Comparison of Chinese 50 ETF put option pricing based on four algorithms

Lambert Lin*
linbei@stu.scu.edu.cn
Financial engineering
Chengdu, China

Yunze Dong
University of Bristol
UK
al18910@bristol.ac.uk

Valklyrs Ye
NingboTech University
China
Yeahruith@163.com

ABSTRACT

The article starts with the traditional Black-Scholes(B-S) option pricing models. Three more models: Long-term and short-term memory networks(LSTM), support vector machine (SVM) and random forest(RF) are introduced to be compared to the B-S model and to each other on 50 ETF put option pricing. It is showed that each model has its advantages when used in different position. The neural network pricing result is better than that of the B-S model From the four evaluation indicators of MD, MSD, MAD and MPD, the absolute values of the four errors of the prediction results of the neural network are all smaller than the absolute values of the corresponding errors of the prediction results of the B-S model.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Option pricing, LSTM, Neural networks, Support vector regressions, Random Forest, 50ETF

ACM Reference Format:
Lambert Lin, Yunze Dong, and Valklyrs Ye. 2021. Comparison of Chinese 50 ETF put option pricing based on four algorithms. In *The 2021 12th International Conference on E-business, Management and Economics (ICEME 2021), July 17–19, 2021, Beijing, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3481127.3481246

1 INTRODUCTION

1.1 Black-Scholes model

Black-Scholes model is a mathematical model in the financial market, which can be used to price European options. The partial differential equation of derivative Black-Scholes is the equation that satisfies the price of every derivative related to the underlying asset. In any short period of time, it can be considered that there is a perfect correlation between the price of the derivative product and the price of the underlying asset. After the establishment of a combination of the derivative product and the underlying asset, the profit or loss caused by the change in the price of the underlying asset can always be offset by the profit or loss brought by the derivative product, so the price change of such a portfolio in a short period of time is determined. The return on this trading portfolio is a risk-free return. We can get the Black-Scholes equation from this Cymbals.

Other assumptions of B-S model:

(1) there is no dividend payment during the option validity period
(2) risk-free interest rate r is constant,
(3) transaction costs and tax costs are not considered and are highly divisible
(4) there is no risk-free arbitrage opportunity in the market
(5) investors can borrow with the same r, and securities trading is continuous.

Many researchers have proposed a lot of changes, but still no changed model can simulate the behavior of the actual option price. From the research results at that time, we can see that the formula derivation of B-S model is carried out under a large number of assumptions, but these conditions often do not exist in practical application. In the case that there is no definite analytical solution of the hypothetical model, the numerical solution can be used to approximate the analytical solution, such as binary tree option pricing method and Monte Carlo simulation method. This kind of option pricing model is proposed by economists and requires certain assumptions to determine the model structure according to some market information.

Market participants change their option pricing strategies from time to time. Parametric option pricing model may not be able to adapt to this rapidly changing market behavior, so it is particularly important to develop non-parametric parameter pricing techniques that can overcome such limitations.

Garcia and Timmermann found that when the same variable as the B-S formula is used as the input variable of the feedforward neural network model, the option pricing prediction value of the neural network has a smaller mean. Das , Phani , Ludovic study the machine learning pricing methods of European options and American options from the point of view of traders, combining reinforcement learning models. AnderSen also proposed the option pricing method by using the utility maximization pricing method.

1.2 Deep learning

However, researchers found that using neural Networks to predict option prices is easy to overestimate options with longer execution periods and underestimate options that are about to expire. because of the over-fitting of neural networks.

ICEME 2021, July 17–19, 2021, Beijing, China

ACM ISBN 978-1-4503-9006-4/21/07…$15.00
https://doi.org/10.1145/3481127.3481246

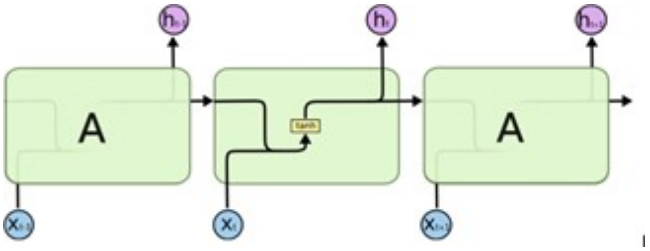

Figure 1: Parameter optimization of Support Vector Machine based on improved grid search method.

