Stock Volatility Forecasting: Adopting LSTM Deep Learning Method and Comparing the Results with GARCH Family Model

Tian Wang
* Corresponding author: FIN1909327@xmu.edu.my

Xiamen University Malaysia, Finance Malaysia, Kuala Lumpur, 266035

Abstract—As a booming industry, information technology has been applied to many other industries. The combination of finance and IT (financial technology) is one of the most representative mergers. Volatility is one of the most important indexes of all financial assets and it is hard to forecast using traditional financial method due to many uncertainties. This paper will use the improved LSTM network to forecast the US stock market, and compare the result with the actual data based on selected GARCH model. After a series of experiments, the predicted volatility is close to the actual volatility and LSTM is applicable in forecasting the stock volatility.

Keywords-LSTM, GARCH, Volatility

1 INTRODUCTION

The development of financial technology has had a deep impact on the financial sector and brought profound changes to it. Among them, after the full application of information processing technology in financial activities, it will have a significant impact on all links and fields of financial activities, playing a comprehensive role in a wide range of social and economic life, and more importantly, making the complicated predication and pricing process of the past feasible and applicable [1].

In this paper, the investigation will mainly focus on the prediction of the stock volatility. Volatility in finance refers to the degree of movement in asset prices and gauges the price's unpredictability. It is crucial in both academic research and the financial business. Volatility is a risk indicator in and of itself, as well as a component of various other indicators, such as the Sharpe ratio, in risk management and performance analysis. Markowitz [2] employed volatility in portfolio theory to quantify the hazards of assets and the total risk of the portfolio. The portfolio creation approach uses volatility as both an input and an optimization objective. Prices of derivatives can be dictated by the volatility of the underlying assets in derivative pricing [3].

2 LITERATURE REVIEW

Since the volatility is important and useful, the prediction of it gets more and more attention these years and there have been many approaches presented. The autoregressive conditional heteroscedasticity (ARCH) and generalized ARCH (GARCH) models proposed by Bollerslev [4]

FFIT 2022, October 28-30, Shenzhen, People's Republic of China

DOI 10.4108/eai.28-10-2022.2328447

are mainly used to predict volatility. Conditional variance is modeled as a function of prior mistakes and variances in this model. Nelson [5] developed an exponential generalized autoregressive conditional heteroscedasticity (EGARCH) model based on the GARCH model. Liu Yifei [6] conducts a research on the Price Volatility of China's Pulp Futures Market Based on GARCH Family Models in 2021.

Another general way to predict the volatility is through machine learning, or specifically, deep learning. Machine learning algorithms are more data-driven than econometric models, which are based on economic assumptions and statistical reasoning. Many publications used neural networks and the GARCH model to create a hybrid model, which was then used to anticipate volatility. Zhou Aimin and Guan Rui (2021) anticipate volatility using a deep learning method [7], also Marcelo Sardelicha & Suresh Manandhar (2018) [8] adopt multimodal deep learning method to predict the short term stock volatility.Among all the deep learning, the mostly used method is combining LSTM with GARCH to forecast volatility. Fang Jia and Boli Yang (2021) [9] use this approach combing comparison with the traditional econometric method with the likelihood-based loss function. However, few of them exam the related GARCH model such as EGARCH and TGARCH model Eduardo Ramos-Pérez & Pablo J. Alonso-González [10] (2021) covered the GARCH model but mainly compared with multi-transformer approach. Besides, the LSTM learning process is improvable.

This paper will firstly use the GARCH family model as the benchmark to build the result based on the whole period, and then compare the improved LSTM deep learning forecast result with the selected GARCH model for the forecast period.

3 METHODOLOGY

3.1 Data collection

The data used in this paper is collected from S&P 500 (S&P 500) Index. The author obtained the closing prices of the index from Dec 18, 2000 until Dec 15, 2021 through yahoo finance, 5282 observations in total.

