# European Volatility Under the Microscope

## Applied Machine Learning for Volatility Forecasting

Tobias Persson[†] Anton Johanson[‡]

Master's Thesis in Finance

Stockholm School of Economics

May 2026

**Abstract**

This paper examines the out-of-sample forecasting performance of twenty model specifications, ranging from HAR-type benchmarks, regularized linear models, and tree-based ensembles to feed-forward neural networks, in predicting one-day-ahead realized volatility on the EURO STOXX 50 index. Using high-frequency intraday trading data spanning the period from January 1999 to August 2020, we compare model performance across a parsimonious set of lagged realized variance components and an extended set augmented with macroeconomic and financial predictors motivated by our European setting. Forecasts are evaluated under both MSE and QLIKE loss functions. We find that the macro-financial predictors enable machine learning models to exploit additional signals, but the gains differ markedly across model classes and loss functions. Tree-based ensembles improve forecast accuracy under both losses, whereas regularized linear models and neural networks deliver gains under MSE only. Two specifications emerge as robust across feature sets and loss functions; HARQ on the parsimonious feature set, and Random Forest on the extended feature set. We identify the lagged realized variance components and the VSTOXX index as the main drivers for predicting next-day volatility across all four model categories.

*Keywords:* Volatility Forecasting, High-frequency, Machine Learning, EURO STOXX 50

**Supervisor:** Tobias Sichert, Department of Finance, Stockholm School of Economics
**Examinator:** Gualtiero Azzalini, Department of Finance, Stockholm School of Economics
**JEL Classification:** C10, C50

**Acknowledgment:** We would like to extend our sincere gratitude to our supervisor Tobias Sichert for his valuable feedback and continued support throughout the development of this degree project.

[†]tobias.ct.persson@outlook.com
[‡]anton13johanson@gmail.com

# Table of Contents

# 1 Introduction

One of the central stylized facts of financial markets is that, although asset returns are essentially unpredictable, return volatility exhibits strong persistence, clustering, and long-term memory behavior. As a result, stock market volatility, approximated by observable squared returns, is highly predictable over time. Motivated by these properties, an extensive body of literature has sought to better understand the realized volatility dynamics that prevail across financial markets.

Volatility forecasting sits at the center of modern risk management, derivatives pricing, and portfolio allocation, since more accurate and precise volatility forecasts translate into better calibrated risk models and more informed capital investments. The origins of the literature trace back to the original works of Engle (1982) and Bollerslev (1986), who introduced the initial ARCH and GARCH models, respectively. Over time, increased access to high-frequency trading data and greater computational power have allowed practitioners to expand this line of work, leading to new specifications such as the Heterogeneous Autoregressive (HAR) set of models initially proposed by Corsi (2009). While the standard HAR framework and its extensions constitute a salient benchmark in volatility forecasting, modern software programs have enabled more advanced machine learning (ML) algorithms to capture nonlinearities and complex relationships among correlated variables, which conventional linear models in the forecasting literature may not detect.

In light of this development, Christensen et al. (2023) provide a comprehensive comparison of conventional linear HAR benchmark models against ML methods, evaluating regularized linear models, tree-based ensembles, and feed-forward neural networks on the individual constituents of the Dow Jones Industrial Average (DJIA). Their main findings are that the forecasting performance of the ML models depends heavily on the set of predictors and conclude that the Random Forest (RF) model and the Neural Networks (NNs) are the preferred ML algorithms in their setting. Subsequent work by Díaz et al. (2024), who examine the S&P 500, reaches a similar conclusion, while broader surveys (Gunnarsson et al. (2024), Mansilla-Lopez et al. (2025)) confirm that most of the literature on volatility forecasting remains concentrated on US equity markets.

The evident lack of research on European equity markets is notable, given that the macro-financial structure of the Eurozone differs from that of the US in ways that directly influence volatility dynamics. The EURO STOXX 50 index, which constitutes the principal large-cap benchmark in the Euro area and underlying asset for the VSTOXX implied volatility index, reflects a multi-sovereign environment characterized by heterogeneous fiscal conditions, sovereign risk differentials, and varying degrees of financial fragmentation (Lane (2012), Corsetti et al. (2013)). In contrast to the single-index structure of the relatively more integrated US system, volatility dynamics in European equity markets is likely to also be shaped by a wide range of supranational factors, including monetary policy by the European Central Bank (ECB) (Bohl et al. (2008)), systemic financial stress as measured by the CISS index (Hollo et al. (2012)), sensitivity to energy

price fluctuations due to the region's status as a net energy importer (Katsampoxakis et al. (2022)), and spillovers from preceding US and subsequent Asian trading sessions (Golosnoy et al. (2015), Wang et al. (2020)). Taken together, these structural differences imply that volatility forecasting models designed for US markets may not directly apply to the European setting without explicitly accounting for these additional factors that are particularly relevant in the European context.

To the best of our knowledge, no prior study has systematically compared the HAR benchmark family with modern ML approaches on the EURO STOXX 50 index. This paper seeks to fill that gap. We extend the methodological framework of Christensen et al. (2023) to a single-index European setting and evaluate twenty model specifications, ranging from HAR benchmark models, to regularized linear methods, tree-based ensembles, and feed-forward neural networks, for one-day-ahead forecasting of realized volatility on the EURO STOXX 50 index. Our final clean sample covers the period from January 1999 to August 2020, thereby encompassing the Dot-com Bubble (2000-2002), the Global Financial Crisis (GFC, 2008-2009), the European Sovereign Debt Crisis (Euro Crisis, 2010-2012), a prolonged low-volatility expansion period (2016-2019), and the Covid-19 market turmoil of March 2020. We compare a parsimonious feature set restricted to the standard lagged realized variance components against an extended set augmented with macroeconomic and financial predictors specific to the European context. We evaluate performance under both the Mean Squared Error (MSE) and Quasi-Likelihood (QLIKE) loss functions, with statistical significance assessed by the Diebold and Mariano (1995) test.

Three main research questions guide the analysis of our paper. First, to what extent does the inclusion of additional macroeconomic and financial predictors impact the forecasting performance of ML models? Second, which specific model, or class of models, most accurately captures the realized volatility dynamics specific to the EURO STOXX 50 index? Third, what variables drive predictive performance across model categories, and is there agreement across model families on the dominant predictors?

Our findings are threefold. First, the extended feature set expands the set of models that significantly outperform the HAR benchmark model under MSE from two to nine, but these gains do not translate uniformly to QLIKE. Tree-based ensembles improve forecast accuracy under both loss functions, whereas regularized linear models, and neural networks deliver gains under MSE only. Second, two models emerge as robust across feature sets and loss functions; HARQ on the parsimonious set, and Random Forest on the extended feature set. Third, our analysis of variable importance reveals broad agreement across model families on the identity of the dominant predictors, with lagged realized variance components and the VSTOXX index emerging as the most informative drivers for predicting next-day volatility.

The remainder of our paper is structured as follows. Section (2) provides a chronological overview of the literature on volatility forecasting. Section (3) describes the data collection and cleaning process, while also motivating the selection of variables in our

two feature sets. Section (4) details the methodological framework, forecasting models, and performance evaluation metrics used in this empirical study. Section (5) presents the empirical results, Section (6) benchmarks our findings in light of our three research questions, and Section (7) provides concluding remarks.

# 2 Literature Review

This section reviews the literature on volatility forecasting by presenting a chronological overview of model development. We begin by introducing the autoregressive models and gradually transition towards modern approaches based on high-frequency data, including linear HAR models and novel ML techniques. The section concludes by examining the existing empirical evidence for volatility forecasting on European equity markets and outlines how our study contributes to the existing strand of literature.

## 2.1 From ARCH Models to Realized Volatility

The origins of the literature on volatility forecasting can be traced back to the contribution of Engle (1982), who introduced Autoregressive Conditional Heteroskedastic (ARCH) models. This framework allows the conditional variance to vary over time as a function of past squared residuals, while the unconditional variance remains constant. Building on this foundation, Bollerslev (1986) extended the ARCH model to the generalized ARCH (GARCH) framework to allow for a more flexible and parsimonious lag structure in which the conditional variance depends on both past squared residuals and past conditional variances. The GARCH framework, which has been shown to outperform high-order ARCH specifications, has since established itself as an important benchmark in forecasting.

Although widely applied in the risk management literature, standard GARCH models and their numerous extensions (see Glosten et al. (1993), Nelson (1991), Engle and Ng (1993), Zakoian (1994), Baillie et al. (1996), Bollerslev and Mikkelsen (1996), Engle et al. (2013)), are typically estimated using daily return data. As a result, they disregard the vast amount of information contained in intraday price movements by treating each trading day as a single observation rather than as a series of smaller movements. This aggregation leads to a loss of information and limits the ability to fully capture the underlying volatility dynamics. Recognizing these limitations, Andersen et al. (2001) proposed exploiting high-frequency intraday price data to construct realized volatility as a model-free and consistent estimator of the latent return variation. In contrast to parametric volatility models, realized volatility can be directly computed from readily available price data, thus providing a more accurate measure of volatility. While traditionally exhibiting a highly skewed pattern, Andersen et al. (2001) show that the logarithmic transformation of realized volatility is approximately normally distributed and exhibits strong persistence, which makes it well-suited for standard time series modeling. Expanding on these

insights, Andersen et al. (2003) demonstrate that relatively simple linear models applied to realized volatility outperform traditional GARCH and stochastic volatility models in forecasting foreign exchange rates. As such, these findings highlight the importance of high-frequency data and mark a seismic shift in the literature by viewing volatility as something measurable rather than latent.

## 2.2 The HAR Model and Related Extensions

Following the seminal contributions of Andersen et al. (2001) and Andersen et al. (2003), which established realized volatility as a persistent and informative predictor of the true underlying volatility, Corsi (2009) introduced the heterogeneous autoregressive (HAR) model as a parsimonious and intuitive framework for volatility forecasting. The model is grounded in the Heterogeneous Market Hypothesis (HMH) of Müller et al. (1997), which asserts that financial markets consist of heterogeneous agents operating at different time horizons rather than a homogeneous group of investors, and it captures the long-memory behavior of realized volatility by approximating it through an additive cascade of volatility components measured over daily, weekly, and monthly horizons. Despite its simplicity, the HAR model effectively mimics the persistence observed in financial market volatility and it has been shown to outperform complex autoregressive models with long lag structures.

Building on this framework, a large body of literature has proposed extensions to the standard HAR model to account for additional stylized facts of financial markets. One common extension is the logarithmic transformation (LogHAR), proposed by Corsi et al. (2012), which improves the statistical properties of realized volatility by reducing skewness and bringing it closer to a normal distribution. Other extensions focus on capturing more salient volatility dynamics. For example, Corsi and Renò (2012) incorporate a persistent leverage effect into the HAR model (LevHAR), capturing the asymmetric effect of negative returns on future volatility. Similarly, Patton and Sheppard (2015) address potential asymmetries by proposing the semivariance HAR (SHAR) model, which decomposes realized volatility into its positive and negative return components. Their findings indicate that negative semivariance carries greater predictive power, suggesting that negative returns have a stronger and more persistent impact on future volatility than its positive counterpart. More recently, Bollerslev et al. (2016) extend the HAR framework by accounting for the time-varying measurement error in realized volatility through realized quarticity. The resulting HARQ model, which adjusts the parameters depending on the level of estimation uncertainty in the realized volatility measure, demonstrates greater persistence during tranquil periods and faster mean reversion during turbulent periods, with meaningful improvements in forecast accuracy compared to the baseline HAR model where parameters are held constant. Taken together, these extensions highlight the flexibility of the HAR framework in capturing key features observed in financial data while maintaining a structure that is relatively easy to implement.

## 2.3 Macroeconomic and Financial Predictors of Volatility

A related strand of the literature examines the extent to which stock market volatility is driven by broader economic conditions, and whether the predictive performance of HAR-type models can be improved by incorporating additional sources of information beyond the lagged volatility components. This line of research complements the autoregressive nature of standard HAR models by emphasizing the role of macroeconomic and financial factors in governing volatility dynamics. In particular, Paye (2012) shows that the persistence observed in realized volatility can be more accurately modeled by conditioning on macroeconomic and financial variables, which suggests that financial market volatility is not only driven by its own past but also by underlying economic fundamentals. This is in line with the findings in Christiansen et al. (2012), Engle et al. (2013), and Nonejad (2017), where macroeconomic indicators play a significant role in predicting future volatility. Extending this argument beyond traditional macroeconomic variables, Wang et al. (2018) demonstrate that crude oil volatility serves as a strong short-term predictor of stock market volatility. Their results suggest that augmenting autoregressive models with commodity-based measures can lead to meaningful improvements in forecast accuracy.

In addition to analyzing the impact of macroeconomic variables on financial markets, related studies also explore the reverse relationship. For instance, Choudhry et al. (2016) investigate how stock market volatility and real economic activity across the US, Canada, Japan, and the UK are interrelated. Using a combination of linear and nonlinear causality tests, they find that stock market volatility is a significant short-term predictor of the business cycle in all four countries, with more pronounced effects during periods of market turbulence. Moreover, their results document the presence of cross-country spillover effects, indicating that volatility and economic activity in the US can also influence economic fundamentals in related markets.

Collectively, this body of research suggests that the HAR family of models, on a standalone basis, may omit relevant information by only relying on lagged volatility components. As such, incorporating a broader set of macroeconomic and financial variables can improve the predictive performance of these models.

## 2.4 Machine Learning Approaches for Volatility Forecasting

The application of ML methods in volatility forecasting is motivated by the limitations of traditional linear models when dealing with high-dimensional datasets, complex interactions among predictors, and potential nonlinear relationships. As emphasized by Varian (2014), advances in data availability and computational power have made it possible to employ more flexible modeling techniques that, in contrast to simple linear models, are capable of capturing these patterns.

Traditional HAR models can be effectively extended by imposing regularization techniques, such as Ridge Regression (RR), the Least Absolute Shrinkage and Selection Oper-

ator (LASSO, LA), and the Elastic Net (EN). These methods address issues of overfitting and multicollinearity by selecting and shrinking the estimated coefficients of variables (Hastie (2009)). On this note, Audrino and Knaus (2016) investigate the ability of LA to recover the lag structure of the HAR model. They find that while LA can approximate the HAR model synthetically when it is assumed to be the true model, it struggles to do so when estimated on real data, which indicates that volatility dynamics may be more complex than what the standard HAR models suggests. In contrast, Ding et al. (2021) document that LA can outperform the HAR model in forecasting realized volatility, attributing this improvement in out-of-sample performance to its ability to account for a more flexible lag structure. Similarly, Wei et al. (2020) show that RR can improve forecasting accuracy by mitigating multicollinearity and reducing overfitting in models augmented with an extended set of predictors. In related work, Audrino et al. (2020) show that sentiment and attention measures, such as social media, news article, and search engine data, can enhance predictive performance when combined with regularization techniques. Extending this line of research, Zhang et al. (2019) compare forecast combination approaches by combining individual forecasts from HAR models with shrinkage methods, such as EN and LA, and find that the shrinkage methods outperform both the combination approaches and the individual HAR forecasts.

Tree-based models constitute another class of models designed to capture nonlinear relationships and interactions among predictors. This framework adopts a nonparametric approach that partitions the predictor set into smaller subsets, which allows it to generate forecasts without any information about the functional form (Leushuis and Petkov (2026)). For instance, Yang et al. (2017) apply Bagging (BG) techniques within the HAR framework to predict the volatility of commodity futures and find that augmenting HAR-type models with BG methods leads to improved forecasting accuracy. In related work, Luong and Dokuchaev (2018) employ Random Forests (RF) to high-frequency data from the S&P ASX 200 and document improved predictive performance for the HAR model when the RF model is applied. Similarly, Gupta and Pierdzioch (2022) show that RF enhances the accuracy of forecasts for crude oil volatility, particularly at longer horizons. Building on this tree-based approach, Mittnik et al. (2015) apply Gradient Boosting (GB) techniques and demonstrate that sequentially constructed decision trees, when used with a large set of financial and macroeconomic predictors, produce superior out-of-sample forecasts compared to GARCH-type benchmarks. Their findings further suggest that this class of models better captures the dynamics of realized volatility and the potentially nonlinear effects of macroeconomic and financial variables.

Advanced research also explores the use of Neural Networks (NNs) in volatility modeling, with mixed evidence about their predictive performance relative to linear benchmarks. As a start, Fernandes et al. (2014) find no support that NN-based models consistently outperform standard HAR specifications when forecasting the VIX, suggesting that increased model flexibility does not necessarily translate into improved predictive accuracy.

On a similar note, Vortelinos (2017) compare NNs with HAR, GARCH, and principal component-based approaches, and report only modest improvements from NNs compared to simpler specifications, indicating that the long-memory property of realized volatility is well-captured by autoregressive linear benchmarks. In contrast, other studies document more favorable results. Miura et al. (2019) show that artificial NNs can improve predictions of realized volatility in cryptocurrency markets, where volatility dynamics are highly nonlinear and subject to large jumps. Building on this argument, Bucci (2020) constructs NN models that incorporate a comprehensive set of macroeconomic and financial predictors and show that these models are able to outperform linear benchmarks, including the HAR model. Importantly, these results indicate that NNs perform relatively better during periods characterized by financial distress, suggesting that realized volatility may display a stronger nonlinear pattern during turbulent times and that this behavior is more accurately captured by flexible modeling approaches.

A comprehensive comparison of ML models and the linear HAR framework is provided by Christensen et al. (2023), which serves as the primary reference paper of our study. The authors evaluate the ML methods described above against the HAR family in forecasting realized volatility, examining both a restricted set of lagged volatility components and an extended feature set that incorporates firm-specific and macroeconomic variables. Their findings highlight that the relative performance of ML models is highly dependent on the information contained within the prediction set. While traditional HAR-type models remain competitive in parsimonious settings, more flexible modeling techniques, such as regularization methods and NNs, deliver significant improvements in forecasting accuracy when a large set of predictors is included. While tree-based models exhibit weaker performance when only lagged volatility components are included, their predictive accuracy increases when incorporating an expanded set of predictors, thus reflecting their ability to capture nonlinearities and interactions.

These results are broadly consistent with the findings of Díaz et al. (2024), who conduct a similar evaluation of ML models and HAR-type specifications in forecasting S&P 500 realized volatility with a comprehensive set of macroeconomic and financial predictors. Their analysis confirms that ML approaches can improve predictive performance, especially in a high-dimensional setting, while also emphasizing that the quality of the feature set is crucial. In particular, they find that variables capturing financial and macroeconomic uncertainty, such as the VIX index, are the main drivers of future volatility, further emphasizing that improvements in forecasting performance of ML models over the HAR benchmark are primarily attributable to the quality of the predictors in the feature set.

## 2.5 Volatility Forecasting in European Markets

While volatility forecasting has been extensively studied in the US context, research on European equity markets remains relatively limited and the existing evidence is fragmented. For example, Harrison and Moore (2012) apply traditional AR and MA models to fore-

cast volatility in Central and Eastern European markets, primarily focusing on emerging economies rather than developed Western markets. In another study, Cipollini and Gallo (2019) decompose the realized volatility of five major European equity indices into common and idiosyncratic components, and examine how these collectively contribute to the volatility of the EURO STOXX 50. More recent work has explored the use of ML models in a European context. Parra-Domínguez and Sanz-Martín (2024) assess the ability of different NN architectures to predict EURO STOXX 50 index levels. Their focus, however, is on price prediction rather than volatility forecasting, and they do not compare the effectiveness of ML methods with traditional linear models.