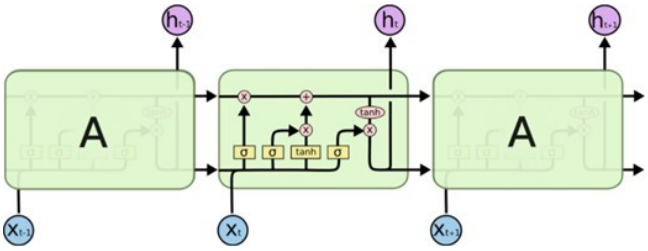

Figure 2: The repeating module in a LSTM contains four interacting layers.

Deep learning methods are more suitable for high- dimensional and nonlinear complex data than traditional machine learning methods. At the same time, financial data are often time series data while recurrent neural network (RNN) is a depth neural network specially designed to deal with series data. Therefore, according to the characteristics of option pricing, Author introduces LSTM model, improved SVM model and Random forest model into option pricing, and then makes an empirical analysis by using Chinese 50ETF options to compare the application prospects of these models in the pricing of financial derivatives.

2 LSTM MODEL

Long-term and short-term memory networks-often referred to only as "LSTM"-are a special kind of RNN, that can learn long-term rules. They were first proposed by Hochreiter & Schmidhuber (1997) and refined and promoted by many people in later work. They are used very well on a variety of problems and are now widely used. LSTM is specifically designed to avoid the problem of long-term dependency. Memorizing information for a long time is actually their default behavior, not something difficult for them to learn! All recurrent neural networks have the form of repetitive module chain of neural networks. In standard RNN, the repeating module will have a very simple structure, such as a single tanh layer.

LSTM also has this chain-like structure, but repetitive modules have different structures. There are four, rather than one neural network layer, as shown in the following Fig. 2.

LSTM can remove or add information to nodes to change the state of information flow, which is regulated by a structure called gate. A gate is a node that can choose to pass through information. They are composed of Sigma neural network layer and point-by-point multiplication operation. The sigmoid layer outputs numbers

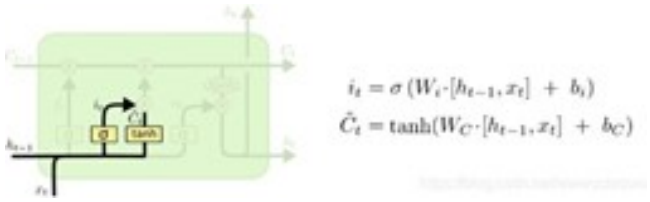

Figure 3: Forgetting gate.

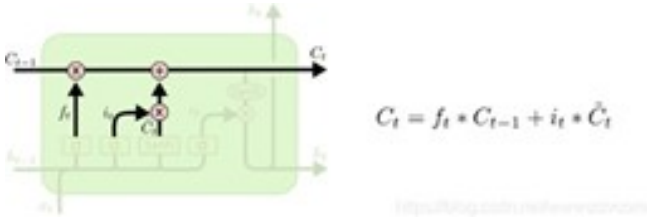

Figure 4: Input gate.

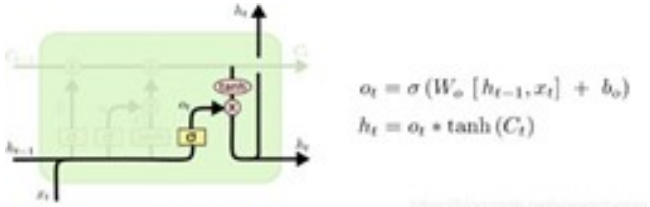

Figure 5: Output gate.

between 0 and 1, describing how much each information vector should pass. LSTM has three such gates for protecting and controlling the state of information flow vectors.

Forgetting gate: (1) the hidden state of the upper layer is merged with the input of the current layer, and the output of 0meme1 is obtained through the sigmoid function to update the C_{t-1} of the upper layer, that is, to control the C_{t-1} of the old state. After the merger of h_{t-1} and x_t, the input is controlled between-1 and 1 through a function, which is equivalent to a process of data normalization, and then the choice of the new state of C_t is decided by it.

Input gate: (2) according to the formula, it is obvious that $C_t * i_t$ controls the input of the new candidate value, $f_{t-1} * C_{t-1}$ controls the input of the old state and the addition of the two results in the output of the new cell state, as the final cell state C_t.

Output gate: (3) C_t contains the information input from the previous layer and the current layer, so the result processed by C_t through the tanh function to obtain the final output result needs to be filtered.

2.1 Model advantages

2.2 Compare to B-S model

B-S partial differential equation of European option Price is

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0 \quad (1)$$

The BS partial differential equation BS model to obtain the price changes of European options needs to be decomposed by a random process under the assumption of Brownian motion, so as to construct a complex and sophisticated mathematical formula with a definite relationship between variables. The LSTM neural network only pays attention to the rationality of the internal structure of the neural network model itself, and does not have any requirements for the structure between variables. That is to say, under the same variable conditions, the option pricing model constructed by the LSTM neural network is similar to the B- The partial differential equations derived by S may not have any structural similarities. Therefore, in terms of financial asset price prediction, it will be of great theoretical and practical significance to use algorithmic models to compare with traditional data models to analyze the pros and cons of the two models.

