# Forecasting Realized Volatility: An Automatic System Using Many Features and Many Machine Learning Algorithms

Sophia Zhengzi Li* and Yushan Tang†

November 18, 2020

**Abstract**

We propose a machine learning-based automatic system for forecasting realized volatility. The system consists of two components: feature engineering and learning algorithm fitting. The system is automatic in that it requires little intervention in terms of feature and model selection. Using this system, we demonstrate large gains based on 118 features with five machine learning algorithms to forecast realized volatility for 173 stocks spanning two decades. High-frequency based realized features from existing risk models (HAR, MIDAS, SHAR, HARQ-F and HExpGl) are particularly important in forecasting daily and weekly realized volatility, whereas implied variance features from put and call options are more crucial in predicting monthly and quarterly realized volatility. Both sparse linear models (LASSO, PCR) and nonlinear algorithms (RF, GBRT, NN) can improve over the commonly used ordinary least squares models. A final ensemble model combining all machine learning algorithms delivers extraordinary performance across all forecast horizons, with $R^2_{OOS}$ relative to HAR prediction equal to 9.3%, 14.0%, 15.0% and 10.4% at daily, weekly, monthly and quarterly forecast horizons.

**Keywords**: Automation; Volatility Forecasting; Machine Learning; High-Frequency Data; Realized Variance.

*Rutgers Business School, Newark, NJ 07102; E-mail: zhengzi.li@business.rutgers.edu.
†Rutgers Business School, Newark, NJ 07102; E-mail: yushan.tang@rutgers.edu.

# 1. Introduction

Forecasting volatility is crucial in risk management and asset pricing in general. The availability of high-frequency price data over the past two decades has spurred the field of modeling and forecasting realized variance $RV$, estimated by summing up squared intraday returns.[1] Most of the existing volatility forecasting models propose a handful of new predictors and then examine them one-by-one within the framework of classical statistical inference. In this paper, we focus on the out-of-sample prediction rather than statistical inference. Our objective is to build an automatic forecasting system that: 1) reduces human intervention in choosing features and algorithms; 2) scales to fit many features while controlling for overfitting; 3) utilizes more flexible and state-of-the-art learning algorithms; and 4) achieves good and consistent out-of-sample performance.

Our system has two main components: feature engineering and learning algorithm fitting. In the feature engineering step, we aim to include many features that might be useful in predicting future volatility. We do not follow traditional approaches that examine each individual feature for its statistical significance. Instead, we let the automatic learning system decide which and how features should be selected at a given point of time. We consider three types of features: the realized variance-based ($RV$-based) features that consist of all 16 predictors proposed by five popular $RV$ forecasting models; the implied variance-based ($IV$-based) features that consist of 102 option implied variances at the stock level across all deltas and with maturity between one and three months; derived features that are transformations of the $IV$- and $RV$- based features. Our feature set is, to the best of our knowledge, the largest that has ever been examined in the volatility forecasting literature. In the learning step, we aim to learn the relation between volatility and features by various learning algorithms. We do not limit the relation to be linear via simple OLS. Instead, we consider methods that are more prediction-oriented and capable of capturing more complicated relations. Specifically, we apply five popular machine learning (ML) algorithms including two linear ones: LASSO and Principal Component Regression (PCR), and three nonlinear

[1] In the early volatility forecasting literature, competing ARCH-type and stochastic volatility models were proposed for estimating and forecasting volatility, see the review articles by Bollerslev, Engle, and Nelson (1994) and Andersen, Bollerslev, Christoffersen, and Diebold (2006). In both types of models, volatility is latent in nature which complicates the model implementation. Andersen and Bollerslev (1998) originally proposed the use of realized volatility for accurately measuring the true latent integrated volatility; Andersen, Bollerslev, Diebold, and Labys (2003) suggested using reduced-form time series forecasting models for realized volatilities.

ones: Random Forecast (RF), Gradient Boosted Regression Trees (GBRT), and Neural Network (NN). We automate the learning process of these models by setting up the procedure that can automatically and dynamically control the model complexity to reduce overfitting.

We illustrate the automatic forecasting system through a large scale experiment that compares different combinations of features and learning algorithms. Our main findings are: 1) including all $RV$-based features from popular $RV$ forecasting models improves over any stand-alone model even through a simple OLS fit; 2) further including all $IV$-based features can improve the out-of-sample forecasting performance; 3) dynamically fitting the same feature set with machine learning algorithms can further improve the performance over OLS; and 4) an ensemble model that uses all features and all machine learning algorithms performs extraordinarily well across forecast horizons and under different market conditions.

We start by constructing a comprehensive dataset comprising 1-minute data of stocks that were ever constituent of the S&P 100 index from the date TAQ data became available to June 2019. We then compute $RV$-based features using this dataset for each date and each stock over the period of January 1996 and June 2019. We further collect $IV$ features from both put and call options with absolute delta between 0.1 and 0.9 and maturity between one and three months from OptionMetrics.[2] Our final stock sample consists of 173 unique stocks that have at least five years of data on all features and response variables over the period of January 1996 and June 2019. Our stock universe is large-scale by the volatility forecasting literature standard.[3] Given the well-known commonalities in the dynamic dependencies of volatilities and spillover effects across assets, we purposely estimate all models across all stocks using panel data which adds power to individual stock fitting.[4]

We begin our evaluation with four OLS-based volatility forecasting models including the MIDAS model by Ghysels, Santa-Clara, and Valkanov (2006) and Ghysels, Sinko, and Valkanov (2007), the SHAR model by Patton and Sheppard (2015), the HARQ-F model by Bollerslev, Patton, and

[2] $IV$s from options with ten-day maturity only became available in November 2005 for a handful of stocks.

[3] Another paper we are aware of that uses such large dataset in the volatility forecasting literature is Patton and Sheppard (2015), which relies on 105 unique stocks that were constituents of the S&P 100 index and with four-year continuous data between June 1997 and July 2008. The focus on S&P 100 stocks helps ensure all stocks are frequently traded and thus their realized features based on intraday data are less subject to measurement error.

[4] Volatility spillover effects and commonalities in the dynamic dependencies are well documented in the traditional GARCH and stochastic volatility models, see Taylor (2005), and Andersen, Bollerslev, Christoffersen, and Diebold (2006) and the references therein. Recent work by Herskovic, Kelly, Lustig, and Nieuwerburgh (2016), Bollerslev, Hood, Huss, and Pedersen (2018), and Herskovic, Kelly, Lustig, and Nieuwerburgh (2021) further highlights the co-movement of stock volatilities over time.

Quaedvlieg (2016b), and the HExpGl model by Bollerslev, Hood, Huss, and Pedersen (2018). The prediction from the popular HAR model by Corsi (2009) is used as a benchmark when calculating out-of-sample $R^2$. We find that the forecasting performance of the stand-alone risk models can be improved once we combine their feature sets. We next show that implied variances contain unique information; once included in the feature set, the forecasting performance of OLS can be further enhanced. Motivated by the "bet-on-sparsity" principle, we provide evidence showing that sparse linear models such as LASSO can improve over OLS. Then we demonstrate nonlinear algorithms can add value to linear methods. Lastly, we show that an ensemble model by combining all machine learning algorithms delivers extraordinary performance over all forecast horizons, with $R^2_{OOS}$ relative to HAR equal to 9.3%, 14.0%, 15.0% and 10.4% at daily, weekly, monthly and quarterly horizons, respectively.

To further understand which features are the most important in forecasting volatilities, we use the ensemble model as an example to calculate variable importance metric for each predictor at each forecast horizon. We find that $RV$-based features are relatively more important at shorter daily and weekly horizons whereas $IV$-based features are more crucial for longer monthly and quarterly horizons. When ranked by their overall variable importance across forecast horizons, the top-four features are all implied variances, further highlighting the importance of using implied variances to forecast realized volatilities.

Our contributions to the literature are on two general aspects: methodology and empirics. On methodology, we propose a modern machine learning-based framework for volatility forecasting. Within this framework, we decompose the volatility forecasting task into two steps: feature engineering and learning algorithm fitting. For features, rather than examining individual features one-by-one to test their significance, we include many features *all* together. We use learning algorithms along with prediction-based model selection procedures to *automatically* and *dynamically* select features. For learning algorithms, we go beyond OLS to include major linear and nonlinear learning algorithms. We do not argue for the dominance of one particular algorithm over another. Instead, we consider the combination of all learning algorithms as long as they are well implemented to avoid overfitting. Therefore, our framework is less prone to human decision-making biases (e.g., cherry-picking of features and models) and interventions (e.g., using one set of features or models for the financial crisis period) and appears to be robust throughout our analyses.

On empirics, we conduct perhaps one of the largest scaled experiments for forecasting realized volatility of individual stocks. Our big dataset consists of intraday high-frequency data and stock-level option data for 173 individual stocks between January 1996 and June 2019. Our giant feature set includes predictors from five popular $RV$-based volatility forecasting models and implied variances ($IV$) with one- to three-month maturity across all deltas. Our learning algorithms consist of major linear and nonlinear machine learning models. Through this comprehensive data and design, we empirically demonstrate the gains of using the new automatic system based on many features and many learning algorithms to forecast realized volatility.

There is a burgeoning interest in applying machine learning (ML) techniques to asset pricing. Gu, Kelly, and Xiu (2020) show that ML methods can generate robust forecasting power to predict stock returns in the cross section and time series. Bianchi, Büchner, and Tamoni (2021) use ML algorithms to predict treasury bonds returns. Bali, Goyal, Huang, F., and Wen (2020) study cross-sectional predictability of corporate bond returns using both stock and bond characteristics via ML. There are also several papers that apply selective ML algorithms on volatility forecasting problems: Audrino and Knaus (2016) use LASSO to forecast realized volatility; Luong and Dokuchaev (2018) forecast realized volatility with random forest algorithm; Rossi (2018) employs Boosted Regression Trees to forecast stock returns and volatility at monthly frequency; Bucci (2020) and Rahimikia and Poon (2020) apply neural network to predict realized volatility; Carr, Wu, and Zhang (2020) rely on Ridge, Feedforward Neural Networks and Random Forecast to predict realized variance of SPX using option price as features. Compared with these work on applying machine learning to volatility forecasting, our work focuses on building the entire learning system that is automatic and robust. We emphasize the benefit of not just one or two particular learning algorithms such as RF or GBRT, but the benefit of the ML-based system that allows us to consider features and algorithms more inclusively, because machines are able to scan, fit, and select them in a more robust and prediction-error-optimized fashion. In addition, our empirical experiment is quite comprehensive in terms of the stock universe, sample history, features, and learning algorithms.

## 2. Data and Response Variable

### *2.1. Data*

We consider a large universe of stocks that were ever a constituent of the S&P 100 index over the period of January 1993 and June 2019. For stocks in this universe, we keep those listed on the New York Stock Exchange (NYSE), National Association of Securities Dealers Automated Quotations (NASDAQ), and American Stock Exchange (AMEX) with share code of 10 or 11, price between \$1 and \$1,000, and daily number of trades greater than or equal to 100. To prepare their intraday price data, we collect minute-by-minute observations of intraday prices from the NYSE trade and quote (TAQ) database by applying the cleaning rules of Bollerslev, Li, and Todorov (2016a), Bollerslev, Li, and Zhao (2020), and Jiang, Li, and Wang (2021) to the TAQ database.[5] We rely on the data between January 1993 and December 1995 to compute some of the high-frequency based realized features that require a longer history for construction; all realized features become available from January 1996.

In addition, we collect implied variances for the same universe of stocks from the volatility surface data in OptionMetrics. The database provides implied volatilities with various maturities and deltas at stock and date level. In our empirical analyses, we rely on implied variances (i.e., squared implied volatilities) from call and put options with maturity of one month (30 days), two months (60 days) and three month (91 days), and absolute delta of 0.1, 0.15, ..., 0.9.[6] Our final stock sample consists of 173 unique stocks with at least five years of data on all features and response variables over the period of January 1996 and June 2019.

### *2.2. Response Variable*

As in every predictive problem, we need to first define what exactly we are trying to predict. In this paper, we aim to predict realized variance ($RV$) which is a consistent estimator of the quadratic variation of the log price process over a given period. Formally, let $p_{i,t}$ denote the natural logarithm of stock $i$'s price on day $t$. We omit subscription $i$ in this section for simplicity and assume the log

[5] Further details on the TAQ data cleaning rules are provided in the Appendix.

[6] Implied variances with ten-day maturity only became available in November 2005 for a handful stocks and are excluded from our analyses due to data limitation.

price follows a generic jump diffusion process:

$$p_t = \int_0^t \mu_s ds + \int_0^t \sigma_s dW_s + J_t, \tag{1}$$

where $\mu$ and $\sigma$ denote the drift and diffusive volatility processes, respectively, $W$ is a standard Brownian motion, and $J$ is a pure jump process, and the unit time interval corresponds to a trading day. It is natural to extend the notation to intraday prices using the notation $p_t, p_{t+1/n}, ..., p_{t+1}$, assuming prices are observed at $n+1$ equally spaced time intervals from day $t$ to day $t+1$. The *annualized* daily $RV$ based on summing over frequently sampled squared returns within a trading day is then:

$$RV_t^d = 252 \times \sum_{i=1}^{n} r_{t-1+i/n}^2, \tag{2}$$

where $r_{t-1+i/n} = p_{t-1+i/n} - p_{t-1+(i-1)/n}$ is the log-return over the $i$th time interval on day $t$. In particular, we include the overnight squared returns in the daily $RV$ estimation to obtain an $RV$ measure for the entire day. As shown in Andersen, Bollerslev, Diebold, and Labys (2001, 2003), $RV$ is a consistent estimator for quadratic variation when the number of intervals $n \to \infty$. Longer horizon $RV$'s (e.g., weekly, monthly, and quarterly) can be estimated by averaging daily $RV$ over the corresponding intervals. Specifically, $RV_{t+1}^{t+h} = \frac{1}{h}\sum_{i=1}^{h} RV_{t+i}^d$ is the $h$-day ahead $RV$, where $h = 5,\ 21,\ 63$ corresponds to weekly, monthly and quarterly $RV$, respectively.

Our research objective is to build better predictive models for the responses of daily, weekly, monthly and quarterly $RV$s. To empirically compute $RV$, we use five-minute sampling frequency commonly employed in the realized volatility literature. To further increase the efficiency of $RV$ estimates, we apply a subsampling approach following Zhang, Mykland, and Aït-Sahalia (2005). Specifically, we compute five different daily $RV$ estimates by starting the trading day at 9:30, 9:31, 9:32, 9:33, and 9:34, respectively, and then average over these five estimates to obtain the final daily $RV$ estimate.

## 3. Features

Our machine learning predictive system consists of two components: features and learning algorithms. Generally speaking, our research design is to first construct input features potentially containing predictive information, and then fit learning algorithms to estimate functions that map features to the response variable, and finally evaluate the performance of our predictions. In this section, we discuss how we construct our feature sets. We consider three types of features: 1) features proposed by popular $RV$-based volatility forecasting models: HAR, SHAR, MIDAS, HARQ-F, and HExpGl; 2) features from option implied variances; and 3) derived features from the first two types of features. We start by reviewing several popular $RV$ forecasting models with a focus on the particular features (predictors) proposed by each model.

### *3.1. HAR*

The Heterogeneous Autoregressive (HAR) model proposed by Corsi (2009) is popular because it is easy to implement yet very effective in practice. The idea is to combine short- (daily), medium- (weekly) with long-term (monthly) volatility components in capturing various empirical properties in return series such as long memory and fat tails. The original HAR is used to forecast volatility up to monthly horizon. Since our longest forecast horizon is quarterly, we augment the HAR model with a quarterly $RV$ term:

$$RV_{t+1}^{t+h} = \beta_0 + \beta_d RV_t^d + \beta_w RV_t^w + \beta_m RV_t^m + \beta_q RV_t^q + \epsilon_t, \tag{3}$$

where $RV_t^w$, $RV_t^m$ and $RV_t^q$ denote the average annualized daily $RV$ over lags 1 to 5, lags 1 to 21, and lags 1 to 63 throughout the paper.

### *3.2. MIDAS*

The mixed data sampling (MIDAS) model of Ghysels, Santa-Clara, and Valkanov (2006) has the following form:

$$RV_{t+1}^{t+h} = \beta_0 + \beta_1 MIDAS_t^k + \epsilon_t, \tag{4}$$

in which the $MIDAS^k$ term is defined by:

$$\begin{aligned} MIDAS_t^k &= \frac{1}{\sum_{i=1}^{L} a_i}(a_1 RV_t^d + a_2 RV_{t-1}^d + ... + a_L RV_{t-L+1}^d), \\ a_i &= (\frac{i}{L})^{\theta_1 - 1}(1 - \frac{i}{L})^{\theta_2 - 1}\Gamma(\theta_1 + \theta_2)\Gamma(\theta_1)^{-1}\Gamma(\theta_2)^{-1}, i = 1, ..., L, \end{aligned} \tag{5}$$

where $\Gamma(\cdot)$ denotes the Gamma function; the superscript $k$ in $MIDAS^k$ can take values of $d, w, m, q$, representing the resulting $MIDAS$ term from predicting $h = 1,\ 5,\ 21,\ 63$-day ahead $RV$. The $MIDAS$ feature can be viewed as a smoothly weighted sum of lagged daily $RV$s. It has three hyperparameters $\theta_1$, $\theta_2$, and $L$ that need to be tuned. Directly mirroring Ghysels, Santa-Clara, and Valkanov (2006) and Bollerslev, Hood, Huss, and Pedersen (2018), we set $\theta_1 = 1$ and $L = 50$. Further guided by Bollerslev, Hood, Huss, and Pedersen (2018) and Ghysels and Qian (2019), we employ a grid search to tune $\theta_2$ for each $h$-day forecast horizon and choose the value that minimizes the Mean Squared Errors (MSE) over the full sample.[7]

### *3.3. SHAR*

We follow Patton and Sheppard (2015) to estimate a Semivariance-HAR (SHAR) model which decomposes daily $RV$ into two realized semivariance components:

$$RV_{t+1}^{t+h} = \beta_0 + \beta_d^+ RVP_t^d + \beta_d^- RVN_t^d + \beta_w RV_t^w + \beta_m RV_t^m + \beta_q RV_t^q + \epsilon_t, \tag{6}$$

where the annualized daily positive and negative semivariances, introduced by Barndorff-Nielsen, Kinnebrock, and Shephard (2010), are defined as:

$$RVP_t^d = 252 \times \sum_{i=1}^{n} r_{t-1+i/n}^2 1_{\{r_{t-1+i/n}>0\}},\ RVN_t^d = 252 \times \sum_{i=1}^{n} r_{t-1+i/n}^2 1_{\{r_{t-1+i/n}<0\}}. \tag{7}$$

Daily realized semivariances provide a natural decomposition of daily $RV$, i.e., $RV_t^d = RVP_t^d + RVN_t^d$. Patton and Sheppard (2015) show that the negative semivariance $RVN^d$ has stronger predictive power on future $RV$s. To mitigate bias in realized semivariance estimates, we also apply the subsampling scheme to construct $RVP^d$ and $RVN^d$.

[7] Due to computational burden, we follow the literature to not perform a rolling grid search for the $\theta_2$ parameter. As a result, the MIDAS feature is not truly out-of-sample but is included for comparison.

### *3.4. HARQ-F*

Bollerslev, Patton, and Quaedvlieg (2016b) propose a HARQ-F model by considering measurement errors in $RV$ estimates. The measurement error may be characterized by the asymptotic (for $n \to \infty$) distribution theory of Barndorff-Nielsen and Shephard (2002):

$$RV_t = IV_t^* + \epsilon_t, \;\; \epsilon_t \sim MN(0, 2\Delta IQ_t), \tag{8}$$

where $IV_t^* \equiv \int_{t-1}^{t} \sigma_s^2 ds$ is the unobservable Integrated Variance, $IQ_t \equiv \int_{t-1}^{t} \sigma_s^4 ds$ denotes the Integrated Quarticity ($IQ$), and MN stands for mixed normal. Using intraday returns, the integrated quarticity for *annualized* daily $RV$ may be consistently estimated by annualized daily realized quarticity ($RQ$):

$$RQ_t^d = 252^2 \times \frac{n}{3} \sum_{i=1}^{n} r_{t-1+i/n}^4. \tag{9}$$

To improve efficiency, we further apply the subsampling method to the daily $RQ$ estimation. Weekly, monthly and quarterly realized quarticities, denoted by $RQ^w$, $RQ^m$ and $RQ^q$ respectively, can be calculated by averaging daily $RQ$ over lags 1 to 5, lags 1 to 21, and lags 1 to 63. The HARQ-F model allows coefficients of lagged $RV$s to vary as a function of $\sqrt{RQ}$:

$$\begin{aligned} RV_{t+1}^{t+h} = \beta_0 + \beta_d RV_t^d + \beta_w RV_t^w + \beta_m RV_t^m + \beta_q RV_t^q \\ + \phi_d RV_t^d \sqrt{RQ_t^d} + \phi_w RV_t^w \sqrt{RQ_t^w} + \phi_m RV_t^m \sqrt{RQ_t^m} + \phi_q RV_t^q \sqrt{RQ_t^q} + \epsilon_t. \end{aligned} \tag{10}$$

Bollerslev, Patton, and Quaedvlieg (2016b) show that by allowing the model parameters to vary explicitly with the degree of measurement error, this model generates significant improvements in the accuracy of the forecasts compared to the forecasts from some of the most popular risk models.

