Stock Volatility Prediction Using LightGBM Based Algorithm

Zhangqi Liu*,
University of Wisconsin-Madison,
Wisconsin, United States,
janceyliu@163.com

Abstract— Nowadays, market volatility prediction is the most prominent terms you will hear in the trading market. Realized volatility is the representation of price movements, market's volatility and the trading risks. A little change happened in volatility will affected the expected return on all assets. In this article, we will use the dataset provided by Kaggle platform to predict the volatility. As a leading global electronic market maker, Optiver is dedicated to continuously improving financial markets, creating better access and prices for options, ETFs, cash equities, bonds and foreign currencies on numerous exchanges around the world. The prediction model we used in our paper is LightGBM, which is an improved version of XGBoost. We conclude some related work about the prediction of volatility. And we compute our model with others. Our model owns the best performance with the lowest RMSPE score 0.221. The RMSPE of Logistic Regression, SVM and Xgboost are respectively 0.069, 0.056 and 0.011 higher than that of LightGBM.

Index Terms— Optiver, LightGBM, realized volatility, Kaggle

I. INTRODUCTION

In financial market, volatility is one of the most important metrics that represents the degree of volatility of financial asset prices. It is a measure of the uncertainty of asset returns and is used to reflect the risk level of financial assets. The higher the volatility, the greater the volatility of financial asset prices, and the stronger the uncertainty of the return on assets; the lower the volatility, the smoother the volatility of financial asset prices, and the greater the certainty of the return on assets.

In addition, the volatility can be classified into four kinds: realized volatility, history volatility, implied volatility and expected volatility. Realized volatility, also called the future volatility, refers to the measurement of the degree of volatility of the return on investment during the validity period of the option. Since the return on investment is a random process, the actual volatility is always an unknown number. In other words, the actual volatility cannot be accurately calculated in advance, and people can only get its estimated value through various methods. In our paper, the dataset is provided by Optiver, a global electronic market marker for predicting the realized volatility.

In the following part, we firstly introduce related work on the volatility. In the section III, we focus on the LightGBM, which is one of industrial applications of GBDT. The experiment section describes the dataset and feature engineering, training parameters and comparative experiment. In the final part, we conclude our work and put forward our improvement expected.

II. RELATED WORK

In this section, we introduce the volatility on the prediction of realized volatility.

It is known to the market trader, besides the realized volatility, there are three other kinds of volatility.

1) History Volatility (HV)

History volatility [1][2] is based on past statistical analysis, assuming that the future is an extension of the past, using historical methods to estimate volatility is similar to estimating the standard deviation of the underlying asset return series. In the stock market, historical volatility reflects past fluctuations in the underlying stock price. However, because stock price fluctuations are difficult to predict, the use of historical volatility to predict warrant prices is generally not accurate.

2) Implied Volatility (IV)

Implied volatility [3][4] is the value of volatility derived from the theoretical price model of warrants by substituting the transaction price of warrants in the market. From a theoretical point of view: the option pricing model (such as the BS model) gives the quantitative relationship between the option price and five basic parameters (underlying stock price, execution price, interest rate, expiration time, volatility), as long as the first four The basic parameters and the actual market price of the option are substituted into the pricing formula as a known quantity, and the only unknown quantity can be solved from it, and its magnitude is the implied volatility.

3) Expected Volatility (EV)

It refers to the use of statistical inference methods to predict the actual volatility results, and use it in the option pricing model to determine the theoretical value of the option. [5][6] Therefore, the predicted volatility is the volatility that people actually use when pricing options in theory.

Typically, we use the realized volatility for prediction. The realized volatility is to assess changes in the returns of investment products by analyzing past returns over a particular period of

time. Uncertainty in investing in a company and / or valuation of potential economic losses / profits can be measured using the volatility / volatility of a company's stock price. In statistics, the most common way to determine volatility is to measure the standard deviation, or volatility of the average rate of return. Market realization volatility or actual volatility is caused by two factors that affect stock prices: the continuous volatility part and the jump part. The ongoing volatility of the stock market is affected by trading volumes during the day. For example, a single block transaction can cause significant fluctuations in the price of a product.

- *Our Contribution*

- We proposed a model based on LightGBM for forecasting the realized volatility.
- In the experiment, we do the feature engineering to get important financial features.
- We introduce our datasets and carry out the feature analysis.

III. METHODOLOGY

In our paper, we describe LightGBM in the detail. LightGBM is the industrial application of GBDT.[7] GBDT is an additive model composed of k trees:

$$\hat{y} = \sum_{k=1}^{K} f_k(x_i), \;\; f_k \in F$$

The training of the overall model becomes a very complex problem because all the base models are considered at the same time. The researchers came up with a greedy solution: train only one base model at a time.

The lift tree uses the addition model and the forward split algorithm to realize the optimization process of learning. When the loss function is a square loss and exponential loss function, each step optimization is simple. However, for the general loss function, it is often not easy to optimize every step. In response to this problem, Freidman proposed a gradient boost algorithm. This is an approximation method that utilizes the fastest descent method, the key of which is to utilize the negative gradient of the loss function at the value of the current model.