Taken together, the literature on European volatility forecasting remains inconclusive, and, to the best of our knowledge, no prior study has systematically compared the HAR framework with novel ML techniques. Furthermore, the influence of macroeconomic and financial variables on volatility dynamics in Europe remains underexplored. Against this backdrop, our study seeks to bridge this gap by delivering a comprehensive evaluation of twenty distinct model specifications for forecasting one-day-ahead realized volatility on the EURO STOXX 50 index, while simultaneously shedding new light on the predictive role of macroeconomic and financial predictors in the European context.

# 3 Data

In this section, we provide an overview of the data gathering and preparation process that we conduct prior to forecasting realized volatility. We begin by explaining the data cleaning process, after which we proceed to present the training-validation-test split, along with the motivation for the construction of our two feature sets. We round-off this section by presenting summary statistics of all variables included in our study.

## 3.1 Data Collection and Preparation

The empirical analysis in this paper is based on 1-minute intraday trading data for the EURO STOXX 50 index, obtained from the Stockholm School of Economics. Prior to data cleaning, the sample spans the period from January 1998 to August 2020. The EURO STOXX 50 index is composed of the 50 largest and most liquid companies in the Eurozone, and it encompasses a broad range of industries, see Appendix (8.1) for an industry and country composition. As such, it is widely regarded as the European equivalent to the S&P 500 index and serves as a natural benchmark for assessing the performance of European equity markets.

Prior to model estimation, we perform a comprehensive data cleaning process on the initial dataset. We begin by excluding weekends and subsequently remove trading days with more than 50% missing intraday observations. This filtering process results in the exclusion of 164 trading days from the original sample due to insufficient intraday

price data. Importantly, these excluded trading days do not appear to be systematically concentrated in any particular period, suggesting that this data cleaning step is unlikely to introduce substantial selection bias to our dataset. Second, smaller data gaps are addressed by forward-filling missing prices within the trading day with the last available observation while remaining missing values at the start of the trading day are backward-filled using the first available price within that day. We note that missing price data appears to be idiosyncratic and reflects data quality issues as opposed to systematic deficiencies. Overall, this approach ensures a complete intraday price series, preserving days with only minor interruptions while excluding those with insufficient data to support reliable realized variance estimation. We further describe the construction of the intraday trading window in Subsection (3.2).

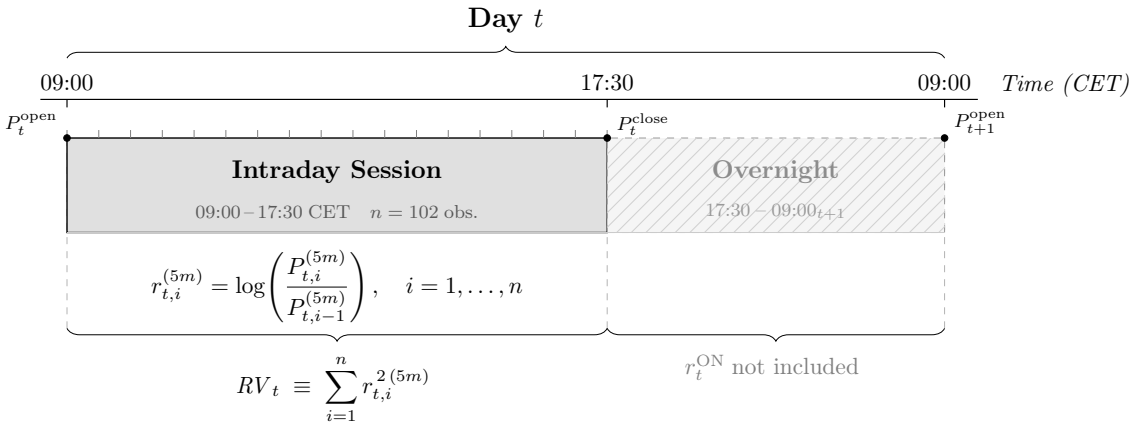
The final sample period is constrained by the availability of the VSTOXX index, which starts recording on January 5th, 1999. Consequently, our final dataset covers the period from 1999-01-05 to 2020-08-06 and includes a total of 5,489 trading day observations. It is important to note that our single-index analysis is based on the EURO STOXX 50 index rather than its individual constituents. As a result, the data is subject to periodic index rebalancing. While working with an index implies that the series captures the evolving structure of European equity markets, it may also absorb discontinuities and jumps around rebalancing dates.

## 3.2 Trading Window

Official sources confirm that the trading window of the EURO STOXX 50 index is active from 09:00 to 17:30 CET (STOXX Ltd. (2025)). While performing diagnostic checks on the intraday data, a sharp decline in trading activity is observed after 17:30, although some observations are still recorded after this mark. These post-close entries likely stem from closing auction mechanics, index adjustments, recalculations, and data vendor artifacts. Given that the data is structurally different outside official opening hours, we restrict our analysis to the regular intraday trading window from 09:00 to 17:30. This approach is consistent with related studies, including Christensen et al. (2023).

While index levels are recorded at 1-minute intervals, we follow standard practice in the volatility forecasting literature and construct 5-minute log returns before proceeding with the realized variance computation. This adjustment mitigates the impact of microstructure noise associated with 1-minute intraday data, such as order imbalances, bid-ask bounces, and related market frictions, which can lead to biased volatility estimates. We illustrate the construction of the intraday trading window in Figure (1).

**Figure 1:** Intraday Trading Window and Realized Variance Construction




*Notes:* The figure illustrates the daily sampling window used to construct the realized variance ($RV_t$) estimator on day $t$ for the EURO STOXX 50 index. The intraday session spans 09:00–17:30 CET, yielding $n = 102$ non-overlapping 5-minute log-returns per trading day. Using 5-minute intervals as opposed to 1-minute intervals is motivated to mitigate the adverse effects of market microstructure noise that inflate higher-frequency estimators. Note that the overnight return ($r_t^{ON}$) is excluded from the estimator, as it is driven predominantly by non-trading-hour noise rather than a continuous process. See Section (4) for further information on this derivation.

## 3.3 Training-Validation-Test Split

Table (1) reports the sequential time series split of the data into the training (70%), validation (10%), and test sets (20%). This percentage split for each partition is in line with the broad forecasting literature, including Christensen et al. (2023). The training set is used for model estimation, the validation set for hyperparameter tuning where applicable, and the test set is reserved for out-of-sample performance evaluation.

**Table 1:** Training, Validation, and Test Set Partition

| | | **Splits** | | |
|---|---|---|---|---|
| | **Full Sample** | **Training** | **Validation** | **Test** |
| Share of Sample | 100% | 70% | 10% | 20% |
| Observations | 5,489 | 3,842 | 548 | 1,099 |
| Start Date | 1999-01-05 | 1999-01-05 | 2014-02-27 | 2016-04-21 |
| End Date | 2020-08-06 | 2014-02-26 | 2016-04-20 | 2020-08-06 |

*Notes:* The table reports the partition of the cleaned sample (5,489 trading days) into training, validation, and test sets. The training set is used for parameter estimation, the validation set for hyperparameter selection in the regularized linear models, Gradient Boosting, and neural network specifications, and the test set for out-of-sample forecast evaluation. For the HAR benchmark models, bagging, and random forest, the training and validation sets are merged and parameters are re-estimated daily via a rolling window, consistent with Christensen et al. (2023).

## 3.4 Feature Sets

As discussed in previous sections, the selection of explanatory variables is a key determinant of predictive performance. While the traditional HAR framework relies on a parsimonious set of lagged volatility measures realized over different time horizons, recent advances in ML have enabled the inclusion of a broader range of macroeconomic and financial variables to improve forecasting performance.

To examine how the information set influences forecasting performance, we follow Christensen et al. (2023) and construct two distinct feature sets. The first, denoted $\mathcal{M}_{\text{HAR}}$, contains only the core variables of the HAR framework, that is the daily, weekly, and monthly realized variance components. Model-specific extensions to the standard HAR model, such as realized quarticity (HARQ) and leverage effect components (Lev-HAR and SHAR), are included only in their respective model specifications. All models considered in this study are first evaluated using only this restricted feature set. The second feature set, denoted $\mathcal{M}_{\text{ALL}}$, augments the baseline $\mathcal{M}_{\text{HAR}}$ set with a broader array of macroeconomic and financial predictors relevant for forecasting realized volatility in a European setting. This expanded information set enables more flexible ML models to capture nonlinear relationships and interactions among predictors, which the HAR lineage does not necessarily accommodate.

A comparison of forecasting performance across these two feature sets, ceteris paribus, allows us to investigate whether improvements in predictive accuracy are primarily attributable to model design or the inclusion of additional informative predictors.

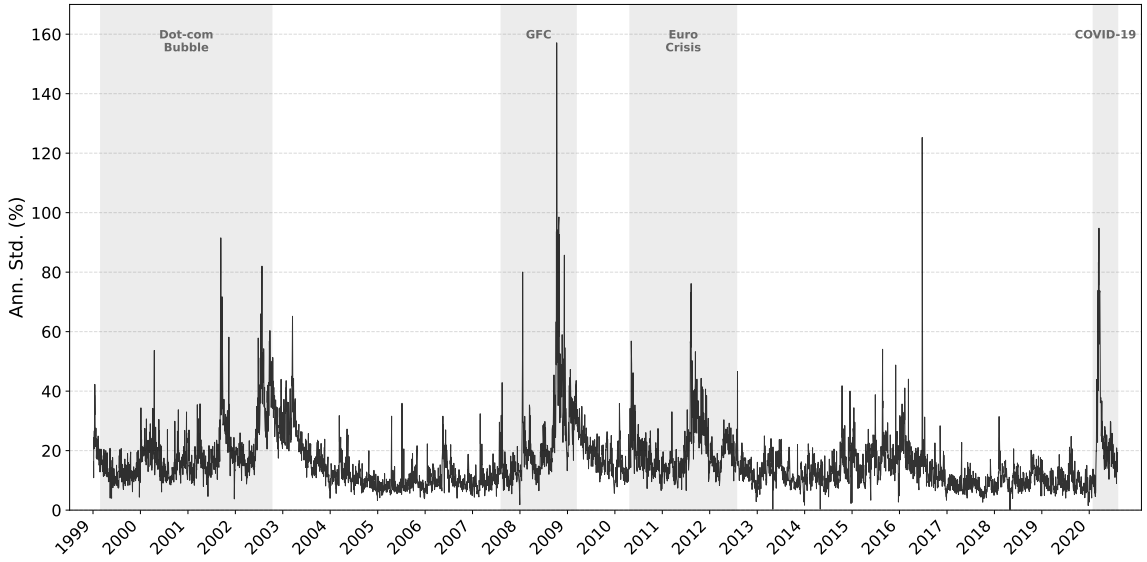
### 3.4.1 Construction of the $\mathcal{M}_{\text{HAR}}$ Feature Set

As noted above, the $\mathcal{M}_{\text{HAR}}$ feature set consists of the standard HAR components, that is daily, weekly, and monthly realized volatility measures. In addition, it includes variables specific to each extension of the HAR framework considered in this study.

All models considered in this study are first evaluated using this restricted feature set. This setup enables a direct comparison between traditional HAR-type models and more flexible ML approaches when estimated on identical information sets, thereby isolating the effect of model complexity.

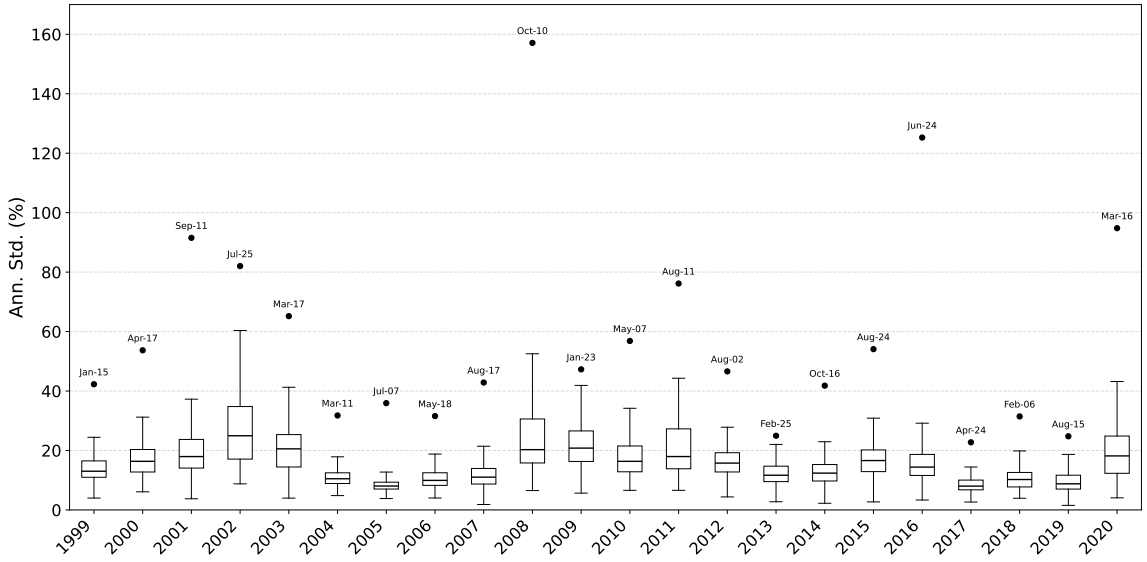
All variables used to construct $\mathcal{M}_{\text{HAR}}$ are derived from the same intraday trading set, sourced from the Stockholm School of Economics. For illustrative purposes, the distribution of annualized daily realized volatility over the full sample period is presented in Figures (2) and (3).

**Figure 2:** Annualized Daily Realized Volatility (1999 - 2020)




*Notes:* The figure plots annualized daily realized volatility for the EURO STOXX 50 index over the sample period January 1999 - August 2020. Shaded regions indicate periods of elevated market stress identified as distinct volatility regimes: the Dot-com Bubble (March 1999 - October 2002), the Global Financial Crisis (GFC, August 2007 - March 2009), the European Sovereign Debt Crisis (Euro Crisis, April 2010 - July 2012), and the Covid-19 pandemic (February 2020 - August 2020). To annualize the numbers, we assume an average of 252 trading days per year.

**Figure 3:** Distribution of Annualized Daily Realized Volatility (1999 - 2020)




*Notes:* The figure illustrates a box plot of the annual distribution of daily realized volatility for the EURO STOXX 50 index over the sample period January 1999 - August 2020. Each box displays the interquartile range of daily annualized realized volatility within a given year, with the horizontal line in the box denoting the median. The dot above each box marks the single most volatile trading day of the year, with the corresponding date indicated above. For descriptive details of each outlier event (see Appendix (8.3)). Note that the box for 2020 only includes data from January till August which marks the end of our sample period. To annualize the numbers, we assume an average of 252 trading days per year.

### 3.4.2 Construction of the $\mathcal{M}_{\text{ALL}}$ Feature Set

While the $\mathcal{M}_{\text{HAR}}$ feature set is restricted to realized measures of volatility derived from high-frequency intraday data, the $\mathcal{M}_{\text{ALL}}$ feature set expands this information by incorporating a broad range of macroeconomic and financial predictors. As highlighted by

previous studies, such an enriched information set can allow ML models to exploit a wider range of economic signals as well as potential nonlinearities and interactions among predictors. This, in turn, provides a more comprehensive framework for forecasting future volatility in a European setting. The expanded set of predictors used in the $\mathcal{M}_{\text{ALL}}$ feature set is justified in the following subsections.

***1. Endogenous Volatility and Risk Expectations***

The VSTOXX index measures the implied volatility of the EURO STOXX 50 index based on option prices with a 30-day maturity. As a forward-looking indicator of market implied uncertainty, it captures the compensation investors require in return for bearing volatility risk and it is therefore widely used in the literature as a proxy for investor sentiment and market fear in European equity markets. The inclusion of the VSTOXX index is further motivated by the well-documented leverage effect and volatility feedback mechanism, whereby an increase in implied volatility is associated with higher realized volatility in subsequent periods. Moreover, Christensen et al. (2023) show that implied volatility to a large extent captures the information contained in historical realized volatility for forecasting. In their analysis, the VIX index is included in levels as a predictor for future stock market volatility, which is in line with the specification adopted by Díaz et al. (2024). Building on this argument, Mittnik et al. (2015) also identify the VIX as one of the primary drivers of realized volatility.

***2. Macroeconomic and Systemic Risk Indicators***

Following Christensen et al. (2023) and Díaz et al. (2024), we also include the Economic Policy Uncertainty (EPU) index as a predictor, originally developed by Baker et al. (2016). The index is constructed by counting the frequency of articles in major European newspapers that jointly contain terms related to economic policy and uncertainty in the Eurozone. These counts are normalized by the total number of articles to ensure comparability over time and across sources. Given that the index is available at a monthly frequency, we employ a one-month lag to avoid look-ahead bias. The use of the EPU index as a predictor of realized volatility is further supported by the work of Audrino et al. (2020), who show that indicators derived from social media, news articles, and search engine data contain predictive information that can be used to improve forecast accuracy.

In addition to the EPU index, we extend the set of macroeconomic and financial predictors by incorporating the Composite Indicator of Systemic Stress (CISS) index developed by Hollo et al. (2012) and provided by the ECB. The CISS index is a daily measure of systemic risk in the European financial system, constructed by aggregating stress signals across multiple market segments, including equity, bond, money, and foreign exchange markets.

***3. Exchange Rate Dynamics***

We extend the framework of Christensen et al. (2023) by incorporating exchange rate

dynamics through the EUR/USD exchange rate, following the broader macro-financial predictor sets considered in Díaz et al. (2024) and Christiansen et al. (2012). In an open economy such as the Eurozone, which is characterized by monetary and financial integration, exchange rate fluctuations capture changes in relative monetary conditions, global risk sentiment, and international capital flows. All of these factors can affect volatility. Moreover, Cipollini and Gallo (2019) show that European equity volatility is largely driven by common factors shared across countries, indicating that countries within the Euro area exhibit similar underlying dynamics.

***4. Commodity and Inflation-linked Factor***

We further extend the set of exogenous predictors by including the European brent crude oil price as a proxy for global commodity and inflation-linked shocks. This choice is motivated by the evidence in Wang et al. (2018), who show that crude oil volatility contains significant predictive information for stock market volatility, particularly at short horizons. Their findings suggest that commodity price dynamics capture broader macroeconomic conditions, including inflationary pressure and global demand fluctuations, which are not fully reflected in traditional volatility models. In a European context, this measure is particularly relevant given the region's status as a net importer of energy and subsequent reliance on global energy markets. As such, fluctuations in oil prices can influence both corporate earnings and perceptions of market uncertainty.