### *3.5. HExpGl*

The Heterogeneous Exponential Realized Volatility with Global Risk Factor (HExpGl) model by Bollerslev, Hood, Huss, and Pedersen (2018) is one of the latest techniques for volatility forecasting. Similar to HAR and MIDAS, HExpGl also constructs features based on daily $RV$ series. The difference is that HExpGl uses exponentially weighted moving averages (EWMA) of lagged daily

$RV$s, whereas HAR uses step functions and MIDAS relies on more complicated functional forms. The EWMA of lagged daily $RV$'s with a pre-specified center-of-mass ($CoM$) is given by:

$$ExpRV_t^{CoM(\lambda)} = \sum_{i=1}^{500} \frac{e^{-i\lambda}}{e^{-\lambda} + e^{-2\lambda} + ... + e^{-500\lambda}} RV_{t-i+1}^d, \tag{11}$$

where $\lambda$ defines the decay rate of the weights and $CoM(\lambda)$ denotes the corresponding center-of-mass $CoM(\lambda) = e^{-\lambda}/(1 - e^{-\lambda})$; conversely, for a given center-of-mass, $\lambda$ can be inferred from $\lambda = log(1 + 1/CoM)$. The center-of-mass for a given $ExpRV$ measure captures the "average" horizon of the lagged $RV$s that it uses. We follow Bollerslev, Hood, Huss, and Pedersen (2018) to consider $ExpRV$ terms with center-of-mass equal to 1, 5, 25 and 125 trading days. Motivated by the cross-asset and cross-market volatility spillover effects, HExpGl also includes the EWMA of a global risk factor $GlRV$ with a center-of-mass equal to 5:

$$ExpGlRV_t^5 = \sum_{i=1}^{500} \frac{e^{-i\lambda}}{e^{-\lambda} + e^{-2\lambda} + ... + e^{-500\lambda}} GlRV_{t-i+1}, \tag{12}$$

where the corresponding $\lambda = log(1 + 1/CoM) = log(1 + 1/5)$. For each day $t$ and each stock $i$, the global risk factor $GlRV$ is computed as the average normalized $RV$ scaled back to the asset's own level of volatility, that is $(\frac{1}{N}\sum_{j=1}^{N} \frac{RV_{j,t}^d}{\overline{RV_j}})\overline{RV_i}$, where $\overline{RV_i}$ is the long-run mean of daily $RV$ for stock $i$ calculated from the beginning of the sample until day $t$. The resulting HExpGl model specification is given by:

$$\begin{aligned} RV_{t+1}^{t+h} = \beta_0 + \beta_1 ExpRV_t^1 + \beta_2 ExpRV_t^5 + \beta_3 ExpRV_t^{25} + \beta_4 ExpRV_t^{125} \\ + \beta_5 ExpGlRV_t^5 + \epsilon_t. \end{aligned} \tag{13}$$

### *3.6. Option-Implied Variances*

In addition to the high-frequency based realized features from existing models, our paper also considers option-implied variances as inputs. Since our forecasting horizon is up to three months, we include all 102 options from put and call options with maturities between one and three months across all deltas to avoid cherry picking a particular option in order to reduce the chance of overfitting. For call option implied variances, we denote these features as $CIV^{jm,\delta}$ with maturity equal to $j$ months ($j = 1, 2, 3$) and delta equal to $\delta$ ($\delta = 0.1, 0.15, ..., 0.9$). For put option implied variances,

we denote these features as $PIV^{jm,\delta}$ with maturity equal to $j$ months ($j = 1, 2, 3$) and delta equal to $\delta$ ($\delta = -0.1, -0.15, ..., -0.9$).

### *3.7. Descriptive Statistics of Features*

Table 1 provides the descriptive statistics of all realized features and selective implied variance features with absolute delta equal to 0.5.[8] Figure 3 plots the average implied variance of call and put options with maturities of one month (30 days), two months (60 days) and three months (91 days) from the entire panel of stocks in our sample as functions of deltas. From the summary statistics reported in Tables A.1 and A.2, the lowest average implied variance from call (put) options is the one with a maturity of three months and a delta of 0.2 (-0.7). Options with longer maturity are associated with lower implied variances.

Table 2 reports the pairwise correlation between features in Table 1. MIDAS features for different forecast horizons have the highest correlations of 0.96 or above with each other perhaps because they are calibrated by fitting highly correlated dependent variables to the same daily $RV$ terms. HARQ-F features (e.g., $RV^k\sqrt{RQ^k}$) have low correlations with other realized features, mostly because these features contain realized quarticities while other realized measures are all linear combinations of $RV$s. Moreover, $IV$-based features $CIV$s and $PIV$s have relatively low correlations with all $RV$-based features, suggesting potentially additive values of the $IV$-based features to $RV$-based features.

## 4. Machine Learning Methodology

This section reviews the five machine learning algorithms we considered in this paper. The first two are linear: Least Absolute Shrinkage and Selection Operator (LASSO) and Principal Component Regression (PCR). The next three are nonlinear: Random Forest (RF), Gradient Boosted Regression Trees (GBRT) and Neural Network (NN).

### *4.1. LASSO*

LASSO tries to improve over OLS by imposing sparsity-encouraging penalties on regression coefficients for variance reduction and model interpretation. Take daily $RV$ prediction as an

[8] Further details on implied variance features across deltas are in Tables A.1 and A.2 in the Appendix for call and put options, respectively.

example, LASSO assumes the same linear regression function as OLS:

$$g^*(z_{i,t};\theta) = z'_{i,t}\theta, \tag{14}$$

where $z'_{i,t}$ is the feature vector for stock $i$ on day $t$ and $\theta$ is the unknown parameter. However, unlike OLS, LASSO estimates $\theta$ through a penalized $L_1$ loss function:

$$\mathcal{L}(\theta;\lambda) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}(RV^d_{i,t+1} - g^*(z_{i,t};\theta))^2 + \lambda\sum_{j=1}^{P}|\theta_j|, \tag{15}$$

where $\lambda$ is the shrinkage parameter controlling the amount of penalty on the coefficients. The special case of $\lambda = 0$ collapses back to OLS. In such case, LASSO/OLS minimizes the training (in-sample) error, thus potentially overfits the data. By imposing the $L_1$ penalty $\lambda\sum_{j=1}^{P}|\theta_j|$, LASSO is capable of setting some of the coefficients to be exactly zero, a very desirable property for two reasons. First, setting coefficients to zero reduces parameter estimation variance and thus brings down the variance part of the prediction error. Second, with zero regression coefficients, the fitted model becomes more interpretable.

A few implementation details are important for achieving better performance of LASSO. First, we need to normalize features before estimating the models so that all features have comparable magnitudes. Otherwise, the single-valued $\lambda$ would have vastly different shrinkage impacts on different features, making it impossible to tune. The normalization is done by only using mean and standard deviation of the training sample to prevent look-ahead bias; we recalculate the mean and standard deviation once per year to be consistent with the expanding-rolling window scheme detailed in Section 4.7. Second, we need to choose $\lambda$ from a wide range of values that can generate coefficient estimates with varying sparsity levels. Otherwise, the estimated $\theta$ might be far from the optimal region in the parameter space.

### *4.2. Principal Component Regression*

The second linear learning algorithm we consider is PCR, which is motivated by the fact that our features for volatility forecasting are often correlated. PCR uses dimension reduction techniques to produce a small number of common factors from the original feature space and then relies on

the derived features as inputs for regressions. Specifically, in the first step, Principal Component Analysis (PCA) is performed on the $P$-dimensional feature space to extract a small number of factors as linear combinations of the original inputs; these factors are orthogonal to each other to prevent redundant information. In the second step, we only take the first $K$ most important principals that preserve the main variability of the original features for fitting the regression. More formally, PCR is defined as following:

$$RV = (Z\Omega_K)\theta_K + \tilde{E}, \tag{16}$$

where $RV$ is the $NT \times 1$ vector of realized variance, $Z$ is the $NT \times P$ matrix of features, $\Omega_K$ is a $P \times K$ orthogonal projection matrix from the $P$-dimensional original feature space onto the $K$-dimensional derived input space; $\theta_K$ is a vector of coefficients corresponding to $K$ derived inputs, and $\tilde{E}$ is an $NT \times 1$ vector of residuals. The projection matrix $\Omega_K$ can be found through singular value decomposition (SVD) of the original feature matrix $Z$.

The hyperparameters for PCR is the number of derived input features $K$. There is a trade-off between dimension reduction and information preservation when choosing $K$. If $K$ is large, more information in the original features are kept and used for prediction. However, overfitting concerns naturally arise as there are more parameters to estimate. If $K$ is small, there is a risk that the second stage regression model misses some useful information in the discarded principal components. In our implementation, we choose $K$ through validation. This gives the unsupervised learning PCA some guidance based on the target. We also standardize all features as in LASSO to ensure the principal components are not dominated by one single feature with extremely large variance. The number of components used in the linear regression is chosen by the smallest MSE on the validation sets. To increase computational speed and also prevent overfitting, we set an upper bound for $K$ equal to 20.

### *4.3. Random Forest*

Our first nonlinear learning algorithm is the random forest (RF) model, which is based on regression trees for modeling nonlinearity. Unlike linear methods reviewed in Sections 4.1 and 4.2 that essentially project the response onto the feature space, tree-based models partition the feature

space into a set of non-overlapping regions as illustrated in Figure 1. The observations within the same region are then fit through a simple model such as a constant. Mathematically, the estimated response function of a regression tree is:

$$\widehat{g^*}(z^*_{i,t};\theta,K,L)=\sum_{k=1}^{K}\theta_k 1_{\{z^*_{i,t}\in C_k(L)\}}, \tag{17}$$

where $C_k(L)$ is one of the $K$ regions pre-determined by the training set. $K$ is the number of regions, $L$ is the tree depth, $1_{\{\cdot\}}$ is an indicator function, and $\theta_k$ is the sample mean of the outcomes for training observations within that region. A very large tree with many regions can capture very fine details of the data but is prone to overfitting. Consider the extreme case where a fully grown tree divides every single training observation in the training set as one region, thus yielding zero training error but very poor out-of-sample performance.

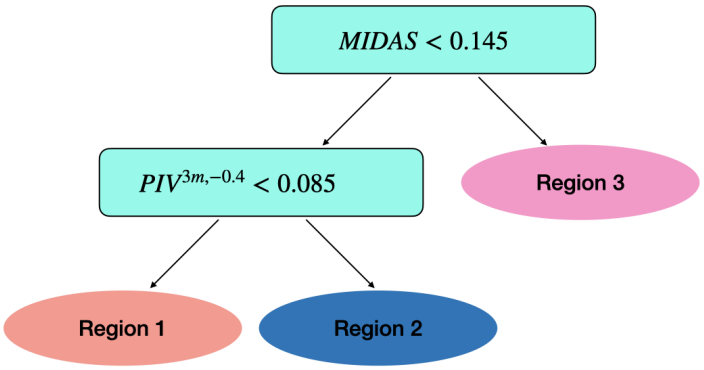



Fig. 1 Illustration of a regression tree model

Random forest reduces the overfitting problem of regression trees through several modifications. First, instead of a single tree, RF generates multiple trees by bootstrapping the training sample and then averaging forecasts from each individual tree to reduce the variance. Second, it implements observation and feature subsampling in the training process to decorrelate individual trees in the forest for further variance reduction.

How large should we grow the trees? As described earlier, deep trees are less biased but very unstable. Our strategy is to grow a large tree and then prune it back to a depth of $L$. We tune the tree depth $L$ via validation where we search for the optimal $L$ that minimizes the validation

error over a grid of values from 1 to 20. For each RF fitting, we bootstrap and average over 500 trees. For each tree, we use 50% of the training observations, and for each node split, we use $log(P)$ features. Tree-based models are insensitive to the scales of the features and thus do not require feature standardization.

### *4.4. Gradient Boosted Regression Trees*

The second nonlinear learning algorithm we investigate is the Gradient Boosted Regression Trees (GBRT), which is based on trees as RF. However, there are two major differences between GBRT and RF. First, GBRT uses trees as base learners in an *additive* fashion whereas RF uses trees in an *average* fashion. At each step, GBRT fits a new tree to explain what has been left unexplained by previous trees, while RF fits a parallel tree to explain the original response. Second, GBRT prefers to use shallow trees because each tree is supposed to be weak, and only by adding many small trees is GBRT expected to gradually achieve good performance. In contrast, RF prefers deep trees because these trees need to be unbiased, and only by averaging many deep trees is RF expected to reduce variance while simultaneously capturing the true relation.

To prevent overfitting, GBRT adds a new tree after shrinking its contribution. Specifically, at every round after fitting a tree to the residuals, we update our $\widehat{g^*}^m(\cdot)$ by adding a shrunk version of the new tree with a shrinkage multiplier $0 < \lambda < 1$, which is called the learning rate. We then update the residuals by subtracting this shrunk tree from the previously predicted values. Other approaches employed by RF to mitigate overfitting problems are also used for GBRT. Specifically, we adopt subsampling for each tree and randomly draw a subset of features at each split. The hyperparameters for GBRT are learning rate $\lambda$ which controls the speed of learning, the maximum tree depth that represents the upper bound for the degree of polynomials and interactions, and the number of trees which prevent overfitting and as a result can balance the in-sample performance and the out-of-sample prediction.

In our implementation, we set the learning rate $\lambda$ low at 0.001 to help prevent the model from overfitting the residuals. We validate the maximum tree depth, $L$, from 1 to 5. The grids with $L > 1$ are set to allow GBRT the ability to include high-order interactions and polynomials. In addition, we use early-stopping rules to help choose the number of trees in the GBRT model. If Mean Squared Errors (MSE) stop decreasing after 50 consecutive rounds, we set the number of trees

to be the round that the MSE stops improving instead of including more trees in our GBRT model. We report the resulting number of trees as model complexity. Note that similar to RF, GBRT is insensitive to the scales of the features. We again use 50% of the training observations for each tree and randomly draw $log(P)$ features at each split.

### *4.5. Feed-Forward Neural Network*

The third nonlinear model we rely on is the traditional feed-forward Neural Network (NN) which uses hidden layers and nonlinear transformations to capture complex nonlinear relations. As shown in Figure 2, the original inputs $X$ pass through one or more hidden layers, which transform these inputs into derived features $Z$. The output layer aggregates the derived features into the ultimate prediction. Transformations are called activation functions in neuron network and are the sources of nonlinearity.

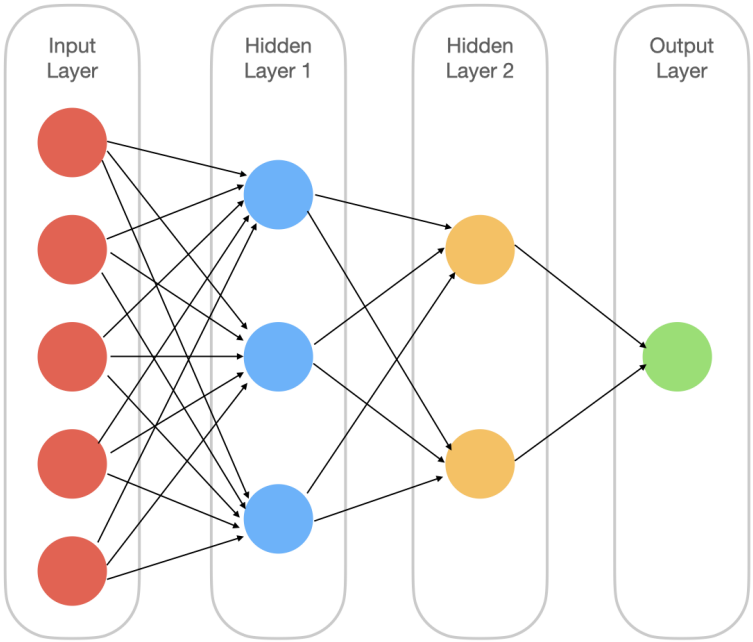



Fig. 2 Illustration of a feed-forward neural network model

In our implementation, we consider a one hidden layer model with 10 neurons. For the activation function, we choose the commonly used rectified linear unit (ReLU) given as:

$$ReLU(x) = max(x, 0). \tag{18}$$

We solve for the parameters in the activation function via stochastic gradient descent (SDG). We choose the adaptive moment estimation (Adam) by Kingma and Ba (2015) for computational efficiency and standardize each feature since neural network is sensitive to feature scales.

### *4.6. Ensemble Model*

In addition to the aforementioned stand-alone learning algorithms, we consider an ensemble model that combines forecasts from different models. The intuition is that no single model is expected to dominate the others under any circumstance. Different models might do well in different scenarios and by combining them together, we can make the forecast more robust. Here, we propose an equal-weighted average of all five machine learning methods as our ensemble forecast and call it AVG:

$$AVG = \frac{1}{5}\sum_{m=1}^{5} \widehat{g^*}^m(z^*_{i,t}). \tag{19}$$

### *4.7. Training and validation*

Machine learning algorithms have key hyperparameters controlling for model complexity. We should tune these parameters based on the *prediction* error rather than the *training* error. Otherwise, learning algorithms especially nonlinear ones will overfit the training data and do poorly out of sample. Accordingly, we adopt a training-validation-testing scheme for model selection and assessment. Specifically, at the end of each year $t$, we divide the sample into three parts: an expanding-window training set consisting of data from data inception (year 1996) to year $t-1$, a validation set consisting of data in year $t$, and a testing set consisting of data in year $t+1$. In other words, we refit our models every year by increasing the training set by one year, and rolling the validation and testing sets one year forward. Our first training sample contains four-year data from year 1996 to 1999 and our first validation sample contains data from year 2000. This scheme leaves us with a total of 19-year predictions between 2001 and 2019 corresponding to 19 fitted models for each learning algorithm. For models that do not require validation (e.g., OLS), we use data up to year $t$ for training and year $t+1$ for testing. Thus, the overall testing sets are the same across models and relative model performance cannot be driven by sample differences.

### *4.8. Performance Evaluation*

Since we focus on prediction rather than statistical inference, we use out-of-sample $R^2$ relative to a benchmark as our main performance measure:

$$R^2_{OOS}(m) = 1 - \frac{\sum_{i,t}(RV_{i.t} - \widehat{RV_{i,t}}^{m})^2}{\sum_{i,t}(RV_{i.t} - \widehat{RV_{i,t}}^{benchmark})^2}, \tag{20}$$

where $\widehat{RV_{i,t}}^{m}$ refers to forecasts from one of the OLS-based or machine learning-based volatility forecasting models, and $\widehat{RV_{i,t}}^{benchmark}$ is the forecast of a benchmark model.[9] A positive $R^2_{OOS}(m)$ indicates that model $m$ achieves smaller out-of-sample prediction mean squared errors than the benchmark model. We consider two benchmarks: one is the prediction from HAR, and the other is the long-run mean equal to the expanding sample mean of $RV$s from the inception date up until day $t$. The long-run mean is a commonly used benchmark and also mirrors the out-of-sample evaluation measure used in the return prediction literature. However, the bar of beating the long-run mean is low because volatilities are persistent and time-varying. HAR is perhaps a better benchmark because it has good volatility forecasting performance empirically, and is also easily implementable and interpretable.

In addition to $R^2_{OOS}(m)$, we also use a modified Diebold and Mariano (1995) (DM) test for pairwise comparison of two models. The DM test is based on the difference in the out-of-sample squared error losses between two forecasting models. More formally, for stock $i$ on day $t$, the loss differential is defined as $d_{i,t} = \left(\widehat{e}^{(1)}_{i,t}\right)^2 - \left(\widehat{e}^{(2)}_{i,t}\right)^2$, where $\widehat{e}^{(1)}_{i,t}$ and $\widehat{e}^{(2)}_{i,t}$ are the prediction error from two models. We then compute the cross-sectional mean of $d_{i,t}$ and denote it by $d_t$. The modified DM test statistic $DM = \overline{d}/\widehat{\sigma}_d$, where $\overline{d}$ and $\widehat{\sigma}_d$ are the mean and Newey-West standard error of $d_t$ over the testing sample.

### *4.9. Variable Importance Metrics*

To further shed lights on how these learning algorithms work for volatility forecasting, we investigate how different features contribute to the prediction at different horizons. Following Gu, Kelly, and Xiu (2020), we use the ranking approach. Let $VI_j$ denote the variable importance for the $j_{th}$

[9] Mirroring Swanson and White (1997) and Bollerslev, Hood, Huss, and Pedersen (2018), we apply an "insanity filter" to avoid deflation in $R^2_{OOS}$. Specifically, we replace any predictions that exceed the maximum outcome value in the training sample with the observed maximum.

variable. We first compute the reduction in $R^2_{OOS}$ relative to HAR from setting all values of variable $j$ to zero within each testing set, and then average the reductions over testing samples to obtain a single variable importance measure. We rank variables based on their $VI_j$ and the higher the rank, the more important a feature is. However, this ranking-based variable importance measure ignores the dependence of feature sets. Thus, $VI_j$ could still capture variable $j$'s predictive power and underestimate its importance. Despite obvious limitations, $VI_j$ is an efficient measure for interpreting models and visualizing feature contributions.

