Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 7816154, 30 pages
https://doi.org/10.1155/2019/7816154

Research Article

An Empirical Study of Machine Learning Algorithms for Stock Daily Trading Strategy

Dongdong Lv[1], **Shuhan Yuan**[2], **Meizi Li**[1,3] **and Yang Xiang**[1]

[1]*College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China*
[2]*University of Arkansas, Fayetteville, AR 72701, USA*
[3]*College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China*

Correspondence should be addressed to Yang Xiang; shxiangyang@tongji.edu.cn

Received 17 October 2018; Revised 3 March 2019; Accepted 19 March 2019; Published 14 April 2019

Academic Editor: Kemal Polat

According to the forecast of stock price trends, investors trade stocks. In recent years, many researchers focus on adopting machine learning (ML) algorithms to predict stock price trends. However, their studies were carried out on small stock datasets with limited features, short backtesting period, and no consideration of transaction cost. And their experimental results lack statistical significance test. In this paper, on large-scale stock datasets, we synthetically evaluate various ML algorithms and observe the daily trading performance of stocks under transaction cost and no transaction cost. Particularly, we use two large datasets of 424 S&P 500 index component stocks (SPICS) and 185 CSI 300 index component stocks (CSICS) from 2010 to 2017 and compare six traditional ML algorithms and six advanced deep neural network (DNN) models on these two datasets, respectively. The experimental results demonstrate that traditional ML algorithms have a better performance in most of the directional evaluation indicators. Unexpectedly, the performance of some traditional ML algorithms is not much worse than that of the best DNN models without considering the transaction cost. Moreover, the trading performance of all ML algorithms is sensitive to the changes of transaction cost. Compared with the traditional ML algorithms, DNN models have better performance considering transaction cost. Meanwhile, the impact of transparent transaction cost and implicit transaction cost on trading performance are different. Our conclusions are significant to choose the best algorithm for stock trading in different markets.

1. Introduction

The stock market plays a very important role in modern economic and social life. Investors want to maintain or increase the value of their assets by investing in the stock of the listed company with higher expected earnings. As a listed company, issuing stocks is an important tool to raise funds from the public and expand the scale of the industry. In general, investors make stock investment decisions by predicting the future direction of stocks' ups and downs. In modern financial market, successful investors are good at making use of high-quality information to make investment decisions, and, more importantly, they can make quick and effective decisions based on the information they have already had. Therefore, the field of stock investment attracts the attention not only of financial practitioner and ordinary investors but also of researchers in academic [1].

In the past many years, researchers mainly constructed statistical models to describe the time series of stock price and trading volume to forecast the trends of future stock returns [2–4]. It is worth noting that the intelligent computing methods represented by ML algorithms also present a vigorous development momentum in stock market prediction with the development of artificial intelligence technology. The main reasons are as follows. (1) Multisource heterogeneous financial data are easy to obtain, including high-frequency trading data, rich and diverse technical indicators data, macroeconomic data, industry policy and regulation data, market news, and even social network data. (2) The research of intelligent algorithms has been deepened. From the early linear model, support vector machine, and shallow neural network to DNN models and reinforcement learning algorithms, intelligent computing methods have made significant improvement. They have been effectively applied to the fields

of image recognition and text analysis. In some papers, the authors think that these advanced algorithms can capture the dynamic changes of the financial market, simulate the trading process of stock, and make automatic investment decisions. (3) The rapid development of high-performance computing hardware, such as Graphics Processing Units (GPUs), large servers, and other devices, can provide powerful storage space and computing power for the use of financial big data. High-performance computer equipment, accurate and fast intelligent algorithms, and financial big data together can provide decision-making support for programmed and automated trading of stocks, which has gradually been accepted by industry practitioners. Therefore, the power of financial technology is reshaping the financial market and changing the format of finance.

Over the years, traditional ML methods have shown strong ability in trend prediction of stock prices [2–16]. In recent years, artificial intelligence computing methods represented by DNN have made a series of major breakthroughs in the fields of Natural Language Processing, image classification, voice translation, and so on. It is noteworthy that some DNN algorithms have been applied for time series prediction and quantitative trading [17–34]. However, most of the previous studies focused on the prediction of the stock index of major economies in the world ([2, 8, 11, 13, 15–17, 22, 29, 30, 32], etc.) or selecting a few stocks with limited features according to their own preferences ([8–11, 14, 17, 20, 22, 26, 31], etc.) or not considering transaction cost ([10, 14, 17, 23], etc.), or the period of backtesting is very short ([2, 8, 9, 11, 17, 20, 22, 27], etc.). Meanwhile, there is no statistical significance test between different algorithms which were used in stock trading ([8–11, 32], etc.). That is, the comparison and evaluation of the various trading algorithms lack large-scale stocks datasets, considering transaction cost and statistical significance test. Therefore, the performance of backtesting may tend to be overly optimistic. In this regard, we need to clarify two concerns based on a large-scale stock dataset: (1) whether the trading strategies based on the DNN models can achieve statistically significant results compared with the traditional ML algorithms without transaction cost; (2) how do transaction costs affect trading performance of the ML algorithm? These problems constitute the main motivation of this research and they are very important for quantitative investment practitioners and portfolio managers. These solutions of these problems are of great value for practitioners to do stock trading.

In this paper, we select 424 SPICS and 185 CSICS from 2010 to 2017 as research objects. The SPICS and CSICS represent the industry development of the world's top two economies and are attractive to investors around the world. The stock symbols are shown in the "Data Availability". For each stock in SPICS and CSICS, we construct 44 technical indicators as shown in the "Data Availability". The label on the T-th trading day is the symbol for the yield of the $T + 1$-th trading day relative to the T-th trading day. That is, if the yield is positive, the label value is set to 1, otherwise 0. For each stock, we choose 44 technical indicators of 2000 trading days before December 31, 2017, to build a stock dataset. After the dataset of a stock is built, we choose the walk-forward analysis (WFA) method to train the ML models step by step. In each step of training, we use 6 traditional ML methods which are support vector machine (SVM), random forest (RF), logistic regression (LR), naïve Bayes model (NB), classification and regression tree (CART), and eXtreme Gradient Boosting algorithm (XGB) and 6 DNN models which are widely in the field of text analysis and voice translation such as Multilayer Perceptron (MLP), Deep Belief Network (DBN), Stacked Autoencoders (SAE), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) to train and forecast the trends of stock price based on the technical indicators. Finally, we use the directional evaluation indicators such as accuracy rate (AR), precision rate (PR), recall rate (RR), F1-Score (F1), Area Under Curve (AUC), and the performance evaluation indicators such as winning rate (WR), annualized return rate (ARR), annualized Sharpe ratio (ASR), and maximum drawdown (MDD)) to evaluate the trading performance of these various algorithms or strategies.

From the experiments, we can find that the traditional ML algorithms have a better performance than DNN algorithms in all directional evaluation indicators except for PR in SPICS; in CSICS, DNN algorithms have a better performance in AR, PR, and F1 expert for RR and AUC. (1) Trading performance without transaction cost is as follows: the WR of traditional ML algorithms have a better performance than those of DNN algorithms in both SPICS and CSICS. The ARR and ASR of all ML algorithms are significantly greater than those of the benchmark index (S&P 500 index and CSI 300 index) and BAH strategy; the MDD of all ML algorithms are significantly greater than that of BAH strategy and are significantly less than that of the benchmark index. In all ML algorithms, there are always some traditional ML algorithms whose trading performance (ARR, ASR, MDD) can be comparable to the best DNN algorithms. Therefore, DNN algorithms are not always the best choice, and the performance of some traditional ML algorithms has no significant difference from that of DNN algorithms; even those traditional ML algorithms can perform well in ARR and ASR. (2) Trading performance with transaction cost is as follows: the trading performance (WR, ARR, ASR, and MDD) of all machine learning algorithms is decreasing with the increase of transaction cost as in actual trading situation. Under the same transaction cost structure, the performance reductions of DNN algorithms, especially MLP, DBN, and SAE, are smaller than those of traditional ML algorithms, which shows that DNN algorithms have stronger tolerance and risk control ability to the changes of transaction cost. Moreover, the impact of transparent transaction cost on SPICS is greater than slippage, while the opposite is true on CSICS. Through multiple comparative analysis of the different transaction cost structures, the performance of trading algorithms is significantly smaller than that without transaction cost, which shows that trading performance is sensitive to transaction cost. The contribution of this paper is that we use nonparametric statistical test methods to compare differences in trading performance for different ML algorithms in both cases of transaction cost and no transaction cost. Therefore, it is helpful for us to select the

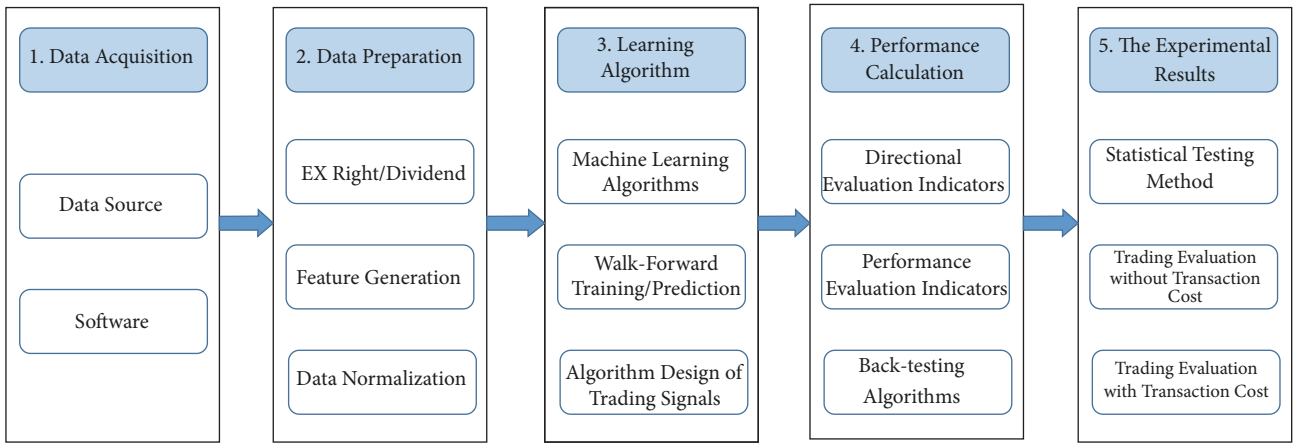

Figure 1: The framework for predicting stock price trends based on ML algorithms.

most suitable algorithm from these ML algorithms for stock trading both in the US stock market and the Chinese A-share market.

The remainder of this paper is organized as follows: Section 2 describes the architecture of this work. Section 3 gives the parameter settings of these ML models and the algorithm for generating trading signals based on the ML models mentioned in this paper. Section 4 gives the directional evaluation indicators, performance evaluation indicators, and backtesting algorithms. Section 5 uses nonparameter statistical test methods to analyze and evaluate the performance of these different algorithms in the two markets. Section 6 gives the analysis of impact of transaction cost on performance of ML algorithms for trading. Section 7 gives some discussions of differences in trading performance among different algorithms from the perspective of data, algorithms, transaction cost, and suggestions for algorithmic trading. Section 8 provides a comprehensive conclusion and future research directions.

2. Architecture of the Work

The general framework of predicting the future price trends of stocks, trading process, and backtesting based on ML algorithms is shown in Figure 1. This article is organized from data acquisition, data preparation, intelligent learning algorithm, and trading performance evaluation. In this study, data acquisition is the first step. Where should we get data and what software should we use to get data quickly and accurately are something that we need to consider. In this paper, we use R language to do all computational procedures. Meanwhile, we obtain SPICS and CSICS from Yahoo finance and Netease Finance, respectively. Secondly, the task of data preparation includes ex-dividend/rights for the acquired data, generating a large number of well-recognized technical indicators as features, and using max-min normalization to deal with the features, so that the preprocessed data can be used as the input of ML algorithms [34]. Thirdly, the trading signals of stocks are generated by the ML algorithms. In this part, we train the DNN models and the traditional ML algorithms by a WFA method; then the trained ML models will predict the direction of the stocks in a future time which is considered as the trading signal. Fourthly, we give some widely used directional evaluation indicators and performance evaluation indicators and adopt a backtesting algorithm for calculating the indicators. Finally, we use the trading signal to implement the backtesting algorithm of stock daily trading strategy and then apply statistical test method to evaluate whether there are statistical significant differences among the performance of these trading algorithms in both cases of transaction cost and no transaction cost.

3. ML Algorithms

3.1. ML Algorithms and Their Parameter Settings. Given a training dataset D, the task of ML algorithm is to classify class labels correctly. In this paper, we will use six traditional ML models (LR, SVM, CART, RF, BN, and XGB) and six DNN models (MLP, DBN, SAE, RNN, LSTM, and GRU) as classifiers to predict the ups and downs of the stock prices [34]. The main model parameters and training parameters of these ML learning algorithms are shown in Tables 1 and 2.

In Tables 1 and 2, features and class labels are set according to the input format of various ML algorithms in R language. *Matrix (m, n)* represents a matrix with m rows and n columns; *Array* (p, m, n) represents a tensor and each layer of the tensor is *Matrix* (m, n) and the height of the tensor is p. c ($h1$, $h2$, $h3$, ...) represents a vector, where the length of the vector is the number of hidden layers and the i-th element of c is the number of neurons of the i-th hidden layer. In the experiment, $m = 250$ represents that we use the data of the past 250 trading days as training samples in each round of WFA; $n = 44$ represents that the data of each day has 44 features. In Table 2, the parameters of DNN models such as activation function, learning rate, batch size, and epoch are all default values in the algorithms of R programs.

3.2. WFA Method. WFA [35] is a rolling training method. We use the latest data instead of all past data to train the model

Table 1: Main parameter settings of traditional ML algorithms.

	Input Features	Label	Main parameters
LR	Matrix(250,44)	Matrix(250,1)	A specification for the model link function is logit.
SVM	Matrix(250,44)	Matrix(250,1)	The kernel function used is Radial Basis kernel; Cost of constraints violation is 1.
CART	Matrix(250,44)	Matrix(250,1)	The maximum depth of any node of the final tree is 20; The splitting index can be Gini coefficient.
RF	Matrix(250,44)	Matrix(250,1)	The Number of trees is 500; Number of variables randomly sampled as candidates at each split is 7.
BN	Matrix(250,44)	Matrix(250,1)	the prior probabilities of class membership is the class proportions for the training set.
XGB	Matrix(250,44)	Matrix(250,1)	The maximum depth of a tree is 10; the max number of iterations is 15; the learning rate is 0.3.

Table 2: Main parameter settings of DNN algorithms.

	Input Features	Label	Learning rate	Dimensions of hidden layers	Activation function	Batch size	Epoch
MLP	Matrix(250,44)	Matrix(250,1)	0.8	c(25,15,10,5)	sigmoid	100	3
DBN	Matrix(250,44)	Matrix(250,1)	0.8	c(25,15,10,5)	sigmoid	100	3
SAE	Matrix(250,44)	Matrix(250,1)	0.8	c(20,10,5)	sigmoid	100	3
RNN	Array(1,250,44)	Array(1,250,1)	0.01	c(10,5)	sigmoid	1	1
LSTM	Array(1,250,44)	Array(1,250,1)	0.01	c(10,5)	sigmoid	1	1
GRU	Array(1,250,44)	Array(1,250,1)	0.01	c(10,5)	sigmoid	1	1

and then apply the trained model to implement the prediction for the out-of-sample data (testing dataset) of the future time period. After that, a new training set, which is the previous training set walk one step forward, is carried out the training of the next round. WFA can improve the robustness and the confidence of the trading strategy in real-time trading.

In this paper, we use ML algorithms and the WFA method to do stock price trend predictions as trading signals. In each step, we use the data from the past 250 days (one year) as the training set and the data for the next 5 days (one week) as the test set. Each stock contains data of 2,000 trading days, so it takes (2000-250)/5 = 350 training sessions to produce a total of 1,750 predictions which are the trading signals of daily trading strategy. The WFA method is as shown in Figure 2.

3.3. The Algorithm Design of Trading Signal. In this part, we use ML algorithms as classifiers to predict the ups and downs of the stock in SPICS and CSICS and then use the prediction results as trading signals of daily trading. We use the WFA method to train each ML algorithm. We give the generating algorithm of trading signals according to Figure 2, which is shown in Algorithm 1.

4. Evaluation Indicators and Backtesting Algorithm

4.1. Directional Evaluation Indicators. In this paper, we use ML algorithms to predict the direction of stock price, so the main task of the ML algorithms is to classify returns. Therefore, it is necessary for us to use directional evaluation indicators to evaluate the classification ability of these algorithms.

The actual label values of the dataset are sequences of sets {DOWN, UP}. Therefore, there are four categories of predicted label values and actual label values, which are expressed as TU, FU, FD, and TD. TU denotes the number of UP that the actual label values are UP and the predicted label values are also UP; FU denotes the number of UP that the actual label values are DOWN but the predicted label values are UP; TD denotes the number of DOWN that the actual label values are DOWN and the predicted label values are DOWN; FD denotes the number of DOWN that the actual label values are UP but the predicted label values are DOWN, as shown in Table 3. Table 3 is a two-dimensional table called confusion matrix. It classifies predicted label values according to whether predicted label values match real label values. The first dimension of the table represents all possible predicted label values and the second dimension represents all real label values. When predicted label values equal real label values, they are correct classifications. The correct prediction label values lie on the diagonal line of the confusion matrix. In this paper, what we are concerned about is that when the direction of stock price is predicted to be UP tomorrow, we buy the stock at today's closing price and sell it at tomorrow's closing price; when we predict the direction of stock price to be DOWN tomorrow, we do nothing. So UP is a "positive" label of our concern.

Table 3: Confusion matrix of two classification results of ML algorithm.