2.2.1 Advantages in deep learning. In the field of deep learning, although multi-layer perceptrons have outstanding performance in many aspects, they cannot analyze the overall logical sequence between input information. However, these sequence data often contain a large amount of content, and this information has complexities between each other. Time relevance, and the length of the information is also different, this is a problem that the traditional multilayer perceptron cannot solve. The recurrent neural network is created to solve a series of problems. In a recurrent neural network, the output of a sequence is related to the previous input. The network will memorize the previous information and use it in the calculation of the current output, that is, there are connections between hidden layer nodes. This network structure shows more Good predictive ability. Recurrent neural networks can process sequences of any length by using neurons with self-feedback, which is more in line with the structure of biological neural networks. Traditional neural networks cannot share the parameters of each network layer, but in the recurrent neural network structure, each layer shares the parameters, which greatly reduces the parameters that need to be learned in the entire network. Fig. 2 is an expanded recurrent neural network structure diagram, where x_t is the input at a certain time, which is an n-dimensional vector, ht represents the hidden state at time t, and ot represents the output at time t, and the weight from the input layer to the hidden layer is U, the weight from the hidden layer to the hidden layer is W, and the weight from the hidden layer to the output layer is V.

From the introduction of the above model, we can see that the Bmurs model needs to decompose the stochastic process under the assumption of Brownian motion, so as to construct a complex and precise mathematical formula, and there is a definite relationship between variables. On the other hand, the LSTM neural network only pays attention to the rationality of the internal structure of the neural network model, but does not have any requirements for the structure between variables, that is to say, under the same variable conditions, the option pricing model constructed by LSTM neural network may not be structurally similar to the partial differential equation (14) derived by Bmurs. Therefore, in the aspect of financial asset price prediction, it is of great theoretical and practical significance to compare the algorithm model with the traditional data model so as to analyze the advantages and disadvantages of the two models.

3 SVM MODEL

Support vector machine (SVM) is a new type of learning machine based on statistical learning theory, which was first proposed by the former Soviet Union professor Vapnik. Unlike traditional learning methods, support vector machines are an approximate implementation of structural risk minimization methods. This inductive principle is based on the fact that the error rate of the learning machine on the test data (that is, the generalization error rate) is bounded by the sum of the training error rate and a term that depends on the Vc dimension (Vapnik-Chervonenkis dimension); In the case of separable mode, the support vector machine has zero value for the previous term and minimizes the second term. Therefore, although the support vector machine does not use the domain knowledge of the problem, it can still provide good generalization performance on the pattern classification problem. This attribute is unique to the support vector machine. What it achieves is the following idea: the input vector x is mapped to a high-dimensional feature space z through a certain pre-selected nonlinear mapping, and the optimal classification hyperplane is constructed in this space, so that the positive and negative examples The separation limit is maximized. Conceptually, support vectors are the data points closest to the decision plane, and they determine the location of the optimal classification hyperplane.

3.1 Improved SVM

Let the sensory characteristic data be N-dimensional, which consists of L groups of data, and the decision-making surface can be expressed as

$$f(x) = \bar{\omega} \cdot g(x) + b \tag{2}$$

In order to minimize structural risk, the optimal classification hyperplane should meet the following conditions

$$y_i \left(\bar{\omega} \cdot g\left(x_i\right) + b\right) \geq 1 \tag{3}$$

When the non-negative relaxation variable ξ_i

$$\begin{cases} \min \frac{1}{2}\|\bar{\omega}\|^2 + c\sum_{i=1}^{n} \xi_i, c \geq 0 \\ \text{s.t } y_i\left[\left(\bar{\omega} \cdot g\left(x_i\right) + b\right)\right] \geq 1 - \xi_i, \xi_i \geq 0 \end{cases} \tag{4}$$

C is the penalty factor, which controls the complexity and generalization ability of the model.

if the Lagrange algorithm is introduced, the optimization problem will be transformed into dual form.

$$\begin{cases} \min \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n} y_i y_j a_i a_j K\left(x_i, x_j\right) - \sum_{i=1}^{n} a_i \\ \text{s.t } \sum_{i=1}^{n} y_i a_i = 0, 0 \leq a_i \leq c \end{cases} \tag{5}$$

$$K\left(x_i, x_j\right) = \left(g\left(x_i\right) \cdot g\left(x_j\right)\right) \tag{6}$$

When introducing the RBF kernel function

$$K\left(x_i, x_j\right) = \exp\left(-g\left\|x_i - x_j\right\|^2\right. \tag{7}$$

G is the kernel function parameter, which controls the range of the input space.