***5. Global Market Spillovers***

To account for global spillover effects, we extend the contribution of Christensen et al. (2023) by incorporating equity market returns from two major international markets. This is motivated by the central position of European markets within the global trading day, where information from earlier trading activity in the US and next-day trading in Asia may influence European volatility dynamics. The inclusion of spillover effects is also supported by Choudhry et al. (2016), who document strong cross-country effects between stock market volatility and economic fundamentals. Their findings highlight that financial market activity in the US can affect real economic activity in other parts of the world, suggesting that stock market volatility can be transmitted across markets. For this reason, we include daily returns from the Nikkei 225 and the S&P 500 index to account for spillovers from Asian and US markets respectively. Due to Europe's position in the global trading window, S&P 500 returns are lagged by one day to ensure that only information available at the time of forecasting is used. We also proceed to forward-fill prices of the respective indices during American and Japanese holidays.

***6. Monthly and Weekly Momentum***

We further augment the $\mathcal{M}_{\text{ALL}}$ feature set by including measures of short and medium term market momentum for the EURO STOXX 50 index. While Christensen et al. (2023) consider only a weekly momentum variable to capture recent price fluctuations of individ-

ual DJIA constituents, we extend this approach by incorporating both weekly and monthly momentum factors in our single-index setting. The inclusion of a monthly momentum factor for volatility forecasting is further motivated by Díaz et al. (2024), who show that it contains predictive information for ML models in general, and the LA method in particular. Both momentum measures are constructed from daily closing prices as cumulative log returns over the past 5 and 22 trading days, respectively.

***7. Term and Credit Spread***

To better capture expectations about future macroeconomic conditions, we extend the analysis of Christensen et al. (2023), who include the US 3-month treasury rate as a predictor, by incorporating measures of term structure dynamics and interbank credit risk in the Euro area. More specifically, we construct the term spread, measured as the difference between the 10-year euro interest rate swap and the 3-month EURIBOR, and the credit spread, approximated as the spread between the 3-month EURIBOR and the 3-month Euro Overnight Index Swap (OIS) rate.

The inclusion of these variables is motivated by the broad empirical literature emphasizing the role of yield curve slopes and credit conditions in forecasting stock market volatility. Several studies, including Díaz et al. (2024) and Mittnik et al. (2015) incorporate term, default, and TED spreads as predictors. In particular, Mittnik et al. (2015) document the TED spread, defined as the spread between LIBOR and T-Bill rate and analogous to the EURIBOR-OIS spread, as one of the most important drivers of the six-month-ahead S&P 500 volatility. On a similar note, Díaz et al. (2024) identify the default spread as an important predictor of one-month ahead S&P 500 volatility for several ML methods, including RR, EN, RF, GB and NNs. Moreover, Christiansen et al. (2012) find that the TED spread is one of the most informative predictors of volatility, while term and default spreads also contribute to improved forecast accuracy. Other studies, including Paye (2012), find that credit related proxies, such as the commercial paper-to-Treasury spread and the default spread, significantly improve volatility forecasts, whereas the impact of the term spread is less pronounced.

## 3.5 Summary Statistics

In the following subsection, we present summary statistics for all variables encompassed by the $\mathcal{M}_{\text{HAR}}$ and $\mathcal{M}_{\text{ALL}}$ feature sets for the entire sample period from 1999-01-05 to 2020-08-06. The summary statistics are presented in Table (2). Further information, particularly pertaining to the autocorrelation structure of the the realized variance components and the correlation matrix for the full predictor set is provided in Appendix (8.2). We also include the indexed evolution of the macroeconomic predictors in the $\mathcal{M}_{\text{ALL}}$ feature set in Appendix (8.4). Finally, a description of all exogenous variables, along with their respective definition, transformation, and source, is provided in Appendix (8.6).

**Table 2:** Summary Statistics of the $\mathcal{M}_{\text{HAR}}$ and $\mathcal{M}_{\text{ALL}}$ Feature Sets

| | Mean | Std. Dev. | Skew. | Exc. Kurt. | Min | Median | Max |
|---|---|---|---|---|---|---|---|
| **Panel A: $\mathcal{M}_{\text{HAR}}$ Feature Set** | | | | | | | |
| $RVD$ | 16.14 | 9.96 | 3.17 | 20.33 | 0.15 | 13.76 | 157.11 |
| $RVW$ | 16.51 | 9.33 | 2.62 | 11.04 | 4.52 | 14.18 | 93.58 |
| $RVM$ | 16.83 | 8.75 | 2.15 | 6.46 | 5.29 | 14.40 | 71.78 |
| $RVD^{+}$ | 11.22 | 6.79 | 2.77 | 13.51 | 0.09 | 9.54 | 83.33 |
| $RVD^{-}$ | 11.37 | 7.66 | 3.80 | 32.94 | 0.12 | 9.50 | 138.64 |
| $RD^{-}$ | −0.44 | 0.80 | −3.37 | 18.27 | −10.54 | 0.00 | 0.00 |
| $RW^{-}$ | −0.21 | 0.38 | −3.83 | 25.44 | −5.41 | 0.00 | 0.00 |
| $RM^{-}$ | −0.10 | 0.19 | −3.51 | 17.55 | −1.91 | 0.00 | 0.00 |
| $RQ$ | 0.65 | 19.50 | 50.39 | 2,681.48 | 0.00 | 0.01 | 1,129.75 |
| **Panel B: Additional Variables Added to Create $\mathcal{M}_{\text{ALL}}$ Feature Set** | | | | | | | |
| $VSTOXX$ | 23.91 | 9.78 | 1.71 | 4.18 | 10.68 | 21.98 | 87.51 |
| $EUR/USD$ | 0.00 | 0.61 | 0.01 | 2.93 | −4.74 | 0.00 | 4.20 |
| $BRENT$ | 0.03 | 2.68 | −2.04 | 81.37 | −64.37 | 0.03 | 41.20 |
| $EPU$ | 4.93 | 0.45 | 0.00 | −0.90 | 4.07 | 4.99 | 6.07 |
| $CISS$ | 0.17 | 0.19 | 1.76 | 3.30 | 0.00 | 0.11 | 0.94 |
| $NIKKEI_{ret}$ | 0.01 | 1.44 | −0.38 | 6.81 | −12.11 | 0.00 | 13.23 |
| $S\&P500_{ret}$ | 0.01 | 1.25 | −0.37 | 10.72 | −12.77 | 0.06 | 10.96 |
| $NIKKEI_{sq}$ | 2.09 | 6.19 | 12.73 | 248.63 | 0.00 | 0.49 | 175.15 |
| $S\&P500_{sq}$ | 1.56 | 5.56 | 13.49 | 263.52 | 0.00 | 0.31 | 162.95 |
| $M1W$ | −0.00 | 3.12 | −0.75 | 5.54 | −27.07 | 0.24 | 16.47 |
| $M1M$ | 0.02 | 6.20 | −1.22 | 5.20 | −47.60 | 0.76 | 22.34 |
| $CREDIT$ | 0.20 | 0.26 | 2.88 | 10.33 | −0.31 | 0.09 | 1.98 |
| $TERM$ | 1.28 | 0.71 | −0.04 | −0.27 | −1.00 | 1.25 | 2.90 |

*Notes:* This table reports summary statistics for the full sample covering the period from 1999-01-05 to 2020-08-06 for both the $\mathcal{M}_{\text{HAR}}$ and the $\mathcal{M}_{\text{ALL}}$ feature sets. The variables $RVD$, $RVW$, $RVM$, $RVD^{+}$, and $RVD^{-}$ are presented as annualized volatility. Moreover, the variables $RD^{-}$, $RW^{-}$, $RM^{-}$, $EUR/USD$, $NIKKEI_{ret}$, $S\&P500_{ret}$, $BRENT$, $M1W$, $M1M$, $CREDIT$, and $TERM$ are shown as percentages. $RQ$ is scaled by $10^{6}$ while $NIKKEI_{sq}$ and $S\&P500_{sq}$ are scaled by $10^{4}$. The $VSTOXX$ and $CISS$ are shown in levels, and the $EPU$ index is shown as the one-month lagged log value of the index. The $CREDIT$ is the 3-month EURIBOR minus the 3-month OIS rate, and the $TERM$ is constructed as the difference between the 10-year Euro interest rate swap and the 3-month EURIBOR. See Tables (9) and (10) in Appendix (8.6) for further information.

# 4 Methodology and Empirical Design

This section outlines the methodological framework and empirical design on which the analysis in this paper is grounded in. We begin by introducing the concepts of realized variance and quarticity, which form the foundation of our volatility measures. We proceed to present the set of models employed for forecasting, detailing both the model categories and the individual model specifications.

## 4.1 Realized Variance and Realized Quarticity

We follow the methodological framework of Andersen et al. (2001) and Corsi (2009) in defining realized variance and constructing the lagged volatility components for the HAR specification. In line with this strand of literature, we assume that the underlying price of an asset $P_t$ follows a continuous diffusion process, which captures the stochastic and time-varying nature of financial markets. This process is defined in Equation (1)

$$d\log(P_t) = \mu_t\, dt + \sigma_t\, dW_t \tag{1}$$

where $\log(P_t)$ denotes the logarithm of the asset price at time $t$, $\mu_t$ is the time-varying drift component, $\sigma_t$ is the instantaneous volatility process, and $W_t$ is a standard Brownian motion. The primary objective of volatility forecasting is to model the latent integrated variance, which for a diffusion process is defined as the integral of the instantaneous variance over a given time interval. This quantity is expressed in Equation (2)

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

where $IV_t$ denotes the integrated variance over the interval $[t-1, t]$, corresponding to one trading day. Since the true latent variance $IV_t$ is not directly observable, it is estimated using high-frequency intraday data. Specifically, we partition each trading day into $n$ equally spaced intervals of length $\Delta$, such that $n = 1/\Delta$. Although 1-minute data is available, we follow the standard approach in the high-frequency literature and construct 5-minute intervals to mitigate microstructure noise, as shown in Equation (3).

$$r_{t,i}^{(5m)} = \log\left(\frac{P_{t-1+i\Delta}^{(5m)}}{P_{t-1+(i-1)\Delta}^{(5m)}}\right), \quad i = 1, \ldots, n \tag{3}$$

As explained by Andersen et al. (2001), the latent integrated variance can be consistently approximated by the sum of squared intraday returns when prices are sampled at a sufficiently high frequency. This estimator, referred to as realized variance, exploits the information contained in high-frequency data and provides a model-free estimate of the latent return variation. The realized variance can be defined according to Equation (4).

$$RV_t \equiv \sum_{i=1}^{n} \left( r_{t,i}^{(5m)} \right)^2 \tag{4}$$

To capture the persistence of volatility across multiple time horizons, we follow Corsi (2009) and construct aggregated measures of realized variance over weekly and monthly intervals. These components are defined as trailing averages of daily realized variance in Equation (5).

$$\overline{RV}_t^{(w)} = \frac{1}{5} \sum_{j=0}^{4} RV_{t-j} \qquad \overline{RV}_t^{(m)} = \frac{1}{22} \sum_{j=0}^{21} RV_{t-j} \tag{5}$$

Under standard regularity conditions, realized variance is a consistent estimator of the latent integrated variance as the sampling frequency increases ($\Delta \to 0$, equivalently $n \to \infty$). In practice, however, observations are recorded at finite frequencies, implying that the realized variance measure is subject to measurement error. Building on the framework of Bollerslev et al. (2016), we account for this measurement error by decomposing the realized variance of a given trading day into the sum of its latent component $IV_t$ and an idiosyncratic error term $\eta_t$, as illustrated in Equation (6)

$$RV_t = IV_t + \eta_t, \qquad \eta_t \mid IQ_t \sim \mathcal{N}(0,\ 2\Delta\, IQ_t) \tag{6}$$

where the variance of the measurement error depends on the latent integrated quarticity, defined according to Equation (7).

$$IQ_t = \int_{t-1}^{t} \sigma_s^4 \, ds \tag{7}$$

The integrated quarticity captures the variation in the underlying volatility process over time and determines the magnitude of the measurement error in realized variance. Periods characterized by more pronounced fluctuations (high $IQ_t$) are associated with noisier estimates of $IV_t$, whereas more tranquil periods (low $IQ_t$) result in more precise estimates. Given that $IQ_t$ is not directly observable, we follow the literature and use realized quarticity as a proxy for the latent variability measure. The realized quarticity can be defined according to Equation (8).

$$RQ_t \equiv \frac{n}{3} \sum_{i=1}^{n} (r_{t,i}^{(5m)})^4 \tag{8}$$

Similar to realized variance, the realized quarticity becomes a consistent estimator of the integrated quarticity as the sampling frequency increases. By relying on higher-order return moments, $RQ_t$ captures the intensity of intraday volatility fluctuations and provides a time-varying proxy for the uncertainty in $RV_t$. Consequently, this decomposition distinguishes between periods where realized variance is measured with high precision and periods characterized by noisy estimates.

## 4.2 HAR Benchmark Models

The linear benchmark models considered in our study are based on the Heterogeneous Autoregressive (HAR) framework. Building on the standard HAR specification, we also include a wide range of extensions that incorporate additional features observed in realized volatility dynamics, including the logarithmic transformation (LogHAR), leverage effect (LevHAR), semivariance decomposition (SHAR), and the time-varying measurement error (HARQ). These models form a uniform linear framework that incorporates different sources of information while maintaining a parsimonious structure and therefore constitutes natural benchmarks, against which more flexible ML methods can be evaluated.

The general linear equation for the standard HAR model, and its extensions, follow the structure of Equation (9), where $Z_t$ denotes the feature set used to predict the one-day-ahead realized variance $RV_{t+1}$. The predictor vector $Z_t \in \mathbb{R}^p$, where $p$ denotes the number of predictors, represents the variables included in the $\mathcal{M}_{\text{HAR}}$ or $\mathcal{M}_{\text{ALL}}$ feature sets.

$$RV_{t+1} = \beta_0 + \beta' Z_t + \varepsilon_{t+1} \tag{9}$$

To evaluate the out-of-sample forecasting performance of the benchmark models, we adopt a rolling-window estimation approach following Christensen et al. (2023). Specifically, each model is estimated via ordinary least squares (OLS) using a fixed-length window corresponding to the combined training and validation sample, comprising of 4,390 observations (see Table (1)). The estimation window is then rolled forward one day at a time across the test sample, and at each step, the model parameters are re-estimated using the information available up to time $t$ to generate one-day-ahead forecasts of realized variance, $RV_{t+1}$. Following the spirit of the insanity filter adoption by Bollerslev et al. (2016), negative forecasts in our study are replaced by the minimum realized variance observed within the corresponding rolling estimation window. This procedure is commonly applied in various versions in the literature to prevent unrealistic negative variance predictions.

### 4.2.1 HAR Model

The Heterogeneous Autoregressive (HAR) model is a parsimonious time-series model designed to capture the long-term memory behavior of realized variance using a simple linear regression framework. It was introduced by Corsi (2009) to model high-frequency based volatility measures such as realized variance.

Because investors react to information over different time scales according to HMH, volatility exhibits persistence across daily, weekly, and monthly horizons. The HAR model in Equation (10) captures this multi-scale structure by explicitly including volatility components measured at different aggregation levels, thereby providing a simple approximation to long-memory dynamics.

$$RV_{t+1} = \beta_0 + \beta_d RV_t + \beta_w \overline{RV}_t^{(w)} + \beta_m \overline{RV}_t^{(m)} + \varepsilon_{t+1} \tag{10}$$

In the equation above, $RV_t$ denotes the realized variance observed at time $t$, while $\overline{RV}_t^{(w)}$ and $\overline{RV}_t^{(m)}$ represent the weekly and monthly trailing averages of realized variance available at time $t$, derived earlier in Equation (5).

### 4.2.2 LogHAR Model

The LogHAR model, constructed by Corsi et al. (2012), is simply the logarithmic transformation of the original HAR specification, where the predictors are logarithms of the realized variance measures as opposed to levels. The LogHAR model is typically applied since realized variance is strictly positive, highly right-skewed, and exhibits strong heteroskedasticity. Estimating the model in logarithms stabilizes the variance of the series and reduces the influence of extreme volatility spikes, and it is defined in Equation (11).

$$\log(RV_{t+1}) = \beta_0 + \beta_d \log(RV_t) + \beta_w \log\left(\overline{RV}_t^{(w)}\right) + \beta_m \log\left(\overline{RV}_t^{(m)}\right) + \varepsilon_{t+1} \tag{11}$$

When the HAR model is estimated in logarithms, forecasts are produced for $\log(RV_{t+1})$. To transform these logarithmic predictions back to levels, we apply the standard log-normal bias adjustment using the residual variance from the rolling in-sample window, as shown in Equation (12).

$$\widehat{RV}_{t+1} = \exp\left(\widehat{\log(RV_{t+1})} + \frac{1}{2}\widehat{\sigma}^2\right) \tag{12}$$

### 4.2.3 LevHAR Model

While the standard HAR model captures volatility dynamics across multiple horizons, it does not account for the well-documented asymmetry whereby negative returns have a larger impact on future volatility than positive returns of similar magnitude. This phenomenon, commonly referred to as the leverage effect, implies that adverse market movements are associated with stronger and more persistent increases in stock market volatility. To account for this asymmetry, Corsi and Renò (2012) extend the standard HAR framework by including additional leverage terms, as shown in Equation (13). These terms augment the model with aggregated negative return components, allowing the LevHAR model to assign greater weight to downside risk.

$$RV_{t+1} = \beta_0 + \beta_d RV_t + \beta_w \overline{RV}_t^{(w)} + \beta_m \overline{RV}_t^{(m)} + \underbrace{\gamma_d RD_t^- + \gamma_w RW_t^- + \gamma_m RM_t^-}_{\text{Leverage effect}} + \varepsilon_{t+1} \tag{13}$$

In the equation above, $RD^- = \min\{0, r_t\}$ represents the negative component of daily

return, while $RW^-$ and $RM^-$ are defined according to Equation (14). Formally, these variables are constructed from the negative components as the minimum of zero and the trailing average return over the past 5 and 22 trailing trading days respectively.

$$RW^- = \min\{0, \frac{1}{5}\sum_{j=0}^{4} r_{t-j}\} \qquad RM^- = \min\{0, \frac{1}{22}\sum_{j=0}^{21} r_{t-j}\} \tag{14}$$

### 4.2.4 SHAR Model

Expanding on the asymmetric impact of positive and negative returns on stock market volatility, the semivariance HAR (SHAR) model, proposed by Patton and Sheppard (2015), decomposes realized variance into its positive and negative semivariance components, allowing volatility to respond differently to upside and downside variation, as shown in Equation (15).

$$RV_{t+1} = \beta_0 + \beta_d^+ RV_t^+ + \beta_d^- RV_t^- + \beta_w \overline{RV}_t^{(w)} + \beta_m \overline{RV}_t^{(m)} + \varepsilon_{t+1} \tag{15}$$

Specifically, realized variance can be expressed as the sum of these components, $RV_t \equiv RV_t^+ + RV_t^-$, where $RV_t^+$ and $RV_t^-$ denote the positive and negative semivariances, constructed as the sum of intraday squared returns conditional on returns being positive and negative, respectively.