		Predicted label values	
		UP	DOWN
Actual label values	UP	TU	FD
	DOWN	FU	TD

In most of classification tasks, AR is generally used to evaluate performance of classifiers. AR is the ratio of the number of correct predictions to the total number of predictions. That is as follows.

$$AR = \frac{(TU + TD)}{(TU + FD + FU + TD)} \quad (1)$$

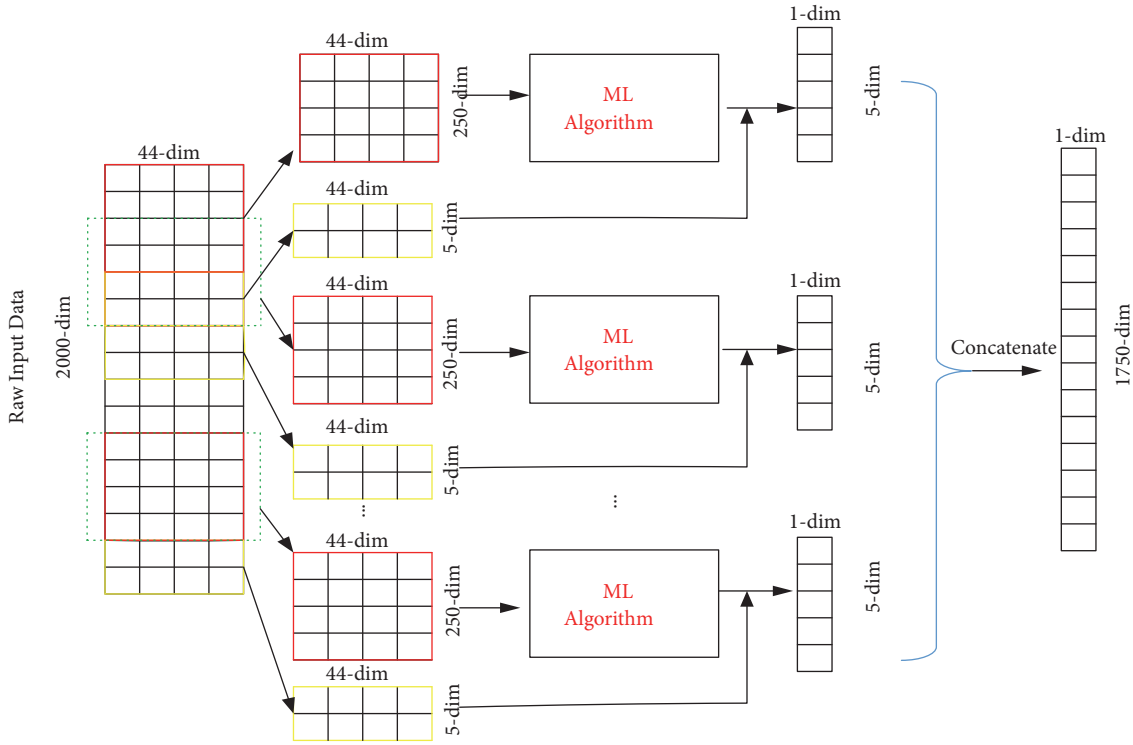

Figure 2: The schematic diagram of WFA (training and testing).

```
Input: Stock Symbols
Output: Trading Signals
(1)  N=Length of Stock Symbols
(2)  L=Length of Trading Days
(3)  P=Length of Features
(4)  k= Length of Training Dataset for WFA
(5)  n= Length of Sliding Window for WFA
(6)  for (i in 1: N) {
(7)       Stock=Stock Symbols[i]
(8)       M=(L-k)/n
(9)       Trading Signal=NULL
(10)      for (j in 1:M) {
(11)           Dataset= Stock[(k+n*(j-1)):(k+n+n*(j-1)), 1:(P+1)]
(12)           Train=Dataset[1:k,1:(1+P)]
(13)           Test= Dataset[(k+1):(k+n),1:P]
(14)           Model=ML Algorithm(Train)
(15)           Probability=Model(Test)
(16)           if (Probability>=0.5) {
(17)              Trading Signal0=1
(18)              } else {
(19)              Trading Signal0=0
(20)                   }
(21)           }
(22)           Trading Signal=c (Trading Signal, Trading Signal0)
(23)      }
(24) return (Trading Signal)
```

Algorithm 1: Generating trading signal in R language.

In this paper, "UP" is the profit source of our trading strategies. The classification ability of ML algorithm is to evaluate whether the algorithms can recognize "UP". Therefore, it is necessary to use PR and RR to evaluate classification results. These two evaluation indicators are initially applied in the field of information retrieval to evaluate the relevance of retrieval results.

PR is a ratio of the number of correctly predicted UP to all predicted UP. That is as follows.

$$PR = \frac{TU}{(TU + FU)} \tag{2}$$

High PR means that ML algorithms can focus on "UP" rather than "DOWN".

RR is the ratio of the number of correctly predicted "UP" to the number of actually labeled "UP". That is as follows.

$$PR = \frac{TU}{(TU + FD)} \tag{3}$$

High RR can capture a large number of "UP" and be effectively identified. In fact, it is very difficult to present an algorithm with high PR and RR at the same time. Therefore, it is necessary to measure the classification ability of the ML algorithm by using some evaluation indicators which combine PR with RR. F1-Score is the harmonic average of PR and AR. F1 is a more comprehensive evaluation indicator. That is as follows.

$$F_1 = 2 * PR * \frac{AR}{(PR + AR)} \tag{4}$$

Here, it is assumed that the weights of PR and RR are equal when calculating F1, but this assumption is not always correct. It is feasible to calculate F1 with different weights for PR and RR, but determining weights is a very difficult challenge.

AUC is the area under ROC (Receiver Operating Characteristic) curve. ROC curve is often used to check the tradeoff between finding TU and avoiding FU. Its horizontal axis is FU rate and its vertical axis is TU rate. Each point on the curve represents the proportion of TU under different FU thresholds [36]. AUC reflects the classification ability of classifier. The larger the value, the better the classification ability. It is worth noting that two different ROC curves may lead to the same AUC value, so qualitative analysis should be carried out in combination with the ROC curve when using AUC value. In this paper, we use R language package "ROCR" to calculate AUC.

4.2. Performance Evaluation Indicator. Performance evaluation indicator is used for evaluating the profitability and risk control ability of trading algorithms. In this paper, we use trading signals generated by ML algorithms to conduct the backtesting and apply the WR, ARR, ASR, and MDD to do the trading performance evaluation [34]. WR is a measure of the accuracy of trading signals; ARR is a theoretical rate of return of a trading strategy; ASR is a risk-adjusted return which represents return from taking a unit risk [37] and the risk-free return or benchmark is set to 0 in this paper; MDD is the largest decline in the price or value of the investment period, which is an important risk assessment indicator.

4.3. Backtesting Algorithm. Using historical data to implement trading strategy is called backtesting. In research and the development phase of trading model, the researchers usually use a new set of historical data to do backtesting. Furthermore, the backtesting period should be long enough, because a large number of historical data can ensure that the trading model can minimize the sampling bias of data. We can get statistical performance of trading models theoretically by backtesting. In this paper, we get 1750 trading signals for each stock. If tomorrow's trading signal is 1, we will buy the stock at today's closing price and then sell it at tomorrow's closing price; otherwise, we will not do stock trading. Finally, we get AR, PR, RR, F1, AUC, WR, ARR, ASR, and MDD by implementing backtesting algorithm based on these trading signals.

5. Comparative Analysis of Different Trading Algorithms

5.1. Nonparametric Statistical Test Method. In this part, we use the backtesting algorithm(Algorithm 2) to calculate the evaluation indicators of different trading algorithms. In order to test whether there are significant differences between the evaluation indicators of different ML algorithms, the benchmark indexes, and the BAH strategies, it is necessary to use analysis of variance and multiple comparisons to give the answers. Therefore, we propose the following nine basic hypotheses for significance test in which *Hja* ($j = 1, 2, 3, 4, 5, 6, 7, 8, 9$) are the null hypothesis, and the corresponding alternative assumptions are *Hjb* ($j = 1, 2, 3, 4, 5, 6, 7, 8, 9$). The level of significance is 0.05.

For any evaluation indicator $j \in \{AR, PR, RR, F1, AUC, WR, ARR, ASR, MDD\}$ and any trading strategy $i \in \{MLP, DBN, SAE, RNN, LSTM, GRU, LR, SVM, NB, CART, RF, XGB, BAH, Benchmark\ index\}$, the null hypothesis a is *Hja*, alternative hypotheses b is *Hjb* ($j = 1, 2, 3, 4, 5, 6, 7, 8, 9$ represent AR, PR, RR, F1, AUC, WR, ARR, ASR, MDD, respectively.).

> *Hja*: the evaluation indicator *j* of all strategies are the same
>
> *Hjb*: the evaluation indicator *j* of all strategies are not the same

It is worth noting that any evaluation indicator of all trading algorithm or strategy does not conform to the basic hypothesis of variance analysis. That is, it violates the assumption that the variances of any two groups of samples are the same and each group of samples obeys normal distribution. Therefore, it is not appropriate to use t-test in the analysis of variance, and we should take the nonparametric statistical test method instead. In this paper, we use the Kruskal-Wallis rank sum test [38] to carry out the analysis of variance. If the alternative hypothesis is established, we will need to further apply the Nemenyi test [39] to do the multiple comparisons between trading strategies.

5.2. Comparative Analysis of Performance of Different Trading Strategies in SPICS. Table 4 shows the average value of

```
Input: TS #TS is trading signals of a stock.
Output: AR, PR, RR, F1, AUC, WR, ARR, ASR, MDD
(1)  N=length of Stock Code List #424 SPICS, and 185 CSICS.
(2)  B_t=Benchmark Index ["Closing Price"] # B is the closing price of benchmark index.
(3)  WR=NULL; ARR=NULL; ASR=NULL; MDD=NULL
(4)  for (i in 1: N) {
(5)      Stock Data=Stock Code List[i]
(6)      P_t=Stock_Data ["Closing Price"]
(7)      Label_t= Stock_Data ["Label"]
(8)      BDRR_t=(B_t-B_t-1)/ B_t-1 # BDRR is the daily return rate of benchmark index.
(9)      DRR_t= (P_t-P_t-1)/P_t-1 #DRR is daily return rate. That is daily return rate of BAH strategy.
(10)     TDRR_t=lag (TS_t)*DRR_t #TDRR is the daily return through trading.
(11)     Table=Confusion_Matrix(TS, Label)
(12)     AR[i]=sum(adj(Table))/sum(Table)
(13)     PR[i]=Table[2, 2]/sum(Table[, 2])
(14)     RR[i]=Table[2, 2]/sum(Table[2, ])
(15)     F1=2*PR[i]*RR[i]/(PR[i]+RR[i])
(16)     Pred=prediction (TS, Label)
(17)     AUC[i]=performance (Pred, measure="auc")@y.values[[1]]
(18)     WR[i]=sum (TDRR>0)/sum(TDRR≠0)
(19)     ARR[i]=Return.annualized (TDRR)# TDRR, BDRR, or DRR can be used.
(20)     ASR[i]=SharpeRatio.annualized (TDRR)# TDRR, BDRR, or DRR can be used.
(21)     MDD[i]=maxDrawDown (TDRR)# TDRR, BDRR, or DRR can be used.
(22)     AR=c (AR, AR[i])
(23)     PR=c (PR, PR[i])
(24)     RR=c (RR, RR[i])
(25)     F1=c (F1, F1[i])
(26)     AUC=c (AUC, AUC[i])
(27)     WR=c (WR, WR[i])
(28)     ARR=c (ARR, ARR[i])
(29)     ASR=c (ASR, ASR[i])
(30)     MDD=c (MDD, MDD[i])
(31)     }
(32)     Performance=cbind (AR, PR, RR, F1, AUC, WR, ARR, ASR, MDD)
(33) return (Performance)
```

Algorithm 2: Backtesting algorithm of daily trading strategy in R language.

Table 4: Trading performance of different trading strategies in the SPICS. Best performance of all trading strategies is in boldface.

	Index	BAH	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM	XGB
AR	—	—	0.5205	0.5189	0.5201	0.5025	0.5013	0.4986	0.6309	0.5476	0.6431	0.6491	0.6235	**0.6600**
PR	—	—	**0.7861**	0.7764	0.7781	0.5427	0.5121	0.4911	0.6514	0.5270	0.6595	0.6474	0.6733	0.6738
RR	—	—	0.5274	0.5263	0.5273	0.5245	0.5253	0.5239	0.6472	0.5762	0.6599	0.6722	0.6325	**0.6767**
F1	—	—	0.6258	0.6217	0.6229	0.5332	0.5183	0.5065	0.6491	0.5480	0.6595	0.6591	0.6517	**0.6751**
AUC	—	—	0.5003	0.5001	0.5002	0.4997	0.5005	0.4992	0.6295	0.5489	0.6418	0.6491	0.6199	**0.6590**
WR	0.5450	0.5235	0.5676	0.5680	0.5683	0.5843	0.5825	0.5844	0.5266	**0.5930**	0.5912	0.5859	0.5831	0.5891
ARR	0.1227	0.1603	**0.3333**	0.3298	0.3327	0.2945	0.2921	0.2935	0.3319	0.2976	0.3134	0.2944	0.3068	0.3042
ASR	0.8375	0.6553	1.5472	1.5415	1.5506	1.5768	1.5575	1.5832	1.3931	1.6241	**1.6768**	1.5822	1.6022	1.6302
MDD	**0.1939**	0.4233	0.3584	0.3585	0.3547	0.3403	0.3489	0.3381	0.3413	0.3428	0.3284	0.3447	0.3429	0.3338

various trading algorithms in AR, PR, RR, F1, AUC, WR, ARR, ASR, and MDD. We can see that the AR, RR, F1, and AUC of XGB are the greatest in all trading algorithms. The WR of NB is the greatest in all trading strategies. The ARR of MLP is the greatest in all trading strategies including the benchmark index (S&P 500 index) and BAH strategy. The ASR of RF is the greatest in all trading strategies. The MDD of the benchmark index is the smallest in all trading strategies. It is worth noting that the ARR and ASR of all ML algorithms are greater than those of BAH strategy and the benchmark index.

(1) Through the hypothesis test analysis of H1a and H1b, we can obtain p value<2.2e-16.

Therefore, there are statistically significant differences between the AR of all trading algorithms. Therefore, we need to make multiple comparative analysis further, as shown in

Table 5: Multiple comparison analysis between the AR of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
DBN	1.0000										
SAE	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	**0.0000**								
LSTM	**0.0000**	**0.0000**	**0.0000**	1.0000							
GRU	**0.0000**	**0.0000**	**0.0000**	0.8273	0.9811						
CART	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**					
NB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**				
RF	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0232**	**0.0000**			
LR	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.7649		
SVM	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.6057	**0.0000**	**0.0000**	**0.0000**	
XGB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0002**	0.2010	**0.0000**

Table 6: Multiple comparison analysis between the PR of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
DBN	0.9999										
SAE	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	**0.0000**								
LSTM	**0.0000**	**0.0000**	**0.0000**	**0.0034**							
GRU	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.1472						
CART	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**					
NB	**0.0000**	**0.0000**	**0.0000**	0.7869	0.5786	**0.0000**	**0.0000**				
RF	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.8056	**0.0000**			
LR	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.9997	**0.0000**	0.2626		
SVM	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0008**	**0.0000**	0.3104	**0.0000**	
XGB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0491**	**0.0000**	0.9999

Table 5. The number in the table is a p value of any two algorithms of Nemenyi test. When p value<0.05, we think that the two trading algorithms have a significant difference, otherwise we cannot deny the null assumption that the mean values of AR of the two algorithms are equal. From Tables 5 and 4, we can see that the AR of all DNN models are significantly lower than those of all traditional ML models. The AR of MLP, DBN, and SAE are significantly greater than those of RNN, LSTM, and GRU. There are no significant differences among the AR of MLP, DBN, and SAE. There are no significant differences among the AR of RNN, LSTM, and GRU.

(2) Through the hypothesis test analysis of H2a and H2b, we can obtain p value<2.2e-16. So, there are statistically significant differences between the PR of all trading algorithms. Therefore, we need to make multiple comparative analysis further, as shown in Table 6. The number in the table is a p value of any two algorithms of Nemenyi test. From Tables 6 and 4, we can see that the PR of MLP, DBN, and SAE are significantly greater than that of other trading algorithms. The PR of LSTM is not significantly different from that of GRU and NB. The PR of GRU is significantly lower than that of all traditional ML algorithms. The PR of NB is significantly lower than that of other traditional ML algorithms.

(3) Through the hypothesis test analysis of H3a and H3b, we can obtain p value<2.2e-16. So, there are statistically significant differences between the RR of all trading algorithms Therefore, we need to make multiple comparative analysis further, as shown in Table 7. The number in the table is a p value of any two algorithms of Nemenyi test. From Tables 7 and 4, we can see that there is no significant difference among the RR of all DNN models, but the RR of any DNN model is significantly lower than that of all traditional ML models. The RR of NB is significantly lower than that of other traditional ML algorithms. The RR of CART is significantly lower than that of other traditional ML algorithms except for NB.

(4) Through the hypothesis test analysis of H4a and H4b, we can obtain p value<2.2e-16. So, there are statistically significant differences between the F1 of all trading algorithms. Therefore, we need to make multiple comparative analysis further, as shown in Table 8. The number in the table is a p value of any two algorithms of Nemenyi test. From Tables 8 and 4, we can see that there is no significant difference among the F1 of MLP, DBN, and SAE. The F1 of MLP, DBN, and SAE are significantly greater than that of RNN, LSTM, GRU, and NB, but are significantly smaller than that of RF, LR, SVM, and XGB. The F1 of GRU and LSTM have no significant difference, but they are significantly smaller than that of all traditional ML algorithms. The F1 of XGB is significantly greater than that of all other trading algorithms.

Table 7: Multiple comparison analysis between the RR of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
DBN	1.0000										
SAE	1.0000	1.0000									
RNN	1.0000	1.0000	1.0000								
LSTM	1.0000	1.0000	1.0000	1.0000							
GRU	0.9999	1.0000	0.9999	1.0000	1.0000						
CART	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**					
NB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**				
RF	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0485**	**0.0000**			
LR	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.0555		
SVM	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0197**	**0.0000**	**0.0000**	**0.0000**	
XGB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0010**	0.9958	**0.0000**

Table 8: Multiple comparison analysis between the F1 of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
DBN	0.9998										
SAE	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	**0.0000**								
LSTM	**0.0000**	**0.0000**	**0.0000**	0.0810							
GRU	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.3489						
CART	0.0861	**0.0061**	**0.0117**	**0.0000**	**0.0000**	**0.0000**					
NB	**0.0000**	**0.0000**	**0.0000**	0.4635	**0.0000**	**0.0000**	**0.0000**				
RF	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0078**	**0.0000**			
LR	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0173**	**0.0000**	1.0000		
SVM	**0.0007**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.9797	**0.0000**	0.3336	0.4825	
XGB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**

Table 9: Multiple comparison analysis between the AUC of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
DBN	1.0000										
SAE	1.0000	1.0000									
RNN	1.0000	1.0000	1.0000								
LSTM	1.0000	1.0000	1.0000	0.9999							
GRU	1.0000	1.0000	1.0000	1.0000	0.9975						
CART	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**					
NB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**				
RF	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0270**	**0.0000**			
LR	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.5428		
SVM	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.3125	**0.0000**	**0.0000**	**0.0000**	
XGB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0002**	0.3954	**0.0000**

(5) Through the hypothesis test analysis of H5a and H5b, we can obtain p value<2.2e-16. So, there are statistically significant differences between the AUC of all trading algorithms. Therefore, we need to make multiple comparative analysis further, as shown in Table 9. The number in the table is a p value of any two algorithms of Nemenyi test. From Tables 9 and 4, we can see that there is no significant difference among the AUC of all DNN models. The AUC of all DNN models are significantly smaller than that of any traditional ML model.