The above optimization problem is transformed into

$$\begin{cases} \min \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j a_i a_j \exp\left(-g\left\|x_i - x_j\right\|\right)^2 - \sum_{i=1}^{n} a_i \\ \text{s.t} \quad \sum_{i=1}^{n} y_i a_i = 0, 0 \leq a_i \leq c \end{cases} \tag{8}$$

It can be seen that the optimization problem depends on two important parameters c and g, which will affect the prediction performance of SVM.

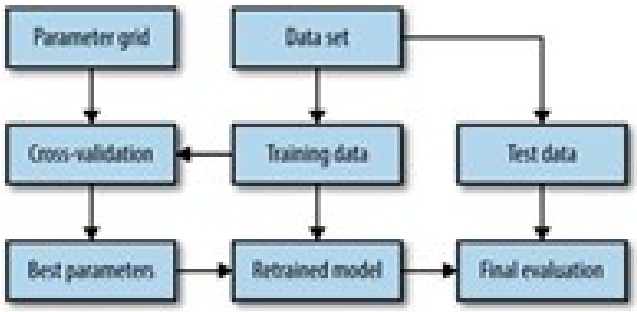

Figure 6: The process of parameter selection and model evaluation with Grid Search.

In order to improve the prediction performance of the model, the grid search method (GS) is introduced to optimize two important parameters in the process of establishing the model.

1. principal: Grid Search is a parameter adjustment method, that is, exhaustive search: in all candidate parameter choices, try each possibility through loop traversal, and the best-performing parameter is the final result.

3.2 Advantages

At the same time, to avoid the phenomenon of over-learning and under- learning of the model, the five-fold cross-validation method is used to optimize the parameters with the minimum root mean square error of the training set as the fitness function. (After the original data set is divided into training set and test set, the test set is not only used as adjustment parameters, but also used to measure the quality of the model,which leads to the final score result better than the actual effect.)When the minimum root mean square error is reached, the obtained c and g are the best parameters.

Five-fold cross-validation method: divides the original data into group of K (K-Fold), makes a verification set for each subset data, and takes the rest of $K-1$ subset data as a training set, so that K models can be obtained. The K models are evaluated in the verification set respectively, and the final error MSE (Mean Squared Error) is added and averaged to get the cross- validation error.

LIU Xiaosheng, ZHANG Zhibang used particle swarm optimization (PSO) to search the optimum solution; then uses grid search method for fine search with small length to get the final solution. The results show that the new algorithm has better prediction accuracy and shorter computation time compared with the traditional grid method.

The traditional parameter optimization method of support vector machine has not been recognized as a unified optimal method, and various improved methods have their own advantages and disadvantages.

4 RANDOM FOREST MODEL

As a newly emerging and highly flexible machine learning algorithm, Random Forest, or RF for short, has a wide range of application prospects, from marketing to health care insurance. It can be used to model marketing simulations and count customer sources.

	Traditional grid search method SVM		Improved grid search method SVM	
	Accuracy rate /%	time /s	Accuracy rate /%	time /s
Group 1	85.7143	227.81	85.7143	26.21
Group 2	96.0000	30.18	98.6667	4.50

Figure 7: LIU Xiaosheng, ZHANG Zhibang's experiment result.

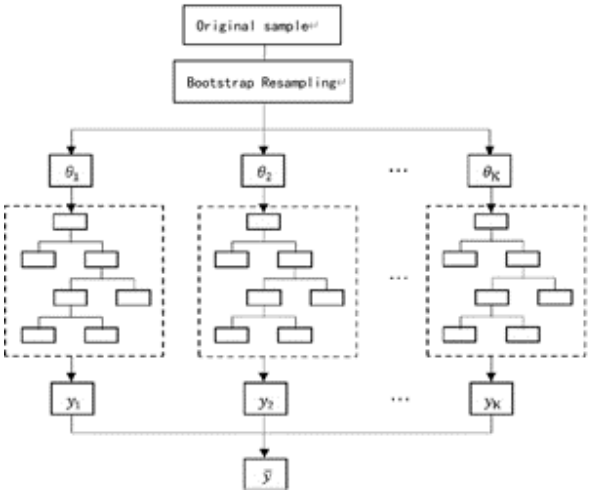

Figure 8: Schematic of random forest algorithm.