3.2 Model and approach

1)GARCH Model. Bollerslev (1986) [4] proposed the GARCH model on the basis of the ARCH model and proved that the GARCH (1,1) model is applicable in most situations. The GARCH model is specifically defined as follows:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i r_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 \,/\, r_t = \sigma_t \epsilon_t \tag{1}$$

The GARCH model requires the data to be smooth, therefore, the daily return is calculated as the logarithms of relative daily closing prices, using the following equation [9]:

$$r_l = log(P_t/P_{t-1}) \tag{2}$$

Since the negative news may impact the stock index more than the positive news, EGARCH and TGARCH are included and the they are defined respectively as follows [10]:

$$\log\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \log\sigma_{t-i}^2 + \sum_{i=1}^{q} (\beta_i e_{t-i} + \gamma_i(|e_{t-i}| - E(|e_{t-i}|))$$

$$\sigma_t = \omega + \sum_{i=1}^{q} \alpha_i |r_{t-i}| + \sum_{i=1}^{0} \gamma_i |r_{t-i}| I_{[r_{t-i<0}]} + \sum_{i=1}^{p} \beta_i \sigma_{t-i} \tag{3}$$

where ω_i, α_i, β_i and γ_i are the parameters to be estimated and $e_t = r_t/\sigma_t$.

2)LSTM Deep Learning Approach. BP neural network and recurrent neural network (RNN) are currently the two most common types of machine learning models used for time series forecasting, but the signals transmitted by BP neural network can only flow in one direction, and the potential impact of data earlier in the time series on later data is not considered. Although RNN implements a weighted connection between the hidden layers on the basis of the BP neural network, the hidden layer at the next point in time can accept the information transmitted at the previous point in time, and can process the context of time series data. However, due to the inherent problems of the RNN model, the gradient in the training process will disappear, and the long-term data dependence cannot be handled. Therefore, these two types of networks are obviously not suitable for processing time series with long memory.

Therefore, LSTM is used. On the basis of RNN, LSTM adds a cell structure that determines whether data is retained or forgotten, which solves the problem of the disappearance of gradients in traditional RNNs, and can learn long-term dependent information well. This kind of cell structure automatically selects data retention and forgetting through three structures equivalent to "gates", namely input gate, forget gate and output gate. The structure of the "gate" is equivalent to a data screening process. Using the non-linear activation function sigmoid, a value from 0 to 1 is output according to the degree of activation. Therefore, when data passes through the "input gate", valid information can be retained and passed to the next moment. Afterwards, the useless information is filtered out through the "forgetting gate", so the time series data with long memory characteristics can be handled well [11].

The LSTM can be described using vector formulas as follows:

(1) Enter the sample data into the "forgotten door" layer. The purpose of this "door" is to eliminate useless information in the past. The calculation formula is:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

$$\sigma(t) = \frac{1}{1+e^{-t}} \tag{4}$$

Among them, σ represents the sigmoid function, W_f is the weight of the "forgotten layer"; x_t represents the current input time series, h_{t-1} represents the output at the previous moment, and b_f is the corresponding bias parameter. By multiplying x_t and h_{t-1} at the previous moment, an n-dimensional vector can be obtained, and then a new vector f_t can be calculated through the sigmoid function, and its value must be in the interval (0,1). After calculation, the lower activation value will be forgotten, and the higher activation value will be left behind.

(2) Input data into the "input gate" to add new information to the current input, which consists of three parts. First, the "input gate" layer uses sigmoid to determine the current input x_t and the last time output h_{t-1} to calculate the retained information i_t; then the tanh function is used to calculate

the input value at this time, and it is multiplied by the input value ~ C_t to get a new vector and add it to the cell state; finally, the "input gate" multiplies the old cell state by the "forgotten gate" to get f_t, in order to forget a part of the previous information and add new input information i_t* ~ C_t, forming a new cell state. The calculation formula is:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

$$\sim C_t = tanh(W_C[h_{t-1}, x_t] + b_C)$$

$$C_t = f_t C_{t-1} + i_t \sim C_t \quad (5)$$

(3) Use the new cell state C_t as the new output value. The initial output is calculated by sigmoid, and the process is not affected by the output of the information learned at the previous time. Use tanh to scale the C_t value to the interval (-1,1), which is the processing of the previously learned information. Then multiply the obtained value with the initial output pair by pair to get the output value. The output value obtained in this way is relatively stable.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t tanh(C_t) \quad (6)$$

Among them, W_o and b_o are the weight vector and bias parameter of the "output gate" layer, respectively.

