Master's Paper of the Department of Statistics, The University of Chicago

Realized Volatility Forecasting with Machine Learning

Yichen Ji

Advisor(s): Dacheng Xiu

Approved _____

Date _____

May 1, 2024

Abstract

Realized volatility (RV) represents a nonparametric ex-post estimate of the return variation. Real-time estimates and forecasts of realized volatility play a crucial role in option pricing, trading, and risk management. This paper investigates the predictive power of machine learning models for forecasting future realized volatility in the equity market. By leveraging high-frequency intraday prices and implied volatilities (IV) derived from equity options, our empirical results within the S&P 500 universe reveal that shallow neural networks deliver superior out-of-sample predictive performance compared to existing OLS-based regression models. Furthermore, our findings are robust and scalable, extending to a much broader U.S. stock universe encompassing over 10,000 stocks spanning from 1996 to 2022.

Contents

1 Introduction

Realized volatility (RV) is a measure of volatility at a lower frequency using data at a higher frequency, derived from the sum of intraday squared returns. This measure provides a more accurate reflection of asset volatility than models based on daily or lower frequency return observations and strong parametric assumptions, as it captures the intra-day price movements that are otherwise averaged out in daily data. The granularity of high-frequency data allows for a nuanced understanding of market dynamics, making RV a pivotal object of interest in financial econometrics.

Given that volatility exhibits a relatively high signal-to-noise ratio compared to asset returns and that the availability of high-frequency intraday data supplies a large-scale dataset sufficient in both the number of data points and diversity of features, it stands to presume that realized volatility can be effectively predicted using machine learning techniques. As highlighted by Kelly et al. (2023), the presence of large conditioning panel information sets and ambiguous functional forms are two pivotal factors that underscore the potential of machine learning in financial research. This perspective echoes the reminder of Cochrane (2009) in that investors utilize conditioning information in ways that are not fully observable to researchers, thereby presenting a significant challenge in encapsulating such behaviors within a parametric statistical model comprehensively. These attributes align closely with the challenges faced in forecasting realized volatility, highlighting the suitability of machine learning approaches in addressing these complexities.

In this thesis, we apply a range of machine learning models for forecasting one-day-ahead realized volatility. The objective is to compare the predictive performance of these models against conventional time-series econometric models, which have traditionally dominated the task of volatility forecasting. Our approach involves a systematic examination of various machine learning algorithms, including LASSO, Principal Component Regression (PCR), Random Forest (RF), Gradient Boosting Regression Trees (GBRT), and neural networks, to identify the most effective technique for capturing the predictability of realized volatility.

We conduct a large-scale empirical analysis in the realm of the S&P 500 equity universe from 1996 to 2022 and demonstrate that shallow neural networks provide superior out-of-sample predictive performance relative to traditional OLS-based regression models and other machine learning methods. We construct a comprehensive set of 122 features motivated by the existing literature. This set comprises 15 realized features, 102 features derived from the implied volatility of call and put equity options across various deltas, and 4 price-volume features that carry potentially predictive information based on overnight returns and trading volume. Additionally, a constant term is included to complete the feature set for our machine learning models. Furthermore, our empirical analysis extends to an expansive dataset encompassing over 10,000 U.S. stocks spanning the period from 1996 to 2022. We show that shallow neural networks continue to dominate other models and outperform out-of-sample in terms of various evaluation metrics.

The rest of the paper is structured as follows. Section 2 conducts a comprehensive review of existing literature on volatility estimators and forecasting models. Section 3 sets up the theoretical formulation of realized volatility measures and provides an overview of the collection of econometric and machine learning models we employ in this paper. Section

4 summarizes the high-frequency intraday data and features we use for machine learning models. Section 5 presents our empirical results in both the S&P 500 universe and a much broader set of U.S. stocks. Section 6 concludes with a discussion of the findings of this study. The implementation details of our end-to-end machine-learning pipeline can be found in Appendix A.

2 Literature Review

This section provides a comprehensive review of existing literature in realized volatility models, alternative robust realized volatility measures, and machine learning applications in economics and finance.

As Engle and Patton (2007) points out, a good volatility model must be able to forecast volatility and incorporate certain stylized facts about volatility, for example, persistence, clustering, (local) mean-reversion, fat tails, leverage effects, asymmetric innovations, etc. On the one hand, ARCH/GARCH-type of models of Engle (1982) and Bollerslev (1986) and stochastic volatility models of Ghysels et al. (1996) are pivotal in financial econometrics for forecasting time-varying return volatilities and modeling conditional distributions. On the other hand, with the availability of high-frequency intraday financial data, Andersen and Bollerslev (1998) suggested using realized squared intraday returns for more accurate measurement of the true latent volatility factor in the ARCH and stochastic volatility models. Andersen et al. (2003) formally introduced the notion of realized volatility as a proxy measure of return variability and provided satisfying theoretical guarantees based on the quadratic variation theory, which motivated further development of modeling and forecasting daily and lower frequency volatility using realized volatility.

2.1 OLS-based Models

Early econometrics literature proposed various time-series volatility models whose features are motivated by structural economics assumptions and constructed based on lagged daily RVs, and whose parameters are estimated by ordinary least squares (OLS).

Ghysels et al. (2006) proposes mixed data sampling (MIDAS) whose regression weights are parameterized by a flexible function form and empirically argues that the realized daily power, calculated as the sum of intraday absolute returns, dominates other daily volatility predictors in consideration such as past daily squared returns, daily absolute returns, and realized volatility, both in-sample and out-of-sample.

The heterogeneous autoregressive (HAR) model of Corsi (2009) is an AR-type model that identifies different reactions to historical realized volatilities of different time horizons, which has arguably become a common benchmark model in the realized volatility forecasting literature.

Busch et al. (2011) analyzes the role of implied volatility in forecasting future realized volatility by including implied volatility from option prices as an additional regressor in the HAR setting and provides evidence for the sizable incremental forecasting power of implied volatility after controlling for possible endogeneity issues in the regressors. Andersen et al. (2007b) discusses the relative informativeness of implied volatility and time series-based volatility forecasts.

Andersen et al. (2007a) suggests the decomposition of RV into its continuous and jump components based on the bipower variation measures of Barndorff-Nielsen and Shephard (2006) and proposes alternative HAR-J and HAR-CJ models that include jump and continuous sample path variability measures as additional regressors in the HAR formulation. Jumps are empirically found much less persistent and predictable, and thus have limited use

for forecasting future volatility.

Building on the realized semivariance estimator proposed by Barndorff-Nielsen et al. (2008b), the semivariance-HAR (SHAR) model of Patton and Sheppard (2015) decomposes realized volatility into signed realized semivariance terms as predictors in the standard HAR formulation and demonstrates that negative realized semivariance plays a more informative role in forecasting future volatility than its positive counterpart.

Realized quarticity (RQ) is a consistent estimator of the integrated quarticity (IQ) that characterizes the asymptotic variance as shown in Barndorff-Nielsen and Shephard (2002). Motivated by the asymptotic theory and a hypothetical link between IQ and the well-known attenuation bias arising from the presence of measurement errors, Bollerslev et al. (2016) proposes the HAR-quarticity (HARQ) model that allows the coefficients to explicitly vary as a function of the realized quarticity and demonstrates superior out-of-sample performance with robustness checks using alternative RV and RQ estimators and model specifications.

Bollerslev et al. (2018) introduces a heterogeneous exponential (HExp) realized volatility model whose features are constructed using an exponentially weighted moving average (EWMA) of lagged realized volatility factors with similar horizons used in the HAR model.

There is no consensus on the impact of overnight returns and trading volume for out-of-sample realized volatility forecasting. Todorova and Souček (2014) shows none of the liquidity measures or overnight returns as additional regressors can contribute significantly to an out-of-sample forecasting improvement. Ahoniemi and Lanne (2013) highlights the importance of choosing how to treat overnight returns when determining the RV estimator to which different out-of-sample volatility forecasts are compared. Liu et al. (2023) draws evidence of the impact of trading volume on forecasting RV from the Chinese stock market.

2.2 ML-based Models

In recent years, there has been a surge of empirical attempts to use machine learning models and neural networks to forecast realized volatilities.

Hillebrand and Medeiros (2010) studies the benefits of bagging (bootstrap aggregation) log-linear models for forecasting realized volatility. Audrino and Knaus (2016) shows equal performance between the HAR model and the least absolute shrinkage and selection operator (LASSO) approach out-of-sample. Luong and Dokuchaev (2018) integrates the HAR model with random forest (RF) algorithms and illustrates improved out-of-sample performance for forecasting both the direction of the realized volatility as a classification problem and realized volatility in a regression setting. Carr et al. (2019) echos the study on the predictability of option price and implied volatility at monthly realized volatility and applies Ridge, random forest, and feed-forward neural network models to improve predictability in the context of volatility indexing.

