Proceedings of the 2nd International Conference on Management Research and Economic Development
DOI: 10.54254/2754-1169/85/20240845

Comparison of the Performance of Different Machine Learning Methods in Predicting VIX Volatility

Yinuo Zhai[1,a,*]

[1]*School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China*
a. Yinuo.Zhai23@student.xjtlu.edu.cn
**corresponding author*

Abstract: As a matter of fact, index volatility has always been one of the key indicators of the state of an index and a reflection of investor confidence and expectations in the market. Among various indicators, the VIX, which is also known as the "Panic Index", has always been viewed by the market as a barometer of the state of the economy. With this in mind, the purpose of this study is to investigate the process of Random Forest, Support Vector Regression as well as XGBoost in predicting VIX volatility and to evaluate their performance. Based on the evaluations, experiments in this study show that XGBoost performs optimally for smaller, low-dimensional time series data. According to the analysis, this study strongly demonstrates the excellent performance as well as great potential of machine learning algorithms in volatility prediction. Overall, these results will provide a feasible solution for volatility prediction with machine learning algorithms.

Keywords: Volatility prediction, random forest, SVR, XGBoost, STL

1. Introduction

From the 2008 subprime mortgage crisis to the 2020 U.S. stock meltdown, the stability and reliability of global capital markets are being challenged time and again, and investor confidence is constantly changing along with the volatility of a range of economic data such as indices, stock prices, and futures prices. In this environment, the study of asset volatility naturally appears to be particularly important [1]. Between the late 19th and mid-20th centuries, the concept of asset volatility was unclear, and investors and analysts focused more on market trends and fundamental analysis. In 1973, the Black-Scholes model marked the birth of modern option pricing theory and laid the foundation for the study of asset volatility [2]. The introduction of volatility derivatives such as the VIX at the beginning of the 21st century made asset volatility a separately traded asset class in the financial markets. Investors began to trade volatility directly, rather than just as a risk management tool [3].

VIX is a measure of expected market volatility calculated by CBOE based on the price of S&P 500 index options, also known as the Panic Index, which reflects the market's naive anticipation of the volatility of the S&P 500 index in the 30 days to come, and is often considered to be an important indicator of investor sentiment and market risk appetite [4]. The VIX rises when the market expects future volatility to increase and falls when the market expects volatility to decrease. During the 2019-2020 COVID-19 epidemic, the VIX rose by approximately 42%, which was directly and positively correlated to the number of deaths and the decline in stock prices [5].

DOI: 10.54254/2754-1169/85/20240845

Volatility forecasting has undoubtedly been the focus of econometricians' attention in the last five years, and the introduction of statistical-like learning methods has led to a rapid leap in research progress in this field. Traditional time series models such as GARCH have been applied to forecast volatility by Awartani and Corradi [6], while Sheikh et al. have applied ARIMA to this area with great success [7]. In addition, traditional machine learning has been viewed as a mainstay of volatility prediction, and its stable ability to fit with high accuracy cannot be ignored. Christensen, Siggaard, and Veliyev make a compelling case that multiple machine learning algorithms, including regularization and regression trees, have a more pronounced long-term memory advantage over heteroskedastic autoregressions (HARs) in volatility forecasting [8]. Wang and Guo went a step further by integrating multiple machine learning methods and proposed the DWT-ARIMA-GSXGB hybrid model [9].

This study focuses on the process of various chosen algorithms in predicting VIX volatility and comparing the performance of the above models. Next, this paper will explain in turn the data chosen, the model chosen and its parameters, the experimental procedure, the model performance comparison, the limitations of this study and the future outlook.

2. Data and Method

2.1. Data and Models

In this study, the historical data of VIX index from January 1, 2010 to December 31, 2023 are extracted from Yahoo Finance, and the date index and daily closing price "Close" are extracted as the basic features of the data. As the VIX data collected by Yahoo Finance is very complete and has been verified and cited by many parties, there are no missing items or unrecognizable data formats, which is indispensable for confirming the feasibility of various machine learning algorithms to be used in this study. This study describes in detail all the machine learning techniques used in this study, including algorithms, models and validation methods, and gives the hyperparameter configurations used in this study, as well as the specific computational formulas.

Random Forest improves prediction performance and problem solving with multiple correlations by constructing multiple decision trees [10]. The core idea of Random Forest is to construct multiple decision trees, each of which is a weak classifier, through the Bagging method of integrated learning. During training, Random Forest takes multiple subsets of samples from the original dataset with put-back and trains a decision tree for each subset. These decision trees also randomly select a portion of features from the feature set during node splitting during training, which increases the diversity and robustness. The final prediction is usually determined by voting (classification problem) or averaging (regression problem) the predictions from all the decision trees.