### 4.2.5 HARQ Model

The HARQ model, introduced by Bollerslev et al. (2016), extends the standard HAR framework by explicitly accounting for the time-varying measurement error inherent in realized variance. As a nonparametric estimate of latent volatility, realized variance is subject to estimation noise whose variance is proportional to integrated quarticity. Ignoring this feature may lead to an errors-in-variables problem, which in turn induces attenuation bias in the estimated coefficients. The HARQ extension account for this by augmenting the standard HAR model with an interaction term between daily realized variance and the square root of realized quarticity, as evident in Equation (16).

$$RV_{t+1} = \beta_0 + \beta_d RV_t + \beta_{dQ}\left(RV_t\sqrt{RQ_t}\right) + \beta_w \overline{RV}_t^{(w)} + \beta_m \overline{RV}_t^{(m)} + \varepsilon_{t+1} \tag{16}$$

In the equation above, $RQ_t$ represents realized quarticity at time $t$, which serves as a proxy for the variability of the measurement error in $RV_t$. The interaction term $RV_t\sqrt{RQ_t}$ captures how said measurement error affects the predictive power of realized variance. Following Bollerslev et al. (2016), the quarticity adjustment is only applied to the daily lag of realized variance. This is because the measurement error is more pronounced at higher frequencies, whereas temporal aggregation over weekly and monthly horizons reduces much of the noise.

### 4.2.6 HAR-X Model

We complete the set of linear benchmark models with the HAR-X specification, following Christensen et al. (2023). The HAR-X model extends the standard HAR framework by augmenting the realized volatility components with additional macroeconomic and financial predictors. Formally, the HAR-X model is given by Equation (17), where $X_t$ denotes the vector of additional predictors used in the $\mathcal{M}_{\text{ALL}}$ feature set and $\gamma$ is the corresponding coefficient vector. As such, the HAR-X specification collapses to the standard HAR model in the $\mathcal{M}_{\text{HAR}}$ feature set when no exogenous predictors are included.

$$RV_{t+1} = \beta_0 + \beta_d RV_t + \beta_w \overline{RV}_t^{(w)} + \beta_m \overline{RV}_t^{(m)} + \gamma' X_t + \varepsilon_{t+1} \tag{17}$$

## 4.3 Regularized Linear Models

Regularized models extend the standard OLS framework by incorporating a penalty term that controls model complexity and mitigates overfitting. The general optimization problem that regularized models aim to solve is expressed in Equation (18).

$$\min_{\beta_0,\beta} \left\{ \sum_{t=1}^{T} (y_t - \beta_0 - X_t\beta)^2 + \lambda \cdot \phi(\beta) \right\} \tag{18}$$

The first component of this expression, that is the sum of squared residuals, $\sum_{t=1}^{T}(y_t - \beta_0 - X_t\beta)^2$, corresponds to the standard OLS loss function and measures how well the model fits the observed data. In the context of this study, $y_t$ denotes actual realized variance at time $t$, while $X_t$ represents the vector of predictors. Minimizing this first term alone yields the OLS estimator, which is optimal in-sample under the traditional assumptions but can suffer from overfitting in settings characterized by a large number of correlated predictors, which is often the case in forecasting applications.

To address these issues, regularized models expand the OLS specification with a penalty term, $\lambda \cdot \phi(\beta)$, where $\phi(\beta)$ is a function of the regression coefficients and $\lambda \geq 0$ is a tuning parameter that controls the strength of regularization. As $\lambda$ increases, large coefficients are penalized more heavily. The inclusion of this penalty term introduces bias into the parameter estimates while also reducing the variance of the estimators, which potentially leads to improved out-of-sample forecasting performance.

Different specifications of the penalty term $\phi(\beta)$ give rise to different regularization methods. For instance, a quadratic penalty term corresponds to the Ridge Regression (RR), an absolute value penalty term leads to the LASSO (LA), and a combination of both yields the Elastic Net (EN). In the following subsections, we follow the framework of Christensen et al. (2023) and present these three regularized linear models used in the empirical analysis. However, before introducing the individual specifications, we summarize the methodological choices that are held constant across all regularized models to ensure that differences in out-of-sample performance are attributable only to the penalty

structure and the selected feature set.

First, all predictors are standardized to zero mean and unit variance using moments computed from the rolling training portion of each estimation window. These standardization parameters re-estimated at every forecast origin to avoid look-ahead bias. Standardizing the predictors further ensures that the $L_1$, $L_2$, and combined penalty terms operate on a comparable scale across coefficients.

Second, the out-of-sample forecasts are generated using a fixed-length rolling window of size $n_{\text{train}} + n_{\text{val}} = 4{,}390$ observations, as summarized in Table (1). At each forecast origin, the window advances by one trading day, hyperparameters are re-tuned using the validation portion of that window, and a one-day-ahead forecast of realized variance is produced. Importantly, after hyperparameter selection, the model is re-estimated using only the rolling training portion rather than the combined training and validation sample. This procedure follows from the methodology in Christensen et al. (2023) and ensures consistency across model specifications.

Third, hyperparameters are selected by minimizing MSE over logarithmically spaced grids of penalty values. For each candidate value or, in the case of the EN, each combination of values, the model is estimated on the rolling training portion and evaluated on the rolling validation portion. The set of hyperparameters that yields the lowest validation MSE is then used to produce the forecasts.

Finally, we adopt the same insanity filter as prescribed before across all models, implying that any negative forecast predictions are replaced with the minimum observed realized variance in the estimation window.

### 4.3.1 Ridge Regression

The Ridge Regression (RR), originally introduced by Hoerl and Kennard (1970), extends the standard OLS framework by imposing a $L_2$ penalty term on the coefficient vector. The quadratic penalty term, which shrinks the estimated coefficients toward zero without setting them exactly to zero, is given by Equation (19).

$$\phi(\beta) = \sum_{j=1}^{p} \beta_j^2 \tag{19}$$

By penalizing large coefficients more heavily, this model reduces estimator variance and mitigates overfitting. In volatility forecasting, this is particularly relevant since there are often a large number of correlated predictors, and standard OLS specifications may struggle to handle such complexity and generate unbiased forecasts.

The degree of regularization is governed by the ridge penalty parameter $\lambda$, which is selected using validation-based grid search. We consider a logarithmically spaced grid of candidate values defined as $\lambda_k \in \{10^{-5}, \ldots, 10^{5}\}$ with $K = 1000$. This construction ensures broad coverage of both near-OLS and heavily regularized models.

### 4.3.2 LASSO

The LASSO (LA) regression, introduced by Tibshirani (1996), is another form of regularized linear regression that imposes an $L_1$ penalty term on the coefficient vector. In contrast to the ridge regression, which applies a quadratic penalty, the LASSO penalty is defined as the sum of the absolute value of the coefficients:

$$\phi(\beta) = \sum_{j=1}^{p} |\beta_j|. \tag{20}$$

This alternative structure for the penalty term implies that the loss function of the LA can effectively shrink coefficients exactly to zero, thereby performing variable selection. Similarly to RR, the LA penalty parameter $\lambda$ is selected using a validation-based grid search. In this case, we consider a logarithmically spaced grid of candidate values given by $\lambda_k \in \{10^{-10}, \ldots, 10^{1}\}$ with $K = 1000$. This allows the model to approximate the unregularized OLS solution for sufficiently small values of $\lambda$, while also spanning levels of penalization that may induce sparsity in the estimated coefficient vector when the feature set includes irrelevant or weakly informative variables.

### 4.3.3 Elastic Net

The Elastic Net (EN), introduced by Zou and Hastie (2005), combines the penalty structures of LA and RR, with the goal of addressing the limitations associated with each of these two individual models. While RR retains all predictors but fails to perform variable selection, LA can induce sparsity but may become unstable when the predictors are highly correlated. EN mitigates these issues by combining both the $L_1$ and $L_2$ penalty terms into a single penalty function:

$$\phi(\beta) = \alpha \sum_{j=1}^{p} |\beta_j| + (1 - \alpha) \sum_{j=1}^{p} \beta_j^2. \tag{21}$$

In Equation (21), $\alpha \in [0, 1]$ denotes the mixing parameter governing the contribution of the two penalty terms. Setting $\alpha = 1$ yields the LA specification, while $\alpha = 0$ corresponds to the RR model, implying that the EN nests both models as special cases. The EN specification therefore introduces two hyperparameters. The first is the overall penalty parameter $\lambda$, which controls the strength of the regularization, while the second one is the mixing parameter $\alpha$, which determines the relative contribution of the $L_1$ and $L_2$ penalty terms. Both hyperparameters are selected using validation-based grid search. Specifically, we consider a logarithmically spaced grid for the penalty parameter defined as $\lambda_k \in \{10^{-5}, \ldots, 10^{2}\}$ with $K = 1000$, alongside a discrete grid for the mixing parameter $\alpha \in \{0.01, 0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.70, 0.90, 1.00\}$. This scope allows the EN to span a wide range of potential structures, ranging from ridge-dominated shrinkage to LA-like variable selection.

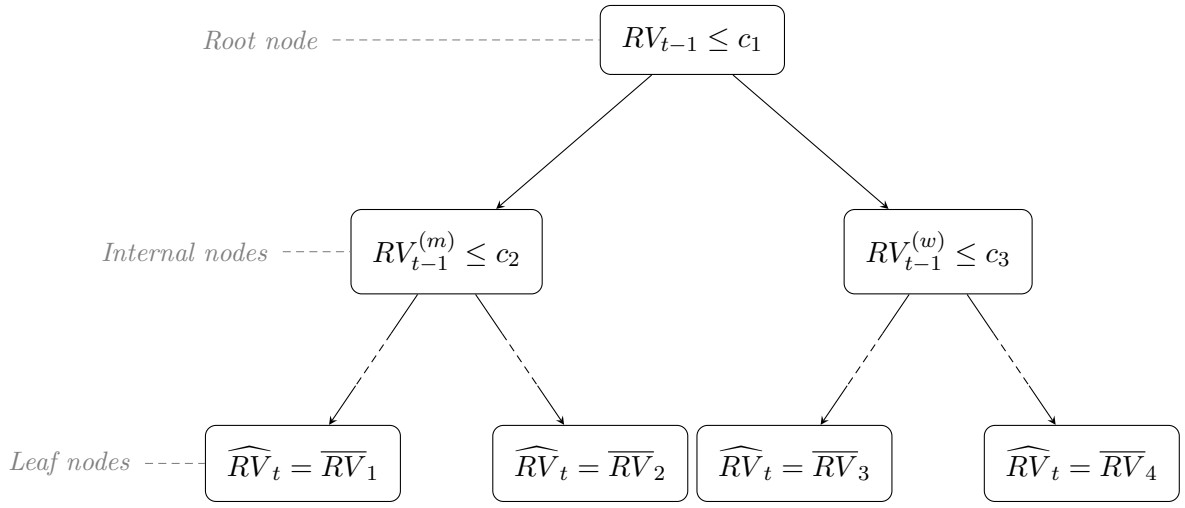
## 4.4 Tree-based Models

Tree-based models constitute a class of ML models particularly well-suited for volatility forecasting, given that they are able to capture many salient characteristics observed in financial data, including nonlinearities, interaction effects, and high dimensionality. As such, this set of models addresses many of the limitations associated with conventional linear regression frameworks.

A single decision tree is a nonparametric ML algorithm that generates predictions by recursively partitioning the predictor space into smaller subsets based on feature values and splitting criteria, thereby forming a hierarchical tree-like structure of decision rules. Predictions are made by averaging the values contained in the terminal nodes of the tree, which are commonly referred to as leaf nodes. Figure (4) provides an illustrative example of a single decision tree. Following the methodology presented in Christensen et al. (2023), we employ three tree-based models that are commonly used in the volatility forecasting literature, Bagging (BG), Random Forest (RF), and Gradient Boosting (GB).

Finally, to ensure economically meaningful predictions, any negative realized variance forecasts generated by all tree-based models are replaced with the minimum realized variance observed within the corresponding estimation window. Moreover, all variables are standardized similarly to the regularized linear models prior to estimation.

**Figure 4:** Illustration of a Regression Tree for Realized Variance




*Notes:* The figure illustrates a stylized regression tree using $\mathcal{M}_{\text{HAR}}$ predictors. Each internal node partitions the predictor space based on threshold values $c_1, c_2, c_3$, and each leaf node reports the average realized variance $\overline{RV}_j$ within that region where $j$ is the total number of leaf nodes in the tree. The dashed segments along the second-level edges indicate that additional splits may occur before reaching the leaf nodes.

### 4.4.1 Bagging

Single decision trees have a relatively rigid structure and are therefore prone to overfitting, instability, and high estimation variance. To mitigate these shortcomings, Breiman (1996) introduced a bootstrapped aggregation technique, commonly referred to as Bagging (BG), whereby multiple independent trees are estimated on resampled subsets of the original dataset and subsequently averaged to produce a final prediction. All subsets of the data are equally large and randomly formed, including replacement. The BG forecast is computed as the average prediction across $B$ regression trees, see Equation (22).

$$\hat{f}(Z_t) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{(b)}(Z_t) \tag{22}$$

In the equation above, $\hat{f}(Z_t)$ denotes the BG forecast, while $\hat{f}^{(b)}(Z_t)$ represents the prediction generated by the $b$th regression tree estimated on the corresponding bootstrap sample. Moreover, $Z_t$ denotes the feature vector and $B$ is the total number of trees. Note that the BG model builds on one primary source of randomness. Each tree $b$ is estimated using a bootstrap sample drawn with replacement from the original dataset, where $\mathcal{D} = \{(x_t, y_t)\}_{t=1}^{T}$ denotes the estimation sample. For each tree $b$, a bootstrap sample $\mathcal{D}^{(b)}$ is constructed, thereby introducing variability across trees and reducing the tendency of individual decision trees to overfit the data.

Following Christensen et al. (2023), the BG model is implemented in an off-the-shelf manner with minimal hyperparameter tuning. The base learner consists of a regression tree with a fixed minimum leaf size of five observations (see Hastie (2009)), and the ensemble is constructed using $B = 500$ trees. Since the BG model does not require hyperparameter tuning, the validation sample is merged with the training sample to form the estimation window.

### 4.4.2 Random Forest

Random forest (RF), introduced by Breiman (2001), constitutes another tree-based ML method designed to improve out-of-sample predictive performance through the aggregation of a large number of regression trees, and builds directly upon bootstrap aggregation (BG). However, while BG reduces estimation variance by averaging independently estimated trees, RF extends this framework by introducing an additional layer of randomness through feature sub-sampling, with the objective to further decorrelate individual trees. Accordingly, the specification for the RF model is defined in Equation (23).

$$\hat{f}(Z_t) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{(b)}(Z_t, \theta^{(b)}) \tag{23}$$

In this equation, $B$ denotes the total number of trees used in the forest and $\hat{f}^{(b)}(\cdot)$ represents the prediction generated by the $b$th regression tree. The term $\theta^{(b)}$ captures

the random subset of predictors drawn at each split during the construction of the tree $b$, and it is the only characteristic that differentiates RF from standard BG. Each regression tree partitions the predictor space into a set of disjoint regions, referred to as leaf nodes, and assigns a constant prediction within each region. Formally, a regression tree can be written as in Equation (24)

$$\hat{f}^{(b)}(Z_t) = \sum_{m=1}^{M_b} c_m^{(b)} \mathbf{1}\{Z_t \in R_m^{(b)}\} \tag{24}$$

where $\left\{R_m^{(b)}\right\}_{m=1}^{M_b}$ denotes the partition of the prediction space generated by tree $b$, and $c_m^{(b)}$ represents the average value of the target variable within region $R_m^{(b)}$. Similarly to the BG implementation, the minimum leaf size is set to five observations, while the ensemble consists of $B = 500$ trees.

The RF specification relies on two primary sources of randomness. First, similarly to the BG model, each tree is estimated using a bootstrap sample drawn with replacement. Second, and more importantly, RF differs from standard BG through random feature sub-sampling at each split in the tree. Rather than considering all $p$ predictors when determining the optimal split, only a randomly selected subset of size $m$ predictors is evaluated. Following the conventional recommendation for regression forests, we define $m$ according to Equation (25).

$$m = \left\lfloor \frac{p}{3} \right\rfloor \tag{25}$$

This specification is well established in the RF literature (see Christensen et al. (2023) and Hastie (2009)), and serves to reduce the correlation between trees. Intuitively, if all predictors were considered at each split, dominant explanatory variables, such as lagged realized variance, would be repeatedly selected across trees, resulting in highly similar and correlated tree structures. By restricting the candidate set of predictors for each tree $b$ according to Equation (25), the RF algorithm encourages individual trees to explore alternative splits of the predictor space, thereby increasing model diversity and improving the effectiveness of averaging. In volatility forecasting, this characteristic is particularly useful, as volatility dynamics commonly exhibit nonlinearities and complex interactions across time horizons and predictors. RF can flexibly capture such patterns without interaction effects or functional forms being specified prior to estimation.

Finally, given that RF, similarly to the BG model, is implemented in an off-the-shelf manner without extensive hyperparameter tuning, the training and validation windows are combined to form the estimation window.

### 4.4.3 Gradient Boosting

Gradient Boosting (GB), originally proposed by Friedman (2001), is a tree-based ML algorithm that constructs an ensemble of regression trees sequentially. Rather than fitting

a single complex model, GB estimates a large number of relatively shallow trees, where each tree is trained to correct the prediction errors of the preceding ensemble. This iterative refinement procedure is grounded in the principle of functional gradient descent, where each new tree is fitted to the negative gradient of the loss function with respect to the current prediction. Formally, the GB algorithm is initialized with a constant prediction following Equation (26):

$$\hat{f}^{(0)}(Z_t) = \arg\min_{\beta_0} \sum_{t \in \text{train}} (RV_t - \beta_0)^2 \tag{26}$$

which, under the MSE loss function, corresponds to the sample mean of realized variance in the training sample. At each subsequent iteration $b = 1, \ldots, B$, a shallow regression tree is fitted to the residuals of the current ensemble, and the prediction is updated according to Equation (27):

$$\hat{f}^{(b)}(Z_t) = \hat{f}^{(b-1)}(Z_t) + \nu \sum_{j=1}^{J_b} \hat{\gamma}_{jb} \, \mathbf{1}_{\{Z_t \in R_{jb}\}}, \tag{27}$$

where $R_{jb}$ denotes the $j$th terminal node of the $b$th tree, $\hat{\gamma}_{jb}$ is the optimal step size associated with that node, and $\nu \in (0, 1]$ is a learning rate that controls the contribution of each tree to the final prediction. The final forecast is provided by the accumulated ensemble, such that $\hat{f}(Z_t) \equiv \hat{f}^{(B)}(Z_t)$.

Each individual tree in the ensemble constitutes a so-called weak learner, that is a model only marginally better than a naive benchmark. However, the sequential aggregation of these weak learners produces a highly flexible nonparametric forecasting framework capable of capturing nonlinearities and interaction effects in the predictor space without requiring these to be specified.

A well-known limitation of GB in financial applications is its sensitivity to outliers. Since the algorithm places disproportionate weight on large residuals when fitting each successive tree, extreme observation, such as volatility spikes during market stress episodes, can distort the learned function. This contrasts with RF, where independent bootstrap sampling and averaging mitigate this issue. Given the pronounced right-skewness and heavy tails typically observed in realized variance, this issue is particularly relevant in volatility forecasting applications.

