The Korean Economic Review
Volume 38, Number 3, Summer 2022, 541-569.
DOI : 10.22841/kerdoi.2022.38.3.007

Multi-Step-Ahead Forecasting of the CBOE Volatility Index in a Data-Rich Environment: Application of Random Forest with Boruta Algorithm*

Byung Yeon Kim** · Heejoon Han***

The CBOE volatility index (VIX) is a representative barometer of the overall sentiment and volatility of the financial market. This paper seeks to apply random forest and its variable importance measure to forecasting the VIX index. Compared to the previous literature which has found it difficult to outperform the pure HAR process in terms of forecasting the VIX index due to its persistent nature, random forest can produce forecasts that are significantly more accurate than the HAR and augmented HAR models for multi-days forecasting horizons. This paper shows that the forecasting accuracy of random forest could be further improved by systematically selecting the optimal number of the most important covariates from a dataset of 298 macro-finance variables, while using the Boruta algorithm which ranks the variables based on random forest's variable importance measure. The superior predictability of this method is more evident with longer forecasting horizons.

JEL Classification: C01, G17
Keywords: Random Forest, Boruta Algorithm, Machine Learning, VIX Index, Volatility Forecasting

I. Introduction

The implied volatility index of the Chicago Board Options Exchange (CBOE), commonly known as the VIX index, represents the market's estimate of the future

Received: Aug. 18, 2021. Revised: March 7, 2022. Accepted: April 25, 2022.

* We would like to thank Chang Sik Kim and two anonymous referees for their helpful comments and suggestions. This paper was supported by Sungkyun Research Fund, Sungkyunkwan University, 2019.

** First Author, Graduate Student, Department of Economics, Sungkyunkwan University, Republic of Korea, E-mail: bryank2242@gmail.com

*** Corresponding Author, Professor, Department of Economics and Department of Quantitative Applied Economics, Sungkyunkwan University, Republic of Korea, E-mail: heejoonhan@skku.edu

volatility of the S&P 500 over the next 30 calendar days. It is derived from the bid/ask quotes of options on the S&P 500 index, and it is disseminated on a real-time basis. As it is calculated directly based on option prices rather than being solved out of an option pricing formula like the Black-Scholes, the VIX index is free from the measurement errors that were present in the previous implied volatility measures.

The VIX index attracts substantial attention in the financial market. Not only is it widely traded in the form of VIX futures for hedging or speculative purposes, but it is also acknowledged as the world's leading barometer of investor sentiment and market volatility. Thus, accurate forecasts of the VIX index in the short and long term can provide crucial information to participants in the financial market.

There are numerous research topics related to implied volatility and the VIX index and, unsurprisingly, there has been research focusing directly on forecasting the VIX index. Degiannakis (2008) considers realized and conditional volatility of the S&P 500 as exogenous covariates in modelling VIX in an ARFIMA model. However, he concludes that the VIX index is hard to forecast, and that it does not seem to be closely connected to the volatility of the underlying index. Konstantinidi et al. (2008) model seven different implied volatility indices including VIX in a multivariate VAR framework, and they confirmed the presence of implied volatility spillover between various markets. However, their method did not succeed in deriving significantly improved forecasts.

Fernandes et al. (2014) apply a heterogeneous autoregressive (HAR) model coupled with neural network approximation to capture non-linearity for forecasting VIX; they conclude that it is very hard to exceed the performance of the pure HAR process due to the highly persistent nature of the VIX index. Conversely, Psaradellis et al. (2016) find significant evidence of strong non-linearity in VIX by employing a HAR process combined with support vector regression model, thereby improving upon the results of the one-day-ahead forecasts of the pure HAR model.[1]

However, most of the literature focuses solely on one-day-ahead forecasts of VIX, while suggesting the multi-day-ahead forecast problem as a topic of future research. Forecasting VIX on a longer horizon can be a significant matter in several aspects. For an investor who adjusts his/her portfolio including VIX futures on a multi-day basis while considering trading costs, multi-day-ahead forecasts may be more useful than one-day-ahead forecasts. For a market participant searching for clues about the future volatility and direction of the overall market, an accurate multi-day-ahead forecast of VIX can provide significant information. Moreover, most of the studies on VIX forecasting consider only a handful of exogenous covariates, if any, and they

[1] Ballestra et al. (2019) consider the directional forecast of VIX Futures instead of the VIX index, and they use a feed-forward neural network model with non-lagged explanatory variables that are available only a few hours before the opening of the CBOE. They find that the neural network model with only one most recent exogenous variable is the superior model.

do not make use of the "big" datasets that are easily available nowadays.

The main motivation of this paper is to fill these research gaps. We focus on multi-day-ahead forecasting of VIX of up to 22 trading days, and we also utilize a high-dimensional dataset including 298 macro-finance variables. We investigate whether the findings of Fernandes et al. (2014) are still valid when utilizing a high-dimensional dataset. Another feature of this paper is to investigate a random forest procedure that systematically selects the most important variables. Specifically, we adopt the Boruta algorithm to select macro-finance variables and choose, via cross-validation, the optimal number of the most important variables that are used for random forest. To the best of our knowledge, this is the first study to apply a systemic variable selection mechanism based on the Boruta algorithm in time series forecasting.

The main findings of this paper are as follows: First, the random forest method provides superior multi-day-ahead out-of-sample forecasts; while it does not produce better one-day-ahead forecasts, it outperforms other benchmark models in 5/10/22-days-ahead forecasts. Moreover, the relative accuracy of the random forest method compared to benchmark models becomes more evident as the forecasting horizon increases.

Second, the random forest method using only the optimal number of the most important covariates from the Boruta algorithm can produce significantly superior out-of-sample forecasts over that using all available covariates. This is consistent with the existing view noted in Kohavi and John (1997) suggesting that it is important to select the most important variables when given a high-dimensional dataset. This finding is still valid when we consider more recent data and the various machine learning methods described in Appendix A.

The rest of the paper is organized as follows: The next section describes our methodology and briefly explains the random forest method and Boruta algorithm. The third section describes the data, forecasting procedure, and benchmark models. The fourth section reports the main results, including the variable selection, choice of optimal number of variables, forecast results, and robustness check. Lastly, the fifth section concludes the paper.

II. Methodology

The recent advances in machine learning (ML) methods and the increased accessibility to "big" datasets have led to opportunities to approach the problem of forecasting economic time series in a novel way. While traditional econometric applications are centered around parameter estimation, ML methods revolve around the problem of prediction—specifically, of producing predictions of $\hat{y}$

from x. While traditional econometric models rely on careful assumptions about the underlying data-generating-process, ML methods seek to discover complex structures that are not specified in advance. They manage to fit complex and very flexible functional forms to the data without simply overfitting, and they produce relatively accurate out-of-sample predictions (Mullainathan and Spiess, 2017).

Medeiros et al. (2019) is one of the recent studies that has highlighted the benefits of applying ML methods to economic time series forecasting. It applies a wide range of ML models to forecasting US inflation, and it finds that ML methods combined with high-dimensional datasets can produce more accurate forecasts than traditional benchmark models. Specifically, it reports that a particular model, random forest (RF) of Breiman (2001), consistently outperforms all other models due to its ability to catch nonlinearities and its variable selection mechanism. Moreover, it reports that the superiority of random forest becomes more evident in settings where the forecasting horizon becomes longer as well as during the periods when the time series is more volatile.

This research seeks to discover how ML methods can provide benefits to forecasting the VIX index, especially with a focus on the random forest method and its variable selection mechanism through variable importance measures. The distinct features of our methodology are that we adopt the *Boruta algorithm* by Kursa et al. (2010) to select the most important covariates and that we choose the optimal number of selected covariates in the random forest method.

One of the strengths of machine learning methods such as RF is their ability to handle datasets with high-dimensional covariates. However, many machine learning algorithms exhibit decreased accuracy when the number of variables is significantly higher than optimal (Kohavi et al. 1997). Thus, when given a high-dimensional dataset, it is often an important matter to distinguish and select out the most important variables, not only for technical efficiency, but also to enhance accuracy in solving the relevant problem. Our results in Section 4 confirm that RF using the optimal number of selected covariates, as opposed to all available high-dimensional data, provides better forecasts.

Our methodology consists of the following steps:

Step 1: Using the Boruta algorithm, obtain the rankings of the covariates in high-dimensional data.

