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

Sophia Zhengzi Li† and Yushan Tang‡

First Draft: November 18, 2020

This Version: November 4, 2021

Abstract

We propose an automatic machine-learning system to forecast realized volatility for S&P 100 stocks using 118 features and five machine learning algorithms. A simple average ensemble model combining all learning algorithms delivers extraordinary performance across forecast horizons, and the improvement in out-of-sample R^2's translates into nontrivial economic gains. We further augment the feature set by including firm characteristics and pure noise terms, and find that the system continues to perform well after including weak or noisy features. Finally, we demonstrate that our learning system is scalable to a broader S&P 500 stock universe via hyperparameter transfer learning for nonlinear models.

JEL Classification: C13, C14, C52, C53, C55, C58.

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

*We thank Tim Bollerslev, Todd Griffith, Hao Jiang, Andrea Tamoni, Allan Timmermann, Yuanyuan Xiao, Dacheng Xiu, Peixuan Yuan, and seminar participants at Rutgers Business School, Johns Hopkins Carey Business School, Michigan State Broad College of Business, NBER-NSF Time Series Conference Virtual Poster Session, and FMA Annual Meeting for their helpful comments and suggestions.

†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 squared intraday returns.[1] Most of the existing RV 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, instead of arguing the dominance of a particular feature or algorithm, we have an ambitious objective: building 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 consider 118 features that might be useful in predicting future volatility, including 16 realized-variance-based (RV-based) features proposed by five popular RV forecasting models and 102 implied-variance-based (IV-based) features across all deltas and with maturity between one and three months. 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 five popular learning algorithms: LASSO, Principal Component Regression (PCR), Random Forecast (RF), Gradient Boosted Regression Trees (GBRT), and Neural Network (NN). These algorithms are more prediction-oriented and capable of capturing complicated relations than simple OLS. Rather than emphasizing the performance of a particular algorithm, we consider a simple combination of all machine learning algorithms. This ensemble model is less prone to human decision-making biases and approves robust throughout our analyses.

We illustrate the automatic forecasting system through perhaps the largest-scale experiment that compares different combinations of features and learning algorithms for 173 S&P 100 stocks and 663 S&P 500 stocks. 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)

[1] 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.

dynamically fitting the same feature set with machine learning algorithms can further improve the performance over OLS; 4) an ensemble model that uses all features and all machine learning algorithms performs extraordinarily well across forecast horizons and under different market conditions; and 5) the improvement in out-of-sample prediction accuracy can translate into nontrivial economic gains for a mean-variance investor who forecasts future volatilities based on the ensemble model.

We start by comparing the predictive power of different features for the S&P 100 stock sample using the same traditional OLS fit. We consider five RV-based feature sets from existing volatility forecasting models including the HAR model by Corsi (2009); 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 Quaedvlieg (2016b); and the HExpGl model by Bollerslev, Hood, Huss, and Pedersen (2018), as well as an additional IV-based feature set. We find that the forecasting performance of any stand-alone RV-based feature set can be improved if we combine them all together, whose performance can be further improved by adding the IV-based features.

After fixing the feature set, our second exercise evaluates learning algorithms in comparison with OLS models. We find that machine learning algorithms can improve performance over that of OLS models. A simple average ensemble model that combines all machine learning algorithms delivers extraordinary performance across forecast horizons, with R^2_{OOS}'s relative to HAR equal to 9.0%, 14.3%, 15.2%, and 10.0% at daily, weekly, monthly, and quarterly horizons. The corresponding relative out-of-sample R^2's further jump to 10.4%, 18.7%, 29.6%, and 27.5% for the most recent decade, indicating that our automatic volatility forecasting system becomes increasingly powerful over time. We next show that the superior out-of-sample prediction performance also translates into economic gains. Under a utility-based framework developed by Bollerslev, Hood, Huss, and Pedersen (2018), we compute the realized utility of a mean-variance investor who trades an asset with a constant Sharpe ratio. We show that at the monthly forecast horizon, using the ensemble model is worth 24 basis points (bps) per year relative to using an OLS model with all features. In other words, the investor is willing to pay 24 bps of his wealth to have access to the ensemble model rather than investing according to OLS.

We further investigate which types of features are the most important in forecasting RV. Our

feature importance evaluation process differs from those in existing studies in two important ways. First, because several subgroups of our feature set are highly correlated, we assign highly correlated features into one group and compute the importance of each group. The motivation is to avoid the dilution effect of per-feature variable importance. Specifically, we divide 118 features into three groups: 1) realized features from the MIDAS and HExpGl models; 2) semi-variances and daily, weekly, monthly, and quarterly RV's as well as $RV\sqrt{RQ}$'s; and 3) implied variance features. Second, instead of setting all values of the features within a group to zero, we consider random permutations of values across observations within the training set for the tested features as suggested by Fisher, Rudin, and Dominici (2019). This is because setting all feature values to zero introduces unintended bias to nonlinear models. We find that all three groups of features contribute at least 10% to the forecast across most horizons and learning algorithms. Interestingly, implied variance features increasingly contribute more to volatility forecasting through time. We conjecture that much of the gain comes from the improved quality of implied variance features as the overall option market becomes more liquid.

To test the robustness of the system, we augment the feature set by including six firm characteristics and six pure noise terms. Firm characteristics have not been widely documented as useful predictors of volatilities. Therefore, if our system is robust, adding these 12 features should not materially change the forecasting performance. Indeed, we find that firm characteristics do not contribute much to volatility forecasting and our system continues to perform well even after including these weak or noisy features. Interestingly, the contribution of firm characteristics is slightly higher at around 3% for nonlinear models RF, GBRT, and NN compared with around 1% for linear models LASSO and PCR. One possible explanation is that firm characteristics may help predict future RV only as interaction terms with existing RV predictors. The noise features, on the other hand, contribute almost nothing to model prediction, indicating that our system and the associated group importance metrics effectively control for false positives.

Is our machine-learning-based automatic system scalable to more stocks? To this end, we examine its performance on a large and different set of 663 S&P 500 stocks. To speed up fitting nonlinear models, we transfer tuning parameters already learned from the S&P 100 stock universe to the S&P 500 universe. We find that tuning parameters based on the original sample perform well in the new sample, and our automatic system consistently delivers significant gains over the

traditional OLS-based approach. In terms of utility gains, the ensemble model delivers 44 more bps per year to the mean-variance investor relative to using OLS for this large stock universe.

We offer two main contributions to the literature: one for our methodology and one for empirics. Regarding methodology, we propose a modern machine-learning-based framework for volatility forecasting. Within this framework, we decompose the volatility forecasting task into feature engineering and learning algorithm fitting steps. In the feature engineering step, 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-oriented 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 as suggested by Wolpert (1996). Instead, we consider combinations across 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 a particular sample period) and appears to be robust throughout our analyses.

Regarding empirics, we conduct perhaps the largest-scale experiment involving the forecasting of realized stock-price volatility. Our big dataset consists of intraday high-frequency data and stock-level option data for 173 S&P 100 stocks and another 663 S&P 500 stocks for the period from January 1996 to June 2019. Our giant feature set includes predictors drawn from five popular RV-based volatility forecasting models and implied variances with one-to-three-month maturity across all deltas. Our learning algorithms consist of major linear and nonlinear machine learning models. With our comprehensive data and unique study design, we empirically demonstrate the gains that can be obtained when using the new automatic system based on many features and learning algorithms to forecast realized volatility.

There is 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 bond returns. Bali, Goyal, Huang, Jiang, and Wen (2020) study cross-sectional predictability of corporate bond returns using both stock and bond characteristics via ML. Li and Rossi (2021) select mutual funds using stock-based fund characteristics via ML. In

addition, several papers apply selective ML algorithms to volatility forecasting problems: Audrino and Knaus (2016) use LASSO to forecast realized volatility; Luong and Dokuchaev (2018) forecast realized volatility with random forest algorithms; Rossi (2018) employs Boosted Regression Trees to forecast stock returns and volatility at monthly frequency; Bucci (2020) and Rahimikia and Poon (2020) apply neural networks to predict realized volatility; and 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 studies that apply machine learning to volatility forecasting, our work focuses on building an entire learning system that is automatic and robust. Again, we emphasize the benefits of using not just one or two particular learning algorithms, such as RF or GBRT, but more specifically the benefits of an ML-based system that allows us to consider features and algorithms more inclusively, because machines are able to scan, fit, and select features in a robust and prediction-error-optimized fashion.

2. Data and Response Variable

2.1. Data

We consider a large universe of stocks that were ever constituents of the S&P 100 index over the period from January 1993 to June 2019, listed on the New York Stock Exchange (NYSE), National Association of Securities Dealers Automated Quotations (NASDAQ), and the 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 the 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 (2021a).[2] In addition to the TAQ data, 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 the stock and date levels. 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 months (91 days), and absolute

[2]Further details regarding the TAQ data-cleaning rules are provided in the Appendix.

delta of 0.1, 0.15, ..., 0.9.[3] Some RV-based features (e.g., features from the HExpGl model) detailed in Section 3 require a longer historical sample for estimation. To ensure that all RV-based features have the same history, we use the sample between 1993 and 1995 to construct their first observations; therefore our features first become available in January 1996. Our final stock sample consists of 173 unique stocks with at least five years of data for all features and response variables over the period from January 1996 to June 2019.

Our S&P 100 stock universe is large-scale by the volatility forecasting literature standard. Another paper we are aware of that uses such a large dataset to forecast volatility 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 errors. In Section 8, we also examine the out-of-sample performance of various models for a different set of 663 S&P 500 stocks. To the best our knowledge, this universe is the largest ever that has been explored in the RV forecasting literature.

2.2. Response Variable

As in every predictive problem, we first need to 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 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 μ and σ denote the drift and diffusive volatility processes, respectively, W is a standard Brownian motion, 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

[3] Implied variances with ten-day maturity only became available in November 2005 for a handful of stocks and are excluded from our analyses because of limited availability of data.

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 ith 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. Formally, the h-day ahead RV is defined as:

$$RV_{t+1}^{t+h} = \frac{1}{h} \sum_{i=1}^{h} RV_{t+i}^d, \tag{3}$$

where $h = 5,\ 21,\ 63$ corresponds to weekly, monthly, and quarterly RV, respectively.[4]

Our research objective is to build better predictive models for the responses of daily, weekly, monthly, and quarterly RVs. To empirically compute RV, we use the 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 separate 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.[5]

3. Features

Our machine learning predictive system consists of two components: features and learning algorithms. Generally speaking, our research design involves first constructing input features that potentially

[4] From a heuristic perspective, French, Schwert, and Stambaugh (1987), Schwert (1989), and Schwert and Seguin (1990) rely on the sum of intra-month squared daily returns to estimate monthly U.S. equity volatilities.

[5] Other consistent but more complicated RV estimators, such as the two-scale RV of Zhang, Mykland, and Aït-Sahalia (2005), the kernel-based RV of Barndorff-Nielsen, Hansen, Lunde, and Shephard (2008), and the pre-averaged RV of Jacod, Li, Mykland, Podolskij, and Vetter (2009), all require the choice of additional parameters. From a theoretical perspective, Andersen, Bollerslev, and Meddahi (2011) and Ghysels and Sinko (2011) show that the simple subsampled 5-minute RV perform on par with or better than these more complicated estimators. Empirically, Liu, Patton, and Sheppard (2015) compare more than 400 different RV estimators across multiple asset classes and conclude that it is difficult to significantly beat the 5-minute RV.

contain predictive information, then fitting learning algorithms to estimate functions that map features to the response variable, and finally evaluating the performance of our predictions. In this section, we discuss how we construct our feature sets. We consider two types of features: 1) features proposed by popular RV-based volatility forecasting models: HAR, SHAR, MIDAS, HARQ-F, and HExpGl; and 2) features from option-implied variances. 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 mix short- (daily), medium- (weekly), and long-term (monthly) volatility components for capturing various empirical properties observed in volatility series such as long memory and fat tails. The original HAR is used to forecast volatility up to monthly horizon. As our longest forecast horizon is quarterly, we augment the HAR model with a quarterly RV term:

$$RV_{t+1}^{t+h} = \beta_0 + \beta_d RV_t^d + \beta_w RV_t^w + \beta_m RV_t^m + \beta_q RV_t^q + \epsilon_t, \tag{4}$$

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) assumes the following form:

$$RV_{t+1}^{t+h} = \beta_0 + \beta_1 MIDAS_t^k + \epsilon_t, \tag{5}$$

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

$$\begin{aligned} MIDAS_t^k &= \frac{1}{\sum_{i=1}^{L} a_i}(a_1 RV_t^d + a_2 RV_{t-1}^d + ... + a_L RV_{t-L+1}^d), \\ a_i &= (\frac{i}{L})^{\theta_1 - 1}(1 - \frac{i}{L})^{\theta_2 - 1}\Gamma(\theta_1 + \theta_2)\Gamma(\theta_1)^{-1}\Gamma(\theta_2)^{-1}, i = 1, ..., L, \end{aligned} \tag{6}$$

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 RVs. It has three hyperparameters θ_1, θ_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 θ_2 for each h-day forecast horizon and choose the value that minimizes the Mean Squared Errors (MSE) over the full sample.[6]

3.3. SHAR

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

$$RV_{t+1}^{t+h} = \beta_0 + \beta_d^+ RVP_t^d + \beta_d^- RVN_t^d + \beta_w RV_t^w + \beta_m RV_t^m + \beta_q RV_t^q + \epsilon_t, \tag{7}$$

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

$$RVP_t^d = 252 \times \sum_{i=1}^{n} r_{t-1+i/n}^2 1_{\{r_{t-1+i/n}>0\}},\ RVN_t^d = 252 \times \sum_{i=1}^{n} r_{t-1+i/n}^2 1_{\{r_{t-1+i/n}<0\}}. \tag{8}$$

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 RVs.[7] To mitigate bias in realized semivariance estimates, we also apply the subsampling scheme to construct RVP^d and RVN^d.

[6] To avoid onerous computational burdens, we follow the literature and do not perform a rolling grid search for the θ_2 parameter. As a result, the MIDAS feature is not truly out-of-sample but is included for comparison.

[7] Patton and Sheppard (2015) also rely on the difference between positive and negative realized semivariances to isolate signed jumps, e.g., $\Delta J_t^2 = RVP_t^d - RVN_t^d$, and show that ΔJ_t^2 negatively predicts future RV. Our Eq. (7) nests the specification of including signed jump variation when $\beta_d^+ = -\beta_d^-$. On a related note, Andersen, Bollerslev, and Diebold (2007) find that unsigned jumps lead to only a slight decrease in future RV.

