest accuracy of 83.22% (Khan et al., 2022). Another study compared the performance of ensemble classifiers with single predictors and neural network for stock price prediction. The results showed that ensemble learning models can achieve better results than the neural network (Subasi et al., 2021).

To improve accuracy and reduce error, most research papers on price prediction incorporated more information such as fundamental and technical analysis to build models (Obthong et al., 2020). Many papers on the stock market prediction by machine learning algorithms used technical indicators to build models (Nti et al., 2020). Some researchers constructed some technical indicators in the financial market, such as relative strength index, moving average convergence divergence and so on (Basak et al., 2019). Other researchers also used some technical indicators of the financial market as input variables to assist stock price predictions by machine learning classification models and found that the support vector machine model achieved the best training accuracy of nearly 0.7 (Zhang et al. 2018). Ampomah et al. (2021) considered forty financial technical indicators to make the model generalizable. Yan and Wang's (2023) research proposed a method to predict the stock price difference by using some financial indicators, such as moving average and momentum. They claimed that their method can reduce the error along with ensemble regressors.

The main purpose of the stock prediction is to help investors earn more money from stock price changes, so the trading strategies based on these prediction models can better demonstrate the performance of models in the field of quantitative trading. Kamalov (2020) predicted significant stock price changes by neural network and provided a trading simulation with positive rates of return based on the best prediction model on four stocks. Dinesh et al. (2021) predicted the stock market trends by forecasting the short and long term moving average lines based on linear regression and classified the trading signals from the crossover of two moving average lines. Wang and Yan (2023) integrated the trading strategy with the predictive machine learning models and the best performance of their model achieved 81% accuracy on the Bitcoin price. The random forest was used to predict the change of the 3-day weighted moving average for financial banking stocks and applied to the bollinger band strategy to obtain higher returns than the traditional strategy (Yan et al., 2023).

In this research, we propose a method that combines some traditional machine learning regression models with technical indicators in the financial market to predict the moving average changes of different days on stocks from the American stock market. Our goal is to convert the regression results into a binary classification problem and find the best-performing model for stock price prediction. We also conduct simulation experiments of trading based on the model's predictions to test the validity of this method.

3. Methodology

We aim to use six machine learning models to predict the changes of moving averages for eight stocks in the American market. The framework is shown in Figure 1. First, we collect the public transaction data of stocks and construct indicator variables based on technical indicators in the financial market. We also define the target predicted variable as the changes of moving averages with different time intervals. Then, we build regression models to predict the target predicted variable and use the prediction results to calculate the close prices for stocks. Moreover, we convert the regression problem into a binary classification problem and obtain the classification results. Finally, we design simulation experiments based on the models proposed in the last part and compare the performance of different machine learning trading strategies that can help investors gain more benefits in reality.

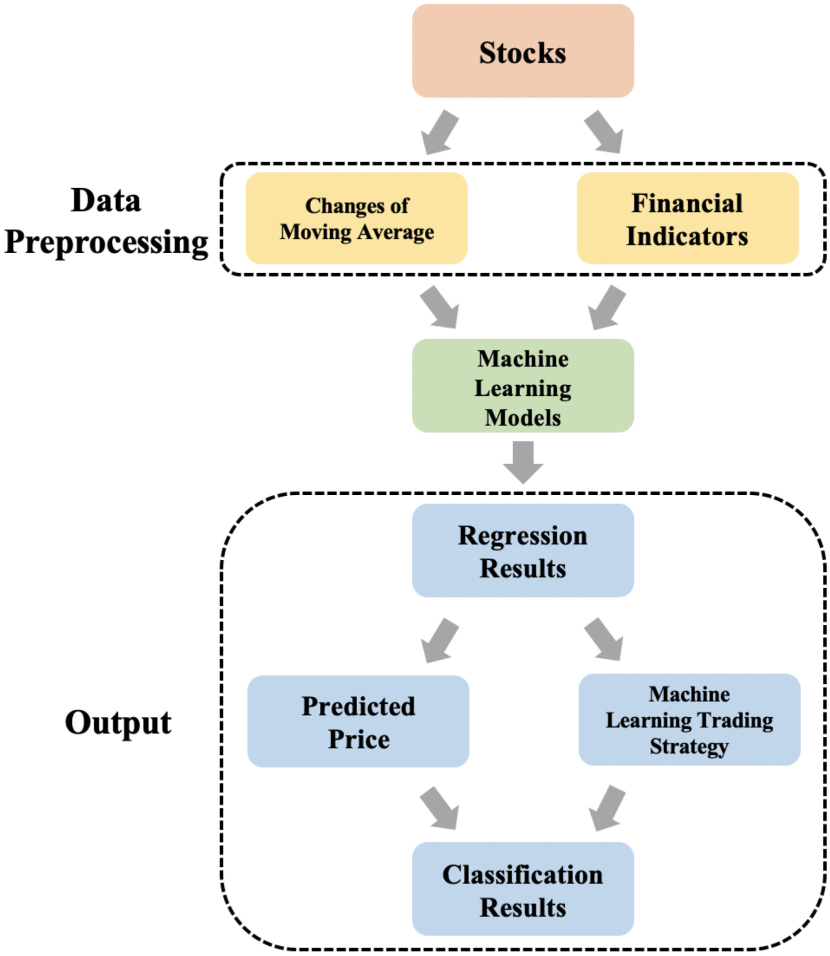


Figure 1. Framework.

3.1. Data collection and variables construction

We select eight stocks from the technology industry listed on NASDAQ and NYSE and collect 5-year data from the Yahoo Finance website, covering the time period from 2017.01.01 to 2021.12.31. Detailed materials of these stocks are listed in Table 1. The original data of these stocks include five independent variables: open, high, low, close and volume. To improve the prediction performance, we also establish some related variables based on financial technical indicators which have been widely used in previous research works.

Table 1. Information of stocks.

Stock	Company	Exchange
AAPL	Apple Inc.	NASDAQ
ADBE	Adobe Inc.	NASDAQ
AMD	Advanced Micro Devices, Inc.	NASDAQ
CRM	Salesforce, Inc.	NYSE
MSFT	Microsoft Corporation	NASDAQ
NOW	ServiceNow, Inc.	NYSE
NVDA	NVIDIA Corporation	NASDAQ
ORCL	Oracle Corporation	NYSE

3.1.1. Moving average (MA)

$$MA(d)_t = \frac{Close_t + Close_{t-1} + \ldots + Close_{t-(d-1)}}{d}, t \geq d \geq 1. \tag{1}$$

MA calculates the average of the close prices over the past d days at day t, which represents the tendency of stock price in the market.

3.1.2. Exponential moving average (EMA)

$$EMA(d)_t = \frac{2}{d+1} \sum_{i=0}^{t-1} (\frac{d-1}{d+1})^i Close_{t-i}, t \geq d \geq 1. \tag{2}$$

Similar to MA, EMA is a technical indicator that can capture the trend of the stock price changes at day t. We also include the EMA of the close price for each stock in this research. The term $(\frac{d-1}{d+1})^i$ is the weight of the corresponding $Close_{t-i}$, which increases as i gets closer to 0. (Zhang et al., 2018).

3.1.3. Relative strength index (RSI)

$$RSI(d)_t = 100 - \frac{100}{1 + \frac{Average\ Gain\ Over\ past\ d\ days}{Average\ Loss\ Over\ past\ d\ days}}, t > 0. \tag{3}$$

RSI is a momentum indicator that measures the degree of overbought and oversold conditions based on the average gain and loss over the past days. When the value of RSI is higher than 70, it indicates an overbought condition and the holder should sell the stock. Conversely, when the value of RSI is lower than 30, it indicates an oversold condition and the investor should buy the stock (Basak et al., 2019).

3.1.4. On balance volume (OBV)

$$OBV_t = \begin{cases} OBV_{t-1} + Volume_t, & \text{if } Close_t > Close_{t-1} \\ OBV_{t-1} - Volume_t, & \text{if } Close_t < Close_{t-1} \\ OBV_{t-1}, & \text{if } Close_t = Close_{t-1} \end{cases}, t > 1. \tag{4}$$

OBV is another momentum indicator that reflects the trend of stock price changes based on the volume flows (Ampomah et al., 2021; Nti et al., 2020). When the OBV line is rising, it suggests that the stock price will increase. When the OBV line is falling, it suggests that the stock price will decrease (Nti et al., 2020).