Step 2: Choose the optimal number of the most important covariates via cross-validation.

Step 3: Using only those selected covariates from the previous step, implement the random forecast method and produce a forecast.

We briefly explain the random forest method and Boruta algorithm in the following subsections.

2.1. Random Forests and Permutation Importance

Random forest has its roots in classification and regression trees (CART). Introduced by Breiman et al. (1984), it is a simple model which partitions the predictor space into rectangles using binary splits, then uses those splits to determine the outcome prediction. That is, it divides the set of possible values of the predictors $X_1, X_2, \ldots, X_P$ into J distinct and non-overlapping regions, $R_1, R_2, \ldots, R_J$. For every observation that falls into region R_J, the same prediction is made, which is simply the mean of the response values for the training observations in R_J. For a regression tree, the objective is to identify the partition $R_1, R_2, \ldots, R_J$ such that the residual sum of squares (RSS, henceforth) given by

$$\sum_{j=1}^{J} \sum_{i \in Rj} (y_i - \hat{y}_{R_j})^2 \tag{1}$$

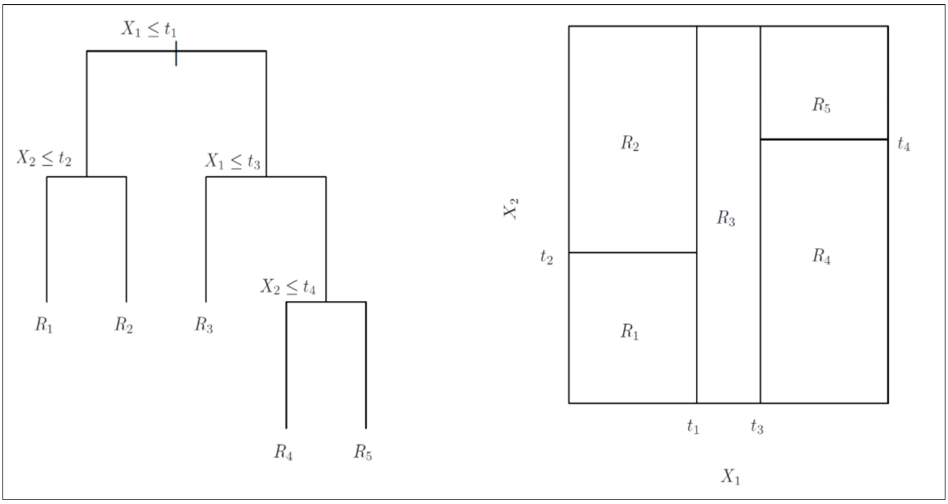
is minimized.

It is apparent that it becomes computationally infeasible to consider every possible partition of the predictor space. As a result, a top-down approach known as recursive binary splitting is utilized. The tree diagram in the left of Figure 1 illustrates a widely used example from Hastie et al. (2001) in which a tree model is grown in a regression setting with two predictors – X_1 and X_2. On the top node (or split) of the tree, the predictor space is partitioned into two regions at $X_1 = t_1$. Then, the region to the left of $X_1 = t_1$ is partitioned at $X_2 = t_2$ and the region to the right is partitioned at $X_1 = t_3$. Finally, the region to the right of $X_1 = t_3$ is partitioned at $X_2 = t_4$. At each node of the tree, the best split is determined such that the decrease in RSS due to the particular split is maximized. It is a greedy approach in that each split only considers the best one at that particular step, rather than looking ahead to also consider the future steps. The resulting partition of the predictor space is illustrated in the right diagram of Figure 1, where the five regions (or rectangles) $R_1, \ldots, R_5$ correspond to the five terminal nodes in the tree diagram.

An obvious question one faces when growing a tree model is how large the tree should be grown. A very large tree could easily overfit the data, whereas a very small tree could miss out on important structures underlying in the data. Cost-complexity pruning is a strategy that is widely used to determine the optimal tree size. The idea is to grow a sufficiently large tree, then prune the tree back to obtain a subtree that minimizes the cost-complexity criterion that penalizes the size of the tree model.

While having low model bias, a single tree model is typically known to be less competitive with the best ML methods in terms of prediction accuracy due to its high variance. Random forest, introduced by Breiman (2001), seeks to reduce the variance of trees through a bootstrap aggregation (or bagging) approach. Thus, the idea is to average many noisy but approximately unbiased trees to achieve stability while taking advantage of the advantages of tree models.

[Figure 1] Example of a Regression Tree



Random forest is an ensemble of a few hundred to thousands of unpruned trees, each trained on a bootstrap sample of the original data. When building a tree from a bootstrapped sample, RF uses m randomly selected input variables at each split.[2] This random selection of potential predictors to be selected ensures that the trees in the forest are decorrelated to each other. For a regression problem, the prediction of RF for a new test point x is defined as

$$\hat{f}_{rf}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x), \tag{2}$$

where B is the number of trees in the whole forest and $T_b(x)$ corresponds to the prediction from the b^{th} tree.

A desirable by-product of the bootstrap sampling process of RF is the presence of out-of-bag (OOB) samples, or the observations that are left out from each bootstrap sampling. These OOB samples can be utilized to measure the importance of each input variable.

When the b^{th} tree is grown, the OOB samples are run down the tree to calculate the OOB mean squared error (MSE):

[2] A typical choice of m with $m = \sqrt{p}$ for classification and $m = p/3$ for regression is known to perform well in most cases.

$$OOBMSE_b = \frac{1}{n_{OOB,b}} \sum_{i=1:i\in OOB_b}^{n} (y_i - \hat{y}_i)^2, \tag{3}$$

where $n_{OOB,b}$ denotes the number of observations in the b^{th} OOB sample. Then, the values of the j^{th} variable X_j are randomly permuted in the OOB data, and the permuted OOB MSE is calculated for the j^{th} variable:

$$OOBMSE_b(X_j\ permuted) = \frac{1}{n_{OOB,b}} \sum_{i=1:i\in OOB_b}^{n} (y_i - \hat{y}_i(X_j permuted))^2 . \tag{4}$$

If X_j does not have a predictive value for the given tree, random permutation of X_j should make a small difference to the OOB MSE. On the other hand, if X_j is used as an important variable within the tree, then random permutation of X_j should lead to a significant increase in OOB MSE. Thus, the increase in MSE due to this permutation averaged over all trees is used as a measure of the importance of X_j:

$$\frac{1}{B}\sum_{b=1}^{B}(OOBMSE_b(X_j permuted) - OOBMSE_b) . \tag{5}$$

This measure of variable importance in RF is known as the *permutation importance*.

2.2. The Boruta Algorithm

Until recently, there have been various methodologies developed for variable selection using RF variable importance measures. These developments have been particularly extensive in the bioinformatics and related fields, i.e., for identifying the important genetic variables for predicting certain disease status such as cancer. However, there does not yet appear to be consensus on a single outperforming variable selection methodology in a RF setting.

Speiser et al. (2019) compare the performance of 13 different RF variable selection procedures that have been developed. It reports the OOB errors as well as the computation time of the different methodologies when they are applied to 311 different datasets. That study reports that the *Boruta algorithm* by Kursa and Rudnicki (2010) is one of the better performing procedures overall in terms of OOB errors, and it is especially preferrable in high-dimensional settings with over 50 predictors.

The Boruta is a wrapper algorithm built around RF that provides a stable selection of the important variables from the dataset. The Z-score, which is derived for each variable by dividing the permutation importance measure by its standard deviation, is used as the measure of selection. Moreover, it extends the dataset by adding variables that are random by design. For each variable in the dataset, it creates a 'shadow attribute' which is obtained by shuffling the values of the original variable.

In detail, the Boruta algorithm consists of the following steps:

1. Extend the dataset by adding copies of all variables.

2. Shuffle the added variables to remove their correlations with the response. (Shadow attributes)

3. Run a random forest on the extended dataset and collect the computed Z scores.

4. Find the maximum Z score among shadow attributes (MZSA), then assign a hit to every variable that scored better than MZSA.

5. For each variable with undetermined importance, perform a two-sided test of equality with the MZSA.

6. Deem the variables that have significantly lower importance than MZSA as 'unimportant' and permanently remove them from the dataset.

7. Deem the variables that have significantly higher importance than MZSA as 'important'.

8. Remove all shadow attributes.

9. Repeat the procedure until the importance is assigned for all the variables or the algorithm has reached the previously set limit of the random forest runs.