Bucci (2020) investigates the predictive performance of feed-forward neural networks (FFNN), recurrent neural networks (RNN), and long short-term memory networks (LSTM), which shows that RNNs outperform traditional econometric methods but LSTM doesn't produce significant forecasting improvement. Reisenhofer et al. (2022) proposes a dilated convolutional neural network (CNN) called HARNet, which generates substantial improvements

relative to the baseline HAR model with the standard OLS fit, but little to no improvements relative to the HAR model with the weighted least squares (WLS) following Patton and Sheppard (2015) or the log-OLS where the input time series is log-transformed. Reisenhofer et al. (2022) also discusses the choice of the objective function for training and prefers the QLIKE loss function proposed by Patton (2011) since it stabilizes the model training and optimization, whereas Rahimikia and Poon (2020) yields substantial degradation in forecasting performance for high volatility days when the loss function changes from MSE to QLIKE.

Rahimikia and Poon (2020), Li and Tang (2022), Christensen et al. (2023), and Zhang et al. (2024) conduct extensive empirical comparative analysis on different machine learning models and neural network architectures such as regularized regression, tree-based regression, principal component regression, neural networks, etc., and report improvement in out-of-sample RV forecasts relative to the HAR benchmark using random forest, feed-forward neural networks, and ensemble models.

2.3 Alternative RV measures

Not only have different volatility models been proposed, but researchers have also improved nonparametric realized volatility estimators that are robust to jumps and microstructure noise, which is closely related to another important strand of literature in jump identification and tests. Important contributions include the bipower variation estimator by Barndorff-Nielsen and Shephard (2006), the two-scale estimator of Zhang et al. (2005), the multi-scale estimator of Zhang (2006), the realized kernel estimator of Barndorff-Nielsen et al. (2008a), the pre-averaged estimator of Podolskij and Vetter (2009), the MinRV and MedRV estimators of Andersen et al. (2012), likelihood-based estimator of Da and Xiu (2021). There is also a growing body of research focused on assessing the forecasting capabilities of these alternative RV estimators. Studies by Hansen and Lunde (2006), Bandi et al. (2008), Andersen et al. (2011), Ghysels and Sinko (2011), Bandi and Russell (2008), Andersen et al. (2011) and Caporin (2023) are notable in this regard.

2.4 ML applications in Economics and Finance

Machine learning and neural networks have shown great potential in finance and economics research in recent years. Athey et al. (2018) discusses integrating machine learning into various aspects of economic research, including policy analysis, causal inference, model selection, and empirical studies in economics. Giglio et al. (2022) surveys recent methodological contributions in asset pricing using factor models and machine learning. Nagel (2021) takes the perspective of machine learning methods as models of investor belief formation and investor learning in high-dimensional environments.

In a closely related canonical problem of empirical asset pricing, namely cross-sectional return prediction, Gu et al. (2020) demonstrates large economic gains using machine learning forecasts and finds that shallow learning outperforms deeper learning in the outperforming tree-based algorithms and neural networks. Chen et al. (2024) investigates deep neural

networks and proposes a stochastic discount factor (SDF) network architecture based on generative adversarial network (GAN) and recurrent neural network (RNN) with long short-term memory (LSTM) cells. Kelly et al. (2024) promotes using the model with the largest number of parameters possible when the true data-generating process (DGP) is unknown since the expected out-of-sample forecast accuracy and portfolio performance are strictly increasing in model complexity when appropriate shrinkage is applied.

3 Methodology

This section characterizes the mathematical formulation of realized volatility (RV) and different proposals for RV estimators we consider in this paper. Here, we focus on the setting of a standard continuous Itô process without jumps for presentation simplicity. One can refer to Andersen and Teräsvirta (2009) and McAleer and Medeiros (2008) for technical and theoretical results in the econometric formulation of realized volatility under more general settings, for example, incorporating jumps and microstructure noise. We also provide a concise introduction to the five machine-learning models we employ in this paper. One can refer to Gu et al. (2020) and Kelly et al. (2023) for a more comprehensive review of machine learning models and their applications in finance.

3.1 Theoretical Foundation

To set out the notation, assume the log price p_t within the active part of a trading day t follows a continuous semimartingale of the form

$$dp_t = \mu_t dt + \sigma_t dW_t \tag{1}$$

or equivalently, in the integral form

$$p_t = \int_{t-1}^{t} \mu_s ds + \int_{t-1}^{t} \sigma_s dW_s \tag{2}$$

where W is a standard Brownian motion, μ_t and σ_t, denote a locally bounded drift and a strictly positive càdlàg instantaneous volatility process, respectively. The *quadratic variation* (QV) of this log-price diffusion process is

$$[p, p]_t = \int_{t-1}^{t} \sigma_s^2 ds \tag{3}$$

The true unobservable volatility construct that integrates the instantaneous volatility over time is called *integrated volatility* (IV) defined as

$$IV_t = \int_{t-1}^{t} \sigma_s^2 ds \tag{4}$$

Notice that the quadratic variation and integrated volatility coincide under setting (1), which is not the case when we consider more general assumptions for the log price process, for example, the jump-diffusion process.

Since we do not obtain a continuous reading from a diffusion process and can only observe intraday price observations in discrete time, the unobservable integrated volatility/quadratic variation can be consistently estimated by the sum of squared intraday log returns which we call *realized volatility* (RV) [1]

[1] Here, we use the terms *volatility* and *variance* interchangeably for both RV and IV. Some literature denotes volatility specifically as the squared root of variance measure.

$$RV_t = \sum_{i=1}^{M} r_{t,i}^2 \xrightarrow{p} IV_t \tag{5}$$

where $r_{t,i}$ denotes the ith Δ-period log return within day t $r_{t,i} = p_{t-1+i\Delta} - p_{t-1+(i-1)\Delta}$ and therefore, the daily log return for the day t is $r_t = \sum_{i=1}^{M} r_{t,i}$.

In general, consider an equally spaced discrete time partition $\{t-k+\frac{j}{M}, j = 1, ..., M \cdot k\}$ of a k-day time interval from $t-k$ to t, $[t-k, t]$, where M is the number of observations per day, and we correspondingly denote $\Delta = \frac{1}{M}$ as the intraday sampling frequency. For example, if one samples once every 5 minutes over a typical 6.5-hour trading session ($\Delta = 5$ mins), $M = 78$ observations will be collected daily.

The semimartingale theory shows that the realized volatility converges to the quadratic variation in probability (Jacod and Protter (1998), Andersen et al. (2003)). Barndorff-Nielsen and Shephard (2002) derives the asymptotic theory of the convergence of realized volatility to integrated volatility: In the absence of price jumps, as the number of intraday observations per day $M \to \infty$ or equivalently, the sampling frequency $\Delta \to 0$,

$$\sqrt{M}\left(\frac{RV_t - IV_t}{\sqrt{2IQ_t}}\right) \xrightarrow{d} N(0,1) \tag{6}$$

where $IQ_t = \int_{t-1}^{t} \sigma_s^4 ds$ denotes *integrated quarticity*, which is independent of the limiting Gaussian distribution and can be consistently estimated by the *realized quarticity* (RQ) statistic:

$$RQ_t = \frac{M}{3} \sum_{i=1}^{M} r_{t,i}^4 \xrightarrow{p} IQ_t \tag{7}$$

Note that the consistency and asymptotic results don't hold in the presence of jumps and microstructure noises. RV consistently estimates the sum of IV and squared jumps if jumps are present. When microstructure effects are non-negligible, RV estimates are asymptotically swamped by noise and fail to converge to the IV of the underlying true equilibrium log price process.

3.2 Econometric Models

3.2.1 HAR

The heterogeneous autoregressive (HAR) model considers different volatility components realized over different time horizons (day, week, month), which echos the Heterogeneous Market Hypothesis proposed by Müller et al. (2008) that agents with daily, weekly, and monthly trading frequencies perceive and respond to, altering the corresponding components of volatility. HAR can be interpreted as a restricted AR(21) model with only 4 parameters to estimate instead of 22 in the standard AR(21) model by imposing economically sensible structural assumptions. Our HAR model specification extends the heterogeneity idea with an additional quarterly RV term:

$$RV_t = \beta_0 + \beta_d RV_{t-1}^d + \beta_w RV_{t-1}^w + \beta_m RV_{t-1}^m + \beta_q RV_{t-1}^q + \epsilon_t, \tag{8}$$

where $RV_{t-1}^l = \frac{1}{l}\sum_{i=1}^{l} RV_{t-i}, l = \{1, 5, 22, 63\}$ is the simple average of daily RVs over different lag horizons (daily, weekly, monthly, quarterly, respectively), and $\{\epsilon_t\}_t$ is a zero mean innovation process.

HAR is widely used as the benchmark model in the volatility forecasting literature due to its simplicity and parsimony.