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):

$$RV_t = IV_t^* + \epsilon_t, \;\; \epsilon_t \sim MN(0, 2\Delta IQ_t), \tag{9}$$

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):

$$RQ_t^d = 252^2 \times \frac{n}{3} \sum_{i=1}^{n} r_{t-1+i/n}^4. \tag{10}$$

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 RVs to vary as a function of $\sqrt{RQ}$:

$$\begin{aligned} RV_{t+1}^{t+h} = \beta_0 + \beta_d RV_t^d + \beta_w RV_t^w + \beta_m RV_t^m + \beta_q RV_t^q \\ + \phi_d RV_t^d \sqrt{RQ_t^d} + \phi_w RV_t^w \sqrt{RQ_t^w} + \phi_m RV_t^m \sqrt{RQ_t^m} + \phi_q RV_t^q \sqrt{RQ_t^q} + \epsilon_t. \end{aligned} \tag{11}$$

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 with 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) represents one of the latest techniques for volatility forecasting. Like 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 RVs, 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:

$$ExpRV_t^{CoM(\lambda)} = \sum_{i=1}^{500} \frac{e^{-i\lambda}}{e^{-\lambda} + e^{-2\lambda} + ... + e^{-500\lambda}} RV_{t-i+1}^d, \tag{12}$$

where λ 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, λ 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 RVs 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:

$$ExpGlRV_t^5 = \sum_{i=1}^{500} \frac{e^{-i\lambda}}{e^{-\lambda} + e^{-2\lambda} + ... + e^{-500\lambda}} GlRV_{t-i+1}, \tag{13}$$

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 long-run mean of RV, 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 period until day t. The resulting HExpGl model specification is given by:

$$\begin{aligned} RV_{t+1}^{t+h} = \beta_0 + \beta_1 ExpRV_t^1 + \beta_2 ExpRV_t^5 + \beta_3 ExpRV_t^{25} + \beta_4 ExpRV_t^{125} \\ + \beta_5 ExpGlRV_t^5 + \epsilon_t. \end{aligned} \tag{14}$$

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.[8] Because our forecasting horizon is up to three months, we include all 102 options from put and call options with maturities between one and three

[8] Christensen and Prabhala (1998) find that volatility implied by S&P 100 index option prices predicts ex-post realized volatility. Busch, Christensen, and Nielsen (2011) further show that implied volatility contains incremental information about future volatility across different asset classes.

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 δ ($\delta = 0.1, 0.15, ..., 0.9$). 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 δ ($\delta = -0.1, -0.15, ..., -0.9$).

3.7. Descriptive Statistics for Features

Table 1 provides the descriptive statistics for all realized features and selected implied variance features with absolute delta equal to 0.5.[9] Figure 3 plots the average implied variances 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.

In Table 2 we report pairwise correlations between features shown in Table 1. MIDAS features for various forecast horizons exhibit 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 weak correlations with other realized features, mostly because these features contain realized quarticities while other realized measures are all linear combinations of daily RVs. Interestingly, IV-based features CIVs and PIVs exhibit relatively weak correlations with all RV-based features, suggesting potentially new information contained in the IV-based features to the RV-based features.

4. Machine Learning Methodology

This section reviews the five machine learning algorithms we investigated 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).

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

4.1. LASSO

LASSO is designed to improve performance over that of OLS by imposing sparsity-encouraging penalties on regression coefficients for variance reduction and model interpretation. Take daily RV prediction as an example, LASSO assumes the same linear regression function as OLS:

$$g^*(z_{i,t};\theta) = z'_{i,t}\theta, \tag{15}$$

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

$$\mathcal{L}(\theta;\lambda) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}(RV^d_{i,t+1} - g^*(z_{i,t};\theta))^2 + \lambda\sum_{j=1}^{P}|\theta_j|, \tag{16}$$

where λ is the shrinkage parameter that controls for the magnitude of the penalty on the coefficients. The special case of $\lambda = 0$ collapses back to OLS. In such a case, LASSO/OLS minimizes the training (in-sample) error, potentially overfitting 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 component of the prediction error. Second, with zero regression coefficients, the fitted model becomes more interpretable.

It is important to consider several implementation details to achieve better performance with LASSO. First, we need to normalize features before estimating the models so that all features have comparable magnitudes. Otherwise, a single λ would have vastly different shrinkage effects on different features, making it impossible to tune. The normalization is done by using only 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 window scheme detailed in Section 4.7. Second, we need to choose λ from a wide range of values that can generate coefficient estimates with varying sparsity levels for the model selection procedure to choose from. Otherwise, the selected θ might be far from the region of optimal fit 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 volatility forecasting features 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 original 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 information redundancy. In the second step, we take only 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 follows:

$$RV = (Z\Omega_K)\theta_K + \tilde{E}, \tag{17}$$

where RV is the $NT \times 1$ vector of realized variances, Z is the $NT \times P$ matrix of features, Ω_K is a $P \times K$ orthogonal projection matrix from the P-dimensional original feature space onto the K-dimensional derived input space, θ_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 Ω_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 is kept and used to make predictions. Overfitting concerns naturally arise, however, 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 a 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:

$$\widehat{g^*}(z^*_{i,t};\theta,K,L) = \sum_{k=1}^{K} \theta_k 1_{\{z^*_{i,t} \in C_k(L)\}}, \tag{18}$$

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 θ_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 into one region, thus yielding zero training error but very poor out-of-sample performance.

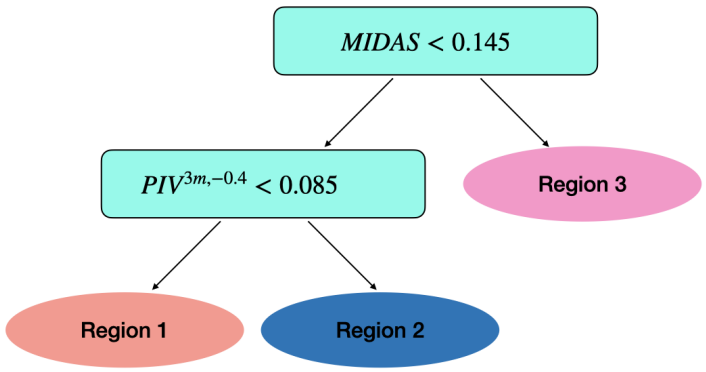


Fig. 1 Illustration of a regression tree model

RF reduces the overfitting problem associated with 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, RF 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 ranging 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 feature location and scale 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 uses the base learner of regression trees as RF. There are, however, two principal 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 using shallow trees because each tree is supposed to be weak, but by adding many small trees GBRT gradually reduces prediction bias while still controlling for variance. 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 discounting its contribution. Specifically, at every round after fitting a tree to the residuals, we update our $\widehat{g^*}(\cdot)$ by adding a shrunken 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 shrunken 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 the learning rate λ 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 prevents overfitting and as a result can balance the in-sample performance with the out-of-sample prediction.

In our implementation, we set the learning rate λ 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 give GBRT the ability to include high-order interactions and polynomials. For subsampling, we again use 50% of the training observations for each tree and $log(P)$ features for each split. In addition, we use early-stopping rules to help us choose the number of trees in the GBRT model: 1) If Mean Squared Errors (MSE) stop decreasing after 50 consecutive rounds, we set the number of trees as the round at which the MSE stops improving instead of including more trees in our GBRT model, and 2) when the total number of trees reaches 20,000. We report the resulting number of trees as model complexity. Like RF, GBRT is location- and scale-invariant so there is no feature standardization.

4.5. Feed-Forward Neural Network

Our third nonlinear model is the 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 final prediction. Transformations are called activation functions in NN and are the sources of nonlinearity.

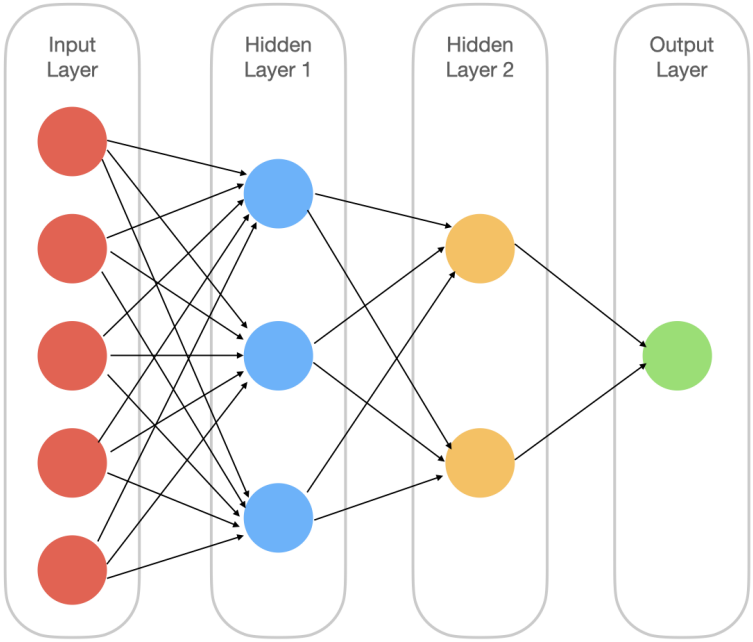


Fig. 2 Illustration of a feed-forward neural network model

In our implementation, we consider a model that has two hidden layers with 5 and 2 neurons,

respectively. For the activation function, we choose the commonly used rectified linear unit (ReLU) given as:

$$ReLU(x) = max(x, 0). \tag{19}$$

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 because NN is sensitive to feature scales. We also use multiple random states when implementing stochastic optimization for hyperparameters and derive predictions by averaging forecasts based on all tuned neural network models with varying starting points.

4.6. Ensemble Model

In addition to the aforementioned stand-alone learning algorithms, we consider an ensemble model that combines forecasts from several models. The intuition is that no single model is expected to dominate the others under any circumstances (Wolpert, 1996). Different models might do well in different scenarios and by combining them 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:

$$AVG = \frac{1}{5} \sum_{m=1}^{5} \widehat{g^*}^m(z_{i,t}^*). \tag{20}$$

4.7. Training and Validation

Machine learning algorithms include key hyperparameters that control for model complexity. We should tune these parameters based on the *prediction* error rather than the *training* error. Otherwise, learning algorithms, especially nonlinear algorithms, will overfit the training data and do poorly out of sample. Accordingly, we adopt a training-validation-testing scheme for model selection and assessment. Given the well-known commonalities in the dynamic dependencies of volatilities and spillover effects across assets, we purposely fit pooled models based on panel data in order to

increase estimation efficiency over stock-by-stock fitting.[10] 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 year t data, and a testing set consisting of year $t+1$ data. 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. For example, our first training sample contains four years of data from year 1996 to 1999, our first validation sample contains data in year 2000, and our first testing sample contains data in year 2001. This scheme leaves us with a total of 19 years of 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 from data inception to year t for training and data in year $t+1$ for testing. Thus, the overall testing sets are the same across models and differences in 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:

$$R^2_{OOS}(m) = 1 - \frac{\sum_{i,t}(RV_{i,t} - \widehat{RV_{i,t}}^{m})^2}{\sum_{i,t}(RV_{i,t} - \widehat{RV_{i,t}}^{benchmark})^2}, \tag{21}$$

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.[11] 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, which equals the expanding sample mean of RVs from the inception date 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 for beating the

[10] 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.

[11] 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 (fall below) the maximum (minimum) outcome value in the training sample with the observed maximum (minimum).

long-run mean is low because volatilities are persistent and time-varying. HAR is perhaps a better benchmark because it has shown 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 $d_{i,t} = (\widehat{e}^{(1)}_{i,t})^2 - (\widehat{e}^{(2)}_{i,t})^2$, where $\widehat{e}^{(1)}_{i,t}$ and $\widehat{e}^{(2)}_{i,t}$ are the prediction errors 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 and West (1987) standard error of d_t over the testing sample.

4.9. Feature Importance Metric

To shed additional light on how these features and learning algorithms work for volatility forecasting, we investigate how different features contribute to the prediction at various horizons. Our feature importance evaluation process differs from those in existing applications of machine learning to asset pricing. For example, Gu, Kelly, and Xiu (2020) compute the reduction in R^2 obtained by setting all values of one feature j to zero within each training set and then averaging the reductions over the training samples to obtain a per-feature importance measure. In our evaluation, we consider per-group feature importance instead of per-feature variable importance by assigning highly correlated features into one group and computing the importance of the entire group. The motivation is to avoid the dilution effect of per-feature variable importance. Consider the following simple example. Suppose two uncorrelated features X_1 and X_2 are equally important so they reduce R^2 equally by 0.5 from the joint model (X_1, X_2). Now suppose a new feature X_3 is introduced and X_3 is highly correlated with X_2 but not with X_1. If the model is estimated in a sensible way, then the variable importance of X_2 measured by its marginal reduction of R^2 will be diluted by X_3 because X_3 might serve as a proxy for X_2 in the model. The dilution phenomenon would be more pronounced when there are many more correlated variables. For our feature set, several subgroups are highly correlated as shown in Table 2. Because of the dilution effect, the per-feature variable importance measure might not truly reflect the importance of that feature. Therefore, we consider per-group feature importance to reduce cross-group correlations and also to reduce the number of

candidates for feature importance evaluation.

Second, instead of setting all values of the features within a group to zero, we consider random permutations of values across observations within the training set for the tested features as suggested by Fisher, Rudin, and Dominici (2019). This is because setting all feature values to zero introduces unintended bias to nonlinear models. To offer a simple example, suppose that we wish to test the marginal contribution of daily realized variance RV_t^d in RF fitting. If we simply set RV_t^d to zero, all observations will fall into one child node at each binary split that uses RV_d^d as the splitting variable, causing severe bias in the prediction. In contrast, permutation breaks the association between features and the true outcome, enabling us to remove the effects of the tested features. Specifically, for each training set and each feature group k, we permutate all values of each feature within group k and record the corresponding R^2. To reduce the permutation variance, we repeat the permutation five times and use average R^2 to compute the reduction in R^2. We then average the reductions in R^2 over different training samples to obtain a single group importance measure GI_k. We rank groups based on their GI_k such that the higher the rank the more important a feature group is.

5. Out-of-Sample Performance of Forecasting Models

In this section we show how machine learning can improve the volatility forecasting performance over that of traditional approaches. We begin by establishing the baseline performance by applying the traditional OLS method to *each* of the feature sets described in Section 3, as is commonly done in the literature. We then show that combining many RV-based features improves performance over that of any stand-alone feature set and that including the new IV-based features adds further value to RV-based features. After fixing the feature set, we demonstrate the benefits of more sophisticated learning algorithms in comparison with the baseline OLS. Finally, we show that an ensemble model combining many learning algorithms delivers extraordinary performance across all forecast horizons.

5.1. OLS-Based Models

In Table 3 we report the out-of-sample performance of OLS-based volatility forecasting models based on the R^2_{OOS} relative to HAR from Eq. (21).[12] The first column lists the model names

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

and the second column summarizes their features. First, we focus on the four popular RV-based models. Among them, MIDAS, SHAR, and HARQ-F outperform HAR across all forecast horizons, as is evident by the positive relative R^2_{OOS}'s. HExpGl outperforms HAR at the daily, weekly, and monthly horizons, while slightly underperforms HAR at the quarterly horizon.