Through an iterative process of eliminating variables deemed unimportant, the Boruta algorithm can deal with both the fluctuating nature of the RF variable importance measure and the interactions between the variables. Figure 2 in Kursa and Rudnicki (2010) shows an example of a Boruta result plot that displays the distribution of Z-scores from the iterations, and from which the ranking of the relative importance between variables can be derived.

The Boruta package available for usage in R is utilized for the results in Section 4. We let the maximum number of iterations of the Boruta algorithm be 100. The mean of the Z-scores from the 100 iterations are extracted from the results of the Boruta algorithm, and the variables are ranked based on this measure. In Section 4, the list of the top-30 ranked variables is reported and utilized for the purpose of selecting an optimal set of variables to forecast VIX in a random forest method.

III. Data and Forecasting Procedure

3.1. Data

We consider the sample period starting from April 5, 1990 and extending to January 15, 2013, matching that of the data used in Fernandes et al. (2014). The sample period includes a total of 5,740 daily observations of the VIX index and all exogenous variables. In Section 4.4, as a robustness check, we also consider recent data ranging from January 27, 2009 to December 31, 2020 with a total of 3,005 daily observations of all variables.

[Figure 2] Logarithm of VIX (April 5, 1990 – January 15, 2013)

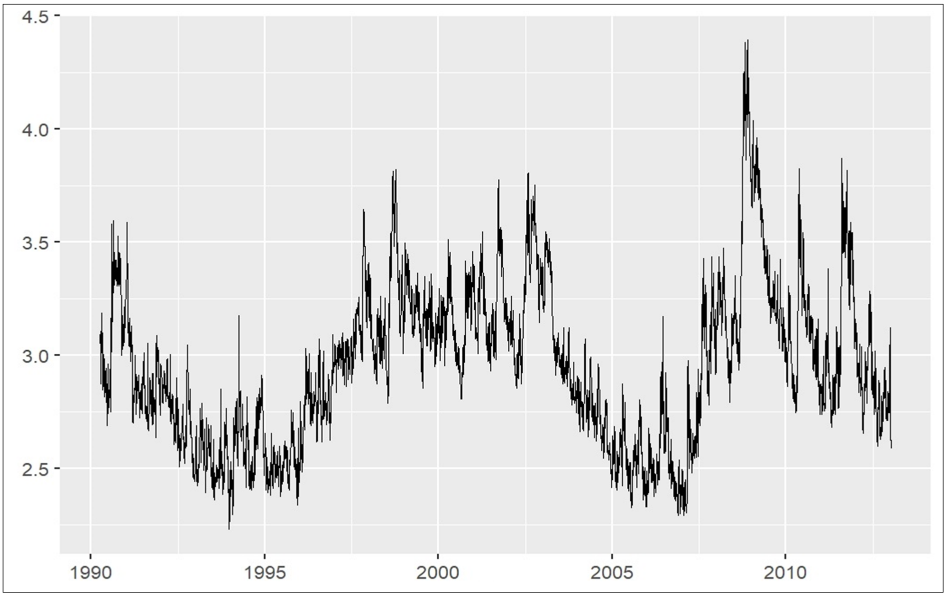


Table 1 lists the descriptive statistics for the logarithm of the VIX index for the period from April 5, 1990 to January 15, 2013. It reports the mean, median, minimum, maximum, standard deviation, skewness, kurtosis, p-values of Jarque-Bera, Augmented Dickey-Fuller (ADF), Phillips-Perron (PP) test results, and the KPSS test statistic. The unit root test results show that the logarithm of the VIX index is stationary, as the null hypothesis of unit root is rejected in the ADF and PP tests and the KPSS test cannot reject the null hypothesis of stationarity.

This research considers 298 other macro-finance variables as exogenous covariates for forecasting the VIX index, and they consist of the following: the k-day continuously compounded returns on the S&P 500 index (k=1, 5, 10, 22, 66) and the first difference of the logarithm of the volume of the S&P 500 index; the k-day continuously compounded returns on the crude oil futures contract; the first difference of the logarithm of US dollar foreign exchange value on seven currencies (Australian dollar, Canadian dollar, Swiss frac, euro, British sterling pound, Japanese yen, and Swedish kroner) and a trade-weighted average of the above foreign exchange values; the yield difference between Moody's seasoned Baa rated corporate bonds and Aaa rated corporate bonds (credit spread); the difference between 10-year and 3-month treasury constant maturity rates (term spread); the first difference of the logarithm of 10 other stock indices (NASDAQ-100, Dow Jones Industrial Average, FTSE ALL-Share index, FTSE-100 index, DAX Performance index, Swiss Market index, Nikkei 225, KOSPI index, Hang Seng index, and the BSE Sensex index); the first difference of the logarithm of world gold price; and the daily returns of the individual S&P 500 composites available since 1990 (266 return series). All data were retrieved from Thomson Reuters Datastream and Federal Reserve Economic Data (FRED).

[Table 1] Descriptive Statistics for Logarithm of VIX (April 5, 1990 – January 15, 2013)

Mean	2.951
Median	2.931
Minimum	2.231
Maximum	4.393
Standard Deviation	0.349
Skewness	0.547
Kurtosis	3.274
Jarque-Bera	0.000
ADF	0.000
PP	0.000
KPSS	0.064

Notes: Jarque-Bera, ADF, and PP respectively represent the p-values of Jarque-Bera, Augmented Dickey-Fuller, and Phillips-Perron tests. KPSS is the KPSS test statistic, and its critical values are 0.119, 0.146, and 0.216 at the levels of 10%, 5%, and 1%, respectively.

3.2. Forecasting Procedure

For direct comparison of results, this study uses the same forecasting methodology as Fernandes et al. (2014). Forecasts are made from a rolling window of a fixed size. For each model, a rolling window of 2,500 time-series observations is used to estimate the model, and 3,240 out-of-sample forecasts are produced (February 29, 2000 – January 15, 2013). Direct forecasts are made with no consideration of forecasting the exogenous covariates, as in Medeiros et al. (2019). Since the covariates are high-dimensional, it is natural to adopt the direct forecasting procedure, rather than the iterated forecasting procedure, for multi-step-ahead forecasts.

Four different forecasting horizons of k-day(s) (k=1, 5, 10, 22) are considered. k-day(s)-ahead forecasts are made and mean squared error (MSE) and mean absolute error (MAE) are calculated for each model and forecasting horizon. The average MSE and MAE of the random forest are compared to those of the benchmark models.

3.3. Benchmark Models

The benchmarks on which the results are compared are the models whose performances are reported upon in Fernandes et al. (2014); namely, they are the random walk (RW) model, Autoregressive model with exogenous variables (ARX), heterogeneous autoregressive (HAR) model of Corsi (2009), and HAR model with exogeneous variables (HARX). The model specifications follow those described in Fernandes et al. (2014). For the models including exogeneous covariates, the 14 variables used in Fernandes et al. (2014) are used.[3] In Section 4.4, as a robustness check, we also consider more machine learning methods.

IV. Results

4.1. Variable Rankings by the Boruta Algorithm

For each forecasting horizon, the full dataset containing the first and second lag[4] of all 299 variables is run on the Boruta algorithm. Each lag of each variable is treated as a separate variable, with the total number of covariates being 598. Of the 598 variables that were run on the Boruta algorithm, the number of variables confirmed to be 'important' are 138, 142, 147, and 123 for the 1-day-ahead, 5-days-ahead, 10-days-ahead, and 22-days-ahead forecasts, respectively. Table 2 lists the rankings of the top-30 variables determined from the Boruta algorithm for each forecasting horizon.[5]

Overall, the variability in the variable rankings among the different forecasting horizon settings does not seem to be large, especially for the variables ranked among the top-20. For all forecasting horizons, the lagged value of the logarithm of VIX recorded the largest mean Z-score, with quite a margin from the exogenous variables. In addition, for all forecasting horizon settings, the top-2 ranked

[3] The 14 variables are S&P k-day return, S&P 500 volume change, oil k-day return, trade-weighted USD change, credit spread, and term spread with k = 1,5,10,22,66.

[4] Following Fernandes et al. (2014), we consider the first and second lag of each covariate. One can use more lagged covariates. We also tried to use the full dataset containing additional third and fourth lag of all variables, and the results were similar.

[5] The full list of the rankings determined by Boruta is available upon request.