## 5. Out-of-Sample Performance of Forecasting Models

Based on the features in Section 3 and the learning algorithms in Section 4, we now show how machine learning can improve the volatility forecasting performance over traditional approaches. We begin with establishing the baseline performance by applying the traditional OLS method to *each* of the feature sets described in Section 3, as is commonly carried out in the literature. We then improve the performance of any stand-alone feature sets by combining *all* of them using OLS. We also show that implied variances contain unique information and can add additional value to the out-of-sample performance. Motivated by the "bet-on-sparsity" principle, we further provide evidence showing sparse linear models such as LASSO can improve over OLS. We next consider several nonlinear algorithms and demonstrate that they add value to linear methods. Finally, we show that an ensemble model delivers the best performance.

### *5.1. OLS-Based Models*

Table 3 reports the out-of-sample performance of OLS-based volatility forecasting models based on the $R^2_{OOS}$ relative to HAR from Eq. (20).[10] The first column lists the model names and the second column summarizes their features. First, we focus on the four popular $RV$-based models. Among them, MIDAS, SHAR, HARQ-F outperform HAR across all forecast horizons, as is evident by the positive relative $R^2_{OOS}$. HExpGl outperforms HAR at the daily, weekly and monthly horizons, and slightly underperforms HAR at the quarterly horizon.

Next, we combine all 16 realized measures from the MIDAS, SHAR, HARQ-F and HExpGl

[10] $R^2_{OOS}$'s relative to the long-run mean of $RV$s for OLS-based models are presented in Table A.3 in the Appendix.

models through OLS.[11] This model, namely $OLS^{RM}$, not only outperforms HAR by large margins across all horizons, but also generally beats individual models at different horizons. Only HARQ-F has a higher relative $R^2_{OOS}$ than $OLS^{RM}$ at the quarterly horizon. Overall, the superior performance of $OLS^{RM}$ illuminates the importance of feature combination in improving volatility forecasting performance.

We then fit OLS to the 102 implied variances ($IV$s) from call and put options with one-, two-, and three-month maturities and denote the model by $OLS^{IV}$. Unlike the realized features, these $IV$ features seem to underperform HAR as measured by the relative $R^2_{OOS}$. However, this does not mean that the $IV$ features are useless in the presence of realized features. Although as stand-alone features, $IV$s are weakly informative, they can still add value as long as they contain orthogonal information to the realized features. To test if there is any additional value of the $IV$ features, we expand the feature set in $OLS^{RM}$ by adding the 102 $IV$ features to the 16 realized features and call the model $OLS^{ALL}$. The row named $OLS^{ALL}$ reports its performance. As can be seen, $OLS^{ALL}$ has the highest relative $R^2_{OOS}$ for the first three forecast horizons among all OLS-based models in Table 3. However, at the quarterly horizon, the relative $R^2_{OOS}$ remains negative at $-0.6\%$ which is worse than several individual $RV$-based models. One possible reason is that, at the quarterly horizon, effective sample size drops significantly and thus we do not have enough data for estimating a dense OLS model with 118 features;[12] we may need sparse or more regularized models. Another point worth noting is that $OLS^{ALL}$ outperforms $OLS^{RM}$, suggesting the additional information contained in $IV$ measures, especially at the first three horizons.

### *5.2. Machine Learning-Based Models*

Having established the initial evidence that increasing the number of features can improve forecast performance but with the need for regularization, we now show how more sophisticated machine learning algorithms can further improve over the traditional OLS models. Table 4 presents the $R^2_{OOS}$'s relative to HAR for the five learning algorithms in Section 4: LASSO, Principal Component Regression (PCR), Random Forecast (RF), Gradient Boosted Regression Trees (GBRT) and Neural

[11] For a given forecast horizon, we include only one $MIDAS$ term corresponding to the same horizon. For instance, in predicting weekly $RV$, we keep the $MIDAS$ term constructed by using coefficients estimated from forecasting weekly $RV$ according to Eqs. (4) and (5).

[12] Our forecasting models try to use as much data as possible by fitting daily updated $RV$s on features ever day for all forecast horizons. However, due to overlapping data, the effective sample size of the data at the quarterly horizon is only about 1/63 of the daily horizon.

Network (NN), and for an ensemble model based on the five individual machine learning models (AVG).[13] Each model is trained using all 118 realized and implied variance features, so $OLS^{ALL}$ serves as a natural benchmark. The second column of Table 4 lists the hyperparameters of each model with tuning parameters in bold, and the last column reports the $R^2_{OOS}$'s relative to HAR. The most obvious pattern is that all machine learning models outperform HAR with positive relative $R^2_{OOS}$s across board. We then begin assessing the out-of-sample performance of each of the five machine learning models.

*Linear Machine learning models:* First, we focus on the two linear learning algorithms LASSO and PCR. The row labeled as LASSO and PCR in Table 4 presents their $R^2_{OOS}$'s relative to HAR.[14] The sparsity-encouraging $LASSO$ model has higher relative $R^2_{OOS}$ than the unregularized $OLS^{ALL}$ across all forecast horizons, indicating the importance of sparsity in enhancing the out-of-sample performance. The dimension-reduction PCR approach underperforms LASSO at daily, weekly and monthly forecast horizons, but exhibits better performance at the quarterly forecast horizon with a relative $R^2_{OOS}$ of 7.8%.

*Nonlinear Machine learning models:* Next, we turn our attention to the three nonlinear learning algorithms: RF, GBRT and NN. To train RF, we set the total number of trees to be 500 and use a subsample of 50% of the observations randomly drawn from each training sample (i.e., subsample = 0.5). At each node split, we randomly select 5 out of the 118 features (i.e., subfeature = $log(118) = 5$). Subsample and subfeature can help decorrelate the trees to reduce overfitting. The maximum tree depth across all trees $L$ is a tuning parameter, which can take any integer value between 1 and 20. The relative $R^2_{OOS}$ of RF from Table 4 is at 3.2% for daily forecast horizon and at 6.4% for weekly forecast horizon, both of which are below the corresponding metrics of $OLS^{ALL}$. However, RF outperforms $OLS^{ALL}$ at monthly and quarterly forecast horizons with relative $R^2_{OOS}$ at 9.5% and 5.4%, respectively. To train GBRT, we impose two early-stopping rules (whichever is met first): 1) when the MSE of the model does not decrease after 50 consecutive iterations, and 2) when the total number of trees reaches 20,000. Both the number of trees $B$ and the maximum tree depth are tuning parameters that we adaptively choose in the validation step; the maximum tree depth can

[13] $R^2_{OOS}$'s relative to the long-run mean of $RV$s for machine learning-based models are presented in Table A.4 in the Appendix.

[14] For LASSO, we validate its shrinkage parameter $\lambda$ from a set of 100 distinct values that covers a wide range of sparsity levels in the corresponding LASSO estimates of regression coefficients. For PCR, we validate the number of principal components as any integer between 1 and 20.

take any integer value between 1 and 5. For the rest of the hyperparameters, we set the learning rate to be 0.001; to grow each tree, we randomly draw 50% of the observations from the training sample; at each node split, we randomly select 5 out of the 118 features (i.e., subfeature = $log(118) = 5$). Overall, GBRT underperforms $OLS^{ALL}$ at daily and weekly forecast horizons with relative $R^2_{OOS}$ equal to 4.7% and 10.2%, but significantly outperforms $OLS^{ALL}$ at monthly and quarterly horizons with relative $R^2_{OOS}$ equal to 10.8% and 6.3%. To train NN, we consider one hidden layer and 10 neurons. We choose the popular rectified linear unit (ReLU) as the activation function. In general, NN performs reasonably well with relative $R^2_{OOS}$ equal to 8.4%, 10.4%, 8.1%, and 3.7% at daily, weekly, monthly, and quarterly forecast horizons, respectively.

*An ensemble model:* Comparing the out-of-sample performance of the five learning algorithms, we find no single model strictly dominates the others. We then consider an ensemble model that combines volatility forecasts from different models. We take a simple average of the five volatility forecast models and name the model as AVG. The motivation is that averaging forecasts from different models can improve the robustness of the model and reduce forecast variance. The out-of-sample performance of AVG shown at the bottom of Table 4 is indeed striking. This average model outperforms all five individual machine learning models at each forecast horizon by a significant margin. Its $R^2_{OOS}$ relative to HAR ranges from 9.3% at daily forecast horizon to up to 15% at monthly forecast horizon, further highlighting the advantage of combining machine learning models in forecasting $RV$s.

### *5.3. Model Complexity*

To help gain insights about model complexity, Figure 4 displays the chosen tuning parameters of LASSO, PCR, RF, and GBRT for each forecast horizon and validation period. For LASSO, Panel A shows the number of selected features with nonzero coefficients ranges from four (at monthly and quarterly forecast horizons by the end of 2001) to 45 (at daily forecast horizon by the end of 2008). Interestingly, as the forecast horizon increases, LASSO tends to select fewer features. For PCR, Panel B shows that the number of selected principal components also varies across forecast horizons and over time. Similar to LASSO, we see that PCR tends to favor more components in shorter-horizon forecast (daily and weekly) than longer horizons (monthly and quarterly).

For RF, Panel C of Figure 4 displays the maximum tree depth across 500 trees over time. The

average maximum tree depth is around 13 across forecast horizons and validation periods. For GBRT, Panel D plots the number of trees $B$ over time. The average number of trees is around 5,000 across forecast horizons and validation periods. The relatively large number of trees is due to our choice of a small learning rate $\lambda$ at 0.001, which requires a large value of $B$ for GBRT to converge. Note that imposing a boundary to $B$ is recommended in the literature (e.g., Zhang and Yu, 2005) because extremely large $B$ can also lead to overfitting. Our choice of 20,000 seems appropriate because this boundary is hit only once (e.g., by the end of 2006 at quarterly forecast horizon), and the overall out-of-sample performance of GBRT from Table 4 is comparable to other machine learning models.

### *5.4. Model Comparison*

The early results in Table 4 reveal that different machine learning models have different strengths over different horizons, and the simple average of all individual machine learning models performs the best across horizons. To better understand how the out-of-sample forecasts from various models are related to each other, we report the pairwise correlation of forecasts between the five machine learning models and $OLS^{ALL}$ in Table 5. The correlations are high, ranging from 0.909 between RF and $OLS^{ALL}$ at quarterly forecast horizon, to 0.997 between LASSO and $OLS^{ALL}$ at daily forecast horizon. This is not surprising since all models presented here employ the same set of features and the same set of responses. In addition, volatilities are very persistent and hence all these predictive models have a high signal-to-noise ratio.[15] Looking across forecast horizons, we see the pairwise correlations between forecasts tend to decrease as the forecast horizon increases. For instance, the correlation between AVG and $OLS^{ALL}$ monotonically decreases from 0.987 at the daily forecast horizon to 0.964 at the quarterly forecast horizon.

Given the high correlation of forecasts from different models, we are interested in formally assessing whether the differences in the out-of-sample performance among models are statistically significant at all. Table 6 reports the Diebold-Mariano (DM) $t$-statistics for pairwise comparisons of a model in the row versus a model in the column. The DM statistics are distributed $N(0,1)$ under the null hypotheses of equal predictive power between models, and thus the magnitude of the test statistics map to $p$-values in the same fashion as regression $t$-statistics; a positive $t$-statistic indicates

[15] For example, when computed relative to the long-run mean of $RV$ instead of HAR, the $R^2_{OOS}$'s of the models in Tables 3 and 4 are all above 53% across different forecast horizons, as shown in Tables A.3 and A.4 in the Appendix.

that the row model outperforms the column model; *, **, and *** denote significance at 10%, 5%, and 1% levels, respectively. At shorter daily and weekly forecast horizons, we find the majority of the $t$-statistics are significant at 5% level, indicating that high correlation between forecasts at these two horizons does not necessarily translate into insignificant difference in out-of-sample performance. At monthly forecast horizon, however, the $t$-statistics comparing the out-of-sample performance among $OLS^{ALL}$ and the five individual machine learning algorithms are generally insignificant. Similarly, at quarterly horizon, the five individual machine learning models generate statistically insignificant forecasting performance from each other at 5% level, yet PCF, RF and GBRT outperform $OLS^{ALL}$ at 1% or 5% level. Mirroring the results in Table 4, the relative strength of each stand-alone machine learning model and $OLS^{ALL}$ depends on the forecast horizon. In sharp contrast, the simple average of all machine learning models AVG outperforms the rest of the models across all forecast horizons at 1% or 5% significance level in most cases.

### *5.5. Subsample*

So far, we have established the superior out-of-sample performance of the learning algorithms over the full sample period. How does the relative performance of each model change over time? Could the 2008 financial crisis disrupt the relation between features and future $RV$s fitted using historical data? Which model performs the best in the most recent decade? To answer these questions, we further divide the 2001–2019 testing sample into three subperiods (2001-2007, 2008-2009, and 2010-2019), and calculate the $R^2_{OOS}$'s relative to HAR for both OLS-based and ML-based models. Table 7 summarizes the out-of-sample performance of all models over the three subperiods.

Panel A reports the results for the pre-crisis period between January 2001 and December 2007. Among OLS-based models, $OLS^{ALL}$ performs the best at daily, weekly and monthly forecast horizons with relative $R^2_{OOS}$ between 5.2% and 8.5%; at quarterly forecast horizon, HARQ-F has the highest relative $R^2_{OOS}$ at 6.7%. Turning to the ML-based models, AVG has the greatest performance across all forecast horizons, with relative $R^2_{OOS}$ ranging from 6.7% to 20.9%. In particular, its relative $R^2_{OOS}$'s at monthly and quarterly horizons are almost twice the magnitude of the second best model.

Panel B shows the out-of-sample performance of all models between January 2008 and December 2009, the period covering the financial crisis and its aftermath. $OLS^{ALL}$ continues to outperform

the rest of the OLS-based models at daily and weekly forecast horizons, whereas MIDAS dominates at monthly horizon and HARQ-F beats other OLS-based models at quarterly horizon. Among ML-based models, NN performs the best at daily forecast horizon with relative $R^2_{OOS}$ equal to 13.2%, and LASSO dominates the other models at both weekly and monthly horizons with relative $R^2_{OOS}$ at 14.9% and 9.5%, respectively. At the longer quarterly horizon, PCR has the highest relative $R^2_{OOS}$ equal to 5.7%. However, the overall winning model is still the ensemble model AVG. Although at a given forecast horizon, AVG cannot beat all of the stand-alone models, it consistently delivers top performance across all horizons.

Panel C presents the relative $R^2_{OOS}$'s for each model during the post-crisis period between January 2010 and June 2019. Interestingly, the performance of $OLS^{ALL}$ in this period is quite impressive with relative $R^2_{OOS}$ equal to 7.5%, 13.5%, 20.2%, and 15.5% for daily, weekly, monthly and quarterly horizons, respectively. In contrast, the relative $R^2_{OOS}$ of the popular $RV$ forecasting models are all below 6%. A natural question is, where does the stellar performance of $OLS^{ALL}$ come from? We conjecture that much of the gain comes from the better quality of the implied variance features in recent years. For example, the average daily dollar trading volume of stock options has steadily increased over the past two decades, implying that the overall option market is becoming more efficient in incorporating information about future price movements. As more direct evidence, the relative $R^2_{OOS}$'s of $OLS^{IV}$ during the post-crisis period all become positive, in sharp contrast to the mostly negative values in the pre-crisis and crisis periods. This trajectory further illuminates the prominence of including implied variance predictors in forecasting $RV$s. Meanwhile, the ML-based models exhibit even more extraordinary predictive power across forecast horizons than $OLS^{ALL}$ in the last decade. In particular, the ensemble model AVG is associated with relative $R^2_{OOS}$'s of 10.4%, 18.8%, 29.6%, and 27.5% at daily, weekly, monthly, and quarterly horizons, all dominating $OLS^{ALL}$ by significant margins. Taken together, the subsample results further highlight the importance of using machine learning techniques to exploit the rich information content in the giant feature set.

## 6. Which Features Matter?

We rely on the variable importance measure described in Section 4.9 to identify the most important features in forecasting future $RV$s while simultaneously controlling for the rest of the features. To keep the discussion concise, we focus on the average machine learning model AVG.[16] For each forecast horizon, we estimate the reduction in relative $R^2_{OOS}$ from setting all values of a given feature to zero within each testing sample, and then average the reductions over all testing samples to obtain a single variable importance measure.

Figure 5 displays the top-20 most influential features ranked by the variable importance measure for each forecast horizon. Following Gu, Kelly, and Xiu (2020), variable importance for a given forecast horizon is normalized to sum to one for easy interpretation of the relative importance of each feature. Figure 6 reports the rankings of 118 features in terms of overall contribution for AVG across different forecast horizons. All features are ordered based on the sum of their importance rankings over all forecast horizons; the columns correspond to each forecast horizon, and the color gradients within each column indicate the most influential (dark blue) to the least influential (white) features.

Figure 5 reveals several interesting findings. First and foremost, realized features tend to be more important in forecasting shorter daily and weekly $RV$s, whereas implied variance ($IV$) features are more crucial in predicting $RV$s at longer monthly and quarterly forecast horizons. Specifically, the top-seven features in forecasting daily $RV$s from Panel A and the top-three features in forecasting weekly $RV$s from Panel B are all realized measures; on the contrary, the top-seven features in predicting monthly $RV$s from Panel C and the top-four features in predicting quarterly $RV$s from Panel D are all $IV$ features. Since $IV$s included in our feature set all have maturities between one and three months, it is not surprising that they can better predict longer-term $RV$s. Secondly, $IV$s from put options tend to be more important than those from call options in forecasting future $RV$s. An, Ang, Bali, and Cakici (2014) document that stocks with large increases in $IV$s from put (call) options in the previous month tend to have negative (positive) future returns.[17] Another well-documented effect in the volatility literature is the asymmetric relationship between return and

[16] Variable importance results for the five individual machine learning models LASSO, PCR, RF, GBRT, and NN are shown in Figures A.1 to A.5 in the Appendix.

[17] An, Ang, Bali, and Cakici (2014) explain their findings based on an informed trading story. Basically, an bearish (bullish) informed trader, betting that a stock will decrease (increase) in value, can buy a put (call), which increases put (call) option volatilities this period, and hence the stock price drops (increases) the following period.

volatility, i.e., extremely negative equity returns are often associated with heightened volatilities.[18] Through both channels, $IV$s from put options tend to better predict future $RV$s than $IV$s from call options. Third, variable importance is more evenly distributed across the top-20 features at longer forecast horizons but skews to the top few features at shorter forecast horizons. Given our variable importance measures for each forecast horizon are normalized to sum to one, we can interpret the importance measure of each feature as its relative contribution to the overall importance of the top-20 features in percentage. At daily forecast horizon in Panel A, the most important feature $RVN^d$ contributes around 26% to the overall importance, and the second most important feature $MIDAS$ contributes around 23%; at weekly forecast horizon in Panel B, the most important feature $MIDAS$ alone contributes 28.4%. On the other hand, the most influential features $PIV^{3m,-0.3}$ at monthly forecast horizon in Panel C and $PIV^{3m,-0.3}$ at quarterly horizon in Panel D only contribute 8.4% and 10.3% respectively to the overall importance, while the least influential features $PIV^{3m,-0.15}$ and $PIV^{1m,-0.75}$ from Panels C and D each contribute around 2%. This pattern suggests that including more features is particularly important in forecasting longer-horizon $RV$s.

Figure 6 further displays the rankings of all 118 volatility predictors based on the sum of their importance rankings over forecast horizons. The results are generally inline with Figure 5. The top-four most important features across horizons are all $IV$ features, namely $PIV^{3m,-0.4}$, $CIV^{3m,0.4}$, $PIV^{3m,-0.1}$ and $PIV^{2m,-0.1}$. The only two realized measures ranked among the top ten are $RV^d\sqrt{RQ^d}$ and $RV^w\sqrt{RQ^w}$. Features such as $RVN^d$ and $MIDAS$ are among the most influential variables in forecasting daily and weekly volatilities; they do not receive high overall rankings because of their relatively low rankings in forecasting longer-horizon $RV$s.