Next, we combine all 16 realized measures from the MIDAS, SHAR, HARQ-F, and HExpGl models through OLS.[13] This model, namely OLS^{RM}, not only outperforms HAR by wide margins across all horizons, but it also generally beats individual models in most cases. 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 (IVs) 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}'s. Yet this does not mean that IV features are useless in the presence of realized features. Although IVs are weakly informative as stand-alone features, they can still add value as long as they contain information that is orthogonal to the realized features. To test whether there is any additional value gained from 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 labeled 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. At the quarterly horizon, however, the relative R^2_{OOS} remains negative at -0.6%, which is worse than several individual RV-based models. The result might reflect the fact that, at the quarterly horizon, effective sample size drops significantly and thus we do not have enough data to estimate a dense OLS model with 118 features.[14] In such a case, we may need sparse or more regularized models. Another point worth noting is that OLS^{ALL} outperforms OLS^{RM} at the first three horizons, indicating the additional information contained in IV measures.

[13] 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. (5) and (6).

[14] Our forecasting models are designed to use as much data as possible by fitting daily updated RVs on daily updated features for all forecast horizons. Because of overlapping data, however, the effective sample size of the data at the quarterly horizon is only about 1/63 of the sample size at the daily horizon.

5.2. Machine-Learning-Based Models

Having established the initial evidence that increasing the number of features can improve forecast performance through a simple OLS fit, we now show that performance can be further improved by using learning algorithms other than OLS. Table 4 presents the R^2_{OOS}'s relative to HAR for the five learning algorithms discussed in Section 4: LASSO, Principal Component Regression (PCR), Random Forecast (RF), Gradient Boosted Regression Trees (GBRT), and Neural Network (NN), and for an ensemble model based on the five individual machine learning models (AVG).[15] 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 in the last column we report 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 the 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 "LASSO" and "PCR" in Table 4 presents their R^2_{OOS}'s relative to HAR.[16] The sparsity-encouraging $LASSO$ model has higher relative R^2_{OOS}'s 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 the 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 the daily forecast horizon and at 6.4% for the

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

[16] For LASSO, we validate its shrinkage parameter λ 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.

weekly forecast horizon, both of which are below the corresponding metrics of OLS^{ALL}. However, RF outperforms OLS^{ALL} at the monthly and quarterly forecast horizons with relative R^2_{OOS}'s 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, and the maximum tree depth can take any integer value between 1 and 5. For the remaining 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 the daily and weekly forecast horizons with relative R^2_{OOS}'s equal to 4.7% and 10.2%, but significantly outperforms OLS^{ALL} at the monthly and quarterly horizons with relative R^2_{OOS} equal to 10.8% and 6.3%. To train NN, we consider two hidden layers with five and two neurons, respectively. We choose the popular rectified linear unit (ReLU) as the activation function. In general, NN performs fairly well with relative R^2_{OOS} equal to 10.5%, 16.7%, 14.3%, and 4.8% at the daily, weekly, monthly, and quarterly forecast horizons, respectively.

An ensemble model: Comparing the out-of-sample performance of the five learning algorithms, we find that no single model strictly dominates the others. We then consider an ensemble model that combines volatility forecasts from different models.[17] 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 extraordinary. This average model outperforms the first four individual machine learning models at each forecast horizon by a significant margin. Although the performance of AVG is comparable to or slightly weaker than that of NN at daily to monthly horizons, it significantly dominates NN at the quarterly horizon with an improvement in R^2_{OOS} relative to HAR of more than 5%. Overall, the relative R^2_{OOS} of AVG ranges from 9.0% at the daily forecast horizon to up to 15.2% at the monthly forecast horizon, further highlighting the advantage of combining machine learning models in forecasting RVs.

[17] See Timmermann (2006) for an extensive survey of forecast combination.

5.3. Model Complexity

To help generate insights into 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 that 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 the 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. Here we observe that, like LASSO, PCR tends to favor more components in shorter-horizon forecasts (daily and weekly) than longer-horizon forecasts (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 reflects our choice of a small learning rate λ of 0.001, which requires a large value of B for GBRT to converge. Note that imposing a boundary on 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 the quarterly forecast horizon), and the overall out-of-sample performance of GBRT indicated in Table 4 is comparable to that of other machine learning models.

5.4. Model Comparison

The early results reported in Table 4 reveal that different machine learning models have varying strengths over different horizons, and the simple average of all individual machine learning models performs quite well across horizons. To better understand how the out-of-sample forecasts from various models are related to each other, we report pairwise correlations of forecasts between the five machine learning models and OLS^{ALL} in Table 5. The correlations are strong, ranging from 0.909 between RF and OLS^{ALL} at the quarterly forecast horizon, to 0.997 between LASSO and OLS^{ALL} at the daily forecast horizon. This is not surprising because 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 high signal-to-noise ratios. Consequently, a large part of

each model's prediction may well capture the easy-to-predict part of the signal, inducing strong correlations among them.[18] Looking across forecast horizons, we see that the pairwise correlations between forecasts tend to decrease as the forecast horizon increases. For instance, the correlation between AVG and OLS^{ALL} decreases monotonically from 0.988 at the daily forecast horizon to 0.966 at the quarterly forecast horizon.

Given the strong correlation of forecasts between models, we are interested in formally assessing whether the differences in the out-of-sample performance between models are statistically significant at all. In Table 6 we report the Diebold-Mariano (DM) t-statistics for pairwise comparisons of performance for a model in the row versus a model in the column. The DM statistics are distributed $\mathcal{N}(0,1)$ 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 that the row model outperforms the column model; *, **, and *** denote significance at the 10%, 5%, and 1% levels, respectively. At the shorter daily and weekly forecast horizons, we find that the majority of the t-statistics are significant at the 5% level, indicating that a strong correlation between forecasts at these two horizons does not necessarily translate into an insignificant difference in out-of-sample performance. At the monthly forecast horizon, however, the t-statistics comparing the out-of-sample performance between OLS^{ALL} and the first four individual machine learning algorithms are generally insignificant. NN and AVG, on the other hand, significantly outperform OLS^{ALL} and the other individual machine learning algorithms in most cases. At the quarterly horizon, four out of the five individual machine learning models significantly outperform OLS^{ALL}. It is worth noting that the ensemble model AVG not only outperforms OLS^{ALL} at the 1% level, but also significantly dominates each individual machine learning algorithm except for PCR. Mirroring the results reported in Table 4, the relative strength of each stand-alone machine learning model and OLS^{ALL} may depend on the forecast horizon. In sharp contrast, the simple average of all machine learning models AVG performs very well across all forecast horizons at the 1% or 5% significance level in most cases.

[18] 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 forecast horizons, as shown in Tables A.3 and A.4 in the Appendix.

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 2008-2009 financial crisis disrupt the relation between features and future RVs fitted using historical data?[19] 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. In Table 7 we summarize 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}'s between 5.2% and 8.5%; at the quarterly forecast horizon, HARQ-F has the highest relative R^2_{OOS} at 6.7%. Turning to the ML-based models, AVG exhibits superior performance across all forecast horizons, with relative R^2_{OOS}'s ranging from 6.6% to 20.5%.

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 remaining OLS-based models at the daily and weekly forecast horizons, whereas MIDAS dominates at the monthly horizon and HARQ-F beats other OLS-based models at the quarterly horizon. Among ML-based models, NN performs the best at the daily forecast horizon with a relative R^2_{OOS} equal to 13.4%, and it performs on par with LASSO at the weekly horizon with a relative R^2_{OOS} equal to 14.9%. LASSO dominates at the monthly horizon with a relative R^2_{OOS} of 9.5%. At the longer quarterly horizon, PCR achieves the highest relative R^2_{OOS} equal to 5.7%. The overall winning model remains, however, the ensemble model AVG. Although AVG cannot beat all of the stand-alone models at a given forecast horizon, it consistently delivers top performance across 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} during this period is quite impressive with relative R^2_{OOS}'s equal to 7.5%, 13.5%, 20.2%, and 15.5% at the daily, weekly,

[19] Schwert (2011) suggests that the financial crisis was associated with high levels of stock market volatility, but volatility exhibits more mean-reversion than in the past.

monthly, and quarterly horizons. In contrast, the relative R^2_{OOS}'s of the popular RV forecasting models are all below 6%. A natural question is, what explains the stellar performance of OLS^{ALL}? 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 for stock options has increased steadily over the past two decades, implying that the overall option market is becoming more efficient in incorporating information about future price movements. To mention 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 sheds additional light on the importance of including implied variance predictors in forecasting RVs. 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%, 18.7%, 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. Utility Gains

We have demonstrated the statistical improvement in terms of relative R^2_{OOS} achieved by using many features and machine learning algorithms for forecasting RV. A natural question is then to what extent the increase in relative R^2_{OOS}'s can translate into economic gains. In this section, we gauge the utility gains of a mean-variance investor who invests in a risky asset with time-varying volatility and a constant Sharpe ratio.

6.1. Framework

Following the framework of Bollerslev, Hood, Huss, and Pedersen (2018), we assume that the investor's expected utility at time t under mean-variance utility can be expressed as $E_t(u(W_{t+1})) = E_t(W_{t+1}) - \frac{1}{2}\gamma^A Var_t(W_{t+1})$, where γ^A reflects the investor's absolute risk aversion. Assuming that the investor allocates a ω_t fraction of his wealth to the risky asset with return r_{t+1} and the remaining fraction to the risk-free asset with return r^f_t, his wealth in period $t+1$ is $W_{t+1} = W_t(1+r^f_t+\omega_t r^e_{t+1})$,

where $r^e_{t+1} \equiv r_{t+1} - r^f_t$ is the excess return. After dropping constant terms that depend only on time t variables, the expected utility is:

$$U(\omega_t) = W_t \left(\omega_t E_t(r^e_{t+1}) - \frac{\gamma}{2}\omega_t^2 Var_t(r^e_{t+1}) \right) = W_t \left(\omega_t E_t(r^e_{t+1}) - \frac{\gamma}{2}\omega_t^2 E_t(RV_{t+1}) \right), \tag{22}$$

where $\gamma \equiv \gamma^A W_t$ denotes the investor's relative risk aversion. Assuming a constant conditional Sharpe ratio $SR \equiv E_t(r^e_{t+1})/\sqrt{E_t(RV_{t+1})} = 0.4$ and $\gamma = 2$, Bollerslev, Hood, Huss, and Pedersen (2018) show that the expected utility per unit of wealth (after dropping constant terms) under model θ can be expressed as:

$$U(\omega_t^\theta)/W_t = 8\% \frac{\sqrt{E_t(RV_{t+1})}}{\sqrt{E_t^\theta(RV_{t+1})}} - 4\% \frac{E_t(RV_{t+1})}{E_t^\theta(RV_{t+1})}. \tag{23}$$

Replacing $E_t(RV_{t+1})$ with the true and observed RV_{t+1}, we obtain the realization of the expected utility per unit of wealth, referred to as the realized utility:

$$RU(\omega_t^\theta) = U(\omega_t^\theta)/W_t = 8\% \frac{\sqrt{RV_{t+1}}}{\sqrt{E_t^\theta(RV_{t+1})}} - 4\% \frac{RV_{t+1}}{E_t^\theta(RV_{t+1})}. \tag{24}$$

The upper bound of the average realized utility over stocks and time is 4% when the risk model perfectly predicts the realized RV in each period, or $E_t^\theta(RV_{t+1}) = RV_{t+1}$. To put the 4% figure into perspective, from Eq. (22), $U(0) = 0$ when $\omega_t = 0$ or the investor allocates 100% of his wealth to the risk-free asset. Thus, the investor is willing to give up 4% of his wealth to have access to a perfect risk model instead of investing only in the risk-free asset.

In addition to the above frictionless setting, we consider the cost of implementing the risk-targeted positions. As in Bollerslev, Hood, Huss, and Pedersen (2018), we focus on the monthly forecast horizon and assume that transaction costs (TC) are linear in the absolute magnitude of the change in the positions. For the benchmark transaction cost estimate, we use the median bid-ask spread for each stock over the last 90 trading days of the sample. Note that one-way transactions are often assumed in the literature to cost half of the bid-ask spread. We use the full spread as Bollerslev, Hood, Huss, and Pedersen (2018) to account for the market-impact component of the transaction cost. The realized utility after TC is simply the difference between the realized utility and the transaction cost. We take the average of the realized utilities before/after TC over stocks and the

testing sample to obtain the final average numbers.

6.2. Results

In Panel A of Table 8 we summarize the average realized utilities before TC based on risk models OLS^{ALL} and AVG. The column labeled "All stocks" reports the results for all 173 stocks in our main sample. OLS^{ALL} and AVG deliver average realized utilities of 3.47% and 3.71%, respectively, with a utility difference of 0.24% reported in the third row. To further assess the utility gain of AVG over OLS^{ALL} among stocks with varying levels of volatility, we sort the 173 stocks into quintiles based on their median monthly RV over the testing sample, and compute the average realized utility for each quintile of stocks. The utility difference between the two models for the lowest volatility quintile is 0.19% and that for the highest volatility quintile is 0.34%, indicating that the utility gained by using AVG over OLS^{ALL} is more pronounced for high-volatility stocks. The last row reports the DM t-statistics contrasting AVG with OLS^{ALL}, all of which are significant at the 1% level for various stock universes. The average realized utilities after TC reported in Panel B exhibit similar patterns: AVG generates significantly higher realized utility over OLS^{ALL} across various stock samples and the gain is more remarkable for more volatile stocks.

Another important factor that can be used to gauge the economic significance is the performance of a risk model during turbulent periods when volatility forecasting is more challenging. In Figure 5, we present a scatter plot of the average realized utilities per calendar month obtained using AVG and OLS^{ALL}. Most of the data points are above the 45-degree line, indicating that AVG typically generates higher realized utility than OLS^{ALL}. While both models produce realized utility close to the 4% maximum during the majority of the periods, the utility gained by using AVG as opposed to OLS^{ALL} is more pronounced when OLS^{ALL} generates utility that is considerably below 4% (i.e., the periods when volatility is hard to predict).

7. Feature Importance

7.1. Which Features Matter?

We rely on the group importance measure described in Section 4.9 to assess the importance of each group of features for forecasting future RVs while simultaneously controlling for the rest of

the feature groups. Specifically, we divide the 118 features into three groups. The first group "MIDAS & ExpRV" includes the $MIDAS$ feature, $ExpRV^1$, $ExpRV^5$, $ExpRV^{25}$, $ExpRV^{125}$, and $ExpGlRV$, all of which are smoothly weighted sums of lagged daily RV's. The second group "RV& RQ" includes RV^d, RV^w, RV^m, RV^q, RVP^d, RVN^d, $RV^d\sqrt{RQ^d}$, $RV^w\sqrt{RQ^w}$, $RV^m\sqrt{RQ^m}$, and $RV^q\sqrt{RQ^q}$, all of which are based on simple RV and/or RQ terms. And the third group "Implied Variance" includes 102 implied variance features. For each forecast horizon and each model, we estimate the reduction in R^2 from permutating all values of a feature group within each training sample, and then average the reductions in R^2 over all training samples to obtain a single group importance measure. As our group importance measures for each forecast horizon are normalized to sum to one, we can interpret the importance measure of each group as its relative contribution to the overall importance in percentage.