(6) Through the hypothesis test analysis of H6a and H6b, we can obtain p value<2.2e-16. So, there are statistically significant differences between the WR of all trading algorithms. Therefore, we need to make multiple comparative analysis further, as shown in Table 10. The number in the table is p value of any two algorithms of Nemenyi test. From Tables 4

Table 10: Multiple comparison analysis between the WR of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	Index	BAH	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
BAH	**0.0000**												
MLP	**0.0000**	**0.0000**											
DBN	**0.0000**	**0.0000**	1.0000										
SAE	**0.0000**	**0.0000**	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**								
LSTM	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.9974							
GRU	**0.0011**	**0.0000**	**0.0001**	**0.0000**	**0.0000**	1.0000	0.9961						
CART	**0.0000**	0.9998	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**					
NB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0031**	**0.0000**	**0.0038**	**0.0000**				
RF	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0118**	**0.0001**	**0.0140**	**0.0000**	1.0000			
LR	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	1.0000	0.8508	1.0000	**0.0000**	**0.0432**	0.1177		
SVM	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	1.0000	1.0000	1.0000	**0.0000**	**0.0001**	**0.0006**	0.9780	
XGB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.2660	**0.0084**	0.2927	**0.0000**	0.9831	0.9989	0.7627	**0.0376**

Table 11: Multiple comparison analysis between the ARR of any two trading strategies. The p value of the two trading strategies with significant difference is in boldface.

	Index	BAH	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
BAH	**0.0000**												
MLP	**0.0000**	**0.0000**											
DBN	**0.0000**	**0.0000**	1.0000										
SAE	**0.0000**	**0.0000**	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	**0.0001**	**0.0006**	**0.0001**								
LSTM	**0.0000**	**0.0000**	**0.0000**	**0.0002**	**0.0000**	1.0000							
GRU	**0.0000**	**0.0000**	**0.0001**	**0.0008**	**0.0001**	1.0000	1.0000						
CART	**0.0000**	**0.0000**	1.0000	1.0000	1.0000	**0.0001**	**0.0000**	**0.0001**					
NB	**0.0000**	**0.0000**	**0.0021**	**0.0094**	**0.0022**	1.0000	0.9998	1.0000	**0.0018**				
RF	**0.0000**	**0.0000**	0.7978	0.9524	0.8036	0.1685	0.0874	0.1962	0.7745	0.5861			
LR	**0.0000**	**0.0000**	**0.0002**	**0.0012**	**0.0002**	1.0000	1.0000	1.0000	**0.0002**	1.0000	0.2408		
SVM	**0.0000**	**0.0000**	0.2375	0.4806	0.2427	0.7029	0.5214	0.7457	0.2178	0.9778	0.9999	0.8015	
XGB	**0.0000**	**0.0000**	0.0674	0.1856	0.0694	0.9423	0.8466	0.9576	0.0600	0.9996	0.9905	0.9739	1.0000

Table 12: Multiple comparison analysis between the ASR of any two trading strategies. The p value of the two trading strategies with significant difference is in boldface.

	Index	BAH	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
BAH	0.9667												
MLP	**0.0000**	**0.0000**											
DBN	**0.0000**	**0.0000**	1.0000										
SAE	**0.0000**	**0.0000**	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	0.8763	0.7617	0.8998								
LSTM	**0.0000**	**0.0000**	0.9922	0.9701	0.9949	1.0000							
GRU	**0.0000**	**0.0000**	0.6124	0.4563	0.6537	1.0000	0.9996						
CART	**0.0000**	**0.0000**	**0.0002**	**0.0005**	**0.0002**	**0.0000**	**0.0000**	**0.0000**					
NB	**0.0000**	**0.0000**	**0.0467**	**0.0233**	0.0557	0.9529	0.7037	0.9971	**0.0000**				
RF	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0291**	**0.0042**	0.1062	**0.0000**	0.8010			
LR	**0.0000**	**0.0000**	0.7506	0.6025	0.7859	1.0000	1.0000	1.0000	**0.0000**	0.9872	0.0602		
SVM	**0.0000**	**0.0000**	0.1759	0.1020	0.2010	0.9982	0.9399	1.0000	**0.0000**	1.0000	0.4671	0.9998	
XGB	**0.0000**	**0.0000**	**0.0099**	**0.0044**	**0.0122**	0.7548	0.3776	0.9470	**0.0000**	1.0000	0.9681	0.8791	0.9997

and 10, we can see that the WR of MLP, DBN, and SAE have no significant difference, but they are significantly higher than that of BAH and benchmark index, and significantly lower than that of other trading algorithms. The WR of RNN, LSTM, and GRU have no significant difference, but they are significantly higher than that of CART and significantly lower than that of NB and RF. The WR of LR is not significantly different from that of RF, SVM, and XGB.

(7) Through the analysis of the hypothesis test of H7a and H7b, we obtain p value<2.2e-16. Therefore, there are significant differences between the ARR of all trading strategies including the benchmark index and BAH. We need to do further multiple comparative analysis, as shown in Table 11. From Tables 4 and 11, we can see that the ARR of the benchmark index and BAH are significantly lower than that of all ML algorithms. The ARR of MLP, DBN, and SAE are significantly greater than that of RNN, LSTM, GRU, NB, and LR, but not significantly different from that of CART, RF, SVM, and XGB; there is no significant difference between the ARR of MLP, DBN, and SAE. The ARR of RNN, LSTM, and GRU are significantly less than that of CART, but they are not significantly different from that of other traditional ML algorithms. In all traditional ML algorithms, the ARR of CART is significantly greater than that of NB and LR, but, otherwise, there is no significant difference between ARR of any other two algorithms.

(8) Through the hypothesis test analysis of H8a and H8b, we obtain p value<2.2e-16. Therefore, there are significant differences between ASR of all trading strategies including the benchmark index and BAH. The results of our multiple comparative analysis are shown in Table 12. From Tables 4 and 12, we can see that the ASR of the benchmark index and BAH are significantly smaller than that of all ML algorithms. The ASR of MLP and DBN are significantly greater than that of CART and are significantly smaller than that of NB, RF, and XGB, but there is no significant difference between MLP, DBN, and other algorithms. The ASR of SAE is significantly greater than that of CART and significantly less than that of RF and XGB, but there is no significant difference between SAE and other algorithms. The ASR of RNN and LSTM

Table 13: Multiple comparison analysis between the MDD of any two trading strategies. The p value of the two trading strategies with significant difference is in boldface.

	Index	BAH	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
BAH	**0.0000**												
MLP	**0.0000**	**0.0052**											
DBN	**0.0000**	**0.0031**	1.0000										
SAE	**0.0000**	**0.0012**	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	0.1645	0.2243	0.3556								
LSTM	**0.0000**	**0.0000**	0.6236	0.7173	0.8511	1.0000							
GRU	**0.0000**	**0.0000**	**0.0245**	**0.0381**	0.0760	1.0000	0.9860						
CART	**0.0000**	**0.0000**	0.1496	0.2057	0.3309	1.0000	1.0000	1.0000					
NB	**0.0000**	**0.0000**	0.0786	0.1136	0.1999	1.0000	0.9994	1.0000	1.0000				
RF	**0.0000**	**0.0000**	**0.0002**	**0.0004**	**0.0012**	0.8964	0.4248	0.9980	0.9109	0.9713			
LR	**0.0000**	**0.0000**	0.5451	0.6428	0.7935	1.0000	1.0000	0.9933	1.0000	0.9998	0.5015		
SVM	**0.0000**	**0.0000**	0.2433	0.3194	0.4734	1.0000	1.0000	0.9999	1.0000	1.0000	0.8155	1.0000	
XGB	**0.0000**	**0.0000**	**0.0103**	**0.0167**	**0.0360**	0.9998	0.9462	1.0000	0.9999	1.0000	0.9998	0.9685	0.9989

Table 14: Trading performance of different trading strategies in CSICS. Best performance of all trading strategies is in boldface.

	Index	BAH	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM	XGB
AR	—	—	**0.5175**	0.5167	0.5163	0.5030	0.4993	0.4993	0.5052	0.5084	0.5090	0.5084	0.5112	0.5087
PR	—	—	**0.7548**	0.7436	0.7439	0.5414	0.4964	0.4956	0.5022	0.5109	0.5128	0.4967	0.5695	0.5026
RR	—	—	0.5252	0.5250	0.5248	0.5234	0.5224	0.5223	0.5279	0.5307	0.5311	**0.5318**	0.5295	0.5315
F1	—	—	**0.6150**	0.6108	0.6108	0.5320	0.5086	0.5082	0.5143	0.5192	0.5214	0.5132	0.5483	0.5164
AUC	—	—	0.5027	0.5024	0.5020	0.5006	0.4995	0.4996	0.5049	0.5078	0.5082	**0.5086**	0.5074	0.5085
WR	0.5222	0.5090	0.5559	0.5565	0.5564	0.5681	0.5720	0.5717	0.5153	0.5317	0.5785	**0.5809**	0.5716	0.5803
ARR	0.0633	0.2224	0.5731	0.5704	0.5678	0.5248	0.5165	0.5113	0.5534	**0.6125**	0.4842	0.5095	0.5004	0.4938
ASR	0.2625	0.4612	1.4031	1.4006	1.3935	1.4880	1.5422	1.5505	1.2232	1.1122	1.4379	**1.5582**	1.4231	1.4698
MDD	**0.4808**	0.6697	0.6082	0.6086	0.6130	0.5648	0.5456	0.5429	0.5694	0.7469	0.5695	0.5410	0.5775	0.5632

are significantly greater than that of CART and significantly less than that of RF, but there is no significant difference between RNN, LSTM, and other algorithms. The ASR of GRU is significantly greater than that of CART, but there is no significant difference between GRU and other traditional ML algorithms. In all traditional ML algorithms, the ASR of all algorithms are significantly greater than that of CART, but otherwise, there is no significant difference between ASR of any other two algorithms.

(9) Through the hypothesis test analysis of H9a and H9b, we obtain p value<2.2e-16. Therefore, there are significant differences between MDD of trading strategies including the benchmark index and the BAH. The results of multiple comparative analysis are shown in Table 13. From Tables 4 and 13, we can see that MDD of any ML algorithm is significantly greater than that of the benchmark index but significantly smaller than that of BAH strategy. The MDD of MLP and DBN are significantly smaller than those of GRU, RF, and XGB, but there is no significant difference between MLP, DBN, and other algorithms. The MDD of SAE is significantly smaller than that of XGB, but there is no significant difference between SAE and other algorithms. Otherwise, there is no significant difference between MDD of any other two algorithms.

In a word, the traditional ML algorithms such as NB, RF, and XGB have good performance in most directional evaluation indicators such as AR, PR, and F1. The DNN algorithms such as MLP have good performance in PR and ARR. In traditional ML algorithms, the ARR of CART, RF, SVM, and XGB are not significantly different from that of MLP, DBN, and SAE; the ARR of CART is significantly greater than that of LSTM, GRU, and RNN, but otherwise the ARR of all traditional ML algorithms are not significantly worse than that of LSTM, GRU, and RNN. The ASR of all traditional ML algorithms except CART are not significantly worse than that of the six DNN models; even the ASR of NB, RF, and XGB are significantly greater than that of some DNN algorithms. The MDD of RF and XGB are significantly less that of MLP, DBN, and SAE; the MDD of all traditional ML algorithms are not significantly different from that of LSTM, GRU, and RNN. The ARR and ASR of all ML algorithms are significantly greater than that of BAH and the benchmark index; the MDD of any ML algorithm is significantly greater than that of the benchmark index, but significantly less than that of BAH strategy.

5.3. Comparative Analysis of Performance of Different Trading Strategies in CSICS. The analysis methods of this part are similar to Section 5.2. From Table 14, we can see that the AR, PR, and F1 of MLP are the greatest in all trading algorithms. The RR, AUC, WR, and ASR of LR are the greatest in all trading algorithms, respectively. The ARR of NB is the

Table 15: Multiple comparison analysis between the AR of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
DBN	1.0000										
SAE	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	**0.0000**								
LSTM	**0.0000**	**0.0000**	**0.0000**	0.1857							
GRU	**0.0000**	**0.0000**	**0.0000**	0.4439	1.0000						
CART	**0.0000**	**0.0000**	**0.0000**	0.9765	**0.0024**	**0.0131**					
NB	**0.0000**	**0.0001**	**0.0002**	**0.0022**	**0.0000**	**0.0000**	0.1810				
RF	**0.0000**	**0.0002**	**0.0005**	**0.0007**	**0.0000**	**0.0000**	0.0941	1.0000			
LR	**0.0000**	**0.0000**	**0.0000**	**0.0076**	**0.0000**	**0.0000**	0.3454	1.0000	1.0000		
SVM	**0.0217**	0.0766	0.1309	**0.0000**	**0.0000**	**0.0000**	**0.0003**	0.8314	0.9352	0.6360	
XGB	**0.0000**	**0.0001**	**0.0001**	**0.0025**	**0.0000**	**0.0000**	0.1930	1.0000	1.0000	1.0000	0.8168

Table 16: Multiple comparison analysis between the PR of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
DBN	1.0000										
SAE	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	**0.0000**								
LSTM	**0.0000**	**0.0000**	**0.0000**	**0.0000**							
GRU	**0.0000**	**0.0000**	**0.0000**	**0.0000**	1.0000						
CART	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.9906	0.9781					
NB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.1716	0.1234	0.8940				
RF	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0319**	**0.0205**	0.5271	1.0000			
LR	**0.0000**	**0.0000**	**0.0000**	**0.0000**	1.0000	1.0000	0.9951	0.2099	0.0422		
SVM	**0.0000**	**0.0000**	**0.0000**	0.1157	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	
XGB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.9922	0.9811	1.0000	0.8836	0.5086	0.9960	**0.0000**

highest in all trading strategies. The MDD of CSI 300 index (benchmark index) is the smallest in all trading strategies. The WR, ARR, and ASR of all ML algorithms are greater than those of the benchmark index and BAH strategy.

(1) Through the hypothesis test analysis of H1a and H1b, we can obtain p value<2.2e-16. Therefore, there are significant differences between the AR of all trading algorithms. Therefore, we need to do further multiple comparative analysis and the results are shown in Table 15. The number in the table is a p value of any two algorithms of Nemenyi test. From Tables 14 and 15, we can see that the AR of MLP, DBN, and SAE have no significant difference, but they are significantly greater than that of all other trading algorithms except for SVM. The AR of GRU is significantly smaller than that of all traditional ML algorithms. There is no significant difference between the AR of any two traditional ML algorithms except for CART and SVM.

(2) Through the hypothesis test analysis of H2a and H2b, we can obtain p value<2.2e-16. Therefore, there are significant differences between the PR of all trading algorithms. Therefore, we need to do further multiple comparative analysis and the results are shown in Table 16. The number in the table is a p value of any two algorithms of Nemenyi test. From Tables 14 and 16, we can see that the PR of MLP, DBN, and SAE are significantly greater than that of all other trading algorithms, and the PR of MLP, DBN, and SAE have no significant difference. The PR of SVM is significantly greater than that of all other traditional ML algorithms which have no significant difference between any two algorithms except for SVM. The PR of RNN is significantly greater than that of all traditional ML algorithms except for SVM. The PR of GRU and LSTM are not significantly different from that of all traditional ML algorithms except for SVM and LR.

(3) Through the hypothesis test analysis of H3a and H3b, we can obtain p value<2.2e-16. Therefore, there are significant differences between the RR of all trading algorithms. Therefore, we need to do further multiple comparative analysis and the results are shown in Table 17. The number in the table is a p value of any two algorithms of Nemenyi test. From Tables 14 and 17, we can see that the RR of all DNN models are not significantly different. There is no significant difference among the RR of all traditional ML algorithms. The RR of RNN, GRU, and LSTM are significantly smaller than that of any traditional ML algorithm except for CART.

(4) Through the hypothesis test analysis of H4a and H4b, we can obtain p value<2.2e-16. Therefore, there are significant differences between the F1 of all trading algorithms. Therefore, we need to do further multiple comparative analysis and

Table 17: Multiple comparison analysis between the RR of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
DBN	1.0000										
SAE	1.0000	1.0000									
RNN	0.9996	0.9996	1.0000								
LSTM	0.9309	0.9314	0.9781	0.9999							
GRU	0.9660	0.9663	0.9916	1.0000	1.0000						
CART	0.9744	0.9742	0.9225	0.5809	0.1509	0.2138					
NB	0.1093	0.1088	0.0574	**0.0075**	**0.0004**	**0.0007**	0.8861				
RF	0.0537	0.0534	**0.0260**	**0.0028**	**0.0001**	**0.0002**	0.7544	1.0000			
LR	**0.0330**	**0.0328**	**0.0152**	**0.0015**	**0.0001**	**0.0001**	0.6498	1.0000	1.0000		
SVM	0.3444	0.3434	0.2170	**0.0434**	**0.0033**	**0.0059**	0.9920	1.0000	0.9998	0.9991	
XGB	**0.0193**	**0.0192**	**0.0085**	**0.0007**	**0.0000**	**0.0000**	0.5344	1.0000	1.0000	1.0000	0.9960

Table 18: Multiple comparison analysis between the F1 of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
DBN	1.0000										
SAE	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	**0.0000**								
LSTM	**0.0000**	**0.0000**	**0.0000**	**0.0000**							
GRU	**0.0000**	**0.0000**	**0.0000**	**0.0000**	1.0000						
CART	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.7211	0.6670					
NB	**0.0000**	**0.0000**	**0.0000**	**0.0136**	**0.0132**	**0.0099**	0.8664				
RF	**0.0000**	**0.0000**	**0.0000**	0.0786	**0.0016**	**0.0011**	0.5162	1.0000			
LR	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.9440	0.9208	1.0000	0.5675	0.2181		
SVM	**0.0000**	**0.0000**	**0.0000**	**0.0178**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	
XGB	**0.0000**	**0.0000**	**0.0000**	**0.0001**	0.3138	0.2679	1.0000	0.9937	0.8849	0.9964	**0.0000**

Table 19: Multiple comparison analysis between the AUC of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
DBN	1.0000										
SAE	0.9999	1.0000									
RNN	0.9945	0.9985	1.0000								
LSTM	0.5273	0.6382	0.9259	0.9937							
GRU	0.8448	0.9102	0.9958	1.0000	1.0000						
CART	0.6921	0.5835	0.2356	0.0801	**0.0014**	**0.0096**					
NB	**0.0002**	**0.0001**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.2616				
RF	**0.0001**	**0.0001**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.2002	1.0000			
LR	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.0930	1.0000	1.0000		
SVM	**0.0027**	**0.0014**	**0.0001**	**0.0000**	**0.0000**	**0.0000**	0.6454	1.0000	0.9999	0.9980	
XGB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.1257	1.0000	1.0000	1.0000	0.9993

the results are shown in Table 18. The number in the table is a p value of any two algorithms of Nemenyi test. From Tables 14 and 18, we can see that the F1 of MLP, DBN, and SAE have no significant difference, but they are significantly greater than that of all other trading algorithms. There is no significant difference among traditional ML algorithms except SVM, and the F1 of SVM is significantly greater than that of all other traditional ML algorithms.