The generation of a decision tree is a recursive process that looks for the best division attributes at each intermediate node. The recursion stops when the current node contains samples belonging to the same class and no further division is required. The current attribute set is empty, or all samples take the same value on all attributes and cannot be divided. The current node contains an empty set of samples that cannot be divided.

The dataset chosen for this experiment is small and has a strong seasonal periodicity, and the data are relatively flat overall, so the risk of overfitting can be ignored in this experiment. It should be noted that as an integrated algorithm, Random Forest consumes more computational resources, so in the hyperparameter optimization stage, this experiment uses random search or k-fold cross-validation instead of grid search for this model.

Scholars are reasonably certain that support vector regression is one of the most mathematically beautiful machine learning algorithms available today. SVR improves generalization by minimizing the balance between prediction error and smoothness. The core idea of support vector regression is

DOI: 10.54254/2754-1169/85/20240845

to find an optimal function that minimizes the error between anticipated and true values. In most cases, this function is a nonlinear mapping that maps the data into a high-dimensional feature space such that there is a clearly identifiable linear relationship among the variables in this high-dimensional space. Specifically, the goal of SVR is to find an optimal hyperplane such that none of the sample data points are at a distance of no less than 1 to that hyperplane and there is no misclassification.

The basic mathematical formulation of SVR can be described by the following optimization problem. Given a training dataset$\{(x_1, y_1),(x_2, y_2), \ldots, (x_N, y_N)\}$,where$x_i$is the input vector and$y_i$is the corresponding output value. The goal of SVR is to minimize the following objective function:

$$\frac{1}{2}\|w\|^2 + \mathrm{C}\sum_{i=1}^{N}\left[\max\left(0,1 - \mathrm{y_i}(\mathrm{w^T x_i} + \mathrm{b})\right)\right] \tag{1}$$

Here, w is the normal vector, and b is the displacement term, C is the regularization parameter. The constraints are:

$$y_i(w^T x_i + b) \leq 1 \ \ for\ all\ i = 1, \ldots, N \tag{2}$$

XGBoost is an implementation of the Gradient Boosting Decision Tree (GBDT) based algorithm that builds a powerful predictive model by integrating multiple weak classifiers. Similarly, the mathematics of XGBoost is based on the gradient boosting framework, which is an iterative algorithm that optimizes a given loss function by constructing a series of decision trees. XGBoost uses a weighted squared loss function, which for the regression problem is defined as:

$$\mathrm{L}(\theta) = -\sum_{\mathrm{i}=1}^{\mathrm{n}}[\mathrm{y_i}\log(\theta^{\mathrm{T}}\mathrm{x_i}) + (1 - \mathrm{y_i})\log(1 - \theta^{\mathrm{T}}\mathrm{x_i})] \tag{3}$$

Among them, θ is the parameter of the model, $\mathrm{y_i}$ is the true label, and $\mathrm{x_i}$ is the feature vector. In each iteration, the algorithm selects a subset (usually based on the prediction error from the previous round) and grows a new tree on this subset. During the growth process, the algorithm performs pre-pruning (depth-wise) and post-pruning (alpha-wise) to avoid overfitting, and stops the growth by gain-pruning (gamma-wise) if the added tree brings less lift than a certain threshold.

The three hyperparametric optimization methods used in this experiment will be described below: k-fold cross-validation, Grid Search and Random Search. The core idea of K-fold CV is to reduce the risk of evaluation bias and overfitting by dividing the dataset into K equal-sized subsets. The following are the main steps of K-fold CV:

- Data division: the entire dataset is randomly divided into K equal-sized subsets.
- Training and validation: select one subset at a time as the validation set and merge the remaining K-1 subsets as the training set. Then train the model on the training set and test it on the validation set.
- Repeat process: this process can be seen as a K-times recursive procedure, each time selecting a different subset as the validation set to ensure that each subset is used as a validation set once.
- Results averaging: the results of the K training and validation sessions are averaged to obtain an accuracy that better represents the model's performance on unknown data.

In general, the commonly used Grid Search finds the optimal configuration by traversing all possible combinations of hyperparameters, which inevitably results in a significant use of computational resources. In contrast, Random Search randomly selects combinations for evaluation with relatively low computational cost, especially when the hyperparameter space is large. In a sense, Random Search is a "compromise" between computational efficiency and optimization, so to improve efficiency, it is possible to combine the strategies of Random Search and lattice search: first, Random

DOI: 10.54254/2754-1169/85/20240845

Search is performed to narrow down the search range, and then Grid Search is used to find the optimal parameters within this smaller range.