[Table 2] Variable Rankings Determined by Boruta Algorithm

	1-day-ahead	5-days-ahead	10-days-ahead	22-days-ahead
1	CBOEVIX(1)	CBOEVIX(1)	CBOEVIX(1)	CBOEVIX(1)
2	CBOEVIX(2)	CBOEVIX(2)	CBOEVIX(2)	CBOEVIX(2)
3	BaaAaa(1)	BaaAaa(1)	BaaAaa(1)	T10Y3M(1)
4	BaaAaa(2)	BaaAaa(2)	BaaAaa(2)	BaaAaa(1)
5	T10Y3M(1)	T10Y3M(1)	T10Y3M(1)	BaaAaa(2)
6	T10Y3M(2)	T10Y3M(2)	T10Y3M(2)	T10Y3M(2)
7	SP_66day(1)	SP_66day(2)	SP_66day(1)	SP_66day(1)
8	SP_66day(2)	SP_66day(1)	SP_66day(2)	SP_66day(2)
9	SP_22day(2)	SP_22day(2)	Oil_66day(1)	Oil_66day(2)
10	SP_22day(1)	SP_22day(1)	SP_22day(1)	Oil_66day(1)
11	SP_10day(1)	Oil_66day(1)	SP_22day(2)	SP_22day(2)
12	SP_10day(2)	Oil_66day(2)	Oil_66day(2)	SP_22day(1)
13	SP_5day(1)	SP_10day(2)	SP_10day(2)	SP_10day(2)
14	Oil_66day(2)	SP_10day(1)	SP_10day(1)	SP_10day(1)
15	Oil_66day(1)	SP_5day(1)	SP_5day(1)	SP_5day(1)
16	SP_5day(2)	SP_5day(2)	SP_5day(2)	SP_5day(2)
17	S&P_1day(1)	Oil_22day(2)	Oil_22day(2)	Oil_22day(2)
18	S&P_MV(1)	Oil_22day(1)	Oil_22day(1)	Oil_22day(1)
19	S&P_1day(2)	S&P_1day(1)	S&P_1day(1)	FITB(2)
20	S&P_MV(2)	S&P_MV(1)	S&P_MV(1)	FITB(1)
21	Oil_22day(2)	S&P_1day(2)	S&P_1day(2)	KEY(2)
22	DJINDUS(1)	S&P_MV(2)	S&P_MV(2)	AIG(1)
23	Oil_22day(1)	BAC(1)	BAC(1)	BAC(1)
24	GE(1)	DJINDUS(1)	FITB(2)	@HBAN(1)
25	BAC(1)	FITB(2)	GE(1)	AIG(2)
26	NASA100(1)	GE(1)	FITB(1)	BAC(2)
27	BAC(2)	BAC(2)	AIG(2)	S&P_1day(1)
28	DAXINDX(1)	DAXINDX(1)	DJINDUS(1)	KEY(1)
29	GE(2)	FITB(1)	NASA100(2)	S&P_MV(1)
30	DAXINDX(2)	DJINDUS(1)	BAC(2)	S&P_MV(2)

Notes: The following are the abbreviations used. CBOEVIX: lagged values of the VIX index; BaaAaa: credit spread; T10Y3M: term spread; SP_kday: k-day(s) returns on the S&P 500 index; Oil_kday: k-day(s) returns on oil futures; DJINDUS: daily returns on the Dow Jones Industrial Average; NASA100: daily returns on Nasdaq-100 Index; DAXINDX: daily returns on the DAX Performance Index; and S&P_MV: daily change in market volume of the S&P 500. All other abbreviations not mentioned correspond to the daily returns on individual S&P 500 composites, which are presented by their ticker symbols as listed on the New York Stock Exchange and the Nasdaq Stock Exchange. The numbers in parenthesis refer to the lag of each variable.

exogenous variables were credit spread and term spread (credit spread ranked first for 1/5/10-day(s)-ahead forecast setting while term spread ranked first for 22-days-ahead forecast setting). The list is followed by the continuously compounded multiple-days-returns on S&P 500 and oil futures and the daily change rate in the

S&P 500 index and the volume of the S&P 500. It is interesting to note that, aside from trade-weighted USD change, these top-20 variables roughly correspond to the 14 exogenous variables used in Fernandes et al. (2014).

The difference in rankings among the forecasting horizons seems to be more visible for variables ranked afterward, most of which are composed of daily returns on other stock market indices and daily returns on individual prices of S&P 500 composites. For the 1/5/10-day(s)-ahead forecast settings, at least two of the three main stock market indices (NASDAQ-100, Dow Jones Industrial Average, and DAX Performance index) made it into the top-30 list. Comparatively, the list for 22-days-ahead forecast setting is dominated by the daily return series of the individual S&P 500 composites.

These rankings are used to select the best subset of variables. That is, the optimal number of top ranked variables is derived, and this is considered as the dataset to be used for random forest.

4.2. The Optimal Number of Variables

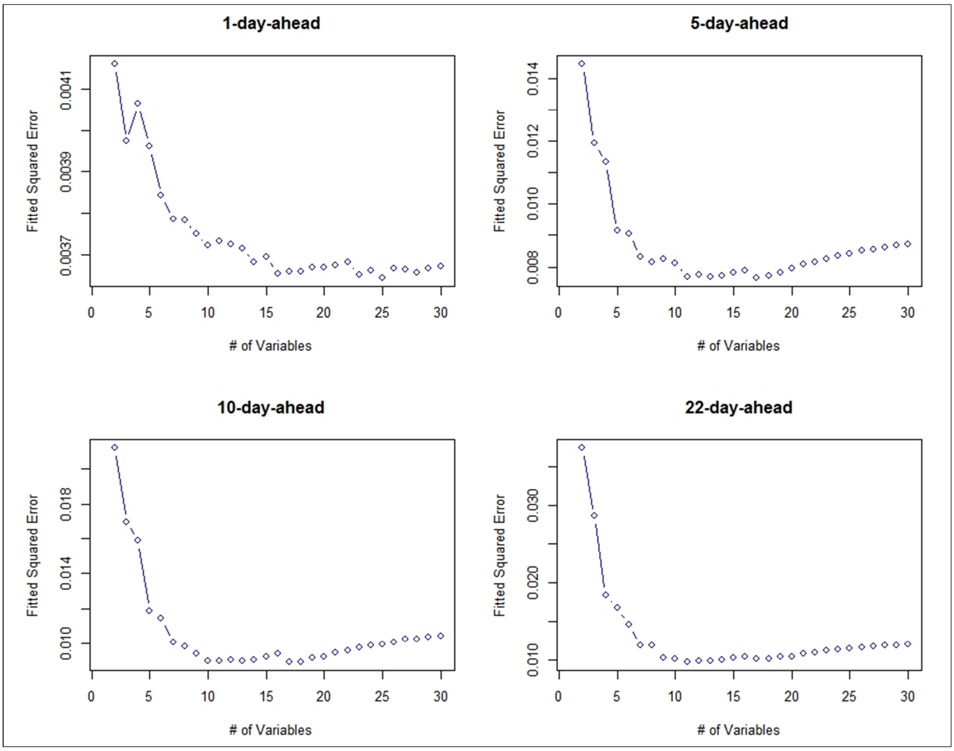
A cross validation procedure was conducted to determine the optimal number of variables to be used in a random forest method. The procedure is straight forward: First, start with a dataset with no exogenous variables using only the first and second lags of VIX as input variables into random forest. The dataset is run on the random forest and the in-sample OOB MSE is recorded. Next, add on to the dataset one exogenous variable at a time based on the order of the rankings decided by the Boruta algorithm in Table 2. Finally, record the in-sample OOB MSE from each dataset and find the number of variables that produces the smallest forecast error.

Figure 3 plots the change in the in-sample error as the dataset is sequentially expanded for each forecasting horizon. For the 1-day-ahead forecast, the picture seems less clear as it does not show a visible optimal point in terms of OOB MSE. However, for the 5/10/22-days-ahead forecasts, the results show a clear increasing trend in OOB MSE after a minimal point. According to the results, the optimal number of variables are 25, 17, 17, and 11 for the 1/5/10/22-day(s)-ahead forecasts, respectively. Thus, among the 598 covariates that are considered, the best subset of variables is derived for forecasting VIX in a random forest method while using the top-25/17/17/11 ranked variables according to the mean Z-scores derived from the Boruta algorithm.