Following Christensen et al. (2023), the GB model is estimated using a rolling window framework, in which the hyperparameters are re-selected at each forecast origin based on the validation portion of the current window. Specifically, the tree depth is tuned over $\{1, 2\}$, the learning rate over $\{0.01, 0.1\}$, and the number of trees over $\{50, 100, \ldots, 500\}$, while all remaining parameters are set to their default values. The combination of hyperparameters that minimizes the MSE on the validation set is selected, and the model is re-estimated on the rolling training portion prior to generating the one-day-ahead forecast.

## 4.5 Neural Networks

To complement the previously presented set of models, we implement a family of feed-forward neural networks (NNs) to forecast next-day realized variance. In contrast to conventional linear regression models, NNs are capable of approximating highly nonlinear and complex functional relationships between predictors and the target variable, characteristics that make NNs particularly suitable for volatility forecasting. Following Christensen et al. (2023), we estimate four different NN architectures, denoted $\text{NN}_1$, $\text{NN}_2$, $\text{NN}_3$, and $\text{NN}_4$. Figure (5) illustrates the architecture of the respective NNs. Across these specifications, the primary intended methodological difference is the depth and width of the hidden layer structure. This implies that all remaining components of the modeling framework, including data pre-processing, optimization, regularization, and forecasting methodology are held fixed across the four architectures.

### 4.5.1 Feed-forward Neural Network Architectures

A feed-forward neural network maps the predictor vector available at time $t$, denoted $Z_t$, into a forecast of next-day realized variance through a sequence of linear transformations and nonlinear activation functions. Information flows sequentially through the network from the input layer to one or several hidden layers and then to the final output layer, which produces a prediction $\hat{f}(Z_t)$. We let $L$ denote the number of hidden layers in a given NN architecture and let $N_l$ represent the total number of neurons in layer $l$. The general equation for the $l$th hidden layer is expressed in Equation (28).

$$a_t^{(\theta_{l+1},b_{l+1})} = g_l\left(\sum_{j=1}^{N_l} \theta_j^{(l)} a_t^{(\theta_l,b_l)} + b^{(l)}\right), \quad 1 \leq l \leq L \tag{28}$$

In this equation, $g_l(\cdot)$ denotes the activation function for layer $l$, $\theta^{(l)}$ is the corresponding weight matrix, and $b^{(l)}$ represents the bias vector serving as an activation threshold. Moreover, $a_t^{(\theta_l,b_l)}$ denotes the input for layer $l+1$, while $a_t^{(\theta_{l+1},b_{l+1})}$ denotes the resulting output for that layer. The output layer uses a linear activation function, which is appropriate for a regression problem where the target variable is continuous. The final forecast is constructed by composing the transformation across all layers, as illustrated in Equation (29), where each layer adds a new level of transformation to the NN.

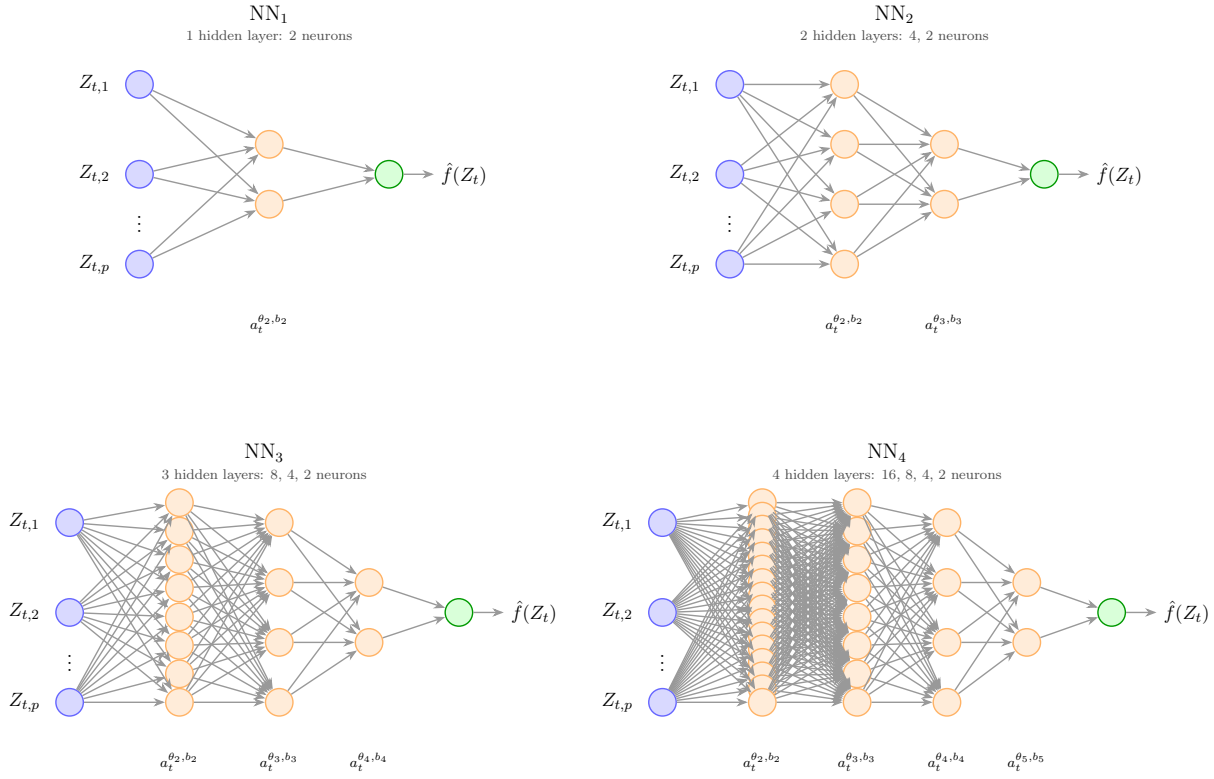
$$\hat{f}(Z_t) = (g_L \circ \cdots \circ g_1)(Z_t) \tag{29}$$

In accordance with Christensen et al. (2023), we adopt the Leaky Rectified Linear Unit (L-ReLU) activation function proposed by Maas et al. (2013) for all hidden layers. The L-ReLU activation function is defined according to Equation (30).

$$\text{L-ReLU}(x) = \begin{cases} cx & \text{if } x < 0 \\ x & \text{otherwise} \end{cases} \tag{30}$$

Compared with the standard ReLU activation function, L-ReLU allows for a small non-zero slope, denoted $c$, when the input is negative. This mitigates the so-called 'dying ReLU' problem, which occurs when neurons in the network become stuck generating zero for all inputs and thus essentially die. Following Christensen et al. (2023), the slope parameter in Equation (30) is set to $c = 0.01$.

**Figure 5:** Feed-Forward Neural Network Architectures: $\text{NN}_1$, $\text{NN}_2$, $\text{NN}_3$, and $\text{NN}_4$




*Notes:* The figure illustrates the four feed-forward neural network architectures considered in this study, following Christensen et al. (2023) and inspired by the geometric pyramid structure of Masters (1993) and Gu et al. (2020). Each network receives $p$ input features $Z_{t,1}, \ldots, Z_{t,p}$, where $p = 3$ for the $\mathcal{M}_{\text{HAR}}$ feature set and $p = 16$ for the extended $\mathcal{M}_{\text{ALL}}$ feature set. $\text{NN}_1$ is a shallow network with a single hidden layer of 2 neurons; $\text{NN}_2$ has two hidden layers of 4 and 2 neurons; $\text{NN}_3$ has three hidden layers of 8, 4, and 2 neurons; $\text{NN}_4$ has four hidden layers of 16, 8, 4, and 2 neurons. All hidden layers apply a Leaky ReLU activation function with negative slope $c = 0.01$ and are subject to dropout regularization with a keep rate of 0.8. The output layer returns the one-step-ahead forecast $\hat{f}(Z_t)$ via a linear activation function.

### 4.5.2 Estimation and Regularization

As previously noted, all components of the neural network framework, except for the NN architectures themselves, are held fixed across specifications. Consequently, any differences in out-of-sample performance can be attributed to differences in network depth and width as opposed to variations in the estimation procedure.

First, the NNs are estimated using the chronological training, validation and test split, previously described in Table (1). In contrast to models that are re-estimated recursively over time, each NN is fitted once on the initial training sample and subsequently evaluated on the fixed validation and test periods. Importantly, the network is not re-estimated on the combined training and validation samples prior to forecasting. This fixed-window design follows the empirical implementation of Christensen et al. (2023) and reflects the comparatively high computational cost associated with NN estimation.

Second, both the predictors and the target variable are standardized using moments computed from the initial training sample only. The same transformation is then applied to the validation and test sets. Standardizing the predictors improves numerical stability during optimization, while standardizing the target variable facilitates estimation procedure by ensuring that the loss function is evaluated on a comparable scale across observations. Forecasts are subsequently transformed back to the original realized variance scale before proceeding with performance evaluation.

Third, the network parameters are estimated by minimizing the MSE loss function in the standardized target space. Optimization is performed using the Adam optimizer with a learning rate of 0.001, following Christensen et al. (2023). Training is conducted using mini-batches of size 64, and the training observations are reshuffled at the beginning of each epoch. A maximum of 500 epochs is allowed and the initial weights are drawn using Xavier (Glorot) normal initialization, while all bias terms are initialized to zero.

Fourth, regularization is implemented through a combination of dropout and early stopping. Dropout is applied after each hidden layer activation during training and is used to reduce overfitting by preventing the network from becoming overly dependent on a small subset of neurons. It works by randomly deactivating neurons in a given layer $l$ during training, according to Equation (31).

$$\tilde{a}_t^{(\theta_l, b_l)} = d^{(\theta_l, b_l)} \odot a_t^{(\theta_l, b_l)} \quad \text{where} \quad d_i^{(\theta_l, b_l)} \sim \text{Bernoulli}(p) \tag{31}$$

In this equation, $d^{(\theta_l, b_l)}$ is a masked vector performing the drop-out through element-wise multiplication with layer $a_t^{(\theta_l, b_l)}$. Following Christensen et al. (2023), the drop-out rate is set at $p = 0.8$, implying that we keep 80% of the neurons in each hidden layer $l$ during training. Dropout is applied only during training and is turned off when generating validation and test forecasts. To further control for overfitting, early stopping is implemented using the validation set MSE. After each training epoch, the model is evaluated on the full validation set, and training is terminated if the validation loss fails to improve for 100 consecutive epochs. The parameter values from the epoch with the lowest validation MSE are then retained.

Fifth, since NN performance may depend on the random initialization of the network weights, each architecture is trained 100 times using different random seeds. The resulting models are ranked according to their validation set MSE. We follow the approach of Christensen et al. (2023) and retain two forecast variants for each architecture: the forecast

generated by the single best performing network and the forecast obtained by averaging the predictions from the ten best performing networks. These two variants are denoted with superscripts 1 and 10, respectively, in Section (5). This procedure reduces sensitivity to any particular random initialization while maintaining a comparable estimation design across the four architectures.

Finally, since realized variance is strictly non-negative, we again apply an insanity filer to the NN forecasts, meaning that any forecast falling below the minimum realized variance observed in the pre-test sample is replaced by this minimum value.

## 4.6 Out-of-Sample Performance Evaluation

To evaluate the predictive performance of the forecasting models estimated on the two feature sets $\mathcal{M}_{\text{HAR}}$ and $\mathcal{M}_{\text{ALL}}$, we employ a set of out-of-sample evaluation measures commonly used in the volatility forecasting literature. Following the methodological framework of Christensen et al. (2023), our primary loss function is the Mean Squared Error (MSE), which is complemented by the Quasi-Likelihood (QLIKE) loss function. To assess whether differences in forecasting performance are statistically significant, we further conduct pairwise tests of equal predictive accuracy using the framework introduced by Diebold and Mariano (1995). Finally, we supplement the MSE with a set of additional loss functions to examine the robustness of our findings.

### 4.6.1 Relative Mean Squared Error

Let $\hat{f}(Z_t)$ denote the one-day-ahead forecast of realized variance ($RV_{t+1}$), generated at time $t$ using the feature set $Z_t$. Depending on the specification, the feature set corresponds to either the $\mathcal{M}_{\text{HAR}}$ or the $\mathcal{M}_{\text{ALL}}$ feature set. The MSE loss function for a given specification is then defined by Equation (32).

$$\mathcal{L}_{\text{MSE}}(\hat{f}(Z_t)) = \frac{1}{T} \sum_{t \in \text{OOS}} \left( RV_{t+1} - \hat{f}(Z_t) \right)^2 \tag{32}$$

In this equation, $T$ refers to the number of observations (trading days) in the out-of-sample test period, and $RV_{t+1}$ denotes the actual realized variance observed on trading day $t+1$. Since the MSE performance measure serves as the primary loss function underlying the estimation of all models in our study, it provides a natural and consistent basis for evaluating out-of-sample forecasting performance.

To facilitate model comparison, we additionally report a pairwise *relative MSE* score for each model pair $(i, j)$, defined according to Equation (33)

$$\text{Relative MSE}_{i,j} = \frac{\text{MSE}_i}{\text{MSE}_j} \tag{33}$$

where model $j$ serves as the benchmark specification. A relative MSE ratio below one indicates that model $i$ achieves a lower forecast error and therefore outperforms the bench-

mark model $j$, whereas a ratio above one has the opposite implication. The complete set of pairwise relative MSE scores for all models is reported in Section (5).

### 4.6.2 Relative Quasi-Likelihood

In addition to the MSE loss function, we extend the out-of-sample performance evaluation of Christensen et al. (2023) by computing the QLIKE loss function. As discussed by Patton and Sheppard (2009) and Patton (2011), QLIKE is particularly well-suited for evaluating competing volatility forecasting models as it remains robust to measurement errors in the realized variance estimator, which itself is an inherently noisy proxy for the latent conditional variance process. In contrast, alternative loss functions may generate distorted model rankings when the volatility proxy is noisy. Compared with the MSE, which penalizes squared forecast deviations in levels, the QLIKE loss function evaluates forecasts based on the ratio between realized and predicted variance, thereby providing a likelihood-based assessment of model accuracy. As a result, forecast errors tend to be distributed more evenly across the evaluation sample, whereas the MSE may become disproportionately influenced by a small number of extremely volatile trading days. Formally, the QLIKE loss function is defined according to Equation (34).

$$\mathcal{L}_{\text{QLIKE}}(\hat{f}(Z_t)) = \frac{1}{T} \sum_{t \in \text{OOS}} \left[ \frac{RV_{t+1}}{\hat{f}(Z_t)} - \log \left( \frac{RV_{t+1}}{\hat{f}(Z_t)} \right) - 1 \right] \tag{34}$$

where $T$ denotes the number of observations in the out-of-sample period, $RV_{t+1}$ represents the realized variance observed on trading day $t+1$, and $\hat{f}(Z_t)$ is the one-day-ahead forecast generated by the model using the predictor vector $Z_t$.

An important feature of the QLIKE loss function is its asymmetric penalty structure, whereby an underestimation of volatility is penalized more heavily than an overestimation of the same magnitude. This characteristic is particularly relevant in financial applications, where underestimating volatility can lead to severe consequences from a risk management perspective. By assigning greater weight to such adverse forecast errors, QLIKE provides an economically meaningful assessment of predictive performance. Note that the QLIKE score goes towards infinity as forecasts approach zero, which makes the loss function sensitive to the minimum value imposed by the non-negativity constraint for models that frequently produce negative forecasts. Taken together, these properties make QLIKE both a theoretically and practically relevant complement to the MSE loss function in the model performance evaluation. We compute pairwise *relative QLIKE* score analogously to the relative MSE metric to facilitate model comparison, defined in Equation (35).

$$\text{Relative QLIKE}_{i,j} = \frac{\text{QLIKE}_i}{\text{QLIKE}_j} \tag{35}$$

As before, model $j$ serves as the benchmark specification. A relative QLIKE ratio below one indicates that model $i$ outperforms the benchmark model, while a ratio above

one implies the opposite. The complete set of pairwise relative QLIKE scores for the $\mathcal{M}_{\text{HAR}}$ and $\mathcal{M}_{\text{ALL}}$ feature sets is reported in Section (5).

### 4.6.3 Additional Loss Functions

While the MSE and QLIKE loss functions constitute the primary evaluation measures in the empirical analysis, alternative loss functions may provide complementary insights to evaluate model performance. Accordingly, in addition to the relative MSE and QLIKE measures, we also report the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) for each model across both feature sets, presented in Equation (36) and (37) respectively. These complementary results are not part of our main analysis but are presented in Appendix (8.8), specifically in Table (11) and (12).

$$\mathcal{L}_{MAE}(\hat{f}(Z_t)) = \frac{1}{T} \sum_{t \in \text{OOS}} \left| RV_{t+1} - \hat{f}(Z_t) \right| \tag{36}$$

$$\mathcal{L}_{RMSE}(\hat{f}(Z_t)) = \sqrt{\frac{1}{T} \sum_{t \in \text{OOS}} \left( RV_{t+1} - \hat{f}(Z_t) \right)^2} \tag{37}$$

### 4.6.4 Diebold-Mariano Test

While the relative MSE and QLIKE measure quantifies the magnitude of differences in forecasting accuracy across models, it does not provide a formal assessment of whether such differences are statistically significant. We therefore complement the relative MSE and QLIKE measures with the Diebold and Mariano (1995) test for equal predictive accuracy. For a given model pair $(i, j)$, the loss differential is defined in Equation (38) for MSE and Equation (39) for QLIKE.

$$d_{ij,t}^{MSE} = \left( RV_{t+1} - \hat{f}_i(Z_t) \right)^2 - \left( RV_{t+1} - \hat{f}_j(Z_t) \right)^2 \tag{38}$$

$$d_{ij,t}^{QLIKE} = \left[ \frac{RV_{t+1}}{\hat{f}_i(Z_t)} - \log\left( \frac{RV_{t+1}}{\hat{f}_i(Z_t)} \right) - 1 \right] - \left[ \frac{RV_{t+1}}{\hat{f}_j(Z_t)} - \log\left( \frac{RV_{t+1}}{\hat{f}_j(Z_t)} \right) - 1 \right] \tag{39}$$

In these specifications, a negative value of $d_{ij,t}$ implies that model $i$ produces a smaller squared error than model $j$ at time $t$. The null hypothesis of equal predictive accuracy, $H_0 : \mathbb{E}[d_{ij,t}] = 0$, is tested against the one-sided alternative $H_1 : \mathbb{E}[d_{ij,t}] > 0$, implying superior predictive accuracy of model $j$. The Diebold-Mariano (DM) test statistic is computed according to Equation (40).

$$DM = \frac{\bar{d}}{\sqrt{\widehat{\text{Var}}(\bar{d})}} \tag{40}$$

In this equation, $\bar{d}$ denotes the sample mean of the loss differential series, and $\widehat{\text{Var}}(\bar{d})$

is a heteroskedasticity and autocorrelation consistent (HAC) estimator that accounts for potential serial correlation in $d_{ij,t}$. Under the null hypothesis of equal predictive accuracy, the DM statistic asymptotically follows a standard normal distribution $DM \xrightarrow{d} \mathcal{N}(0,1)$.