Panels A to F in Figure 6 display the group importance across various forecast horizons for LASSO, PCR, RF, GBRT, NN, and AVG, respectively. This figure reveals several interesting findings. First, all three groups of features contribute significantly to the forecast across different horizons and learning algorithms. For example, each feature group contributes at least 10% in almost all settings with the only exception being "RV& RQ" for fitting LASSO at monthly and quarterly horizons. Second, the MIDAS and ExpRV terms tend to be more important for LASSO, jointly contributing around 70% to the overall prediction. This might be because the MIDAS and ExpRV features are already well-engineered through smoothing and denoising the raw data. Unlike the raw features such as IVs, $MIDAS$ and $ExpRV$ terms can be viewed as competent encoders that represent the predictive structure in the data and thus are more likely to be directly picked up by linear models. Third, the implied variance features become more important over forecasting horizons. Since IV-based features all have maturities between one and three months, it may not be surprising that they can better predict longer-term RVs.

To further assess the relative importance of each feature group over time, we focus on AVG and report its group importance based on 118 features across forecast horizons for each training sample in our out-of-sample analyses. Our first training sample is from January 1996 to December 1999, and our last training sample is from January 1996 to December 2017. By the end of each training sample, we calculate group importance based on the reduction in R^2 from permutating all values of a given group of features within that training sample, and then normalize group importance

per each training sample and each forecast horizon to sum to one. Figure 7 displays the group importance for each training sample. Overall, implied variance features grow increasingly important over time for all forecast horizons. At the daily (weekly) forecast horizon, the group importance of "IV" terms increases from 13.52% (14.56%) for the first training sample to 49.14% (55.92%) for the last training sample, and the trend is similar at longer monthly and quarterly forecast horizons. In the meanwhile, the importance of the other two feature groups shrinks significantly over time. For example, at the monthly forecast horizon, the importance of "MIDAS & ExpRV" terms decreases from 60% by the end of year 1999 to 25.86% by the end of year 2017, and that of "RV& RQ" terms decreases from 23.5% to 15.39%. These observations are consistent with the stellar performance of the pure IV-based OLS model during the most recent period reported in Panel C of Table 7, and further highlight the crucial role implied variance features played in forecasting realized variances.

7.2. Firm Characteristics and Pure Noise Features

Our 118 features are all volatility-based features and, given the persistence of volatility, they are naturally strong predictors of future RVs. One may wonder about how our new system can handle other features such as weak features or even pure noise features. To address these questions, we consider two new feature sets: firm characteristics and pure random noises. In the volatility forecasting literature, firm characteristics have not been widely documented as useful predictors of future realized volatility. On the other hand, it might be reasonable to hypothesize that firm characteristics such as size might be indirectly (through interaction or nonlinearity) helpful in volatility forecasting. To examine firm characteristics, we consider six features: $Size$, BM, Mom, Ret^d, Ret^m, and $ILLQ^m$. $Size$ is the product of the closing price and the number of shares outstanding, updated each day. BM is the book-to-market ratio in June of year t, which is computed as the ratio of the book value of common equity in fiscal year $t-1$ to the market value of equity in December of year $t-1$. Mom is the cumulative returns from prior day 252 to day 21 for a given day t. Ret^d and Ret^m refer to past daily and 21-day returns. $ILLQ^m$ is the illiquidity measure of Amihud (2002), which is the average daily ratio of the absolute stock return to the dollar trading volume over the past 21 days.

The second new feature set is pure noise, with which we can test how well our system handles false positives. We generate six random noise terms that mimic the distributional properties of the

volatility-based features. Let $r_{i,j,t}$ denote the j-th noise term for stock i on day t. We simulate the panel of noises for each $1 \leq i \leq N$ and each $1 \leq j \leq 6$ from the following model:

$$r_{i,j,t} = 0.2(1-\rho_j) + \rho_j r_{i,j,t-1} + u_{i,j,t}, \quad u_{i,j,t} \sim \mathcal{N}(0, 0.25^2(1-\rho_j^2)), \tag{25}$$

where $\rho_j \in \{0.2, 0.4, 0.6, 0.8, 0.9, 0.99\}$ is the first-order autocorrelation of noise j. By construction, each noise term will have a mean of 0.2 and a standard deviation of 0.25, and they cover a wide range of persistence levels.

Panel B of Table 9 presents the R^2_{OOS}'s relative to HAR for fitting OLS^{ALL} and the ML-based models to the newly expanded set of 130 features. In Panel A of the same table we replicate the results reported in Table 4 using the original 118 features for ease of comparison. Overall, the augmented feature set generates very similar results to these using the original 118 features across different models. For OLS^{ALL}, the additional features slightly improve the relative R^2_{OOS}'s at daily, weekly, and monthly horizons, but reduce the relative R^2_{OOS} at the quarterly horizon as a result of overfitting more predictors. For LASSO, PCR, and RF, the average performance of each model over forecast horizons stays about the same using either 118 or 130 features. For GBRT, the R^2_{OOS}'s based on 130 features are higher across all horizons, but the improvement over the original feature set is rather small, ranging from 0.2% at the quarterly horizon to 0.9% at the weekly horizon. For NN, the additional features produce higher relative R^2_{OOS}'s at the daily and weekly horizons, but deliver worse performance at the monthly and quarterly horizons. For the ensemble model AVG, the additional features show minimal improvement in the relative R^2_{OOS}'s at the first three horizons, and identical relative R^2_{OOS}'s at the quarterly horizon.

Figure 8 displays the group importance plots based on all 130 features for each individual ML model and the ensemble model AVG. In addition to the three groups of features from the original 118-feature set, we include two new groups, "Firm Char" and "Noise," each of which contains six firm characteristics and six pure noise terms, respectively. There are several intriguing observations. First, the importance of the first three groups is largely aligned with what Figure 6 shows based on 118 features. Secondly, firm characteristics as a group contribute modestly to RV prediction, with group importance ranging from 0.31% for LASSO at the quarterly horizon to 4.2% for NN at the same quarterly horizon. Across models, the contribution of firm characteristics is relatively greater

for nonlinear models RF, GBRT, and NN at around 3% over various forecast horizons compared with around 1% for linear models LASSO and PCR. One possible explanation is that firm characteristics can help predict future RV only as interaction terms with existing RV predictors. Lastly, the noise features contribute almost nothing to model prediction, indicating that our ML-based models and the associated group importance metrics effectively control for false positives.

8. Predicting Realized Variances for S&P 500 Stocks

We have demonstrated that our machine-learning-based automatic system can improve volatility forecasting performance both statistically and economically. Is the learning system scalable to more stocks? In this section, we examine the out-of-sample performance of our system on a broader set of S&P 500 stocks. To speed up fitting nonlinear models, we transfer tuning parameters already learned from the original S&P 100 stock universe to the new S&P 500 universe. We find that tuning parameters learned from the original stock sample transfer well to the new sample and the resulting automatic system consistently generates significant gains.

To this end, we consider 663 unique stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 500 index but are not members of the S&P 100 index between January 1996 and June 2019, and apply the same data filters as described in Section 2.1 to this stock sample. Because hyperparameter tuning for nonlinear models becomes more time-consuming as the sample grows, we directly transfer the tuning parameters for RF (i.e., maximum tree depth) and GBRT (i.e., # of trees and maximum tree depth) obtained from 173 S&P 100 stocks to 663 S&P 500 stocks, and retrain both models without validating these tuning parameters.[20] The idea is inspired by transfer learning, which is designed to explore the possibility that learned knowledge from one sample can be applied to a new sample.[21] The remaining ML-based as well as OLS-based models can be estimated efficiently and thus are completely recalibrated using the S&P 500 stock sample without hyperparameter transfer.

Table 10 summarizes the out-of-sample performance of all models for 663 S&P 500 stocks. Among OLS-based models, OLS^{ALL} using all 118 predictors outperforms the remaining models at

[20] All hyperparameters for NN are pre-specified and thus do not require hyperparameter tuning.

[21] Jiang, Kelly, and Xiu (2021b) apply the image-based convolutional neural networks (CNNs) trained using daily data to lower-frequency and international data for return prediction problems. See Pan and Yang (2009) for a comprehensive survey on transfer learning.

daily, weekly, and monthly forecast horizons with R^2_{OOS}'s relative to HAR between 4.9% and 8.6%. OLS^{ALL} slightly underperforms HARQ-F at the quarterly forecast horizon but beats the rest of the OLS-based models across horizons, indicating that the 118 features identified earlier remain powerful volatility predictors for this broader universe. Note that the relative R^2_{OOS}'s of OLS^{IV} become more negative between −13.6% and −20.3% for the S&P 500 stock sample in contrast to the relative R^2_{OOS}'s between −2.1% and −9.8% for the S&P 100 sample as reported in Table 4. This is likely because S&P 500 stocks tend to have fewer liquid option contracts than S&P 100 stocks and the associated implied variance features are prone to measurement errors and biases. Yet, we find that implied variance features for S&P 500 stocks still contain information orthogonal to the realized features, as evident by the better performance of OLS^{ALL} over that of OLS^{RM} at each forecast horizon.

Turning to the ML-based models, LASSO outperforms OLS^{ALL} across all horizons by small margins, while NN outperforms OLS^{ALL} by wide margins with relative R^2_{OOS}'s ranging from 4.3% to 15.1%. The other three ML models PCR, RF, and GBRT produce out-of-sample performance comparable to OLS^{ALL}, indicating the success of hyperparameter transfer for the latter two models. For the ensemble model AVG, it consistently delivers higher relative R^2_{OOS} than OLS^{ALL} across forecast horizons, and the pairwise Diebold-Mariano t-statistics comparing the out-of-sample forecast performance between AVG and OLS^{ALL} are all significant at the 1% level. Moreover, we find that the average realized utility derived from AVG is 0.44% higher than that from OLS^{ALL} for the S&P 500 stocks, in comparison to the 0.24% utility difference between AVG and OLS^{ALL} for the S&P 100 stocks as reported in Table 8. In a nutshell, our automatic system continues to perform well on this broader S&P 500 stock universe.

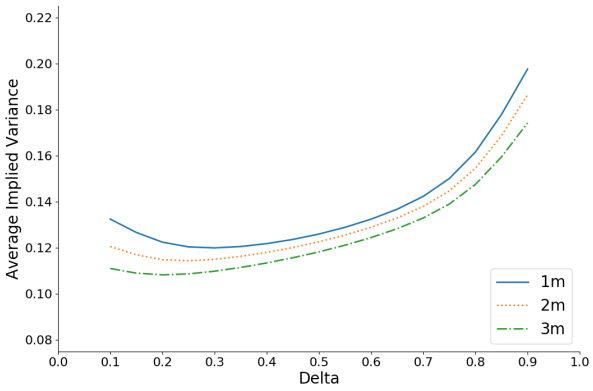
9. Conclusion

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

requires little human intervention for choosing predictors and models. Using 118 features and five machine learning algorithms (LASSO, PCR, RF, GBRT, and NN) to forecast realized volatility for 173 S&P 100 stocks spanning two decades, we show that the automatic system delivers robust and superior performance across forecasting horizons and time periods. A simple average ensemble model combining all machine learning algorithms produces out-of-sample R^2's relative to HAR model predictions equal to 9.0%, 14.3%, 15.2%, and 10.0% at daily, weekly, monthly, and quarterly forecast horizons over the full sample. In the most recent decade, the corresponding out-of-sample R^2's relative to HAR jump to 10.4%, 18.7%, 29.6%, and 27.5%, indicating the importance of forecasting volatility via such a powerful automatic system in the next decade.

Based on a utility framework, we further demonstrate that the improvement in out-of-sample prediction accuracy can translate into nontrivial economic gains for a mean-variance investor. Among all features we study, implied variance features increasingly contribute more to RV prediction in the recent period, perhaps due to their improved quality as the overall option market becomes more liquid. After augmenting our feature set by including six firm characteristics and six pure noise terms, we find that firm characteristics that are important in return prediction do not contribute much to volatility forecasting and the importance of noise terms is nearly zero, indicating that our system effectively controls for false positives. Lastly, we show that tuning parameters learned from S&P 100 stocks transfer well to a different set of S&P 500 stocks and our learning system is scalable to this broader universe.

Panel A: Implied variance from call options



Panel B: Implied variance from put options

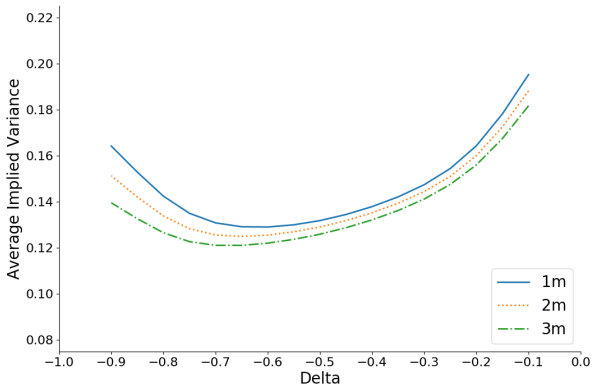


Fig. 3 Average implied variance from call and put options

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 variances from call (put) options with 0.05 increment.