3.1.5. Moving average convergence divergence (MACD)

$$MACD_t = EMA(12)_t - EMA(26)_t, \tag{5}$$

$$SingalLine_t = EMA(MACD_t, 9)_t, t > 0. \tag{6}$$

MACD is a momentum indicator derived from MA and captures the difference between the short-term and long-term EMA at day t (i.e., EMA with $d = 12$ represents the short-term trend, and EMA with $d = 26$ represents the long-term trend). When $MACD_t$ is lower than $SingalLine_t$, it is a signal of a sell opportunity for investors. When $MACD_t$ is higher than $SingalLine_t$, it is a signal of a buying opportunity for investors (Basak et al., 2019).

3.1.6. Rate of change (ROC)

$$ROC(d)_t = \frac{Close_t - Close_{t-d}}{Close_{t-d}}, t > d > 0. \tag{7}$$

ROC is the ratio of the price change to the price d days ago at day t. Similarly, ROC is also a momentum indicator that forecasts the change in the stock close price.

3.1.7. Stochastic oscillator

$$\%K(d)_t = 100 \times \frac{Close_{t-i} - LowestLow(d)_t}{HighestHigh(d)_t - LowestLow(d)_t}, \tag{8}$$

$$\%D(d)_t = 100 \times \frac{\sum_{i=0}^{3}(Close_{t-i} - LowestLow(d)_{t-i})}{\sum_{i=0}^{3}(HighestHigh(d)_{t-i} - LowestLow(d)_{t-i})}, t > d > 0. \tag{9}$$

$\%K(d)_t$ and $\%D(d)_t$ are technical indicators that provide the oversold and overbought conditions of the stock price (Nti et al., 2020). To compute these two indicators, $LowestLow(d)_t$ denotes the lowest Low price over past d days at day t, and $HighestHigh(d)_t$ denotes the highest High price over past d days at day t. Based on these two indicators, the stock price is increasing if $\%K(d)_t$ is higher than $\%D(d)_t$, and it has falling trend if $\%K(d)_t$ is lower than the $\%D(d)_t$ (Lai et al., 2019).

3.1.8. Commodity channel index (CCI)

$$CCI(d)_t = \frac{\frac{High_t + Low_t + Close_t}{3} - MA(d)_t}{0.015 \times \frac{\sum_{i=0}^{d} \left| \frac{High_{t-i} + Low_{t-i} + Close_{t-i}}{3} - MA(d)_{t-i} \right|}{d}}, t > d > 0. \tag{10}$$

CCI is an indicator that detects whether the stock price deviates from its normal distribution (Zhang et al., 2018). The CCI index is calculated based on the high, low, close, and d-day moving average of the close price, with 0.015 as the constant factor. The value of the CCI index ranges from negative infinity to positive infinity.

Table 2. Variables construction.

Name of variable	Formula	Name of variable	Formula
Open	$Open_t, t > 0$	MA20_1	$MA(20)_{t-1}, t > 20$
High	$High_t, t > 0$	MA20_increasement	$MA(20)_t - MA(20)_{t-1}, t > 20$
Low	$Low_t, t > 0$	EMA5	$EMA(5)_t, t \geq 5$
Close	$Close_t, t > 0$	EMA10	$EMA(10)_t, t \geq 10$
Close_1	$Close_{t-1}, t > 1$	EMA20	$EMA(20)_t, t \geq 20$
Close_increasement	$Close_t - Close_{t-1}, t > 1$	RSI	$RSI(12)_t, t > 0$
Volume	$Volume_t, t > 0$	OBV	$OBV_t, t \geq 2$
Volume_1	$Volume_{t-1}, t > 1$	ROC5	$ROC(5)_t, t > 5$
Volume_increasement	$Volume_t - Volume_{t-1}, t > 1$	ROC10	$ROC(10)_t, t > 10$
MA5	$MA(5)_t, t \geq 5$	ROC20	$ROC(20)_t, t > 20$
MA5_1	$MA(5)_{t-1}, t > 5$	MACD	$MACD_t, t > 0$
MA5_increasement	$MA(5)_t - MA(5)_{t-1}, t > 5$	MACDsignal	$SingalLine_t, t > 0$
MA10	$MA(10)_t, t \geq 10$	MACDhist	$MACD_t - SingalLine_t, t > 0$
MA10_1	$MA(10)_{t-1}, t > 10$	slowk	$\%K(3)_t, t > 0$
MA10_increasement	$MA(10)_t - MA(10)_{t-1}, t > 10$	slowd	$\%D(3)_t, t > 0$
MA20	$MA(20)_t, t \geq 20$	CCI	$CCI(10)_t, t > 0$

We use the technical indicators in the financial market and the variables shown in Table 2 for this research. To avoid the impact of variables with large values, we standardize the data using the formula $\frac{data-\mu}{\sigma}$, where μ is the mean and σ is the standard deviation of each constructed variables in the data. Three types of moving average, like 5-day, 10-day and 20-day, are used to represent the trend of stock price changes. We set the $Value(d)_t$ as the predicted value with d equal to 5, 10 and 20. And then we find out which type is the best predictor of stock price changes.

$$Value(d)_t = MA(d)_{t+1} - MA(d)_t = \frac{Close_{t+1} - Close_{t-(d-1)}}{d}. \tag{11}$$

The formula below is used to calculate the predicted stock price at day $t + 1$ for each stock based on the regression models.

$$Predicted\ Close_{t+1} = Predicted\ Value(d)_t \times d + Close_{t-(d-1)}. \tag{12}$$

In this research, we have 32 independent variables for building the prediction model and dependent variable $Value(d)_t$ with d equal to 5, 10 and 20 as the prediction result. We split the whole data of each stock into a training set and a test set with a ratio of 8:2, which are 979 and 245, respectively.

3.2. *Machine learning model*

We employ six popular machine learning models in the research design: decision tree, support vector machine, bootstrap aggregating, random forest, adaptive boosting, and categorical boosting. We use the Scikit-Learn and CatBoost packages in Python to build these models. We also use the GridSearchCV module to find the best parameters (like kernel, loss function, etc.) for improving the performance of the models. We provide some basic and key descriptions of each machine learning model in this section.

3.2.1. Decision tree

Decision tree is a traditional machine learning algorithm with a tree-like structure. It is a simple and interpretable prediction model (Hindrayani et al., 2020). The decision tree model has one root node (input value) and many leaf nodes (outputs). Unlike the decision tree classification model, the outputs on the leaf nodes are values instead of categories. The final predicted value is the average of all output values in the decision tree. We use the Scikit-Learn package from Python to build the prediction model. Scikit-Learn builds the model by the classification and regression tree (CART) (Géron, 2022).

For the dataset $D = \{X_i, Y_i\}_{i=1,...,n}$, there are regions $\{R_m\}_{m=1,2}$ with a corresponding average output value $\{C_m\}_{m=1,2}$ and $X = \{X_i\}_{i=1,...,n}$. For the regression tree model, it finds a feature split s that minimizes the squared errors of the output value C_m with the corresponding Y_i and produces the regression tree $T_1(X)$:

$$f_1(X) = T_1(X) = \begin{cases} C_1, & \text{if } X < s, \\ C_2, & \text{if } X \geq s. \end{cases} \tag{13}$$

Based on residuals of the $f_1(X_i)$ with Y_i, it obtains $T_2(X)$ by the same step and updates the tree model:

$$f_2(X) = f_2(X) + T_2(X). \tag{14}$$

After that, it repeats the steps until the stop conditions (normally the squared errors are below a certain threshold) are met at the corresponding tree $f_j(X)$ and the $f(X) = f_j(X)$ is the final decision tree model.