# 5 Empirical Results

In the following subsections, we present the empirical findings of our study. We begin by evaluating the forecasting performance of each model using only the predictors included in the $\mathcal{M}_{\text{HAR}}$ feature set. We then proceed to compare these results with the forecasting performance obtained when the models are estimated using the more extensive $\mathcal{M}_{\text{ALL}}$ feature set, which incorporates additional macroeconomic and financial predictors.

## 5.1 Results from $\mathcal{M}_{\text{HAR}}$ Feature Set

This subsection presents the out-of-sample forecasting results for all models estimated using the $\mathcal{M}_{\text{HAR}}$ feature set. The evaluation period consists of 1,099 trading days spanning the period from 2016-04-21 to 2020-08-06. Table (3) reports the pairwise relative MSE results while Table (4) presents the corresponding pairwise comparisons under the QLIKE loss function. In both tables, entries below one indicate that the column model produces a lower expected loss than the row model. Statistical significance is assessed using the one-sided Diebold and Mariano (1995) test for equal predictive accuracy.

### 5.1.1 Pairwise Relative MSE

Observing the results in Table (3) yields five compelling insights. First, HARQ emerges as the single best performing model on the $\mathcal{M}_{\text{HAR}}$ feature set under the MSE loss function. All competing models produce higher MSE values relative to HARQ. Compared with HAR, HARQ achieves a relative MSE of 0.846, with the corresponding Diebold and Mariano (1995) test indicating significance at the 10% level. Against the remaining HAR extensions, HARQ improves forecast accuracy by roughly 10% to 18%, and these differences are all statistically significant at the 10% level. Effectively, $\mathcal{M}_{\text{HAR}}$ leaves little room through which the remaining ML models can extract additional information in a statistically significant manner.

Second, LogHAR is the only other model, apart from HARQ, that outperforms HAR under $\mathcal{M}_{\text{HAR}}$. LogHAR improves upon HAR by approximately 6.4% (relative MSE of 0.936) and is statistically significant at the 10% level, whereas LevHAR delivers only modest forecast gains relative to the baseline HAR (0.951) and the difference is not statistically significant. SHAR performs marginally worse than HAR with a relative MSE of 1.031. These findings suggest that, within a restricted information set that only contains lagged realized variance components, the leverage and semivariance refinements of

**Table 3:** Pairwise Relative MSE and Diebold-Mariano Tests on $\mathcal{M}_{\text{HAR}}$ Feature Set

| | Benchmark Models | | | | | | Regularized Models | | | Tree-Based Models | | | Neural Networks | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **HAR** | **HAR-X** | **LogHAR** | **LevHAR** | **SHAR** | **HARQ** | **RR** | **LA** | **EN** | **BG** | **RF** | **GB** | $\mathbf{NN}_1^1$ | $\mathbf{NN}_1^{10}$ | $\mathbf{NN}_2^1$ | $\mathbf{NN}_2^{10}$ | $\mathbf{NN}_3^1$ | $\mathbf{NN}_3^{10}$ | $\mathbf{NN}_4^1$ | $\mathbf{NN}_4^{10}$ |
| **HAR** | – | 1.000 | **0.936** | 0.951 | 1.031 | **0.846** | 1.028 | 1.039 | 1.039 | 0.941 | 0.916 | 0.973 | 0.966 | 1.009 | 0.994 | 0.997 | 1.005 | 1.003 | 1.006 | 1.045 |
| **HAR-X** | 1.000 | – | **0.936** | 0.951 | 1.031 | **0.846** | 1.028 | 1.039 | 1.039 | 0.941 | 0.916 | 0.973 | 0.966 | 1.009 | 0.994 | 0.997 | 1.005 | 1.003 | 1.006 | 1.045 |
| **LogHAR** | 1.068 | 1.068 | – | 1.016 | 1.101 | **0.904** | 1.098 | 1.110 | 1.110 | 1.005 | 0.979 | 1.040 | 1.032 | 1.078 | 1.062 | 1.065 | 1.074 | 1.072 | 1.075 | 1.117 |
| **LevHAR** | 1.052 | 1.052 | 0.984 | – | 1.084 | **0.890** | 1.081 | 1.093 | 1.093 | 0.989 | 0.964 | 1.024 | 1.016 | 1.061 | 1.046 | 1.048 | 1.057 | 1.055 | 1.058 | 1.099 |
| **SHAR** | 0.970 | 0.970 | 0.908 | 0.922 | – | **0.821** | 0.997 | 1.008 | 1.008 | 0.913 | 0.889 | 0.944 | 0.937 | 0.979 | 0.965 | 0.967 | 0.975 | 0.973 | 0.976 | 1.014 |
| **HARQ** | 1.182 | 1.182 | 1.106 | 1.124 | 1.218 | – | 1.215 | 1.228 | 1.228 | 1.112 | 1.083 | 1.150 | 1.141 | 1.192 | 1.175 | 1.178 | 1.188 | 1.186 | 1.189 | 1.235 |
| **RR** | 0.973 | 0.973 | **[0.911]** | 0.925 | 1.003 | **(0.823)** | – | 1.011 | 1.011 | 0.915 | **0.892** | 0.947 | **(0.940)** | 0.982 | 0.967 | 0.970 | 0.978 | 0.976 | 0.979 | 1.017 |
| **LA** | **(0.962)** | **(0.962)** | **[0.901]** | 0.915 | 0.992 | **(0.814)** | **(0.989)** | – | 1.000 | 0.905 | **0.882** | **0.937** | **(0.929)** | 0.971 | 0.957 | 0.959 | 0.967 | 0.966 | 0.968 | 1.006 |
| **EN** | **(0.963)** | **(0.963)** | **[0.901]** | 0.915 | 0.992 | **(0.815)** | **(0.989)** | 1.000 | – | 0.905 | **0.882** | **0.937** | **(0.930)** | 0.971 | 0.957 | 0.960 | 0.968 | 0.966 | 0.969 | 1.006 |
| **BG** | 1.063 | 1.063 | 0.995 | 1.011 | 1.096 | **0.900** | 1.093 | 1.105 | 1.104 | – | 0.974 | 1.035 | 1.027 | 1.072 | 1.057 | 1.060 | 1.069 | 1.067 | 1.070 | 1.111 |
| **RF** | 1.091 | 1.091 | 1.022 | 1.038 | 1.125 | **0.924** | 1.122 | 1.134 | 1.134 | 1.027 | – | 1.062 | 1.054 | 1.101 | 1.085 | 1.088 | 1.097 | 1.095 | 1.098 | 1.141 |
| **GB** | 1.027 | 1.027 | **0.962** | 0.977 | 1.059 | **(0.869)** | 1.056 | 1.068 | 1.067 | 0.966 | **(0.941)** | – | 0.992 | 1.036 | 1.022 | 1.024 | 1.033 | 1.031 | 1.034 | 1.074 |
| $\mathbf{NN}_1^1$ | 1.035 | 1.035 | 0.969 | 0.985 | 1.067 | **0.876** | 1.064 | 1.076 | 1.076 | 0.974 | 0.949 | 1.008 | – | 1.044 | 1.029 | 1.032 | 1.041 | 1.039 | 1.042 | 1.082 |
| $\mathbf{NN}_1^{10}$ | 0.991 | 0.991 | 0.928 | 0.943 | 1.022 | **0.839** | 1.019 | 1.030 | 1.030 | 0.932 | 0.908 | 0.965 | **0.957** | – | 0.986 | 0.988 | 0.997 | 0.995 | 0.997 | 1.036 |
| $\mathbf{NN}_2^1$ | 1.006 | 1.006 | 0.941 | 0.956 | 1.037 | 0.851 | 1.034 | 1.045 | 1.045 | 0.946 | 0.922 | 0.979 | 0.971 | 1.015 | – | 1.003 | 1.011 | 1.009 | 1.012 | 1.051 |
| $\mathbf{NN}_2^{10}$ | 1.003 | 1.003 | 0.939 | 0.954 | 1.034 | **0.849** | 1.031 | 1.043 | 1.042 | 0.944 | 0.919 | 0.976 | 0.969 | 1.012 | 0.997 | – | 1.008 | 1.007 | 1.009 | 1.049 |
| $\mathbf{NN}_3^1$ | 0.995 | 0.995 | 0.931 | 0.946 | 1.025 | 0.842 | 1.022 | 1.034 | 1.033 | 0.936 | 0.911 | 0.968 | 0.961 | 1.003 | 0.989 | 0.992 | – | 0.998 | 1.001 | 1.040 |
| $\mathbf{NN}_3^{10}$ | 0.997 | 0.997 | 0.933 | 0.948 | 1.027 | **0.843** | 1.024 | 1.036 | 1.035 | 0.937 | 0.913 | 0.970 | 0.963 | 1.005 | **0.991** | 0.993 | 1.002 | – | 1.003 | 1.042 |
| $\mathbf{NN}_4^1$ | 0.994 | 0.994 | 0.930 | 0.945 | 1.024 | 0.841 | 1.021 | 1.033 | 1.032 | 0.935 | 0.911 | 0.967 | 0.960 | 1.003 | 0.988 | 0.991 | 0.999 | 0.997 | – | 1.039 |
| $\mathbf{NN}_4^{10}$ | 0.957 | 0.957 | 0.896 | 0.910 | 0.986 | 0.810 | 0.983 | 0.994 | 0.994 | 0.900 | 0.877 | 0.931 | 0.924 | 0.965 | **0.951** | 0.954 | 0.962 | 0.960 | **0.963** | – |

*Notes:* Each entry $(i, j)$ reports the out-of-sample MSE of row model $i$ relative to column model $j$, computed over the test period 2016-04-21 to 2020-08-06 (1,099 observations). A value below unity indicates that column model $i$ outperforms row model $j$. Significant values denote rejection of equal predictive accuracy in favor of the column model based on one-sided Diebold and Mariano (1995) tests with Newey-West standard errors (5 lags). The notation for significance level is represented as follows; **number** denotes $p < 0.10$, **(number)** denotes $p < 0.05$, and **[number]** denotes $p < 0.01$.

the LevHAR and SHAR models do not outperform HAR at the daily forecasting horizon. The strong performance of the baseline HAR is further supported by its strong autocorrelation, where all three realized variance components exhibit strong persistence across multiple lags (see Figure (10) in Appendix (8.2)). As expected, the HAR-X model collapses mechanically to HAR in this feature set because of the absence of additional exogenous regressors. Therefore, the row and column values for HAR and HAR-X are identical by construction.

Third, the regularized linear models fail to reject the null of equal predictive accuracy relative to the benchmark HAR model. In particular, both LA and EN generate significantly higher MSE values than the HAR benchmark (1.039 respectively), while RR (1.028) performs somewhat better relative to its regularized counterparts, although without delivering a statistically significant improvement over the benchmark. Table (3) further shows that HAR significantly outperforms both LA and EN at the 5% level, whereas LogHAR outperform all three regularized models at the 1% level. These findings suggest that disciplined linear shrinkage methods provide little benefit when the information set is restricted to only three highly persistent HAR-type predictors in our European single-index setting. Moreover, the virtually identical results obtained for LA and EN, matching to three-decimal points, indicate that the EN penalty function frequently collapses to the $L_1$ specification during cross-validation. As a result, EN is effectively reduced to the variable selection of LA in a substantial share of the estimation windows.

Fourth, the tree-based models generally perform better than the regularized linear specifications, with BG, RF, and GB all producing relative MSE entries below one relative to the HAR benchmark. In particular, RF achieves the strongest performance among the tree-based methods with a relative MSE of 0.916, followed by BG and GB with relative MSE values of 0.941 and 0.973, respectively. However, none of these improvements are statistically significant according to the Diebold and Mariano (1995) test. At the same time, the HAR benchmark does not significantly outperform any of the tree-based models either, suggesting that the forecasting performance of these specifications remains statistically indistinguishable on the $\mathcal{M}_{\text{HAR}}$ feature set.

Finally, the relative MSE entries of all NN architecture and ensemble specification against HAR cluster tightly around one, ranging from 0.966 ($\text{NN}_1^1$) to 1.045 ($\text{NN}_4^{10}$), with no statistically significant differences relative to HAR in either direction for any model architecture. This finding is consistent with the economic interpretation that, with a three predictor information set, the marginal value of capturing nonlinearities and complex patterns is limited, and without additional macroeconomic or financial predictors, there is little room for deeper NN architectures to learn from these interactions in the data. The pattern is also internally consistent across architectures. The top-10 ensemble produce relative MSE values close to, or marginally worse than, the best seed specifications, while additional depth of the architecture provides no consistent improvement in forecasting performance. We return to this discussion in Section (6).

### 5.1.2 Pairwise Relative QLIKE

Re-evaluating the same forecasts under the QLIKE loss function in Table (4) delivers a markedly sharper picture, both in terms of the magnitude of pairwise differences and in their statistical significance. Three observations stand out.

First, the overall ranking observed under MSE is largely preserved under the QLIKE loss function, although the performance differences between models become more pronounced. HARQ remains among the strongest performing models, achieving a relative QLIKE score of 0.792 compared to HAR, which corresponds to an improvement of approximately 20% that is now significant at the 1% level, compared to 10% level under the MSE loss function. The other benchmark HAR extension that materially improves on HAR is LogHAR, with a relative QLIKE value of 0.834, again significant at the 1% level. Notably, the performance of LevHAR reverses sign under QLIKE relative to its MSE result. While the model produced a relative MSE score compared to HAR of 0.951, and statistically indistinguishable from HAR, its QLIKE score relative to HAR increases to 1.733, with strong rejection in the opposite direction. This indicates that LevHAR produces forecasts whose proportional accuracy is materially worse than HAR even though its MSE performance is of a similar magnitude. This pattern emerges naturally when a model occasionally underestimates volatility during extremely volatile days because the QLIKE loss function asymmetrically penalizes such underestimations more aggressively than the MSE score, making these predictions substantially more costly under a likelihood-based evaluation.

Second, the QLIKE comparison broadens the set of models that significantly outperform HAR. In addition to HARQ and LogHAR, both RF and BG now significantly outperform HAR at the 1% level, with relative QLIKE scores of 0.781 and 0.797, respectively. By contrast, the remaining model families perform substantially worse relative to HAR under QLIKE and carry on no Diebold and Mariano (1995) test significance. RR, LA, and EN record relative QLIKE entries of 1.686, 1.765 and 1.761 relative to HAR, while GB produces a value of 1.403. Similarly, the NN family records relative QLIKE scores ranging from 1.189 ($\text{NN}_4^1$) to 1.640 ($\text{NN}_3^1$). None of these models beat HAR under QLIKE, and reading the corresponding cells in the opposite direction, we observe Diebold and Mariano (1995) rejections in favor of HAR at the 1% significance mark. Disciplined linear shrinkage, GB trees, and the NN family therefore not only fail to outperform HAR under QLIKE, but are in many cases significantly worse. The economic intuition is that these models produce forecasts whose proportional accuracy is systematically inferior to the HAR benchmark on the $\mathcal{M}_{\text{HAR}}$ feature set. The two model classes that improve upon HAR under both MSE and QLIKE are are therefore the HAR-type extensions, HARQ and LogHAR, along with the bootstrap-aggregated tree-based models BG and RF. What changes under QLIKE is primarily the strength of the rejection. HARQ and LogHAR move from significance at the 10% under MSE to the 1% level under QLIKE, while BG and RF transition from MSE insignificance to a 1% significance level under QLIKE.