(5) Through the hypothesis test analysis of H5a and H5b, we can obtain p value<2.2e-16. Therefore, there are significant differences between the AUC of all trading algorithms. Therefore, we need to do further multiple comparative analysis and the results are shown in Table 19. The number in the table is a p value of any two algorithms of Nemenyi test. From Tables 14 and 19, we can see that the AUC of all DNN models have no significant difference. There is no significant difference

Table 20: Multiple comparison analysis between the WR of any two trading algorithms. The p value of the two trading strategies with significant difference is in boldface.

	Index	BAH	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
BAH	0.4117												
MLP	**0.0000**	**0.0000**											
DBN	**0.0000**	**0.0000**	1.0000										
SAE	**0.0000**	**0.0000**	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	**0.0002**	**0.0006**	**0.0000**								
LSTM	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.9772							
GRU	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.9911	1.0000						
CART	0.9931	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**					
NB	**0.0031**	**0.0000**	**0.0001**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**				
RF	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0205**	0.6437	0.5358	**0.0000**	**0.0000**			
LR	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0010**	0.1611	0.1105	**0.0000**	**0.0000**	1.0000		
SVM	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.9914	1.0000	1.0000	**0.0000**	**0.0000**	0.5322	0.1090	
XGB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**

Table 21: Multiple comparison analysis between the ARR of any two trading strategies. The p value of the two trading strategies with significant difference is in boldface.

	Index	*BAH*	*MLP*	*DBN*	*SAE*	*RNN*	*LSTM*	GRU	CART	NB	RF	LR	SVM
BAH	**0.0007**												
MLP	**0.0000**	**0.0000**											
DBN	**0.0000**	**0.0000**	1.0000										
SAE	**0.0000**	**0.0000**	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	0.4790	0.6355	0.7182								
LSTM	**0.0000**	**0.0000**	0.2512	0.3806	0.4630	1.0000							
GRU	**0.0000**	**0.0000**	0.2235	0.3454	0.4249	1.0000	1.0000						
CART	**0.0000**	**0.0000**	0.8301	0.9217	0.9542	1.0000	0.9999	0.9998					
NB	**0.0000**	**0.0000**	1.0000	1.0000	1.0000	0.2920	0.1295	0.1125	0.6517				
RF	**0.0000**	**0.0000**	**0.0020**	**0.0048**	**0.0076**	0.8705	0.9735	0.9806	0.5393	**0.0006**			
LR	**0.0000**	**0.0000**	0.2058	0.3222	0.3995	1.0000	1.0000	1.0000	0.9996	0.1019	0.9845		
SVM	**0.0000**	**0.0000**	1.0000	0.0803	0.1114	0.9993	1.0000	1.0000	0.9659	**0.0165**	0.9999	1.0000	
XGB	**0.0000**	**0.0000**	1.0000	**0.0333**	**0.0484**	0.9916	0.9997	0.9998	0.8789	**0.0057**	1.0000	0.9999	1.0000

between the AUC of all traditional ML algorithms. The AUC of all traditional ML algorithms except for CART are significantly greater than that of any DNN model. There is no significant difference among the AUC of MLP, SAE, DBN, RNN, and CART.

(6) Through the hypothesis test analysis of H6a and H6b, we can obtain p value<2.2e-16. Therefore, there are significant differences between the WR of all trading algorithms. Therefore, we need to do further multiple comparative analysis and the results are shown in Table 20. The number in the table is a p value of any two algorithms of Nemenyi test. From Tables 14 and 20, we can see that the WR of BAH and benchmark index have no significant difference, but they are significantly smaller than that of any ML algorithm. The WR of MLP, DBN, and SAE are significantly smaller than that of the other trading algorithms, but there is no significant difference between the WR of MLP, DBN, and SAE. The WR of LSTM and GRU have no significant difference, but they are significantly smaller than that of XGB and significantly greater than that of CART and NB. In traditional ML models, the WR of NB and CART are significantly smaller than that of other algorithms. The WR of XGB is significantly greater than that of all other ML algorithms.

(7) Through the analysis of the hypothesis test of H7a and H7b, we obtain p value<2.2e-16.

Therefore, there are significant differences between the ARR of all trading strategies including the benchmark index and BAH strategy. Therefore, we need to do further multiple comparative analysis and the results are shown in Table 21. From Tables 14 and 21, we can see that ARR of the benchmark index and BAH are significantly smaller than that of all trading algorithms. The ARR of MLP is significantly higher than that of RF, but there is no significant difference between MLP and other algorithms. The ARR of SAE and DBN are significantly higher than that of RF and XGB, but they are not significantly different from ARR of other algorithms. The ARR of NB is significantly higher than that of RF, SVM, and XGB. But, otherwise, there is no significant difference

Table 22: Multiple comparison analysis between the ASR of any two trading strategies. The p value of the two trading strategies with significant difference is in boldface.

	Index	BAH	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
BAH	0.8877												
MLP	**0.0000**	**0.0000**											
DBN	**0.0000**	**0.0000**	1.0000										
SAE	**0.0000**	**0.0000**	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	0.9099	0.8862	0.8114								
LSTM	**0.0000**	**0.0000**	0.3460	0.3080	0.2239	0.9999							
GRU	**0.0000**	**0.0000**	0.2132	0.1853	0.1270	0.9981	1.0000						
CART	**0.0000**	**0.0000**	**0.0158**	**0.0195**	**0.0327**	**0.0000**	**0.0000**	**0.0000**					
NB	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.7298				
RF	**0.0000**	**0.0000**	1.0000	1.0000	1.0000	0.9968	0.7444	0.5789	**0.0018**	**0.0000**			
LR	**0.0000**	**0.0000**	0.1181	0.1003	0.0650	0.9879	1.0000	1.0000	**0.0000**	**0.0000**	0.4044		
SVM	**0.0000**	**0.0000**	1.0000	1.0000	1.0000	0.9849	0.5952	0.4238	**0.0042**	**0.0000**	1.0000	0.2704	
XGB	**0.0000**	**0.0000**	0.9937	0.9902	0.9746	1.0000	0.9878	0.9532	**0.0001**	**0.0000**	1.0000	0.8723	0.9998

Table 23: Multiple comparison analysis between the MDD of any two trading strategies. The p value of the two trading strategies with significant difference is in boldface.

	Index	BAH	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM
BAH	**0.0000**												
MLP	**0.0000**	**0.0006**											
DBN	**0.0000**	**0.0004**	1.0000										
SAE	**0.0000**	**0.0023**	1.0000	1.0000									
RNN	**0.0000**	**0.0000**	**0.0320**	**0.0421**	**0.0111**								
LSTM	**0.0000**	**0.0000**	**0.0002**	**0.0003**	**0.0000**	0.9947							
GRU	**0.0000**	**0.0000**	**0.0001**	**0.0001**	**0.0000**	0.9767	1.0000						
CART	**0.0000**	**0.0000**	0.1238	0.1538	0.0521	1.0000	0.9241	0.8305					
NB	**0.0000**	0.1875	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**				
RF	**0.0000**	**0.0000**	0.1180	0.1469	0.0493	1.0000	0.9298	0.8401	1.0000	**0.0000**			
LR	**0.0000**	**0.0000**	**0.0001**	**0.0002**	**0.0000**	0.9881	1.0000	1.0000	0.8821	**0.0000**	0.8898		
SVM	**0.0000**	**0.0000**	0.3285	0.3839	0.1701	0.9999	0.7011	0.5424	1.0000	**0.0000**	1.0000	0.6216	
XGB	**0.0000**	**0.0000**	**0.0308**	**0.0405**	**0.0106**	1.0000	0.9951	0.9783	1.0000	**0.0000**	1.0000	0.9890	0.9998

between any other two algorithms. Therefore, the ARR of most traditional ML models are not significantly worse than that of the best DNN model.

(8) Through the hypothesis test analysis of H8a and H8b, we obtain p value<2.2e-16. Therefore, There are significant differences between ASR of all trading strategies including the benchmark index and BAH strategy. The results of multiple comparative analysis are shown in Table 22. From Tables 14 and 22, we can see that the ASR of the benchmark index and BAH are significantly smaller than that of all trading algorithms. The ASR of all ML algorithms are significantly higher than that of CART and NB, but there is no significant difference between the ASR of CART and NB. Beyond that, there is no significant difference between any other two algorithms. Therefore, the ASR of all traditional ML models except NB and CART are not significantly worse than that of any DNN model.

(9) Through the hypothesis test analysis of H9a and H9b, we obtain p value<2.2e-16. Therefore, there are significant differences between the MDD of these trading strategies including the benchmark index and the BAH strategy. The results of multiple comparative analysis are shown in Table 23. From Tables 14 and 23, we can see that the MDD of the benchmark index is significantly smaller than that of other trading strategies including BAH strategy. The MDD of BAH is significantly greater than that of all trading algorithms except NB. The MDD of MLP, DBN, and SAE are significantly lower than that of NB, but significantly higher than that of RNN, LSTM, GRU, LR, and XGB. The MDD of NB is significantly greater than that of all other trading algorithms. Beyond that, there is no significant difference between any other two algorithms. Therefore, all ML algorithms expect NB, especially LSTM, RNN, GRU, LR, and XGB, can play a role in controlling trading risk.

In a word, some DNN models such as MLP, DBN, and SAE have good performance in AR, PR, and F1; traditional ML algorithms such as LR and XGB have good performance in AUC and WR. The ARR of some traditional ML algorithms such as CART, NB, LR, and SVM are not significantly different from that of the six DNN models. The ASR of the

six DNN algorithms are not significantly different from all traditional ML models except NB and CART. The MDD of LR and XGB are significantly smaller than those of MLP, DBN, and SAE, and are not significantly different from that of LSTM, GRU, and RNN. The ARR and ASR of all ML algorithms are significantly greater than those of BAH and benchmark index; the MDD of all ML algorithms are significantly smaller than that of the benchmark index but significantly greater than that of BAH strategy.

From the above analysis and evaluation, we can see that the directional evaluation indicators of some DNN models are very competitive in CSICS, while the indicators of some traditional ML algorithms have excellent performance in SPICS. Whether in SPICS or CSICS, the ARR and ASR of all ML algorithms are significantly greater than that of the benchmark index and BAH strategy, respectively. In all ML algorithms, there are always some traditional ML algorithms which are not significantly worse than the best DNN model for any performance evaluation indicator (ARR, ASR, and MDD). Therefore, if we do not consider transaction cost and other factors affecting trading, performance of DNN models are alternative but not the best choice when they are applied to stock trading.

In the same period, the ARR of any ML algorithm in CSICS is significantly greater than that of the same algorithm in SPICS (p value <0.001 in the Nemenyi test). Meanwhile, the MDD of any ML algorithm in CSICS is significantly greater than that of the same algorithm in SPICS (p value <0.001 in the Nemenyi test). The results show that the quantitative trading algorithms can more easily obtain excess returns in the Chinese A-share market, but the volatility risk of trading in Chinese A-share market is significantly higher than that of the US stock market in the past 8 years.

6. The Impact of Transaction Cost on Performance of ML Algorithms

Trading cost can affect the profitability of a stock trading strategy. Transaction cost that can be ignored in long-term strategies is significantly magnified in daily trading. However, many algorithmic trading studies assume that transaction cost does not exist ([10, 17], etc.). In practice, frictions such as transaction cost can distort the market from the perfect model in textbooks. The cost known prior to trading activity is referred to as transparent such as commissions, exchange fees, and taxes. The costs that has to be estimated are known as implicit, including comprise bid-ask spread, latency or slippage, and related market impact. This section focuses on the transparent and implicit cost and how do they affect trading performance in daily trading.

6.1. Experimental Settings and Backtesting Algorithm. In this part, the transparent transaction cost is calculated by a certain percentage of transaction turnover for convenience; the implicit transaction cost is very complicated in calculation, and it is necessary to make a reasonable estimate for the random changes of market environment and stock prices. Therefore, we only discuss the impact of slippage on trading performance.

The transaction cost structures of American stocks are similar to that of Chinese A-shares. We assume that transparent transaction cost is calculated by a percentage of turnover such as less than 0.5% [40, 41] and 0.2% and 0.5% in the literature [42]. It is different for the estimation of slippage.

In some quantitative trading simulation software such as JoinQuant [43] and Abuquant [44], the slippage is set to 0.02. The transparent transaction cost and implicit transaction cost are charged in both directions when buying and selling. It is worth noting that the transparent transaction cost varies with the different brokers, while the implicit transaction cost is related to market liquidity, market information, network status, trading software, etc.

We set slippages s = {s0=0, s1=0.01, s2=0.02, s3=0.03, s4=0.04}; the transparent transaction cost c = {c0=0, c1=0.001, c2=0.002, c3=0.003, c4=0.004, c5=0.005}. For different {s, c} combinations, we study the impact of different transaction cost structures on trading performance. We assume that buying and selling positions are one unit, so the turnover is the corresponding stock price. When buying stocks, we not only need to pay a certain percentage cost of the purchase price, but also need to pay an uncertain slippage cost. That is, we need to pay a higher price than the real-time price P_{t-1} when we are buying. But, when selling stocks, we not only need to pay a certain percentage cost of the selling price, but also to pay an uncertain slippage cost. Generally speaking, we need to sell at a price lower than the real-time price P_t. It is worth noting that our trading strategy is self-financing. If ML algorithms predict the continuous occurrence of buying signals or selling signals, i.e., $|TradeSignal_t - TradeSignal_{t-1}| = 0$, we will continue to hold or do nothing, so the transaction cost at this time is 0. when $|TradeSignal_t - TradeSignal_{t-1}| = 1$, it is indicated that the position may be changed from holding to selling or from empty position to buying. At this time, we would pay transaction cost due to the trading operation. Finally, we get a real yield is

$$
\begin{aligned}
Ret_t &\leq \frac{ClosePrice_t - ClosePrice_{t-1}}{ClosePrice_{t-1}}.\\
P_t &= ClosePrice_t \\
&\quad * \left(1 - c * \left|TradeSignal_t - TradeSignal_{t-1}\right|\right) \\
&\quad - s * \left|TradeSignal_t - TradeSignal_{t-1}\right| \\
P_{t-1} &= ClosePrice_{t-1} \\
&\quad * \left(1 + c * \left|TradeSignal_{t-1} - TradeSignal_{t-2}\right|\right) \\
&\quad + s * \left|TradeSignal_{t-1} - TradeSignal_{t-2}\right| \\
Ret_t &= \frac{P_t - P_{t-1}}{P_{t-1}}
\end{aligned}
\tag{5}
$$

where $ClosePrice_t$ denotes the t-th closing price, $TradeSignal_t$ denotes the t-th trading signal, P_t denotes the t-th executing price, and Ret_t denotes the t-th return rate.

We propose a backtesting algorithm with transaction cost based on the above analysis, as is shown in Algorithm 3.

```
Input: TS #TS is trading signals of a stock.
       s # s is slippage.
       c # c is transparent transaction cost.
Output: WR, ARR, ASR, MDD
(1)  N=length of Stock Code List #424 SPICS, and 185 CSICS.
(2)  WR=NULL; ARR=NULL; ASR=NULL; MDD=NULL
(3)  for (i in 1: N) {
(4)      Stock Data=Stock Code List[i]
(5)      ClosePrice_t=Stock_Data ["Closing Price"]
(6)      P_t =ClosePrice_t *(1-c*abs(TS_t-TS_t-1)) - s*abs(TS_t-TS_t-1)
(7)      P_t-1 =ClosePrice_t*(1+c*abs(TS_t-TS_t-1)) + s*abs(TS_t-TS_t-1)
(8)      Ret_t= (P_t- P_t-1)/ P_t # Ret is the return rate series.
(9)      TDRR=lag (TS)*Ret #TDRR is the daily return through trading.
(10)     WR[i]=sum (TDRR>0)/sum(TDRR≠0)
(11)     ARR[i]=Return.annualized (TDRR)
(12)     ASR[i]=SharpeRatio.annualized (TDRR)
(13)     MDD[i]=maxDrawDown (TDRR)
(14)     WR=c (WR, WR[i]);
(15)     ARR=c (ARR, ARR[i]);
(16)     ASR=c (ASR, ASR[i]);
(17)     MDD=c (MDD, MDD[i])
(18)      }
(19) return (WR, ARR, ASR, MDD)
```

Algorithm 3: Backtesting algorithm with transaction cost in R language.

6.2. Analysis of Impact of Transaction Cost on the Trading Performance of SPICS. Transaction cost is one of the most important factors affecting trading performance. In US stock trading, transparent transaction cost can be charged according to a fixed fee per order or month, or a floating fee based on the volume and turnover of each transaction. Sometimes, customers can also negotiate with broker to determine transaction cost. The transaction cost charged by different brokers varies greatly. Meanwhile, implicit transaction cost is not known beforehand and the estimations of them are very complex. Therefore, we assume that the percentage of turnover is the transparent transaction cost for ease of calculation. In the aspect of implicit transaction cost, we only consider the impact of slippage on trading performance.

(1) Analysis of Impact of Transaction Cost on WR. As can be seen from Table 24, WR is decreasing with the increase of transaction cost for any trading algorithm, which is intuitive. When the transaction cost is set to (s, c) = (0.04, 0.005), the WR of each algorithm is the lowest. Compared with setting (s, c) = (0, 0), the WR of MLP, DBN, SAE, RNN, LSTM, GRU, CART, NB, RF, LR, and SVM to XGB are reduced by 5.80%, 5.97%, 5.91%, 15.83%, 18.04%, 13.95%, 21.71%, 16.04%, 22.16%, 18.54%, 18.50%, and 25.97%, respectively. Therefore, MLP, DBN, and SAE are more tolerant to transaction cost. Generally speaking, the DNN models have stronger capacity to accommodate transaction cost than the traditional ML models. From the single trading algorithm such as MLP, if we do not consider slippage, i.e., s=0, the average WR of MLP is 0.5510 under transaction cost structures { (s0, c1), (s0, c2), (s0, c3), (s0, c4), (s0, c5) }; if we do not consider transparent transaction cost, i.e., c=0, the average WR of MLP is 0.5618 under transaction cost structures { (s1, c0), (s2, c0), (s3, c0), (s4, c0) }; so transparent transaction cost has greater impact than slippage. Through multiple comparative analysis, the WR under the transaction cost structure (s1, c0) is not significantly different from the WR without transaction cost for MLP, DBN, and SAE. The WR under all other transaction cost structures are significantly smaller than the WR without transaction cost. For all trading algorithms except for MLP, DBN, and SAE, the WR under the transaction cost structure { (s1, c0), (s2, c0) } are not significantly different from the WR without transaction cost; the WR under all other transaction cost structures are significantly smaller than the WR without transaction cost.