In GBDT, the residuals are actually the reverse gradient of the minimum mean-party loss function about the predicted values.

$$r = y - f_{m-1}(x)$$

$$\frac{\alpha(\frac{1}{2}(y - F_k(x)^2)}{\alpha F_k(x)} = \mathrm{r} = y - F_k(x)$$

Residual GBDT is prone to outliers; Subsequent models will be unusually concerned about misdivided values.

LightGBM improve GBDT from 7 aspects.

1. Unilateral gradient sampling algorithm;
2. Histogram algorithm;
3. Mutually exclusive feature bundling algorithm;
4. Vertical growth algorithm based on Leaf-wise with maximum depth;
5. Optimal segmentation of category characteristics;
6. Feature parallels and data parallels;
7. Cache optimization

IV. EXPERIMENTS

- **Experimental Data**

Our dataset is provided by Optiver, which is a famous electronic market marker. It is committed to continuously improving the financial market and creating better opportunities and prices for options, ETFs, stocks, bonds and foreign currencies on many exchanges around the world.

This dataset contains stock market data relevant to the practical execution of trades in the financial markets. In particular, it includes order book snapshots and executed trades. With one second resolution, it provides a uniquely fine-grained look at the micro-structure of modern financial markets. To do the experiments, we have 429k training dataset. And the hidden test set contains data that can be used to construct features to predict roughly 150,000 target values.

- Training dataset

Target: The realized volatility computed over the 10 minute's window following the feature data under the same stock/time_id. There is no overlap between feature and target data.

- Order Book

We have the book.parquet files, which is partitioned by stock_id. The parameters and their definition are shown in the following Table 1.

Table 1: book feature

stock_id	ID code for the stock. Not all stock IDs exist in every time bucket.
time_id	ID code for the time bucket.
seconds_in_bucket	Number of seconds from the start of the bucket, always starting from 0.
bid_price[1/2]	Normalized prices of the most/second most competitive buy level.
ask_price[1/2]	Normalized prices of the most/second most competitive sell level.
bid_size[1/2]	The number of shares on the most/second most competitive buy level.
ask_size[1/2]	The number of shares on the most/second most competitive sell level.

The term order book refers to an electronic list of buy and sell orders for a specific security or financial instrument organized by price level. An order book lists the number of shares being bid on or offered at each price point.

➢ Trade

Meanwhile, an order book is a representation of trading intention on the market, however the market needs a buyer and seller at the same price to make the trade happen. Therefore, sometimes when someone wants to do a trade in a stock, they check the order book and find someone with counter-interest to trade with.

The trade_[train/test].parquet partitioned by stock_id. Contains data on trades that actually executed. Usually, in the market, there are more passive buy/sell intention updates (book updates) than actual trades, therefore one may expect this file to be sparser than the order book. It has the other three fields.

Table 2: trade features

price	The average price of executed transactions happening in one second. Prices have been normalized and the average has been weighted by the number of shares traded in each transaction.
size	The sum number of shares traded.
order_count	The number of unique trade orders taking place.

➢ WAP[8]

A fair book valuation must consider two factors: the level and size of the order. In this paper, we use the weighted average price (WAP) to calculate the instantaneous stock valuation and calculate the realized volatility as our goal.

The WAP formula can be written as follows, which takes into account the top price and quantity information:

$$WAP = \frac{BidPrice_1 * AskSize_1 + AskPrice_1 * BidSize_1}{BidSize_1 + AskSize_1}$$

➢ Log returns [9]

The traders need to compare the stock price between yesterday and today. Returns are widely used in finance area, however log returns are preferred whenever some mathematical modelling is required.

Calling St the price of the stock S at time t, we can define the log return between $t1$ and $t2$ as:

$$r_{t_1,t_2} = \log\left(\frac{S_{t_2}}{S_{t_1}}\right)$$

we look at log returns over fixed time intervals, so with 10-minute log return we mean rt=rt−10min,t .

➢ Realized volatility

When we trade options, a valuable input to our models is the standard deviation of the stock log returns. The standard deviation will be different for log returns computed over longer or shorter intervals, for this reason it is usually normalized to a 1-year period and the annualized standard deviation is called volatility.

We will compute the log returns over all consecutive book updates and we define the realized volatility, σ as the squared root of the sum of squared log returns.

$$\sigma = \sqrt{\sum_t r_{t-1,t}^2}$$

Where we use WAP as price of the stock to compute log returns.

● **Experimental settings**

Our experimental settings are shown in the following table 3, we train our model using Pytorch.