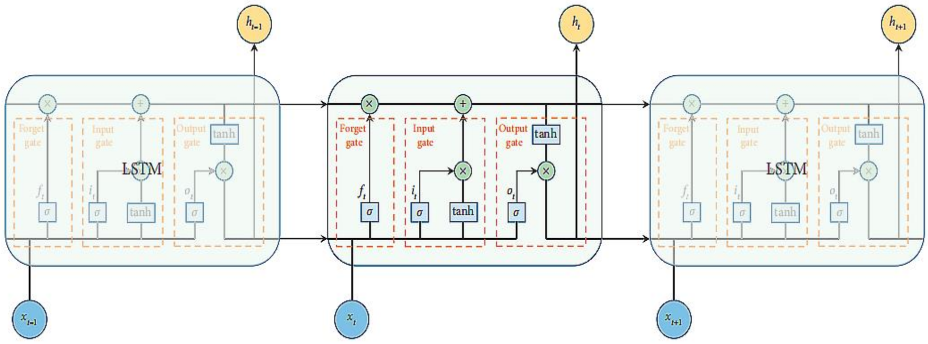

Figure 1. The framework of the LSTM model [9]

As compared with the previous researches that divide the whole process into two stages, that is, training and testing, this paper will include another stage called verification set to improve the testing result. The data is separated into three pieces according to a 7:1.5:1.5 ratio for training set, verification set, and test set respectively. The training set is used to train models, the verification set is used to fine-tune hyperparameters, and the test set is used to test models.

4 RESULTS AND DISCUSSION

4.1 GARCH analysis (via Eviews)

In order to fit the GARCH model, the time series must be smoothly processed. As mentioned above, the logarithms of relative daily closing prices are used to get the daily return as shown in figure 2.

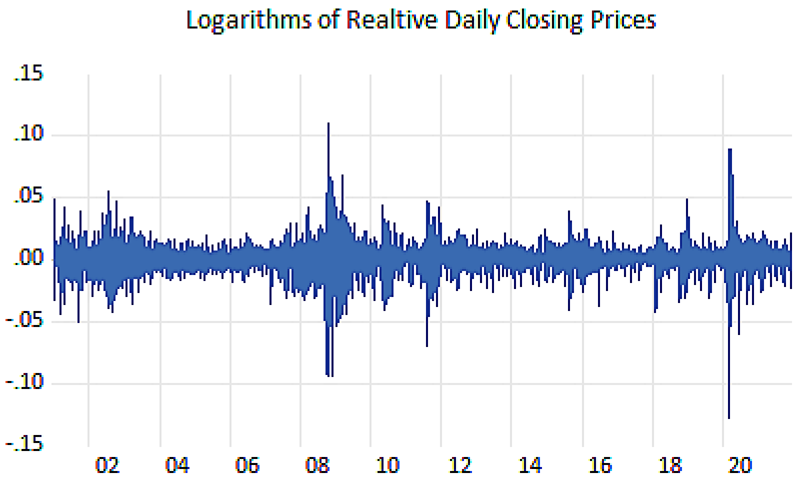

Figure 2. Daily Return of the S&P 500 Index

The LM test is needed to see whether the residual term is autocorrelated or not. Table 1 indicates the level of significance is extremely high, therefore, the residual is autocorrelated, meaning that the heteroskedasticity exists and GARCH model can be built.