3.2.2 MIDAS

The mixed data sampling (MIDAS) model is a unified framework that enables flexible choices of data sampling frequency and window length, which can be viewed as a smoothly weighted moving average of lagged daily RVs and specified in the following beta polynomial form:

$$\begin{aligned} RV_t &= \beta_0 + \beta_1 MIDAS_{t-1} + \epsilon_t, \\ MIDAS_t &= \frac{1}{\sum_{i=1}^{L} a_i} \sum_{i=0}^{L} a_{i+1} RV_{t-i}, \\ a_i &= \left(\frac{i}{L}\right)^{\theta_1 - 1} \left(1 - \frac{i}{L}\right)^{\theta_2 - 1} \frac{\Gamma(\theta_1 + \theta_2)}{\Gamma(\theta_1)\Gamma(\theta_2)}, i = 1, ..., L. \end{aligned} \tag{9}$$

where $\Gamma(\cdot)$ denotes the Gamma function, L is the lookback window cutoff hyperparameter, θ_1, θ_2 are hyperparameters that control the Beta function specification.

Essentially, the MIDAS model provides a tractable structure to parameterize the weights of the lagged RVs. For example, the HAR model can be interpreted as a special case of MIDAS with a_i being step functions instead of Beta functions. Note that the hyperparameters L, θ_1, θ_2 in (9) need to be tuned. We follow the choice of Li and Tang (2022) where $\theta_1 = 1, L = 50$, and θ_2 is selected by grid search that minimizes the MSE over the full sample.

3.2.3 SHAR

The semivariance-HAR (SHAR) model leverages the realized semivariance (RS) estimator proposed by Barndorff-Nielsen et al. (2008b) that decomposes the total variation into the signed components, that is, variation due to only negative or positive returns. The realized semivariances are defined as:

$$\begin{aligned} RS_t^+ &= \sum_{i=1}^{M} r_{t,i}^2 \mathbb{I}\{r_{t,i} > 0\}, \\ RS_t^- &= \sum_{i=1}^{M} r_{t,i}^2 \mathbb{I}\{r_{t,i} < 0\}. \end{aligned} \tag{10}$$

Notice that $RV_t = RS_t^+ + RS_t^-$. We adopt the main HAR model extension proposed in Patton and Sheppard (2015) that only decomposes the most recent daily RV, RV^d:

$$RV_t = \beta_0 + \beta_d^+ RS_{t-1}^{d+} + \beta_d^- RS_{t-1}^{d-} + \beta_w RV_{t-1}^w + \beta_m RV_{t-1}^m + \beta_q RV_{t-1}^q + \epsilon_t, \tag{11}$$

3.2.4 HARQ-F

The HARQ-F model further exploits the heteroskedasticity in the measurement error based on the HAR model by including the realized quarticity terms over different time horizons:

$$\begin{aligned} RV_t &= \beta_0 + (\beta_d + \phi_d\sqrt{RQ^d_{t-1}})RV^d_{t-1} + (\beta_w + \phi_w\sqrt{RQ^w_{t-1}})RV^w_{t-1} + (\beta_m + \phi_m\sqrt{RQ^m_{t-1}})RV^m_{t-1} + \\ &\quad (\beta_q + \phi_q\sqrt{RQ^q_{t-1}})RV^q_{t-1} + \epsilon_t \\ &= \beta_0 + \beta_d RV^d_{t-1} + \beta_w RV^w_{t-1} + \beta_m RV^m_{t-1} + \beta_q RV^q_{t-1} + \phi_d RV^d_{t-1}\sqrt{RQ^d_{t-1}} + \\ &\quad \phi_w RV^w_{t-1}\sqrt{RQ^w_{t-1}} + \phi_m RV^m_{t-1}\sqrt{RQ^m_{t-1}} + \phi_q RV^q_{t-1}\sqrt{RQ^q_{t-1}} + \epsilon_t \end{aligned} \tag{12}$$

According to Bollerslev et al. (2016), we can rewrite the asymptotic distribution (6) as

$$RV_t = IV_t + \eta_t, \eta_t \sim N(0, 2\Delta IQ_t) \tag{13}$$

where the estimation error η_t follows a Normal distribution conditional on the realization of integrated quarticity IQ_t.

Under some mild assumptions, the variance term $2\Delta IQ_t$ is directly linked to the attenuation bias observed in the HAR coefficients. The HARQ model compensates for this attenuation bias in HAR forecasts by accounting for the existing uncertainty in realized variance measurements: on days with a low variance in measurement errors, the daily realized volatility (RV) offers a stronger signal for predicting the following day's volatility compared to days with high variance, and vice versa.

3.2.5 HEXP

The Heterogeneous Exponential (HExp) realized volatility model is based on a mixture of exponentially weighted moving average (EWMA) volatility factors:

$$RV_t = \beta_0 + \beta_1 ExpRV^1_{t-1} + \beta_5 ExpRV^5_{t-1} + \beta_{25} ExpRV^{25}_{t-1} + \beta_{125} ExpRV^{125}_{t-1} + \epsilon_t \tag{14}$$

where

$$ExpRV^{\mathrm{CoM}(\lambda)}_t = \sum_{i=1}^{500} \frac{e^{-i\lambda}}{e^{-\lambda} + e^{-2\lambda} + \cdots + e^{-500\lambda}} RV_{t-i+1} \tag{15}$$

is the $EMWA$ of lagged daily RVs and $CoM(\lambda)$ is called the pre-specified center-of-mass ($CoM(\lambda) \in \{1, 5, 25, 125\}$ in (14)), which is formally defined as the weighted-average period for the lags used

$$\mathrm{CoM}(\lambda) \equiv \frac{\sum_{t=0}^{\infty} e^{-\lambda t} t}{\sum_{t=0}^{\infty} e^{-\lambda t}} = \frac{e^{-\lambda}}{1 - e^{-\lambda}} \tag{16}$$

and the decay rate can be reversely computed by $\lambda = \log(1 + \frac{1}{CoM})$. As specified in the footnote of Bollerslev et al. (2018), we ignore the specification that the sum in (15) only uses the first 500 lags to achieve a simple formula and the influence of the remaining lags is numerically negligible.

The authors also propose a global risk factor (GlRV), defined as the average normalized RVs across all assets, to capture spillover effects in realized volatility within and across asset classes. Correspondingly, $ExpGlRV^m$ is defined as the m-day center-of-mass EWMA of the realizations of GlRV, resulting in the HExpGl realized volatility model after including this global risk factor:

$$RV_t = \beta_0 + \beta_1 ExpRV_{t-1}^1 + \beta_5 ExpRV_{t-1}^5 + \beta_{25} ExpRV_{t-1}^{25} + \beta_{125} ExpRV_{t-1}^{125} + \beta_5^{Gl} ExpGlRV_{t-1}^5 + \epsilon_t \tag{17}$$

3.3 Machine Learning Models

3.3.1 LASSO

Least Absolute Shrinkage and Selection Operator (LASSO) is a regression method that induces sparsity and performs both variable selection and regularization to enhance the prediction accuracy and interpretability of the statistical model it produces. LASSO is particularly effective in scenarios where the number of features exceeds the number of observations or in cases of multicollinearity among predictors.

LASSO introduces a penalty to the regression model, which is equivalent to the absolute value of the coefficients' magnitude. This penalty is regulated by a tuning parameter, λ, which determines the intensity of the penalty. As λ increases, more coefficients are driven to zero. Such sparsity is one of LASSO's key features, as it allows the method to automatically exclude less influential or redundant predictors, making it highly suitable for models with a large number of predictors. The optimization problem at the core of LASSO is convex and efficient algorithms such as coordinate descent are typically employed to find the global minimum. This robust and computationally efficient approach makes LASSO an attractive option for handling large datasets.

3.3.2 Principal Component Regression (PCR)

Principal Component Regression (PCR) is a statistical technique that marries the concepts of principal component analysis (PCA) and multiple linear regression. It's particularly useful in situations where multicollinearity exists among predictor variables, or the dataset features a large number of predictors relative to the number of observations. In PCR, PCA is first employed to transform the original predictors into a new set of orthogonal components. These components are linear combinations of the original variables and are selected in such a way that they capture the maximum variance within the dataset. By focusing on these principal components instead of the original predictor variables, PCR effectively reduces the dimensionality of the data.

The application of PCR proceeds by using these principal components as the new predictor variables in a linear regression model. Typically, only a subset of the principal components—those explaining the most variance—are retained for the regression model to mitigate overfitting. This dimensionality reduction allows the model to focus on the most significant features of the data, ignoring the noise and less informative variables. PCR is therefore particularly good at dealing with complex datasets where traditional regression models might struggle due to overfitting or high degrees of freedom relative to the number of available observations. However, PCR may discard components that contain important predictive information if those components do not contribute significantly to variance, potentially leading to the loss of valuable insights. Additionally, the method relies heavily on the assumption that high variance directions correlate with important predictive features, which may not always hold, especially in complex datasets where low variance components could be crucial.

3.3.3 Random Forest (RF)

Random Forest is an ensemble learning method that builds on the simplicity of decision trees by combining multiple such trees to improve the overall model's accuracy and robustness. Each tree in a Random Forest is constructed using a random subset of the data features and samples, a technique known as bootstrap aggregating, or bagging. This randomness helps to decrease the model's variance without substantially increasing its bias, making Random Forest particularly effective in dealing with overfitting, a common problem with individual decision trees. The trees operate as a collective, where each tree contributes a vote towards the final prediction, enhancing the predictive performance across diverse datasets.