Retention and loss can also be used to predict the risk of disease and the susceptibility of patients.

4.1 Principle

1) According to whether the variables are discrete or continuous, random forest algorithm can be divided into random forest classification.

(RFC) algorithm and random forest regression (RFR) algorithm. The ideas and construction methods of the two are basically the same. Both use the Bootstrap re-sampling method to extract multiple subsample sets from the original sample set, and construct the decision tree for each subsample set respectively. The decision tree construction algorithm uses the CART method proposed by Breiman in 1984. The difference lies in the different rules adopted by the classification and regression algorithm when each leaf node of the decision tree is split. In the prediction, each decision tree will output its own prediction results based on the input variables, and then get the final results according to certain rules. The minority is subordinate to the majority in the random forest classification algorithm, and the average value is used as the predicted value in the random forest regression algorithm. The basic structure of the random forest algorithm can be shown in Figure 8.

2) Decision tree

Decision tree is a tree structure in which each internal node represents a test on an attribute, each branch represents a test output, and each leaf node represents a category. Common decision tree algorithms are C4.5, ID3 and CART.

3) Integrated learning

Integrated learning solves a single prediction problem by building a combination of several models. It works by generating multiple classifiers/ models, each of which learns and makes predictions independently. These predictions are finally combined into a single prediction, so it is better than any single category to make predictions.

4) The generation of random forest

As mentioned earlier, there are many classification trees in random forests. To classify an input sample, you need to input the input sample into each tree for classification. To make a vivid metaphor: a meeting is held in the forest to discuss whether a certain animal is a mouse or a squirrel. Each tree must independently express its views on this issue, that is, each tree must vote. Whether the animal is a mouse or a squirrel depends on the voting situation. The category with the most votes is the classification result of the forest. Each tree in the forest is independent, and the predictions made by 99.9% unrelated trees cover all situations, and these predictions will cancel each other out. The prediction results of a few excellent trees will be beyond the “noise" and make a good prediction. The classification results of several weak classifiers are voted to form a strong classifier. This is the idea of random forest bagging. If the size of the training set is N, for each tree, N training samples are randomly selected from the training set with replacement, as the training set of the tree

5) Advantages

1. Based on the theorem of large numbers, the problem of overfit does not occur easily, and the generalization error is usually less than the error of the decision tree.
2. because parallel operations can be implemented, the learning speed is fast, and it is still very efficient when dealing with a large number of data.
3. it can deal with a large number of as many as thousands of independent variables.
4. It can evaluate the importance of all variables without worrying about the problem of multiple collinearity faced by general regression problems.
5. It has a high tolerance for missing values and does not need to do any processing on the missing data. it can still maintain a certain degree of accuracy; it is not sensitive to outliers and is robust in the case of more random interference.
6. The classification tree algorithm in random forest naturally includes the interaction of variables (Interaction) (Cutler,etal.,2007) [31]. This interaction is often ignored in traditional models because of its complexity.
7. Random forests can estimate the error of the model by out-of-bag error For the classification problem, the error is the error rate of the classification; for the regression problem, the error is the variance of the residual. Each classification tree in a random forest is generated after re- sampling the original records. About 1 stroke 3 records per re-sampling were not extracted. Those that are not extracted naturally form a control data set. Therefore, the random forest does not need to reserve another part of the data for cross-validation, its own algorithm is similar to cross-validation, and the out-of-bag error is the unbiased estimation of the prediction error

4.2 Random forest option pricing model parameters

There are three more important parameters in the random forest model:The first is mtry, which is the number of candidate feature variables extracted at each node of the decision tree during the construction of the decision tree;The second is nodesize, which is the minimum size of the data set at the node of the decision tree as a criterion for judging when the decision tree stops growing during the decision tree generation processThe third is ntree, which is the number of decision trees in the basic structure of the random forest algorithm, ntree.These three parameters have a greater impact on the goodness of fit of the random forest and the generalization error of the model. According to the previous definition, the goodness of fit of the random forest is the R square after bringing all the samples back into the random forest for prediction. The generalization error satisfies the following expression, where it measures the correlation between trees and measures the classification effect of the model. It can be seen from the formula that the lower the correlation between the tree and the tree and the better the classification effect of the decision tree, the smaller the upper bound of the generalization error:

5 EMPIRICAL RESULTS

5.1 Selection of data samples and variable

The original data sample comes from the daily frequency data of flush database, covering a total of 769 trading days from the first day of listing of 50ETF options on February 9, 2016 to April 4, 2019. The options in the original sample cover all existing and expired 50ETF options.