Table 1 Residual Autocorrelation LM Test Result

F-Statistic	633.2260	Prob. F(1,5278)		0.000
Obs*R-squared	565.6075	Prob. Chi-Square(1)		0.000
Variable	Coefficient	Std. Error	T-Statistic	Prob.
C	0.000102	7.58E-06	13.47191	0.000
RESID^2(-1)	0.327297	0.013007	25.16398	0.000

The most common and classic GARCH (1,1) model is built using the normal distribution and the result is presented in Table 2 below. According to the result, the parameter of the

GARCH and ARCH are both significant and positive and the sum of them is less than one, which is reasonable because it indicates the influence of past fluctuations on the future gradually weakens.

Table 2 GARCH Model of the Selected Data

Variable	Coefficient	Std. Error	Z-Statistic	Prob.
C	2.52E-06	2.06E-07	12.24427	0.0000
RESID(-1)^2	0.127273	0.007396	17.20761	0.0000
GARCH(-1)	0.853084	0.007769	109.8065	0.0000
Akaike info criterion (AIC)	-6.502112			

Figure 3 represents the squared test after the GARCH model. It is apparent that the autocorrelation is not significant for any lag included, which means that the residual of the model is well extracted by GARCH (1,1).

Date: 12/16/21 Time: 20:14
Sample (adjusted): 12/19/2000 12/15/2021
Included observations: 5281 after adjustments

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob*
		1	-0.014	-0.014	0.9977	0.318
		2	0.024	0.024	4.1597	0.125
		3	0.005	0.006	4.2881	0.232
		4	0.020	0.019	6.3096	0.177
		5	-0.008	-0.008	6.6372	0.249
		6	-0.008	-0.009	6.9520	0.325
		7	-0.008	-0.008	7.2614	0.402
		8	0.003	0.003	7.2982	0.505
		9	-0.006	-0.005	7.4591	0.589
		10	0.038	0.038	15.037	0.131
		11	-0.005	-0.004	15.175	0.175
		12	-0.005	-0.007	15.315	0.225
		13	-0.007	-0.008	15.610	0.271
		14	0.000	-0.001	15.610	0.338
		15	-0.014	-0.013	16.668	0.339
		16	-0.001	-0.000	16.670	0.407
		17	-0.004	-0.003	16.765	0.470
		18	-0.016	-0.017	18.146	0.446
		19	0.004	0.004	18.222	0.508
		20	-0.002	-0.003	18.244	0.571
		21	-0.007	-0.007	18.497	0.617
		22	-0.001	-0.001	18.506	0.676
		23	0.008	0.008	18.828	0.711
		24	-0.024	-0.024	21.776	0.593
		25	-0.014	-0.015	22.890	0.584
		26	-0.027	-0.027	26.731	0.424
		27	-0.005	-0.006	26.878	0.470
		28	-0.011	-0.008	27.568	0.487
		29	-0.003	-0.003	27.609	0.539
		30	0.004	0.004	27.676	0.588
		31	-0.007	-0.007	27.911	0.626
		32	-0.011	-0.011	28.521	0.643
		33	0.014	0.012	29.615	0.636
		34	-0.013	-0.011	30.550	0.638
		35	0.005	0.004	30.663	0.678
		36	0.002	0.004	30.685	0.719

*Probabilities may not be valid for this equation specification.

Figure 3. Correlogram of Standardized Residuals Squared Test on the Built GARCH Model

Table 3 and Table 4 elucidate the results of EGARCH as well as TGARCH respectively (based on GARCH(1,1)). It can be seen that in EGARCH, the parameter of C(4) is negative and significant, meaning that bad news does have more impact on the stock index. TGARCH shows and similar result as EGARCH. From Table 5, the parameter of TGARCH item is positive and significant, therefore, bad news have more impact on the stock index compared with good news.