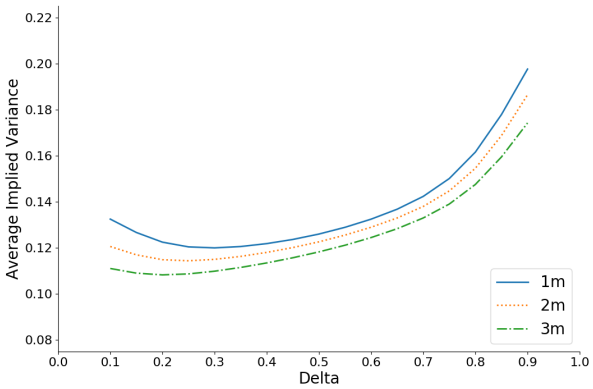
## 7. Conclusion

This paper proposes an automatic system for forecasting realized volatility. The system consists of two components: feature engineering and learning algorithm fitting. The feature engineering component includes many features that potentially contain predictive information of future volatility. The learning component consists of linear and nonlinear learning algorithms for estimating the predictive relation between volatility and features. This system requires little human intervention in

[18]The asymmetric relationship between return and volatility can be explained by leverage and volatility feedback effects. See Bollerslev, Litvinova, and Tauchen (2006) and the references therein.

choosing predictors and models. Using 118 features and five machine learning algorithms to forecast realized volatility for 173 stocks spanning two decades, we show that an automatic system can deliver robust and superior performance across forecasting horizons and time periods. Among all features, high-frequency based realized features from existing risk models are particularly important in predicting realized volatility at shorter daily and weekly horizons, whereas implied variance features from put and call options are more crucial in forecasting realized volatility at longer monthly and quarterly horizons.

Panel A: Implied variance from call options




Panel B: Implied variance from put options

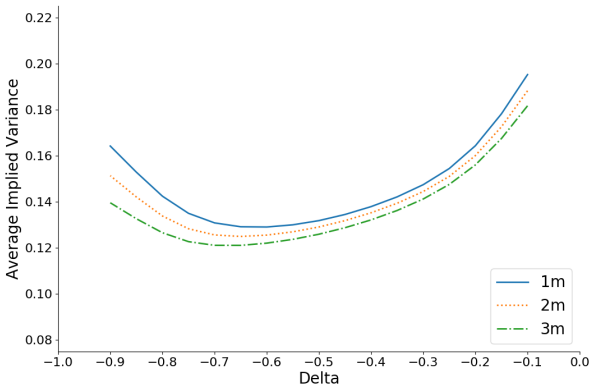



Fig. 3 Average implied variance from call and put options

This figure plots the average implied variance from call and put options for the entire panel of stocks in our sample as functions of delta. Panel A (B) displays the average implied variance from call (put) options with maturity equal to one month (30 days), two months (60 days), and three months (91 days). Delta ranges from 0.1 (-0.9) to 0.9 (-0.1) for implied variance from call (put) options with 0.05 increment.

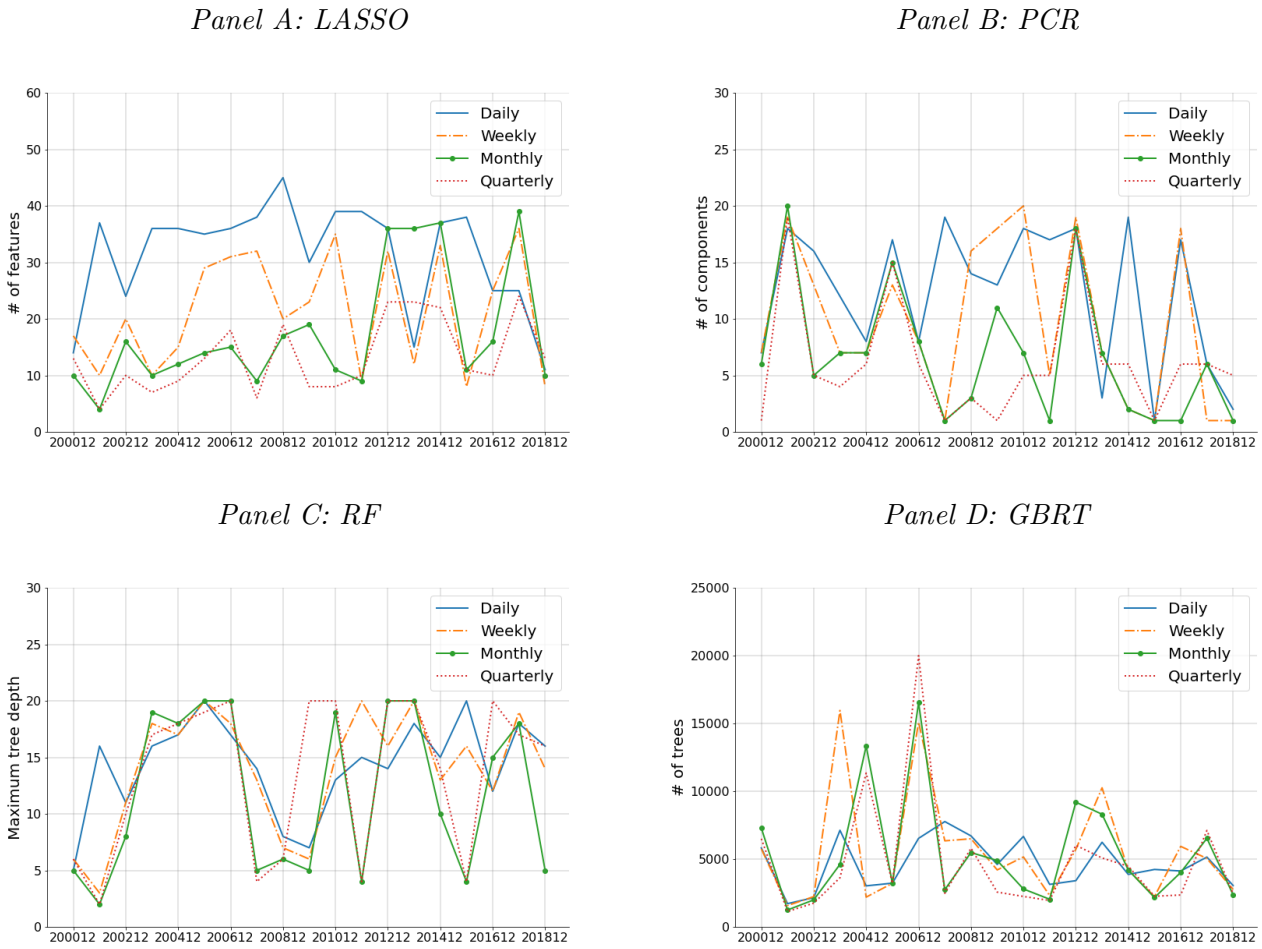



Fig. 4 Model complexity over time

This figure displays the complexity of LASSO, Principal Component Regression (PCR), Random Forest (RF), and Gradient Boosted Regression Trees (GBRT) validated using each training and validation sample in our out-of-sample analyses for different forecast horizons. Our first training sample is from January 1996 to December 1999 and our first validation sample is from January 2000 to December 2000; our last training sample is from January 1996 to December 2017 and our last validation sample is from January 2018 to December 2018. By the end of each validation sample, we report the number of selected features with nonzero coefficients for LASSO, the number of principal components for PCR, the maximum tree depth for RF, and the total number of trees for GBRT.

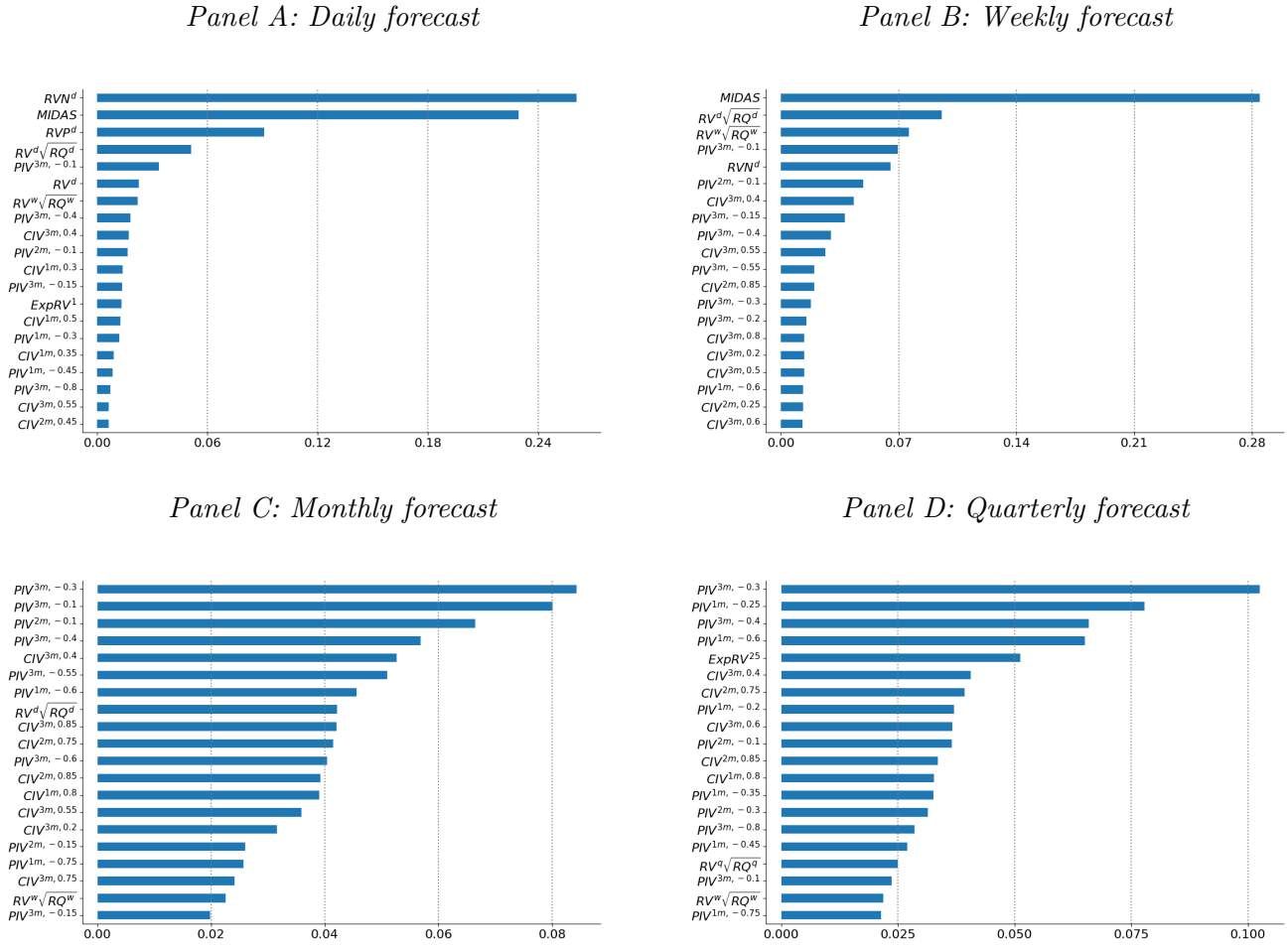



Fig. 5 Variable importance for AVG: top-20 features

This figure displays the top 20 most influential variables among all 118 features for AVG (i.e., the simple average of five individual machine learning models) across different forecast horizons. To calculate variable importance for each variable, we first compute the reduction in $R^2_{OOS}$ relative to HAR from setting all values of a given variable to zero within each testing sample, and then average the reductions over all testing samples to obtain a single variable importance measure.


Fig. 6 Variable importance for AVG: all 118 features

This figure displays the rankings of all 118 features in terms of overall contribution for AVG (i.e., the simple average of five individual machine learning models) across different forecast horizons. Variable importance is defined in Figure 5. All variables are ordered based on the sum of their importance rankings over all forecast horizons. The columns correspond to each forecast horizon, and the color gradients within each column indicate the most influential (dark blue) to the least influential (white) variables.

Table 1 Descriptive statistics

This table reports the descriptive statistics of all realized features and selective implied variance features with absolute delta equal to 0.5. Statistics for implied variances with absolute delta ranging from 0.1 to 0.9 are presented in Tables A.1 and A.2 in the Appendix. The sample consists of 173 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 100 index between January 1996 and June 2019 with share code 10 or 11, prices between \$1 and \$1000, daily number of trades greater than or equal to 100, and at least five years of data on all features and response variables. Superscriptions $d$, $w$, $m$, and $q$ are abbreviations for daily, weekly, monthly, and quarterly construction interval or forecast horizon. $MIDAS^k$ $(k = d, w, m, q)$ denotes the smoothly weighted moving average of 50 lagged realized variances using validated polynomials from Eqs. (4) and (5) in forecasting realized variance at horizon $k$. $RV^k$ $(k = d, w, m, q)$ is the daily, weekly, monthly or quarterly realized variance. $RVP^d$ and $RVN^d$ are the daily realized positive and negative semivariances, respectively. $RV^k\sqrt{RQ^k}$ $(k = d, w, m, q)$ is the product of the realized variance and the square root of the realized quarticity with the same construction interval $k$. $ExpRV^i$ $(i = 1, 5, 25, 125)$ is the exponentially weighted moving average of the past 500-day realized variances using the corresponding center-of-mass $i$ from Eq. (11). $ExpGlRV$ is the exponentially weighted moving average of the global risk factor with a 5-day center-of-mass from Eq. (12). $CIV^{jm,0.5}$ and $PIV^{jm,-0.5}$ are implied variances from call and put options with absolute delta equal to 0.5 and maturity equal to $j$ months $(j = 1, 2, 3)$.

| | Mean | Std | Skewness | Kurtosis | Min | P5 | Median | P95 | Max | AR(1) | AR(5) | AR(21) | AR(63) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $MIDAS^d$ | 0.145 | 0.243 | 7.575 | 106.664 | 0.000 | 0.019 | 0.076 | 0.478 | 9.918 | 0.969 | 0.839 | 0.629 | 0.457 |
| $MIDAS^w$ | 0.145 | 0.236 | 7.428 | 102.063 | 0.001 | 0.020 | 0.078 | 0.471 | 9.218 | 0.985 | 0.905 | 0.688 | 0.489 |
| $MIDAS^m$ | 0.145 | 0.233 | 7.344 | 99.184 | 0.001 | 0.020 | 0.079 | 0.468 | 8.728 | 0.991 | 0.933 | 0.725 | 0.508 |
| $MIDAS^q$ | 0.145 | 0.228 | 7.217 | 94.686 | 0.001 | 0.021 | 0.081 | 0.464 | 7.995 | 0.995 | 0.960 | 0.780 | 0.534 |
| $RV^d$ | 0.144 | 0.299 | 9.249 | 152.262 | 0.000 | 0.014 | 0.065 | 0.507 | 13.071 | 0.581 | 0.466 | 0.366 | 0.280 |
| $RV^w$ | 0.148 | 0.265 | 7.899 | 115.693 | 0.001 | 0.017 | 0.073 | 0.509 | 10.743 | 0.945 | 0.656 | 0.508 | 0.382 |
| $RV^m$ | 0.150 | 0.247 | 8.076 | 119.655 | 0.001 | 0.021 | 0.081 | 0.487 | 9.197 | 0.993 | 0.945 | 0.682 | 0.482 |
| $RV^q$ | 0.151 | 0.235 | 7.504 | 97.368 | 0.002 | 0.024 | 0.087 | 0.471 | 6.838 | 0.999 | 0.989 | 0.910 | 0.612 |
| $RVP^d$ | 0.072 | 0.158 | 11.268 | 251.878 | 0.000 | 0.006 | 0.031 | 0.255 | 9.462 | 0.513 | 0.414 | 0.324 | 0.248 |
| $RVN^d$ | 0.070 | 0.155 | 10.070 | 189.623 | 0.000 | 0.006 | 0.030 | 0.252 | 8.155 | 0.495 | 0.400 | 0.317 | 0.238 |
| $RV^d\sqrt{RQ^d}$ | 0.257 | 3.111 | 45.794 | 3351.632 | 0.000 | 0.000 | 0.007 | 0.504 | 448.391 | 0.259 | 0.169 | 0.116 | 0.079 |
| $RV^w\sqrt{RQ^w}$ | 0.281 | 2.272 | 25.820 | 1024.559 | 0.000 | 0.001 | 0.012 | 0.733 | 205.110 | 0.853 | 0.281 | 0.180 | 0.116 |
| $RV^m\sqrt{RQ^m}$ | 0.285 | 2.023 | 29.907 | 1495.195 | 0.000 | 0.001 | 0.020 | 0.964 | 257.709 | 0.973 | 0.837 | 0.315 | 0.176 |
| $RV^q\sqrt{RQ^q}$ | 0.287 | 1.820 | 29.610 | 1455.167 | 0.000 | 0.002 | 0.031 | 1.026 | 149.275 | 0.994 | 0.962 | 0.783 | 0.291 |
| $ExpRV^1$ | 0.148 | 0.274 | 8.341 | 128.445 | 0.000 | 0.017 | 0.072 | 0.508 | 11.939 | 0.875 | 0.625 | 0.477 | 0.357 |
| $ExpRV^5$ | 0.149 | 0.254 | 8.064 | 120.020 | 0.002 | 0.020 | 0.079 | 0.496 | 9.601 | 0.976 | 0.863 | 0.626 | 0.445 |
| $ExpRV^{25}$ | 0.151 | 0.235 | 7.447 | 97.015 | 0.004 | 0.024 | 0.087 | 0.476 | 7.584 | 0.997 | 0.978 | 0.869 | 0.624 |
| $ExpRV^{125}$ | 0.154 | 0.200 | 5.232 | 42.188 | 0.012 | 0.029 | 0.097 | 0.464 | 3.324 | 1.000 | 0.997 | 0.978 | 0.890 |
| $ExpGlRV$ | 0.178 | 0.294 | 6.849 | 81.002 | 0.008 | 0.030 | 0.094 | 0.603 | 8.923 | 0.993 | 0.942 | 0.743 | 0.520 |
| $CIV^{1m,0.5}$ | 0.126 | 0.167 | 5.913 | 63.823 | 0.000 | 0.022 | 0.077 | 0.384 | 6.516 | 0.972 | 0.921 | 0.793 | 0.635 |
| $CIV^{2m,0.5}$ | 0.123 | 0.158 | 5.953 | 65.884 | 0.000 | 0.023 | 0.076 | 0.367 | 4.989 | 0.982 | 0.946 | 0.840 | 0.670 |
| $CIV^{3m,0.5}$ | 0.118 | 0.146 | 5.643 | 57.782 | 0.000 | 0.023 | 0.075 | 0.348 | 3.574 | 0.988 | 0.959 | 0.868 | 0.700 |
| $PIV^{1m,-0.5}$ | 0.132 | 0.200 | 11.795 | 305.546 | 0.001 | 0.023 | 0.079 | 0.395 | 8.652 | 0.977 | 0.930 | 0.803 | 0.642 |
| $PIV^{2m,-0.5}$ | 0.129 | 0.191 | 12.798 | 359.667 | 0.001 | 0.025 | 0.080 | 0.377 | 8.652 | 0.985 | 0.953 | 0.852 | 0.681 |
| $PIV^{3m,-0.5}$ | 0.126 | 0.181 | 13.983 | 431.430 | 0.002 | 0.027 | 0.080 | 0.359 | 8.642 | 0.990 | 0.965 | 0.880 | 0.714 |

## Table 2 Feature correlation

This table reports the correlations of all realized features and selective implied variance features with absolute delta equal to 0.5. The sample consists of 173 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 100 index between January 1996 and June 2019 with share code 10 or 11, prices between $1 and $1000, daily number of trades greater than or equal to 100, and at least five years of data on all features and response variables. Superscriptions $d$, $w$, $m$, and $q$ are abbreviations for daily, weekly, monthly, and quarterly construction interval or forecast horizon. $MIDAS^k$ $(k = d, w, m, q)$ denotes the smoothly weighted moving average of 50 lagged realized variances using validated polynomials from Eqs. (4) and (5) in forecasting realized variance at horizon $k$. $RV^k$ $(k = d, w, m, q)$ is the daily, weekly, monthly or quarterly realized variance. $RVP^d$ and $RVN^d$ are the daily realized positive and negative semivariances, respectively. $RV^k\sqrt{RQ^k}$ $(k = d, w, m, q)$ is the product of the realized variance and the square root of the realized quarticity with the same construction interval $k$. $ExpRV^i$ $(i = 1, 5, 25, 125)$ is the exponentially weighted moving average of the past 500-day realized variances using the corresponding center-of-mass $i$ from Eq. (11). $ExpGlRV$ is the exponentially weighted moving average of the global risk factor with a 5-day center-of-mass from Eq. (12). $CIV^{jm,0.5}$ and $PIV^{jm,-0.5}$ are implied variances from call and put options with absolute delta equal to 0.5 and maturity equal to $j$ months $(j = 1, 2, 3)$.