4.3. Forecasting Results

Table 3 shows the results of the out-of-sample forecasts of each model for each forecasting horizon. The table reports the average MSE, MAE, and their relative ratios compared to the RW model. Among the four benchmark models from

[Figure 3] Number of Variables and In-Sample OOB MSE



Notes: For each forecasting horizon, the changes in in-sample out-of-bag (OOB) mean squared error (MSE) are plotted as variables that are added to the dataset based on the order of the rankings decided by the Boruta algorithm.

Fernandes et al. (2014), the pure HAR model shows the best performance overall, except for the 22-days-ahead forecast horizon in which the HARX model records a smaller forecast error by a slight margin. Fernandes et al. (2014) claims that the relative success of the pure HAR model can be attributed to the very persistent nature of the VIX index, and that it is difficult to outperform the pure HAR process.[6]

Comparing the above results to the performance of random forest on different datasets shows a substantially different picture. First, we consider a random forecast method using the same information as the benchmark models in Fernandes et al. (2014). The *RF(14)* in Table 3 shows the performance of random forest when using the dataset with only the exogenous covariates that were used by the benchmark

[6] Fernandes et al. (2014) also reports the forecasting results of the HAR model augmented with neural network (NNHARX). NNHARX outperforms HAR and HARX models only in the 22-days-ahead forecast setting, but the difference in forecast errors is negligible and statistically insignificant. Thus, it can be said that their results show little evidence of non-linearity.

[Table 3] Forecasting Performances at Different Horizons

	MSE	%	MAE	%	MSE	%	MAE	%	MSE	%	MAE	%	MSE	%	MAE	%
	One Day Ahead				*Five Days Ahead*				*Ten Days Ahead*				*Twenty-two Days Ahead*			
RW	0.0040		0.0458		0.0142		0.0891		0.0219		0.1115		0.0427		0.1540	
ARX	0.0040	1.00	0.0459	1.00	0.0136	0.96	0.0877	0.98	0.0214	0.98	0.1107	0.99	0.0408	0.95	0.1498	0.97
HAR	0.0039	0.97	0.0454	0.99	0.0133	0.94	0.0873	0.98	0.0209	0.96	0.1095	0.98	0.0399	0.93	0.1497	0.97
HARX	0.0040	1.00	0.0458	1.00	0.0136	0.96	0.0875	0.98	0.0214	0.98	0.1103	0.99	0.0409	0.96	0.1491	0.97
RF(14)	0.0044	1.10	0.0490	1.07	0.0104	0.73	0.0767	0.86	0.0135	0.62	0.0870	0.78	0.0177	0.41	0.0992	0.64
RF(598)	0.0047	1.19	0.0507	1.11	0.0126	0.89	0.0854	0.96	0.0170	0.78	0.0993	0.89	0.0244	0.57	0.1180	0.77
RF_Selected	0.0044	1.10	0.0492	1.07	0.0098	0.69	0.0744	0.84	0.0125	0.57	0.0837	0.75	0.0152	0.35	0.0916	0.59

Notes: Forecasting performances of different models for the test period from February 29, 2000 to January 15, 2013 (3,240 daily observations). The results of the benchmark models (RW/ARX/HAR/HARX) are derived using the same model specifications as those reported in Fernandes et al. (2014).

models in Fernandes et al. (2014). While the forecasting performance of RF(14) is even worse than the RW model for the 1-day-ahead forecast, the forecasting error drops significantly for 5/10/22-days-ahead forecasts compared to the pure and augmented HAR models.

Moreover, the relative accuracy of random forest compared to the RW and the linear benchmarks becomes more evident as the forecasting horizon increases. That is, the gap in forecasting error between the benchmark models and RF increases as the forecasting horizon approaches the 30-calendar-days-ahead threshold. The relative MAE of RF(14) compared to the RW model are 1.08, 0.86, 0.78, and 0.64 for the 1/5/10/22-day(s)-ahead forecasting horizons, respectively. This corroborates the results of Medeiros et al. (2019) which find that the forecasting superiority of random forest compared to the linear models becomes more evident with longer forecasting horizons.

The *RF(598)* in Table 3 shows the performance of random forest when the dataset includes all the first and second lags of 299 variables, whereas *RF_Selected* shows the performance of random forest using the dataset with the optimal number of the most important variables based on the Boruta algorithm, as derived through the variable selection process described in sections 4.1 and 4.2. The results show that selecting the optimal number of the most important variables using the Boruta algorithm further enhances the performance significantly. For the longer three forecasting horizons, RF_Selected is able to produce forecasts that are significantly more accurate than RF(14).[7]

Table 4 lists the p-values of the unconditional Giacomini-White (2006) test at different forecasting horizons. For the 1-day-ahead-forecasts, the forecasting ability of HAR is compared to all other models. It can be seen that the outperformance of HAR model in this setting is significant at the 5% level.

For the 5/10/22-days-ahead forecasts, the performance of RF is compared to those of the benchmark models. Using the same covariates, the superior accuracy of RF(14) is statistically significant at the 0.1% level compared to all linear benchmark models. Moreover, the superior predictive ability of RF_Selected is also significant at the 0.1% level compared to RF(14) and RF(598) . Thus, we can conclude that the gains from systematical variable selection using the Boruta algorithm is also statistically significant.

Table 5 reports the test results of model confidence sets (MCS) of Hansen et al. (2011) at different forecasting horizons. The test is based on squared error losses. The shaded cells are the models included in the 50% MCS, along with their p-values. The results are unambiguous, with only one model included in the MCS for

[7] We also calculated out-of-sample R^2 based on the Mincer-Zarnowitz regression. RF-Selected exhibits the highest R^2 for the longer three forecasting horizons. This result is not surprising because it is well known that out-of-sample R^2 based on the Mincer-Zarnowitz regression is equivalent to MSE if forecasts are unbiased.

each forecasting horizon, and all with a uniform p-value of 1. Thus, the HAR model seems to be the best model for the 1-day-ahead forecasts, while RF_Selected outperforms all other models for multi-day-ahead forecasts.

[Table 4] Giacomini-White Test for Predictive Ability

One Day Ahead	
	HAR
RW	0.0246
ARX	0.0056
HARX	0.0008
RF_14	0.0000
RF_298	0.0000
RF_Selected	0.0000

Five Days Ahead		
	RF_14	RF_Selected
RW	0.0000	0.0000
ARX	0.0000	0.0000
HAR	0.0000	0.0000
HARX	0.0000	0.0000
RF_14		0.0000
RF_298		0.0000

Ten Days Ahead		
	RF_14	RF_Selected
RW	0.0000	0.0000
ARX	0.0000	0.0000
HAR	0.0000	0.0000
HARX	0.0000	0.0000
RF_14		0.0000
RF_298		0.0000

Twenty-two Days Ahead		
	RF_14	RF_Selected
RW	0.0000	0.0000
ARX	0.0000	0.0000
HAR	0.0000	0.0000
HARX	0.0000	0.0000
RF_14		0.0000
RF_298		0.0000

Notes: The p-values of the Giacomini-White test for superior predictive ability between the HAR model and the other models for one-day-ahead setting, as well as between RF_14 & RF_Selected models and the other models for longer forecasting horizons.

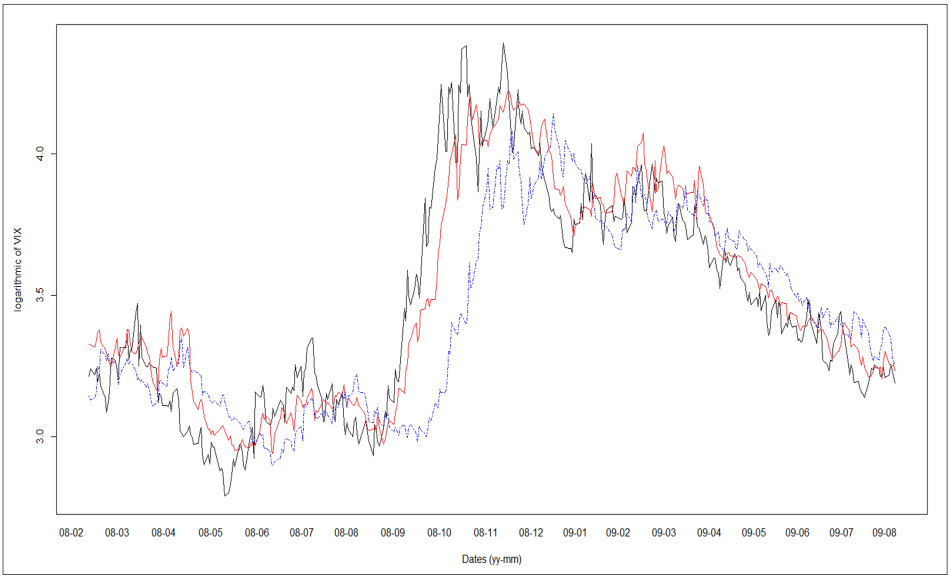
[Table 5] MCS Test Results

	Model Confidence Set			
	1-day	5-day	10-day	22-day
RW				
ARX				
HAR	1			
HARX				
RF_14				
RF_298				
RF_Selected		1	1	1

Notes: For each forecasting horizon setting, the shaded cells show the models that are included in the 50% Model Confidence Set (MCS), using squared error as the loss function. The MCS p-values are reported, where a higher p-value indicate that the model is more likely to be the "best" model.

