The Comparison of LSTM, LGBM, and CNN in Stock Volatility Prediction

Jiabao Li[1,*]

[1] *Department of Engineering and Physical Sciences, School of Computing, The University of Leeds Leeds, The United Kingdom*
[*] *Corresponding author. Email: ml2022jl@leeds.ac.uk*

ABSTRACT
In financial markets, volatility reflects the magnitude of price fluctuations. Forecasting volatility will be an important measure of the future direction of the market. Measuring and prdedicting stock market volatility has received increasing attention from academics and the industry over the past few years. This paper will focus on predicting the actual volatility of stocks using CNN, LightGBM, and LSTM models, using a data-set from Kaggle to make predictions. The paper gives a throughout analysis of the comparison for the performance of the three models. After testing with the chosen dataset, it was found that LGBM is more suitable for the task of predicting short-term stock volatility.

Keywords: *Stock price volatility, Volatility prediction, Machine learning.*

1. INTRODUCTION

Volatility is developing in financial theory as well as in practical application. With the availability of high-frequency data, new avenues have been taken to study the volatility of financial asset returns. In addition to directly modeling high-frequency returns, intra-day returns are also used to construct non-parametric, low-frequency (daily) volatility indicators, called realized volatility[1]. Volatility represents the magnitude of a stock's price movement. The more the stock price rises or falls, the more intense the price action is, the greater the volatility, and vice versa. In financial markets, volatility reflects the magnitude of price movements and high volatility is associated with periods of market turmoil and large price swings, while low volatility describes calmer and quieter markets [2]. Volatility can be used to measure the riskiness of stock as well. A stock with high volatility has a high degree of uncertainty about its price movement and when investors buy it, they may face a big raise or a big fall. For example, many small-cap and junk stocks tend to have high volatility. For example, many small-cap and junk stocks tend to have high volatility. Stocks with low volatility, on the other hand, have very stable price movements and face only small gains and losses after buying them, such as large-cap blue stocks. With the further development of artificial intelligence and deep learning, more and more data science-related techniques are spreading to other fields, and the development of neural networks and various related algorithms are making it possible to forecast stock volatility.

Measuring and predicting stock market volatility has received increasing attention from academics and the industry over the past few years. Miura et al. 's research aggregates realized volatility values using minute sampled bitcoin returns over a 3-hour interval. Next, using the RV time series, this work predicts future values based on past samples using several machine learning methods, ANN (MLP, GRU, LSTM), SVM, and ridge regression, which are implemented with heterogeneous regression with optimized lag parameters for volatility (HARRV) models. The results of this paper show that ridge regression performs best and supports the autoregressive dynamics assumed by the HARRV model. The neural network-based approach is followed by the mean squared error values, while SVM shows the worst performance [3]. Exactly forecasting multivariate volatility is essential for the financial industry. Bucci manifests the first attempt to model multivariate volatility through artificial neural networks, which is aiming at detecting nonlinear dynamics and long-term dependencies in the realized covariance series. The Cholesky-Artificial Neural Networks specification possesses two strengths here. On the one hand, the application of the Cholesky decomposition

affirms positive definite predictions. On the other hand, the utilization of artificial neural networks agrees to nominate nonlinear relations without any specific distributional assumption. Out-of-sample comparisons demonstrate that artificial neural networks cannot strongly excel over the competing models. However, long-memory detecting networks, like the Nonlinear Autoregressive model process with exogenous input and long short-term memory, have higher predictive accuracy than existing econometric models [4]. Bucci presents a large literature in the field of finance that has used artificial neural networks as a forecasting method in the last few decades as well. The main advantage of this approach is the possibility of approximating any linear and non-linear behavior without knowing the structure of the data generation process. This makes it suitable for forecasting time series that exhibit long memory and non-linear dependencies, such as conditional volatility. In this research, the comparison of the predictive performance of feedforward and recurrent neural networks (RNN), with a particular focus on the recently developed Long Short Term Memory (LSTM) networks and NARX networks, with traditional econometric methods. The results show that recurrent neural networks can outperform all traditional econometric methods. In addition, capturing long-term dependence through long-term short-term memory and NARX models appears to improve prediction accuracy also in a highly unstable framework [5].