3.2.2. Support vector machine (SVM)

SVM is a single predictor that can solve both classification and regression problems, like the decision tree model. For a dataset $D = \{X_i, Y_i\}_{i=1,...,n}$, the goal of the SVM prediction model is to find an objective function that makes the predicted value $f(x)$ close to the real value Y_i. The objective function is:

$$f(X) = w^T X + b. \tag{15}$$

$X_i \in X$ and w, b are parameters. There is a margin bounded by hyperplanes on both sides of the objective function. The SVM regression model tries to fit as many samples as possible between these two hyperplanes and minimizes $||w||$ to increase the margin (Géron, 2022). The problem of the SVM regression model becomes an optimization problem:

$$\min_{w,b} \frac{1}{2}||w||^2 + C \sum_{i=1}^{m} l_\epsilon(Y_i - f(X_i)). \tag{16}$$

C is a fixed constant and l_ϵ is an ϵ-insensitive loss function. We use Lagrange multipliers to get the partial derivatives of parameters w and b (Collobert and Bengio, 2001).

3.2.3. Bootstrap aggregating (Bagging)

Bagging is a type of ensemble learning model that combines many single predictors. For the bagging regression model, the predicted value is the average of all output results from each predictor (Breiman, 1996). The decision tree model is usually the original predictor of the bagging. For a bagging model with predictors, the formula of the predicted value $f(x)$ is:

$$f(X) = \frac{1}{M}\sum_{m=1}^{M} f_m(x). \tag{17}$$

$\{f_m\}_{m=1,\ldots,M}$ is the predicted value of each predictor and x is the corresponding random sample set, $x \in X$. The bagging model randomly takes a sample from the training dataset to train different predictors and allows a sample to be sampled by the same predictor more than once (Géron, 2022). Figure 2 shows the main process of the bagging model.

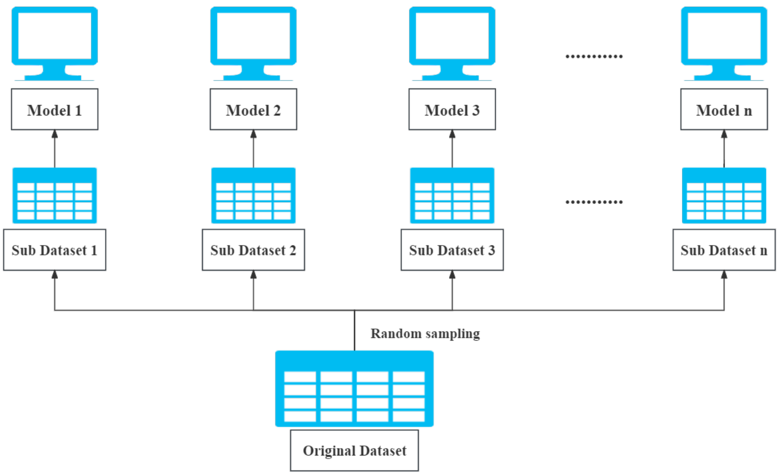


Figure 2. Structure of bagging.

3.2.4. Random forest

Random forest is an ensemble model of the decision tree models. It is an improvement based on the bagging model (Géron, 2022). Unlike the bagging model with the decision tree predictors, the random forest model selects both the sample dataset and the features to split randomly (Breiman, 2001). The decision tree predictors in a random forest regression model are independent of each other. The final predicted value is the average of the prediction results from each predictor. The randomness of sampling and feature selection makes this model less likely to overfit (Breiman, 2001). In practice, random forest has high stability and good predictive ability, which makes it perform well in other financial research fields, such as option pricing (Li and Yan, 2023).

3.2.5. Adaptive boosting (Adaboost)

Adaboost is a popular ensemble learning model and one of the boosting methods that combines several weak predictors into an effective prediction model (Géron, 2022). The steps of adaboost are shown in Figure 3.

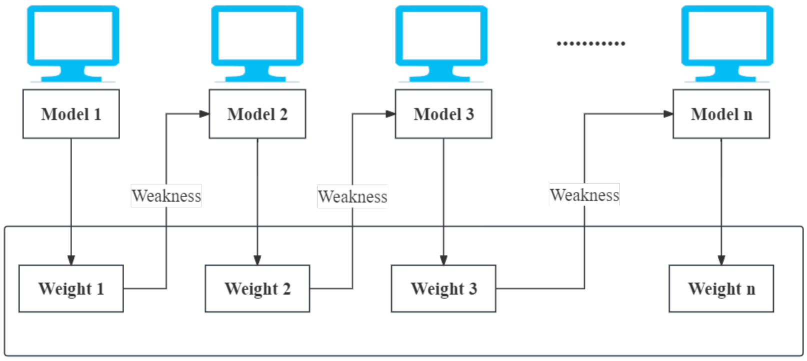


Figure 3. Structure of adaboost.

For the adaboost model, there is a high correlation among each predictor (Géron, 2022). The errors from the previous predictor are used to update the weight of samples for the next predictor and then this step is repeated (Freund and Schapire, 1997). In the dataset $D = \{X_i, Y_i\}_{i=1,\ldots,n}$, $w_{1,i} = \frac{1}{n}$ $(i = 1, \ldots, n)$ is the initial weight for the training set. For the k-th weak predictor, it computes the relative error (i.e, linear, squared or exponential error) $e_{k,i}$ for each sample and obtains the error rate for the adaboost regression model as $e_k = \sum_{i=1}^{n} w_{k,i} e_{k,i}$, where w_{ki} denotes the weight of samples in the k-th iteration. Based on the weight of this weak predictor $\alpha_k = \frac{e_k}{1-e_k}$, this model updates the weight of samples for the $(k+1)$-th predictor, which is:

$$w_{k+1,i} = \frac{w_{k,i}}{\sum_{i=1}^{n} w_{k,i} \alpha_k^{1-e_{k,i}}} \alpha_k^{1-e_{k,i}}. \tag{18}$$

This algorithm repeats the steps and combines all weak predictors with their respective weights. Finally, it obtains a strong predictor and makes the prediction.

3.2.6. Categorical boosting (Catboost)

Catboost is an open-source library based on decision trees that combines the gradient boosting algorithm with categorical features in a machine learning technique (Prokhorenkova et al., 2018). Catboost uses oblivious decision trees and splits on the same feature for each level of trees. The catboost algorithm does not use a simple way of applying the average label value to represent the categorical feature. It adds a prior value to improve the greedy target-based statistics to calculate the frequency of a certain category and reduce the effects of low-frequency categories. Catboost can also combine different features into new features and transform them to numerical values for a new split in a tree (Prokhorenkova et al., 2018). Compared with extreme gradient boosting and light gradient boosting machine, the catboost algorithm can handle categorical features and use an ordered boosting to solve the prediction shift problem. The main steps of catboost for solving the prediction shift problem are: (1) there is no feature combination in the first split; (2) for the later splits, they combine all features with features of previous levels. The catboost model can also deal with small data sets.

3.3. Evaluation indicator

In order to evaluate the performance of machine learning models from different perspectives, we use three well-known evaluation indicators of regression: mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). All indicators estimate the error between the real value y_t and predicted value $\hat{y}_t$ for the regression prediction models and generally. The lower the value of MAE, MSE, and RMSE, the better the model fitting.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|, \tag{19}$$

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2, \tag{20}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}. \tag{21}$$

In these formulas, y_t can be $Real\ Value(d)_t$, which represents the real change of d-day moving average on day t. $\hat{y}_t$ can be $Predicted\ Value(d)_t$, which represents the predicted change of d-day moving average on day t. According to the framework, we aim to obtain both the regression results from these models and the transformed classification results. Based on the $Predicted\ Value(d)_t$ value, we assign two labels to indicate the trend of stock price changes.

$$ValueBinary(d)_t = \begin{cases} 1, & \text{if } Value(d)_t > 0, \\ -1, & \text{if } Value(d)_t \leq 0. \end{cases} \tag{22}$$

When the predicted value is positive, we give the $ValueBinary(d)_t$ a label of 1; otherwise, we consider the stock price decreasing and give a label of −1 to $ValueBinary(d)_t$. Then, the research based on machine learning regression models becomes a binary classification problem. For a classification problem, it is usually evaluated by a confusion matrix.

Table 3. Confusion matrix.