2.2. Evaluation metrics

In this study, MSE, MAE and R2 will be used as quantitative measures of model performance and visualized for comparison. The mean square error (MSE) is a measure of the average of the squares of the differences between the predicted values of a model and the actual observed values.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{1}\left(Y_i - \overline{Y_i}\right)^2 \tag{4}$$

The mean absolute error (MAE) is the average of the absolute values of the differences between the predicted and actual values.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \overline{Y_i}\right| \tag{5}$$

R-squared is the percentage of variability in the dependent variable that can be explained by the regression model. It has a value between 0 and 1 and is calculated by the formula:

$$R^2 = 1 - \frac{sum\ of\ squares\ of\ the\ residuals}{total\ sum\ of\ squares} \tag{6}$$

Hyperparameters of Random Forest are n_estimators: 200, min_samples_split: 2, min_samples_leaf: 2, max_depth: None. Hyperparameters of SVR: kernel are 'linear', gamma: 'scale', C: 0.1. Hyperparameters of XGBoost are colsample_bytree: 1, learning_rate: 0.1, max_depth: 3, min_child_weight: 1, n_estimators: 150, subsample: 1.

3. Results and Discussion

3.1. Feature engineering

To visualize the general trend of the VIX data, this study first visualizes the data. The data curve is rather irregular and rough, so a 20-day moving average is calculated instead. By averaging data points over a certain time window, a moving average reduces the impact of short-term fluctuations or "noise" and reveals the trend behind the data (seen from Fig. 1). 20-day moving averages are a much quicker way of reflecting recent price changes than longer-term moving averages, such as 50-day or 200-day moving averages, and are suitable for capturing medium-term trends. It is also suitable for the size of the dataset chosen for this experiment.

DOI: 10.54254/2754-1169/85/20240845

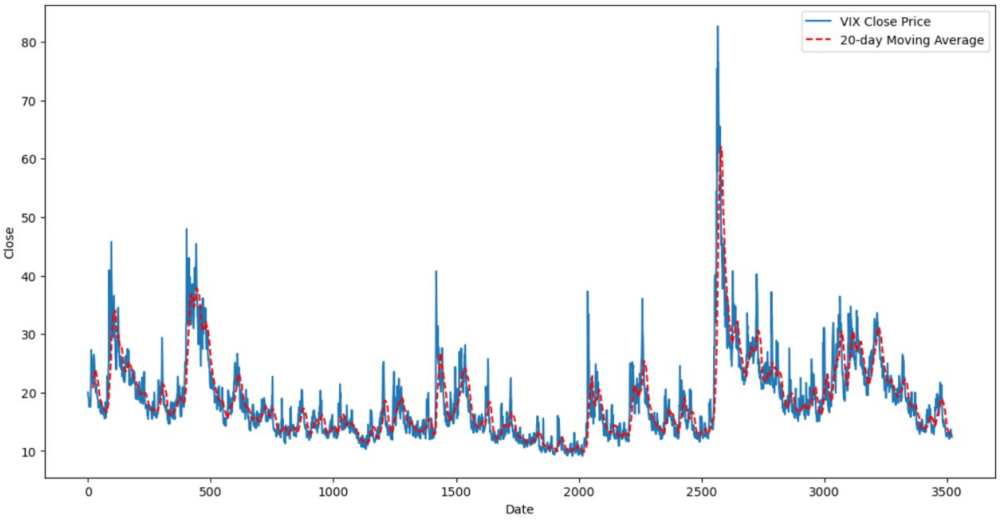

Figure 1: Price for VIX (Photo/Picture credit: Original).

In general, long-term time series data such as indices, stocks, etc. usually have obvious long-term trends and seasonal cycles, and are also bound to be disturbed by noise (random components), so it is indispensable to perform time series decomposition. In this experiment, STL is utilized as a method for time series decomposition. STL splits the time series data into multiple seasonal cycles, and for each seasonal cycle, Loess regression is used to estimate the trend and seasonal components.Loess is a nonparametric locally weighted regression method that estimates the trend by sliding a window through the data and using a locally weighted regression within each window. This captures the local trend in the data well while being insensitive to noise. Finally, the Loess regression results within each period are combined to obtain the trend and seasonal components of the entire time series. The trend and seasonal components are then subtracted from the original time series to obtain the remaining stochastic component, also known as the residual (seen from Fig. 2).