(2) Analysis of Impact of Transaction Cost on ARR. As can be seen from Table 25, ARR is decreasing with the increase of transaction cost for any trading algorithm. Undoubtedly, when the transaction cost is set to (s, c) = (0.04, 0.005), the ARR of each algorithm is the lowest. Compared with the settings without transaction cost, the ARR of MLP, DBN, and SAE reduce by 40.31%, 41.57%, and 40.93%, respectively, while the ARR of other trading algorithms decrease by more than 100% compared with those without transaction cost. Therefore, excessive transaction cost can lead to serious losses in accounts. For a general setting of s and c, i.e., (s, c) = (0.02, 0.003), ARR of MLP, DBN, and SAE decrease by 23.26%, 24.00%, and 23.61%, respectively, while the ARR of other algorithms decrease by more than 50% and that of CART and XGB decrease by more than 100%. Therefore, MLP, DBN, and SAE are more tolerant to high transaction cost. From single trading algorithm such as RNN, if we do not consider slippage, i.e., s=0, the average ARR of RNN is 0.1434 under

Table 24: The WR of SPICS for daily trading with different transaction cost. The result that there is no significant difference between performance without transaction cost and that with transaction cost is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM	XGB
(s0, c0)	0.5676	0.5680	0.5683	0.5843	0.5825	0.5844	0.5266	0.5930	0.5912	0.5859	0.5831	0.5891
(s0, c1)	0.5615	0.5618	0.5621	0.5669	0.5626	0.5694	0.5052	0.5752	0.5663	0.5654	0.5630	0.5599
(s0, c2)	0.5560	0.5560	0.5564	0.5507	0.5442	0.5553	0.4847	0.5585	0.5429	0.5463	0.5438	0.5324
(s0, c3)	0.5507	0.5506	0.5510	0.5353	0.5269	0.5418	0.4652	0.5424	0.5205	0.5278	0.5253	0.5061
(s0, c4)	0.5457	0.5454	0.5460	0.5206	0.5101	0.5289	0.4470	0.5277	0.4996	0.5105	0.5084	0.4818
(s0, c5)	0.5410	0.5407	0.5412	0.5071	0.4946	0.5170	0.4303	0.5138	0.4803	0.4947	0.4924	0.4594
(s1, c0)	**0.5649**	**0.5653**	**0.5656**	**0.5778**	**0.5751**	**0.5785**	**0.5190**	**0.5861**	**0.5824**	**0.5782**	**0.5759**	**0.5788**
(s1, c1)	0.5594	0.5597	0.5600	0.5619	0.5571	0.5648	0.4991	0.5700	0.5597	0.5595	0.5574	0.5521
(s1, c2)	0.5539	0.5539	0.5544	0.5458	0.5389	0.5508	0.4788	0.5534	0.5364	0.5406	0.5383	0.5249
(s1, c3)	0.5487	0.5486	0.5491	0.5305	0.5215	0.5375	0.4597	0.5375	0.5141	0.5223	0.5200	0.4988
(s1, c4)	0.5438	0.5436	0.5441	0.5163	0.5052	0.5249	0.4418	0.5231	0.4938	0.5054	0.5034	0.4750
(s1, c5)	0.5394	0.5390	0.5396	0.5032	0.4902	0.5132	0.4255	0.5097	0.4751	0.4900	0.4880	0.4532
(s2, c0)	0.5628	0.5631	0.5635	**0.5729**	**0.5697**	**0.5741**	**0.5134**	**0.5811**	**0.5758**	**0.5726**	**0.5704**	**0.5713**
(s2, c1)	0.5573	0.5574	0.5578	0.5568	0.5515	0.5602	0.4931	0.5647	0.5527	0.5535	0.5515	0.5442
(s2, c2)	0.5518	0.5518	0.5523	0.5408	0.5336	0.5463	0.4729	0.5482	0.5297	0.5349	0.5326	0.5172
(s2, c3)	0.5469	0.5467	0.5472	0.5260	0.5164	0.5334	0.4543	0.5330	0.5082	0.5171	0.5150	0.4918
(s2, c4)	0.5421	0.5417	0.5423	0.5120	0.5004	0.5209	0.4368	0.5187	0.4881	0.5004	0.4986	0.4685
(s2, c5)	0.5377	0.5373	0.5379	0.4993	0.4858	0.5096	0.4209	0.5055	0.4699	0.4855	0.4835	0.4473
(s3, c0)	0.5606	0.5609	0.5612	0.5678	0.5640	0.5694	0.5074	0.5759	0.5690	0.5668	0.5646	0.5635
(s3, c1)	0.5552	0.5553	0.5558	0.5518	0.5460	0.5556	0.4872	0.5595	0.5460	0.5478	0.5459	0.5365
(s3, c2)	0.5499	0.5498	0.5503	0.5362	0.5284	0.5422	0.4675	0.5434	0.5233	0.5295	0.5273	0.5099
(s3, c3)	0.5450	0.5448	0.5453	0.5216	0.5114	0.5292	0.4491	0.5284	0.5020	0.5120	0.5098	0.4849
(s3, c4)	0.5405	0.5401	0.5407	0.5080	0.4960	0.5172	0.4321	0.5146	0.4827	0.4959	0.4940	0.4622
(s3, c5)	0.5362	0.5357	0.5363	0.4955	0.4816	0.5063	0.4166	0.5017	0.4651	0.4815	0.4793	0.4417
(s4, c0)	0.5587	0.5589	0.5593	0.5630	0.5588	0.5652	0.5019	0.5710	0.5627	0.5613	0.5593	0.5562
(s4, c1)	0.5533	0.5533	0.5537	0.5470	0.5407	0.5513	0.4815	0.5544	0.5395	0.5424	0.5404	0.5290
(s4, c2)	0.5480	0.5479	0.5484	0.5316	0.5232	0.5380	0.4620	0.5386	0.5171	0.5242	0.5220	0.5026
(s4, c3)	0.5432	0.5429	0.5435	0.5172	0.5067	0.5253	0.4440	0.5241	0.4964	0.5070	0.5051	0.4782
(s4, c4)	0.5388	0.5384	0.5391	0.5041	0.4917	0.5137	0.4274	0.5106	0.4775	0.4914	0.4897	0.4564
(s4, c5)	0.5347	0.5341	0.5347	0.4918	0.4774	0.5029	0.4123	0.4979	0.4602	0.4773	0.4752	0.4361

Table 25: The ARR of SPICS for daily trading with different transaction cost. The result that there is no significant difference between performance without transaction cost and that with transaction cost is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM	XGB
(s0, c0)	0.3332	0.3296	0.3325	0.2943	0.2920	0.2934	0.3317	0.2975	0.3133	0.2942	0.3067	0.3041
(s0, c1)	**0.3128**	**0.3087**	**0.3118**	0.2439	0.2327	0.2521	0.2266	0.2496	0.2384	0.2336	0.2400	0.2137
(s0, c2)	0.2924	0.2879	0.2912	0.1936	0.1735	0.2108	0.1216	0.2017	0.1636	0.1731	0.1734	0.1234
(s0, c3)	0.2721	0.2672	0.2706	0.1433	0.1143	0.1697	0.0168	0.1539	0.0889	0.1126	0.1070	0.0333
(s0, c4)	0.2519	0.2464	0.2500	0.0931	0.0553	0.1285	-0.0878	0.1062	0.0143	0.0522	0.0406	-0.0567
(s0, c5)	0.2316	0.2257	0.2294	0.0430	-0.0037	0.0875	-0.1924	0.0586	-0.0602	-0.008	-0.0257	-0.1466
(s1, c0)	**0.3249**	**0.3212**	**0.3242**	**0.2778**	**0.2729**	**0.2794**	**0.2953**	**0.2818**	**0.2898**	**0.2747**	**0.2855**	**0.2762**
(s1, c1)	0.3046	0.3004	0.3035	0.2274	0.2136	0.2381	0.1902	0.2339	0.2149	0.2141	0.2188	0.1859
(s1, c2)	0.2842	0.2796	0.2829	0.1771	0.1544	0.1969	0.0853	0.1861	0.1401	0.1536	0.1523	0.0957
(s1, c3)	0.2639	0.2588	0.2623	0.1269	0.0953	0.1557	-0.0195	0.1383	0.0654	0.0931	0.0858	0.0056
(s1, c4)	0.2437	0.2381	0.2417	0.0767	0.0363	0.1146	-0.1241	0.0906	-0.0091	0.0328	0.0195	-0.0844
(s1, c5)	0.2234	0.2174	0.2211	0.0266	-0.0226	0.0735	-0.2285	0.0430	-0.0836	-0.0275	-0.0468	-0.1742
(s2, c0)	**0.3167**	**0.3129**	**0.3159**	**0.2613**	**0.2538**	**0.2654**	**0.2589**	**0.2661**	**0.2663**	**0.2551**	**0.2643**	**0.2484**
(s2, c1)	0.2964	0.2921	0.2952	0.2110	0.1946	0.2241	0.1539	0.2182	0.1914	0.1946	0.1977	0.1581
(s2, c2)	0.276	0.2713	0.2746	0.1607	0.1354	0.1829	0.0490	0.1704	0.1167	0.1341	0.1312	0.0679
(s2, c3)	0.2557	0.2505	0.2540	0.1104	0.0763	0.1418	-0.0557	0.1227	0.0420	0.0737	0.0647	-0.0221
(s2, c4)	0.2355	0.2298	0.2334	0.0603	0.0173	0.1007	-0.1603	0.0750	-0.0325	0.0134	-0.0016	-0.1120
(s2, c5)	0.2153	0.2091	0.2129	0.0102	-0.0416	0.0597	-0.2647	0.0274	-0.1069	-0.0469	-0.0678	-0.2018
(s3, c0)	**0.3085**	**0.3046**	**0.3076**	0.2448	0.2348	0.2514	0.2225	0.2504	0.2428	0.2356	0.2431	0.2206
(s3, c1)	0.2881	0.2838	0.2869	0.1945	0.1756	0.2102	0.1175	0.2026	0.1680	0.1751	0.1765	0.1304
(s3, c2)	0.2678	0.2630	0.2663	0.1442	0.1164	0.1690	0.0127	0.1548	0.0933	0.1146	0.1100	0.0402
(s3, c3)	0.2476	0.2422	0.2457	0.0940	0.0574	0.1279	-0.0919	0.1071	0.0187	0.0543	0.0436	-0.0498
(s3, c4)	0.2273	0.2215	0.2252	0.0439	-0.0016	0.0868	-0.1964	0.0595	-0.0558	-0.006	-0.0227	-0.1397
(s3, c5)	0.2071	0.2008	0.2047	-0.0062	-0.0605	0.0458	-0.3008	0.0119	-0.1302	-0.0662	-0.0889	-0.2294
(s4, c0)	0.3003	0.2962	0.2993	0.2284	0.2157	0.2374	0.1862	0.2348	0.2193	0.2161	0.2219	0.1929
(s4, c1)	0.2799	0.2754	0.2786	0.1781	0.1565	0.1962	0.0813	0.1870	0.1445	0.1556	0.1554	0.1026
(s4, c2)	0.2597	0.2547	0.2580	0.1278	0.0974	0.1551	-0.0235	0.1392	0.0699	0.0952	0.0889	0.0125
(s4, c3)	0.2394	0.2339	0.2375	0.0776	0.0384	0.1140	-0.1281	0.0915	-0.0047	0.0349	0.0226	-0.0774
(s4, c4)	0.2192	0.2132	0.2169	0.0275	-0.0205	0.0729	-0.2326	0.0439	-0.0792	-0.0254	-0.0437	-0.1673
(s4, c5)	0.1989	0.1926	0.1964	-0.0225	-0.0794	0.0319	-0.3369	-0.0037	-0.1535	-0.0856	-0.1099	-0.2570

the transaction cost structures { (s0, c1), (s0, c2), (s0, c3), (s0, c4), (s0, c5) }; if we do not consider transparent transaction cost, i.e., c=0, the average ARR of RNN is 0.2531 under the transaction cost structure { (s1, c0), (s2, c0), (s3, c0), (s4, c0) }; so transparent transaction cost has greater impact than slippage. Through multiple comparative analysis, the ARR under the transaction cost structures { (s1, c0), (s2, c0), (s3, c0), (s0, c1) } are not significantly different from the ARR without transaction cost for MLP, DBN, and SAE; the ARR under all other transaction cost structures are significantly smaller than the ARR without transaction cost. For all trading algorithms except for MLP, DBN, and SAE, the ARR under the transaction cost structures { (s1, c0), (s2, c0) } are not significantly different from the ARR without transaction cost; the ARR under all other transaction cost structures are significantly smaller than the ARR without transaction cost.

(3) Analysis of Impact of Transaction Cost on ASR. As can be seen from Table 26, ASR is decreasing with the increase of transaction cost for any trading algorithm. Undoubtedly, when the transaction cost is set to (s, c) = (0.04, 0.005), the ASR of each algorithm is the lowest. Compared with setting without transaction cost, the ASR of MLP, DBN, and SAE reduce by 39.97%, 41.23%, and 40.66%, respectively, while the ASR of other trading algorithms reduce by more than 90% compared with the case of no transaction cost. Therefore, excessive transaction cost will significantly reduce ASR. For a general setting of s and c, i.e., (s, c) = (0.02, 0.003), the ASR of MLP, DBN, and SAE decrease by 22.62%, 23.36% and 23.02% respectively. while the ASR of other algorithms decrease by more than 50%; the ASR of CART and XGB decrease by more than 100%. Therefore, MLP, DBN, and SAE are more tolerant to transaction cost. From single trading algorithm such as NB, if we do not consider slippage, i.e., s=0, the average ASR of NB is 0.8052 under the transaction cost structure {(s0, c1), (s0, c2), (s0, c3), (s0, c4), (s0, c5)}; if we do not consider transparent transaction cost, i.e., c=0, the average ASR of NB is 1.4182 under the transaction cost structures {(s1, c0), (s2, c0), (s3, c0), (s4, c0) }; so transparent transaction cost has greater impact than slippage. Through multiple comparative analysis, the ASR under the transaction cost structures {(s1, c0), (s2, c0), (s3, c0), (s0, c1)} are not significantly different from the ASR without transaction cost for MLP, DBN, and SAE; the ASR under all other transaction cost structures are significantly smaller than the ASR without transaction cost. For all trading algorithms except for MLP, DBN, and SAE, the ASR under the transaction cost structures {(s1, c0), (s2, c0)} are not significantly different from the ASR without transaction cost; the ASR under all other transaction cost structures are significantly smaller than the ASR without transaction cost.

(4) Analysis of Impact of Transaction Cost on MDD. As can be seen from Table 27, MDD increases with the increase of transaction cost for any trading algorithm. Undoubtedly, when the transaction cost is set to (s, c) = (0.04, 0.005), the MDD of each algorithm increases to the highest level. In this case, compared with the settings without transaction cost, the MDD of MLP, DBN, and SAE increase by 9.32%, 11.08%, and 10.32%, respectively. The MDD of other trading algorithms increase by more than 80% compared with those without considering transaction cost. Therefore, excessive transaction cost can cause serious potential losses to the account. For a general setting of s and c, i.e., (s, c) = (0.02, 0.003), the MDD of MLP, DBN, and SAE increase by 4.83%, 5.80%, and 5.33%, respectively, while the MDD of other algorithms increase by more than 35%, and the MDD of CART, RF, and XGB increase by more than 100%. Therefore, MLP, DBN, and SAE are more tolerant to transaction cost. As a whole, the DNN models have stronger capacity to accommodate transaction cost than the traditional ML models. From single trading algorithm such as GRU, if we do not consider slippage, i.e., s=0, the average MDD of GRU is 0.4459 under the transaction cost structures {(s0, c1), (s0, c2), (s0, c3), (s0, c4), (s0, c5)}; if we do not consider transparent transaction cost, i.e., c=0, the average MDD of GRU is 0.3559 under the transaction cost structures {(s1, c0), (s2, c0), (s3, c0), (s4, c0)}; so transparent transaction cost has greater impact than slippage. Through multiple comparative analysis, the MDD under any the transaction cost structure is not significantly different from the MDD without transaction cost for MLP, DBN, and SAE. For all trading algorithms except for MLP, DBN, and SAE such as LR, the MDD under the transaction cost structures {(s0, c1), (s1, c0), (s2, c0), (s3, c0)} are not significantly different from the MDD without transaction cost; the MDD under all other transaction cost structures are significantly greater than the MDD without transaction cost.

Through the analysis of the Table 27 performance evaluation indicators, we find that trading performance after considering transaction cost will be worse than that without considering transaction cost as is in actual trading situation. It is noteworthy that the performance changes of DNN algorithms, especially MLP, DBN, and SAE, are very small after considering transaction cost. This shows that the three algorithms have good tolerance to changes of transaction cost. Especially for the MDD of the three algorithms, there is no significant difference with that with no transaction cost. So, we can consider applying them in actual trading. Meanwhile, we conclude that the transparent transaction cost has greater impact on the trading performances than the slippage for SPICS. This is because the prices of SPICS are too high when the transparent transaction cost is set to a certain percentage of turnover. In actual transactions, special attention needs to be paid to the fact that the transaction performance under most transaction cost structures is significantly lower than the trading performance without considering transaction cost. It is worth noting that the performance of traditional ML algorithm is not worse than that of DNN algorithms without considering transaction cost, while the performance of DNN algorithms is better than that of traditional ML algorithms after considering transaction cost.