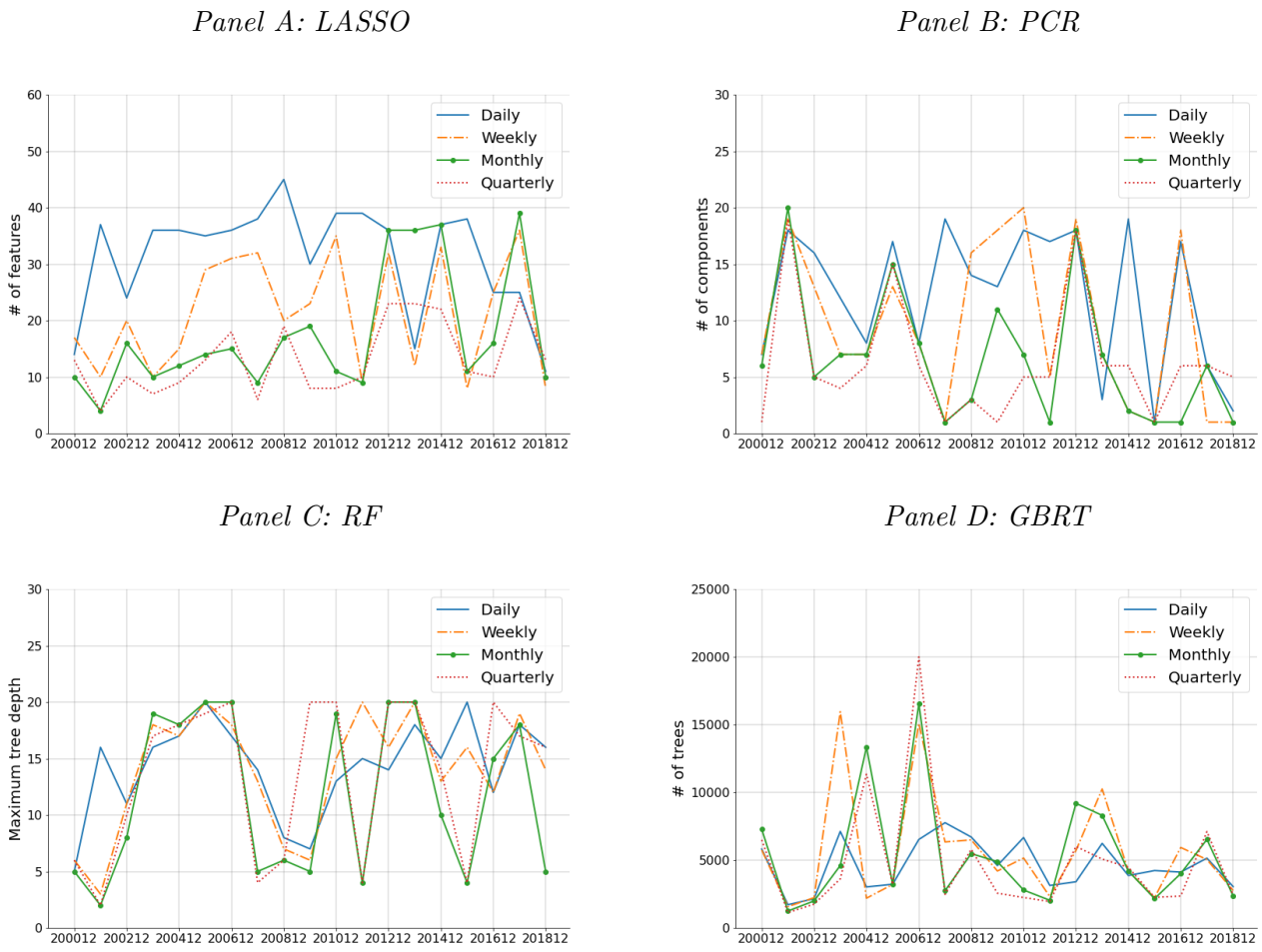


Fig. 4 Model complexity over time

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 across various 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.

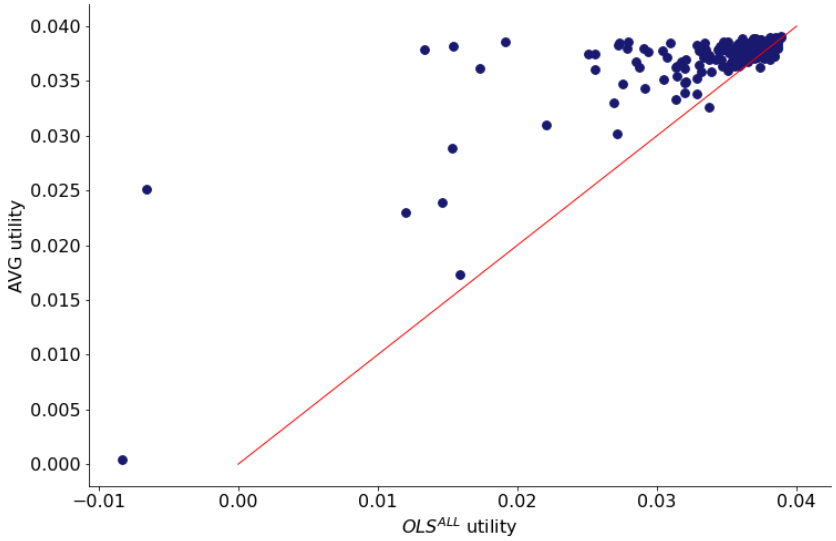
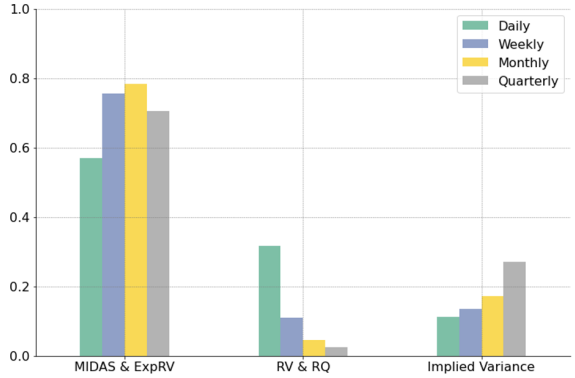


Fig. 5 Realized utility: AVG versus OLS^{ALL}

This figure shows the average realized utilities per calendar month obtained using AVG on the y-axis and OLS^{ALL} on the x-axis. AVG denotes the simple average of five individual machine learning models, and OLS^{ALL} denotes the simple OLS model with all 118 realized and implied variance features as joint predictors. Realized utilities are calculated without transaction costs according to Eq. (24).

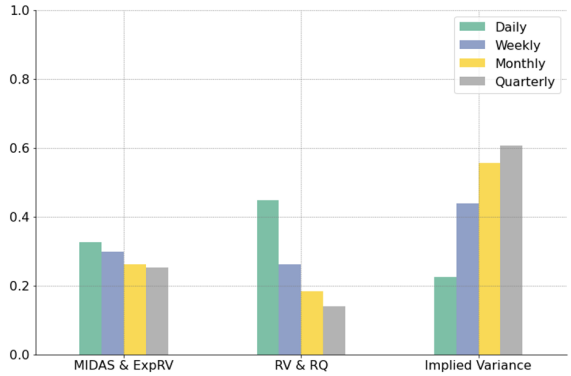
Panel A: LASSO

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	56.93%	75.60%	78.35%	70.49%
RV & RQ	31.79%	10.90%	4.56%	2.45%
Implied Variance	11.28%	13.50%	17.09%	27.06%



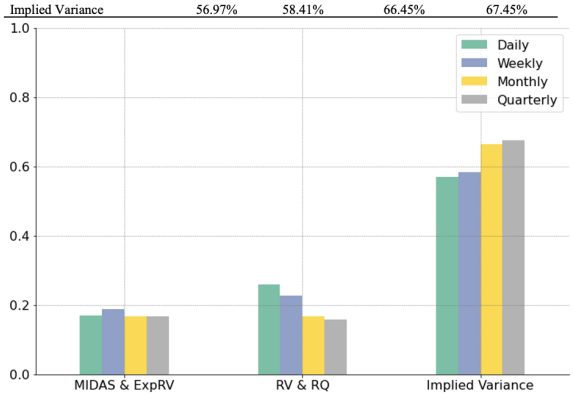
Panel B: PCR

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	32.56%	29.83%	26.10%	25.25%
RV & RQ	44.84%	26.23%	18.41%	14.04%
Implied Variance	22.60%	43.94%	55.49%	60.71%



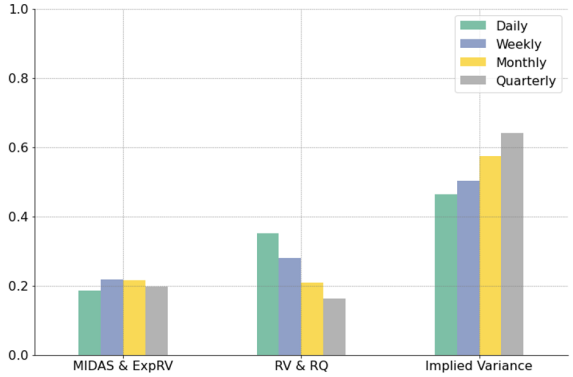
Panel C: RF

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	17.04%	18.85%	16.71%	16.69%
RV & RQ	26.00%	22.74%	16.83%	15.87%
Implied Variance	56.97%	58.41%	66.45%	67.45%



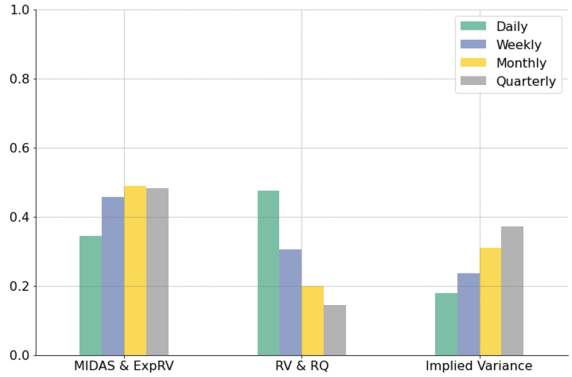
Panel D: GBRT

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	18.62%	21.70%	21.62%	19.66%
RV & RQ	35.08%	28.00%	20.93%	16.32%
Implied Variance	46.30%	50.31%	57.45%	64.03%



Panel E: NN

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	34.41%	45.75%	49.02%	48.31%
RV & RQ	47.61%	30.59%	20.03%	14.40%
Implied Variance	17.98%	23.66%	30.95%	37.29%



Panel F: AVG

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	31.71%	38.41%	38.23%	33.97%
RV & RQ	36.25%	23.57%	15.62%	13.03%
Implied Variance	32.04%	38.02%	46.15%	53.00%

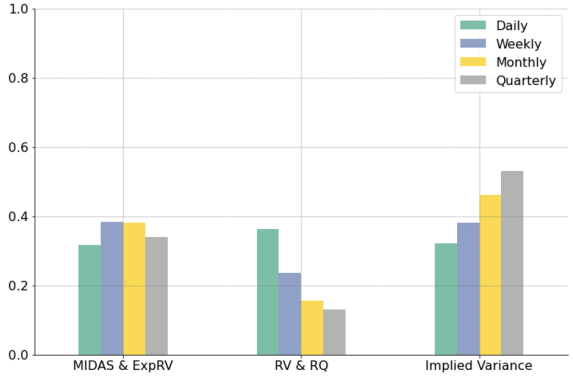


Fig. 6 Group importance based on 118 Features

This figure displays the group importance based on 118 features for LASSO, PCR, RF, GBRT, NN, and AVG across various forecast horizons. The first group "MIDAS & ExpRV" includes the $MIDAS$ term for the corresponding forecast horizon, $ExpRV^1$, $ExpRV^5$, $ExpRV^{25}$, $ExpRV^{125}$, and $ExpGlRV$. The second group "RV& RQ" includes RV^d, RV^w, RV^m, RV^q, RVP^d, RVN^d, $RV^d\sqrt{RQ^d}$, $RV^w\sqrt{RQ^w}$, $RV^m\sqrt{RQ^m}$, and $RV^q\sqrt{RQ^q}$. The third group "Implied Variance" includes $CIV^{jm,\delta}$ and $PIV^{jm,-\delta}$, where $j = 1, 2, 3$, and $\delta = 0.1, 0.15, ..., 0.9$. To calculate group importance, we first compute the reduction in R^2 from permutating all values of a given group of features within each training sample, and then average the reductions of R^2 over all training samples to obtain a single group importance measure. Group importance for each forecast horizon is normalized to sum to one.

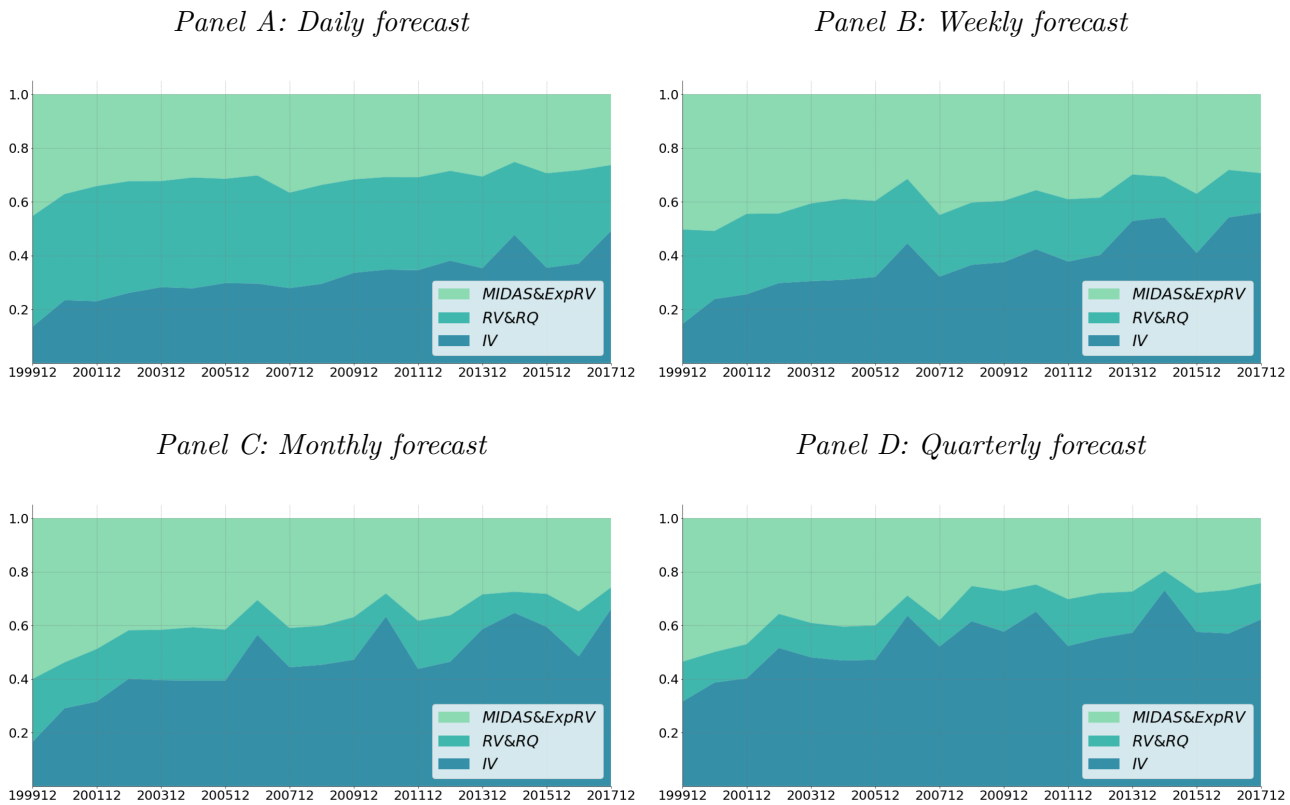
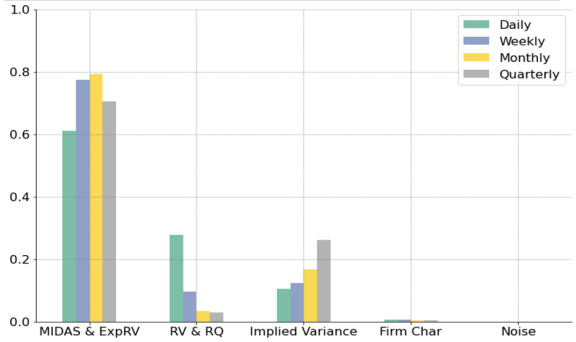


Fig. 7 Group importance for AVG over time

This figure displays the group importance based on 118 features for AVG across forecast horizons for each training sample in our out-of-sample analyses. Our first training sample is from January 1996 to December 1999, and our last training sample is from January 1996 to December 2017. The first group "MIDAS & ExpRV" includes the $MIDAS$ term for the corresponding forecast horizon, $ExpRV^1$, $ExpRV^5$, $ExpRV^{25}$, $ExpRV^{125}$, and $ExpGlRV$. The second group "RV& RQ" includes RV^d, RV^w, RV^m, RV^q, RVP^d, RVN^d, $RV^d\sqrt{RQ^d}$, $RV^w\sqrt{RQ^w}$, $RV^m\sqrt{RQ^m}$, and $RV^q\sqrt{RQ^q}$. The third group "Implied Variance" includes $CIV^{jm,\delta}$ and $PIV^{jm,-\delta}$, where $j = 1, 2, 3$, and $\delta = 0.1, 0.15, ..., 0.9$. By the end of each training sample, we calculate group importance based on the reduction in R^2 from permutating all values of a given group of features within that training sample. Group importance per each training sample and each forecast horizon is normalized to sum to one.