**Table 4:** Pairwise Relative QLIKE and Diebold-Mariano Tests on $\mathcal{M}_{\text{HAR}}$ Feature Set

| | Benchmark Models | | | | | | Regularized Models | | | Tree-Based Models | | | Neural Networks | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **HAR** | **HAR-X** | **LogHAR** | **LevHAR** | **SHAR** | **HARQ** | **RR** | **LA** | **EN** | **BG** | **RF** | **GB** | $\mathbf{NN}_1^1$ | $\mathbf{NN}_1^{10}$ | $\mathbf{NN}_2^1$ | $\mathbf{NN}_2^{10}$ | $\mathbf{NN}_3^1$ | $\mathbf{NN}_3^{10}$ | $\mathbf{NN}_4^1$ | $\mathbf{NN}_4^{10}$ |
| **HAR** | – | 1.000 | **[0.834]** | 1.733 | 1.013 | **[0.792]** | 1.686 | 1.765 | 1.761 | **[0.797]** | **[0.781]** | 1.403 | 1.394 | 1.227 | 1.222 | 1.377 | 1.640 | 1.448 | 1.189 | 1.486 |
| **HAR-X** | 1.000 | – | **[0.834]** | 1.733 | 1.013 | **[0.792]** | 1.686 | 1.765 | 1.761 | **[0.797]** | **[0.781]** | 1.403 | 1.394 | 1.227 | 1.222 | 1.377 | 1.640 | 1.448 | 1.189 | 1.486 |
| **LogHAR** | 1.199 | 1.199 | – | 2.077 | 1.215 | 0.950 | 2.021 | 2.116 | 2.111 | 0.955 | 0.937 | 1.682 | 1.671 | 1.470 | 1.465 | 1.651 | 1.966 | 1.736 | 1.425 | 1.782 |
| **LevHAR** | **[0.577]** | **[0.577]** | **[0.481]** | – | **[0.585]** | **[0.457]** | 0.973 | 1.019 | 1.017 | **[0.460]** | **[0.451]** | **0.810** | **0.805** | **(0.708)** | **(0.705)** | **0.795** | 0.947 | 0.836 | **[0.686]** | 0.858 |
| **SHAR** | 0.987 | 0.987 | **[0.823]** | 1.710 | – | **[0.782]** | 1.664 | 1.742 | 1.738 | **[0.786]** | **[0.771]** | 1.385 | 1.376 | 1.210 | 1.206 | 1.359 | 1.618 | 1.429 | 1.173 | 1.467 |
| **HARQ** | 1.262 | 1.262 | 1.053 | 2.187 | 1.279 | – | 2.128 | 2.228 | 2.223 | 1.006 | 0.986 | 1.771 | 1.759 | 1.548 | 1.542 | 1.738 | 2.070 | 1.828 | 1.501 | 1.876 |
| **RR** | **[0.593]** | **[0.593]** | **[0.495]** | 1.027 | **[0.601]** | **[0.470]** | – | 1.047 | 1.044 | **[0.473]** | **[0.463]** | **[0.832]** | **[0.827]** | **[0.727]** | **[0.725]** | **[0.817]** | 0.972 | **[0.859]** | **[0.705]** | **[0.881]** |
| **LA** | **[0.567]** | **[0.567]** | **[0.473]** | 0.982 | **[0.574]** | **[0.449]** | **[0.955]** | – | 0.998 | **[0.451]** | **[0.443]** | **[0.795]** | **[0.790]** | **[0.695]** | **[0.692]** | **[0.780]** | **(0.929)** | **[0.821]** | **[0.674]** | **[0.842]** |
| **EN** | **[0.568]** | **[0.568]** | **[0.474]** | 0.984 | **[0.575]** | **[0.450]** | **[0.957]** | 1.002 | – | **[0.452]** | **[0.444]** | **[0.797]** | **[0.792]** | **[0.696]** | **[0.694]** | **[0.782]** | **(0.931)** | **[0.822]** | **[0.675]** | **[0.844]** |
| **BG** | 1.255 | 1.255 | 1.047 | 2.174 | 1.272 | 0.994 | 2.116 | 2.215 | 2.210 | – | 0.980 | 1.761 | 1.750 | 1.539 | 1.533 | 1.728 | 2.058 | 1.818 | 1.492 | 1.865 |
| **RF** | 1.280 | 1.280 | 1.068 | 2.217 | 1.297 | 1.014 | 2.158 | 2.259 | 2.254 | 1.020 | – | 1.796 | 1.784 | 1.570 | 1.564 | 1.763 | 2.099 | 1.854 | 1.522 | 1.902 |
| **GB** | **[0.713]** | **[0.713]** | **[0.595]** | 1.235 | **[0.722]** | **[0.565]** | 1.202 | 1.258 | 1.255 | **[0.568]** | **[0.557]** | – | 0.994 | **[0.874]** | **[0.871]** | 0.981 | 1.169 | 1.032 | **[0.847]** | 1.059 |
| $\mathbf{NN}_1^1$ | **[0.717]** | **[0.717]** | **[0.598]** | 1.243 | **[0.727]** | **[0.568]** | 1.210 | 1.266 | 1.263 | **[0.572]** | **[0.560]** | 1.006 | – | **[0.880]** | **[0.876]** | 0.988 | 1.176 | 1.039 | **[0.853]** | 1.066 |
| $\mathbf{NN}_1^{10}$ | **[0.815]** | **[0.815]** | **[0.680]** | 1.413 | **[0.826]** | **[0.646]** | 1.375 | 1.439 | 1.436 | **[0.650]** | **[0.637]** | 1.144 | 1.137 | – | 0.996 | 1.123 | 1.337 | 1.181 | **(0.969)** | 1.212 |
| $\mathbf{NN}_2^1$ | **[0.818]** | **[0.818]** | **[0.683]** | 1.418 | **[0.829]** | **[0.648]** | 1.380 | 1.445 | 1.441 | **[0.652]** | **[0.639]** | 1.148 | 1.141 | 1.004 | – | 1.127 | 1.342 | 1.185 | **[0.973]** | 1.216 |
| $\mathbf{NN}_2^{10}$ | **[0.726]** | **[0.726]** | **[0.606]** | 1.258 | **[0.736]** | **[0.575]** | 1.224 | 1.282 | 1.279 | **[0.579]** | **[0.567]** | 1.019 | 1.012 | **[0.891]** | **[0.887]** | – | 1.191 | 1.052 | **[0.863]** | 1.079 |
| $\mathbf{NN}_3^1$ | **[0.610]** | **[0.610]** | **[0.509]** | 1.057 | **[0.618]** | **[0.483]** | 1.028 | 1.076 | 1.074 | **[0.486]** | **[0.476]** | **[0.856]** | **[0.850]** | **[0.748]** | **[0.745]** | **[0.840]** | – | **[0.883]** | **[0.725]** | **[0.906]** |
| $\mathbf{NN}_3^{10}$ | **[0.690]** | **[0.690]** | **[0.576]** | 1.196 | **[0.700]** | **[0.547]** | 1.164 | 1.219 | 1.216 | **[0.550]** | **[0.539]** | 0.969 | **(0.963)** | **[0.847]** | **[0.844]** | **[0.951]** | 1.132 | – | **[0.821]** | 1.026 |
| $\mathbf{NN}_4^1$ | **[0.841]** | **[0.841]** | **[0.702]** | 1.457 | **[0.852]** | **[0.666]** | 1.418 | 1.485 | 1.481 | **[0.670]** | **[0.657]** | 1.180 | 1.173 | 1.032 | 1.028 | 1.158 | 1.379 | 1.218 | – | 1.250 |
| $\mathbf{NN}_4^{10}$ | **[0.673]** | **[0.673]** | **[0.561]** | 1.166 | **[0.682]** | **[0.533]** | 1.135 | 1.188 | 1.185 | **[0.536]** | **[0.526]** | 0.944 | **[0.938]** | **[0.825]** | **[0.822]** | **[0.927]** | 1.103 | **[0.974]** | **[0.800]** | – |

*Notes:* Each entry $(i, j)$ reports the out-of-sample QLIKE loss of row model $i$ relative to column model $j$, computed over the test period 2016-04-21 to 2020-08-06 (1,099 observations). A value below unity indicates that column model $i$ outperforms row model $j$. Significant values denote rejection of equal predictive accuracy in favour of the column model based on one-sided Diebold and Mariano (1995) tests with Newey–West standard errors (5 lags). The notation for significance level is represented as follows; **number** denotes $p < 0.10$, **(number)** denotes $p < 0.05$, and **[number]** denotes $p < 0.01$.

Third, beyond the comparisons against HAR and HARQ, two within-table cross-comparisons are worth noting. The LA and EN rows remain nearly identical, further supporting the conclusion that the EN specification frequently collapses onto LA on the restricted three predictor $\mathcal{M}_{\text{HAR}}$ feature set. Within the NN family, differentiation across architecture depth and across best-seed versus top-10 ensemble is small relative to the broader gap between the NN family as a whole and the HAR benchmark.

### 5.1.3 Cross Loss Function Synthesis

Taken together, Tables (3) and (4) tell a compelling story. Under the MSE loss function, only HARQ and LogHAR reject the null hypothesis of equal predictive accuracy against the HAR benchmark, and both only at a 10% significance level, whereas no regularized linear model, tree-based method, or NN specification deliver a significant improvement relative to HAR. Under the QLIKE loss function, the relative ranking of the models remain broadly unchanged, but the relative gap between models in forecast accuracy widens. HARQ remains dominant, and the number of models that significantly outperform HAR expands from two to four. Specifically, HARQ and LogHAR strengthen from significance at the 10% level under MSE to the 1% under QLIKE, while BG and RF also enter the rejection region at the 1% level. The regularized methods, GB, and the NN family fail to improve under QLIKE. In fact, the corresponding Diebold and Mariano (1995) test statistic instead reject equal predictive accuracy in favor of HAR at the 1% level for all these specifications.

Above all, the $\mathcal{M}_{\text{HAR}}$ results support three key findings. First, HARQ is the best performing model on the restricted three predictor information set under both the MSE and QLIKE loss functions. Second, the QLIKE comparison expands the set of models that significantly outperform HAR from two to four, adding BG and RF to HARQ and LogHAR, while the regularized linear models, GB, and the NN specifications are revealed as significantly less accurate than HAR on the proportional-error metric. Third, the joint evidence across the two loss functions indicate that the $\mathcal{M}_{\text{HAR}}$ feature set leaves little room for more complex non-HAR specifications to consistently overturn the leading benchmark at the one-day-ahead forecasting horizon. This finding motivates the extension to the richer $\mathcal{M}_{\text{ALL}}$ feature set, which incorporates additional macroeconomic and financial predictors

## 5.2 Results from $\mathcal{M}_{\text{ALL}}$ Feature Set

This subsection presents the out-of-sample forecasting results for all models estimated using the $\mathcal{M}_{\text{ALL}}$ feature set. As previously noted, the evaluation period consists of 1,099 trading days spanning from 2016-04-06 to 2020-08-06. The five HAR benchmark models remain unchanged from the $\mathcal{M}_{\text{HAR}}$ setting and serve as fixed reference forecasts, whereas the HAR-X model is re-estimated using the extended $\mathcal{M}_{\text{ALL}}$ feature set, along with the

regularized, tree-based, and NN models. Pairwise relative MSE results are reported in Table (5) and the QLIKE counterpart is presented in Table (6). Entries below one indicate that the column model produces a lower expected loss than the row model and statistical significance is assessed using the one-sided Diebold and Mariano (1995) test.

### 5.2.1 Pairwise Relative MSE

Analyzing Table (5) yields three important findings. First, the rejection front against the HAR benchmark widens substantially relative to the $\mathcal{M}_{\text{HAR}}$ results. The HAR row now records statistically significant rejections at the 10% level for HAR-X (relative MSE of 0.929), LogHAR (0.936), HARQ (0.846), LA (0.908), EN (0.908), RF (0.852), $\text{NN}_1^{10}$ (0.880), $\text{NN}_2^1$ (0.881) and $\text{NN}_2^{10}$ (0.895). Whereas the corresponding $\mathcal{M}_{\text{HAR}}$ specification identified only HARQ and LogHAR as significant improvements relative to HAR at the 10% significance level, the addition of the macroeconomic and financial variables in the $\mathcal{M}_{\text{ALL}}$ feature set introduces six new model specifications into the rejection region. The largest gains relative to HAR are obtained by HARQ and RF, which reduce the MSE by approximately 15.4% and 14.8%, respectively. These are followed by the shallow NN ensembles ($\text{NN}_1^{10}$ and $\text{NN}_2^1$), which improve forecast accuracy by roughly 12%, while the regularized LA and EN models both improve upon HAR by approximately 9.2%. These results suggest that the richer $\mathcal{M}_{\text{ALL}}$ set allows several ML models to extract incremental predictive information which was not available under the $\mathcal{M}_{\text{HAR}}$ feature set.

Second, although a larger number of models significantly outperform HAR under the $\mathcal{M}_{\text{ALL}}$ feature set, no specification delivers a statistically significant improvement relative to HARQ under the MSE loss function. The HARQ row of Table (5) records relative MSE entries ranging from 1.006 for RF to 1.218 for SHAR, yet none of these models produce statistically significant Diebold and Mariano (1995) test rejections. RF emerges as the closest competitor to HARQ, with a relative MSE value of 1.006, but even this specification fails to reject the null hypothesis of equal predictive accuracy. The persistence of HARQ as the leading model is therefore in line with the conclusions drawn from the $\mathcal{M}_{\text{HAR}}$ feature set. Even when the information set expands from three lagged realized variance components to the substantially richer $\mathcal{M}_{\text{ALL}}$ feature set, the explicit correction for measurement error in HARQ continues to deliver superior forecast accuracy under MSE and it proves hard to beat even for more complex nonlinear specifications.

Third, several internal patterns within the $\mathcal{M}_{\text{ALL}}$ model specifications are economically informative. The LA and EN columns again produce nearly identical results, confirming that cross-validation continues to push the EN $L_1$ mixing parameter onto the LA boundary in our European single-index setting. Within the tree-based family, RF clearly dominates BG (0.852 relative to 0.895 against HAR), while GB underperforms both alternative tree-based specifications with a relative score of 0.922 against HAR. Within the NN family, the best performing specifications are the comparatively shallow architectures. In particular, $\text{NN}_1^{10}$, $\text{NN}_2^1$, and $\text{NN}_2^{10}$ enter the rejection region against HAR, whereas the deeper $\text{NN}_3$ and

**Table 5:** Pairwise Relative MSE and Diebold-Mariano Tests on $\mathcal{M}_{\text{ALL}}$ Feature Set

| | Benchmark Models | | | | | | Regularized Models | | | Tree-Based Models | | | Neural Networks | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **HAR** | **HAR-X** | **LogHAR** | **LevHAR** | **SHAR** | **HARQ** | **RR** | **LA** | **EN** | **BG** | **RF** | **GB** | $\mathbf{NN}_1^1$ | $\mathbf{NN}_1^{10}$ | $\mathbf{NN}_2^1$ | $\mathbf{NN}_2^{10}$ | $\mathbf{NN}_3^1$ | $\mathbf{NN}_3^{10}$ | $\mathbf{NN}_4^1$ | $\mathbf{NN}_4^{10}$ |
| **HAR** | – | **0.929** | **0.936** | 0.951 | 1.031 | **0.846** | 0.922 | **0.908** | **0.908** | 0.895 | **0.852** | 0.922 | 0.944 | **0.880** | **0.881** | **0.895** | 0.924 | 0.943 | 0.961 | 0.946 |
| **HAR-X** | 1.077 | – | 1.008 | 1.024 | 1.110 | **0.911** | 0.993 | 0.978 | 0.978 | 0.964 | 0.917 | 0.992 | 1.017 | **0.947** | 0.949 | 0.963 | 0.995 | 1.015 | 1.035 | 1.018 |
| **LogHAR** | 1.068 | 0.992 | – | 1.016 | 1.101 | **0.904** | 0.985 | 0.970 | 0.971 | 0.956 | 0.910 | 0.985 | 1.009 | **0.940** | **0.941** | **0.956** | 0.987 | 1.008 | 1.027 | 1.010 |
| **LevHAR** | 1.052 | 0.977 | 0.984 | – | 1.084 | **0.890** | 0.970 | 0.955 | 0.955 | 0.941 | **0.896** | 0.969 | 0.993 | 0.925 | 0.927 | 0.941 | 0.972 | 0.992 | 1.011 | 0.994 |
| **SHAR** | 0.970 | **0.901** | 0.908 | 0.922 | – | **0.821** | 0.894 | **0.881** | **0.881** | 0.868 | **0.826** | 0.894 | 0.916 | **0.853** | **0.855** | **0.868** | 0.896 | 0.915 | 0.932 | 0.917 |
| **HARQ** | 1.182 | 1.098 | 1.106 | 1.124 | 1.218 | – | 1.089 | 1.073 | 1.074 | 1.058 | 1.006 | 1.089 | 1.116 | 1.039 | 1.041 | 1.057 | 1.092 | 1.114 | 1.136 | 1.117 |
| **RR** | 1.085 | 1.008 | 1.015 | 1.031 | 1.118 | **0.918** | – | **0.985** | **0.985** | 0.971 | 0.924 | 1.000 | 1.024 | **0.954** | **0.956** | **(0.971)** | 1.002 | 1.023 | 1.042 | 1.026 |
| **LA** | 1.101 | 1.023 | 1.030 | 1.047 | 1.135 | 0.932 | 1.015 | – | 1.000 | 0.985 | 0.938 | 1.015 | 1.040 | 0.968 | 0.970 | 0.985 | 1.017 | 1.038 | 1.058 | 1.041 |
| **EN** | 1.101 | 1.023 | 1.030 | 1.047 | 1.135 | 0.931 | 1.015 | 1.000 | – | 0.985 | 0.937 | 1.014 | 1.040 | 0.968 | 0.970 | 0.985 | 1.017 | 1.038 | 1.058 | 1.041 |
| **BG** | 1.117 | 1.038 | 1.046 | 1.062 | 1.152 | 0.945 | 1.030 | 1.015 | 1.015 | – | **0.951** | 1.030 | 1.055 | 0.983 | 0.984 | 1.000 | 1.032 | 1.054 | 1.074 | 1.056 |
| **RF** | 1.174 | 1.091 | 1.099 | 1.117 | 1.210 | 0.994 | 1.083 | 1.067 | 1.067 | 1.051 | – | 1.082 | 1.109 | 1.033 | 1.035 | 1.051 | 1.085 | 1.107 | 1.129 | 1.110 |
| **GB** | 1.085 | 1.008 | 1.016 | 1.032 | 1.118 | 0.918 | 1.000 | 0.986 | 0.986 | 0.971 | 0.924 | – | 1.025 | 0.954 | 0.956 | 0.971 | 1.002 | 1.023 | 1.043 | 1.026 |
| $\mathbf{NN}_1^1$ | 1.059 | 0.984 | 0.991 | 1.007 | 1.092 | **0.896** | 0.976 | 0.962 | 0.962 | 0.948 | **0.902** | 0.976 | – | **0.931** | **0.933** | **0.948** | 0.978 | 0.999 | 1.018 | 1.001 |
| $\mathbf{NN}_1^{10}$ | 1.137 | 1.056 | 1.064 | 1.081 | 1.172 | 0.962 | 1.048 | 1.033 | 1.033 | 1.018 | 0.968 | 1.048 | 1.074 | – | 1.002 | 1.017 | 1.050 | 1.072 | 1.093 | 1.075 |
| $\mathbf{NN}_2^1$ | 1.135 | 1.054 | 1.062 | 1.079 | 1.170 | 0.960 | 1.046 | 1.031 | 1.031 | 1.016 | 0.966 | 1.046 | 1.072 | 0.998 | – | 1.016 | 1.048 | 1.070 | 1.091 | 1.073 |
| $\mathbf{NN}_2^{10}$ | 1.118 | 1.038 | 1.046 | 1.063 | 1.152 | 0.946 | 1.030 | 1.015 | 1.015 | 1.000 | 0.952 | 1.030 | 1.055 | 0.983 | 0.985 | – | 1.032 | 1.054 | 1.074 | 1.057 |
| $\mathbf{NN}_3^1$ | 1.082 | 1.005 | 1.013 | 1.029 | 1.116 | 0.916 | 0.998 | 0.983 | 0.983 | 0.969 | 0.922 | 0.998 | 1.022 | 0.952 | 0.954 | 0.969 | – | 1.021 | 1.040 | 1.024 |
| $\mathbf{NN}_3^{10}$ | 1.060 | 0.985 | 0.993 | 1.008 | 1.093 | 0.897 | 0.978 | 0.963 | 0.963 | 0.949 | 0.903 | 0.977 | 1.001 | 0.933 | 0.934 | 0.949 | **(0.980)** | – | 1.019 | 1.003 |
| $\mathbf{NN}_4^1$ | 1.041 | 0.967 | 0.974 | 0.989 | 1.073 | **0.880** | 0.959 | **0.945** | **0.945** | 0.931 | 0.886 | 0.959 | 0.983 | 0.915 | **0.917** | **0.931** | **(0.961)** | **(0.981)** | – | 0.984 |
| $\mathbf{NN}_4^{10}$ | 1.058 | 0.982 | 0.990 | 1.006 | 1.090 | **0.895** | 0.975 | **0.961** | **0.961** | 0.947 | 0.901 | 0.975 | 0.999 | **0.930** | **0.932** | **0.946** | **[0.977]** | 0.997 | 1.016 | – |

*Notes:* Each entry $(i, j)$ reports the out-of-sample MSE of row model $i$ relative to column model $j$, computed over the test period 2016-04-21 to 2020-08-06 (1,099 observations). A value below unity indicates that column model $i$ outperforms row model $j$. Significant values denote rejection of equal predictive accuracy in favor of the column model based on one-sided Diebold and Mariano (1995) tests with Newey-West standard errors (5 lags). The notation for significance level is represented as follows; **number** denotes $p < 0.10$, **(number)** denotes $p < 0.05$, and **[number]** denotes $p < 0.01$.

$NN_4$ architectures cluster slightly below one but remain statistically indistinguishable from the HAR benchmark. Moreover, the top-10 ensembles do not systematically outperform their respective best-seed counterparts within a given architecture.

Taken together, the relative $\mathcal{M}_{\text{ALL}}$ MSE results differ materially from the findings from the corresponding $\mathcal{M}_{\text{HAR}}$ feature set. The HAR benchmark no longer acts as a sharp separator, given that nine alternative models now produce significantly lower MSE values than HAR at the 10% significance level. The competitive landscape in terms of predictive performance therefore shifts from a narrow HAR versus HARQ and LogHAR comparison under the restricted information set toward a substantially broader set of competitive regularized, tree-based, and shallow NN specifications that produce forecasts which are statistically different from HAR. Nevertheless, HARQ remains the best performing model on the EURO STOXX 50 index under the MSE loss function in our setting.