6.3. Analysis of Impact of Transaction Cost on the Trading Performance of CSICS. Similar to Section 6.2, we will discuss the impact of transaction cost on trading performance of CSICS in the followings. In the Chinese A-share market, the transparent transaction cost is usually set to a certain

Table 26: The ASR of SPICS for daily trading with different transaction cost. The result that there is no significant difference between performance without transaction cost and that with transaction cost is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM	XGB
(s0, c0)	1.5468	1.5412	1.5502	1.5764	1.5571	1.5828	1.3927	1.6237	1.6763	1.5818	1.6018	1.6298
(s0, c1)	**1.4562**	**1.4478**	**1.4577**	1.3000	1.2317	1.3565	0.9346	1.3554	1.2650	1.2472	1.2456	1.1294
(s0, c2)	1.3639	1.3527	1.3635	1.0191	0.9015	1.1262	0.4657	1.0822	0.8482	0.9074	0.8846	0.6236
(s0, c3)	1.2702	1.2561	1.2679	0.7355	0.5690	0.8929	-0.0085	0.8061	0.4298	0.5652	0.5216	0.1182
(s0, c4)	1.1751	1.1581	1.1709	0.4511	0.2366	0.6580	-0.4823	0.5291	0.0139	0.2234	0.1596	-0.3812
(s0, c5)	1.079	1.0591	1.0728	0.1678	-0.0931	0.4227	-0.9500	0.2534	-0.3959	-0.1152	-0.1987	-0.8696
(s1, c0)	**1.5121**	**1.5057**	**1.5149**	**1.4927**	**1.4606**	**1.5119**	**1.2449**	**1.5424**	**1.5582**	**1.4825**	**1.4974**	**1.4886**
(s1, c1)	1.4208	1.4116	1.4217	1.2146	1.1334	1.2841	0.7823	1.2722	1.1447	1.1459	1.1394	0.9859
(s1, c2)	1.3279	1.3159	1.3270	0.9325	0.8020	1.0526	0.3105	0.9977	0.7267	0.8048	0.7772	0.4792
(s1, c3)	1.2337	1.2188	1.2308	0.6482	0.4690	0.8185	-0.1646	0.7209	0.3082	0.4622	0.4139	-0.0257
(s1, c4)	1.1382	1.1204	1.1333	0.3637	0.1368	0.5831	-0.6374	0.4438	-0.1068	0.1208	0.0524	-0.5231
(s1, c5)	1.0417	1.0210	1.0348	0.0807	-0.1920	0.3478	-1.1025	0.1686	-0.5146	-0.2169	-0.3047	-1.0083
(s2, c0)	**1.477**	**1.4699**	**1.4792**	**1.4081**	**1.3632**	**1.4403**	**1.0946**	**1.4601**	**1.4390**	**1.3822**	**1.3922**	**1.3462**
(s2, c1)	1.3851	1.3752	1.3854	1.1285	1.0343	1.2111	0.6283	1.1883	1.0237	1.0439	1.0325	0.8416
(s2, c2)	1.2916	1.2789	1.2901	0.8454	0.7021	0.9785	0.1547	0.9128	0.6049	0.7019	0.6696	0.3346
(s2, c3)	1.1969	1.1812	1.1934	0.5607	0.3688	0.7437	-0.3205	0.6355	0.1868	0.3591	0.3063	-0.1690
(s2, c4)	1.1010	1.0825	1.0955	0.2763	0.0372	0.5081	-0.7916	0.3586	-0.2268	0.0183	-0.0545	-0.6639
(s2, c5)	1.0041	0.9827	0.9967	-0.006	-0.2905	0.2729	-1.2533	0.0841	-0.6323	-0.3179	-0.4101	-1.1454
(s3, c0)	**1.4415**	**1.4337**	**1.4432**	1.3227	1.2650	1.3679	0.9425	1.3770	1.3189	1.2810	1.2862	1.2030
(s3, c1)	1.3490	1.3384	1.3488	1.0419	0.9348	1.1375	0.4733	1.1039	0.9023	0.9413	0.9252	0.6971
(s3, c2)	1.2551	1.2416	1.2529	0.7580	0.6019	0.9040	-0.0013	0.8275	0.4832	0.5988	0.5618	0.1904
(s3, c3)	1.1599	1.1435	1.1558	0.4731	0.2688	0.6688	-0.4757	0.5500	0.0658	0.2562	0.1988	-0.3116
(s3, c4)	1.0636	1.0443	1.0575	0.1891	-0.0620	0.4331	-0.9442	0.2735	-0.3460	-0.0836	-0.1608	-0.8034
(s3, c5)	0.9664	0.9443	0.9584	-0.0924	-0.3883	0.1982	-1.4018	0.0000	-0.7489	-0.4183	-0.5146	-1.2808
(s4, c0)	1.4058	1.3972	1.4068	1.2368	1.1662	1.2950	0.7892	1.2933	1.1983	1.1793	1.1796	1.0591
(s4, c1)	1.3127	1.3013	1.3119	0.9549	0.8350	1.0634	0.3180	1.0190	0.7807	0.8386	0.8177	0.5527
(s4, c2)	1.2183	1.204	1.2155	0.6705	0.5018	0.8293	-0.1568	0.7421	0.3617	0.4958	0.4542	0.0468
(s4, c3)	1.1226	1.1055	1.1179	0.3857	0.1691	0.5938	-0.6295	0.4647	-0.0545	0.1537	0.0918	-0.4529
(s4, c4)	1.026	1.006	1.0193	0.1021	-0.1607	0.3582	-1.0948	0.1888	-0.4642	-0.1849	-0.2665	-0.9413
(s4, c5)	0.9286	0.9057	0.9199	-0.1783	-0.4853	0.1238	-1.5478	-0.0836	-0.8642	-0.5177	-0.6182	-1.4142

Table 27: The MDD of SPICS for daily trading with different transaction cost. The result that there is no significant difference between performance without transaction cost and that with transaction cost is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM	XGB
(s0, c0)	0.3583	0.3584	0.3547	0.3403	0.3489	0.3381	0.3413	0.3428	0.3284	0.3447	0.3429	0.3338
(s0, c1)	**0.3629**	**0.3638**	**0.3594**	**0.3779**	**0.3986**	**0.3636**	**0.4072**	**0.3712**	**0.3843**	**0.3963**	**0.3972**	**0.4203**
(s0, c2)	**0.3677**	**0.3695**	**0.3647**	0.4302	0.4707	0.3968	0.5168	0.4127	0.4842	0.4729	0.4787	0.5735
(s0, c3)	**0.3727**	**0.3756**	**0.3703**	0.4990	0.5639	0.4376	0.6564	0.4709	0.6172	0.5682	0.5844	0.7335
(s0, c4)	**0.3781**	**0.3821**	**0.3764**	0.5767	0.6612	0.4873	0.7804	0.5424	0.7377	0.6653	0.6913	0.8447
(s0, c5)	**0.3839**	**0.3890**	**0.3828**	0.6529	0.746	0.5444	0.8730	0.6162	0.8272	0.7501	0.7808	0.9118
(s1, c0)	**0.3596**	**0.3600**	**0.3560**	**0.3500**	**0.36130**	**0.3446**	**0.3574**	**0.3502**	**0.3414**	**0.3585**	**0.3569**	**0.3540**
(s1, c1)	**0.3642**	**0.3655**	**0.3609**	**0.3907**	0.4156	**0.3717**	0.4320	**0.3814**	0.4067	0.4154	0.4170	0.4541
(s1, c2)	**0.3691**	**0.3712**	**0.3662**	0.4466	0.4936	0.4066	0.5534	0.4270	0.5172	0.4971	0.5049	0.6168
(s1, c3)	**0.3742**	**0.3774**	**0.3720**	0.5183	0.5895	0.4495	0.6928	0.4885	0.6499	0.5937	0.6129	0.7662
(s1, c4)	**0.3796**	**0.3839**	**0.3781**	0.5961	0.6842	0.5013	0.8105	0.5610	0.7631	0.6884	0.7161	0.8649
(s1, c5)	**0.3856**	**0.3909**	**0.3847**	0.671	0.7646	0.5595	0.8946	0.6342	0.8451	0.7691	0.800	0.9235
(s2, c0)	**0.3609**	**0.3615**	**0.3573**	**0.3607**	**0.3756**	**0.3517**	**0.3770**	**0.3586**	**0.3586**	**0.3739**	**0.3727**	**0.3787**
(s2, c1)	**0.3656**	**0.3671**	**0.3623**	0.4047	0.4349	0.3805	0.4627	**0.3929**	0.4339	0.4365	0.4397	0.4929
(s2, c2)	**0.3705**	**0.3729**	**0.3678**	0.4642	0.5176	0.4171	0.5916	0.4424	0.5504	0.5218	0.5327	0.6577
(s2, c3)	**0.3756**	**0.3792**	**0.3736**	0.5377	0.6143	0.4624	0.7277	0.5067	0.6805	0.6183	0.6402	0.7939
(s2, c4)	**0.3812**	**0.3859**	**0.3799**	0.6155	0.7056	0.5161	0.8380	0.5796	0.7856	0.7099	0.7388	0.8816
(s2, c5)	**0.3873**	**0.3930**	**0.3866**	0.6887	0.7816	0.5751	0.9126	0.6517	0.8606	0.7864	0.8172	0.9331
(s3, c0)	**0.3622**	**0.3631**	**0.3588**	**0.3729**	**0.3912**	**0.3594**	**0.4004**	**0.3685**	**0.3795**	**0.3909**	**0.3909**	**0.4081**
(s3, c1)	**0.3669**	**0.3687**	**0.3639**	0.4200	0.4555	0.3901	0.4966	0.4062	0.4622	0.4587	0.4642	0.5334
(s3, c2)	**0.3719**	**0.3746**	**0.3694**	0.4826	0.5423	0.4286	0.6295	0.4589	0.5833	0.5465	0.5607	0.6936
(s3, c3)	**0.3772**	**0.3811**	**0.3754**	0.5572	0.6377	0.4762	0.7609	0.5248	0.7081	0.6420	0.6657	0.8175
(s3, c4)	**0.3829**	**0.3879**	**0.3818**	0.6341	0.7254	0.5314	0.8620	0.5980	0.8054	0.7300	0.7593	0.8956
(s3, c5)	**0.3894**	**0.3954**	**0.3888**	0.7056	0.7972	0.5909	0.9275	0.6689	0.8740	0.8022	0.8325	0.9412
(s4, c0)	**0.3635**	**0.3647**	**0.3602**	**0.3861**	**0.4082**	**0.3678**	0.4274	**0.3798**	0.4015	0.4096	0.4114	0.4396
(s4, c1)	**0.3683**	**0.3704**	**0.3654**	0.4362	0.4774	0.4005	0.5325	0.4211	0.4908	0.4814	0.4894	0.5730
(s4, c2)	**0.3734**	**0.3765**	**0.3712**	0.5013	0.5664	0.4410	0.6667	0.4758	0.6142	0.5707	0.5875	0.7253
(s4, c3)	**0.3790**	**0.3833**	**0.3775**	0.5765	0.6600	0.4905	0.7906	0.5429	0.7330	0.6644	0.6894	0.8375
(s4, c4)	**0.3851**	**0.3904**	**0.3841**	0.6521	0.7439	0.5470	0.8824	0.6162	0.8230	0.7485	0.7782	0.9074
(s4, c5)	**0.3917**	**0.3981**	**0.3913**	0.7218	0.8115	0.6067	0.9395	0.6857	0.8858	0.8165	0.8463	0.9480

percentage of turnover, and it is the same as the assumption in the experimental settings. As in the US stock market, the smallest unit of price change is 0.01 (one tick). It is reasonable to set slippage to be 0.01-0.05. Of course, it should be noted that the prices fluctuation may be more intense when closing than that in the middle of a trading day.

(1) Analysis of Impact of Transaction Cost on WR. As can be seen from Table 28, the WR is decreasing with the increase of transaction cost for any trading algorithm. When the transaction cost is set to (s, c) = (0.04, 0.005), the WR of each algorithm is the smallest. Compared with the settings without transaction cost, the WR of MLP, DBN, SAE, RNN, LSTM, GRU, CART, NB, RF, LR, SVM, and XGB are reduced by 6.71%, 6.88%, 6.97%, 22.69%, 17.26%, 15.48%, 24.30%, 14.91%, 24.84%, 21.12%, 21.12%, and 29.19%, respectively. For a general setting of s and c, i.e., (s, c) = (0.02, 0.003), the WR of MLP, DBN, and SAE decrease by 4.10%, 4.20%, and 4.30%, respectively, while the WR of other algorithms decrease by more than 9%; the WR of CART, RF, and XGB decrease by more than 15%. Therefore, MLP, DBN, and SAE are more tolerant to transaction cost. From single trading algorithm such as LSTM, if we do not consider slippage, i.e., s=0, the average WR of DBN is 0.5417 under the transaction cost structures { (s0, c1), (s0, c2), (s0, c3), (s0, c4), (s0, c5) }; if we do not consider transparent transaction cost, i.e., c=0, the average WR of LSTM is 0.5304 under the transaction cost structures { (s1, c0), (s2, c0), (s3, c0), (s4, c0) }; so transparent transaction cost has smaller impact than slippage. Through multiple comparative analysis, the WR under the transaction cost structures { (s0, c1), (s0, c2), (s1, c0) } are not significantly different from the WR without transaction cost for MLP, DBN, SAE, and NB; the WR under all other transaction cost structures are significantly smaller than the WR without transaction cost. For all trading algorithms except for MLP, DBN, SAE, and NB, the WR under the transaction cost structure (s0, c1) is not significantly different from the WR without transaction cost; the WR under all other transaction cost structures are significantly smaller than the WR without transaction cost.

Table 28: The WR of CSICS for daily trading with different transaction cost. The result that there is no significant difference between performance without transaction cost and that with transaction cost is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM	XGB
(s0, c0)	0.5559	0.5565	0.5564	0.5681	0.5720	0.5717	0.5153	0.5317	0.5785	0.5809	0.5716	0.5803
(s0, c1)	**0.5523**	**0.5527**	**0.5525**	**0.5525**	**0.5608**	**0.5620**	**0.5009**	**0.5227**	**0.5612**	**0.5665**	**0.5571**	**0.5595**
(s0, c2)	**0.5488**	**0.5492**	**0.5489**	0.5389	0.5512	0.5535	0.4879	**0.5149**	0.5460	0.5542	0.5445	0.5411
(s0, c3)	0.5453	0.5456	0.5452	0.5258	0.5414	0.5451	0.4747	0.5068	0.5313	0.5418	0.5320	0.5234
(s0, c4)	0.5417	0.5419	0.5414	0.5127	0.5320	0.5368	0.4622	0.4991	0.5172	0.5297	0.5202	0.5067
(s0, c5)	0.5383	0.5383	0.5379	0.5004	0.5230	0.5286	0.4504	0.4917	0.5036	0.5180	0.5088	0.4905
(s1, c0)	**0.5494**	**0.5499**	**0.5497**	0.5444	0.5541	0.5558	0.4925	**0.5170**	0.5520	0.5584	0.5492	0.5488
(s1, c1)	0.5456	0.5459	0.5456	0.5286	0.5424	0.5456	0.4775	0.5080	0.5342	0.5437	0.5345	0.5275
(s1, c2)	0.5421	0.5423	0.5420	0.5161	0.5335	0.5377	0.4654	0.5007	0.5207	0.5320	0.5231	0.5110
(s1, c3)	0.5386	0.5388	0.5383	0.5036	0.5246	0.5296	0.4530	0.4931	0.5065	0.5205	0.5116	0.4946
(s1, c4)	0.5353	0.5354	0.5349	0.4915	0.5156	0.5218	0.4419	0.4861	0.4937	0.5095	0.5007	0.4795
(s1, c5)	0.5323	0.5323	0.5317	0.4808	0.5076	0.5148	0.4315	0.4796	0.4817	0.4995	0.4905	0.4652
(s2, c0)	0.5431	0.5434	0.5431	0.5219	0.5368	0.5403	0.4707	0.5036	0.5269	0.5374	0.5286	0.5189
(s2, c1)	0.5395	0.5397	0.5393	0.5078	0.5266	0.5314	0.4573	0.4956	0.5115	0.5242	0.5154	0.5005
(s2, c2)	0.5360	0.5361	0.5357	0.4960	0.5181	0.5237	0.4458	0.4886	0.4985	0.5134	0.5046	0.4853
(s2, c3)	0.5331	0.5331	0.5326	0.4850	0.5100	0.5167	0.4352	0.4818	0.4864	0.5031	0.4945	0.4711
(s2, c4)	0.5300	0.5300	0.5293	0.4743	0.5018	0.5093	0.4252	0.4752	0.4746	0.4931	0.4846	0.4572
(s2, c5)	0.5273	0.5271	0.5266	0.4648	0.4946	0.5029	0.4159	0.4692	0.4639	0.4838	0.4755	0.4445
(s3, c0)	0.5373	0.5374	0.5371	0.5019	0.5216	0.5265	0.4514	0.4917	0.5049	0.5186	0.5098	0.4932
(s3, c1)	0.5341	0.5341	0.5336	0.4902	0.5128	0.5188	0.4399	0.4846	0.4917	0.5074	0.4990	0.4775
(s3, c2)	0.5312	0.5312	0.5306	0.4798	0.5052	0.5122	0.4303	0.4782	0.4804	0.4977	0.4896	0.4644
(s3, c3)	0.5281	0.5281	0.5275	0.4696	0.4976	0.5049	0.4206	0.4718	0.4689	0.4879	0.4799	0.4509
(s3, c4)	0.5252	0.5251	0.5245	0.4598	0.4901	0.4984	0.4110	0.4657	0.4581	0.4785	0.4707	0.4378
(s3, c5)	0.5226	0.5223	0.5218	0.4510	0.4833	0.4924	0.4023	0.4602	0.4481	0.4701	0.4621	0.4264
(s4, c0)	0.5325	0.5325	0.5321	0.4860	0.5089	0.5150	0.4360	0.4816	0.4870	0.5030	0.4949	0.4723
(s4, c1)	0.5294	0.5294	0.5289	0.4753	0.5010	0.5079	0.4258	0.4750	0.4752	0.4928	0.4849	0.4588
(s4, c2)	0.5266	0.5265	0.5259	0.4653	0.4937	0.5013	0.4164	0.4690	0.4642	0.4838	0.4761	0.4458
(s4, c3)	0.5238	0.5236	0.5230	0.4562	0.4864	0.4948	0.4073	0.4632	0.4541	0.4747	0.4672	0.4336
(s4, c4)	0.5211	0.5208	0.5203	0.4475	0.4798	0.4891	0.3985	0.4577	0.4440	0.4662	0.4586	0.4218
(s4, c5)	0.5186	0.5182	0.5176	0.4392	0.4733	0.4832	0.3901	0.4524	0.4348	0.4582	0.4509	0.4109

(2) Analysis of Impact of Transaction Cost on ARR. As can be seen from Table 29, ARR is decreasing with the increase of transaction cost for any trading algorithm. Undoubtedly, when the transaction cost is set to (s, c) = (0.04, 0.005), the ARR of each algorithm is the smallest. Compared with the settings without transaction cost, the ARR of MLP, DBN, and SAE reduce by 50.73%, 51.75%, and 52.25%, respectively. While the ARR of other trading algorithms decrease by more than 100% compared with those algorithms without transaction cost. Therefore, excessive transaction cost can lead to serious losses in the accounts. For a general setting of s and c, i.e., (s, c) = (0.02, 0.003), ARR of MLP, DBN, and SAE decrease by 27.41%, 27.97%, and 28.25% respectively, while the ARR other algorithms decrease by more than 50% and that of CART, NB, RF, and XGB decrease by more than 100%. Therefore, MLP, DBN, and SAE are more tolerant to transaction cost. From single trading algorithm such as SAE, if we do not consider slippage, i.e., s=0, the average ARR of SAE is 0.5040 under the transaction cost structure {(s0, c1), (s0, c2), (s0, c3), (s0, c4), (s0, c5)}; if we do not consider transparent transaction cost, i.e., c=0, the average ARR of SAE is 0.4468 under the transaction cost structures {(s1, c0), (s2, c0), (s3, c0), (s4, c0)}; so transparent transaction cost has smaller impact than slippage. Through multiple comparative analysis, the ARR under the transaction cost structures {(s0, c1), (s0, c2), (s0, c3), (s0, c1), (s1, c1)} are not significantly different from the ARR without transaction cost for MLP, DBN, and SAE; the ARR under all other transaction cost structures are significantly smaller than the ARR without transaction cost. For RNN, LSTM, GRU, CART, RF, LR, and SVM, the ARR under the transaction cost structures {(s0, c1), (s0, c2), (s1, c0)} are not significantly different from the ARR without transaction cost; the ARR under all other transaction cost structures are significantly smaller than the ARR without transaction cost. For NB and XGB, the ARR under the transaction cost structures {(s0, c1), (s1, c0)} are not significantly different from the ARR without transaction cost; the ARR under all other transaction cost structures are significantly smaller than the ARR without transaction cost.