Table 3 EGARCH Model of the Selected Data

Variable	Coefficient	Std Error	Z-Statistic	Prob.
C(2)	-0.394238	0.022006	-17.91489	0.0000
C(3)	0.165471	0.009102	18.17893	0.0000
C(4)	-1.40440	0.005024	-27.95631	0.0000
C(5)	0.971403	0.001899	511.4370	0.0000
AIC	-6.539323			

Table 4 TGARCH Model of the Selected Data

Variable	Coefficient	Std. Error	Z-Statistic	Prob.
C	2.36E-06	1.48E-07	16.00065	0.000
RESID (-1)^2	0.007603	0.004289	1.772390	0.000
RESID (-1)^2*_RESID (-1)<0)	0.176345	0.009342	18.87635	0.000
GARCH(-1)	0.880401	0.006235	141.2007	0.000
AIC	-6.529965			

After considering both the level of significant and the Akaike Info Criterion (AIC) factors, EGARCH Model is finally chosen as the base model for further comparison between the estimated data from the LSTM forecast and actual data later, and the negative news do impact the stock index more than the positive news. The function of the EGARCH (1,1) is as follows:

$$\log\sigma_t^2 = -0.3942 + 0.9714\log\sigma_{t-1}^2 + 0.1655\,|e_{t-1}| - 0.1404 e_{t-1} \tag{7}$$

LSTM Approach (via Python)

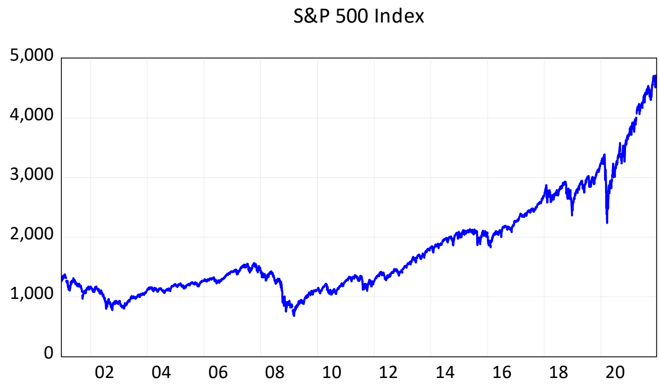

Figure 4. The Actual Closing Price of the S&P 500 Index

Table 5 Summary of the LSTM Model

Layer (type)	**Output shape**	**Param #**
Lstm (LSTM)	(None, 50, 96)	37632
Lstm_1 (LSTM)	(None, 50, 96)	74112
Lstm_2 (LSTM)	(None, 48)	27840
Dropout (Dropout)	(None, 48)	0
Dense (Dense)	(None, 5)	245
Total params: 139,829		
Trainable params: 139,829		
Non-trainable params: 0		

According to Table 5, when modeling the LSTM model, the author used a 4-layer network, with the first layer being the Lstm layer (dimension; 96), the second layer being the Lstm layer (dimension; 96), and the third layer being the Lstm layer (dimension; 48). The fourth layer is the dropout layer

(dropout=0.2, used to prevent over-fitting), the fourth layer being the fully connected layer as well (the number of neurons is 5). Estimate parameters adopts Adam optimizer. The learning rate employs the LR attenuation approach, and the maximum number of iterations is set to 100. The loss transformation trends of the training and verification sets are shown in Figure 5.