Table 3: experimental settings

objective	rmse
boosting	gbdt
feature_fraction	0.7
Learning rate	0.2
bagging_fraction	0.8

● **Experimental Results and Analyses**

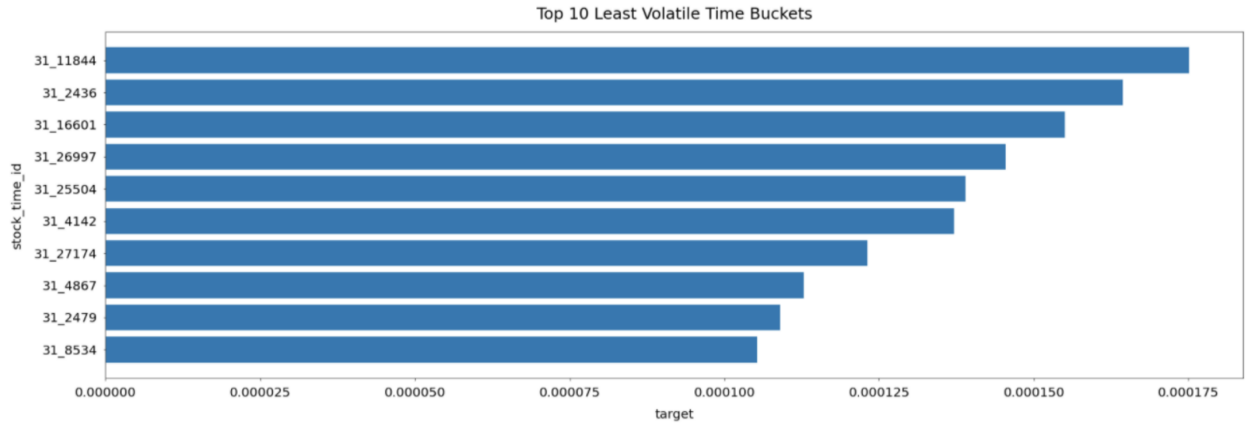

Figure 1: top 10 least volatile time buckets

The top 10 least volatile time buckets is shown in the figure 1. From the figure 1, we can see the stock_time_id 31_11944 owns the least target.

The wap value of one stock in 600s is shown in the following Figure 2.

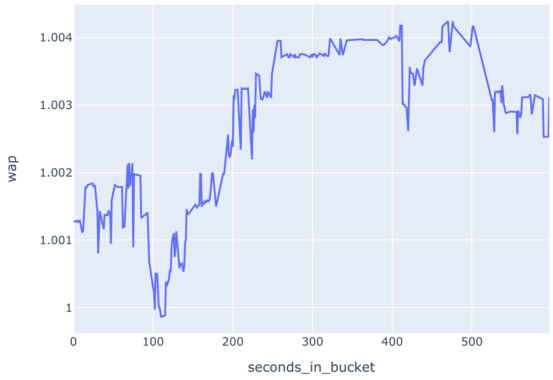

Figure 2. Wap value of stocks

To compare our model with other models, we evaluate our result using the metrics of RMSPE. The definition of RMSPE is shown as this:

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

The experimental results of competing models and our model are shown in table 4.

Table 4: performance of different models

Models	RMSPE
Logistic Regression	**0.290**
SVM	**0.267**
XGBoost	**0.232**
LightGBM	**0.221**

From Table 4, we can see that our model owns the best performance with the lowest RMSPE score 0.221. On the contrary, other classical models are hardly owns the performance like LightGBM. The RMSPE of Logistic Regression, SVM and Xgboost are respectively 0.069, 0.056 and 0.011 higher than that of LightGBM.

V. Conclusion

Realized volatility is the representation of price movements, market's volatility and the trading risks. A little change happened in volatility will affected the expected return on all assets. In this article, we will use the dataset provided by Kaggle platform to predict the volatility. In the section II, we conclude the related work, Section IV shows the experiments results and then we conclude the whole work of this article in the section V. The prediction model we used in our paper is LightGBM, which is an Improved version of XGBoost. We conclude some related work about the prediction of volatility. Our model owns the best performance with the lowest RMSPE score 0.221. On the contrary, other classical models are hardly owns the performance like LightGBM. The RMSPE of Logistic Regression, SVM and Xgboost are respectively 0.069, 0.056 and 0.011 higher than that of LightGBM.,

References

[1] Barndorffff-Nielsen, O. E. and N. Shephard (2004). Power and bipower variation with stochastic volatility and jumps. Journal of fifinancial econometrics 2 (1), 1–37.

[2] Bollerslev, T., A. J. Patton, and R. Quaedvlieg (2016). Exploiting the errors: A simple approach for improved volatility forecasting. Journal of Econometrics 192 (1), 1–18.

[3] Diebold, F. and R. Mariano (1995). Comparing predictive accuracy. Journal of Business Economic Statistics 13 (3), 253–63.

[4] Gu, S., B. Kelly, and D. Xiu (2020). Empirical asset pricing via machine learning. The Review of Financial Studies 33 (5), 2223–2273

[5] Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: a simple way to prevent neural networks from overfifitting. The journal of machine learning research 15 (1), 1929–1958.

[6] West, K. D. (1996). Asymptotic inference about predictive ability. Econometrica: Journal of the Econometric Society, 1067–1084.

[7] Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems 30 (2017): 3146-3154.

[8] Barndorffff-Nielsen, O. E., P. R. Hansen, A. Lunde, and N. Shephard (2009). Realized kernels in practice: Trades and quotes.

[9] A. Khan, K. Khan, B.B. Baharudin, Frequent patterns mining of stock data using hybrid clustering association algorithm, in: Proceedings of the International Conference on Information Management and Engineering, ICIME'09, 2009.