Table 29: The ARR of CSICS for daily trading with different transaction cost. The result that there is no significant difference between performance without transaction cost and that with transaction cost is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM	XGB
(s0, c0)	0.5728	0.5702	0.5675	0.5246	0.5162	0.5110	0.5531	0.6122	0.484	0.5092	0.5001	0.4935
(s0, c1)	**0.5521**	**0.549**	**0.5463**	**0.4522**	**0.4697**	**0.4707**	**0.4490**	**0.5072**	**0.4095**	**0.4494**	**0.4331**	**0.4044**
(s0, c2)	**0.5314**	**0.5279**	**0.5251**	**0.3799**	**0.4232**	**0.4305**	**0.3450**	0.4023	**0.3351**	**0.3896**	**0.3662**	0.3154
(s0, c3)	**0.5107**	**0.5068**	**0.5039**	0.3077	0.3767	0.3904	0.2411	0.2976	0.2608	0.3299	0.2994	0.2266
(s0, c4)	0.4901	0.4858	0.4828	0.2356	0.3303	0.3503	0.1374	0.1930	0.1866	0.2703	0.2327	0.1379
(s0, c5)	0.4695	0.4648	0.4617	0.1636	0.2840	0.3102	0.0339	0.0886	0.1125	0.2108	0.1661	0.0493
(s1, c0)	**0.5248**	**0.5216**	**0.5186**	**0.3917**	**0.4250**	**0.4302**	**0.3437**	**0.4210**	**0.3463**	**0.3963**	**0.3746**	**0.3318**
(s1, c1)	**0.5042**	**0.5005**	**0.4975**	0.3195	0.3785	0.3900	0.2399	0.3162	0.2720	0.3366	0.3078	0.2429
(s1, c2)	0.4835	0.4795	0.4764	0.2474	0.3321	0.3499	0.1362	0.2116	0.1978	0.2770	0.2411	0.1542
(s1, c3)	0.463	0.4585	0.4553	0.1754	0.2858	0.3099	0.0327	0.1072	0.1237	0.2174	0.1745	0.0656
(s1, c4)	0.4424	0.4375	0.4342	0.1035	0.2396	0.2699	-0.0707	0.0029	0.0497	0.15800	0.1079	-0.0229
(s1, c5)	0.4219	0.4165	0.4132	0.0317	0.1934	0.2299	-0.174	-0.1013	-0.0242	0.0986	0.0415	-0.1113
(s2, c0)	0.4774	0.4736	0.4704	0.2599	0.3344	0.3501	0.1363	0.2310	0.2095	0.2842	0.2500	0.1711
(s2, c1)	0.4568	0.4526	0.4493	0.1879	0.2881	0.3100	0.0328	0.1265	0.1354	0.2247	0.1834	0.0825
(s2, c2)	0.4363	0.4316	0.4282	0.1159	0.2419	0.2700	-0.0706	0.0222	0.0614	0.1652	0.1168	-0.006
(s2, c3)	0.4158	0.4107	0.4072	0.0441	0.1957	0.2301	-0.1739	-0.0820	-0.0125	0.1058	0.0504	-0.0944
(s2, c4)	0.3953	0.3898	0.3862	-0.0276	0.1495	0.1902	-0.2770	-0.1861	-0.0863	0.0465	-0.0160	-0.1827
(s2, c5)	0.3748	0.3689	0.3653	-0.0992	0.1034	0.1503	-0.3799	-0.2900	-0.1600	-0.0127	-0.0823	-0.2708
(s3, c0)	0.4305	0.4261	0.4226	0.1289	0.2446	0.2706	-0.0694	0.0421	0.0737	0.1729	0.1263	0.0115
(s3, c1)	0.4100	0.4052	0.4016	0.0570	0.1984	0.2306	-0.1726	-0.0621	-0.0003	0.1135	0.0598	-0.0769
(s3, c2)	0.3895	0.3843	0.3806	-0.0147	0.1522	0.1907	-0.2757	-0.1662	-0.0741	0.0542	-0.0066	-0.1652
(s3, c3)	0.3691	0.3634	0.3597	-0.0863	0.1062	0.1509	-0.3787	-0.2701	-0.1478	-0.0050	-0.0729	-0.2533
(s3, c4)	0.3487	0.3426	0.3388	-0.1578	0.0601	0.1111	-0.4815	-0.3739	-0.2214	-0.0642	-0.1391	-0.3414
(s3, c5)	0.3283	0.3217	0.3179	-0.2293	0.0142	0.0713	-0.5842	-0.4775	-0.2949	-0.1233	-0.2052	-0.4293
(s4, c0)	0.3841	0.3791	0.3754	-0.0013	0.1554	0.1917	-0.2734	-0.1457	-0.0614	0.0623	0.0033	-0.1471
(s4, c1)	0.3637	0.3582	0.3544	-0.0729	0.1093	0.1518	-0.3764	-0.2497	-0.1351	0.0031	-0.0630	-0.2353
(s4, c2)	0.3433	0.3374	0.3335	-0.1445	0.0633	0.1120	-0.4792	-0.3535	-0.2087	-0.0561	-0.1292	-0.3233
(s4, c3)	0.3229	0.3166	0.3126	-0.2159	0.0173	0.0723	-0.5819	-0.4572	-0.2823	-0.1152	-0.1953	-0.4113
(s4, c4)	0.3025	0.2958	0.2918	-0.2873	-0.0286	0.0326	-0.6844	-0.5607	-0.3557	-0.1742	-0.2613	-0.4991
(s4, c5)	0.2822	0.2751	0.2710	-0.3585	-0.0744	-0.0071	-0.7868	-0.6641	-0.4290	-0.2331	-0.3273	-0.5868

(3) Analysis of Impact of Transaction Cost on ASR. As can be seen from Table 30, ASR is decreasing with the increase of transaction cost for any trading algorithm. Undoubtedly, when the transaction cost is set to (s, c) = (0.04, 0.005), the ASR of each algorithm is the smallest. Compared with the settings without transaction cost, the ASR of MLP, DBN, and SAE reduce by 48.99%, 50.11%, and 50.70%, respectively, while the ASR of other trading algorithms decrease by more than 100% compared with those without transaction cost. Therefore, excessive transaction cost can lead to serious losses in the accounts. For a general setting of s and c, i.e., (s, c) = (0.02, 0.003), ASR of MLP, DBN, and SAE decrease by 26.01%, 26.61%, and 26.94%, respectively, while the ASR other algorithms decrease by more than 50% and that of CART, NB, RF, and XGB decrease by more than 100%. Therefore, MLP, DBN, and SAE are more tolerant to transaction cost. From single trading algorithm such as LSTM, if we do not consider slippage, i.e., s=0, the average ASR of LSTM is 1.1129 under the transaction cost structures {(s0, c1), (s0, c2), (s0, c3), (s0, c4), (s0, c5)}; if we do not consider transparent transaction cost, i.e., c=0, the average ASR of LSTM is 0.8837 under the transaction cost structures {(s1, c0), (s2, c0), (s3, c0), (s4, c0)}; so transparent transaction cost has smaller impact than slippage. Through multiple comparative analysis, the ASR under the transaction cost structures {(s0, c1), (s0, c2), (s0, c3), (s0, c1), (s1, c1)} are not significantly different from the ASR without transaction cost for MLP, DBN, and SAE; the ASR under all other transaction cost structures are significantly smaller than the ASR without transaction cost. For LSTM and GRU, the ASR under the transaction cost structures {(s0, c1), (s0, c2), (s1, c0)} are not significantly different from the ASR without transaction cost; the ASR under all other transaction cost structures are significantly smaller than the ASR without transaction cost. For RNN, NB, RF, LR, and SVM, the ASR under the transaction cost structures {(s0, c1), (s1, c0)} are not significantly different from the ASR without transaction cost; the ASR under all other transaction cost structures are significantly smaller than the ASR without transaction cost. For CART and XGB, the ASR under the transaction cost structure (s0, c1) are

Table 30: The ASR of CSICS for daily trading with different transaction cost. The result that there is no significant difference between performance without transaction cost and that with transaction cost is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM	XGB
(s0, c0)	1.4027	1.4003	1.3931	1.4876	1.5418	1.5501	1.2229	1.1119	1.4375	1.5578	1.4227	1.4694
(s0, c1)	**1.3525**	**1.3488**	**1.3413**	**1.2736**	**1.4005**	**1.4268**	**0.9763**	**0.9356**	**1.2078**	**1.3697**	**1.2253**	**1.1938**
(s0, c2)	**1.3019**	**1.2969**	**1.2890**	1.0579	**1.2578**	**1.3023**	0.7263	0.7584	0.9763	1.1796	1.0263	0.9161
(s0, c3)	**1.2509**	**1.2445**	**1.2364**	0.8411	1.1139	1.1765	0.4736	0.5804	0.7436	0.9882	0.8263	0.6372
(s0, c4)	1.1996	1.1918	1.1834	0.6237	0.9690	1.0499	0.2191	0.4021	0.5104	0.7958	0.6255	0.3581
(s0, c5)	1.1479	1.1388	1.1301	0.4063	0.8235	0.9224	-0.0364	0.2236	0.2773	0.6029	0.4244	0.0795
(s1, c0)	**1.2996**	**1.2953**	**1.2872**	**1.1103**	**1.2817**	**1.3133**	0.7570	**0.7905**	**1.0278**	**1.2127**	**1.0676**	0.9843
(s1, c1)	**1.2486**	**1.2430**	**1.2346**	0.8938	1.1379	1.1876	0.5050	0.6126	0.7955	1.0215	0.8678	0.7058
(s1, c2)	1.1972	1.1903	1.1816	0.6766	0.9932	1.0610	0.2511	0.4343	0.5625	0.8293	0.6672	0.4269
(s1, c3)	1.1455	1.1373	1.1282	0.4591	0.8477	0.9336	-0.0040	0.2558	0.3294	0.6365	0.4662	0.1483
(s1, c4)	1.0935	1.0839	1.0746	0.2420	0.7016	0.8056	-0.2594	0.0775	0.0968	0.4436	0.2654	-0.1291
(s1, c5)	1.0413	1.0303	1.0206	0.0257	0.5554	0.6771	-0.5143	-0.1003	-0.1346	0.2510	0.0651	-0.4045
(s2, c0)	1.1936	1.1875	1.1785	0.7293	1.0157	1.0710	0.2851	0.4662	0.6145	0.8624	0.7084	0.4971
(s2, c1)	1.142	1.1345	1.1252	0.5128	0.8708	0.9441	0.0321	0.2883	0.3826	0.6707	0.5083	0.2199
(s2, c2)	1.0901	1.0812	1.0717	0.2964	0.7253	0.8166	-0.2213	0.1105	0.1510	0.4787	0.3083	-0.0564
(s2, c3)	1.0379	1.0277	1.0178	0.0807	0.5794	0.6887	-0.4745	-0.067	-0.0797	0.2870	0.1086	-0.3310
(s2, c4)	0.9854	0.9739	0.9637	-0.1339	0.4335	0.5605	-0.7267	-0.2437	-0.3088	0.0958	-0.0902	-0.6030
(s2, c5)	0.9328	0.9199	0.9094	-0.3469	0.2878	0.4323	-0.9771	-0.4195	-0.5359	-0.0942	-0.2878	-0.8719
(s3, c0)	1.0862	1.0781	1.0683	0.3537	0.7496	0.8304	-0.1751	0.1456	0.2086	0.5184	0.3534	0.0217
(s3, c1)	1.0342	1.0247	1.0147	0.1393	0.6047	0.7033	-0.4254	-0.0309	-0.0204	0.3282	0.1550	-0.2509
(s3, c2)	0.9819	0.9711	0.9607	-0.0742	0.4596	0.5760	-0.6751	-0.2068	-0.2481	0.1384	-0.0426	-0.5214
(s3, c3)	0.9294	0.9172	0.9066	-0.2862	0.3146	0.4485	-0.9232	-0.3818	-0.4741	-0.0504	-0.2392	-0.7891
(s3, c4)	0.8767	0.8632	0.8522	-0.4965	0.1700	0.3211	-1.1693	-0.5558	-0.6978	-0.2380	-0.4343	-1.0534
(s3, c5)	0.8239	0.809	0.7977	-0.7045	0.0258	0.1940	-1.4125	-0.7283	-0.9188	-0.4240	-0.6276	-1.3135
(s4, c0)	0.9785	0.9684	0.9580	-0.0099	0.4878	0.5943	-0.6140	-0.1663	-0.1821	0.1862	0.0089	-0.4325
(s4, c1)	0.9263	0.9148	0.9040	-0.2205	0.3439	0.4679	-0.8592	-0.3403	-0.4063	-0.0010	-0.1862	-0.6982
(s4, c2)	0.8738	0.8609	0.8499	-0.4296	0.2003	0.3415	-1.1027	-0.5132	-0.6285	-0.1871	-0.3800	-0.9608
(s4, c3)	0.8212	0.8070	0.7957	-0.6366	0.0571	0.2153	-1.3437	-0.6849	-0.8482	-0.3718	-0.5722	-1.2198
(s4, c4)	0.7684	0.7528	0.7413	-0.8412	-0.0854	0.0894	-1.5818	-0.8551	-1.0652	-0.5547	-0.7625	-1.4747
(s4, c5)	0.7155	0.6986	0.6868	-1.0431	-0.2271	-0.0359	-1.8162	-1.0236	-1.2788	-0.7356	-0.9505	-1.7248

not significantly different from the ASR without transaction cost; the ASR under all other transaction cost structures are significantly smaller than the ASR without transaction cost.

(4) Analysis of Impact of Transaction Cost on MDD. As can be seen from Table 31, MDD increases with the increase of transaction cost for any transaction algorithm. Undoubtedly, when the transaction cost is set to (s, c) = (0.04, 0.005), the MDD of each algorithm increases to the highest level. In this case, compared with the setting without transaction cost, the MDD of MLP, DBN, and SAE increase by 10.31%, 11.35%, and 10.83%, respectively. The MDD of the other transaction algorithms increases by more than 30% compared with those without transaction cost. Therefore, excessive transaction cost can cause serious potential losses to the account. For a general setting of s and c, i.e., (s, c) = (0.02, 0.003), the MDD of MLP, DBN, and SAE increase by 4.31%, 4.81%, and 4.80%, respectively. While the MDD of the other algorithms increase by more than 20%, the MDD of CART, RF, and XGB increase by more than 60%. Therefore, MLP, DBN, and SAE are more tolerant to transaction cost. From a single trading algorithm such as RNN, if we do not consider slippage, i.e., s=0, the average MDD of RNN is 0.7402 under the transaction cost structures { (s0, c1), (s0, c2), (s0, c3), (s0, c4), (s0, c5) }; if we do not consider transparent transaction cost, i.e., c=0, the average MDD of RNN is 0.7754 under the transaction cost structures { (s1, c0), (s2, c0), (s3, c0), (s4, c0) }; so transparent transaction cost has smaller impact than slippage. Through multiple comparative analysis, the MDD under most of the transaction cost structures are not significantly different from the MDD without transaction cost for MLP, DBN, and SAE. It shows that the three algorithms have higher tolerance for transaction cost. For all trading algorithms except for MLP, DBN, and SAE, the MDD under the transaction cost structures { (s0, c1), (s0, c2), (s1, c0) } are not significantly different from the MDD without transaction cost; the MDD under all other transaction cost structures are significantly greater than the MDD without transaction cost. It is worth noting that the MDD of GRU under the transaction cost

Table 31: The MDD of CSICS for daily trading with different transaction cost. The result that there is no significant difference between performance without transaction cost and that with transaction cost is in boldface.