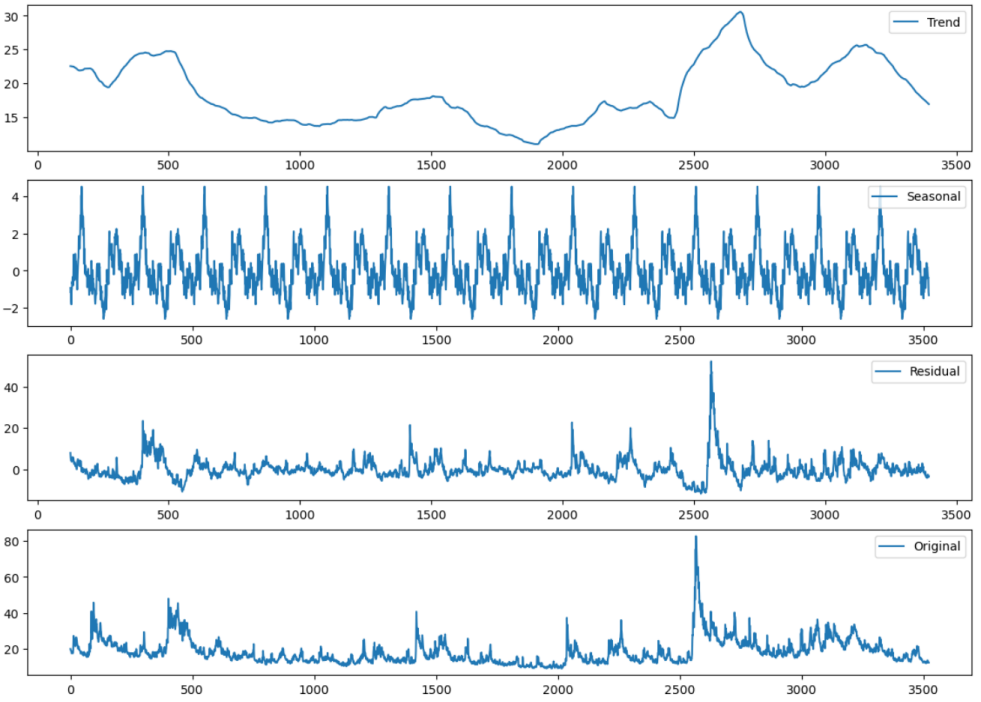

Figure 2: Seasonal decomposition (Photo/Picture credit: Original).

DOI: 10.54254/2754-1169/85/20240845

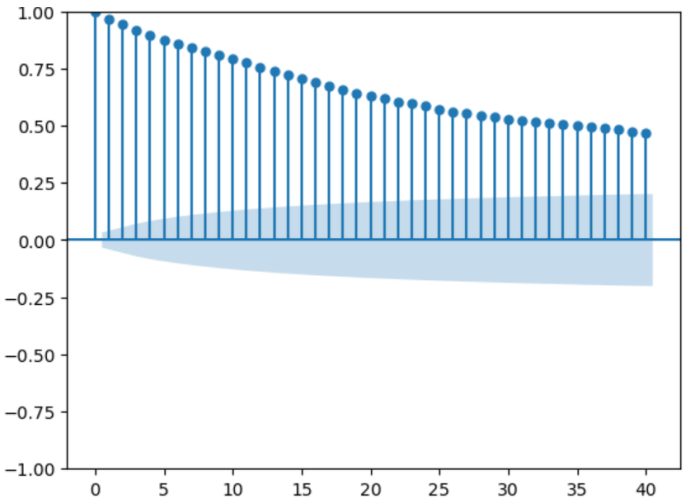

Figure 3: ACF plot (Photo/Picture credit: Original).

Empirically, as a time series, an index usually has significant autocorrelation, so this experiment next applies the autocorrelation function (ACF) to test the autocorrelation of the data (as shown in Fig. 3). Plotting the ACF requires the calculation of the autocorrelation coefficient, which is the degree of correlation between two series at a given delay, usually measured by the Pearson correlation coefficient. For the time series $x_1, x_2, \ldots, x_n$, The autocorrelation coefficient ρ_l at lag l can be calculated by the following equation:

$$\rho_l = \frac{\sum_{t=1}^{n-l}(x_t-\overline{x})(x_{t+1}-\overline{x})}{\sqrt{\sum_{t=1}^{n-l}(x_t-\overline{x})^2}\sqrt{\sum_{t=1}^{n-l}(x_{t+1}-\overline{x})^2}} \tag{7}$$