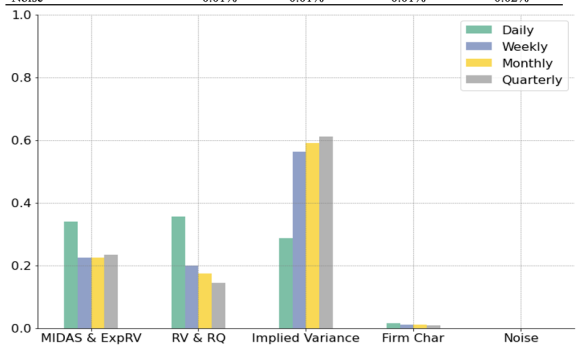
Panel A: LASSO

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	61.12%	77.42%	79.29%	70.52%
RV & RQ	27.72%	9.52%	3.51%	2.91%
Implied Variance	10.53%	12.47%	16.79%	26.26%
Firm Char	0.63%	0.59%	0.41%	0.31%
Noise	0.00%	0.00%	0.00%	0.00%



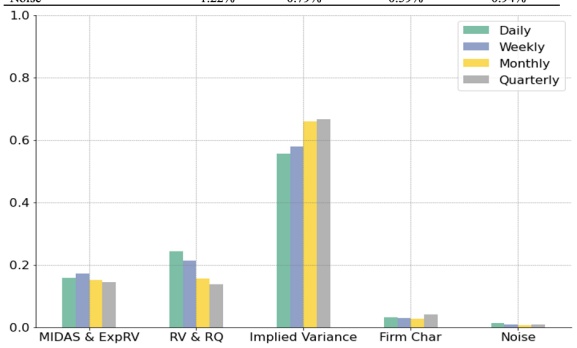
Panel B: PCR

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	34.00%	22.51%	22.52%	23.38%
RV & RQ	35.63%	20.01%	17.42%	14.52%
Implied Variance	28.76%	56.31%	59.06%	61.18%
Firm Char	1.61%	1.17%	0.99%	0.90%
Noise	0.01%	0.01%	0.01%	0.02%



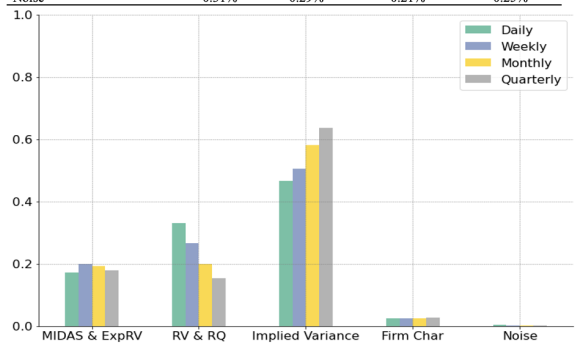
Panel C: RF

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	15.74%	17.15%	15.13%	14.55%
RV & RQ	24.42%	21.29%	15.63%	13.65%
Implied Variance	55.51%	57.90%	65.88%	66.68%
Firm Char	3.11%	2.87%	2.77%	4.18%
Noise	1.22%	0.79%	0.59%	0.94%



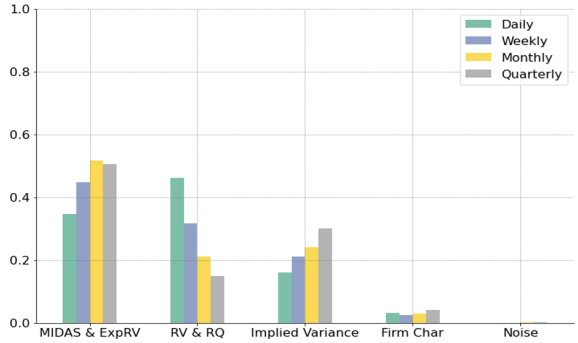
Panel D: GBRT

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	17.29%	19.96%	19.37%	17.94%
RV & RQ	33.09%	26.53%	19.85%	15.45%
Implied Variance	46.71%	50.64%	58.04%	63.59%
Firm Char	2.60%	2.58%	2.54%	2.78%
Noise	0.31%	0.29%	0.21%	0.25%



Panel E: NN

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	34.69%	44.77%	51.65%	50.45%
RV & RQ	46.18%	31.59%	21.02%	14.96%
Implied Variance	15.99%	21.22%	24.21%	30.15%
Firm Char	3.11%	2.37%	3.02%	4.20%
Noise	0.03%	0.04%	0.09%	0.24%



Panel F: AVG

	Daily	Weekly	Monthly	Quarterly
MIDAS & ExpRV	31.25%	35.49%	36.74%	32.97%
RV & RQ	33.03%	22.14%	15.30%	12.41%
Implied Variance	32.18%	39.36%	44.95%	50.49%
Firm Char	2.89%	2.61%	2.71%	3.60%
Noise	0.65%	0.41%	0.30%	0.54%

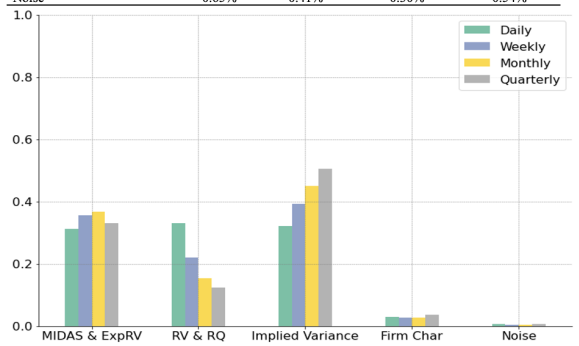


Fig. 8 Group importance based on 130 Features

This figure displays the group importance based on 130 features for LASSO, PCR, RF, GBRT, NN, and AVG across different forecast horizons. The 130-predictor feature set includes the 118 features used in the main analyses, six firm characteristics, and six noise terms. The first group "MIDAS & ExpRV" includes the $MIDAS$ term for the corresponding forecast horizon, $ExpRV^1$, $ExpRV^5$, $ExpRV^{25}$, $ExpRV^{125}$, and $ExpGlRV$. The second group "RV& RQ" includes RV^d, RV^w, RV^m, RV^q, RVP^d, RVN^d, $RV^d\sqrt{RQ^d}$, $RV^w\sqrt{RQ^w}$, $RV^m\sqrt{RQ^m}$, and $RV^q\sqrt{RQ^q}$. The third group "Implied Variance" includes $CIV^{jm,\delta}$ and $PIV^{jm,-\delta}$, where $j = 1, 2, 3$, and $\delta = 0.1, 0.15, ..., 0.9$. The fourth group "Firm Char" includes $Size$, BM, Mom, Ret^d, Ret^m, and $ILLQ^m$. The last group "Noise" includes six noise terms generated according to Eq. (25). To calculate group importance, we first compute the reduction in R^2 from permutating all values of a given group of features within each training sample, and then average the reductions of R^2 over all training samples to obtain a single group importance measure. Group importance for each forecast horizon is normalized to sum to one.

Table 1 Descriptive statistics

This table reports the descriptive statistics for 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. Superscripts d, w, m, and q are abbreviations of daily, weekly, monthly, and quarterly construction intervals or forecast horizons. $MIDAS^k$ $(k = d, w, m, q)$ denotes the smoothly weighted moving average of 50 lagged realized variances using validated polynomials from Eqs. (5) and (6) 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. (12). $ExpGlRV$ is the exponentially weighted moving average of the global risk factor with a 5-day center-of-mass from Eq. (13). $CIV^{jm,0.5}$ 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	P1	P5	Median	P95	P99	AR(1)	AR(5)	AR(21)	AR(63)
$MIDAS^d$	0.145	0.243	7.575	106.664	0.012	0.019	0.076	0.478	1.143	0.969	0.839	0.629	0.457
$MIDAS^w$	0.145	0.236	7.428	102.063	0.013	0.020	0.078	0.471	1.116	0.985	0.905	0.688	0.489
$MIDAS^m$	0.145	0.233	7.344	99.184	0.013	0.020	0.079	0.468	1.103	0.991	0.933	0.725	0.508
$MIDAS^q$	0.145	0.228	7.217	94.686	0.014	0.021	0.081	0.464	1.089	0.995	0.960	0.780	0.534
RV^d	0.144	0.299	9.249	152.262	0.009	0.014	0.065	0.507	1.349	0.581	0.466	0.366	0.280
RV^w	0.148	0.265	7.899	115.693	0.011	0.017	0.073	0.509	1.258	0.945	0.656	0.508	0.382
RV^m	0.150	0.247	8.076	119.655	0.014	0.021	0.081	0.487	1.135	0.993	0.945	0.682	0.482
RV^q	0.151	0.235	7.504	97.368	0.017	0.024	0.087	0.471	1.103	0.999	0.989	0.910	0.612
RVP^d	0.072	0.158	11.268	251.878	0.004	0.006	0.031	0.255	0.684	0.513	0.414	0.324	0.248
RVN^d	0.070	0.155	10.070	189.623	0.003	0.006	0.030	0.252	0.687	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	4.198	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	5.394	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	4.888	0.973	0.837	0.315	0.176
$RV^q\sqrt{RQ^q}$	0.287	1.820	29.610	1455.167	0.001	0.002	0.031	1.026	4.252	0.994	0.962	0.783	0.291
$ExpRV^1$	0.148	0.274	8.341	128.445	0.011	0.017	0.072	0.508	1.269	0.875	0.625	0.477	0.357
$ExpRV^5$	0.149	0.254	8.064	120.020	0.013	0.020	0.079	0.496	1.168	0.976	0.863	0.626	0.445
$ExpRV^{25}$	0.151	0.235	7.447	97.015	0.017	0.024	0.087	0.476	1.093	0.997	0.978	0.869	0.624
$ExpRV^{125}$	0.154	0.200	5.232	42.188	0.021	0.029	0.097	0.464	1.017	1.000	0.997	0.978	0.890
$ExpGlRV$	0.178	0.294	6.849	81.002	0.021	0.030	0.094	0.603	1.454	0.993	0.942	0.743	0.520
$CIV^{1m,0.5}$	0.126	0.167	5.913	63.823	0.015	0.022	0.077	0.384	0.819	0.972	0.921	0.793	0.635
$CIV^{2m,0.5}$	0.123	0.158	5.953	65.884	0.016	0.023	0.076	0.367	0.771	0.982	0.946	0.840	0.670
$CIV^{3m,0.5}$	0.118	0.146	5.643	57.782	0.016	0.023	0.075	0.348	0.719	0.988	0.959	0.868	0.700
$PIV^{1m,-0.5}$	0.132	0.200	11.795	305.546	0.016	0.023	0.079	0.395	0.860	0.977	0.930	0.803	0.642
$PIV^{2m,-0.5}$	0.129	0.191	12.798	359.667	0.018	0.025	0.080	0.377	0.807	0.985	0.953	0.852	0.681
$PIV^{3m,-0.5}$	0.126	0.181	13.983	431.430	0.019	0.027	0.080	0.359	0.755	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. Superscripts d, w, m, and q are abbreviations of daily, weekly, monthly, and quarterly construction intervals or forecast horizons. $MIDAS^k$ $(k = d, w, m, q)$ denotes the smoothly weighted moving average of 50 lagged realized variances using validated polynomials from Eqs. (5) and (6) 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. (12). $ExpGlRV$ is the exponentially weighted moving average of the global risk factor with a 5-day center-of-mass from Eq. (13). $CIV^{jm,0.5}$ 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. Superscripts d, w, m, and q are abbreviations of daily, weekly, monthly, and quarterly construction intervals or forecast horizons. $MIDAS$ denotes the smoothly weighted moving average of 50 lagged realized variances using validated polynomials from Eqs. (5) and (6) 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. (12). $ExpGlRV$ is the exponentially weighted moving average of the global risk factor with a 5-day center-of-mass from Eq. (13). $CIV^{jm,\delta}$ 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. (21).

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 reported 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. (21).