	MLP	DBN	SAE	RNN	LSTM	GRU	CART	NB	RF	LR	SVM	XGB
(s0, c0)	0.6082	0.6086	0.6130	0.5648	0.5456	0.5429	0.5694	0.7469	0.5695	0.5410	0.5775	0.5632
(s0, c1)	**0.6115**	**0.6122**	**0.6168**	**0.6133**	**0.5759**	**0.5665**	**0.6272**	**0.7765**	**0.6317**	**0.5843**	**0.6249**	**0.6419**
(s0, c2)	**0.6150**	**0.6163**	**0.6207**	**0.6731**	**0.6098**	**0.5922**	**0.6966**	**0.8088**	**0.7041**	**0.6353**	**0.6820**	**0.7296**
(s0, c3)	**0.6188**	**0.6208**	**0.6247**	0.7426	0.6468	0.6194	0.7674	0.8418	0.7777	0.6901	0.7435	0.8158
(s0, c4)	**0.6229**	**0.6256**	**0.6289**	0.8083	0.6844	0.6471	0.8338	0.8728	0.8426	0.7453	0.8045	0.8858
(s0, c5)	**0.6273**	**0.6305**	**0.6333**	0.8637	0.7200	0.6763	0.8887	0.8986	0.8934	0.7979	0.8563	0.9324
(s1, c0)	**0.6137**	**0.6145**	**0.6191**	**0.6577**	**0.6033**	**0.5870**	**0.6847**	**0.8049**	**0.6893**	**0.6265**	**0.6725**	**0.7088**
(s1, c1)	**0.6174**	**0.6187**	**0.6230**	0.7181	0.6384	**0.6135**	0.7485	0.8350	0.7556	0.6766	0.7304	0.7848
(s1, c2)	**0.6213**	**0.6234**	**0.6270**	0.7818	0.6746	0.6408	0.8115	0.8641	0.8180	0.7292	0.7859	0.8542
(s1, c3)	**0.6255**	**0.6282**	**0.6313**	0.8413	0.7107	0.6680	0.8670	0.8890	0.8721	0.7797	0.8387	0.9087
(s1, c4)	**0.6298**	**0.6331**	**0.6359**	0.8884	0.7451	0.6964	0.9104	0.9099	0.9134	0.8267	0.8828	0.9455
(s1, c5)	**0.6343**	**0.6382**	**0.6406**	0.9237	0.7776	0.7254	0.9432	0.9269	0.9426	0.8674	0.9169	0.9684
(s2, c0)	**0.6214**	**0.6230**	**0.6269**	0.7549	0.6700	0.6369	0.7815	0.8546	0.7857	0.7131	0.7661	0.8131
(s2, c1)	**0.6256**	**0.6277**	**0.6312**	0.8084	0.7032	0.6627	0.8327	0.8785	0.8387	0.7597	0.8138	0.8688
(s2, c2)	**0.6299**	**0.6327**	**0.6357**	0.8579	0.7350	0.6887	0.8793	0.8998	0.8851	0.8037	0.8576	0.9142
(s2, c3)	**0.6344**	**0.6379**	**0.6406**	0.8991	0.7664	0.7163	0.9174	0.9179	0.9207	0.8449	0.8955	0.9473
(s2, c4)	**0.6391**	**0.6431**	**0.6456**	0.9305	0.7963	0.7444	0.9467	0.9335	0.9468	0.8802	0.9250	0.9690
(s2, c5)	**0.6443**	0.6487	**0.6511**	0.9528	0.8230	0.7721	0.9667	0.9457	0.9650	0.9091	0.9471	0.9820
(s3, c0)	**0.6299**	**0.6320**	**0.6357**	0.8218	0.7242	0.6808	0.8419	0.8857	0.8488	0.7794	0.8291	0.8725
(s3, c1)	**0.6345**	**0.6373**	**0.6406**	0.8645	0.7528	0.7063	0.8839	0.9047	0.8891	0.8181	0.8667	0.9131
(s3, c2)	**0.6394**	**0.6427**	**0.6459**	0.9018	0.7812	0.7325	0.9188	0.9216	0.9217	0.8538	0.8998	0.9442
(s3, c3)	**0.6446**	**0.6481**	**0.6514**	0.9313	0.8084	0.7591	0.9461	0.9361	0.9463	0.8857	0.9271	0.9661
(s3, c4)	0.6500	0.6542	0.6572	0.9529	0.8329	0.7859	0.9656	0.9481	0.9642	0.9122	0.9481	0.9801
(s3, c5)	0.6561	0.6610	0.6634	0.9681	0.8555	0.8113	0.9789	0.9576	0.9764	0.9338	0.9633	0.9885
(s4, c0)	**0.6399**	**0.6432**	**0.6460**	0.8670	0.7666	0.7240	0.8832	0.9079	0.8893	0.8270	0.8710	0.9087
(s4, c1)	**0.6453**	0.6496	**0.6519**	0.9005	0.7926	0.7490	0.9165	0.9242	0.9197	0.8585	0.9005	0.9388
(s4, c2)	0.6511	0.6561	0.6583	0.9290	0.8169	0.7741	0.9433	0.9382	0.9435	0.8877	0.9262	0.9611
(s4, c3)	0.6573	0.6630	0.6649	0.9508	0.8399	0.7989	0.9629	0.9497	0.9615	0.9128	0.9467	0.9766
(s4, c4)	0.6640	0.6704	0.6719	0.9664	0.8611	0.8225	0.9768	0.9591	0.9744	0.9331	0.9619	0.9863
(s4, c5)	0.6709	0.6777	0.6794	0.9774	0.8804	0.8445	0.9862	0.9665	0.9831	0.9498	0.9730	0.9921

structure (s1, c1) is not significantly different from the MDD without transaction cost.

Through the Table 31 analysis, we find that trading performance will become worse and worse with the increase of transaction cost. Moreover, excessive transaction cost may cause huge losses. Especially, for some traditional ML algorithm, the ARR and ASR of those algorithms will become negative. MDD of the algorithms will become close to 100% when transaction cost is increasing. DNN models, especially MLP, DBN, and SAE, are more tolerant to the changes of transaction cost and are more suitable for actual trading activities. Meanwhile, the experimental results indicate that the impact of slippage on trading performance is greater than the transparent transaction cost because the prices of CSICS are generally small. We conclude that a certain percentage of turnover will generate smaller transaction cost. Through multiple comparative analysis, we find that the performance of these algorithms under most of transaction cost structures may be significantly worse than those without considering transaction cost. The finding shows that the trading performance of these algorithms is very sensitive to transaction cost, which needs to be paid enough attention to in actual trading activities.

7. Discussion

Forecasting the future ups and downs of stock prices and making trading decisions are always challenging tasks. However, more and more investors are attracted to participate in trading activities by high return of stock market, and high risk promotes investors to try their best to construct profitable trading strategies. Meanwhile, the fast changing of financial markets, the explosive growth of big financial data, the increasing complexity of financial investment instruments, and the rapid capture of trading opportunities provide more and more research topics for academic circles. In this paper, we apply some popular and widely used ML algorithms to do stock trading. Our purpose is to explore whether there are significant differences in stock trading performance among different ML algorithms. Moreover, we study whether we can

find highly profitable trading algorithms in the presence of transaction cost.

Financial data, which is generated in changing financial market, are characterized by randomness, low signal-to-noise ratio, nonlinearity, and high dimensionality. Therefore, it is difficult to find inherent patterns in financial big data by using algorithms. In this paper, we also prove this point.

When using ML algorithms to predict stock prices, the directional evaluation indicators are not as good as expected. For example, the AR, PR, and RR of LSTM and RNN are about 50%-55%, which are only slightly better than random guess. On the contrary, some traditional ML algorithms such as XGB have stronger ability in directional predictions of stock prices. Therefore, those simple models are less likely to cause overfitting when capturing intrinsic patterns of financial data and can make better predictions about the directions of stock price changes. Actually, we assume that sample data are independent and identically distributed when using ML algorithm to classify tasks. DNN algorithms such as LSTM and RNN make full use of autocorrelation of financial time series data, which is doubtful because of the characteristics of financial data. Therefore, the prediction ability of these algorithms may be weakened because of the noise of historical lag data.

From the perspective of trading algorithms, traditional ML models map the feature space to the target space. The parameters of the learning model are quite few. Therefore, the learning goal can be better accomplished in the case of fewer data. The DNN models mainly connect some neurons into multiple layers to form a complex DNN structure. Through the complex structure, the mapping relationships between input and output are established. As the number of neural network layers increases, the weight parameters can be automatically adjusted to extract advanced features. Compared with the traditional ML models, DNN models have more parameters. So their performance tends to increase as the amount of data grows. Complex DNN models need a lot of data to avoid underfitting and overfitting. However, we only use the data for 250 trading days (one year) as training set to construct trading model, and then we predict stock prices in the next week. So, too few data may lead to poor performance in the directional and performance predictions.

In the aspect of transaction cost, it is unexpected that DNN models, especially MLP, DBN, and SAE, have stronger adaptability to transaction cost than traditional ML models. In fact, the higher PR of MLP, DBN, and SAE indicate that they can identify more trading opportunities with higher positive return. At the same time, DNN model can adapt to the changes of transaction cost structures well. That is, compared with traditional ML models, the reduction of ARR and ASR of DNN models are very small when transaction cost increases. There especially is no significant difference between the MDD of DNN models under most of transaction cost structures and that without considering transaction cost. This is further proof that DNN models can effectively control downside risk. Therefore, DNN algorithms are better choices than traditional ML algorithm in actual transactions. In this paper, we divide transaction cost into transparent transaction cost and implicit transaction cost. In different markets, the impact of the two transaction cost on performance is different. We can see that transparent transaction cost is a larger impact than implicit transaction cost in SPICS while they are just the opposite in CSICS, because the prices of SPICS are higher than that of CSICS. While we have taken full account of the actual situation in real trading, the assumption of transaction cost in this paper is relatively simple. Therefore, we can consider the impact of opportunity cost and market impact cost on trading performance in future research work.

This paper makes a multiple comparative analysis of trading performance for different ML algorithms by means of nonparameter statistical testing. We comprehensively discuss whether there are significant differences among the algorithms under different evaluation indicators in both cases of transaction cost and no transaction cost. We show that the DNN algorithms have better performance in terms of profitability and risk control ability in the actual environment with transaction cost. Therefore, DNN algorithms can be used as choices for algorithmic trading and quantitative trading.

8. Conclusion

In this paper, we apply 424 SPICS in the US market and 185 CSICS in the Chinese market as research objects, select data of 2000 trading days before December 31, 2017, and build 44 technical indicators as the input features for the ML algorithms, and then predict the trend of each stock price as trading signal. Further, we formulate trading strategies based on these trading signals, and we do backtesting. Finally, we analyze and evaluate the trading performance of these algorithms in both cases of transaction cost and no transaction cost.

Our contribution is to compare the significant differences between the trading performance of the DNN algorithms and the traditional ML algorithms in the Chinese stock market and the American stock market. The experimental results in SPICS and CSICS show that some traditional ML algorithms have a better performance than DNN algorithms in most of the directional evaluation indicators. DNN algorithms which have the best performance indicators (WR, ARR, ASR, and MDD) among all ML algorithms are not significantly better than those traditional ML algorithms without considering transaction cost. With the increase of transaction cost, the transaction performance of all ML algorithms will become worse and worse. Under the same transaction cost structure, the DNN algorithms, especially the MLP, DBN, and SAE, have lower performance degradation than the traditional ML algorithm, indicating that the DNN algorithms have a strong tolerance to the changes of transaction cost. Meanwhile, the transparent transaction cost and implicit transaction cost are different impact for the SPICS and CSICS. The experimental results also reveal that the transaction performance of all ML algorithms is sensitive to transaction cost, and more attention is needed in actual transactions. Therefore, it is essential to select the competitive algorithms for stock trading according to the trading performance, adaptability to transaction cost, and the risk control ability of the algorithms both in the American stock market and Chinese A-share market.

With the rapid development of ML technology and the convenient access to financial big data, future research work can be carried out from the following aspects: (1) using ML algorithms to implement dynamic optimal portfolio among different stocks; (2) using ML algorithms to do high-frequency trading and statistical arbitrage; (3) considering the impact of more complex implicit transaction cost such as opportunity cost and market impact cost on stock trading performance. The solutions of these problems will help to develop an advanced and profitable automated trading system based on financial big data, including dynamic portfolio construction, transaction execution, cost control, and risk management according to the changes of market conditions and even the changes of investor's risk preferences of over time.

Data Availability

We have shared our data availability (software codes and experimental data) in a website and can be found at https://figshare.com/account/articles/7238345.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (nos. 71571136, 61802258), in part by Technology Commission of Shanghai Municipality (no. 16JC1403000).

Supplementary Materials

The supplementary materials submitted along with our manuscript include program codes of every algorithm, datasets, and the main result of this work. The materials have been uploaded to the Figshare database (https://doi.org/10.6084/m9.figshare.7569032). *(Supplementary Materials)*

References

[1] R. C. Cavalcantea, R. C. Brasileirob, V. F. Souzab, J. P. Nobregab, and A. I. Oliveir, "Computational intelligence and financial markets: a survey and future directions," *Expert Systems with Applications*, vol. 55, pp. 194–211, 2016.

[2] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.

[3] J. Chen, "SVM application of financial time series forecasting using empirical technical indicators," in *Proceedings of the International conference on information networking and automation, ICINA '10*, pp. 1–77, China, October 2010.

[4] C. Q. Xie, "The optimization of share price prediction model based on support vector machine," in *Proceedings of the International conference on control, automation and systems engineering, CASE '11*, pp. 1–4, Singapore, July 2011.

[5] P. Ladyzynski, K. Zbikowski, and P. Grzegorzewski, "Stock trading with random forests, trend detection tests and force index volume indicators," in *Proceedings of the International Conference on Artificial Intelligence and Soft Computing, ICAISC '13*, pp. 441–452, Poland, June 2013.

[6] J. Zhang, S. Cui, Y. Xu, Q. Li, and T. Li, "A novel data-driven stock price trend prediction system," *Expert Systems with Applications*, vol. 97, pp. 60–69, 2018.

[7] D. Ruta, "Automated trading with machine learning on big data," in *Proceedings of the 3rd IEEE International Congress on Big Data, BigData Congress '14*, pp. 824–830, USA, July 2014.

[8] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 1, pp. 259–268, 2015.

[9] L. Luo and X. Chen, "Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction," *Applied Soft Computing*, vol. 13, no. 2, pp. 806–816, 2013.

[10] K. Zbikowski, "Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1797–1805, 2015.

[11] R. Dash and P. K. Dash, "A hybrid stock trading framework integrating technical analysis with machine learning techniques," *The Journal of Finance and Data Science*, vol. 2, no. 1, pp. 42–57, 2016.

[12] M. Gorenc Novak and D. Velušček, "Prediction of stock price movement based on daily high prices," *Quantitative Finance*, vol. 16, no. 5, pp. 793–826, 2015.

[13] R. Cervello-Royo, "Stock market trading rule based on pattern recognition and technical analysis: forecasting the DJIA index with intraday data," *Expert Systems with Applications*, vol. 42, no. 14, pp. 5963–5975, 2015.

[14] R. C. Brasileiro, V. L. F. Souza, and A. L. I. Oliveira, "Automatic trading method based on piecewise aggregate approximation and multi-swarm of improved self-adaptive particle swarm optimization with validation," *Decision Support Systems*, vol. 104, pp. 79–91, 2017.

[15] Y. Chen and X. Wang, "A hybrid stock trading system using genetic network programming and mean conditional value-at-risk," *European Journal of Operational Research*, vol. 240, no. 3, pp. 861–871, 2015.

[16] L. S. Malagrino, N. T. Roman, and A. M. Monteiro, "Forecasting stock market index daily direction: a bayesian network approach," *Expert Systems with Applications*, vol. 105, pp. 11–22, 2018.

[17] W. Bao, J. Yue, and Y. L. Rao, "A deep learning framework for financial time series using stacked autoencoders and long short term memory," *PLoS ONE*, vol. 12, no. 7, pp. 1–24, 2017.

[18] F. Thomas and K. Chrisstopher, "Deep learning with long short-term memory networks for financial market predictions," *Fau Discussion Papers in Economics*, vol. 270, no. 2, pp. 1–32, 2017.

[19] N. Makickiene, A. V. Rutkauskas, and A. Maknickas, "Investigation of financial market prediction by recurrent neural network," *Innovative Info Technologies for Science, Business and Education*, vol. 2, no. 11, pp. 1–24, 2011.

[20] L. D. Persio, "Recurrent neural networks approach to the financial forecast of Google assets," *International Journal of Mathematics and Computers in Simulation*, vol. 11, pp. 1–7, 2017.

[21] C. L. Dunis, J. Laws, and B. Evans, "Trading futures spread portfolios: applications of higher order and recurrent networks," *European Journal of Finance*, vol. 14, no. 6, pp. 503–521, 2008.

[22] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies," *Expert Systems with Applications*, vol. 83, pp. 187–205, 2017.

[23] C. Krauss, A. Do, and N. Hucket, "Deep neural networks, gradient-boosted trees, random forests: statistical arbitrage on the S and P500," *European Journal of Operational Research*, vol. 259, pp. 689–702, 2017.

[24] T.-J. Hsieh, H.-F. Hsiao, and W.-C. Yeh, "Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm," *Applied Soft Computing*, vol. 11, no. 2, pp. 2510–2525, 2011.

[25] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, no. 1, pp. 11–24, 2014.

[26] V. Vella and W. L. Ng, "Enhancing risk-adjusted performance of stock market intraday trading with Neuro-Fuzzy systems," *Neurocomputing*, vol. 141, pp. 170–187, 2014.

[27] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[28] M. Dixon, "Sequence classification of the limit order book using recurrent neural networks," *Journal of Computational Science*, vol. 24, pp. 277–286, 2018.

[29] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.

[30] G. Shen, Q. Tan, H. Zhang, P. Zeng, and J. Xu, "Deep learning with gated recurrent unit networks for financial sequence predictions," *Procedia Computer Science*, vol. 131, pp. 895–903, 2018.

[31] O. B. Sezer, M. Ozbayoglu, and E. Dogdu, "A deep neural-network based stock trading system based on evolutionary optimized technical analysis parameters," *Procedia Computer Science*, vol. 114, pp. 473–480, 2017.

[32] H. P. Hu, L. Tang, S. H. Zhang, and H. Y. Wang, "Predicting the direction of stock markets using optimized neural networks with Google Trends," *Neurocomputing*, vol. 285, pp. 188–195, 2018.

[33] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Fau Discussion Papers in Economics*, vol. 270, no. 2, pp. 1–32, 2017.

[34] D. D. Lv, Z. H. Huang, M. Z. Li, and Y. Xiang, "Selection of the optimal trading model for stock investment in different industries," *PLoS ONE*, vol. 14, no. 2, 2019.

[35] R. Pardo, *The Evaluation and Optimization of Trading Strategies*, John Wiley and Sons, Hoboken, New Jersey, NJ, USA, 2nd edition, 2008.

[36] B. Lantz, *Machine Learning with R*, Packt Publishing, Birmingham, UK, 2nd edition, 2015.

[37] I. Aldridge, *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems*, John Wiley and Sons, Hoboken, New Jersey, NJ, USA, 2nd edition, 2014.

[38] M. Hollander and D. A. Wolfe, *Nonparametric Statistical Methods*, John Wiley and Sons, Hoboken, New Jersey, NJ, USA, 1973.

[39] P. B. Nemenyi, *Distribution-free multiple comparisons [Ph.D. dissertations]*, State University of New York, 1963.

[40] BaiKe, "Securities Transaction cost," 2018, https://baike.baidu.com/item/.

[41] GuCheng, "How to calculate the transaction cost of American stock," SouthMoney, 2016, http://www.southmoney.com/zhishi/gprm/970810.html.

[42] J. Moody, L. Wu, Y. Liao, and M. Saffell, "Performance functions and reinforcement learning for trading systems and portfolios," *Journal of Forecasting*, vol. 17, no. 56, pp. 441–470, 1998.

[43] JoinQuant, "White Horse Stock Strategy," Xueqiu, 2017, https://xueqiu.com/8287840120/87475118.

[44] Abu, "Quantitative Investment: Slippage Strategy and Transaction Cost," Chinese Software Developer Network (CSDN), 2017, https://blog.csdn.net/bbfamily1314/article/details/78284986.