Figure 4 compares the forecasts of RF_Selected and the HAR model for the period around the 2008 global financial crisis, specifically for the time stretching from February 2008 to August 2009. The black line shows the actual value of logarithmic of VIX, which reached its all-time peak in October 2008. The red and blue lines show the 22-days-ahead forecasts of RF_Selected and the HAR model, respectively. It can be seen that the forecasts by RF_Selected are more accurate than those of the HAR model. RF_Selected catches the sharp upward trend of VIX much faster after the collapse of Lehman Brothers in September 2008, and it does the same for the downward trend after the peak of the financial crisis. Overall, while the gap in forecasting error between the two models seem to be consistent over the whole forecasting period, it is in such highly volatile periods when the gap between the two models becomes much more evident.

[Figure 4] Comparison of Forecasts Between RF and HAR Model



Notes: The logarithm of VIX from February 13, 2008 to August 12, 2009 (black line), along with the 22-days-ahead forecasts of RF_Selected (red line) and HAR (blue line) models.

4.4. Robustness Check with More Machine Learning Methods

As a robustness check, we consider more machine learning methods and compare forecasts for the most recent 2-year period (January 2, 2019 to December 31, 2020, 505 daily observations). We adopt the same forecasting procedure and a rolling window of the same size (2,500 daily observations) is utilized. Thus, the dataset of our second sample runs from January 27, 2009 to December 31, 2020 with a total of 3,005 daily observations of all variables.

Along with the benchmark models reported in section 4.3, the following ML methods are included as additional benchmarks; the least absolute shrinkage and selection operator (LASSO), adaptive LASSO (adaLASSO), elastic net (Elnet),

[Table 6] Forecasting Performances for Second Sample

	MSE	%	MAE	%	MSE	%	MAE	%	MSE	%	MAE	%	MSE	%	MAE	%
	One Day Ahead				*Five Days Ahead*				*Ten Days Ahead*				*Twenty-two Days Ahead*			
RW	0.0067		0.0578		0.0288		0.1188		0.0570		0.1696		0.1282		0.2480	
ARX	0.0069	1.03	0.0563	0.97	0.0305	1.06	0.1149	0.97	0.0578	1.01	0.1577	0.93	0.1183	0.92	0.2198	0.89
HAR	0.0068	1.01	0.0564	0.98	0.0295	1.02	0.1160	0.98	0.0557	0.98	0.1553	0.92	0.1125	0.88	0.2136	0.86
HARX	0.0069	1.03	0.0567	0.98	0.0302	1.05	0.1144	0.96	0.0576	1.01	0.1565	0.92	0.1197	0.93	0.2215	0.89
LASSO	0.0068	1.01	0.0560	0.97	0.0301	1.04	0.1146	0.96	0.0562	0.99	0.1515	0.89	0.1158	0.90	0.2150	0.87
adaLASSO	0.0068	1.01	0.0567	0.98	0.0295	1.02	0.1139	0.96	0.0554	0.97	0.1531	0.90	0.1143	0.89	0.2163	0.87
Elnet	0.0069	1.02	0.0559	0.97	0.0309	1.07	0.1152	0.97	0.0568	1.00	0.1518	0.89	0.1160	0.90	0.2153	0.87
adaElnet	0.0066	0.99	0.0562	0.97	0.0298	1.03	0.1138	0.96	0.0552	0.97	0.1528	0.90	0.1132	0.88	0.2158	0.87
CSR	0.0066	0.99	0.0555	0.96	0.0289	1.00	0.1138	0.96	0.0541	0.95	0.1529	0.90	0.1079	0.84	0.2089	0.84
tFact	0.0066	0.98	0.0557	0.96	0.0291	1.01	0.1162	0.98	0.0548	0.96	0.1499	0.88	0.1142	0.89	0.2182	0.88
NN	0.0102	1.51	0.0690	1.19	0.0311	1.08	0.1237	1.04	0.0558	0.98	0.1641	0.97	0.0849	0.66	0.1989	0.80
RF(14)	0.0085	1.26	0.0614	1.06	0.0241	0.84	0.1067	0.90	0.0357	0.63	0.1236	0.73	0.0556	0.43	0.1498	0.60
RF(598)	0.0094	1.40	0.0639	1.11	0.0303	1.05	0.1202	1.01	0.0466	0.82	0.1440	0.85	0.0820	0.64	0.1794	0.72
RF_Selected	0.0085	1.27	0.0617	1.07	0.0230	0.80	0.1041	0.88	0.0323	0.57	0.1163	0.69	0.0460	0.36	0.1380	0.56

Notes: Forecasting performances of different models for the test period from January 2, 2019 to December 31, 2020 (505 daily observations), using a rolling window of 2,500 daily observations. The results of the benchmark models (RW/ARX/HAR/HARX) are derived using the same model specifications as those reported in Fernandes et al. (2014).

adaptive elastic net (adaElnet), complete subset regression (CSR), target factors (tFact), and a deep neural network (NN)[8] with two hidden layers and 32 and 16 nodes in each hidden layer. The model specifications of the ML benchmarks are described in Appendix A.

For random forest, the same variable selection process as described in sections 4.1 and 4.2 was used to determine the predictors for RF_Selected. The rankings of the variables decided by the Boruta algorithm show little variation from those reported in Table 2, particularly among the top-20 variables.[9] From the cross-validation procedure, the optimal numbers of variables for RF are found to be 21, 12, 14, and 14 for the 1/5/10/22-day(s)-ahead forecast horizons, respectively.

Table 6 provides the forecasting results at different horizons. Among the benchmark models, the pure HAR model is no longer the outperforming model when the ML benchmarks are added. For the 1/5/10-day(s)-ahead forecasts, the CSR reports the smallest errors among the benchmarks. However, the difference compared to the rest of the benchmark models is quite minimal. For the 22-days-ahead forecasts, the NN model records the smallest forecast error among the benchmark models.

The comparison between RF and the benchmarks shows a picture that is in line with that of the previous section. While RF(14) performs worse than the RW model for the 1-day horizon, it produces forecasts that are more accurate than all other benchmark models in multi-days horizons. The Giacomini-White test results in Table 7 confirm that the predictive ability of RF(14) is superior to those of the benchmarks for the multi-step-ahead settings. For the 5-days horizon, the null hypotheses of equal predictability between RF(14) and the benchmarks are rejected at the 5% significance level, except for two cases (against LASSO and adaElnet) where they are only rejected at the 10% level. For the 10-days and 22-days horizons, RF(14) outperforms all benchmark models at the 1% level.

Moreover, as was the case in the previous section, the gap in forecast error between RF and the linear benchmark models is magnified for the longer forecasting horizons. The relative MAE of RF(14) to the RW model are 1.06, 0.90, 0.73, and 0.60 for the 1/5/10/22-day(s) horizons, respectively.

The accuracy of RF_Selected demonstrates the benefits of using the Boruta algorithm for variable selection. For the multi-step-ahead forecasts, RF_Selected is clearly the outperforming model overall with the smallest forecast error. In the 5-day horizon, the Giacomini-White test between RF_Selected and all other models except RF(14) is rejected at the 5% level. For 10-day and 22-day horizons, RF_Selected significantly outperforms all other models including RF(14) at the 1%

[8] The optimal numbers of layers and nodes for NN are decided in advance based on cross validation procedures.