The Light Gradient Boost Machine will be an important tool for forecasting achieved volatility. The Light Gradient Boost Machine algorithm is based on a gradient boost decision tree[6]. The gradient boost decision tree (GBDT) is a widely used machine learning algorithm due to its efficiency, accuracy, and interpretability [7]. GBDT has achieved state-of-the-art performance in many machine learning tasks, such as multi-class classification, click prediction, and learning ranking [8,9]. In recent years, with the advent of big data (both in terms of the number of features and number of instances), GBDT faces new challenges, especially in terms of the trade-off between accuracy and efficiency. Traditional implementations of GBDT require scanning all instances of data for each feature to estimate all possible segmentation. Data instances to estimate the information gain of all possible segmentation points. As a result, their computational complexity would be proportional to the number of features and the number of instances. This makes these implementations very time-consuming when dealing with large data. LightGBM is a new GBDT algorithm that incorporates two new techniques: gradient-based one-sided sampling and proprietary feature bundling to handle a large number of data instances and a large number of features, respectively [10]. GBDT is an integrated model of a decision tree, trained sequentially. In each iteration, the GBDT learns the decision tree by fitting negative gradients (also known as residuals).

This paper will first implement the code of LGBM, LSTM, and CNN for forecasting volatility, express the accuracy of forecasting by RMSPE and compare the overall forecasting RMSPE results with those of individual stocks to evaluate the performance of the three models under such tasks.

This paper will mainly present the comparison and evaluation of the performance of LSTM, LGBM, and CNN in stock volatility prediction.

2. METHOD

2.1. Data acquisition

In terms of data collection, it first browsed through the data provided by Kaggle (https://www.kaggle.com/c/optiver-realized-volatility-prediction/data) for various stock prices prediction projects, and finally found a highly granular order book of buy and sell orders for short time intervals and data on actual executed trades.

Each stock has a total of 7 elements, including stock_id, time_id, seconds_in_bucket, bid_price, ask_price,bid_size,ask_size, order_count, target.

In terms of data pre-processing, firstly some features have been added that were not available, such as log return, bid/ask spread, and weighted averaged price. A fair book value assessment must take into account two factors: the level and size of the order. It can use the weighted average price (WAP) to calculate the instantaneous valuation of a stock and target the actual volatility.

2.2. Data prediction

In this experiment, each of the three models is trained through data with existing features. Afterward, the data from the three models are tested with the yi'zhi'gu dataset that has been prepared to produce RMSPE values of the predicted values to compare the ability of the three models in terms of stock price volatility. The three models are decision tree, LSTM, and CNN.

The Light Gradient Boost Machine will be an important tool for forecasting achieved volatility. The Light Gradient Boost Machine algorithm is based on a gradient boost decision tree. The gradient boost decision tree (GBDT) is a widely used machine learning algorithm due to its efficiency, accuracy, and interpretability. GBDT has achieved state-of-the-art performance in many machine learning tasks, such as multi-class classification, click prediction, and learning to rank. In recent years, with the advent of big data

(both in terms of the number of features and number of instances), GBDT faces new challenges, especially in terms of the trade-off between accuracy and efficiency. Traditional implementations of GBDT require scanning all instances of data for each feature to estimate all possible segmentation. Data instances to estimate the information gain of all possible segmentation points. As a result, their computational complexity would be proportional to the number of features and the number of instances. This makes these implementations very time-consuming when dealing with large data.

Long Short Term Memory (LSTM) is a special type of RNN that is designed to solve the problem of gradient disappearance and gradient explosion during the training of long sequences. A recurrent Neural Network (RNN) is a type of neural network for processing continuous data. Compared with ordinary neural networks, it can handle sequentially changing data, but it still has the problem of long-term dependence, which arises when the nodes of a neural network have gone through many stages of computation and the features of the previous longer time slices have been covered. LSTM is made up of a series of LSTM Units, which are used to control the flow and loss of features. For example, the meaning of a word may change depending on what is mentioned above, and LSTM is a good solution to this type of problem. In simple terms, this means that an LSTM can perform better than a normal RNN on longer sequences.