Confusion matrix		*Predicted ValueBinary*$(d)_t$	
		1	−1
Real ValueBinary$(d)_t$	1	True Positive (TP)	False Negative (FN)
	−1	False Positive (FP)	True Negative (TN)

For the binary values $Real\ Value(d)_t$ and $Predicted\ Value(d)_t$, we use Accuracy, Precision, Recall, and F1-score to measure the performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{23}$$

$$Precision = \frac{TP}{TP + FP}, \tag{24}$$

$$Recall = \frac{TP}{TP + FN}, \tag{25}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{26}$$

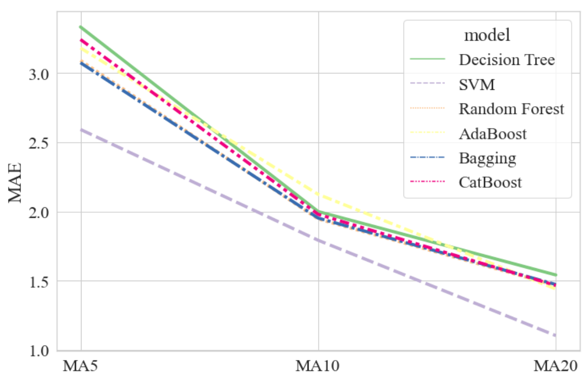
In general, accuracy measures the proportion of samples where the predicted results match the actual results among all samples. Precision evaluates the proportion of true positive samples among all samples with the predicted value $ValueBinary(d)_t$ of 1. Recall indicates the proportion of true positive samples among all samples that have the same value as the actual result. F1-score reflects the robustness of each model. For a model with better prediction performance, the values of all these classification evaluation indicators are expected to be as high as possible.

4. Result and discussion

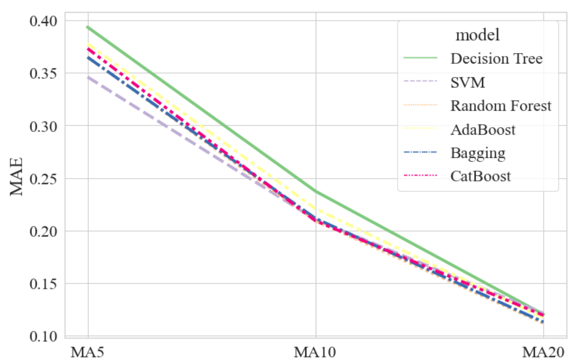
As the methodology designs, this research predicts the changes of moving average by machine learning regression models and transforms the predicted value into a binary classification problem to compare the performance of each prediction model. In this section, we first present the prediction results of these six regression machine learning models based on the eight stocks in the American stock market and then provide the transformed classification results compared with the results from corresponding classification models directly.

4.1. Regression result

The prediction regression results of each stock are shown below, and Figures 4–6 show the best and worst samples of MAE, MSE, and RMSE results based on six machine learning regression models, respectively. There is a decreasing trend on all evaluation indicators of prediction models for these eight stocks under different predictions of the changes in moving average. When we predict the 20-day change of moving average, it contains more information from the previous data and the model has a better prediction performance with the corresponding training set.



(a) NOW



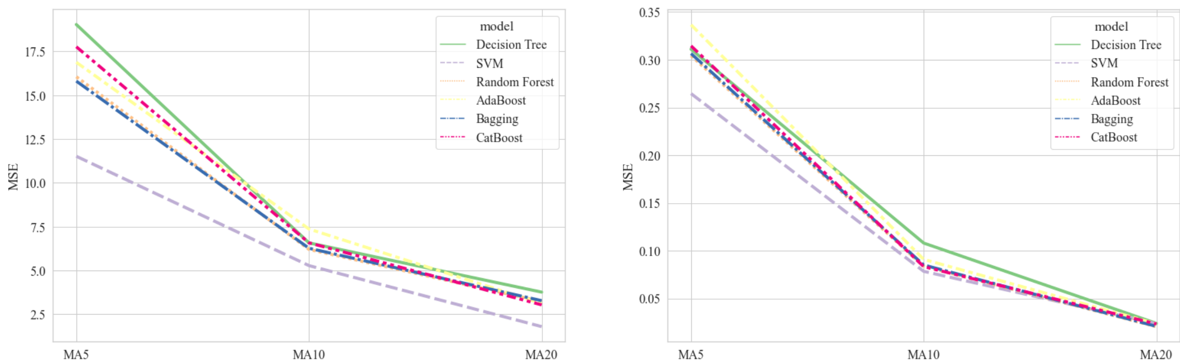
(b) ORCL

Figure 4. MAE results for stocks (2 samples).

Table 4. MAE results for ORCL.

Model	MA5	MA10	MA20
Decision tree	0.3934	0.2373	0.1202
SVM	0.3460	0.2104	0.1211
Bagging	0.3648	0.2115	0.1129
Random forest	0.3650	0.2092	0.1118
Adaboost	0.3776	0.2207	0.1170
Catboost	0.3731	0.2091	0.1193

Figure 4 gives the comparisons of MAE results among six machine learning prediction models for two samples of NOW and ORCL under different predicted targets. For most stocks, the result of the SVM regression model is much better than others and the lowest MAE value is less than 0.13, although all models have great prediction performance on the ORCL stock. The MAE values for the ORCL stock based on different models are shown in Table 4. In contrast to the performance of the SVM model, another single predictor decision tree always has the worst regression results on predicting different stock price changes, which is nearly 0.4. The largest difference between the MAE values from the SVM and decision tree models is near 1 when the predicted variable is the 5-day change of moving average for the ADBE stock. Moreover, four ensemble learning prediction models usually have similar performance for most stocks and even have smaller MAE values than the SVM model.

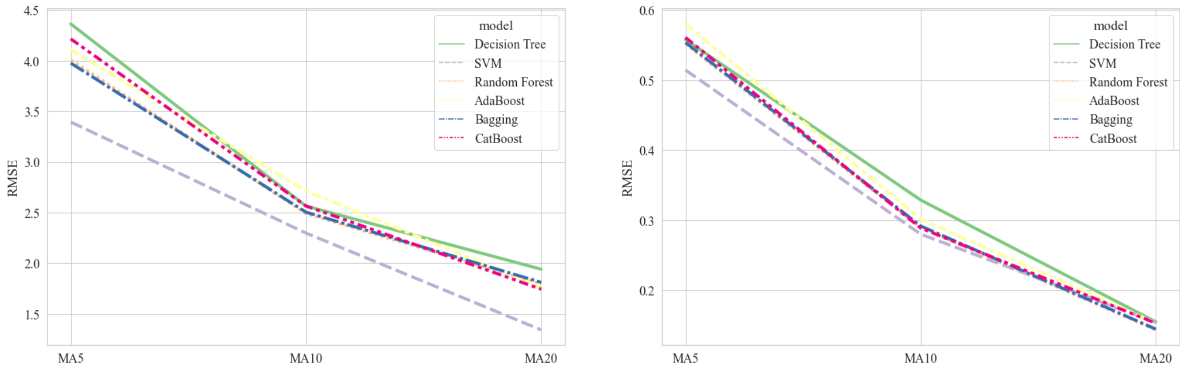


(a) NOW (b) ORCL

Figure 5. MSE results for stocks (2 samples).

Table 5. MSE results for ORCL.

Model	MA5	MA10	MA20
Decision tree	0.3110	0.1080	0.0240
SVM	0.2645	0.0784	0.0237
Bagging	0.3065	0.0850	0.0208
Random forest	0.3042	0.0838	0.0205
Adaboost	0.3366	0.0909	0.0235
Catboost	0.3145	0.0834	0.0233



(a) NOW (b) ORCL

Figure 6. RMSE results for stocks (2 samples).

Table 6. RMSE results for ORCL.

Model	MA5	MA10	MA20
Decision tree	0.5577	0.3287	0.1549
SVM	0.5143	0.2801	0.1540
Bagging	0.5536	0.2916	0.4447
Random forest	0.5516	0.2895	0.1434
Adaboost	0.5802	0.3016	0.1533
Catboost	0.5608	0.2888	0.1529

The MSE results for two sample stocks in Figure 5 have a similar performance. The differences in MSE values among six prediction models for predicting the 5-day moving average change are larger than the MAE values for most stocks. But there is no doubt that the SVM model also has the best performance on predicting moving average changes for different periods. Table 5 also provides the information that the SVM model has relatively lower MSE results for the ORCL stock. In addition, similar to the MAE results, there is a very close relationship among the four ensemble learning models. Figure 6 and Table 6 show the RMSE results and there is a similar distribution of results because the RMSE value is the square root of the MSE value.