On top of the original sample, the sample is screened and adjusted, excluding the option data with daily trading volume of less than 100, as this part of the options are mainly deep imaginary options. The options adjusted by dividends are excluded as well, the adjustment of dividends as 50ETF options involves the revision of exercise price, unit share and transaction price. If this part of the data is included in the sample, it will increase the complexity of the model.

A total of 10 explanatory variables are selected. In the depiction of the underlying asset 50ETF. Author uses 5-day, and 180-day annualized volatility to describe the volatility level and volatility changes of the underlying assets.

5.1.1 Programming. Author use Pathon ane Matlab code to program LSTM, RF, SVM models The code can be found in Appendix

5.2 Comparison

Researchers usedd four models to empirically test European put options, the models are: BS parameter model; support vector machine model; LSTM model, random forest prediction model.

Among them, the support vector machine model uses the standard Black- Schoks model parameters as input, and uses the RBF kernel function. The grid search method is used for cross-validation and the model parameters are selected.

The cross-comparison of different methods shows that the hybrid model using the SVM-BS method is better than the pure BS classic

Table 1: Explanatory variable of SSE 50ETF Option

Explanatory variable	Abbreviation
Option exercise price	STRKP
Remaining expiration time	EXP
Option stock	HLDV
50ETF closing price	UDP
Range of rise and fall of 50ETF in the past 5 trading days	UNYLD5D
50ETF turnover rate of the day	TRNVR
Annualized volatility of 50ETF in recent 5 trading days	UDVLT5D
Risk-free interest rate	RATE
Annualized volatility of 50ETF in recent 180 trading days	UDVLT180D
One-year rate of return	RATE1Y

Table 2: Comparison table of put option on B-S model and LSTM model

Model	MSE	RMSE	MAE	MAPE
B-S Monte Carlo	0.001638	0.010472	0.034571	0.296554
LSTM	0.000251	0.013348	0.013348	0.114719

model, while the random forest model predicts the short-term trend better.

In order to compare the prediction effects of the model, the experiment will be conducted separately for call options and put options. In this experiment, a total of 280 call option price predictions and 372 put option price predictions were obtained. According to the characteristics of the 50ETF option, the input of the model is: 50ETF opening price (S), the option expiration exercise price, the 6-month treasury bond yield, the option expiration time/365, 90-day historical volatility. The output of the model is: option forecast value. In the tensorflow deep learning environment, a two-layer LSTM network structure is constructed, the loss function judgment standard adopts the mean square error, the parameter learning algorithm adopts stochastic gradient descent, and each mini-batch is set to 30. Epoch = 10

5.2.1 LSTM and B-S. (1) The neural network pricing result is better than that of the B-S model From the four evaluation indicators of MD, MSD, MAD and MPD, the absolute values of the four errors of the prediction results of the neural network are all smaller than the absolute values of the corresponding errors of the prediction results of the B-S model. Therefore, it can be seen from the research that BP neural network has its own superiority in warrant prediction. This is exactly in line with our financial theory. Because parametric models such as B-S have a series of assumptions and preconditions, these assumptions are often not in line with the reality of the market, so there is a large pricing deviation. However, non-parametric models such as artificial neural networks do not require any assumptions, as long as there are more training data. Nowadays, the social financial data with highly developed information is very rich and easy to obtain. After reasonable processing, it can be established. Market data-driven model, and achieved better pricing results.

Table 3: Comparison table of put option on improved SVM model and LSTM model

Model	MSE	RMSE	MAE	MAPE
Improved SVM	0.000505	0.022903	0.03449	0.183513
LSTM	0.000251	0.013348	0.013348	0.114719

The long-term prediction accuracy of the neural network model is less than the short-term prediction accuracy.

When the trained ANN model makes predictions, it can be found that the longer the backward prediction time, the greater the error. Make a backward forecast when the difference between the predicted value and the true value is minimal. This is also consistent with previous studies. Neural networks often achieve very good results in short-term predictions, while long-term predictions are not effective. Therefore, when using neural networks for prediction, it is very important to use a shorter prediction period.

The model structure is optimal when the number of neurons in the hidden layer is twice the number of input variables.

When the number of neurons is 6, the absolute value of each error index is the smallest. The effect of 5 neurons with similar structure is also relatively good. When far away from the structure, the various error indicators increase. This also shows that it is not that the more neurons, the better, but that there is an optimal number of neurons. Studies have found that the number of neurons predicts the best effect according to the rules proposed by Wong.

5.2.2 Comparison of improved SVM model and LSTM. (1) The prediction accuracy of the model is related to the number of deep learning trainings. From the perspective of the LSTM network itself, as the number of model training increases, the loss function gradually decreases and tends to converge, and the prediction accuracy begins to rise. It can also be found from the experimental results that, for both call options and put options, the predictive effect of the model after 100 times of training is far higher than that of the model after 10 times of training.