CNN (Convolutional Neural Network) is a feed-forward neural network with artificial neurons that respond to a portion of the surrounding units in the coverage area and can be used in areas such as speech recognition, image processing, and image recognition. A CNN is essentially an input-to-output mapping that can learn a large number of mapping relationships between inputs and outputs without Without the need for any precise mathematical expressions between inputs and outputs, the network can map between input-output pairs as long as the convolutional network is trained with known patterns. Convolutional networks perform tutored training, so their sample set is made up of vector pairs shaped like: (input vector, ideal output vector). All of these vector pairs are supposed to be derived from the actual 'run' of the system that the network is about to simulate.

The experiments will train these models with data from the prepared training set and evaluate them with data from a dedicated test set. In addition, in the training process using market data from multiple stocks, it first stratifies the training data for each stock using K-Fold and performs cross-validation to adjust the parameters of the model. In the evaluation step, it will use another test set for prediction to ensure that no cheat occurs.

2.3. Exploit the relationship between realized volatility and market data.

In this experiment, root-mean-square percentage error is mainly used for analysis. RMSPE is based on RMSE(root-mean-square error). The root means the square error is the square root of the ratio of the square of the deviation of the predicted value from the true value to the number of observations n. In practical measurements, the number of observations n is always finite and the true value can only be replaced by the most trustworthy (best) value.

When the RMSPE converges to 1 or is equal to 1, it indicates that there is a higher preponderance of errors, i.e. the model may be incorrect or completely unsuitable for the task.

When the RMSPE converges to 0, which means that there is less error, i.e. the model and its parameters are appropriate and well suited to the task.

When RMSPE = 0, the training results are the same as the validation set, which means that there is an overfitting problem or there is a cheat in the training process, which means that the test data is present in the training set.

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}((y_i - \overline{y}_i)/y_i)^2} \quad (1)$$

2.4. Implementation details

The adopted method is implemented using Python, TensorFlow, Keras, and scikit-learn package. As for Light Gradient Boost Machine(LGBM), it sets the parameter ‘boosting_type’ to GBDT, 'objective’ to regression, ‘metric’ to None, and ‘n_jobs’ to None. In CNN, It applies the dense of the Keras layer, which is denoted as ‘normal’.

3. RESULTS AND DISCUSSION

3.1. Prediction performance using multiple stocks in different models

This section aims to evaluate those four models using RMSPE as the evaluation score. The value range of RMSPE is RMSPE$\in$[0,1], the less it is, the more accurate the prediction result is.

Figure 1, shows the distribution of the prediction results for a stock with id 0 in the stock market, where the blue is the actual price of the stock and the orange is the price obtained by the model.

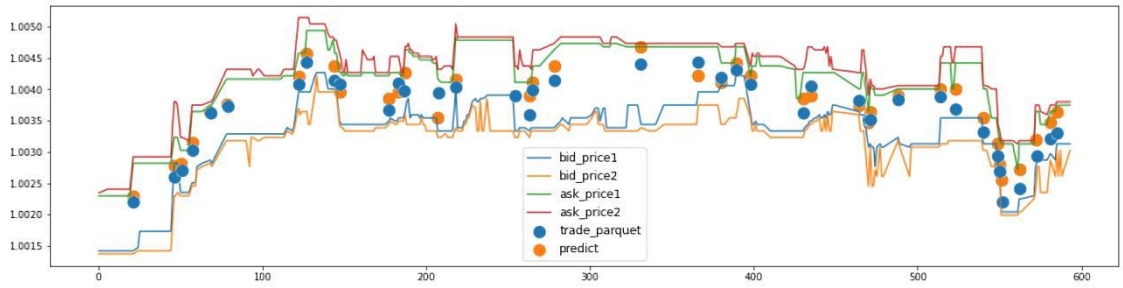

Figure 1 The prediction result of CNN.

Table 1 and Table 2 show the prediction results using the three models for individual stocks and the full test data-set respectively. The scores show the ability of the models to predict results.

Table 1. The RMSPE score of four stock volatility predictions by using Three models.

	CNN	LSTM	LGBM
Stock_id0	0.21762	0.3422	0.25082
Stock_id1	0.2230	0.3516	0.25133
Stock_id2	0.22573	0.3373	0.24897
Stock_id3	0.22164	0.3360	0.24967

Table 2. The RMSPE score of overall stock volatility prediction by using Three models.