Based on the regression evaluation indicators, we can easily find that the well-known SVM regression model has the best relative prediction performance for each stock with different prediction variables. Although the four ensemble learning models use different methods to combine many single predictors, their regression performance is always very close for each stock. However, the popular prediction model with a simple structure-the decision tree model almost performs with the largest prediction errors for each situation compared to other models.

Using the methods in this research, we can calculate the predicted close price of each stock by the corresponding formula and there are sample results that compare the predicted close price of the CRM stock in the test set period with different machine learning prediction models under 5-day, 10-day and 20-day moving average changes.

Table 7. Regression results for CRM under moving average 5.

Model	MAE	MSE	RMSE
Decision tree	1.1131	2.4301	1.5588
SVM	0.7980	1.2867	1.1343
Bagging	0.9182	1.5425	1.2419
Random forest	0.9136	1.5624	1.2499
Adaboost	0.9133	1.5942	1.2626
Catboost	0.9489	1.7389	1.3186

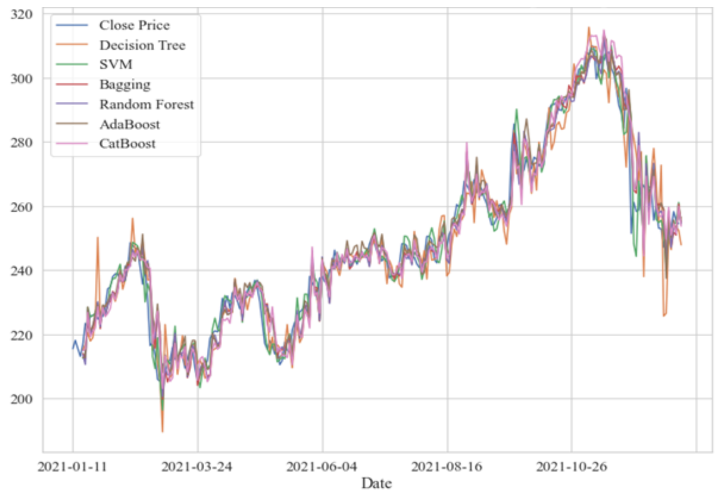


Figure 7. Predicted close price for CRM under moving average 5.

Figure 7 shows the predicted close price for the CRM stock with the 5-day change of the moving average. The prediction performance of the decision tree model has significant fluctuations compared to the actual close price, while other models have smaller fluctuations above and below the actual one. Table 7 shows the corresponding regression results for each predictive model and the SVM model has the lowest MAE, MSE and RMSE.

When the prediction variable becomes a medium-term trend (10-day change of the moving average), it seems that almost all models have a more consistent prediction performance in the middle of the experiment period in Figure 8. However, compared with the result of the 5-day trend prediction variable, the decision tree model has larger errors in prediction values, and even some predicted outliers. The corresponding regression results of the CRM stock for different models in Table 8 can also provide the similar information.

Table 8. Regression results for CRM under moving average 10.

Model	MAE	MSE	RMSE
Decision tree	0.6172	0.8827	0.9395
SVM	0.4473	0.3920	0.6261
Bagging	0.4893	0.4562	0.6754
Random forest	0.4988	0.4688	0.6847
Adaboost	0.5531	0.5832	0.7637
Catboost	0.5609	0.5794	0.7612

Figure 8. Predicted close price for CRM under moving average 10.

Table 9. Regression results for CRM under moving average 20.

Model	MAE	MSE	RMSE
Decision tree	0.4256	0.3371	0.5806
SVM	0.2103	0.0892	0.2987
Bagging	0.4893	0.4562	0.6754
Random forest	0.3486	0.2048	0.4526
Adaboost	0.3713	0.2396	0.4895
Catboost	0.3580	0.2185	0.4675

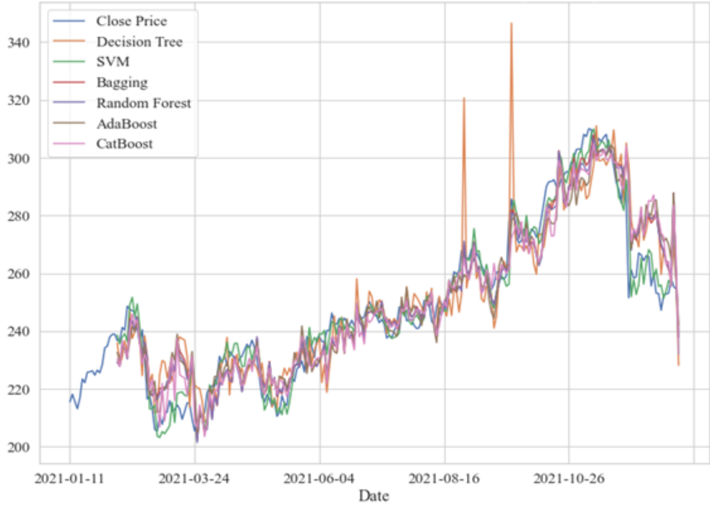


Figure 9. Predicted close price for CRM under moving average 20.

Table 10. Predicted close price for CRM under moving average 20 (10 samples).

Date	Close price	Predicted price	Error	Rate of error
2021-12-09	264.320007	263.007006	−1.313001	−0.4967%
2021-12-10	266.029999	268.173104	2.143105	0.8056%
2021-12-13	265.760010	266.827696	1.067686	0.4017%
2021-12-14	255.589996	258.056676	2.466680	0.9651%
2021-12-15	260.040009	261.481437	1.441428	0.5543%
2021-12-16	253.119995	255.427142	2.307147	0.9115%
2021-12-17	252.929993	255.947815	3.017822	1.1931%
2021-12-20	247.210007	251.325556	4.115549	1.6648%
2021-12-21	252.550003	253.897347	1.347344	0.5335%
2021-12-22	252.800003	257.549671	4.749668	1.8788%

Figure 9 shows the predicted close price of the CRM stock with the corresponding predicted 20-day change of the moving average. Unlike the short-term and medium-term trend predictions, the long-term close price prediction based on the six machine learning models has relatively poor performance. At the end of the test set period, the predicted close price deviates significantly from the actual close price of the CRM stock, except in the SVM model. A similar pattern also occurs at the beginning of this period, when the price drops rapidly and rebounds. Similarly, the regression results for predicting the price of the CRM stock in Table 9 reveals the SVM model performs well since it recives the lowest MAE, MSE and RMSE results. It seems that the SVM regression model with the linear kernel can capture this trend better than other models.

In Figures 8 and 9, there are huge jumps for the CRM price prediction based on the decision tree under 10-day and 20-day moving average changes. Due to the large noise in the stock market, the decision tree model is prone to overfitting in the face of the prediction based on the data set of stocks. It is obvious that the performance of the decision tree model is inferior to other models in predicting the stock price or the trend of stock price changes. Moreover, in other related work, we also find that the decision tree model performs generally compared to other models (Wang, 2023). Although the close prices predicted by different machine learning regression models oscillate around the actual close price, these models can reflect the trend of changes under different moving average changes. The SVM model can especially follow the actual price changes better in most cases.

Table 10 gives ten samples of the comparison between the actual and the predicted close price for CRM stock based on the SVM model under 20-day moving average changes. We can see that the differences between the predicted and actual close price are not very large and the lowest error rate of the prediction model in this period is 0.4017%, which means that machine learning prediction models based on the change of moving average can have good regression performance.

4.2. Classification result

As the research transforms the predicted value into a binary classification problem, it also provides some average classification results based on all 8 stocks in Tables 11–13.

Table 11. Average transformed classification results under moving average 5.