Model	Hyperparameter (Tuning parameter in **bold**)	Daily	Weekly	Monthly	Quarterly
		R^2_{OOS} relative to HAR			
LASSO	**# of shrinkage parameters (λ): 100** $\lambda_{min}/\lambda_{max}$: 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** # of trees: 500 Subsample: 0.5 Subfeature: log(# of features)	3.2%	6.4%	9.5%	5.4%
GBRT	**# of trees (B)** **Maximum tree depth (L): 1, 2, ..., 5** Learning rate: 0.001 Subsample: 0.5 Subfeature: log(# of features) Early-stopping rules (whichever met first): 1) No reduction in MSE after 50 iterations 2) Max # of trees hits 20,000	4.7%	10.2%	10.8%	6.3%
NN	# of hidden layer: 2 # of neurons: (5, 2) Activation function: ReLU	10.5%	16.7%	14.3%	4.8%
AVG		9.0%	14.3%	15.2%	10.0%

Table 5 Forecast correlation

This table reports the correlation of volatility forecasts from various models for the entire panel of stocks across 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 as 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	0.986	0.987	0.979	0.971	0.977	1.000	
AVG	0.988	0.991	0.989	0.983	0.989	0.994	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	0.987	0.985	0.970	0.970	0.979	1.000	
AVG	0.983	0.989	0.985	0.985	0.992	0.992	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	0.981	0.979	0.971	0.968	0.976	1.000	
AVG	0.974	0.988	0.987	0.985	0.993	0.990	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	0.976	0.974	0.966	0.947	0.966	1.000	
AVG	0.966	0.983	0.984	0.976	0.990	0.987	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 that the model denoted by the label to the left of a given row outperforms the model denoted by the label above the corresponding 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	10.67***	9.95***	13.09***	8.68***	11.97***	
AVG	4.75***	4.07***	10.16***	8.17***	12.28***	-6.86***

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	7.36***	6.13***	6.1***	5.25***	5.88***	
AVG	2.64***	2.23**	6.27***	5.16***	5.45***	-3.69***

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	3.26***	1.51	2.31**	2.00**	2.04**	
AVG	2.63***	2.07**	3.32***	2.81***	4.29***	0.74

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	-0.5	0.46		
NN	1.72*	0.89	-1.03	-0.24	-0.74	
AVG	3.09***	3.12***	0.83	1.78*	2.71***	2.78***

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., a simple OLS model with all 16 realized features as predictors), OLS^{IV} (i.e., a simple OLS model with all 102 implied variance features as predictors), and OLS^{ALL} (i.e., a 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 included in each subsample period according to Eq. (21). 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	8.1%	16.6%	20.2%	16.1%	13.4%	14.9%	6.6%	-2.2%	11.1%	19.6%	30.1%	23.4%
	AVG	6.6%	13.7%	19.5%	20.5%	11.3%	12.1%	8.8%	3.4%	10.4%	18.7%	29.6%	27.5%

Table 8 Realized utility

This table reports the average realized utilities from holding volatility-targeted positions based on the out-of-sample predictions of monthly RV from OLS^{ALL} (i.e., a simple OLS model using all 118 features) and AVG (i.e., a simple average of forecasts from the five individual machine learning models). Average realized utilities are calculated by averaging the realized utility from Eq. (24) over the testing sample and a given stock universe. The column labeled "All stocks" includes all 173 stocks in our main analyses. We further sort stocks into quintiles based on their median monthly RV over the entire testing sample, and report the average realized utilities for each subgroup. Panels A and B present the average realized utilities before and after transaction costs (TC). The row labeled "Utility difference btw AVG and OLS^{ALL}" reports the utility difference between AVG and OLS^{ALL}. The row labeled "DM test contrasting AVG and OLS^{ALL}" reports the Diebold-Mariano (DM) t-statistics comparing the average realized utility of AVG with that of OLS^{ALL}. Positive numbers indicate that AVG generates higher average realized utility than OLS^{ALL}. *, **, and *** indicate significance at the 10%, 5%, and 1% levels, respectively.

	Panel A: Realized utility before TC						*Panel B: Realized utility after TC*					
	All stocks	Low-Vol	2	3	4	High-Vol	All stocks	Low-Vol	2	3	4	High-Vol
OLS^{ALL}	3.47%	3.49%	3.53%	3.56%	3.32%	3.42%	3.34%	3.35%	3.40%	3.44%	3.18%	3.29%
AVG	3.71%	3.68%	3.69%	3.69%	3.73%	3.76%	3.63%	3.62%	3.62%	3.62%	3.66%	3.68%
Utility difference btw AVG and OLS^{ALL}	0.24%	0.19%	0.16%	0.13%	0.41%	0.34%	0.29%	0.27%	0.22%	0.18%	0.47%	0.38%
DM test contrasting AVG and OLS^{ALL}	8.06***	7.51***	6.37***	6.94***	5.68***	2.69***	9.46***	9.73***	8.14***	8.81***	6.19***	2.95***

Table 9 Out-of-sample prediction relative to HAR: Firm Characteristics and noise terms

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 using all 130 predictors, including 118 predictors used in the main analyses, six firm characteristics, and six pure noise terms. 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. In Panel A we replicate the results reported in Table 4 for 118 features, and in Panel B we report the results based on 130 features. Firm characteristics include $Size$, BM, Mom, Ret^d, Ret^m, and $ILLQ^m$. $Size$ is the product of the closing price and the number of shares outstanding, updated each day. BM is the book-to-market ratio in June of year t, which is computed as the ratio of the book value of common equity in fiscal year $t-1$ to the market value of equity in December of year $t-1$. Mom is the cumulative returns from prior day 252 to day 21 for a given day t. Ret^d and Ret^m refer to the past daily and 21-day returns. $ILLQ^m$ is the illiquidity measure of Amihud (2002), which is the average daily ratio of the absolute stock return to the dollar trading volume over the past 21 days. The noise terms are generated according to Eq. (25). 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 reported 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. (21).

Model	Hyperparameter	*Panel A: 118 Features*				*Panel B: 130 Features*			
		Daily	Weekly	Monthly	Quarterly	Daily	Weekly	Monthly	Quarterly
		R^2_{OOS} Relative to HAR							
OLS^{ALL}		7.6%	11.6%	7.3%	-0.6%	7.8%	12.0%	7.4%	-2.0%
LASSO	**# of shrinkage parameters (λ): 100** $lambda_{min}/\lambda_{max}$: 0.001	8.0%	12.1%	11.3%	2.6%	8.1%	12.5%	11.4%	2.2%
PCR	**# of components: 1,2,. . . ,20**	5.5%	4.8%	8.1%	7.8%	5.3%	3.4%	7.9%	8.0%
RF	**Maximum tree depth: 1,2,. . . ,20** # of trees: 500 subsample: 0.5 subfeature: log(# of features)	3.2%	6.4%	9.5%	5.4%	2.9%	5.7%	10.0%	5.8%
GBRT	**# of trees (B)** **Maximum tree depth (L): 1, 2, ..., 5** Learning rate: 0.001 Subsample: 0.5 Subfeature: log(# of features) Early-stopping rules (whichever met first): 1) No reduction in MSE after 50 iterations 2) Max # of trees hits 20,000	4.7%	10.2%	10.8%	6.3%	5.2%	11.3%	11.1%	6.5%
NN	# of hidden layer: 2 # of neurons: (5, 2) activation function: ReLU	10.5%	16.7%	14.3%	4.8%	10.7%	17.1%	12.9%	-0.3%
AVG		9.0%	14.3%	15.2%	10.0%	9.4%	14.9%	15.5%	10.0%

Table 10 Out-of-sample predictions relative to HAR for S&P 500 stocks

This table reports the out-of-sample R^2 relative to the HAR model for OLS-based and ML-based volatility forecasting models across different forecast horizons for a different set of S&P 500 stocks. The sample consists of 663 stocks listed on NYSE/AMEX/NASDAQ that have ever been included in the S&P 500 index but not members of 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 OLS-based model consist of either model-specific predictors or all 118 predictors as detailed in Table 3, and those of each ML-based model consist of all 118 predictors. Our ML-base 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). Hyperparameters for each ML-based model are provided in Table 4. We directly transfer the resulting tuning parameters for RF (i.e., maximum tree depth) and GBRT (i.e., # of trees and maximum tree depth) based on the original 173 S&P 100 stocks to this different set of 663 S&P 500 stocks and retrain both models without validating these tuning parameters. The remaining ML-based as well as OLS-based models are completely recalibrated using the new stock sample without hyperparameter transfer. R^2_{OOS} for each model is calculated relative to the prediction from HAR using the entire panel of stocks according to Eq. (21).

	Model	Tuning parameters transferred	Daily	Weekly	Monthly	Quarterly
			R^2_{OOS} *Relative to HAR*			
	MIDAS		0.2%	1.5%	0.7%	-0.8%
	SHAR		0.8%	1.0%	0.8%	0.4%
	HARQ-F		1.2%	1.7%	1.3%	1.1%
OLS	HExpGl		0.5%	2.0%	1.7%	-0.3%
	OLS^{RM}		3.3%	5.2%	3.4%	0.5%
	OLS^{IV}		-15.1%	-20.3%	-18.4%	-13.6%
	OLS^{ALL}		4.9%	8.6%	6.5%	0.8%
	LASSO		5.0%	9.1%	7.8%	2.2%
	PCR		4.3%	7.6%	3.6%	4.5%
	RF	Maximum tree depth	4.7%	7.2%	5.1%	3.2%
ML	GBRT	# of trees & maximum tree depth	5.1%	9.1%	5.6%	1.7%
	NN		8.5%	15.1%	12.0%	4.3%
	AVG		7.3%	12.8%	10.8%	6.6%

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, or *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

Tables A.1 and A.2 provide descriptive statistics for 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.

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

This table reports the descriptive statistics for 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 δ ($\delta = 0.1, 0.15, ..., 0.9$).

	Mean	Std	Skewness	Kurtosis	P1	P5	Median	P95	P99	AR(1)	AR(5)	AR(21)	AR(63)
$CIV^{1m,0.1}$	0.132	0.162	5.073	45.518	0.018	0.026	0.083	0.399	0.809	0.959	0.887	0.735	0.606
$CIV^{1m,0.15}$	0.127	0.159	5.322	50.446	0.016	0.023	0.079	0.384	0.789	0.960	0.897	0.755	0.619
$CIV^{1m,0.2}$	0.122	0.158	5.549	55.227	0.015	0.022	0.076	0.373	0.780	0.963	0.904	0.770	0.628
$CIV^{1m,0.25}$	0.120	0.157	5.674	57.932	0.014	0.021	0.074	0.368	0.777	0.966	0.911	0.780	0.634
$CIV^{1m,0.3}$	0.120	0.158	5.724	58.915	0.014	0.021	0.073	0.367	0.778	0.970	0.917	0.787	0.638
$CIV^{1m,0.35}$	0.121	0.159	5.755	59.372	0.014	0.021	0.074	0.369	0.783	0.972	0.921	0.792	0.640
$CIV^{1m,0.4}$	0.122	0.161	5.808	60.398	0.014	0.021	0.074	0.372	0.789	0.973	0.923	0.793	0.639
$CIV^{1m,0.45}$	0.124	0.164	5.907	63.664	0.015	0.021	0.075	0.377	0.802	0.972	0.922	0.793	0.638
$CIV^{1m,0.5}$	0.126	0.167	5.913	63.823	0.015	0.022	0.077	0.384	0.819	0.972	0.921	0.793	0.635
$CIV^{1m,0.55}$	0.129	0.171	5.889	62.294	0.015	0.022	0.079	0.393	0.841	0.970	0.919	0.789	0.630
$CIV^{1m,0.6}$	0.132	0.176	5.895	62.027	0.016	0.023	0.081	0.403	0.864	0.966	0.914	0.784	0.623
$CIV^{1m,0.65}$	0.137	0.181	5.882	61.501	0.016	0.024	0.084	0.416	0.891	0.960	0.907	0.776	0.613
$CIV^{1m,0.7}$	0.142	0.187	5.861	61.276	0.017	0.026	0.088	0.433	0.925	0.948	0.896	0.764	0.599
$CIV^{1m,0.75}$	0.150	0.195	5.805	60.347	0.018	0.028	0.093	0.454	0.966	0.923	0.870	0.738	0.574
$CIV^{1m,0.8}$	0.162	0.206	5.682	58.129	0.019	0.030	0.102	0.485	1.020	0.880	0.824	0.695	0.538
$CIV^{1m,0.85}$	0.178	0.220	5.445	53.879	0.020	0.032	0.114	0.530	1.092	0.818	0.758	0.631	0.492
$CIV^{1m,0.9}$	0.198	0.239	5.103	47.650	0.021	0.035	0.126	0.591	1.184	0.751	0.688	0.560	0.445
$CIV^{2m,0.1}$	0.121	0.146	5.109	46.855	0.017	0.024	0.076	0.360	0.741	0.975	0.928	0.809	0.662
$CIV^{2m,0.15}$	0.117	0.144	5.292	50.536	0.016	0.023	0.074	0.351	0.728	0.975	0.932	0.821	0.671
$CIV^{2m,0.2}$	0.115	0.144	5.449	53.516	0.015	0.021	0.072	0.345	0.721	0.977	0.936	0.830	0.676
$CIV^{2m,0.25}$	0.114	0.145	5.544	55.179	0.015	0.021	0.071	0.344	0.721	0.978	0.940	0.836	0.678
$CIV^{2m,0.3}$	0.115	0.146	5.599	56.113	0.015	0.021	0.072	0.346	0.724	0.980	0.943	0.839	0.678
$CIV^{2m,0.35}$	0.116	0.149	5.666	57.435	0.015	0.021	0.072	0.349	0.733	0.981	0.945	0.841	0.677
$CIV^{2m,0.4}$	0.118	0.151	5.790	60.751	0.015	0.022	0.073	0.354	0.743	0.982	0.946	0.841	0.675
$CIV^{2m,0.45}$	0.120	0.154	5.884	63.347	0.015	0.022	0.075	0.360	0.755	0.983	0.946	0.841	0.673
$CIV^{2m,0.5}$	0.123	0.158	5.953	65.884	0.016	0.023	0.076	0.367	0.771	0.982	0.946	0.840	0.670
$CIV^{2m,0.55}$	0.126	0.162	5.917	64.703	0.016	0.023	0.078	0.375	0.791	0.981	0.944	0.837	0.665
$CIV^{2m,0.6}$	0.129	0.166	5.854	62.121	0.016	0.024	0.080	0.385	0.812	0.979	0.942	0.834	0.660
$CIV^{2m,0.65}$	0.133	0.171	5.800	60.035	0.017	0.025	0.083	0.397	0.838	0.976	0.937	0.828	0.653
$CIV^{2m,0.7}$	0.138	0.176	5.764	59.159	0.018	0.026	0.086	0.412	0.870	0.968	0.929	0.819	0.642
$CIV^{2m,0.75}$	0.145	0.183	5.742	59.932	0.019	0.028	0.091	0.430	0.907	0.952	0.911	0.800	0.623
$CIV^{2m,0.8}$	0.155	0.192	5.625	56.961	0.020	0.030	0.099	0.454	0.954	0.920	0.876	0.764	0.593
$CIV^{2m,0.85}$	0.169	0.203	5.395	52.099	0.020	0.033	0.110	0.490	1.012	0.863	0.817	0.704	0.548
$CIV^{2m,0.9}$	0.186	0.218	5.096	46.747	0.021	0.035	0.123	0.539	1.084	0.795	0.747	0.633	0.498
$CIV^{3m,0.1}$	0.111	0.133	5.217	50.273	0.016	0.023	0.071	0.329	0.674	0.985	0.954	0.860	0.702
$CIV^{3m,0.15}$	0.109	0.133	5.350	53.231	0.015	0.021	0.069	0.324	0.665	0.986	0.956	0.867	0.710
$CIV^{3m,0.2}$	0.108	0.133	5.457	55.509	0.015	0.021	0.069	0.322	0.660	0.987	0.958	0.871	0.714
$CIV^{3m,0.25}$	0.109	0.134	5.519	56.604	0.015	0.021	0.069	0.322	0.663	0.988	0.959	0.872	0.714
$CIV^{3m,0.3}$	0.110	0.135	5.556	57.126	0.015	0.021	0.070	0.325	0.669	0.988	0.960	0.873	0.713
$CIV^{3m,0.35}$	0.112	0.137	5.572	57.149	0.015	0.021	0.071	0.329	0.677	0.988	0.960	0.872	0.711
$CIV^{3m,0.4}$	0.113	0.140	5.593	57.318	0.015	0.022	0.072	0.335	0.689	0.988	0.960	0.871	0.707
$CIV^{3m,0.45}$	0.116	0.143	5.614	57.425	0.016	0.022	0.073	0.341	0.703	0.988	0.960	0.870	0.704
$CIV^{3m,0.5}$	0.118	0.146	5.643	57.782	0.016	0.023	0.075	0.348	0.719	0.988	0.959	0.868	0.700
$CIV^{3m,0.55}$	0.121	0.150	5.641	57.356	0.016	0.024	0.077	0.357	0.735	0.987	0.958	0.866	0.696
$CIV^{3m,0.6}$	0.124	0.154	5.646	57.158	0.017	0.024	0.079	0.366	0.756	0.986	0.957	0.864	0.691
$CIV^{3m,0.65}$	0.128	0.159	5.776	61.676	0.017	0.025	0.081	0.377	0.780	0.984	0.955	0.860	0.685
$CIV^{3m,0.7}$	0.133	0.165	5.979	70.691	0.017	0.026	0.085	0.391	0.809	0.980	0.950	0.853	0.676
$CIV^{3m,0.75}$	0.139	0.172	5.953	69.854	0.018	0.028	0.089	0.407	0.845	0.971	0.940	0.841	0.662
$CIV^{3m,0.8}$	0.148	0.179	5.696	59.898	0.019	0.030	0.096	0.430	0.888	0.947	0.914	0.813	0.635
$CIV^{3m,0.85}$	0.159	0.188	5.377	51.591	0.019	0.032	0.106	0.459	0.941	0.901	0.865	0.764	0.594
$CIV^{3m,0.9}$	0.174	0.200	5.100	46.722	0.019	0.033	0.118	0.496	1.001	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 for 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 δ ($\delta = -0.9, -0.85, ..., -0.1$).