where $\overline{x}$ is the mean of the time series and n is the length of the series. This experiment observed that the autocorrelation coefficient is positive and decreases slowly with increasing lag, which suggests that the series has long-term correlation and may be a trend or seasonal component. From the results of the STL time series decomposition and autocorrelation test, it is easy to see that the data chosen for this experiment have strong autocorrelation, seasonal periodicity, and long-term smoothness, which suggests that one does not need to introduce additional factors (e.g., new stocks or indices) as features. Therefore, this experiment uses the 20-day moving averages of the selected VIX historical opening, closing, high, low, and trading volume of all the above indicators as the features, and divides the training set and test set according to 8:2.

3.2. Model Performance

Table 1 shows the performance data and learning curves for each model in this experiment (using linear regression as the baseline). The learning curve measures the model's ability to learn, prediction accuracy and fit. The learning curves of the three models studied in this experiment reflect a high degree of consistency: the training score stabilizes at 1 for a long period of time, while the validation score grows linearly at a high rate until the training sample is 1000, and then grows steadily and nearly flat after 1000. This implies that there is no overfitting of the models and the fitting accuracy is relatively high. It can be seen that Random Forest, Support Vector Regression, and XGBoost all have very excellent performance on the dataset of this experiment, with XGBoost's performance being slightly stronger than the other models. The learning curve of SVR fluctuates a lot and there is a big difference between the MSE and MAE, which is most likely since MSE amplifies the prediction error when the prediction error of the data is not uniformly distributed. It is also possible that the

DOI: 10.54254/2754-1169/85/20240845

kernel function of SVR is set as "linear" in this experiment, but the complexity of the data is suitable for other kernels, such as polynomial kernel, radial basis function (RBF) kernel.

Table 1: Model performances of Random Forest, SVR and XGBoost.

	MSE	MAE	R-squared
Random Forest	0.038025	0.024807	0.999211
SVR	0.006594	0.080464	0.999863
XGBoost	0.0022	0.0354	1.0000

4. Conclusion

To sum up, this study compares the performances of three models for VIX prediction. This experiment compares various models under the same dataset, but they require different preprocessing of the data, which is exactly what needs to be improved in this experiment, and future research should be carried out on data preprocessing for deep adaptation to specific models. In addition, in the feature engineering step, this experiment lacks smoothness analysis, which is important for autocorrelation analysis. Finally, the dataset selected for this experiment is of low dimensionality and complexity, which leads to the quantitative measures of performance of different models being very close to each other, making it difficult to compare their differences in depth. Future research could introduce more sophisticated machine learning and deep learning models, as well as higher dimensional datasets, to expand the generalizability and reliability of the conclusions of this experiment. Meanwhile, future research can be combined with more cutting-edge work, such as in the feature engineering stage where multimodal data can be introduced to construct new features, such as analyzing users' panic about the economy from news and social media based on machine learning sentiment analysis and causal inference.

References

[1] *Debesh, B. (2013) International Journal of Scientific and Research Publications, 3, 10.*

[2] *Black, F., Scholes, M. (1973) The pricing of options and corporate liabilities. Journal of political economy, 81(3), 637-654.*

[3] *Fernandes, M., Medeiros, M.C., Scharth, M. (2014). Modeling and predicting the CBOE market volatility index. Journal of Banking & Finance, 40, 1-10.*

[4] *Zhang, J.E., Zhu, Y. (2006) VIX futures. Journal of Future Market, 26, 521-531.*

[5] *Grima, S., Özdemir, L., Özen, E., Romānova, I. (2021) The Interactions between COVID-19 Cases in the USA, the VIX Index and Major Stock Markets. International Journal of Financial Studies, 9(2), 26.*

[6] *Awartani, B.M., Corradi, V. (2005) Predicting the volatility of the S&P-500 stock index via GARCH models: the role of asymmetries. International Journal of forecasting, 21(1), 167-183.*

[7] *Idrees, S.M., Alam, M.A., Agarwal, P. (2019) A prediction approach for stock market volatility based on time series data. IEEE Access, 7, 17287-17298.*

[8] *Christensen, K., Siggaard, M., Veliyev, B. (2023). A machine learning approach to volatility forecasting. Journal of Financial Econometrics, 21(5), 1680-1727.*

[9] *Wang, Y., Guo, Y. (2020). Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost. China Communications, 17(3), 205-221.*

[10] *Biau, G., Scornet, E. (2016). A random forest guided tour. Test, 25, 197-227.*