Model	Accuracy	Precision	Recall	F1-score
Decision tree	0.7413	0.7434	0.7413	0.7412
SVM	0.8199	0.8240	0.8199	0.8204
Bagging	0.8102	0.8110	0.8102	0.8098
Random forest	0.8158	0.8167	0.8158	0.8150
Adaboost	0.7903	0.7964	0.7903	0.7841
Catboost	0.7898	0.7915	0.7898	0.7886

To compare the performance of each machine learning algorithm on the stock presented in Table 1, Table 11 shows the average classification results transformed from the predicted results by the six regression machine learning models based on the 5-day moving average changes. As a simple predictor, the decision tree model has the worst prediction performance with an accuracy of 0.7413, while the SVM model has the highest accuracy of 0.8199. Moreover, the bagging model seems to perform better than the boosting models and the random forest model has the best performance among these ensemble models, in which the classification results are close to the SVM model.

Table 12. Average transformed classification results under moving average 10.

Model	Accuracy	Precision	Recall	F1-score
Decision tree	0.8148	0.8153	0.8148	0.8141
SVM	0.8842	0.8871	0.8842	0.8844
Bagging	0.8760	0.8773	0.8760	0.8758
Random forest	0.8781	0.8794	0.8781	0.8777
Adaboost	0.8719	0.8752	0.8719	0.8708
Catboost	0.8617	0.8617	0.8617	0.8611

Table 12 presents the transformed classification results of the 10-day moving average changes prediction. When the predicted variable is a medium-term trend of stocks, the SVM model still has the best classification result, and the accuracy exceeds 0.88. The decision tree model, which has the lowest accuracy, also has an accuracy that is larger than 0.8, which is a good prediction result. Moreover, under the 10-day change of moving average prediction, the classification results of these ensemble learning models also increase by around 0.07.

Table 13. Average transformed classification results under moving average 20.

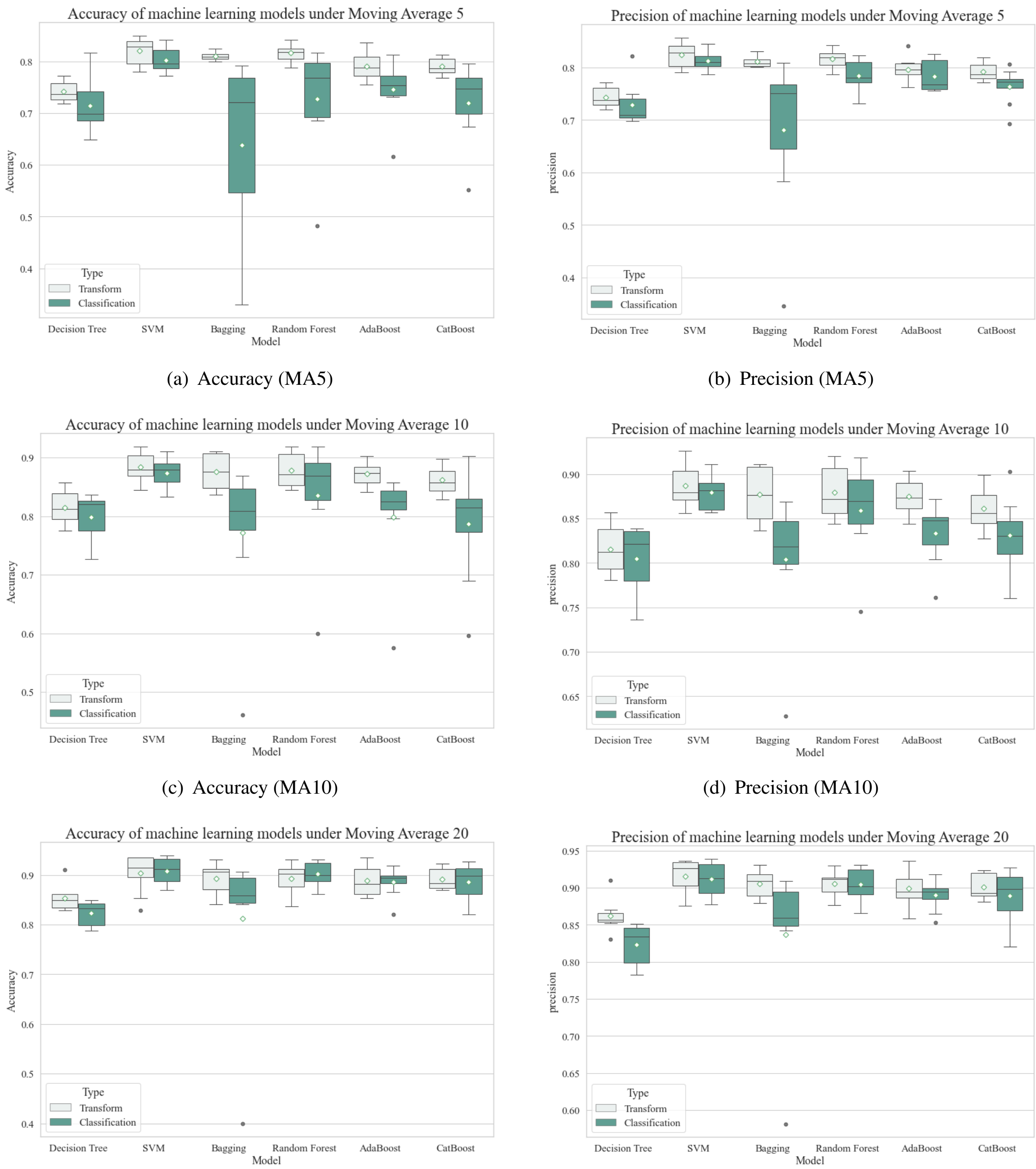
Model	Accuracy	Precision	Recall	F1-score
Decision tree	0.8531	0.8618	0.8531	0.8527
SVM	0.9031	0.9155	0.9031	0.9046
Bagging	0.8929	0.9056	0.8929	0.8932
Random forest	0.8923	0.9061	0.8923	0.8928
Adaboost	0.8888	0.8994	0.8888	0.8875
Catboost	0.8918	0.9013	0.8918	0.8919

Table 13 shows the average transformed classification results of the six machine learning models under the 20-day trend change prediction. In this situation, two single predictors, the decision tree and SVM models, still have the worst and best performance of the stock price trend prediction, respectively. The SVM model has an average accuracy of 0.9031, which is the highest one in the whole study. For ensemble learning models, the bagging model also performs better than the boosting models, while the accuracy of the adaboost model is lower than others.

We also compares the classification results by the corresponding machine learning classification models with the transformed classification results. Figures 10 and 11 give a comparison between the transformed classification results and the classification results predicted by machine learning classification models directly. Overall, the transformation results approach to the prediction by machine learning regression models have better prediction performance.

When the predictions are for the 5-day and 10-day moving average changes, the classification results by transformation are better than those predicted by the classification model directly. Especially for the bagging classification model, it seems not stable to predict the moving average changes, as the boxes of each classification evaluation indicator in the box plots are much larger than others for predicting the 5-day moving average changes. For the 20-day change of moving average prediction, not all transformed classification results are better than those predicted by direct classification models. The average values of the accuracy, recall and F1-score based on the adaboost model from the transformed classification results are lower than those from the direct classification results.

By combining the regression results and the classification results, we can see that the SVM model always has the best performance on predicting different moving average changes, while the decision tree model does not perform well enough. Although the four ensemble learning models with more complex structures have relatively lower results than the SVM model, the prediction performance of these models is stable for different stocks due to the middle prediction errors and the relatively small boxes in the box plots of classification results for different predictions. Moreover, while the 20-day change of moving average prediction has the best regression and classification results, the improvement of results becomes less significant as the number of trend days increases, and the classification results for predicting the 20-day trend are not as good as those predicted directly by the corresponding machine learning classification models.