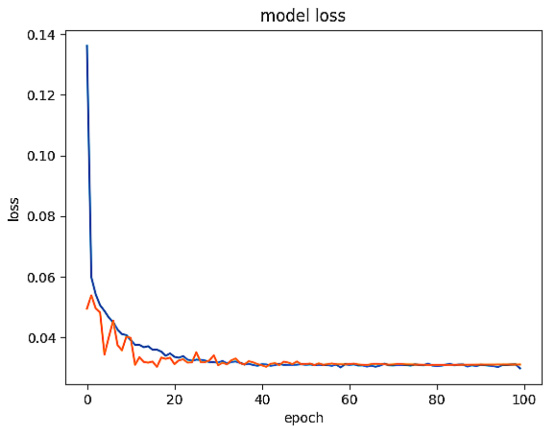

Figure 5. Loss Trend of the Learning Result

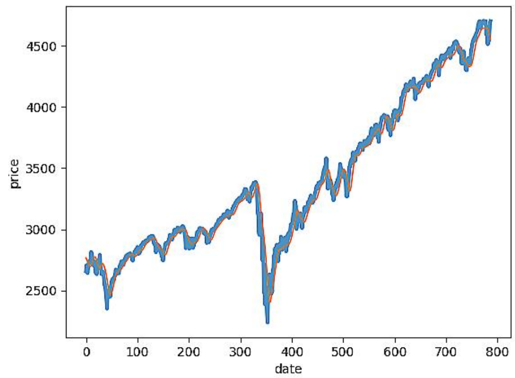

Figure 6. Comparison Between the Prediction and Actual Index

It is shown that the prediction effect on the test set is not awful, and the model has no over-fitting after hyper-parameter adjustment. In terms of model evaluation, the author uses the MAE criterion as the evaluation standard, the Naive estimation (using the S&P 500 Index of the previous day as the estimated value of the next day) as the base model, and finally obtains the Lstm model prediction result due to the Naive estimation of the base model. MAE is obtained as 79.129, since nearly all of the S&P 500 index are four-digit number, therefore, the deviation here is acceptable.

Figure 6 shows the prediction (orange line) and actual (blue line) index for the testing data set. It can be seen that after adding a validation set to LSTM, the error at the peak is reduced a lot compared with many previous studies. The prediction of the daily return can also be get as Figure 7 below.

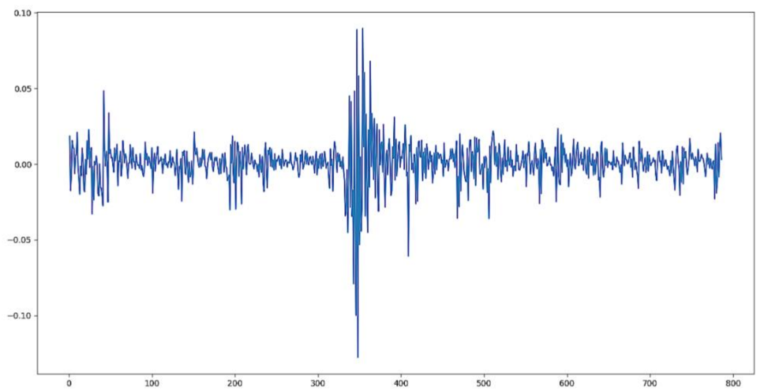

Figure 7. Prediction Daily Return from LSTM Model

Comparison with GARCH

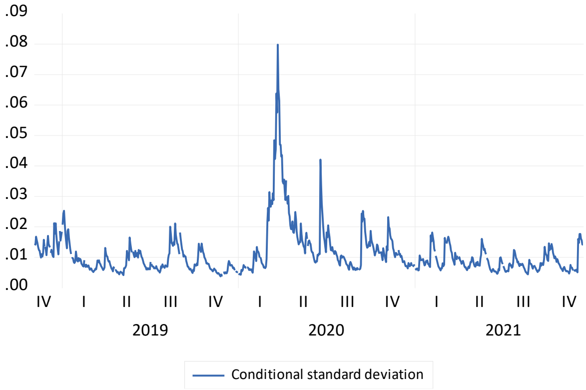

Figure 8. Forecast Daily volatility

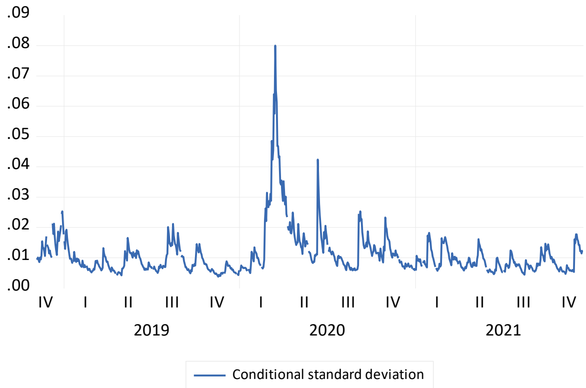

Figure 9. Actual Daily Volatility

After obtaining the forecast daily return, the daily volatility graph can be obtained as Figure 8. Compared with Figure 9, which is the daily volatility using actual daily return, the two figures overlap with each other a lot. Besides, although the Akaike Info Criterion of using the forecast data (-6.438) is a little bit higher (the lower the better), it is really closed to that of based on authentic data (-6.441).