| Variable | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) | (11) | (12) | (13) | (14) | (15) | (16) | (17) | (18) | (19) | (20) | (21) | (22) | (23) | (24) | (25) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (1) $MIDAS^d$ | 1.00 | | | | | | | | | | | | | | | | | | | | | | | | |
| (2) $MIDAS^w$ | 0.99 | 1.00 | | | | | | | | | | | | | | | | | | | | | | | |
| (3) $MIDAS^m$ | 0.98 | 1.00 | 1.00 | | | | | | | | | | | | | | | | | | | | | | |
| (4) $MIDAS^q$ | 0.96 | 0.99 | 1.00 | 1.00 | | | | | | | | | | | | | | | | | | | | | |
| (5) $RV^d$ | 0.86 | 0.82 | 0.80 | 0.77 | 1.00 | | | | | | | | | | | | | | | | | | | | |
| (6) $RV^w$ | 0.97 | 0.95 | 0.94 | 0.91 | 0.81 | 1.00 | | | | | | | | | | | | | | | | | | | |
| (7) $RV^m$ | 0.92 | 0.95 | 0.96 | 0.98 | 0.73 | 0.88 | 1.00 | | | | | | | | | | | | | | | | | | |
| (8) $RV^q$ | 0.82 | 0.86 | 0.88 | 0.90 | 0.65 | 0.77 | 0.89 | 1.00 | | | | | | | | | | | | | | | | | |
| (9) $RVP^d$ | 0.80 | 0.77 | 0.75 | 0.73 | 0.90 | 0.76 | 0.68 | 0.62 | 1.00 | | | | | | | | | | | | | | | | |
| (10) $RVN^d$ | 0.78 | 0.75 | 0.73 | 0.71 | 0.90 | 0.74 | 0.67 | 0.61 | 0.65 | 1.00 | | | | | | | | | | | | | | | |
| (11) $RV^d\sqrt{RQ^d}$ | 0.49 | 0.44 | 0.42 | 0.39 | 0.72 | 0.46 | 0.37 | 0.30 | 0.64 | 0.61 | 1.00 | | | | | | | | | | | | | | |
| (12) $RV^w\sqrt{RQ^w}$ | 0.70 | 0.67 | 0.65 | 0.62 | 0.59 | 0.78 | 0.61 | 0.48 | 0.54 | 0.52 | 0.54 | 1.00 | | | | | | | | | | | | | |
| (13) $RV^m\sqrt{RQ^m}$ | 0.67 | 0.70 | 0.71 | 0.71 | 0.53 | 0.66 | 0.79 | 0.63 | 0.49 | 0.47 | 0.39 | 0.71 | 1.00 | | | | | | | | | | | | |
| (14) $RV^q\sqrt{RQ^q}$ | 0.59 | 0.61 | 0.63 | 0.65 | 0.46 | 0.56 | 0.68 | 0.80 | 0.43 | 0.42 | 0.29 | 0.50 | 0.71 | 1.00 | | | | | | | | | | | |
| (15) $ExpRV^1$ | 0.96 | 0.93 | 0.91 | 0.88 | 0.93 | 0.95 | 0.84 | 0.74 | 0.85 | 0.84 | 0.59 | 0.72 | 0.63 | 0.53 | 1.00 | | | | | | | | | | |
| (16) $ExpRV^5$ | 0.98 | 0.98 | 0.98 | 0.97 | 0.82 | 0.96 | 0.95 | 0.84 | 0.76 | 0.75 | 0.45 | 0.72 | 0.74 | 0.62 | 0.94 | 1.00 | | | | | | | | | |
| (17) $ExpRV^{25}$ | 0.90 | 0.93 | 0.94 | 0.96 | 0.71 | 0.85 | 0.96 | 0.97 | 0.67 | 0.66 | 0.35 | 0.57 | 0.73 | 0.76 | 0.82 | 0.92 | 1.00 | | | | | | | | |
| (18) $ExpRV^{125}$ | 0.76 | 0.79 | 0.80 | 0.83 | 0.60 | 0.71 | 0.80 | 0.90 | 0.57 | 0.56 | 0.26 | 0.40 | 0.50 | 0.62 | 0.68 | 0.76 | 0.89 | 1.00 | | | | | | | |
| (19) $ExpGlRV$ | 0.62 | 0.63 | 0.64 | 0.64 | 0.50 | 0.58 | 0.60 | 0.57 | 0.48 | 0.47 | 0.20 | 0.28 | 0.30 | 0.29 | 0.56 | 0.60 | 0.60 | 0.59 | 1.00 | | | | | | |
| (20) $CIV^{1m,0.5}$ | 0.83 | 0.85 | 0.85 | 0.86 | 0.69 | 0.79 | 0.84 | 0.82 | 0.63 | 0.65 | 0.32 | 0.50 | 0.58 | 0.58 | 0.77 | 0.84 | 0.85 | 0.78 | 0.59 | 1.00 | | | | | |
| (21) $CIV^{2m,0.5}$ | 0.82 | 0.84 | 0.85 | 0.86 | 0.67 | 0.78 | 0.83 | 0.83 | 0.62 | 0.63 | 0.31 | 0.49 | 0.58 | 0.58 | 0.76 | 0.83 | 0.86 | 0.79 | 0.59 | 0.98 | 1.00 | | | | |
| (22) $CIV^{3m,0.5}$ | 0.82 | 0.84 | 0.85 | 0.86 | 0.67 | 0.77 | 0.84 | 0.84 | 0.62 | 0.62 | 0.31 | 0.48 | 0.58 | 0.59 | 0.75 | 0.82 | 0.87 | 0.81 | 0.60 | 0.97 | 0.99 | 1.00 | | | |
| (23) $PIV^{1m,-0.5}$ | 0.78 | 0.80 | 0.80 | 0.81 | 0.64 | 0.75 | 0.80 | 0.79 | 0.59 | 0.60 | 0.32 | 0.53 | 0.63 | 0.63 | 0.73 | 0.80 | 0.82 | 0.73 | 0.52 | 0.90 | 0.89 | 0.88 | 1.00 | | |
| (24) $PIV^{2m,-0.5}$ | 0.77 | 0.78 | 0.79 | 0.80 | 0.62 | 0.73 | 0.79 | 0.79 | 0.58 | 0.58 | 0.31 | 0.52 | 0.62 | 0.64 | 0.71 | 0.78 | 0.82 | 0.74 | 0.51 | 0.88 | 0.88 | 0.88 | 0.99 | 1.00 | |
| (25) $PIV^{3m,-0.5}$ | 0.75 | 0.77 | 0.78 | 0.79 | 0.61 | 0.72 | 0.78 | 0.79 | 0.57 | 0.56 | 0.31 | 0.50 | 0.61 | 0.64 | 0.69 | 0.77 | 0.81 | 0.74 | 0.50 | 0.85 | 0.86 | 0.87 | 0.98 | 0.99 | 1.00 |

Table 3 Out-of-sample prediction relative to HAR: OLS-based models

This table reports the out-of-sample $R^2$ relative to the HAR model for OLS-based volatility forecasting models across different forecast horizons. The sample consists of 173 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 100 index between January 1996 and June 2019 with share code 10 or 11, prices between \$1 and \$1000, daily number of trades greater than or equal to 100, and at least five years of data on all features and response variables. The full out-of-sample evaluation period is from January 2001 to June 2019. Superscriptions $d$, $w$, $m$, and $q$ are abbreviations for daily, weekly, monthly, and quarterly construction interval or forecast horizon. $MIDAS$ denotes the smoothly weighted moving average of 50 lagged realized variances using validated polynomials from Eqs. (4) and (5) for the corresponding forecast horizon. $RV^k$ ($k = d, w, m, q$) is the daily, weekly, monthly or quarterly realized variance. $RVP^d$ and $RVN^d$ are the daily realized positive and negative semivariances, respectively. $RV^k\sqrt{RQ^k}$ ($k = d, w, m, q$) is the product of the realized variance and the square root of the realized quarticity with the same construction interval $k$. $ExpRV^i$ ($i = 1, 5, 25, 125$) is the exponentially weighted moving average of the past 500-day realized variances using the corresponding center-of-mass $i$ from Eq. (11). $ExpGlRV$ is the exponentially weighted moving average of the global risk factor with a 5-day center-of-mass from Eq. (12). $CIV^{jm,\delta}$ and $PIV^{jm,-\delta}$ are implied variances from call and put options with absolute $\delta = 0.1, 0.15, ..., 0.9$ and maturity equal to $j$ months ($j = 1, 2, 3$). Our OLS-based models include MIDAS, SHAR, HARQ-F, HExpGl, $OLS^{RM}$ (i.e., simple OLS model with all 16 realized features as predictors), $OLS^{IV}$ (i.e., simple OLS model with all 102 implied variance features as predictors), and $OLS^{ALL}$ (i.e., simple OLS model with all 118 realized and implied variance features as joint predictors). $R^2_{OOS}$ for each model is calculated relative to the prediction from HAR using the entire panel of stocks according to Eq. (20).

| Model | Features | Daily | Weekly | Monthly | Quarterly |
|---|---|---|---|---|---|
| | | $R^2_{OOS}$ relative to HAR | | | |
| MIDAS | $MIDAS$ term for the corresponding forecast horizon | 1.1% | 3.8% | 4.4% | 1.5% |
| SHAR | $RVP^d$, $RVN^d$, $RV^w$, $RV^m$, $RV^q$ | 1.5% | 1.6% | 1.3% | 0.6% |
| HARQ-F | $RV^d$, $RV^w$, $RV^m$, $RV^q$, $RV^d\sqrt{RQ^d}$, $RV^w\sqrt{RQ^w}$, $RV^m\sqrt{RQ^m}$, $RV^q\sqrt{RQ^q}$ | 2.1% | 2.8% | 3.4% | 4.8% |
| HExpGl | $ExpRV^1$, $ExpRV^5$, $ExpRV^{25}$, $ExpRV^{125}$, $ExpGlRV$ | 0.1% | 2.6% | 2.2% | -1.4% |
| $OLS^{RM}$ | $MIDAS$ term for the corresponding forecast horizon, $RV^d$, $RV^w$, $RV^m$, $RV^q$, $RVP^d$, $RVN^d$, $RV^d\sqrt{RQ^d}$, $RV^w\sqrt{RQ^w}$, $RV^m\sqrt{RQ^m}$, $RV^q\sqrt{RQ^q}$, $ExpRV^1$, $ExpRV^5$, $ExpRV^{25}$, $ExpRV^{125}$, $ExpGlRV$ (# of features = 16) | 4.9% | 6.5% | 5.4% | 1.9% |
| $OLS^{IV}$ | $CIV^{jm,\delta}$ and $PIV^{jm,-\delta}$, $j = 1, 2, 3$, $\delta = 0.1, 0.15, ..., 0.9$ (# of features = 102) | -9.8% | -7.4% | -2.8% | -2.1% |
| $OLS^{ALL}$ | All 118 Features (16 realized features + 102 $IV$ features) | 7.6% | 11.6% | 7.3% | -0.6% |

Table 4 Out-of-sample predictions relative to HAR: Machine learning-based models

This table reports the out-of-sample $R^2$ relative to the HAR model for machine learning-based volatility forecasting models across different forecast horizons.The sample consists of 173 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 100 index between January 1996 and June 2019 with share code 10 or 11, prices between \$1 and \$1000, daily number of trades greater than or equal to 100, and at least five years of data on all features and response variables. The full out-of-sample evaluation period is from January 2001 to June 2019. The features of each model consist of all 118 predictors detailed in Table 3. Our machine learning based models include LASSO, Principal Component Regression (PCR), Random Forest (RF), Gradient Boosted Regression Trees (GBRT), Neural Network (NN), and a simple average of forecasts from the five individual machine learning models (AVG). Tuning parameters for each model are in **bold**. $R^2_{OOS}$ for each model is calculated relative to the prediction from HAR using the entire panel of stocks according to Eq. (20).

| Model | Hyperparameter (Tuning parameter in **bold**) | Daily | Weekly | Monthly | Quarterly |
|---|---|---|---|---|---|
| | | $R^2_{OOS}$ relative to HAR | | | |
| LASSO | **# of shrinkage parameters ($\lambda$): 100**<br>$\lambda_{min}/\lambda_{max}$: 0.001 | 8.0% | 12.1% | 11.3% | 2.6% |
| PCR | **# of components: 1, 2, ..., 20** | 5.5% | 4.8% | 8.1% | 7.8% |
| RF | **Maximum tree depth ($L$): 1, 2, ..., 20**<br># of trees: 500<br>Subsample: 0.5<br>Subfeature: log(# of features) | 3.2% | 6.4% | 9.5% | 5.4% |
| GBRT | **# of trees ($B$)**<br>**Maximum tree depth ($L$): 1, 2, ..., 5**<br>Learning rate: 0.001<br>Subsample: 0.5<br>Subfeature: log(# of features)<br>Early-stopping rules (whichever met first):<br>1) No reduction in MSE after 50 iterations<br>2) Max # of trees hit 20,000 | 4.7% | 10.2% | 10.8% | 6.3% |
| NN | # of hidden layer: 1<br># of neurons: 10<br>Activation function: ReLU | 8.4% | 10.4% | 8.1% | 3.7% |
| AVG | | 9.3% | 14.0% | 15.0% | 10.4% |

Table 5 Forecast correlation

This table reports the correlation of volatility forecasts from different models for the entire panel of stocks across different forecast horizons. The sample consists of 173 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 100 index between January 1996 and June 2019 with share code 10 or 11, prices between $1 and $1000, daily number of trades greater than or equal to 100, and at least five years of data on all features and response variables. The full out-of-sample evaluation period is from January 2001 to June 2019. The features of each model consist of all 118 predictors detailed in Table 3. Our models include a simple OLS model using all features ($OLS^{ALL}$), LASSO, Principal Component Regression (PCR), Random Forest (RF), Gradient Boosted Regression Trees (GBRT), Neural Network (NN), and a simple average of forecasts from the five individual machine learning models (AVG).

*Panel A: Daily forecast*

| | $OLS^{ALL}$ | LASSO | PCR | RF | GBRT | NN | AVG |
|---|---|---|---|---|---|---|---|
| $OLS^{ALL}$ | 1.000 | | | | | | |
| LASSO | 0.997 | 1.000 | | | | | |
| PCR | 0.985 | 0.990 | 1.000 | | | | |
| RF | 0.953 | 0.958 | 0.953 | 1.000 | | | |
| GBRT | 0.963 | 0.967 | 0.967 | 0.982 | 1.000 | | |
| NN | 0.973 | 0.974 | 0.971 | 0.953 | 0.965 | 1.000 | |
| AVG | 0.987 | 0.991 | 0.989 | 0.981 | 0.989 | 0.986 | 1.000 |

*Panel B: Weekly forecast*

| | $OLS^{ALL}$ | LASSO | PCR | RF | GBRT | NN | AVG |
|---|---|---|---|---|---|---|---|
| $OLS^{ALL}$ | 1.000 | | | | | | |
| LASSO | 0.991 | 1.000 | | | | | |
| PCR | 0.968 | 0.974 | 1.000 | | | | |
| RF | 0.947 | 0.957 | 0.955 | 1.000 | | | |
| GBRT | 0.963 | 0.973 | 0.968 | 0.986 | 1.000 | | |
| NN | 0.970 | 0.966 | 0.957 | 0.962 | 0.965 | 1.000 | |
| AVG | 0.982 | 0.988 | 0.984 | 0.985 | 0.992 | 0.984 | 1.000 |

*Panel C: Monthly forecast*

| | $OLS^{ALL}$ | LASSO | PCR | RF | GBRT | NN | AVG |
|---|---|---|---|---|---|---|---|
| $OLS^{ALL}$ | 1.000 | | | | | | |
| LASSO | 0.981 | 1.000 | | | | | |
| PCR | 0.963 | 0.977 | 1.000 | | | | |
| RF | 0.938 | 0.957 | 0.959 | 1.000 | | | |
| GBRT | 0.952 | 0.972 | 0.971 | 0.989 | 1.000 | | |
| NN | 0.972 | 0.968 | 0.958 | 0.947 | 0.959 | 1.000 | |
| AVG | 0.975 | 0.988 | 0.987 | 0.983 | 0.992 | 0.980 | 1.000 |

*Panel D: Quarterly forecast*

| | $OLS^{ALL}$ | LASSO | PCR | RF | GBRT | NN | AVG |
|---|---|---|---|---|---|---|---|
| $OLS^{ALL}$ | 1.000 | | | | | | |
| LASSO | 0.975 | 1.000 | | | | | |
| PCR | 0.954 | 0.971 | 1.000 | | | | |
| RF | 0.909 | 0.932 | 0.941 | 1.000 | | | |
| GBRT | 0.936 | 0.959 | 0.961 | 0.984 | 1.000 | | |
| NN | 0.957 | 0.955 | 0.952 | 0.939 | 0.954 | 1.000 | |
| AVG | 0.964 | 0.981 | 0.984 | 0.976 | 0.989 | 0.978 | 1.000 |

Table 6 Forecast comparison using Diebold-Mariano tests

This table reports pairwise Diebold-Mariano $t$-statistics comparing the out-of-sample forecast performance among seven models across different forecast horizons. The sample consists of 173 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 100 index between January 1996 and June 2019 with share code 10 or 11, prices between \$1 and \$1000, daily number of trades greater than or equal to 100, and at least five years of data on all features and response variables. The full out-of-sample evaluation period is from January 2001 to June 2019. The features of each model consist of all 118 predictors detailed in Table 3. Our models include a simple OLS model using all features ($OLS^{ALL}$), LASSO, Principal Component Regression (PCR), Random Forest (RF), Gradient Boosted Regression Trees (GBRT), Neural Network (NN), and a simple average of forecasts from the five individual machine learning models (AVG). Positive numbers indicate the model in the row outperforms the model in the column. *, **, and *** indicate significance at the 10%, 5%, and 1% levels, respectively.

*Panel A: Daily forecast*

| | $OLS^{ALL}$ | LASSO | PCR | RF | GBRT | NN |
|---|---|---|---|---|---|---|
| LASSO | 3.30*** | | | | | |
| PCR | -7.02*** | -8.87*** | | | | |
| RF | -4.62*** | -5.06*** | -2.32** | | | |
| GBRT | -4.93*** | -5.88*** | -1.20 | 2.27** | | |
| NN | 1.70* | 1.05 | 6.10*** | 4.49*** | 5.02*** | |
| AVG | 6.08*** | 5.60*** | 12.91*** | 7.67*** | 11.93*** | 1.52 |

*Panel B: Weekly forecast*

| | $OLS^{ALL}$ | LASSO | PCR | RF | GBRT | NN |
|---|---|---|---|---|---|---|
| LASSO | 1.05 | | | | | |
| PCR | -3.22*** | -3.36*** | | | | |
| RF | -2.22** | -2.42** | 1.15 | | | |
| GBRT | -0.90 | -1.48 | 3.07*** | 2.38** | | |
| NN | -0.83 | -1.08 | 3.12*** | 2.39** | 0.04 | |
| AVG | 2.21** | 1.71* | 6.33*** | 5.14*** | 4.34*** | 3.37*** |

*Panel C: Monthly forecast*

| | $OLS^{ALL}$ | LASSO | PCR | RF | GBRT | NN |
|---|---|---|---|---|---|---|
| LASSO | 1.62* | | | | | |
| PCR | 0.19 | -1.12 | | | | |
| RF | 0.64 | -0.53 | 0.48 | | | |
| GBRT | 1.12 | -0.18 | 1.13 | 0.94 | | |
| NN | 0.35 | -1.04 | 0.03 | -0.52 | -1.15 | |
| AVG | 2.56** | 1.84* | 3.26*** | 2.73*** | 4.31*** | 2.89*** |

*Panel D: Quarterly forecast*

| | $OLS^{ALL}$ | LASSO | PCR | RF | GBRT | NN |
|---|---|---|---|---|---|---|
| LASSO | 1.38 | | | | | |
| PCR | 2.71*** | 1.66* | | | | |
| RF | 1.92** | 0.89 | -0.68 | | | |
| GBRT | 1.95** | 1.30 | -0.50 | 0.46 | | |
| NN | 1.31 | 0.35 | -1.41 | -0.78 | -1.69* | |
| AVG | 3.13*** | 3.16*** | 0.97 | 2.00** | 3.12*** | 3.96*** |

Table 7 Out-of-sample prediction relative to HAR: Subsample analysis

This table reports the out-of-sample $R^2$ relative to the HAR model for OLS-based and machine learning based volatility forecasting models across different forecast horizons over three subsample periods. The sample consists of 173 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 100 index between January 1996 and June 2019 with share code 10 or 11, prices between \$1 and \$1000, daily number of trades greater than or equal to 100, and at least five years of data on all features and response variables. The full out-of-sample evaluation period is from January 2001 to June 2019. The features of each model consist of all 118 predictors detailed in Table 3. Our OLS-based models include MIDAS, SHAR, HARQ-F, HExpGl, $OLS^{RM}$ (i.e., simple OLS model with all 16 realized features as predictors), $OLS^{IV}$ (i.e., simple OLS model with all 102 implied variance features as predictors), and $OLS^{ALL}$ (i.e., simple OLS model with all 118 realized and implied variance features as joint predictors). Our machine learning based models include LASSO, Principal Component Regression (PCR), Random Forest (RF), Gradient Boosted Regression Trees (GBRT), Neural Network (NN), and a simple average of forecasts from the five individual machine learning models (AVG). $R^2_{OOS}$ for each model is calculated relative to the prediction from HAR using the panel of stocks in each subsample period according to Eq. (20). Panels A, B and C report $R^2_{OOS}$ relative to HAR for the pre-crisis (2001-2007), crisis (2008-2009) and post-crisis (2010-2019) periods, respectively.