(a) Accuracy (MA5)

(b) Precision (MA5)

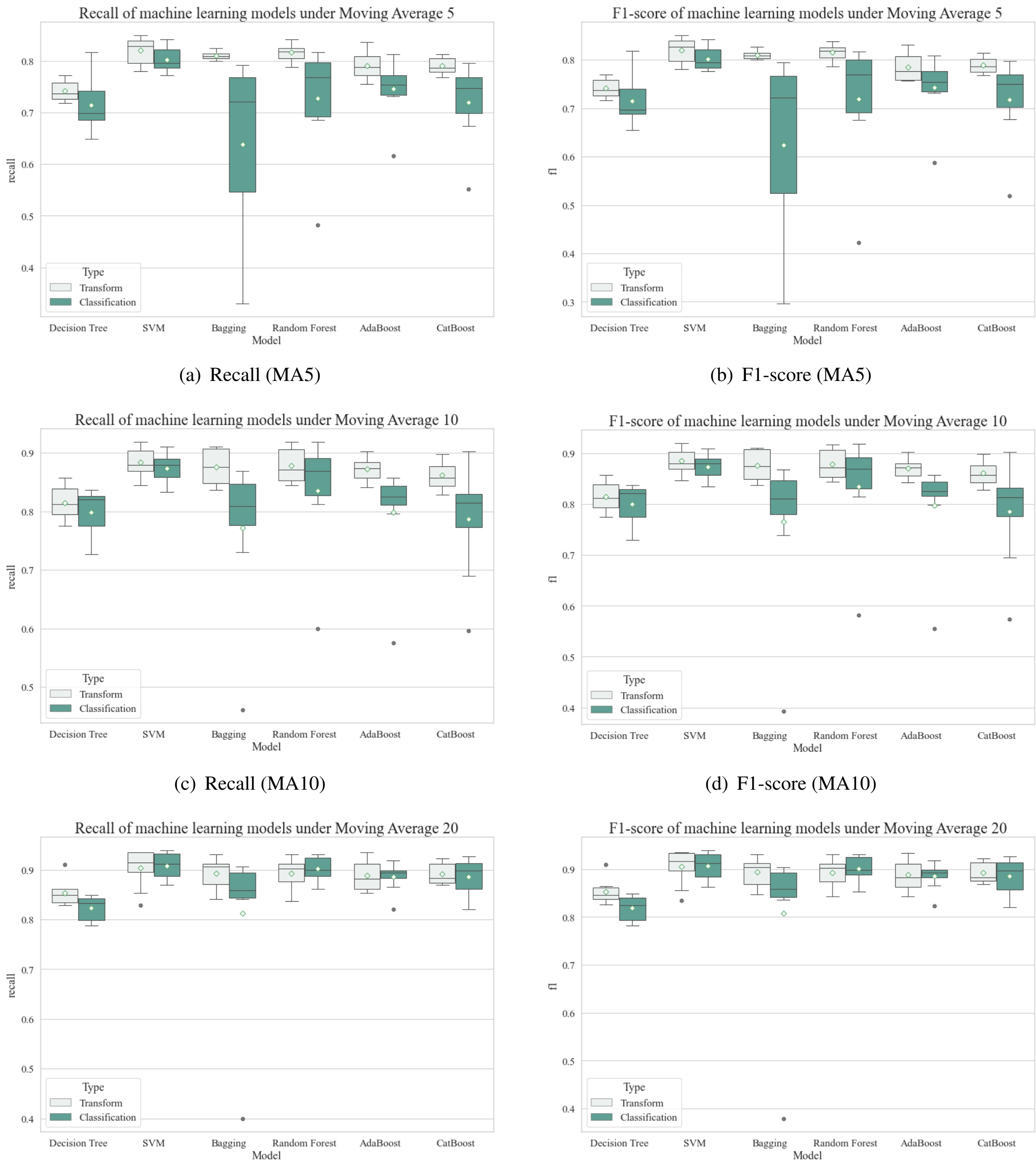
(c) Accuracy (MA10)

(d) Precision (MA10)

(e) Accuracy (MA20)

(f) Precision (MA20)

Figure 10. Comparisons of accuracy and precision results.



(a) Recall (MA5) (b) F1-score (MA5)

(c) Recall (MA10) (d) F1-score (MA10)

(e) Recall (MA20) (f) F1-score (MA20)

Figure 11. Comparisons of recall and F1-score results.

5. Simulation trading experiment

In the previous section, we find that the predictions on different moving average changes based on machine learning regression models can have great prediction performance for both regression and classification results. To help investors understand the role of this method in investing, this section conducts simulation trading experiments related to the transformed classification problem to test whether the method can help investors to earn profits or not. We convert the transformed binary classification problem in equation (22) into a simple machine learning trading strategy: if the $Predicted\ ValueBinary(d)_t$ is equal to 1 and there is no position of this stock, the investor buys the stock; then, if the $Predicted\ ValueBinary(d)_t$ becomes -1 and the investor has a position, the investor sells the stock.

In the simulation trading experiments, we assume there is an investor with \$100,000 investment capital who wants to invest in only one stock during the period of the test set. As there are some transaction fees for trading stocks, we set them as \$2.5 per \$10,000 in this research. We use the backtrader package in Python to complete the experiments.

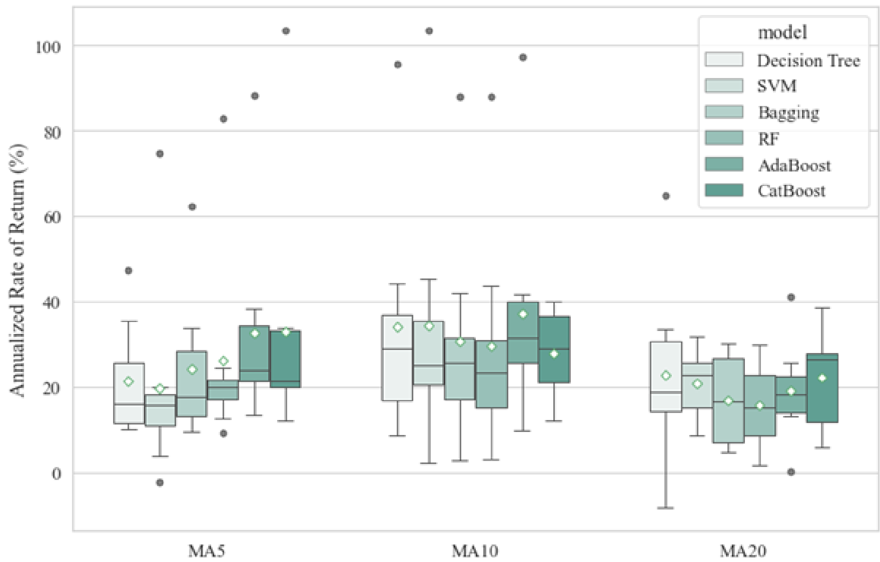


Figure 12. Comparisons of annualized rate of return

Figure 12 shows the annualized rate of return for three types of moving average changes by the machine learning trading strategy. Generally, the annualized rate of return is good when it exceeds 10% and almost all strategies of the transformed method can achieve such high results. For predicting the 5-day and 10-day change of moving averages, the trading strategies based on the six machine learning prediction models have an upward trend, while the average annualized rate of return for each stock decreases for predicting the 20-day trend.

Table 14. Average results for machine learning trading strategies.

Model	Balance	Return	Annualized rate of return (%)	Max drawdown (%)
Decision tree	125213.6	25213.56	26.07	14.20
SVM	124198.9	24198.87	25.02	13.16
Bagging	123127.2	23127.15	23.90	14.28
Random forest	123043.4	23043.42	23.82	13.58
Adaboost	128600.7	28600.74	29.57	13.68
Catboost	126800.0	26800.01	27.70	13.82

We also compute the max drawdown for each trading strategy, which indicates the maximum possible loss for the investor. Table 14 shows the average results of trading strategies based on different machine learning prediction models. Compared with other models, the trading strategy of the adaboost model can help the investor earn the most money with an average annualized rate of return of 29.57%. However, although the annualized rate of return for the SVM model trading strategy is only 25.02%, this strategy has the lowest average max drawdown among others. It suggests that using the SVM trading strategy could be the least risky one for the investor.

From the results of the simulation trading experiments, we find that for different stocks, almost any machine learning trading strategy can make money for the investor. Especially based on the prediction models of the 10-day change of moving average, the average annualized rates of return of these six machine learning trading strategies all exceed 20%. Moreover, the adaboost and SVM trading strategies have the best performance on the average annualized rate of return and max drawdown, respectively.