[9] The rankings from the Boruta algorithm for the second sample are available upon request.

level. Moreover, RF_Selected is the only model included in the 50% MCS with a p-value of 1 for the multi-step horizons.

[Table 7] Giacomini-White Test for Predictive Ability (Second Sample)

One Day Ahead	
	CSR
RW	0.0002
ARX	0.1001
HAR	0.0023
HARX	0.0350
LASSO	0.1111
adaLASSO	0.0027
Elnet	0.1858
adaElnet	0.0483
tFact	0.2086
NN	0.0000
RF_14	0.0000
RF_298	0.0000
RF_Selected	0.0000

Five Days Ahead		
	RF_14	RF_Selected
RW	0.0185	0.0031
ARX	0.0434	0.0157
HAR	0.0043	0.0013
HARX	0.0476	0.0175
LASSO	0.0525	0.0260
adaLASSO	0.0488	0.0186
Elnet	0.0477	0.0245
adaElnet	0.0519	0.0186
CSR	0.0293	0.0123
tFact	0.0032	0.0012
NN	0.0002	0.0000
RF_14		0.1281
RF_298		0.0000

Ten Days Ahead		
	RF_14	RF_Selected
RW	0.0000	0.0000
ARX	0.0001	0.0000
HAR	0.0000	0.0000
HARX	0.0002	0.0000
LASSO	0.0010	0.0001
adaLASSO	0.0002	0.0000
Elnet	0.0016	0.0002
adaElnet	0.0001	0.0000
CSR	0.0000	0.0000
tFact	0.0010	0.0001
NN	0.0000	0.0000
RF_14		0.0010
RF_298		0.0000

Twenty-two Days Ahead		
	RF_14	RF_Selected
RW	0.0000	0.0000
ARX	0.0021	0.0011
HAR	0.0008	0.0009
HARX	0.0026	0.0013
LASSO	0.0025	0.0017
adaLASSO	0.0014	0.0008
Elnet	0.0026	0.0019
adaElnet	0.0015	0.0008
CSR	0.0006	0.0006
tFact	0.0015	0.0007
NN	0.0001	0.0000
RF_14		0.0020
RF_298		0.0010

Notes: The p-values of the Giacomini-White test for superior predictive ability between the CSR model against the other models for one-day-ahead setting, as well as between RF_14 & RF_Selected models against the other models for multi-days-ahead forecasting horizons.

[Table 8] MCS Test (Second Sample)

	Model Confidence Set			
	1-day	5-day	10-day	22-day
RW	1			
ARX	0.7112			
HAR	0.9844			
HARX	0.5536			
LASSO	0.9660			
adaLASSO	0.9950			
Elnet	0.7202			
adaElnet	1			
CSR	1			
tFact	1			
RF_14				
RF_298				
RF_Selected		1	1	1

Notes: For each forecasting horizon setting, the shaded cells show the models that are included in the 50% Model Confidence Set (MCS), using squared error as the loss function. The MCS p-values are reported, where a higher p-value indicates that the model is more likely to be the "best" model.

V. Conclusion

The paper seeks to apply random forest and its variable importance measure to forecasting the CBOE Volatility Index (VIX). In particular, it seeks to improve upon the multi-days-ahead forecasting of VIX relative to those reported in the previous literature. Compared to the results of Fernandes et al. (2014), which find it is very hard to beat the pure HAR process in forecasting VIX, random forest could produce forecasts that are significantly more accurate than the HAR and augmented HAR models for multi-days forecasting horizons. Moreover, the superior predictability of random forest compared to the RW and benchmark linear models becomes more apparent as the forecasting horizon becomes longer. This is consistent with Medeiros et al.'s (2019) findings in the context of forecasting US inflation.

Further improvements in forecasting performance are attained through a systematic selection of covariates among a high-dimensional dataset. Utilizing the Boruta algorithm, the rankings of the variables are extracted based on the permutation importance measure of random forest. Adopting only the optimal number of the selected most important covariates significantly enhances the forecasting accuracy of random forest. It seems clear that variable selection functions as a crucial factor affecting the predictability of random forest.

The robustness of the main results of the paper are confirmed through forecasting on the most recent period from 2018 to 2020. Moreover, compared to various other ML methods, the random forest method utilizing the Boruta algorithm provides superior multi-step-ahead forecasts.

While this paper focuses solely on the random forest, it would be interesting to investigate other ML methods that can capture the nonlinear characteristics of VIX, especially deep learning methods such as long short-term memory. It would also be interesting to investigate whether our methodology can provide better multi-step-ahead forecasts of macroeconomic time series. These issues represent future research topics.

Appendix A

1. The Shrinkage Methods

To improve ordinary least squares (OLS) regression, the family of shrinkage methods takes the form of penalized regression. This is similar to an OLS regression in that the objective is to minimize the residual sum of squares (RSS), but they add a term that imposes a size constraint on the coefficient estimates;

$$\hat{\beta} = arg\ min_{\beta_j}\left[\sum_{i=1}^{n}(y_i - \boldsymbol{X}_i\hat{\beta}) + \sum_{j=1}^{p} p(\beta_j;\lambda)\right].$$

The different models among these methods are distinguished by the penalty function $p(\beta_j;\lambda)$, which regularizes the coefficient estimates and shrinks the coefficients of variables with less explanatory power. The shrinkage penalty term depends on the tuning parameter λ, which regulates the amount of shrinkage imposed on the coefficients; a higher λ results in a stronger shrinkage of the regression coefficients, while $\lambda = 0$ would reduce the model to an OLS with no shrinkage.

1.1. Least Absolute Shrinkage and Selection Operator (LASSO)

LASSO was proposed by Tibshirani (1996), in which the penalty function is given as

$$\sum_{j=1}^{p} p(\beta_j;\lambda) = \lambda \sum_{j=1}^{p} |\beta_j|.$$

Compared to the primitive shrinkage method—ridge regression, described by Hoerl & Kennard (1970)—the L_1 penalty of LASSO is able to shrink the less relevant variables to exactly zero via soft thresholding, and it thus presents the feature of variable selection. Moreover, due to the absolute value operator in the penalty term, LASSO does not have a closed form solution, and it is computed through algorithmic methods.

1.2. Adaptive LASSO

Consistency of variable selection by LASSO is only achieved under strict conditions, and Zou (2006) proposed the adaptive LASSO to overcome this issue.

The penalty term includes a weighting parameter that is derived from a first-step estimation. The penalty function is given as

$$\sum_{j=1}^{p} p(\beta_j;\lambda,w_j)=\lambda\sum_{j=1}^{p} w_j \,|\,\beta_j\,|,$$

where adaptive weights $w_j = |\frac{1}{\hat{\beta}_j}|^{-1}$ are used to penalize different coefficients in the LASSO penalty. Adaptive LASSO can be solved through the same efficient algorithm used to solve LASSO.

1.3. Elastic Net

The elastic net is a compromise between ridge regression and LASSO. While it retains the variable selection feature of LASSO, it also shrinks the coefficients of correlated variables toward each other similar to the ridge regression. The penalty function takes the form of a weighted mean of ridge and LASSO penalties:

$$\sum_{j=1}^{p} p(\beta_j;\lambda)=\lambda\sum_{j=1}^{p} \alpha\beta_j^2+(1-\alpha)\,|\,\beta_j\,|.$$

The elastic net includes the special cases of LASSO ($\alpha=0$) and ridge regression ($\alpha=1$). In this paper, the α parameter is set to be 0.5. This paper also considers an adaptive version of elastic net which includes adaptive weights as in adaptive LASSO.

2. Complete Subset Regression

Another possible approach to handling a high-dimensional dataset is subset selection for linear regression. While there are a number of strategies that can be used for subset selection, testing all possible combinations of predictor variables is computationally demanding and becomes infeasible when there is a very large number of candidate variables.

Complete subset regression (CSR) proposed by Elliott et al. (2013, 2015) takes an ensemble approach. For a given set of potential regressors, CSR combines forecasts from all possible linear regression models while keeping the number of predictors fixed. For a dataset with K possible regressors, the number of k-variate models ($k \le K$) is $n_{k,K} = \frac{K!}{((K-k)!k!)}$. The set of models for a fixed value k is referred to as a complete subset, and the final forecast made by CSR is the equal-weighted average of forecasts from all models within the complete subset indexed by k. In this paper, we use $k = 4$ to calculate the CSR forecasts.