One of the key strengths of Random Forest is its versatility in handling both classification and regression tasks effectively. For regression tasks, such as forecasting realized volatility, the final prediction is typically the average of the predictions from all trees in the forest. This ensemble approach not only captures complex interactions between features but also provides a measure of feature importance based on how often each feature is used to split data across all trees, offering insights into which predictors are most influential. Moreover, Random Forest models are less sensitive to outliers and can handle non-linear data without the need for transformation, making them highly suitable for complex econometric analyses where traditional linear models might fail.

3.3.4 Gradient Boosting Regression Tree (GBRT)

Gradient Boosting Regression Tree (GBRT) is a powerful ensemble learning technique that builds on the concept of boosting, where multiple weak models (typically decision trees) are trained sequentially to correct the errors made by the strong model i.e. the composition of multiple previous weak models. Each tree in a gradient boosting model is fitted on the residual errors of the preceding tree, gradually improving the model's accuracy by focusing on difficult cases that earlier predictors struggled with. This process continues until a stopping criterion is reached, which could be either a predefined maximum number of iterations or an indication of overfitting in the strong model, as determined through performance evaluations on a separate validation dataset. This iterative error-correction process enhances the model's performance, particularly in complex regression tasks where the relationships

between variables are not straightforward. Gradient boosting effectively reduces bias and variance, making it highly adept at creating precise models from complex, high-dimensional data.

The key advantage of GBRT lies in its flexibility and robustness. It allows for the optimization of different loss functions, which can be tailored to the specific needs of the task, such as least squares for regression or logistic loss for classification. This makes it extensively adaptable across various types of data and predictive modeling challenges. Additionally, GBRT includes several tunable parameters such as the number of trees, depth of trees, and learning rate, which control the model's complexity and speed of learning. These features make gradient boosting a highly effective tool for tasks that require nuanced control over model training. However, GBRT can be computationally intensive and prone to overfitting, especially with noisy data or when too many trees are used without adequate regularization. Additionally, the model's iterative nature makes it slower for training compared to some other algorithms, requiring careful tuning and validation to achieve optimal performance.

3.3.5 Neural Networks (NN)

Neural networks are a cornerstone of modern machine learning and deep learning, distinguished by their ability to model complex and nonlinear relationships through layers of interconnected nodes, or neurons. Our study focuses on the simple feed-forward neural network architecture, also known as multilayer perceptron (MLP). In a feed-forward neural network, information moves in only one direction—from the input layer, through one or more hidden layers, to the output layer. Each neuron in a layer receives input from the previous layer, processes it using a weighted sum followed by a nonlinear activation function, and passes the output to the next layer. This architecture allows the network to capture complex patterns and interactions in the data, making feed-forward neural networks highly effective for a wide range of prediction tasks.

One of the main advantages of feed-forward neural networks is their flexibility and capacity for customization. By adjusting the number of layers and the number of neurons within each layer, these networks can be tailored to specific complexities of the data they are intended to model. Moreover, their ability to learn non-linear and high-dimensional mappings from data makes them particularly suited for challenging tasks like forecasting realized volatility in financial markets, where traditional linear models often fall short. Additionally, the training of feed-forward networks through backpropagation—a method for adjusting the weights of the network by minimizing a loss function—ensures that the model continuously improves its accuracy by learning directly from the data, adapting to new patterns as they emerge.

4 Data and Variables

This section outlines the data and variables for our empirical analysis.

4.1 Data

We consider the S&P 500 constituents and broader U.S. stocks as our two main sample universes, encompassing 1,000 and 10,014 listed securities, respectively. Our sample begins in January 1996 and ends in December 2022, totaling 27 years. The primary datasets include 1-minute observations of intraday prices sourced from the NYSE Trade and Quote (TAQ), supplemented with overnight return and trading volume data obtained from the WRDS Center for Research in Security Prices (CRSP), and options implied volatility data from OptionMetrics. The latter focuses on call-and-put options with maturities ranging from one to three months and absolute deltas varying from 0.1 to 0.9. The extensive scale of our sample universe is advantageous for examining the out-of-sample performance of machine learning models in forecasting realized volatility.

The screening process for inclusion in the U.S. stock dataset is executed in several steps to ensure both the quality and relevance of the data for our empirical study. Initially, the universe of potential securities exhaustively includes all stocks and ETFs defined in the CRSP database. To refine this list, several screening criteria are applied: each security is required to have data available for at least 100 trading days within the sample period from 1996 to 2022, possess implied volatility data from OptionMetrics, and have a start date before January 1, 2020, to exclude very recent entries, as well as an end date after January 1, 2000, to ensure their presence during the test periods.

As a result of these inclusion criteria, from an initial set of $35,960$ unique Permno identifiers found in the CRSP database, only $24,705$ securities can be matched with implied volatility data based on their CUSIP identifiers. The dataset is further reduced as many securities were excluded due to a lack of available implied volatility data in OptionMetrics. Ultimately, the dataset is narrowed down to $10,014$ unique assets. Compared to the S&P 500 dataset, the increase in the number of daily RV observations is approximately 6.2 times greater than that observed in the S&P 500 subset (29779115 versus 4804550 observations), rather than the anticipated 10 times, largely because many of the newly included securities had shorter lifespans compared to those in the S&P 500.

4.2 Features and Response Variable

Our feature set includes several distinct groups. Firstly, we construct 15 realized features, inspired by the HAR-class of econometric models discussed in Section 3.2. Secondly, 102 implied volatility features are derived from the implied volatilities of call and put equity options, spanning absolute deltas of $\delta = 0.1, 0.15, \ldots, 0.9$, and maturities of 1, 2, and 3 months. Additionally, 4 features are developed based on overnight return and trading volume data. An intercept term is also included for both OLS-based and machine-learning models. In total, we consider a comprehensive set of 122 features, which are summarized in Table 1. Our response variable is the 1-day ahead realized volatility RV_{t+1} in log-scale.

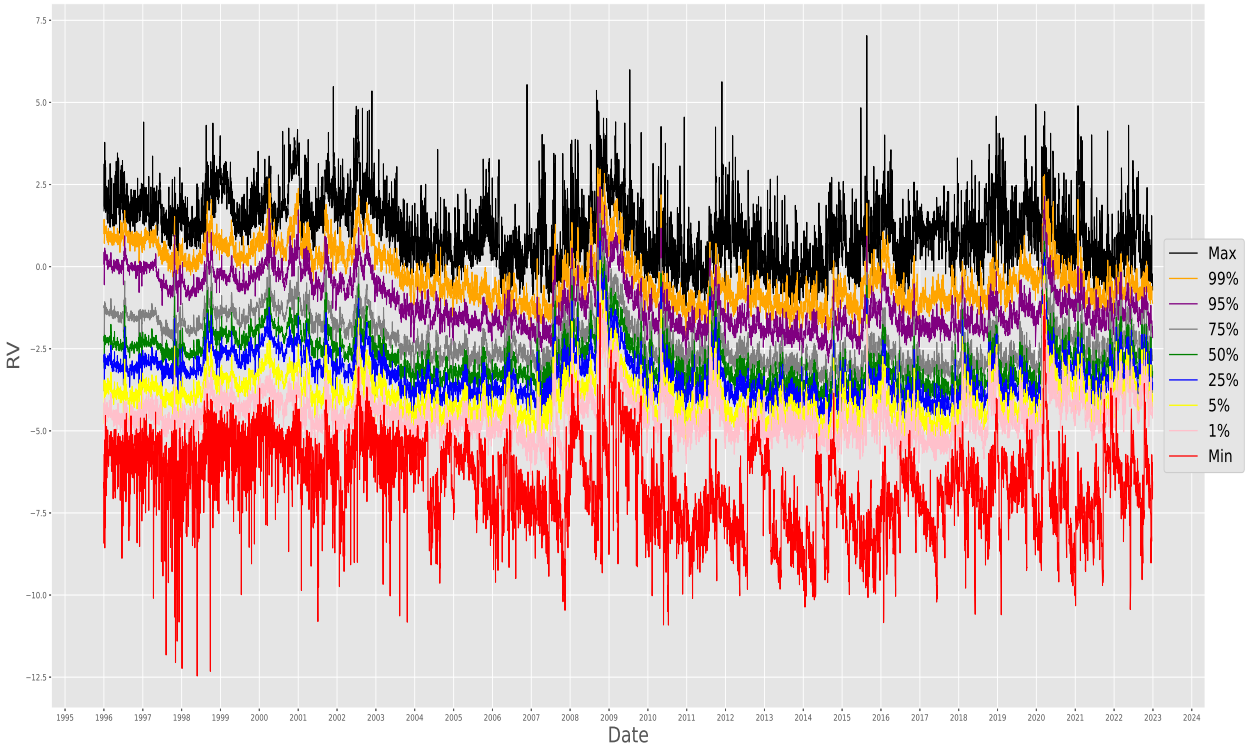

Figure 1: Quantiles of RVs, S&P 500

This figure displays the maximum, minimum, 99th, 95th, 75th, 50th, 25th, 5th, and 1st percentiles of daily realized volatilities in logarithm for stocks in the US stock universe from 1996 to 2022. The daily realized volatilities are computed using intraday high-frequency returns sampled at a 5-minute frequency.