6. Conclusions

In summary, the method of transforming results from regression machine learning models to classification results is feasible and could help investors identify potential investment opportunities. In this research, we use six popular machine learning prediction models, which are decision tree, SVM, bagging, random forest, adaboost and catboost, to predict different moving average changes based on variables of financial technical indicators. With regression machine learning models, we obtain the best regression result for the 20-day change of moving average prediction for the ORCL stock by the random forest model, which has an MSE value of less than 0.025. The SVM model has the lowest average regression prediction error. Then, by transforming predicted values into classification problems, this research achieves the best average accuracy of 0.9031 when we consider the transformed classification results from the 20-day trend prediction by the SVM model. It is worth noting that the 10-day Change of moving Average prediction is a turning point for the improvement of the model performance, though we always get the best prediction performance from the 20-day change of moving average prediction. This is also reflected in the simulation trading experiments, in which the machine learning trading strategies based on 10-day moving average changes have the highest average annualized rate of return. The results of simulation trading experiments also confirm that this method can be a reference for investors, as most machine learning trading strategies are profitable for both short-term and long-term prediction strategies.

Despite conducting a large number of experiments in this research, there is still room for improvement in the future. Since the results show that the 10-day change of moving average prediction has the best performance in the simulation trading, we can construct more variables like fundamental indicators to help build machine learning models. Although we employ some traditional machine learning models to make predictions and obtain satisfactory experimental results, more models can have good performance on stock price prediction, which we may consider in future studies. For example, a CNN-BiLSTM-Attention model had better accuracy of predicting the stock price index than the traditional model long short term memory (Zhang et al., 2023). Besides, some new models with mode decomposition helped to improve the performance on the stock price prediction. The new hybrid model VML sliced the stock price series to different window series and used variational mode decomposition to decompose the window series into subseries, then made preditions on each subseries to reduce the MSE (Liu et al., 2022). Except the choice of predictive models, we can also change the setting of labels and improve the design of related trading strategies. Moreover, in this research, although we collect data from eight stocks, we only conduct experiments on each stock separately. We can try to assign weights to these stocks to form an optimal portfolio for investors. Ultimately, our goal is to build models with high performance and to provide valuable guidance for investors.

Use of AI tools declaration

The authors affirm that no artificial intelligence (AI) tools are used in the creation of this work.

Funding

The authors have received no financial assistance from any source in the preparation of this work.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

Ampomah EK, Qin Z, Nyame G, et al. (2021) Stock market decision support modeling with tree-based AdaBoost ensemble machine learning models. *Informatica* 44. https://doi.org/10.31449/inf.v44i4.3159

Basak S, Kar S, Saha S, et al. (2019) Predicting the direction of stock market prices using tree-based classifiers. *N Am J Econ Financ* 47: 552–567. https://doi.org/10.1016/j.najef.2018.06.013

Breiman L (1996) Bagging predictors. *Mach Learn* 24: 123–140.

Breiman L (2001) Random forests. *Mach Learn* 45: 5–32.

Collobert R, Bengio S (2001) SVMTorch: Support vector machines for large-scale regression problems. *J Mach Learn Res* 1: 143–160.

Dinesh S, Rao N, Anusha SP, et al. (2021) Prediction of Trends in Stock Market using Moving Averages and Machine Learning. *the 6th International Conference for Convergence in Technology*: 1–5. https://doi.org/10.1109/I2CT51068.2021.9418097

Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput System Sci* 55: 119–139. https://doi.org/10.1006/jcss.1997.1504

Géron A (2022) Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. *O'Reilly Media, Inc.*

Henrique BM, Sobreiro VA, Kimura H (2018) Stock price prediction using support vector regression on daily and up to the minute prices. *J Financ Data Sci* 4: 183–201. https://doi.org/10.1016/j.jfds.2018.04.003

Hindrayani KM, Fahrudin TM, Aji RP, et al. (2020) Indonesian stock price prediction including covid19 era using decision tree regression. *the 3rd International Seminar on Research of Information Technology and Intelligent Systems*: 344–347. https://doi.org/10.1109/ISRITI51436.2020.9315484

Kamalov F (2020) Forecasting significant stock price changes using neural networks. *Neural Comput Appl* 32: 17655-017667. https://doi.org/10.1007/s00521-020-04942-3

Khaidem L, Saha S, Dey SR (2016) Predicting the direction of stock market prices using random forest. *arXiv preprint*: 1605.00003. https://doi.org/10.48550/arXiv.1605.00003

Khan W, Ghazanfar MA, Azam MA, et al. (2022) Stock market prediction using machine learning classifiers and social media, news. *J Amb Intel Hum Comp* 13: 3433–3456. https://doi.org/10.1007/s12652-020-01839-w

Lai CY, Chen RC, Caraka RE (2019) Prediction stock price based on different index factors using LSTM. *2019 International conference on machine learning and cybernetics*: 1–6. https://doi.org/10.1109/ICMLC48188.2019.8949162

Li Y, Yan K (2023) Prediction of Barrier Option Price Based on Antithetic Monte Carlo and Machine Learning Methods. *Cloud Comput Data Sci* 4: 77–86. https://doi.org/10.37256/ccds.4120232110

Liu C, Wang J, Xiao D, et al. (2016) Forecasting S&P 500 stock index using statistical learning models. *Open J Stat* 6: 1067–1075. https://doi.org/10.4236/ojs.2016.66086

Liu T, Ma X, Li S, et al. (2021) A stock price prediction method based on meta-learning and variational mode decomposition. *Knowledge-Based Syst* 252: 109324. https://doi.org/10.1016/j.knosys.2022.109324

Nti IK, Adekoya AF, Weyori BA (2020) A systematic review of fundamental and technical analysis of stock market predictions. *Artif Intell Rev* 53: 3007–3057. https://doi.org/10.1007/s10462-019-09754-z

Obthong M, Tantisantiwong N, Jeamwatthanachai W, et al. (2020) A survey on machine learning for stock price prediction: Algorithms and techniques. Available from: https://eprints.soton.ac.uk/437785/

Prokhorenkova L, Gusev G, Vorobev A, et al. (2018) CatBoost: unbiased boosting with categorical features. *Adv Neural Inform Proc Syst* 31.

Subasi A, Amir F, Bagedo K, et al. (2021) Stock Market Prediction Using Machine Learning. *Procedia Comput Sci* 194: 173–179. https://doi.org/10.1016/j.procs.2021.10.071

Vijh M, Chandola D, Tikkiwal VA, et al. (2020) Stock closing price prediction using machine learning techniques. *Procedia Comput Sci* 167: 599–606. https://doi.org/10.1016/j.procs.2020.03.326

Wang Y (2023) A study on stock price prediction based on machine learning models. *Master dissertation, University of Macau*: 1–56.

Wang Y, Yan K (2022) Prediction of Significant Bitcoin Price Changes Based on Deep Learning. *5th International Conference on Data Science and Information Technology (DSIT)*: 1–5. https://doi.org/10.1109/DSIT55514.2022.9943971

Wang Y, Yan K (2023) Application of Traditional Machine Learning Models for Quantitative Trading of Bitcoin. *Artif Intell Evol* 4: 34–48. https://doi.org/10.37256/aie.4120232226

Yan K, Wang Y (2023) Prediction of Bitcoin prices' trends with ensemble learning models. *5th International Conference on Computer Information Science and Artificial Intelligence* 12566: 900–905. https://doi.org/10.1117/12.2667793

Yan K, Wang Y, Li Y (2023) Enhanced Bollinger Band Stock Quantitative Trading Strategy Based on Random Forest. *Artif Intell Evol* 4: 22–33. https://doi.org/10.37256/aie.4120231991

Zhang C, Ji Z, Zhang J, et al. (2018) Predicting Chinese stock market price trend using machine learning approach. *the 2nd International Conference on Computer Science and Application Engineering*: 1–5. https://doi.org/10.1145/3207677.3277966

Zhang J, Ye L, Lai Y (2023) Stock Price Prediction Using CNN-BiLSTM-Attention Model. *Mathematics* 11: 1985. https:// doi.org/10.3390/math11091985