3. Target Factors

Numerous forecasting methodologies using factor augmented models have recently been developed. The idea of these factor models is to first estimate the factors from a large number of predictors using the method of principal components, and then to augment these factors to a linear forecasting equation. To refine the factor augmented forecasting methodology, Bai & Ng (2008) proposed targeting the predictors using hard and soft thresholding rules. The underlying rationale is that computing the principal components from all predictors may result in noisy factors, and that only the predictors with high forecasting power should be used.

This paper uses the hard thresholding method suggested in Bai & Ng (2008) and implemented by Medeiros et al. (2019). Let y_i be the dependent variable or the logarithm of VIX, let $X_j (j = 1, \ldots, p)$ be the candidate predictors, and let W_i be a set of controls. Following Bai & Ng (2008), the lagged values of y_i and a constant are used as W_i.

1. For $j = 1, \ldots, p$, perform a regression of y_{i+h} on W_i and X_j and compute the t-statistics corresponding to the coefficient of X_j.
2. Choose a significance level α and find the set of significant variables $z_i(\alpha)$ based on the computed t-statistics.
3. Estimate the factors F_i from $z_i(\alpha)$.
4. Regress y_{i+h} on W_i and f_i, where $f_i \subset F_i$ and the number of factors in f_i is decided using BIC.

4. Computation and Codes

All ML methods in Section 4 are estimated in R using standard and well-established packages. The R codes are available online at https://github.com/bryank2242/Forecasting-VIX. We modified the R codes provided by Medeiros et al. (2019), which are available online at https://github.com/gabrielrvsc/. Readers are referred to Section 4.7 in Medeiros et al. (2019) for more details of computer codes and tuning parameters. The Boruta package is available at https://cran.r-project.org/web/packages/.

References

Bai, J., and S. Ng (2008), "Forecasting Economic Time Series using Targeted Predictors," *Journal of Econometrics*, 146, 304–317.

Ballestra, L. V., A. Guizzardi, and F. Palladini (2019), "Forecasting and Trading on the VIX Futures Market: A Neural Network Approach Base on Open to Close Returns and Coincident Indicators," *International Journal of Forecasting*, 35, 1250–1262.

Breiman, L. (2001), Random Forests, *Machine Learning*, 45, 5–32.

Breiman, L., J. Friedman, R. Olshen, and C. Stone (1984), *Classification and Regression Trees*, Wadsworth Books.

Corsi, A. (2009), "A Simple Approximate Long-memory Model of Realized Volatility," *Journal of Financial Econometrics*, 7, 174–196.

Degiannakis, S. A. (2008), Forecasting VIX. *Journal of Money, Investment and Banking*, 4, 5–19.

Elliott, G., A. Gargano, and A. Timmermann (2013), "Complete Subset Regressions," *Journal of Econometrics*, 177(2), 357–373.

Elliott, G., A. Gargano, and A. Timmermann (2015), "Complete Subset Regressions with Large Dimensional Sets of Predictors," *Journal of Economic Dynamics and Control*, 54, 86–110.

Fernandes, M., M. C. Medeiros, and M. Scharth (2014), "Modeling and Predicting the CBOE Market Volatility Index," *Journal of Banking and Finance*, 40, 1–10.

Giacomini, R., and H. White (2006), "Tests of Conditional Predictive Ability," *Econometrica*, 74, 1545–1578.

Hansen, P. R., A. Lunde, and J. M. Nason (2011), "The Model Condence Set," *Econometrica*, 79(2), 453–497.

Hastie, T., R. Tibshirami, and J. Friedman (2001), *The Elements of Statistical Learning; Data Mining, Inference and Prediction*, Springer.

Hoerl, A. E., and R. W. Kennard (1970), "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, 12(1), 55–67.

Kohavi, R., and G. H. John (1997), "Wrappers for Feature Subset Selection," *Artificial Intelligence*, 97, 273–324.

Konstantinidi, E., G. Skiadopoulos, and E. Tzagkaraki (2008), "Can the Evolution of Implied Volatility be Forecasted? Evidence from European and US Implied Volatility Indices," *Journal of Banking and Finance*, 32, 2401–2411.

Kursa, M. B., and W. R. Rudnicki (2010), "Feature Selection with the Boruta Package," *Journal of Statistical Software*, 36, 1–13.

Medeiros, M. C., G. F. Vasconcelos, A. Veiga, and E. Zilberman (2019), "Forecasting Inflation in a Data-rich Environment: The Benefits of Machine Learning Methods," *Journal of Business & Economic Statistics*, 1–45.

Mullainathan, S. and J. Spiess (2017), "Machine Learning: An Applied Econometric Approach," *Journal of Economic Perspectives*, 31(2), 87–106.

Psaradellis, I., and G. Sermpinis (2016), "Modelling and Trading the U.S. Implied Volatility

Indices. Evidence from the VIX, VXN and VXD Indices," *International Journal of Forecasting*, 32, 1268–1283.

Speiser, J. L., M. E. Miller, J. Tooze, and E. Ip (2019), "A Comparison of Random Forest Variable Selection Methods for Classification Prediction Modeling," *Expert Systems with Applications*, 134, 93–101.

Tibshirani, R. (1996), "Regression Shrinkage and Selection Via the LASSO," *Journal of the Royal Statistical Society. Series B (Methodological),* 58, 267–288.

Zou, H. (2006), "The Adaptive Lasso and its Oracle Properties," *Journal of the American Statistical Association*, 101(476), 1418–1429.

풍부한 데이터를 활용한 CBOE 변동성 지수(VIX) 예측: 보루타 연산(Boruta algorithm)과 랜덤 포레스트(random forest)의 적용*

김 병 연** · 한 희 준***

초 록 시카고 옵션거래소(CBOE) 변동성 지수(VIX)는 향후 미국 주식시장 변동성에 대한 투자자들의 기대를 반영하는 지표로서, 오늘날 금융시장의 변동성과 전반적인 심리에 대한 대표적인 척도이다. VIX 지수 예측과 관련한 기존 논문들은 단순한 HAR(Heterogeneous Autoregressive) 모형의 예측력이 우수하다는 것을 보였는데, 대부분의 경우 예측에 사용한 설명변수들의 개수가 제한되어 있고 또한 1단계 앞 예측(one-step-ahead forecasting)만을 고려하고 있다. 본고는 고차원(high-dimension)의 설명 변수(총 298개의 거시/금융 변수)를 사용하면서 다양한 기계학습(machine learning) 기법들을 적용하여 VIX 지수의 다단계 앞 예측(multi-step-ahead forecasting)을 분석한다. 특히 랜덤 포레스트(random forest) 환경에서 변수 선택(variable selection)과 최적 변수의 개수를 결정하는 새로운 방법을 제시하고, 이 방법이 다단계 앞 예측(multi-step-ahead forecasting)에 우수함을 보이고 있다. 구체적인 예측 절차는 1) 보루타 연산(Boruta algorithm)을 통해 변수 중요도(variable importance)의 순위를 도출하고, 2) 교차 검증(cross validation)을 통해 최적 변수의 개수를 결정하고, 3) 중요도가 높은 변수들을 2)에서 정해진 개수만큼만 사용하여 랜덤 포레스트를 실시하는 것이다. 다양한 표본 기간 및 예측 기간을 고려해도, 이 방법을 통해 통계적으로 유의하게 우월한 VIX 지수의 다단계 앞 예측치(multi-step-ahead forecast)들을 구할 수 있음을 보이고 있다.

핵심 주제어: 랜덤 포레스트, 보루타 연산, 기계학습, VIX 지수, 변동성 예측

경제학문헌목록 주제분류: C01, G17

투고 일자: 2021. 8. 18. 심사 및 수정 일자: 2022. 3. 7. 게재 확정 일자: 2022. 4. 25.

* 이 논문은 성균관대학교의 2019학년도 성균학술연구비에 의하여 연구되었다. 논문에 대한 유익한 조언을 해준 김창식 교수님과 익명의 두 심사자께 깊은 감사를 표한다.

** 제1저자, 성균관대학교 경제학과 석사 / SC제일은행 Risk Anaytics Analyst, e-mail: bryank2242@gmail.com

** 교신저자, 성균관대학교 경제학과 / 퀀트응용경제학과 교수, e-mail: heejoonhan@skku.edu