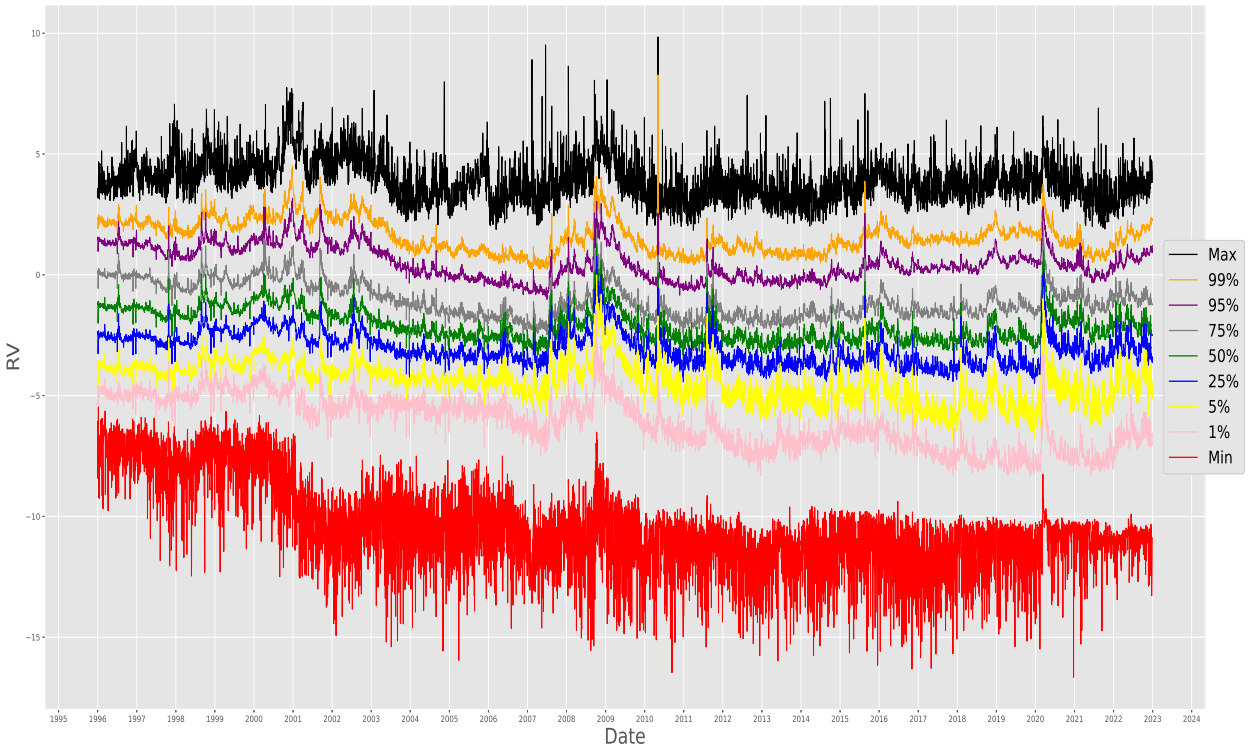

Figure 2: Quantiles of RVs, US Stocks

This figure displays the maximum, minimum, 99th, 95th, 75th, 50th, 25th, 5th, and 1st percentiles of daily realized volatilities in logarithm for stocks in the US stock universe from 1996 to 2022. The daily realized volatilities are computed using intraday high-frequency returns sampled at a 5-minute frequency.

Table 1: List of features by model

Model	Features
HAR	RV^d, RV^w, RV^m, RV^q
MIDAS	$MIDAS^d$
SHAR	$RS^{d+}, RV^d, RV^w, RV^m, RV^q$
HARQ	$RV^d, RV^w, RV^m, RV^q, RQ^d, RQ^w, RQ^m, RQ^q$
HEXP	$ExpRV^1, ExpRV^5, ExpRV^{25}, ExpRV^{125}, ExpGlRV^5$
OLSRM	$RV^d, RV^w, RV^m, RV^q, MIDAS^d, RS^{d+}, RQ^d, RQ^w, RQ^m, RQ^q,$
	$ExpRV^1, ExpRV^5, ExpRV^{25}, ExpRV^{125}, ExpGlRV^5$ (15 realized features)
OLSRM4	$RV^d, RV^w, RV^m, RV^q, MIDAS^d, RS^{d+}, RQ^d, RQ^w, RQ^m, RQ^q,$
	$ExpRV^1, ExpRV^5, ExpRV^{25}, ExpRV^{125}, ExpGlRV^5,$
	OVN, OVN^2, Vo, LVP (15 realized features + 4 price volume features)
OLSIV	$CIV^{im,\delta}, PIV^{im,\delta}, i = 1, 2, 3, \delta = \pm 0.1, \pm 0.15, ..., \pm 0.9$ (102 IV features)
OLSALL	All 122 features
	(15 realized features + 102 IV features + 4 price volume features + 1 intercept)
ML	All 122 features
	(15 realized features + 102 IV features + 4 price volume features + 1 intercept)

This table reports the list of features for each model. Daily, weekly, monthly, and quarterly frequency are abbreviated using superscripts d, w, m, q, respectively. The realized features are all well-defined in Section 3.2. OVN is the daily overnight log return calculated using the holding period return data from CRSP. OVN^2 is the polarized square of overnight return, calculated by taking the square and keeping the sign of OVN. Vo is the trading volume for each security obtained from CRSP. LVP is the log of the product of trading volume and price. $CIV^{im,\delta}$ and $PIV^{im,\delta}$ are implied volatilities from call and put options with $\delta = \pm 0.1, \pm 0.15, ..., \pm 0.9$ with maturity equal to i months, $i = 1, 2, 3$. $OLSRM$ is the simple OLS regression model with all 15 realized features as predictors. $OLSRM4$ is the same as $OLSRM$ but with the inclusion of 4 price volume features as regressors. $OLSIV$ is the OLS model with all 102 implied volatility features as predictors. $OLSALL$ is the OLS model with all 122 features as predictors (including an intercept term). Machine learning (ML) models include $LASSO$, PCR, RF, $GBRT$, and NN introduced in Section 3.3.

5 Empirical Results

This section introduces the performance evaluation metrics and shows the out-of-sample volatility forecasting comparison of all 14 models we consider as listed in Table 1 and discusses our empirical findings for both the S&P 500 universe and the U.S. stocks universe. Appendix A reports the implementation details for our end-to-end machine learning pipeline such as model training scheme and hyperparameter tuning.

5.1 Out-of-Sample Performance Evaluation

Different metrics can lead to different or even incorrect rankings of volatility forecasts. Patton (2011) discusses the robustness of different loss function candidates and proves the mean squared error (MSE) and quasi-likelihood (QLIKE) are robust to the ranking of competing volatility forecasts in the presence of noise in the volatility proxy (RV is a volatility proxy to the latent IV). Patton and Sheppard (2009) argues that QLIKE has the highest power in the Diebold–Mariano (DM) test which compares the forecast accuracy of two forecasting models. Based on these findings, we suggest investigating the following 3 metrics together for out-of-sample forecasting performance evaluation:

- $R^2 = 1 - \frac{\sum_{i,t}\left(RV_{i,t} - \widehat{RV_{i,t}}\right)^2}{\sum_{i,t}\left(RV_{i,t} - \widehat{RV_{i,t}}^{\text{benchmark}}\right)^2}$,
- Mean squared error (MSE) $= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\#\mathcal{T}_{\text{test}}}\sum_{t\in\mathcal{T}_{\text{test}}}\left(\text{RV}_{i,t} - \widehat{\text{RV}}_{i,t}\right)^2$
- Quasi-likelihood (QLIKE) $= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\#\mathcal{T}_{\text{test}}}\sum_{t\in\mathcal{T}_{\text{test}}}\left[\frac{\exp(\text{RV}_{i,t})}{\exp\left(\widehat{\text{RV}}_{i,t}\right)} - \left(\text{RV}_{i,t} - \widehat{\text{RV}}_{i,t}\right) - 1\right]$

where $RV_{i,t}$ is the true realized volatility, $\widehat{\text{RV}}_{i,t}$ is the RV forecast of our model, HAR is the benchmark model in R^2, N is the number of securities, $\mathcal{T}_{test}$ is the test time.

5.2 Main Findings

Table 2 reports the out-of-sample results for forecasting future realized volatility for each model in different metrics in the S&P 500 universe. Neural networks outperform all other models in all 3 evaluation metrics. Among the OLS-based models, OLSALL demonstrates sizable improvement in all evaluation metrics compared to the HAR benchmark leveraging all 122 features in our feature set. There is no significant difference between OLSRM and OLSRM4, which implies that the additional price-volume features don't provide predictive information for the 1-day ahead RV. OLSIV is the worst-performing model among all models, illustrating little predictive power using implied volatility alone for realized volatility forecasting. Among machine learning models, LASSO, PCR, and RF achieve similar out-of-sample forecasting abilities to OLSALL, and GBRT underperforms the worst.