	CNN	LSTM	LGBM
Stock_id0	0.22121	0.3347	0.2501

At the same time, Table 3 presents the time spent by the three models during training, and it can be seen that LGBM took the least amount of time during training, using only 157 seconds.

Table 3. The using time of training three models

	CNN	LSTM	LGBM
Use of Time	312(seconds)	874(seconds)	157(seconds)

Meanwhile, it found that many reasons affect the accuracy of the prediction results, which may be the setting of the model parameters or the amount of data.

In the process of implementing the specific code for the model, it can be seen that different features have different importance in the training process, which also affects the prediction results, as shown in Figure 2, where it can be seen that the feature: log_return1 has the greatest impact on the results of the model.

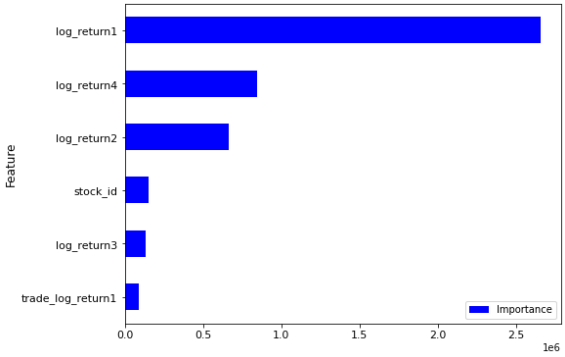

Figure 2 The importance of Features

4. CONCLUSION

The purpose of this paper is to analyze these two aspects using the stock data cited. First, it applies three different methods to predict the realized volatility of a stock. The CNN model achieves an RMSPE below 0.23 for all individual stocks, indicating accurate performance. In terms of training time, the LGBM model has a greater advantage. It then also investigated the stock volatility trends across the dataset for prediction. Finally, it obtained a performance comparison between the three models. From the experiments, it can be seen that the LGBM is more suitable for predicting stock volatility for short periods on the same operating environment and hardware platform, combining the training time and the accuracy of the model predictions.The conclusions presented in this paper are intended to provide some reference in the choice of models for stock forecasting, and the excellent performance of the LGBM model in this task broadens the choice and provides an example basis for the study of forecasting models with higher performance.

In the future, to make the prediction results more accurate, it can use more powerful neural networks, such as CNN with LGBM, to achieve accurate predictions, or integrate more dimensions of stock data and find more features that may influence the prediction results for training.

REFERENCES

[1] Corsi, F., Mittnik, S., Pigorsch, C., Pigorsch, U. (2008). The Volatility of Realized Volatility.

Econometric Reviews, 27(1-3), 46 – 78. https://doi.org/10.1080/07474930701853616

[2] Optiver Co. Ltd. Optiver Realized Volatility Prediction. [Online]. [Accessed 15 September, 2021]. https://www.kaggle.com/c/optiver-realized-volatility-prediction

[3] Miura, R., Pichl, L., & Kaizoji, T. (2019). Artificial Neural Networks for Realized Volatility Prediction in Cryptocurrency Time Series. Advances in Neural Networks – ISNN 2019, 11554, 165 – 172. https://doi.org/10.1007/978-3-030-22796-8_18

[4] Kambouroudis, D. S., McMillan, D. G., & Tsakou, K. (2021). Forecasting realized volatility: The role of implied volatility, leverage effect, overnight returns, and volatility of realized volatility. The Journal of Futures Markets, 41(10), 1618 – 1639. https://doi.org/10.1002/fut.22241

[5] Bucci, A. (2020). Cholesky – ANN models for predicting multivariate realized volatility. Journal of Forecasting, 39(6), 865 – 876. https://doi.org/10.1002/for.2664

[6] Guolin Ke, Qi Meng, Thomas Finely, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems 30 (NIP 2017).https://papers.nips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[7] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. Annals of statistics, pages 1189 – 1232, 2001.

[8] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In Proceedings of the 16th international conference on World Wide Web, pages 521 – 530. ACM, 2007.

[9] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. Learning, 11(23-581):81, 2010

[10] Bucci, A. (2020). Realized Volatility Forecasting with Neural Networks. Journal of Financial Econometrics, 18(3), 502 – 531. https://doi.org/10.1093/jjfinec/nbaa008