| | | *Panel A: Pre-crisis (2001-2007)* | | | | *Panel B: Crisis (2008-2009)* | | | | *Panel C: Post-crisis (2010-2019)* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Daily | Weekly | Monthly | Quarterly | Daily | Weekly | Monthly | Quarterly | Daily | Weekly | Monthly | Quarterly |
| | | $R^2_{OOS}$ relative to HAR | | | | | | | | | | | |
| OLS | MIDAS | -0.4% | 1.0% | -2.4% | -0.9% | 3.9% | 7.1% | 8.3% | 2.1% | 0.4% | 3.1% | 5.1% | 4.3% |
| | SHAR | 1.1% | 1.3% | 1.1% | 0.6% | 2.1% | 2.1% | 1.5% | 0.6% | 1.5% | 1.2% | 1.1% | 0.8% |
| | HARQ-F | 1.9% | 3.0% | 3.9% | 6.7% | 3.4% | 3.6% | 3.2% | 4.0% | 1.1% | 1.4% | 3.0% | 5.3% |
| | HExpGl | 0.2% | 2.5% | 3.5% | 3.1% | -0.3% | 3.6% | 1.3% | -4.2% | 0.3% | 1.1% | 2.6% | 6.0% |
| | $OLS^{RM}$ | 4.0% | 5.8% | 4.3% | 2.3% | 6.6% | 7.2% | 4.8% | 0.6% | 4.4% | 6.3% | 10.2% | 10.3% |
| | $OLS^{IV}$ | -12.5% | -13.0% | -1.5% | 2.9% | -15.4% | -11.9% | -8.4% | -5.6% | 0.4% | 8.4% | 15.5% | 9.1% |
| | $OLS^{ALL}$ | 5.2% | 8.5% | 5.6% | -1.6% | 11.2% | 13.5% | 4.8% | -2.5% | 7.5% | 13.5% | 20.2% | 15.1% |
| ML | LASSO | 5.7% | 8.9% | 9.3% | 4.4% | 11.8% | 14.9% | 9.5% | -1.0% | 7.4% | 12.8% | 22.3% | 23.0% |
| | PCR | 2.6% | 7.0% | 8.7% | 8.1% | 10.1% | -2.2% | 4.5% | 5.7% | 5.3% | 12.1% | 20.0% | 21.9% |
| | RF | 0.6% | 1.9% | 9.9% | 7.9% | 0.0% | 2.0% | 4.0% | 1.6% | 10.8% | 20.2% | 29.2% | 25.3% |
| | GBRT | -0.5% | 3.6% | 9.2% | 10.9% | 7.6% | 11.7% | 7.0% | 1.9% | 9.8% | 18.1% | 28.4% | 24.3% |
| | NN | 3.8% | 6.7% | 8.5% | 11.4% | 13.2% | 9.1% | 3.2% | -0.7% | 10.3% | 18.0% | 25.1% | 13.7% |
| | AVG | 6.7% | 13.5% | 19.4% | 20.9% | 11.9% | 11.3% | 8.4% | 3.9% | 10.4% | 18.8% | 29.6% | 27.5% |

# Appendix

### *A.1. High-Frequency Data Cleaning*

We begin by removing entries that satisfy at least one of the following criteria: a price less than or equal to zero; a trade size less than or equal to zero; corrected trades, i.e., trades with Correction Indicator, CORR, other than 0, 1, or 2; and an abnormal sale condition, i.e., trades for which the Sale Condition, COND, has a letter code other than @, *, E, F, @E, @F, *E, and *F. We then assign a single value to each variable for each second. If one or multiple transactions have occurred in that second, we calculate the sum of volumes, the sum of trades, and the volume-weighted average price within that second. If no transaction has occurred in that second, we enter zero for volume and trades. For the volume-weighted average price, we use the entry from the nearest previous second. Motivated by our analysis of the trading volume distribution across different exchanges over time, we purposely incorporate information from all exchanges covered by the TAQ database.

### *A.2. Additional Results*

Figures A.1 to A.5 report the rankings of 118 features in terms of overall contribution for each individual machine learning model across different forecast horizons. Tables A.1 and A.2 provide descriptive statistics of implied variance features across deltas from call and put options, respectively. Tables A.3 and A.4 present the out-of-sample performance of OLS-based and machine learning-based forecasting models using $R^2_{OOS}$ relative to the long-run mean of $RV$.

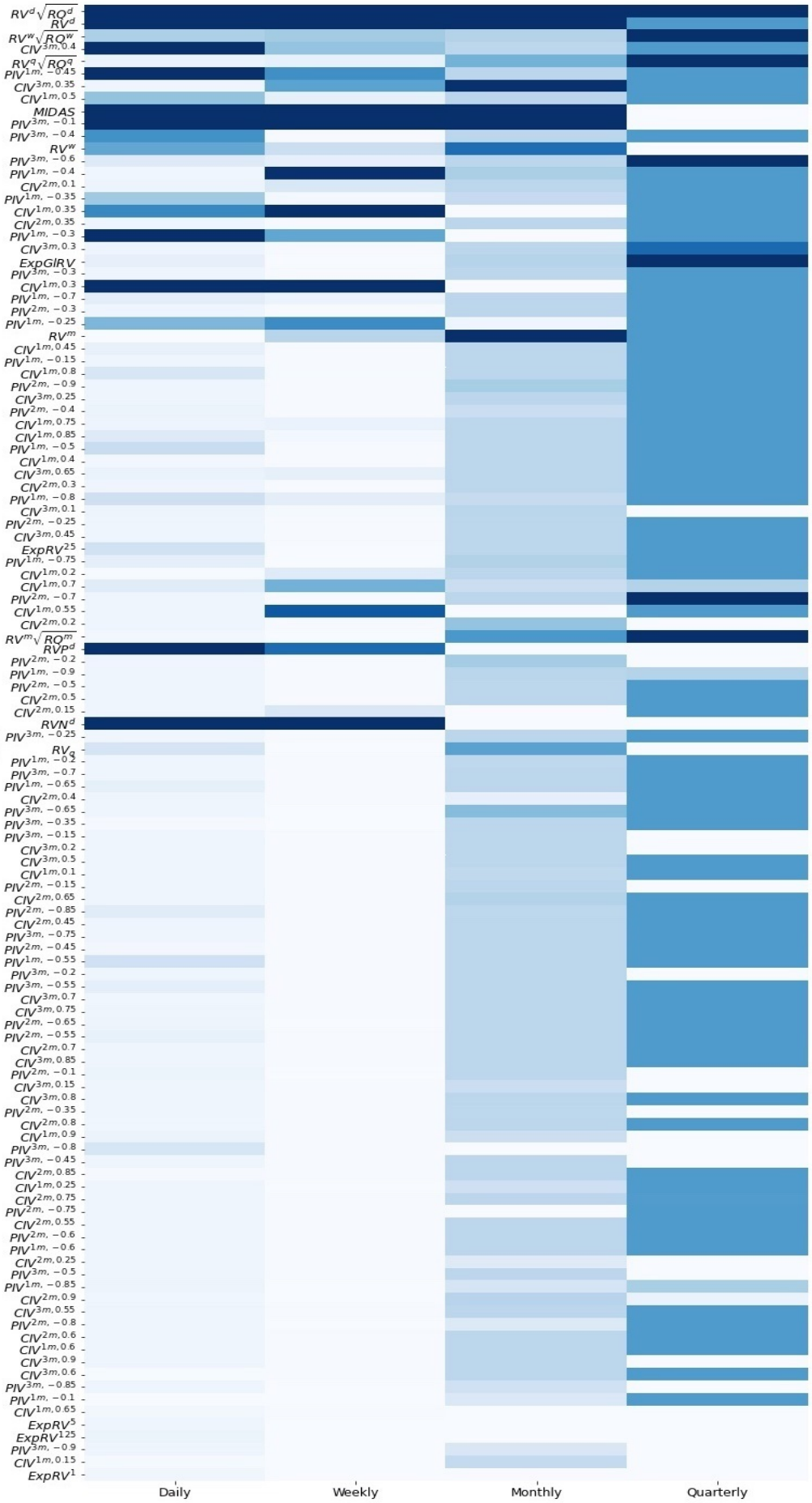





Fig. A.1 Variable importance for LASSO: all 118 features

This figure displays the rankings of all 118 features in terms of overall contribution for LASSO across different forecast horizons. Variable importance is defined in Figure 5. All variables are ordered based on the sum of their importance rankings over all forecast horizons. The columns correspond to each forecast horizon, and the color gradients within each column indicate the most influential (dark blue) to the least influential (white) variables.

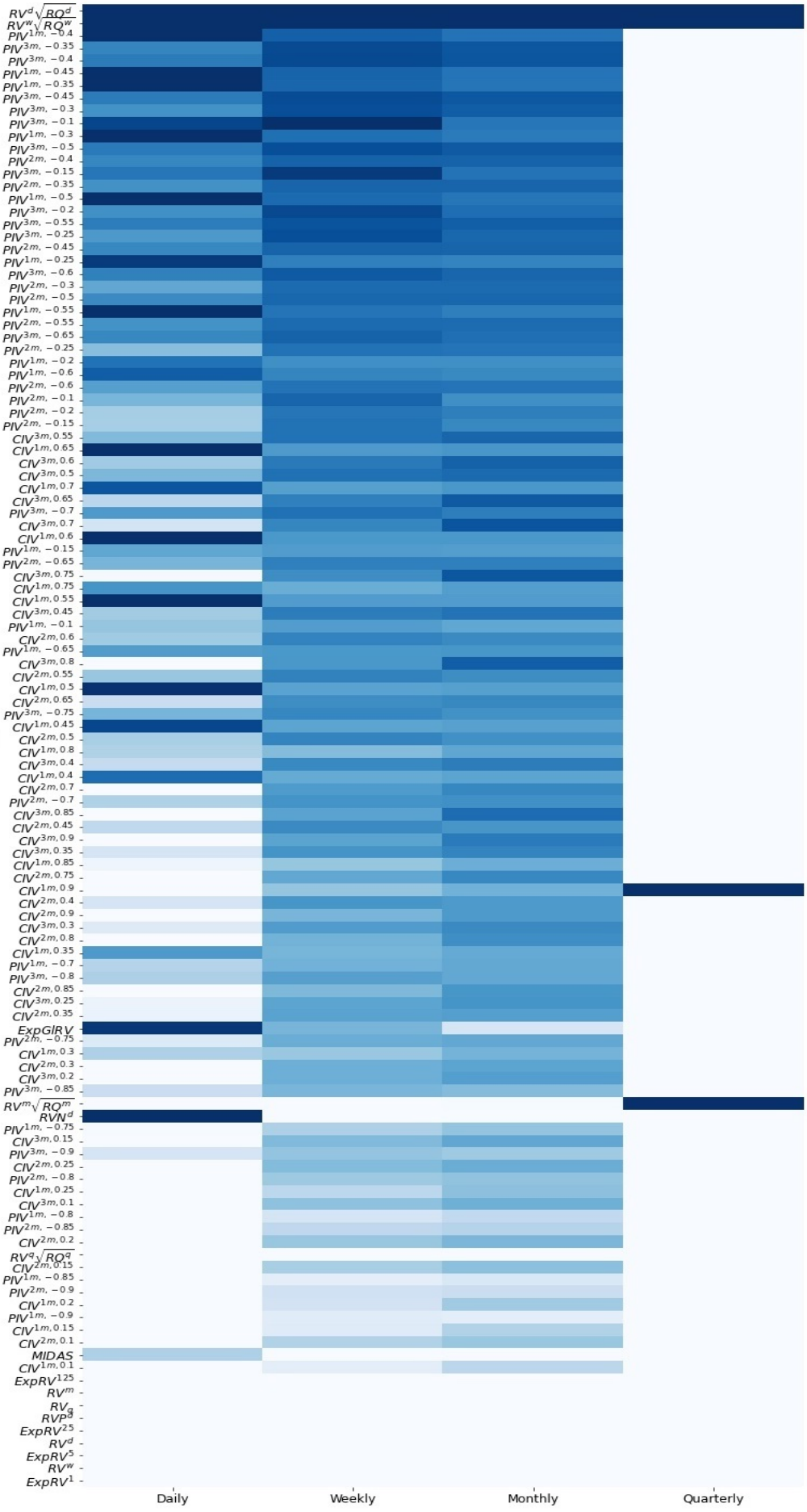





Fig. A.2 Variable importance for PCR: all 118 features

This figure displays the rankings of all 118 features in terms of overall contribution for Principal Component Regressions (PCR) across different forecast horizons. Variable importance is defined in Figure 5. All variables are ordered based on the sum of their importance rankings over all forecast horizons. The columns correspond to each forecast horizon, and the color gradients within each column indicate the most influential (dark blue) to the least influential (white) variables.

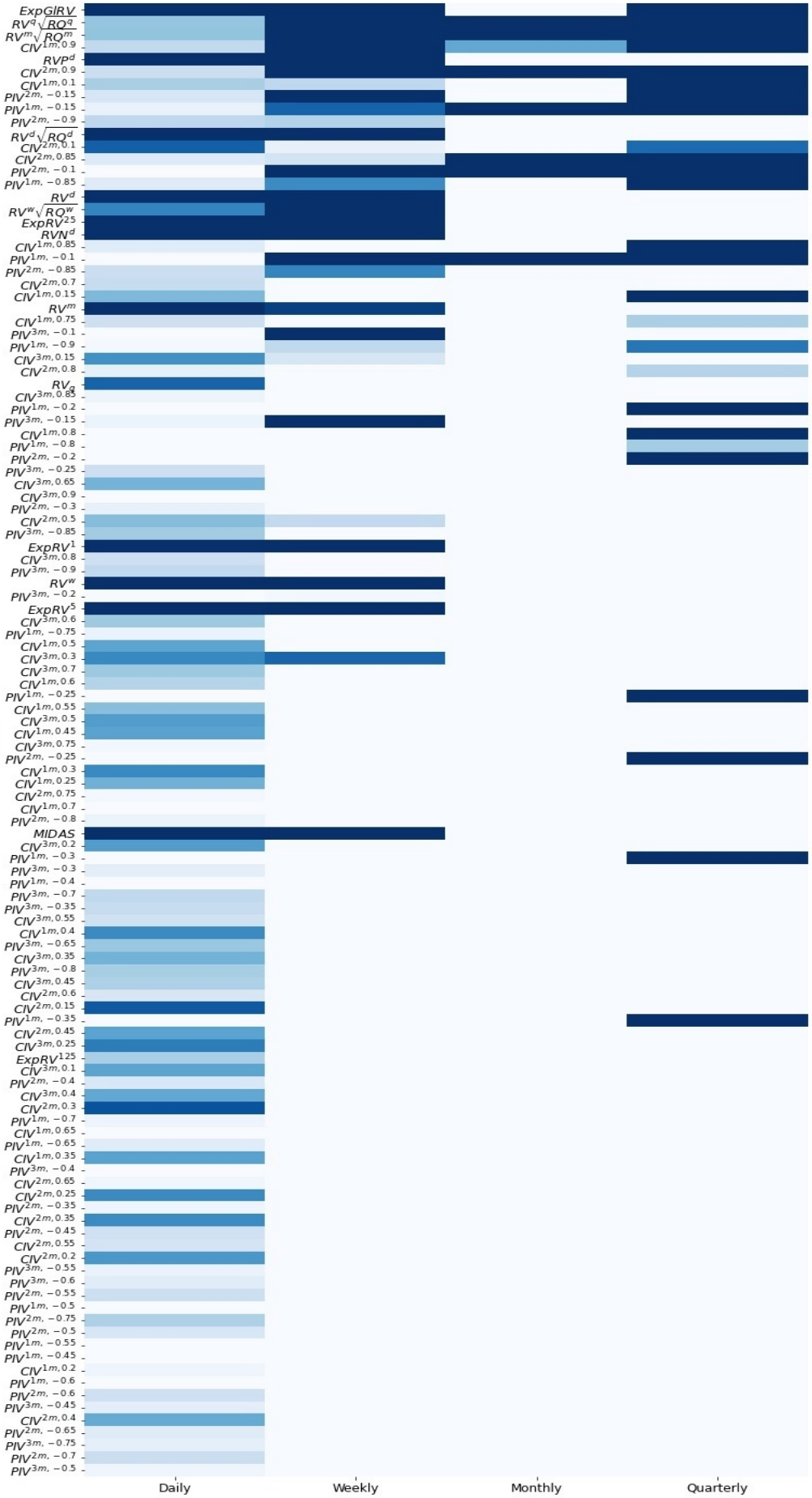





Fig. A.3 Variable importance for RF: all 118 features

This figure displays the rankings of all 118 features in terms of overall contribution for Random Forest (RF) across different forecast horizons. Variable importance is defined in Figure 5. All variables are ordered based on the sum of their importance rankings over all forecast horizons. The columns correspond to each forecast horizon, and the color gradients within each column indicate the most influential (dark blue) to the least influential (white) variables.

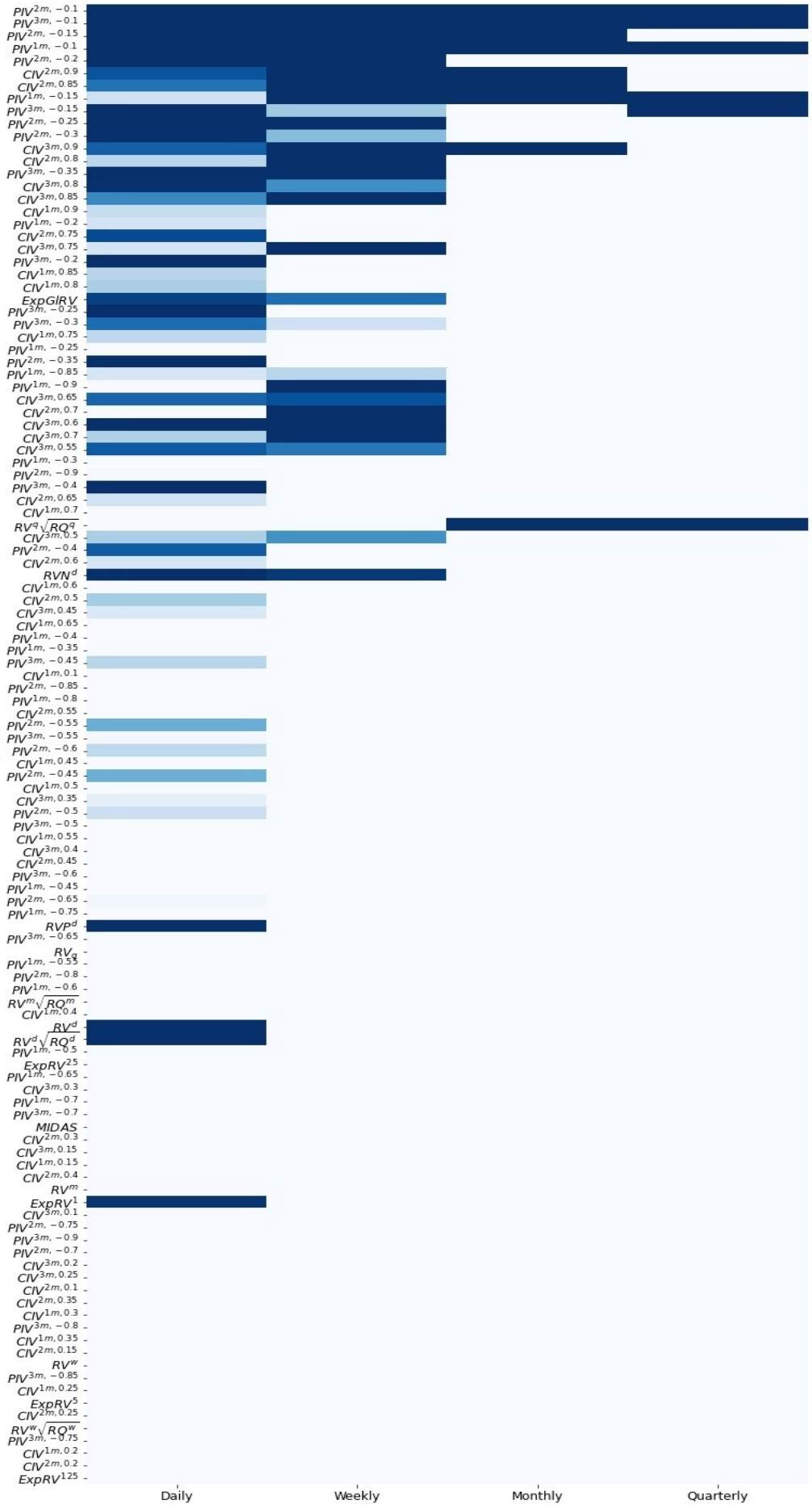





Fig. A.4 Variable importance for GBRT: all 118 features

This figure displays the rankings of all 118 features in terms of overall contribution for Gradient Boosted Regression Trees (GBRT) across different forecast horizons. Variable importance is defined in Figure 5. All variables are ordered based on the sum of their importance rankings over all forecast horizons. The columns correspond to each forecast horizon, and the color gradients within each column indicate the most influential (dark blue) to the least influential (white) variables.