For the U.S. stock universe, Table 3 shows that neural networks still outperform all other models among all evaluation metrics except for QLIKE. Since neural networks achieve the

best winsorized QLIKE score after taking the extreme values into account, we can continue to conclude that the neural network model is the best-performing forecasting model. OLSIV and GBRT are the worst-performing models out-of-sample.

Model	R2	MSE	MSE*	QLike	QLike*
HAR	0.7052	0.3970	0.3962	0.4039	0.3737
MIDAS	0.6995	0.4047	0.4039	0.4018	0.3729
SHAR	0.7057	0.3963	0.3955	0.4029	0.3735
HARQ	0.7187	0.3787	0.3780	0.3912	0.3601
HEXP	0.7071	0.3944	0.3936	0.4015	0.3721
OLSRM	0.7201	0.3768	0.3761	0.3880	0.3583
OLSRM4	0.7202	0.3768	0.3761	0.3874	0.3578
OLSIV	0.6096	0.5257	0.5248	0.4471	0.4128
OLSALL	0.7276	0.3668	0.3660	0.3673	0.3366
LASSO	0.7276	0.3668	0.3661	0.3667	0.3368
PCR	0.7216	0.3748	0.3740	0.3734	0.3416
RF	0.7204	0.3765	0.3758	0.3681	0.3373
GBRT	0.7068	0.3948	0.3941	0.3854	0.3567
NN	**0.7321**	**0.3607**	**0.3599**	**0.3576**	**0.3245**

Table 2: Out-of-sample Forecasting Performance, S&P 500 Stocks

This table presents evaluation metrics for out-of-sample RV forecasting performance in the S&P 500 universe. $*$ denotes the winsorized metric where RV extreme values beyond 99.99^{th} percentile are replaced by the boundary value at 99.99^{th} percentile. The best-performing model is highlighted in bold in each column.

We also compare the differences in the out-of-sample forecasting results between using individual fitting and pooled fitting for 9 OLS-based models in Table 5 and Table 6 in Appendix B. Individual fitting refers to individual estimation for each asset, and pooled fitting means utilizing panel data to get one set of parameter estimates applied to all assets. All of the evaluation metrics show that MIDAS is the best-performing model using individual fitting and OLSALL is the best-performing model using pooled fitting. We also find that pooled fitting outperforms individual fitting for all models and evaluation metrics in both the S&P 500 and the U.S. stock universes, and the contrast is more significant when the model employs more features, especially for OLSIV and OLSALL, which demonstrates the predictive benefit of pooled panel estimation exploiting the commonality of volatility dynamics.

We conduct a pairwise Diebold-Mariano (DM) test (Diebold and Mariano (2002)) to compare the differences in forecasting accuracy between different forecasting models. A positive test statistic indicates that the model corresponding to the row performs better than the model corresponding to the column. Table 9 and Table 10 in Appendix B present the test statistics for all of 14 competing models for the S&P 500 and the U.S. stocks universe, respectively. In both cases, OLSALL is the best-performing OLS-based model, and the neural network model remarkably outperforms all other competing models.

Model	R2	MSE	MSE*	QLike	QLike*
HAR	0.7849	0.5708	0.5680	1.5628	0.4886
MIDAS	0.7815	0.5798	0.5771	1.4024	0.4914
SHAR	0.7850	0.5706	0.5678	1.6462	0.4883
HARQ	0.7884	0.5615	0.5587	1.6487	0.4864
HEXP	0.7863	0.5670	0.5643	1.4541	0.4827
OLSRM	0.7897	0.5580	0.5552	1.7167	0.4819
OLSRM4	0.7898	0.5578	0.5550	1.6645	0.4817
OLSIV	0.5109	1.2980	1.2951	1.4857	1.0282
OLSALL	0.7906	0.5557	0.5529	1.5204	0.4758
LASSO	0.7904	0.5563	0.5535	1.5025	0.4758
PCR	0.7861	0.5675	0.5647	1.3597	0.4781
RF	0.7905	0.5561	0.5533	1.2245	0.4594
GBRT	0.7756	0.5954	0.5926	**1.1476**	0.4855
NN	**0.7954**	**0.5428**	**0.5400**	1.4290	**0.4509**

Table 3: Out-of-sample Forecasting Performance, US Stocks

This table presents evaluation metrics for out-of-sample RV forecasting performance in the U.S. stock universe. $*$ denotes the winsorized metric where RV extreme values beyond 99.99^{th} percentile are replaced by the boundary value at 99.99^{th} percentile. The best-performing model is highlighted in bold in each column.

6 Conclusions and Discussion

In conclusion, our empirical analysis shows that shallow neural network delivers superior out-of-sample forecasting performance compared to OLS-based HAR-class models, linear machine learning models, and tree-based machine learning models.

I would like to highlight several points for further discussion:

- Many econometric conclusions, traditionally derived from empirical analyses with limited sample sizes, may not be sustainable in the current context due to the modern scale of data now available. When models and results premised on smaller datasets are subjected to validation with substantially larger and more diverse datasets, these data-dependent empirical outcomes may not hold.

- Machine learning models often exhibit high turnover in their portfolio constructions, making it challenging to achieve substantial net-of-fee excess returns in practical applications. Moreover, the predictability uncovered by these models tends to concentrate on stocks with high arbitrage and transaction costs, which limits their value to certain institutional investors. A crucial prerequisite for the successful implementation of machine learning models in the market is the ability to effectively tweak nonlinear models to focus on stocks with lower arbitrage and transaction costs.

- Domain knowledge in economic and finance theory is indispensable for effectively integrating machine learning into financial modeling as addressed by Giglio et al. (2022). In investment practice, however, emphasizing the economic rationale behind the explanatory factors in forecasting models does not necessarily enhance profit-making

opportunities; rather, forecasting accuracy is paramount. Focusing on whether explanatory factors have an economic underpinning may not invariably contribute to the stability of predictions.

- The economic implications of the gains from machine learning volatility forecasts that outperform existing HAR models remain an open problem as pointed out by Kelly et al. (2023). Understanding the practical benefits and the incremental value added by advanced machine learning techniques compared to traditional models is crucial for further adoption in financial practices.

A Appendix: Implementation

A.1 Insanity Filter

Following Swanson and White (1997), Bollerslev et al. (2018), and Li and Tang (2022), we use an insanity filter to avoid deflation in out-of-sample R^2, that is, we replace any RV forecast that exceeds the maximum in the training sample with the observed maximum, and vice versa. Having this insanity filter allows for empirically more meaningful model comparisons. We report the effect of the insanity filter in Table 7 and 8 for the S&P 500 and the U.S. stock universe, respectively.

A.2 Training Scheme

We employ a rolling window approach to train our machine learning models using pooled panel data from the entire stock universe. Specifically, each window spans seven years, allocated as 5 years for training, 1 year for validation, and 1 year for testing. This method preserves the temporal and chronological order of the train-validation-test sequence, ensuring that the models do not inadvertently use future information. We refit the models annually by shifting the training, validation, and testing windows forward by one year.

A.3 Choice of Tuning Hyperparameters

Table 4 provides the tuning hyperparameters for the machine learning models.

Model	Hyper-parameter	Value
Lasso	Number of λ's	100
	$\lambda_{Min}/\lambda_{Max}$	10^{-4}
	Maximum iteration	10^6
PCR	Maximum components	50
	Minimum variance ratio	10^{-6}
	SVD Solver	Full
RF	Number of trees	100
	Maximum depth	[10,15]
	Minimum sample at leaf node	10
	Number of features to consider for best split	$\sqrt{p}$
	Loss function	MSE
GBRT	Number of trees	1000
	Learning rate	$[10^{-1}, 10^{-2}, 10^{-3}]$
	Maximum depth	3
	Minimum sample at leaf node	10
	Number of features to consider for best split	$\sqrt{p}$
	Validation fraction	10%
	Loss function	MSE
NN	Architecture	$50 \times 10 \times 10 \times 5$
	Training batch size	10000
	Validation frequency	20
	Epoch	100
	Learning rate	$[0.01, 0.003, 0.001, 0.0003, 0.0001]$
	Patience threshold N_{pthres}	100
	Loss function	QLike

Table 4: Hyperparameters for Machine Learning Models

Note: This table reports the hyperparameters for the five machine learning models we considered in the paper, Lasso, Principal Component Regression (PCR), Random Forest (RF), Gradient Boosting Regression (GBRT), and Neural Network (NN).