(2) The experimental results of option pricing methods based on deep learning algorithms are better than SVM vector machine model option pricing methods. First of all, from the comparison of the prediction results, the MSE of the call option predicted by the LSTM neural network under the condition of 100 iterations is 0.00000 251. Secondly, the predicted value of the put option LSTM MSE is 0.00 617, which is also smaller than the value of the SVM method. In the long run, the improved SVM method also has advantages. The parameter optimization based on the improved grid search method SVM proposed in the article has been proved by experiments that the improved algorithm can take advantage of the global particle swarm algorithm to quickly converge to the optimal interval and accelerate the grid The early search speed of the search algorithm, and the use of fine search with small steps in the later search, further improve the classification accuracy. The results obtained also prove that the improved grid search method SVM has faster search speed.

5.2.3 Evaluation of the random forest model. Although the random forest algorithm can be used for regression problems, the basicidea of the algorithm is classification. In the process of random forest

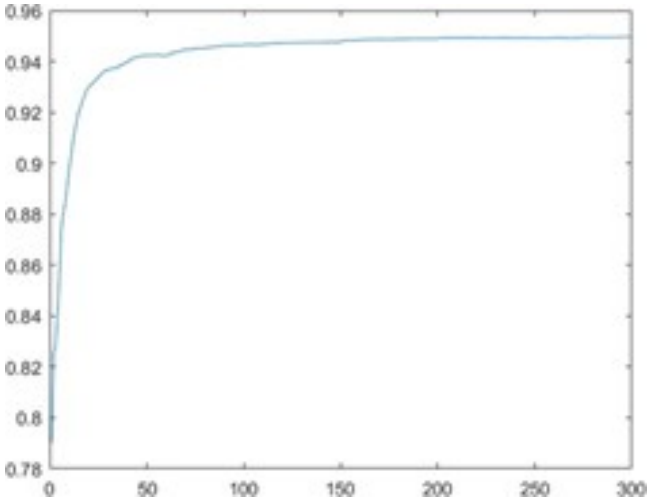

Figure 9: Goodness of fitting of random forest model under nodesize=5, mtry=16

operation, the characteristic value of the object is input into the decision tree, and the object is classified at the node according to the split feature and its value used by the node, and this process is repeated again and again, and finally reaches the root node of the decision tree. as a result, the classification result of the decision tree of the object is obtained. For the random forest model, the variable selection and optimal value of each node of the decision tree are obtained through the training of training samples, which means that the variable value of each node for segmentation can not exceed the range of training sample variables. Therefore, if the characteristic value of the variable used for the algorithm prediction is beyond the range of the training sample variable, the accuracy of the prediction will be greatly reduced.

6 CONCLUSIONS

At present, the application of artificial intelligence in the financial field has become a research hotpot, and the pricing of financial derivatives under machine learning methods is particularly worthy of attention. However, the existing research mainly focuses on the use of traditional feed forward neural networks for option pricing research, and neural networks have not been widely used in option pricing research due to the black box calculation and poor generalization ability of neural networks. The deep learning algorithm has a good model generalization ability, and according to recent research literature, it can also be found that the accuracy of deep learning in financial asset price prediction has surpassed the traditional financial measurement model.

It is necessary to biuld an application pricing model that combines model applications with different requirements such as time-consuming, data sample size, factor type and quantity, predicted window period, forecast content, etc., in order to achieve higher accuracy and fit Forecast requirements for application.

REFERENCES

[1] Henrik Amilon. 2003. A neural network versus Black–Scholes: a comparison of pricing and hedging performances. *Journal of Forecasting* 22, 4 (2003), 317–335.
[2] Henrik Amilon. 2003. A neural network versus Black–Scholes: a comparison of pricing and hedging performances. *Journal of Forecasting* 22, 4 (2003), 317–335.
[3] Fischer Black and Myron Scholes. 2019. The pricing of options and corporate liabilities. In *World Scientific Reference on Contingent Claims Analysis in Corporate Finance: Volume 1: Foundations of CCA and Equity Valuation*. World Scientific, 3–21.
[4] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.
[5] Christine A Brown and David M Robinson. 2002. Skewness and kurtosis implied by option prices: A correction. *Journal of Financial Research* 25, 2 (2002), 279–282.
[6] F Dan Foresee and Martin T Hagan. 1997. Gauss-Newton approximation to Bayesian learning. In *Proceedings of international conference on neural networks (ICNN'97)*, Vol. 3. IEEE, 1930–1935.
[7] Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, Vol. 1. IEEE, 278–282.
[8] Tin Kam Ho. 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 8 (1998), 832–844.
[9] Robert W Kolb and James A Overdahl. 2003. *Futures, options, and swaps*. Vol. 5. Blackwell.
[10] Daewon Lee and Jaewook Lee. 2007. Equilibrium-based support vector machine for semisupervised classification. *IEEE Transactions on Neural Networks* 18, 2 (2007), 578–583.
[11] Xiaosheng Liu and Zhibang Zhang. 2019. Parameter optimization of Support Vector Machine based on improved grid search method. *IEEE Transactions on Neural Networks* 40, 1 (2019), 5–9.
[12] Vladimir Vapnik, Esther Levin, and Yann Le Cun. 1994. Measuring the VC-dimension of a learning machine. *Neural computation* 6, 5 (1994), 851–876.