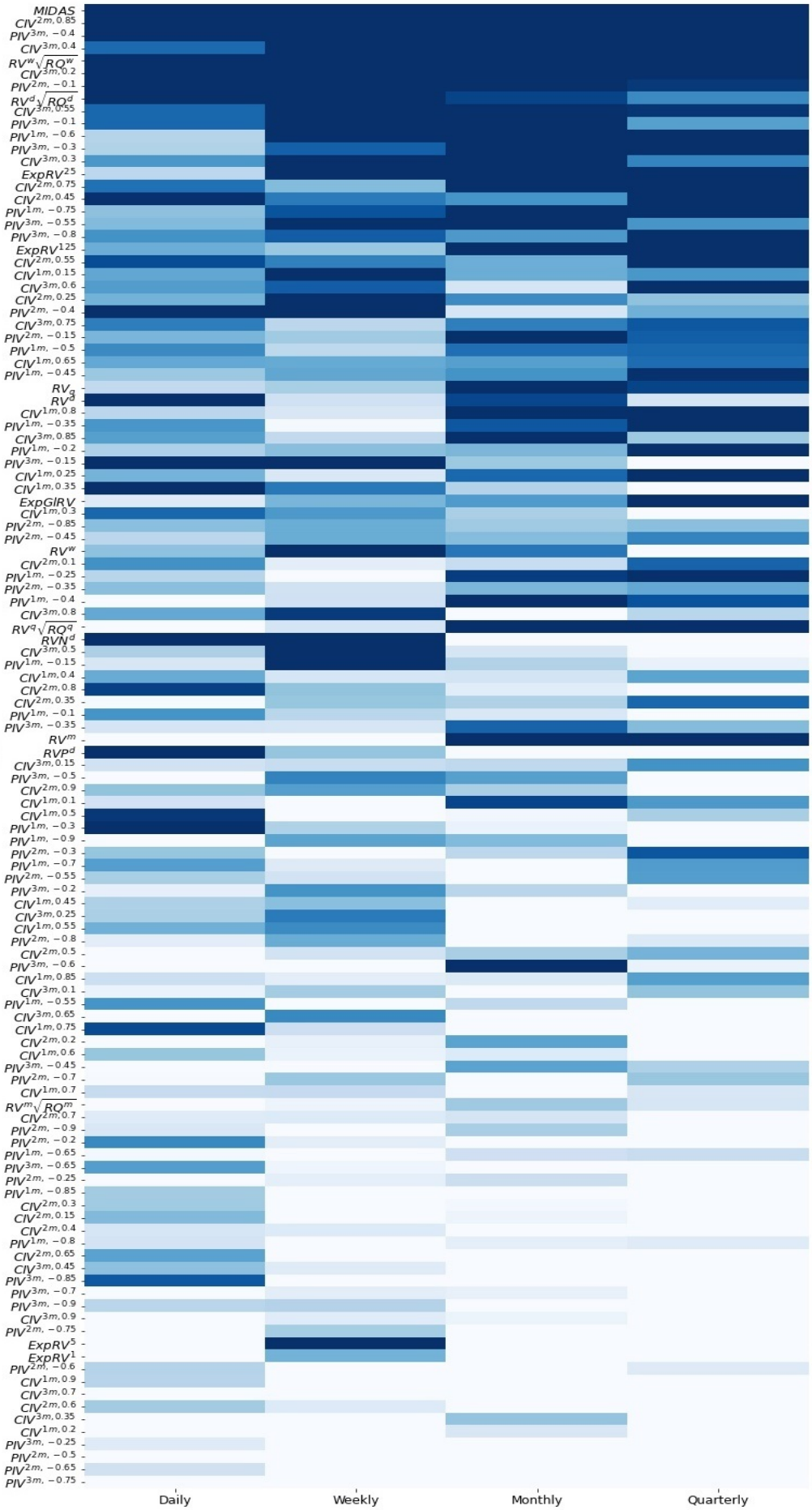



Fig. A.5 Variable importance for NN: all predictors

This figure displays the rankings of all 118 volatility predictors in terms of overall contribution for Neural Network (NN) across different forecast horizons. Variable importance is defined in Figure 5. All variables are ordered based on the sum of their importance rankings over all forecast horizons. The columns correspond to each forecast horizon, and the color gradients within each column indicate the most influential (dark blue) to the least influential (white) variables.

Table A.1 Descriptive statistics of implied variances from call options

This table reports the descriptive statistics of implied variances from call options with delta ranging from 0.1 to 0.9. The sample consists of 173 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 100 index between January 1996 and June 2019 with share code 10 or 11, prices between $1 and $1000, daily number of trades greater than or equal to 100, and at least five years of data on all features and response variables. $CIV^{jm,\delta}$ denotes the implied variance from call options with maturity equal to $j$ months ($j = 1, 2, 3$) and delta equal to $\delta$ ($\delta = 0.1, 0.15, ..., 0.9$).

| | Mean | Std | Skewness | Kurtosis | Min | P5 | Median | P95 | Max | AR(1) | AR(5) | AR(21) | AR(63) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $CIV^{1m,0.1}$ | 0.132 | 0.162 | 5.073 | 45.518 | 0.007 | 0.026 | 0.083 | 0.399 | 3.764 | 0.959 | 0.887 | 0.735 | 0.606 |
| $CIV^{1m,0.15}$ | 0.127 | 0.159 | 5.322 | 50.446 | 0.005 | 0.023 | 0.079 | 0.384 | 3.763 | 0.960 | 0.897 | 0.755 | 0.619 |
| $CIV^{1m,0.2}$ | 0.122 | 0.158 | 5.549 | 55.227 | 0.004 | 0.022 | 0.076 | 0.373 | 3.880 | 0.963 | 0.904 | 0.770 | 0.628 |
| $CIV^{1m,0.25}$ | 0.120 | 0.157 | 5.674 | 57.932 | 0.003 | 0.021 | 0.074 | 0.368 | 4.092 | 0.966 | 0.911 | 0.780 | 0.634 |
| $CIV^{1m,0.3}$ | 0.120 | 0.158 | 5.724 | 58.915 | 0.003 | 0.021 | 0.073 | 0.367 | 4.127 | 0.970 | 0.917 | 0.787 | 0.638 |
| $CIV^{1m,0.35}$ | 0.121 | 0.159 | 5.755 | 59.372 | 0.001 | 0.021 | 0.074 | 0.369 | 4.138 | 0.972 | 0.921 | 0.792 | 0.640 |
| $CIV^{1m,0.4}$ | 0.122 | 0.161 | 5.808 | 60.398 | 0.001 | 0.021 | 0.074 | 0.372 | 4.184 | 0.973 | 0.923 | 0.793 | 0.639 |
| $CIV^{1m,0.45}$ | 0.124 | 0.164 | 5.907 | 63.664 | 0.000 | 0.021 | 0.075 | 0.377 | 5.333 | 0.972 | 0.922 | 0.793 | 0.638 |
| $CIV^{1m,0.5}$ | 0.126 | 0.167 | 5.913 | 63.823 | 0.000 | 0.022 | 0.077 | 0.384 | 6.516 | 0.972 | 0.921 | 0.793 | 0.635 |
| $CIV^{1m,0.55}$ | 0.129 | 0.171 | 5.889 | 62.294 | 0.001 | 0.022 | 0.079 | 0.393 | 5.044 | 0.970 | 0.919 | 0.789 | 0.630 |
| $CIV^{1m,0.6}$ | 0.132 | 0.176 | 5.895 | 62.027 | 0.004 | 0.023 | 0.081 | 0.403 | 4.914 | 0.966 | 0.914 | 0.784 | 0.623 |
| $CIV^{1m,0.65}$ | 0.137 | 0.181 | 5.882 | 61.501 | 0.004 | 0.024 | 0.084 | 0.416 | 5.068 | 0.960 | 0.907 | 0.776 | 0.613 |
| $CIV^{1m,0.7}$ | 0.142 | 0.187 | 5.861 | 61.276 | 0.004 | 0.026 | 0.088 | 0.433 | 5.489 | 0.948 | 0.896 | 0.764 | 0.599 |
| $CIV^{1m,0.75}$ | 0.150 | 0.195 | 5.805 | 60.347 | 0.004 | 0.028 | 0.093 | 0.454 | 5.687 | 0.923 | 0.870 | 0.738 | 0.574 |
| $CIV^{1m,0.8}$ | 0.162 | 0.206 | 5.682 | 58.129 | 0.004 | 0.030 | 0.102 | 0.485 | 5.702 | 0.880 | 0.824 | 0.695 | 0.538 |
| $CIV^{1m,0.85}$ | 0.178 | 0.220 | 5.445 | 53.879 | 0.004 | 0.032 | 0.114 | 0.530 | 5.883 | 0.818 | 0.758 | 0.631 | 0.492 |
| $CIV^{1m,0.9}$ | 0.198 | 0.239 | 5.103 | 47.650 | 0.004 | 0.035 | 0.126 | 0.591 | 6.239 | 0.751 | 0.688 | 0.560 | 0.445 |
| $CIV^{2m,0.1}$ | 0.121 | 0.146 | 5.109 | 46.855 | 0.005 | 0.024 | 0.076 | 0.360 | 3.450 | 0.975 | 0.928 | 0.809 | 0.662 |
| $CIV^{2m,0.15}$ | 0.117 | 0.144 | 5.292 | 50.536 | 0.004 | 0.023 | 0.074 | 0.351 | 3.522 | 0.975 | 0.932 | 0.821 | 0.671 |
| $CIV^{2m,0.2}$ | 0.115 | 0.144 | 5.449 | 53.516 | 0.003 | 0.021 | 0.072 | 0.345 | 3.561 | 0.977 | 0.936 | 0.830 | 0.676 |
| $CIV^{2m,0.25}$ | 0.114 | 0.145 | 5.544 | 55.179 | 0.002 | 0.021 | 0.071 | 0.344 | 3.583 | 0.978 | 0.940 | 0.836 | 0.678 |
| $CIV^{2m,0.3}$ | 0.115 | 0.146 | 5.599 | 56.113 | 0.002 | 0.021 | 0.072 | 0.346 | 3.616 | 0.980 | 0.943 | 0.839 | 0.678 |
| $CIV^{2m,0.35}$ | 0.116 | 0.149 | 5.666 | 57.435 | 0.001 | 0.021 | 0.072 | 0.349 | 3.823 | 0.981 | 0.945 | 0.841 | 0.677 |
| $CIV^{2m,0.4}$ | 0.118 | 0.151 | 5.790 | 60.751 | 0.001 | 0.022 | 0.073 | 0.354 | 4.381 | 0.982 | 0.946 | 0.841 | 0.675 |
| $CIV^{2m,0.45}$ | 0.120 | 0.154 | 5.884 | 63.347 | 0.000 | 0.022 | 0.075 | 0.360 | 4.783 | 0.983 | 0.946 | 0.841 | 0.673 |
| $CIV^{2m,0.5}$ | 0.123 | 0.158 | 5.953 | 65.884 | 0.000 | 0.023 | 0.076 | 0.367 | 4.989 | 0.982 | 0.946 | 0.840 | 0.670 |
| $CIV^{2m,0.55}$ | 0.126 | 0.162 | 5.917 | 64.703 | 0.000 | 0.023 | 0.078 | 0.375 | 5.315 | 0.981 | 0.944 | 0.837 | 0.665 |
| $CIV^{2m,0.6}$ | 0.129 | 0.166 | 5.854 | 62.121 | 0.001 | 0.024 | 0.080 | 0.385 | 5.202 | 0.979 | 0.942 | 0.834 | 0.660 |
| $CIV^{2m,0.65}$ | 0.133 | 0.171 | 5.800 | 60.035 | 0.004 | 0.025 | 0.083 | 0.397 | 4.894 | 0.976 | 0.937 | 0.828 | 0.653 |
| $CIV^{2m,0.7}$ | 0.138 | 0.176 | 5.764 | 59.159 | 0.004 | 0.026 | 0.086 | 0.412 | 6.173 | 0.968 | 0.929 | 0.819 | 0.642 |
| $CIV^{2m,0.75}$ | 0.145 | 0.183 | 5.742 | 59.932 | 0.003 | 0.028 | 0.091 | 0.430 | 7.588 | 0.952 | 0.911 | 0.800 | 0.623 |
| $CIV^{2m,0.8}$ | 0.155 | 0.192 | 5.625 | 56.961 | 0.003 | 0.030 | 0.099 | 0.454 | 7.030 | 0.920 | 0.876 | 0.764 | 0.593 |
| $CIV^{2m,0.85}$ | 0.169 | 0.203 | 5.395 | 52.099 | 0.003 | 0.033 | 0.110 | 0.490 | 6.829 | 0.863 | 0.817 | 0.704 | 0.548 |
| $CIV^{2m,0.9}$ | 0.186 | 0.218 | 5.096 | 46.747 | 0.003 | 0.035 | 0.123 | 0.539 | 5.279 | 0.795 | 0.747 | 0.633 | 0.498 |
| $CIV^{3m,0.1}$ | 0.111 | 0.133 | 5.217 | 50.273 | 0.004 | 0.023 | 0.071 | 0.329 | 3.341 | 0.985 | 0.954 | 0.860 | 0.702 |
| $CIV^{3m,0.15}$ | 0.109 | 0.133 | 5.350 | 53.231 | 0.003 | 0.021 | 0.069 | 0.324 | 3.384 | 0.986 | 0.956 | 0.867 | 0.710 |
| $CIV^{3m,0.2}$ | 0.108 | 0.133 | 5.457 | 55.509 | 0.002 | 0.021 | 0.069 | 0.322 | 3.390 | 0.987 | 0.958 | 0.871 | 0.714 |
| $CIV^{3m,0.25}$ | 0.109 | 0.134 | 5.519 | 56.604 | 0.002 | 0.021 | 0.069 | 0.322 | 3.386 | 0.988 | 0.959 | 0.872 | 0.714 |
| $CIV^{3m,0.3}$ | 0.110 | 0.135 | 5.556 | 57.126 | 0.001 | 0.021 | 0.070 | 0.325 | 3.394 | 0.988 | 0.960 | 0.873 | 0.713 |
| $CIV^{3m,0.35}$ | 0.112 | 0.137 | 5.572 | 57.149 | 0.001 | 0.021 | 0.071 | 0.329 | 3.386 | 0.988 | 0.960 | 0.872 | 0.711 |
| $CIV^{3m,0.4}$ | 0.113 | 0.140 | 5.593 | 57.318 | 0.001 | 0.022 | 0.072 | 0.335 | 3.449 | 0.988 | 0.960 | 0.871 | 0.707 |
| $CIV^{3m,0.45}$ | 0.116 | 0.143 | 5.614 | 57.425 | 0.000 | 0.022 | 0.073 | 0.341 | 3.502 | 0.988 | 0.960 | 0.870 | 0.704 |
| $CIV^{3m,0.5}$ | 0.118 | 0.146 | 5.643 | 57.782 | 0.000 | 0.023 | 0.075 | 0.348 | 3.574 | 0.988 | 0.959 | 0.868 | 0.700 |
| $CIV^{3m,0.55}$ | 0.121 | 0.150 | 5.641 | 57.356 | 0.001 | 0.024 | 0.077 | 0.357 | 3.609 | 0.987 | 0.958 | 0.866 | 0.696 |
| $CIV^{3m,0.6}$ | 0.124 | 0.154 | 5.646 | 57.158 | 0.001 | 0.024 | 0.079 | 0.366 | 3.675 | 0.986 | 0.957 | 0.864 | 0.691 |
| $CIV^{3m,0.65}$ | 0.128 | 0.159 | 5.776 | 61.676 | 0.004 | 0.025 | 0.081 | 0.377 | 4.637 | 0.984 | 0.955 | 0.860 | 0.685 |
| $CIV^{3m,0.7}$ | 0.133 | 0.165 | 5.979 | 70.691 | 0.004 | 0.026 | 0.085 | 0.391 | 6.273 | 0.980 | 0.950 | 0.853 | 0.676 |
| $CIV^{3m,0.75}$ | 0.139 | 0.172 | 5.953 | 69.854 | 0.004 | 0.028 | 0.089 | 0.407 | 6.454 | 0.971 | 0.940 | 0.841 | 0.662 |
| $CIV^{3m,0.8}$ | 0.148 | 0.179 | 5.696 | 59.898 | 0.003 | 0.030 | 0.096 | 0.430 | 5.808 | 0.947 | 0.914 | 0.813 | 0.635 |
| $CIV^{3m,0.85}$ | 0.159 | 0.188 | 5.377 | 51.591 | 0.003 | 0.032 | 0.106 | 0.459 | 4.862 | 0.901 | 0.865 | 0.764 | 0.594 |
| $CIV^{3m,0.9}$ | 0.174 | 0.200 | 5.100 | 46.722 | 0.003 | 0.033 | 0.118 | 0.496 | 4.788 | 0.842 | 0.803 | 0.703 | 0.546 |

Table A.2 Descriptive statistics of implied variances from put options

This table reports the descriptive statistics of implied variances from put options with delta ranging from −0.9 to −0.1. The sample consists of 173 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 100 index between January 1996 and June 2019 with share code 10 or 11, prices between $1 and $1000, daily number of trades greater than or equal to 100, and at least five years of data on all features and response variables. $PIV^{jm,\delta}$ denotes the implied variance from put options with maturity equal to $j$ months ($j = 1, 2, 3$) and delta equal to $\delta$ ($\delta = -0.9, -0.85, ..., -0.1$).