5 CONCLUSION

In this paper, an improved LSTM deep learning network is used to forecast the stock volatility and make comparison between the predicted value with the actual value based on the traditional GARCH family model. As compared with the basic LSTM that is adopted by most of the previous researchers, this paper add on an validation set in order to better adjust the parameters. Meanwhile, the asymmetry of the stock market's performance on positive and negative news is considered as well (EGARCH) for better fitted.

The optimal parameter combination used in the model is obtained through dozens of repeated training. In the process, a lot of adjustments and comparison tests have been done as well.

After the empirical experiment and comparison, the author find that the result using data obtained from the LSTM is extremely close to the result using real data with other errors are so small as to be negligible except a little lagging issue. Therefore, the improved LSTM can be used to precisely forecast the future stock volatility.

However, there are still some minor problems. In order to resolve the problems mentioned above, adding more layers and technical approaches are plausible to be applied. Also, continuous variables are a bad design as the target of prediction, because it will make the prediction space too large, and the search space will be infinite. Therefore, techniques such as tile coding can be used to limit the prediction space. In addition, k-fold method could also be adopted to enlarge the training set. But one thing bear in mind that there are too many random emergencies in the stock market (regardless of the country and nature), and the impact of emergencies on the stock market is highly random. Therefore, there's still a long way to go for forecasting the stock market.

ACKNOWLEDGEMENT

I would like to thank Ms. Li first. She helped me out every time I faced difficulties with the technical issues. Ms. Wang also helped me decide whether the topic that I studied is appropriate and worth doing. She also corrected some of the structure and grammar mistakes to make this paper more professional. I would also like to thank Prof. Venkatesan Guruswami from Carnegie Mellon University, his courses not only taught me some computer knowledge, but also inspired me with such good writing ideas.

REFERENCES

[1] Linting Lin. Analysis of the Mode and Dilemma of Financial Technology Innovation Application [J] Modern Business, 2021(11): 97-99.

[2] H. Markowitz, "Portfolio selection," =e Journal of Finance, vol. 7, no. 1, 1952, pp. 77–91.

[3] F. Black and M. Scholes, "+e pricing of options and corporate liabilities," Journal of Political economy, vol. 81, no. 3, 1973, pp. 637–654.

[4] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," Journal of Econometrics, vol. 31, no. 3, 1986, pp. 307–327.

[5] D. B. Nelson, "Conditional heteroskedasticity in asset returns: a new approach," Econometrica, vol. 59, no. 2, 1991, pp. 347–370.

[6] Liu Yifei. Research on the Price Volatility of my country's Pulp Futures Market Based on GARCH Family Models, 2021. https://10.13691/j.cnki.cn23-1539/f.2021.05.031

[7] Zhou Aimin, Guan Rui. Research on volatility prediction based on deep learning algorithm—— as an example of 50 ETF options on the Shanghai Stock Exchange [J]. North China Finance, 2021, No.536(09): 15-21.

[8] Marcelo Sardelich & Suresh Manandhar. Multimodal deep learning for short-term stock volatility prediction, 2018. PACS: 05.10.-a, 05.40.-a 2010 MSC: 62-07, 62H99.

[9] Fang Jia and Boli Yang. Forecasting Volatility of Stock Index: Deep Learning Model with Likelihood-Based Loss Function, 2021. https://doi.org/10.1155/2021/5511802

[10] Eduardo Ramos-Pérez & Pablo J. Alonso-González. Multi-Transformer: A New Neural Network-Based Architecture for Forecasting S&P Volatility, 2021. https://doi.org/10.3390/math9151794.
[11] Zhang Lei, Sun Shanghong, Wang Yue. Exchange rate prediction based on deep learning LSTM model [J]. Statistics and Decision, 2021, v.37; No.577(13):158-162. DOI:10.13546/j.cnki .tjyjc.2021.13.037.