A LSTM CODE

```
data = pd.read_excel('put.xlsx',encoding='gbk')
data = data[['close', 'ETF closing price','Risk-free rate',
'Implied volatility', 'Volume','
Open interest']] number = data.shape[0]
data_columns = list(data.columns) values = data.values
values = values.astype('float32') n_features = values.shape[1] scaler, scaled =
data_scale(values) n_intervals = 1
reframed = series_to_supervised(scaled, n_intervals, 1)
drop_column_list = ['var'+str(i)+'(t)' for i in range(2, n_features+1)]
reframed.drop(drop_column_list, axis=1, inplace=True)
def split_data_to_train_test_sets(reframed, n_intervals, n_features, rate):
data_numbers = reframed.shape[0] values = reframed.values
n_train = int(data_numbers * rate ) train = values[:n_train, :]
test = values[n_train:, :]
train_X, train_y = train[:, :-1], train[:, -1]
test_X, test_y = test[:, :-1], test[:, -1]
train_X = train_X.reshape((train_X.shape[0], n_intervals, train_X.shape[1]))
test_X = test_X.reshape((test_X.shape[0], n_intervals, test_X.shape[1]))
return train_X, train_y, test_X, test_y
rate = 0.8

train_X, train_y, test_X, test_y = split_data_to_train_test_sets(reframed,
n_intervals, n_features, rate)
n_neurons = 60 loss_function = 'mse' optimizer_function = 'adam'
batch_size = 2 # batch_size: epochs = 100
def create_model(train_X, train_y, test_X, test_y, n_neurons, loss_function,
optimizer_function, batch_size, epochs):
```

B SVM CODE

```
format compact;
clc;
clear;
close all;
data1=xlsread('puttest');
data2=xlsread('putrun');
ts =  data1(:,end);
tsx = data1(:,1:end-1);
tts= data2(:,end);
ttx= data2(:,1:end-1);
tsx = tsx';
ttx=ttx';
[TSX,TSXps] = mapminmax(tsx,-1,1);
[TTX,TTXps] = mapminmax(ttx,-1,1);
TSX = TSX';
TTX = TTX';
[bestmse,bestc,bestg] = SVMcgForRegress(ts,TSX,-10,10,-10,10);
disp('printout');
str = sprintf( 'Best Cross Validation MSE = %g Best c = %g Best g = %g',bestmse,bestc,bestg);
disp(str);
cmd = ['-c ', num2str(bestc), ' -g ', num2str(bestg) , ' -s 3 -p 0.01'];
model = svmtrain(ts,TSX,cmd);
    N2=length(tts);
    [predict_2,mse_2] = svmpredict(tts,TTX,model);
    RMSE=sqrt((sum((tts-predict_2).^2))/N2)
    B=corrcoef(predict_2,tts);
    R2=B(2,1)
    MSE=RMSE*RMSE
    MAE=(sum(abs(tts-predict_2)))/N2
    MAPE=(sum(abs(tts-predict_2)./tts))/N2
figure;
                        plot(tts,'-o');
                        hold on;
                        plot(predict_2,'r-^');
                        legend('actual1','predict');
                        title('SVMpredict','FontSize',12);
                        xlabel('sample','FontSize',12);
                        ylabel('output','FontSize',12);
```

C RF CODE

```
clear all;
clc;
close all;
data1=xlsread('calltest');
data2=xlsread('callrun');
T_train =  data1(:,end);
P_train = data1(:,1:end-1);
T_test= data2(:,end);
P_test= data2(:,1:end-1);
ntree=%inputnumber;
model = regRF_train(P_train,T_train,ntree);
T_sim= regRF_predict(P_test,model);
figure
plot(T_test,'r:o');
hold on
plot(T_sim,'b:*');
```