B Appendix: Tables

	R2		MSE*		QLike*	
Model	Individual	Pooled	Individual	Pooled	Individual	Pooled
HAR	0.6833	0.7052	0.4253	0.3962	0.4305	0.3737
MIDAS	**0.6907**	0.6995	**0.4158**	0.4039	**0.3798**	0.3729
SHAR	0.6834	0.7057	0.4252	0.3955	0.4335	0.3735
HARQ	0.6775	0.7187	0.4332	0.3780	0.5024	0.3601
HEXP	0.6693	0.7071	0.4442	0.3936	0.4701	0.3721
OLSRM	0.6734	0.7201	0.4383	0.3761	0.4894	0.3583
OLSRM4	0.6654	0.7202	0.4492	0.3761	0.5145	0.3578
OLSIV	0.4551	0.6096	0.7317	0.5248	0.8039	0.4128
OLSALL	0.5514	**0.7276**	0.6019	**0.3660**	0.7744	**0.3366**

Table 5: Individual vs Pooled Fit, S&P 500 Stocks

	R2		MSE*		QLike*	
Model	Individual	Pooled	Individual	Pooled	Individual	Pooled
HAR	0.6991	0.7849	0.7953	0.5680	6.1815	0.4886
MIDAS	**0.7434**	0.7815	**0.6777**	0.5771	**0.6883**	0.4914
SHAR	0.6992	0.7850	0.7947	0.5678	5.8682	0.4883
HARQ	0.6379	0.7884	0.9581	0.5587	26.9071	0.4864
HEXP	0.6427	0.7863	0.9452	0.5643	22.3139	0.4827
OLSRM	0.6032	0.7897	1.0501	0.5552	32.1466	0.4819
OLSRM4	0.5933	0.7898	1.0762	0.5550	36.0054	0.4817
OLSIV	0.3112	0.5109	1.8256	1.2951	45.2375	1.0282
OLSALL	0.4051	**0.7906**	1.5765	**0.5529**	64.0068	**0.4758**

Table 6: Individual vs Pooled Fit, U.S. Stocks

	Individual MSE			Pooled MSE		
Model	Pre	Counts	Post	Pre	Counts	Post
HAR	0.5219	1512	0.4265	0.3970	1	0.3970
MIDAS	**0.4165**	0	**0.4165**	0.4047	1	0.4047
SHAR	0.5590	1734	0.4264	0.3963	1	0.3963
HARQ	0.8026	5531	0.4342	0.3787	1	0.3787
HEXP	1.4669	11534	0.4452	0.3944	0	0.3944
OLSRM	2.0154	16484	0.4398	0.3768	0	0.3768
OLSRM4	2.0220	16080	0.4506	0.3768	3	0.3768
OLSIV	1.5692	12449	0.7338	0.5257	1	0.5257
OLSALL	1.6653	14150	0.6040	0.3668	2	0.3668
LASSO	-	-	-	0.3668	0	0.3668
PCR	-	-	-	0.3748	1	0.3748
RF	-	-	-	0.3765	0	0.3765
GBRT	-	-	-	0.3948	0	0.3948
NN	-	-	-	**0.3607**	3	**0.3607**

Table 7: Insanity Filter, S&P 500 Stocks

This table presents the impact of applying an insanity filter on the out-of-sample forecasting performance within the SP 500 universe. It details the Mean Squared Error (MSE) for each model before (Pre) and after (Post) applying the insanity filter, alongside the frequency of the filter's activation (Counts). The table is organized into two panels: the left panel displays results from individual fits, while the right panel, due to computational constraints, exclusively shows pooled fits for the machine learning models. The results demonstrate a significant improvement in the MSE for individual OLS fits post-filter application, particularly when the filter is frequently triggered, although this is not the case for MIDAS. Conversely, in the pooled fits, the insanity filter is seldom triggered, leading to minimal changes in the MSE. Overall, while the insanity filter substantially enhances the performance of individual fits, these still underperform relative to the pooled fits. The column indicating the best performance is highlighted in bold for clarity and emphasis.

Model	Individual MSE			Pooled MSE		
	Pre	Counts	Post	Pre	Counts	Post
HAR	1.3978	54878	0.7985	0.5708	23	0.5708
MIDAS	**0.6993**	2027	**0.6809**	0.5798	11	0.5798
SHAR	1.4003	55707	0.7982	0.5706	23	0.5706
HARQ	2.9585	174428	0.9609	0.5615	28	0.5615
HEXP	4.5651	261614	0.9480	0.5670	21	0.5670
OLSRM	7.0873	425734	1.0529	0.5580	36	0.5580
OLSRM4	6.8668	414680	1.0791	0.5579	46	0.5578
OLSIV	8.1848	481612	1.8279	1.2980	0	1.2980
OLSALL	8.0473	501998	1.5788	0.5558	37	0.5557
LASSO	-	-	-	0.5563	23	0.5563
PCR	-	-	-	0.5675	22	0.5675
RF	-	-	-	0.5561	0	0.5561
GBRT	-	-	-	0.5954	0	0.5954
NN	-	-	-	**0.5428**	11	**0.5428**

Table 8: Insanity Filter, US Stocks

This table extends the analysis from Table 7, focusing on the forecasting performance for a broader set of US stocks. We observe a consistent pattern where the insanity filter is triggered more frequently in individual fits, leading to improved performance. However, a substantial performance gap remains when compared to the pooled fits. Specifically, for US stocks, the insanity filter is activated even more frequently in individual fits than in those involving S&P 500 stocks, indicating heightened sensitivity. In contrast, the pooled fits show robustness, evidenced by fewer trigger events and minimal changes in the Mean Squared Error (MSE). This stability underscores the relative performance strength of the pooled models over individual fits in this broader dataset.

Model	HAR	MIDAS	SHAR	HARQ	HEXP	OLSRM	OLSRM4	OLSIV	OLSALL	LASSO	PCR	RF	GBRT
MIDAS	-50.2	-	-	-	-	-	-	-	-	-	-	-	-
SHAR	28.0	55.6	-	-	-	-	-	-	-	-	-	-	-
HARQ	135.5	153.6	131.0	-	-	-	-	-	-	-	-	-	-
HEXP	43.5	72.5	28.5	-109.6	-	-	-	-	-	-	-	-	-
OLSRM	141.6	169.0	142.6	34.5	126.7	-	-	-	-	-	-	-	-
OLSRM4	139.9	167.4	140.9	33.2	125.1	1.3	-	-	-	-	-	-	-
OLSIV	-27.4	-25.8	-27.6	-31.4	-28.0	-31.8	-31.8	-	-	-	-	-	-
OLSALL	114.8	145.3	113.4	59.3	107.8	52.1	52.3	34.6	-	-	-	-	-
LASSO	112.8	141.4	111.2	56.7	106.3	49.0	48.9	34.6	-1.0	-	-	-	-
PCR	81.8	110.2	79.4	16.1	76.0	8.5	8.4	32.9	-82.8	-95.8	-	-	-
RF	83.0	112.1	80.0	10.2	72.4	1.4	1.2	32.0	-43.6	-42.1	-6.9	-	-
GBRT	4.0	18.0	2.7	-30.5	-0.9	-34.3	-34.4	28.9	-60.3	-61.6	-44.8	-38.9	-
NN	131.2	159.7	129.2	84.1	120.7	74.7	73.7	35.4	28.6	27.2	55.9	87.2	62.5

Table 9: Diebold-Mariano Test, S&P 500 Stocks

Model	HAR	MIDAS	SHAR	HARQ	HEXP	OLSRM	OLSRM4	OLSIV	OLSALL	LASSO	PCR	RF	GBRT
MIDAS	-96.8	-	-	-	-	-	-	-	-	-	-	-	-
SHAR	19.2	99.6	-	-	-	-	-	-	-	-	-	-	-
HARQ	138.8	197.1	137.7	-	-	-	-	-	-	-	-	-	-
HEXP	109.5	145.9	96.4	-75.9	-	-	-	-	-	-	-	-	-
OLSRM	167.5	242.2	170.8	100.2	125.8	-	-	-	-	-	-	-	-
OLSRM4	143.9	218.4	145.1	61.7	106.8	4.4	-	-	-	-	-	-	-
OLSIV	-103.9	-102.7	-104.0	-105.4	-104.4	-105.9	-105.9	-	-	-	-	-	-
OLSALL	156.0	236.7	156.8	89.1	121.3	40.2	55.2	106.5	-	-	-	-	-
LASSO	177.5	248.7	178.6	101.6	139.3	38.5	24.2	106.5	-11.7	-	-	-	-
PCR	55.3	119.0	50.4	-66.3	-8.1	-105.6	-95.7	104.8	-124.9	-142.9	-	-	-
RF	80.6	135.8	79.2	30.0	61.4	10.9	9.5	106.6	-1.9	1.2	63.9	-	-
GBRT	-41.2	-25.9	-41.5	-56.7	-47.6	-62.6	-62.9	104.7	-67.5	-66.9	-47.7	-72.2	-
NN	197.4	278.7	194.9	134.4	176.5	110.1	108.7	108.1	101.5	107.1	177.8	96.2	88.2

Table 10: Diebold-Mariano Test, U.S. Stocks

References

Katja Ahoniemi and Markku Lanne. Overnight stock returns and realized volatility. *International Journal of Forecasting*, 29(4):592–604, 2013.