	Mean	Std	Skewness	Kurtosis	P1	P5	Median	P95	P99	AR(1)	AR(5)	AR(21)	AR(63)
$PIV^{1m,-0.1}$	0.195	0.248	7.924	131.814	0.032	0.044	0.127	0.554	1.150	0.969	0.907	0.759	0.592
$PIV^{1m,-0.15}$	0.178	0.238	8.414	149.091	0.027	0.038	0.113	0.516	1.094	0.971	0.915	0.777	0.601
$PIV^{1m,-0.2}$	0.164	0.229	8.909	168.863	0.023	0.033	0.102	0.484	1.044	0.974	0.921	0.787	0.609
$PIV^{1m,-0.25}$	0.154	0.221	9.388	189.520	0.021	0.029	0.095	0.460	0.997	0.977	0.926	0.793	0.617
$PIV^{1m,-0.3}$	0.147	0.215	9.861	211.084	0.019	0.027	0.090	0.441	0.959	0.979	0.929	0.799	0.626
$PIV^{1m,-0.35}$	0.142	0.209	10.340	233.983	0.018	0.026	0.086	0.425	0.928	0.980	0.931	0.802	0.633
$PIV^{1m,-0.4}$	0.138	0.205	10.834	258.174	0.017	0.025	0.083	0.413	0.901	0.980	0.932	0.804	0.638
$PIV^{1m,-0.45}$	0.134	0.202	11.340	283.167	0.016	0.024	0.081	0.403	0.878	0.979	0.932	0.805	0.642
$PIV^{1m,-0.5}$	0.132	0.200	11.795	305.546	0.016	0.023	0.079	0.395	0.860	0.977	0.930	0.803	0.642
$PIV^{1m,-0.55}$	0.130	0.199	12.163	322.964	0.016	0.023	0.078	0.389	0.846	0.971	0.924	0.798	0.640
$PIV^{1m,-0.6}$	0.129	0.199	12.452	335.076	0.015	0.022	0.078	0.386	0.841	0.963	0.915	0.787	0.633
$PIV^{1m,-0.65}$	0.129	0.201	12.622	340.110	0.015	0.022	0.078	0.386	0.844	0.950	0.900	0.770	0.620
$PIV^{1m,-0.7}$	0.131	0.204	12.605	336.208	0.015	0.022	0.079	0.390	0.854	0.929	0.874	0.741	0.597
$PIV^{1m,-0.75}$	0.135	0.208	12.290	318.686	0.015	0.023	0.082	0.401	0.877	0.895	0.834	0.694	0.561
$PIV^{1m,-0.8}$	0.142	0.216	11.673	288.983	0.016	0.024	0.087	0.423	0.916	0.847	0.781	0.632	0.514
$PIV^{1m,-0.85}$	0.153	0.225	10.804	250.567	0.016	0.026	0.094	0.453	0.979	0.796	0.726	0.567	0.467
$PIV^{1m,-0.9}$	0.164	0.237	9.924	213.367	0.016	0.026	0.102	0.489	1.045	0.753	0.680	0.513	0.428
$PIV^{2m,-0.1}$	0.188	0.236	8.329	148.513	0.035	0.046	0.124	0.526	1.106	0.980	0.940	0.820	0.633
$PIV^{2m,-0.15}$	0.173	0.227	8.852	169.073	0.029	0.040	0.111	0.492	1.050	0.983	0.945	0.832	0.644
$PIV^{2m,-0.2}$	0.160	0.218	9.394	192.250	0.025	0.035	0.101	0.463	0.996	0.985	0.949	0.840	0.652
$PIV^{2m,-0.25}$	0.151	0.211	9.951	217.745	0.023	0.032	0.094	0.440	0.950	0.986	0.952	0.845	0.659
$PIV^{2m,-0.3}$	0.144	0.205	10.522	244.968	0.021	0.029	0.090	0.422	0.909	0.987	0.953	0.849	0.667
$PIV^{2m,-0.35}$	0.139	0.200	11.102	273.437	0.020	0.028	0.086	0.407	0.877	0.987	0.954	0.851	0.673
$PIV^{2m,-0.4}$	0.135	0.196	11.690	303.188	0.019	0.027	0.084	0.395	0.850	0.987	0.954	0.852	0.677
$PIV^{2m,-0.45}$	0.132	0.193	12.280	333.250	0.018	0.026	0.082	0.385	0.826	0.986	0.954	0.853	0.680
$PIV^{2m,-0.5}$	0.129	0.191	12.798	359.667	0.018	0.025	0.080	0.377	0.807	0.985	0.953	0.852	0.681
$PIV^{2m,-0.55}$	0.127	0.190	13.233	381.481	0.017	0.024	0.078	0.371	0.795	0.983	0.950	0.850	0.681
$PIV^{2m,-0.6}$	0.126	0.189	13.609	398.441	0.017	0.024	0.077	0.366	0.785	0.979	0.946	0.845	0.679
$PIV^{2m,-0.65}$	0.125	0.190	13.892	408.974	0.016	0.024	0.077	0.364	0.781	0.972	0.937	0.835	0.672
$PIV^{2m,-0.7}$	0.126	0.192	14.000	409.209	0.016	0.024	0.078	0.365	0.788	0.959	0.922	0.816	0.656
$PIV^{2m,-0.75}$	0.128	0.196	13.872	397.480	0.017	0.024	0.080	0.371	0.805	0.936	0.895	0.782	0.626
$PIV^{2m,-0.8}$	0.134	0.202	13.416	370.667	0.017	0.026	0.084	0.386	0.833	0.901	0.854	0.730	0.582
$PIV^{2m,-0.85}$	0.142	0.210	12.648	331.856	0.017	0.027	0.090	0.409	0.875	0.858	0.807	0.672	0.532
$PIV^{2m,-0.9}$	0.151	0.219	11.790	291.699	0.018	0.028	0.096	0.436	0.928	0.818	0.763	0.621	0.488
$PIV^{3m,-0.1}$	0.182	0.224	8.774	167.904	0.036	0.048	0.121	0.500	1.045	0.988	0.962	0.864	0.664
$PIV^{3m,-0.15}$	0.167	0.215	9.373	194.079	0.031	0.042	0.110	0.469	0.992	0.990	0.964	0.871	0.677
$PIV^{3m,-0.2}$	0.156	0.207	10.044	225.247	0.027	0.037	0.100	0.442	0.940	0.991	0.966	0.875	0.686
$PIV^{3m,-0.25}$	0.148	0.200	10.720	258.679	0.024	0.034	0.094	0.420	0.892	0.992	0.966	0.877	0.694
$PIV^{3m,-0.3}$	0.141	0.194	11.406	294.084	0.023	0.031	0.090	0.403	0.853	0.992	0.966	0.879	0.702
$PIV^{3m,-0.35}$	0.136	0.190	12.104	330.536	0.022	0.030	0.087	0.389	0.821	0.992	0.967	0.881	0.707
$PIV^{3m,-0.4}$	0.132	0.186	12.780	366.438	0.021	0.029	0.084	0.377	0.796	0.991	0.966	0.881	0.710
$PIV^{3m,-0.45}$	0.129	0.183	13.419	400.815	0.020	0.028	0.082	0.367	0.775	0.991	0.966	0.881	0.712
$PIV^{3m,-0.5}$	0.126	0.181	13.983	431.430	0.019	0.027	0.080	0.359	0.755	0.990	0.965	0.880	0.714
$PIV^{3m,-0.55}$	0.124	0.180	14.436	454.755	0.019	0.026	0.078	0.352	0.741	0.989	0.964	0.879	0.715
$PIV^{3m,-0.6}$	0.122	0.180	14.813	470.507	0.018	0.026	0.077	0.348	0.731	0.987	0.962	0.877	0.714
$PIV^{3m,-0.65}$	0.121	0.181	15.100	478.208	0.018	0.025	0.077	0.344	0.726	0.983	0.957	0.872	0.709
$PIV^{3m,-0.7}$	0.121	0.183	15.303	479.964	0.018	0.025	0.077	0.344	0.728	0.975	0.948	0.861	0.697
$PIV^{3m,-0.75}$	0.123	0.186	15.346	474.058	0.018	0.026	0.078	0.347	0.737	0.961	0.932	0.840	0.673
$PIV^{3m,-0.8}$	0.127	0.190	15.059	452.822	0.018	0.026	0.081	0.357	0.753	0.936	0.904	0.805	0.634
$PIV^{3m,-0.85}$	0.133	0.196	14.411	416.578	0.018	0.027	0.086	0.373	0.786	0.902	0.868	0.760	0.586
$PIV^{3m,-0.9}$	0.140	0.203	13.619	376.486	0.018	0.028	0.091	0.393	0.821	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. Superscripts d, w, m, and q are abbreviations of daily, weekly, monthly, and quarterly construction intervals or forecast horizons. $MIDAS$ denotes the smoothly weighted moving average of 50 lagged realized variances using validated polynomials from Eqs. (5) and (6) 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. (12). $ExpGlRV$ is the exponentially weighted moving average of the global risk factor with a 5-day center-of-mass from Eq. (13). $CIV^{jm,\delta}$ 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. (21).

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. (21).

Model	Hyperparameter (Tuning parameter in **bold**)	Daily	Weekly	Monthly	Quarterly
		R^2_{OOS} relative to long-run mean			
LASSO	**# of shrinkage parameters (λ): 100** $\lambda_{min}/\lambda_{max}$: 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** # of trees: 500 Subsample: 0.5 Subfeature: log(# of features)	59.1%	71.4%	72.8%	65.6%
GBRT	**# of trees (B)** **Maximum tree depth (L): 1, 2, ..., 5** Learning rate: 0.001 Subsample: 0.5 Subfeature: log(# of features) Early-stopping rules (whichever met first): 1) No reduction in MSE after 50 iterations 2) Max # of trees hit 20,000	59.8%	72.6%	73.2%	65.9%
NN	# of hidden layer: 2 # of neurons: (5, 2) Activation function: ReLU	62.2%	74.5%	74.3%	65.4%
AVG		61.6%	73.8%	74.5%	67.2%

References

Amihud, Y., 2002. Illiquidity and stock returns: cross-section and time-series effects. Journal of Financial Markets 5, 31–56.

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., 2007. Roughing it up: Including jump components in the measurement, modeling, and forecasting of return volatility. Review of Economics and Statistics 89, 701–720.

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.

Andersen, T. G., Bollerslev, T., Meddahi, N., 2011. Realized volatility forecasting and market microstructure noise. Journal of Econometrics 160, 220–234.

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., Jiang, F., 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., Hansen, P. R., Lunde, A., Shephard, N., 2008. Designing realized kernels to measure the ex-post variation of equity prices in the presence of noise. Econometrica 76, 1481–1536.

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., 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., 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.

Busch, T., Christensen, B. J., Nielsen, M. O., 2011. The role of implied volatility in forecasting future realized volatility and jumps in foreign exchange, stock, and bond markets. Journal of Econometrics 160, 48–57.

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

Christensen, B. J., Prabhala, N. R., 1998. The relation between implied and realized volatility. Journal of Financial Economics 50, 125–150.

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.

Fisher, A., Rudin, C., Dominici, F., 2019. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. Journal of Machine Learning Research 20, 1–81.

French, K. R., Schwert, G. W., Stambaugh, R. F., 1987. Expected stock returns and volatility. Journal of Financial Economics 19, 3–30.

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., 2011. Volatility forecasting and microstructure noise. Journal of Econometrics 160, 257–271.

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.

Jacod, J., Li, Y., Mykland, P. A., Podolskij, M., Vetter, M., 2009. Microstructure noise in the continuous case: The pre-averaging approach. Stochastic Processes and Their Applications 119, 2249–2276.

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

Jiang, J., Kelly, B., Xiu, D., 2021b. (re-)imag(in)ing price trends. Working paper, University of Chicago, Yale University, AQR Capital Management, and NBER.

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

Li, B., Rossi, A., 2021. Selecting mutual funds from the stocks they hold: a machine learning approach. Working paper, Wuhan University and Georgetown University.

Liu, L., Patton, A. J., Sheppard, K., 2015. Does anything beat 5-minute RV? acomparison of realized measures across multiple asset classes. Journal of Econometrics 187, 293–311.

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

Newey, W. K., West, K. D., 1987. A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. Econometrica 55, 703–708.

Pan, S. J., Yang, Q., 2009. A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22, 1345–1359.

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.

Schwert, G. W., 1989. Why does stock volatility change over time? Journal of Finance 44, 1115–1153.

Schwert, G. W., 2011. Stock volatility during the recent financial crisis. European Financial Management 17, 789–805.

Schwert, G. W., Seguin, P. J., 1990. Heteroskedasticity in stock returns. Journal of Finance 45, 1129–1155.

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.

Timmermann, A., 2006. Forecast combinations. In G. Elliott, C.W.J. Granger, and A. Timmermann, eds., Handbook of Economic Forecasting 1, 135–196.

Wolpert, D. H., 1996. The lack of a priori distinctions between learning algorithms. Neural Computation 8, 1341–1390.

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.