| | Mean | Std. | Skewness | Kurtosis | Min | P5 | Median | P95 | Max | AR(1) | AR(5) | AR(21) | AR(63) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $PIV^{1m,-0.1}$ | 0.195 | 0.248 | 7.924 | 131.814 | 0.002 | 0.044 | 0.127 | 0.554 | 8.488 | 0.969 | 0.907 | 0.759 | 0.592 |
| $PIV^{1m,-0.15}$ | 0.178 | 0.238 | 8.414 | 149.091 | 0.002 | 0.038 | 0.113 | 0.516 | 8.498 | 0.971 | 0.915 | 0.777 | 0.601 |
| $PIV^{1m,-0.2}$ | 0.164 | 0.229 | 8.909 | 168.863 | 0.002 | 0.033 | 0.102 | 0.484 | 8.502 | 0.974 | 0.921 | 0.787 | 0.609 |
| $PIV^{1m,-0.25}$ | 0.154 | 0.221 | 9.388 | 189.520 | 0.002 | 0.029 | 0.095 | 0.460 | 8.509 | 0.977 | 0.926 | 0.793 | 0.617 |
| $PIV^{1m,-0.3}$ | 0.147 | 0.215 | 9.861 | 211.084 | 0.002 | 0.027 | 0.090 | 0.441 | 8.513 | 0.979 | 0.929 | 0.799 | 0.626 |
| $PIV^{1m,-0.35}$ | 0.142 | 0.209 | 10.340 | 233.983 | 0.002 | 0.026 | 0.086 | 0.425 | 8.552 | 0.980 | 0.931 | 0.802 | 0.633 |
| $PIV^{1m,-0.4}$ | 0.138 | 0.205 | 10.834 | 258.174 | 0.002 | 0.025 | 0.083 | 0.413 | 8.565 | 0.980 | 0.932 | 0.804 | 0.638 |
| $PIV^{1m,-0.45}$ | 0.134 | 0.202 | 11.340 | 283.167 | 0.002 | 0.024 | 0.081 | 0.403 | 8.639 | 0.979 | 0.932 | 0.805 | 0.642 |
| $PIV^{1m,-0.5}$ | 0.132 | 0.200 | 11.795 | 305.546 | 0.001 | 0.023 | 0.079 | 0.395 | 8.652 | 0.977 | 0.930 | 0.803 | 0.642 |
| $PIV^{1m,-0.55}$ | 0.130 | 0.199 | 12.163 | 322.964 | 0.001 | 0.023 | 0.078 | 0.389 | 8.686 | 0.971 | 0.924 | 0.798 | 0.640 |
| $PIV^{1m,-0.6}$ | 0.129 | 0.199 | 12.452 | 335.076 | 0.001 | 0.022 | 0.078 | 0.386 | 8.735 | 0.963 | 0.915 | 0.787 | 0.633 |
| $PIV^{1m,-0.65}$ | 0.129 | 0.201 | 12.622 | 340.110 | 0.001 | 0.022 | 0.078 | 0.386 | 8.756 | 0.950 | 0.900 | 0.770 | 0.620 |
| $PIV^{1m,-0.7}$ | 0.131 | 0.204 | 12.605 | 336.208 | 0.001 | 0.022 | 0.079 | 0.390 | 8.775 | 0.929 | 0.874 | 0.741 | 0.597 |
| $PIV^{1m,-0.75}$ | 0.135 | 0.208 | 12.290 | 318.686 | 0.002 | 0.023 | 0.082 | 0.401 | 8.793 | 0.895 | 0.834 | 0.694 | 0.561 |
| $PIV^{1m,-0.8}$ | 0.142 | 0.216 | 11.673 | 288.983 | 0.002 | 0.024 | 0.087 | 0.423 | 8.813 | 0.847 | 0.781 | 0.632 | 0.514 |
| $PIV^{1m,-0.85}$ | 0.153 | 0.225 | 10.804 | 250.567 | 0.002 | 0.026 | 0.094 | 0.453 | 8.818 | 0.796 | 0.726 | 0.567 | 0.467 |
| $PIV^{1m,-0.9}$ | 0.164 | 0.237 | 9.924 | 213.367 | 0.002 | 0.026 | 0.102 | 0.489 | 8.834 | 0.753 | 0.680 | 0.513 | 0.428 |
| $PIV^{2m,-0.1}$ | 0.188 | 0.236 | 8.329 | 148.513 | 0.002 | 0.046 | 0.124 | 0.526 | 8.541 | 0.980 | 0.940 | 0.820 | 0.633 |
| $PIV^{2m,-0.15}$ | 0.173 | 0.227 | 8.852 | 169.073 | 0.002 | 0.040 | 0.111 | 0.492 | 8.520 | 0.983 | 0.945 | 0.832 | 0.644 |
| $PIV^{2m,-0.2}$ | 0.160 | 0.218 | 9.394 | 192.250 | 0.002 | 0.035 | 0.101 | 0.463 | 8.505 | 0.985 | 0.949 | 0.840 | 0.652 |
| $PIV^{2m,-0.25}$ | 0.151 | 0.211 | 9.951 | 217.745 | 0.002 | 0.032 | 0.094 | 0.440 | 8.518 | 0.986 | 0.952 | 0.845 | 0.659 |
| $PIV^{2m,-0.3}$ | 0.144 | 0.205 | 10.522 | 244.968 | 0.002 | 0.029 | 0.090 | 0.422 | 8.478 | 0.987 | 0.953 | 0.849 | 0.667 |
| $PIV^{2m,-0.35}$ | 0.139 | 0.200 | 11.102 | 273.437 | 0.002 | 0.028 | 0.086 | 0.407 | 8.516 | 0.987 | 0.954 | 0.851 | 0.673 |
| $PIV^{2m,-0.4}$ | 0.135 | 0.196 | 11.690 | 303.188 | 0.002 | 0.027 | 0.084 | 0.395 | 8.552 | 0.987 | 0.954 | 0.852 | 0.677 |
| $PIV^{2m,-0.45}$ | 0.132 | 0.193 | 12.280 | 333.250 | 0.002 | 0.026 | 0.082 | 0.385 | 8.639 | 0.986 | 0.954 | 0.853 | 0.680 |
| $PIV^{2m,-0.5}$ | 0.129 | 0.191 | 12.798 | 359.667 | 0.001 | 0.025 | 0.080 | 0.377 | 8.652 | 0.985 | 0.953 | 0.852 | 0.681 |
| $PIV^{2m,-0.55}$ | 0.127 | 0.190 | 13.233 | 381.481 | 0.001 | 0.024 | 0.078 | 0.371 | 8.686 | 0.983 | 0.950 | 0.850 | 0.681 |
| $PIV^{2m,-0.6}$ | 0.126 | 0.189 | 13.609 | 398.441 | 0.001 | 0.024 | 0.077 | 0.366 | 8.735 | 0.979 | 0.946 | 0.845 | 0.679 |
| $PIV^{2m,-0.65}$ | 0.125 | 0.190 | 13.892 | 408.974 | 0.001 | 0.024 | 0.077 | 0.364 | 8.744 | 0.972 | 0.937 | 0.835 | 0.672 |
| $PIV^{2m,-0.7}$ | 0.126 | 0.192 | 14.000 | 409.209 | 0.001 | 0.024 | 0.078 | 0.365 | 8.764 | 0.959 | 0.922 | 0.816 | 0.656 |
| $PIV^{2m,-0.75}$ | 0.128 | 0.196 | 13.872 | 397.480 | 0.002 | 0.024 | 0.080 | 0.371 | 8.791 | 0.936 | 0.895 | 0.782 | 0.626 |
| $PIV^{2m,-0.8}$ | 0.134 | 0.202 | 13.416 | 370.667 | 0.002 | 0.026 | 0.084 | 0.386 | 8.788 | 0.901 | 0.854 | 0.730 | 0.582 |
| $PIV^{2m,-0.85}$ | 0.142 | 0.210 | 12.648 | 331.856 | 0.002 | 0.027 | 0.090 | 0.409 | 8.812 | 0.858 | 0.807 | 0.672 | 0.532 |
| $PIV^{2m,-0.9}$ | 0.151 | 0.219 | 11.790 | 291.699 | 0.002 | 0.028 | 0.096 | 0.436 | 8.837 | 0.818 | 0.763 | 0.621 | 0.488 |
| $PIV^{3m,-0.1}$ | 0.182 | 0.224 | 8.774 | 167.904 | 0.002 | 0.048 | 0.121 | 0.500 | 8.392 | 0.988 | 0.962 | 0.864 | 0.664 |
| $PIV^{3m,-0.15}$ | 0.167 | 0.215 | 9.373 | 194.079 | 0.002 | 0.042 | 0.110 | 0.469 | 8.450 | 0.990 | 0.964 | 0.871 | 0.677 |
| $PIV^{3m,-0.2}$ | 0.156 | 0.207 | 10.044 | 225.247 | 0.002 | 0.037 | 0.100 | 0.442 | 8.479 | 0.991 | 0.966 | 0.875 | 0.686 |
| $PIV^{3m,-0.25}$ | 0.148 | 0.200 | 10.720 | 258.679 | 0.002 | 0.034 | 0.094 | 0.420 | 8.488 | 0.992 | 0.966 | 0.877 | 0.694 |
| $PIV^{3m,-0.3}$ | 0.141 | 0.194 | 11.406 | 294.084 | 0.002 | 0.031 | 0.090 | 0.403 | 8.504 | 0.992 | 0.966 | 0.879 | 0.702 |
| $PIV^{3m,-0.35}$ | 0.136 | 0.190 | 12.104 | 330.536 | 0.002 | 0.030 | 0.087 | 0.389 | 8.530 | 0.992 | 0.967 | 0.881 | 0.707 |
| $PIV^{3m,-0.4}$ | 0.132 | 0.186 | 12.780 | 366.438 | 0.002 | 0.029 | 0.084 | 0.377 | 8.548 | 0.991 | 0.966 | 0.881 | 0.710 |
| $PIV^{3m,-0.45}$ | 0.129 | 0.183 | 13.419 | 400.815 | 0.002 | 0.028 | 0.082 | 0.367 | 8.584 | 0.991 | 0.966 | 0.881 | 0.712 |
| $PIV^{3m,-0.5}$ | 0.126 | 0.181 | 13.983 | 431.430 | 0.002 | 0.027 | 0.080 | 0.359 | 8.642 | 0.990 | 0.965 | 0.880 | 0.714 |
| $PIV^{3m,-0.55}$ | 0.124 | 0.180 | 14.436 | 454.755 | 0.002 | 0.026 | 0.078 | 0.352 | 8.671 | 0.989 | 0.964 | 0.879 | 0.715 |
| $PIV^{3m,-0.6}$ | 0.122 | 0.180 | 14.813 | 470.507 | 0.001 | 0.026 | 0.077 | 0.348 | 8.698 | 0.987 | 0.962 | 0.877 | 0.714 |
| $PIV^{3m,-0.65}$ | 0.121 | 0.181 | 15.100 | 478.208 | 0.001 | 0.025 | 0.077 | 0.344 | 8.716 | 0.983 | 0.957 | 0.872 | 0.709 |
| $PIV^{3m,-0.7}$ | 0.121 | 0.183 | 15.303 | 479.964 | 0.001 | 0.025 | 0.077 | 0.344 | 8.749 | 0.975 | 0.948 | 0.861 | 0.697 |
| $PIV^{3m,-0.75}$ | 0.123 | 0.186 | 15.346 | 474.058 | 0.002 | 0.026 | 0.078 | 0.347 | 8.784 | 0.961 | 0.932 | 0.840 | 0.673 |
| $PIV^{3m,-0.8}$ | 0.127 | 0.190 | 15.059 | 452.822 | 0.002 | 0.026 | 0.081 | 0.357 | 8.815 | 0.936 | 0.904 | 0.805 | 0.634 |
| $PIV^{3m,-0.85}$ | 0.133 | 0.196 | 14.411 | 416.578 | 0.002 | 0.027 | 0.086 | 0.373 | 8.830 | 0.902 | 0.868 | 0.760 | 0.586 |
| $PIV^{3m,-0.9}$ | 0.140 | 0.203 | 13.619 | 376.486 | 0.002 | 0.028 | 0.091 | 0.393 | 8.840 | 0.868 | 0.833 | 0.718 | 0.541 |

Table A.3 Out-of-sample prediction relative to long-run mean: OLS-based models

This table reports the out-of-sample $R^2$ relative to the historical mean of realized volatilities for OLS-based volatility forecasting models across different forecast horizons. The sample consists of 173 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 100 index between January 1996 and June 2019 with share code 10 or 11, prices between \$1 and \$1000, daily number of trades greater than or equal to 100, and at least five years of data on all features and response variables. The full out-of-sample evaluation period is from January 2001 to June 2019. Superscriptions $d$, $w$, $m$, and $q$ are abbreviations for daily, weekly, monthly, and quarterly construction interval or forecast horizon. $MIDAS$ denotes the smoothly weighted moving average of 50 lagged realized variances using validated polynomials from Eqs. (4) and (5) for the corresponding forecast horizon. $RV^k$ $(k = d, w, m, q)$ is the daily, weekly, monthly or quarterly realized variance. $RVP^d$ and $RVN^d$ are the daily realized positive and negative semivariances, respectively. $RV^k\sqrt{RQ^k}$ $(k = d, w, m, q)$ is the product of the realized variance and the square root of the realized quarticity with the same construction interval $k$. $ExpRV^i$ $(i = 1, 5, 25, 125)$ is the exponentially weighted moving average of the past 500-day realized variances using the corresponding center-of-mass $i$ from Eq. (11). $ExpGlRV$ is the exponentially weighted moving average of the global risk factor with a 5-day center-of-mass from Eq. (12). $CIV^{jm,\delta}$ and $PIV^{jm,-\delta}$ are implied variances from call and put options with absolute $\delta = 0.1, 0.15, ..., 0.9$ and maturity equal to $j$ months $(j = 1, 2, 3)$. Our OLS-based models include MIDAS, SHAR, HARQ-F, HExpGl, $OLS^{RM}$ (i.e., simple OLS model with all 16 realized features as predictors), $OLS^{IV}$ (i.e., simple OLS model with all 102 implied variance features as predictors), and $OLS^{ALL}$ (i.e., simple OLS model with all 118 realized and implied variance features as joint predictors). $R^2_{OOS}$ for each model at each forecast horizon is calculated relative to the long-run mean of $RV$ using the entire panel of stocks according to Eq. (20).

| Model | Features | Daily | Weekly | Monthly | Quarterly |
|---|---|---|---|---|---|
| | | $R^2_{OOS}$ relative to long-run mean | | | |
| HAR | $RV^d$, $RV^w$, $RV^m$, $RV^q$ | 57.8% | 69.4% | 70.0% | 63.6% |
| MIDAS | $MIDAS$ term for the corresponding forecast horizon | 58.2% | 70.6% | 71.3% | 64.2% |
| SHAR | $RVP^d$, $RVN^d$, $RV^w$, $RV^m$, $RV^q$ | 58.4% | 69.9% | 70.4% | 63.9% |
| HARQ-F | $RV^d$, $RV^w$, $RV^m$, $RV^q$, $RV^d\sqrt{RQ^d}$, $RV^w\sqrt{RQ^w}$, $RV^m\sqrt{RQ^m}$, $RV^q\sqrt{RQ^q}$ | 58.7% | 70.3% | 71.0% | 65.4% |
| HExpGl | $ExpRV^1$, $ExpRV^5$, $ExpRV^{25}$, $ExpRV^{125}$, $ExpGlRV$ | 57.8% | 70.2% | 70.6% | 63.1% |
| $OLS^{RM}$ | $MIDAS$ term for the corresponding forecast horizon, $RV^d$, $RV^w$, $RV^m$, $RV^q$, $RVP^d$, $RVN^d$, $RV^d\sqrt{RQ^d}$, $RV^w\sqrt{RQ^w}$, $RV^m\sqrt{RQ^m}$, $RV^q\sqrt{RQ^q}$, $ExpRV^1$, $ExpRV^5$, $ExpRV^{25}$, $ExpRV^{125}$, $ExpGlRV$ (# of features = 16) | 59.8% | 71.4% | 71.6% | 64.3% |
| $OLS^{IV}$ | $CIV^{jm,\delta}$ and $PIV^{jm,-\delta}$, $j = 1, 2, 3$, $\delta = 0.1, 0.15, ..., 0.9$ (# of features = 102) | 53.6% | 67.2% | 69.1% | 62.9% |
| $OLS^{ALL}$ | All 118 Features (16 realized features + 102 $IV$ features) | 61.0% | 73.0% | 72.2% | 63.4% |

Table A.4 Out-of-sample predictions relative to long-run mean: Machine learning-based models

This table reports the out-of-sample $R^2$ relative to the historical mean of realized volatilities for machine learning-based volatility forecasting models across different forecast horizons. The sample consists of 173 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 100 index between January 1996 and June 2019 with share code 10 or 11, prices between \$1 and \$1000, daily number of trades greater than or equal to 100, and at least five years of data on all features and response variables. The full out-of-sample evaluation period is from January 2001 to June 2019. The features of each model consist of all 118 predictors detailed in Table 3. Our machine learning-based models include LASSO, Principal Component Regression (PCR), Random Forest (RF), Gradient Boosted Regression Trees (GBRT), Neural Network (NN), and a simple average of forecasts from the five individual machine learning models (AVG). Tuning parameters for each model are in **bold**. $R^2_{OOS}$ for each model at each forecast horizon is calculated relative to the long-run mean of $RV$ using the entire panel of stocks according to Eq. (20).

| Model | Hyperparameter (Tuning parameter in **bold**) | Daily | Weekly | Monthly | Quarterly |
|---|---|---|---|---|---|
| | | $R^2_{OOS}$ relative to long-run mean | | | |
| LASSO | **# of shrinkage parameters ($\lambda$): 100**<br>$\lambda_{min}/\lambda_{max}$: 0.001 | 61.1% | 73.1% | 73.4% | 64.6% |
| PCR | **# of components: 1, 2, ..., 20** | 60.1% | 70.9% | 72.4% | 66.5% |
| RF | **Maximum tree depth ($L$): 1, 2, ..., 20**<br># of trees: 500<br>Subsample: 0.5<br>Subfeature: log(# of features) | 59.1% | 71.4% | 72.8% | 65.6% |
| GBRT | **# of trees ($B$)**<br>**Maximum tree depth ($L$): 1, 2, ..., 5**<br>Learning rate: 0.001<br>Subsample: 0.5<br>Subfeature: log(# of features)<br>Early-stopping rules (whichever met first):<br>1) No reduction in MSE after 50 iterations<br>2) Max # of trees hit 20,000 | 59.8% | 72.6% | 73.2% | 65.9% |
| NN | # of hidden layer: 1<br># of neurons: 10<br>Activation function: ReLU | 61.3% | 72.6% | 72.4% | 65.0% |
| AVG | | 61.7% | 73.7% | 74.5% | 67.4% |

# References


An, B.-J., Ang, A., Bali, T. G., Cakici, N., 2014. The joint cross section of stocks and options. Journal of Finance 69, 2279–2337.

Andersen, T. G., Bollerslev, T., 1998. Answering the skeptics: Yes, standard volatility models do provide accurate forecasts. International Economic Review 39(4), 885–905.

Andersen, T. G., Bollerslev, T., Christoffersen, P. F., Diebold, F. X., 2006. Volatility and correlation forecasting. In G. Elliott, C. W. J. Granger, and A. Timmermann, eds. Handbook of Economic Forecasting. North Holland, Amsterdam.

Andersen, T. G., Bollerslev, T., Diebold, F. X., Labys, P., 2001. The distribution of realized exchange rate volatility. Journal of the American Statistical Association 96, 42–55.

Andersen, T. G., Bollerslev, T., Diebold, F. X., Labys, P., 2003. Modeling and forecasting realized volatility. Econometrica 71, 579–625.

Audrino, F., Knaus, S. D., 2016. Lassoing the HAR model: A model selection perspective on realized volatility dynamics. Econometric Review 35, 1485–1521.

Bali, T. G., Goyal, A., Huang, D., F., J., Wen, Q., 2020. The cross-sectional pricing of corporate bonds using big data and machine learning. Working paper, Georgetown University, Swiss Fiance Institute, Singapore Management University, and Central University of Finance and Economics.

Barndorff-Nielsen, O. E., Kinnebrock, S., Shephard, N., 2010. Measuring downside risk: Realised semivariance. In T. Bollerslev, J. Russell, and M. Watson, eds., Volatility and Time Series Econometrics: Essays in Honor of Robert F. Engle.

Barndorff-Nielsen, O. E., Shephard, N., 2002. Econometric analysis of realized volatility and its use in estimating stochastic volatility models. Journal of the Royal Statistical Society Series B 64, 253–280.

Bianchi, D., Büchner, M., Tamoni, A., 2021. Bond risk premia with machine learning. Review of Financial Studies 34(2), 1046–1089.

Bollerslev, T., Engle, R. F., Nelson, D., 1994. ARCH models. In R. F. Engle and D. McFadden, eds., Handbook of Econometrics 4. North Holland, Amsterdam.

Bollerslev, T., Hood, B., Huss, J., Pedersen, L. H., 2018. Risk everywhere: Modeling and managing volatility. Review of Financial Studies 31, 2729–2773.

Bollerslev, T., Li, S. Z., Todorov, V., 2016a. Roughing up beta: continuous vs. discontinuous betas, and the cross-section of expected stock returns. Journal of Financial Economics 120, 464–490.

Bollerslev, T., Li, S. Z., Zhao, B., 2020. Good volatility, bad volatility, and the cross section of stock returns. Journal of Financial and Quantitative Analysis 55, 1–31.

Bollerslev, T., Litvinova, J., Tauchen, G., 2006. Leverage and volatility feedback effects in high-frequency data. Journal of Financial Econometrics 4(3), 353–384.

Bollerslev, T., Patton, A. J., Quaedvlieg, R., 2016b. Exploiting the errors: A simple approach for improved volatility forecasting. Journal of Econometrics 192, 1–18.

Bucci, A., 2020. Realized volatility forecasting with neural networks. Journal of Financial Econometrics 18(3), 502–531.

Carr, P., Wu, L., Zhang, Z., 2020. Using machine learning to predict realized variance. Journal of Investment Management 18(2), 1–16.

Corsi, F., 2009. A simple approximate long-memory model of realized volatility. Journal of Financial Econometrics 7, 174–196.

Diebold, F. X., Mariano, R. S., 1995. Comparing predictive accuracy. Journal of Business and Economic Statistics 13, 253–263.

Ghysels, E., Qian, H., 2019. Estimating MIDAS regressions via OLS with polynomial parameter profiling. Econometrics and Statistics 9, 1–16.

Ghysels, E., Santa-Clara, P., Valkanov, R., 2006. Predicting volatility: Getting the most out of return data sampled at different frequencies. Journal of Econometrics 131, 59–96.

Ghysels, E., Sinko, A., Valkanov, R., 2007. MIDAS regressions: Further results and new directions. Econometric Review 26, 53–90.

Gu, S., Kelly, B., Xiu, D., 2020. Empirical asset pricing via machine learning. Review of Financial Studies 33, 2223–2273.

Herskovic, B., Kelly, B., Lustig, H., Nieuwerburgh, V., 2016. The common factor in idiosyncratic volatility: Quantitative asset pricing implications. Journal of Financial Economics 119, 249–283.

Herskovic, B., Kelly, B., Lustig, H., Nieuwerburgh, V., 2021. Firm volatility in granular networks. Journal of Political Economy, forthcoming.

Jiang, H., Li, S., Wang, H., 2021. Pervasive underreaction: Evidence from high-frequency data. Journal of Financial Economics, forthcoming.

Kingma, D. P., Ba, J., 2015. Adam: A method for stochastic optimization. 3rd International Conference for Learning Representations, Conference paper.

Luong, C., Dokuchaev, N., 2018. Forecasting of realised volatility with the random forests algorithm. Journal of Risk and Financial Management 11(4), 1–15.

Patton, A. J., Sheppard, K., 2015. Good volatility, bad volatility: Signed jumps and the persistence of volatility. Review of Economics and Statistics 97, 683–697.

Rahimikia, E., Poon, S.-H., 2020. Machine learning for realised volatility forecasting. Working paper, University of Manchester.

Rossi, A. G., 2018. Predicting stock market returns with machine learning. Working paper, Georgetown University.

Swanson, N. R., White, H., 1997. Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models. International Journal of Forecasting 13, 439–461.

Taylor, S., 2005. Asset price dynamics, volatility, and prediction. Princeton, NJ: Princeton University Press.

Zhang, L., Mykland, P. A., Aït-Sahalia, Y., 2005. A tale of two time scales: Determining integrated volatility with noisy high-frequency data. Journal of the American Statistical Association 100, 1394–1411.

Zhang, T., Yu, B., 2005. Boosting with early stopping: Convergence and consistency. Annals of Statistics 33, 1538–1579.