Torben G Andersen and Tim Bollerslev. Answering the skeptics: Yes, standard volatility models do provide accurate forecasts. *International economic review*, pages 885–905, 1998.

Torben G Andersen and Timo Teräsvirta. Realized volatility. In *Handbook of financial time series*, pages 555–575. Springer, 2009.

Torben G Andersen, Tim Bollerslev, Francis X Diebold, and Paul Labys. Modeling and forecasting realized volatility. *Econometrica*, 71(2):579–625, 2003.

Torben G Andersen, Tim Bollerslev, and Francis X Diebold. Roughing it up: Including jump components in the measurement, modeling, and forecasting of return volatility. *The review of economics and statistics*, 89(4):701–720, 2007a.

Torben G Andersen, Per Frederiksen, and Arne D Staal. The information content of realized volatility forecasts. *Northwestern University, Nordea Bank, and Lehman Brothers*, 2007b.

Torben G Andersen, Tim Bollerslev, and Nour Meddahi. Realized volatility forecasting and market microstructure noise. *Journal of Econometrics*, 160(1):220–234, 2011.

Torben G Andersen, Dobrislav Dobrev, and Ernst Schaumburg. Jump-robust volatility estimation using nearest neighbor truncation. *Journal of Econometrics*, 169(1):75–93, 2012.

Susan Athey et al. The impact of machine learning on economics. *The economics of artificial intelligence: An agenda*, pages 507–547, 2018.

Francesco Audrino and Simon D Knaus. Lassoing the har model: A model selection perspective on realized volatility dynamics. *Econometric Reviews*, 35(8-10):1485–1521, 2016.

Federico M Bandi and Jeffrey R Russell. Microstructure noise, realized variance, and optimal sampling. *The Review of Economic Studies*, 75(2):339–369, 2008.

Federico M Bandi, Jeffrey R Russell, and Chen Yang. Realized volatility forecasting and option pricing. *Journal of Econometrics*, 147(1):34–46, 2008.

Ole E Barndorff-Nielsen and Neil Shephard. Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 64(2):253–280, 2002.

Ole E Barndorff-Nielsen and Neil Shephard. Econometrics of testing for jumps in financial economics using bipower variation. *Journal of financial Econometrics*, 4(1):1–30, 2006.

Ole E Barndorff-Nielsen, Peter Reinhard Hansen, Asger Lunde, and Neil Shephard. Designing realized kernels to measure the ex post variation of equity prices in the presence of noise. *Econometrica*, 76(6):1481–1536, 2008a.

Ole E Barndorff-Nielsen, Silja Kinnebrock, and Neil Shephard. Measuring downside risk-realised semivariance. *CREATES Research Paper*, (2008-42), 2008b.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

Tim Bollerslev, Andrew J Patton, and Rogier Quaedvlieg. Exploiting the errors: A simple approach for improved volatility forecasting. *Journal of Econometrics*, 192(1):1–18, 2016.

Tim Bollerslev, Benjamin Hood, John Huss, and Lasse Heje Pedersen. Risk everywhere: Modeling and managing volatility. *The Review of Financial Studies*, 31(7):2729–2773, 2018.

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

Thomas Busch, Bent Jesper Christensen, and Morten Ørregaard Nielsen. The role of implied volatility in forecasting future realized volatility and jumps in foreign exchange, stock, and bond markets. *Journal of Econometrics*, 160(1):48–57, 2011.

Massimiliano Caporin. The role of jumps in realized volatility modeling and forecasting. *Journal of Financial Econometrics*, 21(4):1143–1168, 2023.

Peter Carr, Liuren Wu, and Zhibai Zhang. Using machine learning to predict realized variance. *arXiv preprint arXiv:1909.10035*, 2019.

Luyang Chen, Markus Pelger, and Jason Zhu. Deep learning in asset pricing. *Management Science*, 70(2):714–750, 2024.

Kim Christensen, Mathias Siggaard, and Bezirgen Veliyev. A machine learning approach to volatility forecasting. *Journal of Financial Econometrics*, 21(5):1680–1727, 2023.

John Cochrane. *Asset pricing: Revised edition*. Princeton university press, 2009.

Fulvio Corsi. A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2):174–196, 2009.

Rui Da and Dacheng Xiu. When moving-average models meet high-frequency data: Uniform inference on volatility. *Econometrica*, 89(6):2787–2825, 2021.

Francis X Diebold and Robert S Mariano. Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1):134–144, 2002.

Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, pages 987–1007, 1982.

Robert F Engle and Andrew J Patton. What good is a volatility model? In *Forecasting volatility in the financial markets*, pages 47–63. Elsevier, 2007.

Eric Ghysels and Arthur Sinko. Volatility forecasting and microstructure noise. *Journal of Econometrics*, 160(1):257–271, 2011.

Eric Ghysels, Andrew C Harvey, and Eric Renault. 5 stochastic volatility. *Handbook of statistics*, 14:119–191, 1996.

Eric Ghysels, Pedro Santa-Clara, and Rossen Valkanov. Predicting volatility: getting the most out of return data sampled at different frequencies. *Journal of Econometrics*, 131 (1-2):59–95, 2006.

Stefano Giglio, Bryan Kelly, and Dacheng Xiu. Factor models, machine learning, and asset pricing. *Annual Review of Financial Economics*, 14:337–368, 2022.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020.

Peter R Hansen and Asger Lunde. Realized variance and market microstructure noise. *Journal of Business & Economic Statistics*, 24(2):127–161, 2006.

Eric Hillebrand and Marcelo C Medeiros. The benefits of bagging for forecast models of realized volatility. *Econometric Reviews*, 29(5-6):571–593, 2010.

Jean Jacod and Philip Protter. Asymptotic error distributions for the euler method for stochastic differential equations. *The Annals of Probability*, 26(1):267–307, 1998.

Bryan Kelly, Dacheng Xiu, et al. Financial machine learning. *Foundations and Trends® in Finance*, 13(3-4):205–363, 2023.

Bryan Kelly, Semyon Malamud, and Kangying Zhou. The virtue of complexity in return prediction. *The Journal of Finance*, 79(1):459–503, 2024.

Sophia Zhengzi Li and Yushan Tang. Automated risk forecasting. In *Automated Risk Forecasting: Li, Sophia Zhengzi— Tang, Yushan*. [Sl]: SSRN, 2022.

Min Liu, Wei-Chong Choo, Chi-Chuan Lee, and Chien-Chiang Lee. Trading volume and realized volatility forecasting: Evidence from the china stock market. *Journal of Forecasting*, 42(1):76–100, 2023.

Chuong Luong and Nikolai Dokuchaev. Forecasting of realised volatility with the random forests algorithm. *Journal of Risk and Financial Management*, 11(4):61, 2018.

Michael McAleer and Marcelo C Medeiros. Realized volatility: A review. *Econometric reviews*, 27(1-3):10–45, 2008.

Ulrich A Müller, Michel M Dacorogna, Rakhal D Davé, Olivier V Pictet, Richard B Olsen, and J Robert Ward. *Fractals and intrinsic time-a challenge to econometricians*. SSRN, 2008.

Stefan Nagel. *Machine learning in asset pricing*, volume 1. Princeton University Press, 2021.

Andrew J Patton. Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, 160(1):246–256, 2011.

Andrew J Patton and Kevin Sheppard. Evaluating volatility and correlation forecasts. In *Handbook of financial time series*, pages 801–838. Springer, 2009.

Andrew J Patton and Kevin Sheppard. Good volatility, bad volatility: Signed jumps and the persistence of volatility. *Review of Economics and Statistics*, 97(3):683–697, 2015.

Mark Podolskij and Mathias Vetter. Bipower-type estimation in a noisy diffusion setting. *Stochastic processes and their applications*, 119(9):2803–2831, 2009.

Eghbal Rahimikia and Ser-Huang Poon. Machine learning for realised volatility forecasting. *Available at SSRN*, 3707796, 2020.

Rafael Reisenhofer, Xandro Bayer, and Nikolaus Hautsch. Harnet: A convolutional neural network for realized volatility forecasting. *arXiv preprint arXiv:2205.07719*, 2022.

Norman R Swanson and Halbert White. Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models. *International journal of Forecasting*, 13(4):439–461, 1997.

Neda Todorova and Michael Souček. The impact of trading volume, number of trades and overnight returns on forecasting the daily realized range. *Economic modelling*, 36:332–340, 2014.

Chao Zhang, Yihuang Zhang, Mihai Cucuringu, and Zhongmin Qian. Volatility forecasting with machine learning and intraday commonality. *Journal of Financial Econometrics*, 22 (2):492–530, 2024.

Lan Zhang. Efficient estimation of stochastic volatility using noisy observations: A multi-scale approach. *Bernoulli*, 12(6):1019–1043, 2006.

Lan Zhang, Per A Mykland, and Yacine Aït-Sahalia. A tale of two time scales: Determining integrated volatility with noisy high-frequency data. *Journal of the American Statistical Association*, 100(472):1394–1411, 2005.