### 5.2.2 Pairwise Relative QLIKE

Before turning to the main QLIKE analysis in Table (6), two preliminary observations are warranted. Under the $\mathcal{M}_{\text{ALL}}$ feature set, the HAR-X and RR columns produce extremely large relative QLIKE values, where the entries relative to the HAR benchmark equal 1685.4 and 44.9 respectively. The corresponding rows mirror these magnitudes. A closer inspection reveals that these extreme values stem from the fact that the HAR-X and RR models frequently produce negative forecasts during the test period. Since realized variance is strictly non-negative, these forecasts are replaced via the insanity filter using the minimum realized variance observed during the corresponding training window. However, given that these minimum replacements are very small and occur frequently, the resulting QLIKE scores effectively explode and become excessively large, as documented in Table (12) in Appendix (4.6). The economic interpretation of these results is that the HAR-X and RR, on the $\mathcal{M}_{\text{ALL}}$ feature set, fail to generate forecasts that remain within the empirical boundaries for realized variance at a sufficiently high rate to be evaluated accurately under the proportional-error structure of the QLIKE loss function. With this caveat in mind, three key observations follow.

First, the QLIKE-based rejection front against HAR contains five models. The HAR row of Table (6) records statistically significant rejections at the 1% level for LogHAR (relative QLIKE of 0.834), HARQ (0.792), BG (0.739), RF (0.732), and GB (0.887). The LogHAR and HARQ entries remain mechanically identical to those reported under the $\mathcal{M}_{\text{HAR}}$ feature set by construction, whereas the primary changes occur within the tree-based family. Relative to the $\mathcal{M}_{\text{HAR}}$ results, BG and RF strengthen further, with QLIKE values relative to HAR that decrease from 0.797 and 0.781 to 0.739 and 0.732, respectively, under $\mathcal{M}_{\text{ALL}}$. Furthermore, GB transitions from a relative QLIKE score against HAR with no rejection of 1.403 under $\mathcal{M}_{\text{HAR}}$ to a relative score of 0.887 with rejection at the 1% level under $\mathcal{M}_{\text{ALL}}$. These improvements in forecast accuracy suggest that the additional macroeconomic and financial covariates deliver tangible QLIKE gains for all tree-based

**Table 6:** Pairwise Relative QLIKE and Diebold-Mariano Tests on $\mathcal{M}_{\text{ALL}}$ Feature Set

| | Benchmark Models | | | | | | Regularized Models | | | Tree-Based Models | | | Neural Networks | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **HAR** | **HAR-X** | **LogHAR** | **LevHAR** | **SHAR** | **HARQ** | **RR** | **LA** | **EN** | **BG** | **RF** | **GB** | $\mathbf{NN}_1^1$ | $\mathbf{NN}_1^{10}$ | $\mathbf{NN}_2^1$ | $\mathbf{NN}_2^{10}$ | $\mathbf{NN}_3^1$ | $\mathbf{NN}_3^{10}$ | $\mathbf{NN}_4^1$ | $\mathbf{NN}_4^{10}$ |
| **HAR** | – | 1685.437 | **[0.834]** | 1.733 | 1.013 | **[0.792]** | 44.909 | 1.090 | 1.093 | **[0.739]** | **[0.732]** | **[0.887]** | 1.103 | 1.315 | 1.317 | 1.174 | 1.136 | 1.345 | 1.559 | 1.661 |
| **HAR-X** | **[0.001]** | – | **[0.000]** | **[0.001]** | **[0.001]** | **[0.000]** | **[0.027]** | **[0.001]** | **[0.001]** | **[0.000]** | **[0.000]** | **[0.001]** | **[0.001]** | **[0.001]** | **[0.001]** | **[0.001]** | **[0.001]** | **[0.001]** | **[0.001]** | **[0.001]** |
| **LogHAR** | 1.199 | 2020.422 | – | 2.077 | 1.215 | 0.950 | 53.835 | 1.307 | 1.310 | **[0.885]** | **[0.877]** | 1.063 | 1.322 | 1.577 | 1.579 | 1.408 | 1.361 | 1.612 | 1.869 | 1.991 |
| **LevHAR** | **[0.577]** | 972.825 | **[0.481]** | – | **[0.585]** | **[0.457]** | 25.921 | **[0.629]** | **[0.631]** | **[0.426]** | **[0.422]** | **[0.512]** | **[0.636]** | **(0.759)** | **(0.760)** | **[0.678]** | **[0.655]** | **(0.776)** | 0.900 | 0.959 |
| **SHAR** | 0.987 | 1663.146 | **[0.823]** | 1.710 | – | **[0.782]** | 44.315 | 1.076 | 1.078 | **[0.729]** | **[0.722]** | **[0.875]** | 1.088 | 1.298 | 1.299 | 1.159 | 1.120 | 1.327 | 1.538 | 1.639 |
| **HARQ** | 1.262 | 2127.090 | 1.053 | 2.187 | 1.279 | – | 56.678 | 1.376 | 1.379 | **(0.932)** | **[0.923]** | 1.119 | 1.392 | 1.660 | 1.662 | 1.482 | 1.433 | 1.697 | 1.968 | 2.096 |
| **RR** | **[0.022]** | 37.530 | **[0.019]** | **[0.039]** | **[0.023]** | **[0.018]** | – | **[0.024]** | **[0.024]** | **[0.016]** | **[0.016]** | **[0.020]** | **[0.025]** | **[0.029]** | **[0.029]** | **[0.026]** | **[0.025]** | **[0.030]** | **[0.035]** | **[0.037]** |
| **LA** | **(0.917)** | 1545.679 | **[0.765]** | 1.589 | **0.929** | **[0.727]** | 41.186 | – | 1.002 | **[0.677]** | **[0.671]** | **[0.813]** | 1.011 | 1.206 | 1.208 | 1.077 | 1.041 | 1.233 | 1.430 | 1.523 |
| **EN** | **(0.915)** | 1542.180 | **[0.763]** | 1.585 | **(0.927)** | **[0.725]** | 41.092 | **[0.998]** | – | **[0.676]** | **[0.669]** | **[0.811]** | 1.009 | 1.203 | 1.205 | 1.074 | 1.039 | 1.230 | 1.426 | 1.520 |
| **BG** | 1.354 | 2281.905 | 1.129 | 2.346 | 1.372 | 1.073 | 60.803 | 1.476 | 1.480 | – | 0.991 | 1.201 | 1.493 | 1.781 | 1.783 | 1.590 | 1.537 | 1.821 | 2.111 | 2.249 |
| **RF** | 1.367 | 2303.578 | 1.140 | 2.368 | 1.385 | 1.083 | 61.380 | 1.490 | 1.494 | 1.009 | – | 1.212 | 1.507 | 1.798 | 1.800 | 1.605 | 1.552 | 1.838 | 2.131 | 2.270 |
| **GB** | 1.128 | 1900.558 | 0.941 | 1.954 | 1.143 | **(0.894)** | 50.641 | 1.230 | 1.232 | **[0.833]** | **[0.825]** | – | 1.243 | 1.483 | 1.485 | 1.324 | 1.280 | 1.516 | 1.758 | 1.873 |
| $\mathbf{NN}_1^1$ | **[0.907]** | 1528.630 | **[0.757]** | 1.571 | **[0.919]** | **[0.719]** | 40.731 | 0.989 | 0.991 | **[0.670]** | **[0.664]** | **[0.804]** | – | 1.193 | 1.194 | 1.065 | 1.030 | 1.220 | 1.414 | 1.507 |
| $\mathbf{NN}_1^{10}$ | **[0.760]** | 1281.469 | **[0.634]** | 1.317 | **[0.771]** | **[0.602]** | 34.146 | **[0.829]** | **[0.831]** | **[0.562]** | **[0.556]** | **[0.674]** | **[0.838]** | – | 1.001 | **[0.893]** | **[0.863]** | 1.022 | 1.185 | 1.263 |
| $\mathbf{NN}_2^1$ | **[0.759]** | 1279.914 | **[0.633]** | 1.316 | **[0.770]** | **[0.602]** | 34.104 | **[0.828]** | **[0.830]** | **[0.561]** | **[0.556]** | **[0.673]** | **[0.837]** | 0.999 | – | **[0.892]** | **[0.862]** | 1.021 | 1.184 | 1.261 |
| $\mathbf{NN}_2^{10}$ | **[0.852]** | 1435.336 | **[0.710]** | 1.475 | **[0.863]** | **[0.675]** | 38.245 | **(0.929)** | **0.931** | **[0.629]** | **[0.623]** | **[0.755]** | **[0.939]** | 1.120 | 1.121 | – | 0.967 | 1.145 | 1.328 | 1.415 |
| $\mathbf{NN}_3^1$ | **[0.881]** | 1484.306 | **[0.735]** | 1.526 | **[0.892]** | **[0.698]** | 39.550 | **0.960** | 0.962 | **[0.650]** | **[0.644]** | **[0.781]** | 0.971 | 1.158 | 1.160 | 1.034 | – | 1.184 | 1.373 | 1.463 |
| $\mathbf{NN}_3^{10}$ | **[0.744]** | 1253.408 | **[0.620]** | 1.288 | **[0.754]** | **[0.589]** | 33.398 | **[0.811]** | **[0.813]** | **[0.549]** | **[0.544]** | **[0.659]** | **[0.820]** | **[0.978]** | **(0.979)** | **[0.873]** | **[0.844]** | – | 1.159 | 1.235 |
| $\mathbf{NN}_4^1$ | **[0.641]** | 1081.104 | **[0.535]** | 1.111 | **[0.650]** | **[0.508]** | 28.807 | **[0.699]** | **[0.701]** | **[0.474]** | **[0.469]** | **[0.569]** | **[0.707]** | **[0.844]** | **[0.845]** | **[0.753]** | **[0.728]** | **[0.863]** | – | 1.065 |
| $\mathbf{NN}_4^{10}$ | **[0.602]** | 1014.669 | **[0.502]** | 1.043 | **[0.610]** | **[0.477]** | 27.036 | **[0.656]** | **[0.658]** | **[0.445]** | **[0.440]** | **[0.534]** | **[0.664]** | **[0.792]** | **[0.793]** | **[0.707]** | **[0.684]** | **[0.810]** | **[0.939]** | – |

*Notes:* Each entry $(i, j)$ reports the out-of-sample QLIKE loss of row model $i$ relative to column model $j$, computed over the test period 2016-04-21 to 2020-08-06 (1,099 observations). A value below unity indicates that column model $i$ outperforms row model $j$. Significant values denote rejection of equal predictive accuracy in favour of the column model based on one-sided Diebold and Mariano (1995) tests with Newey–West standard errors (5 lags). The notation for significance level is represented as follows; **number** denotes $p < 0.10$, **(number)** denotes $p < 0.05$, and **[number]** denotes $p < 0.01$.

models. In particular, GB appears to benefit the most from the broader information set, which indicates that boosting methods are especially well suited for exploiting nonlinear interactions in environments with a large and correlated predictor set, consistent with the dependence structure illustrated in Figure (11) in Appendix (8.2).

Second, the QLIKE comparison narrows the $\mathcal{M}_{\text{ALL}}$ leadership ordering at the top of the model ranking. Whereas the MSE results in Table (5) record that no specification significantly outperforms HARQ, the corresponding QLIKE results now identify both BG and RF as statistically superior to HARQ. The HARQ row of Table (6) records statistically significant rejections in favor for BG at the 5% level and for RF at the 1% level, corresponding to relative QLIKE measures of 0.932 and 0.923 respectively. This represents a meaningful transition relative to both the $\mathcal{M}_{\text{HAR}}$ and the $\mathcal{M}_{\text{ALL}}$ MSE results. Under the restricted $\mathcal{M}_{\text{HAR}}$ feature set, HARQ remained statistically undefeated under both loss functions. Similarly, under the $\mathcal{M}_{\text{ALL}}$ MSE results, RF emerged as the closest competitor to HARQ (0.6% difference in MSE) but failed to reject the Diebold and Mariano (1995) test for equal predictive accuracy. Under the QLIKE metric for the $\mathcal{M}_{\text{ALL}}$ feature set, however, the same RF specification successfully rejects equal predictive accuracy against HARQ at the 1% level, while BG rejects at the 5% level. As such, the proportional-error metric identifies the bootstrap-aggregated tree-based ensembles as the only specifications in our $\mathcal{M}_{\text{ALL}}$ model space capable of statistically dominating the HARQ benchmark in our EURO STOXX 50 single-index setting.

Third, the regularized linear models and NN family fail to obtain the same QLIKE gains as documented for the tree-based ensembles. Although LA and EN produce statistically significant MSE improvements of 9.2% relative to HAR under the $\mathcal{M}_{\text{ALL}}$ feature set, their relative QLIKE values remain above one (1.090 and 1.093 respectively) and statistically indistinguishable from the benchmark. The same pattern holds for the entire NN family. All NN specifications record relative QLIKE entries above one against HAR, ranging from 1.103 ($\text{NN}_1^1$) to 1.661 ($\text{NN}_4^{10}$), again with no rejection. Furthermore, the reverse direction tests in Table (6) reject the null of equal predictive accuracy in favor of HAR at the 1% level for all regularized and NN models. As a result, it is evident that the MSE-based gains delivered by these specifications on $\mathcal{M}_{\text{ALL}}$ do not translate into superior proportional forecast accuracy. Under the QLIKE loss function, HAR therefore continues to dominate the regularized linear models and the entire NN family despite the inclusion of additional macroeconomic and financial variables.

### 5.2.3 Cross Loss Function Synthesis

Taken together, Table (5) and (6) provide a better picture of the influence of the $\mathcal{M}_{\text{ALL}}$ feature set than either loss function considered in isolation. Whereas only HARQ and LogHAR significantly improved upon HAR under the restricted $\mathcal{M}_{\text{HAR}}$ feature set, the inclusion of macroeconomic and financial variables expand the set of models that significantly outperform HAR to nine specifications, that is HAR-X, LogHAR, HARQ, LA, EN,

RF, $\text{NN}_1^{10}$, $\text{NN}_2^1$, and $\text{NN}_2^{10}$. Despite this increase in competitive models, HARQ remains statistically undefeated under the MSE loss function. The QLIKE comparison, however, produces a more selective ranking of model performance. The rejection front against HAR now contains LogHAR, HARQ, BG, RF, and GB, with the three tree-based models reflecting forecast improvements that are attributable to expanding the information set from $\mathcal{M}_{\text{HAR}}$ to the $\mathcal{M}_{\text{ALL}}$ analysis. More consequentially, the QLIKE results identify BG and RF as significant improvements over HARQ at the 5% and 1% level respectively, marking the first instance in the empirical analysis where any specification statistically dominates the HARQ model. In contrast, the regularized linear models LA and EN, as well as the NN architectures, deliver significant MSE improvements over HAR but fail to retain these under the QLIKE loss function. Under the proportional-error metric, these model classes are significantly less accurate compared to the HAR benchmark, suggesting that their MSE gains are driven primarily by improved fit during periods of extreme volatility rather than by consistently stronger proportional accuracy across the test window. The contrast between the MSE and QLIKE rankings is most pronounced for the NNs, whose best performing MSE specification ($\text{NN}_1^{10}$, $\text{NN}_2^1$, and $\text{NN}_2^{10}$) are also among the models most decisively outperformed by HAR under the QLIKE loss function.

Reflecting upon these findings, the $\mathcal{M}_{\text{ALL}}$ results support three main conclusions. First, the macroeconomic and financial covariates contribute to a material expansion of the set of models that significantly outperform HAR under the MSE loss function, but they do not overturn HARQ as the leading benchmark on this metric. While the forecasting performance of virtually all ML specifications improve under the $\mathcal{M}_{\text{ALL}}$ relative to the $\mathcal{M}_{\text{HAR}}$ feature set, indicating that these models are able to extract additional predictive information from the richer information set, these gains are not enough to overturn HARQ as the leading benchmark under MSE. Second, the QLIKE comparison identifies the bootstrapped-aggregated tree-based ensembles, BG and RF, as the only specifications in the $\mathcal{M}_{\text{ALL}}$ model space capable of statistically outperforming HARQ on the proportional-error metric, with RF doing so at a 1% significance level. This suggests that the tree-based models are particularly effective at converting additional information into incremental forecast improvements. Third, the joint interpretation of the MSE and QLIKE results indicates that the MSE-based improvements documented for the regularized linear models and the NN family under the $\mathcal{M}_{\text{ALL}}$ feature set do not necessarily translate to QLIKE, suggesting that the channels through which these specifications absorb the macroeconomic information differ qualitatively from those exploited by the three-based models. The economic interpretation of these patterns and their implications for the use of ML forecasts across different volatility regimes are discussed further in Section (6).

# 6 Discussion

In this section, we elaborate on the broader implications of the empirical results presented in Section (5) and provide answers to our three research questions. The discussion proceeds in three parts. First, we evaluate how the inclusion of additional macroeconomic and financial predictors affects the forecasting performance of ML approaches in Subsection (6.1). Second, we assess the comparative performance of the different forecasting approaches and discuss which models appear most suitable for forecasting realized volatility on the EURO STOXX 50 index in Subsection (6.2). Third, we analyze the relative importance of the variables across model categories in Subsection (6.3).

## 6.1 Additional Predictors in Machine Learning Models

The transition from the $\mathcal{M}_{\text{HAR}}$ to the $\mathcal{M}_{\text{ALL}}$ feature set holds the model specifications fixed while only expanding the information available to the ML forecasting models. As such, any improvement in out-of-sample performance for a given ML model when moving across feature sets is directly attributable to its ability to extract additional signals from the extended set of macroeconomic and financial predictors. Under the MSE loss function, this extraction is unambiguous and broad-based. As emphasized in the previous section, the relative MSE of all regularized linear models, tree-based models, and feed-forward NN architectures materially improve under the $\mathcal{M}_{\text{ALL}}$ relative to the $\mathcal{M}_{\text{HAR}}$ feature set. Moreover, the set of models that significantly outperform HAR at the 10% significance level under MSE expands from two specifications under $\mathcal{M}_{\text{HAR}}$ (LogHAR and HARQ) to nine under $\mathcal{M}_{\text{ALL}}$ (HAR-X, LogHAR, HARQ, LA, EN, RF, $\text{NN}_1^{10}$, $\text{NN}_2^{1}$, and $\text{NN}_2^{10}$). The size and breadth of this expansion suggests that the specific European macroeconomic and financial predictors in our setting contain meaningful incremental information for forecasting future volatility on the EURO STOXX 50 index, beyond what is contained in lagged realized variance components alone. This finding is consistent with the broader strand of literature emphasizing the role of macroeconomic and financial predictors in forming the basis for volatility dynamics, including Paye (2012), Christiansen et al. (2012), Nonejad (2017), and Engle et al. (2013).

The mechanisms through which the different ML classes exploit this information are also broadly consistent with the respective theoretical motivations. The regularized linear models benefit from the $L_1$ and $L_2$ penalty terms, which selectively absorb informative predictors while shrinking irrelevant and weakly informative coefficients, addressing the high-dimensionality and multicollinearity concerns that arise once the feature set expands beyond the three lagged realized variance components, in line with the general properties documented in Ding et al. (2021). Equivalently, the tree-based models exploit non-additive interactions between the macroeconomic predictors and the lagged realized variance components, capturing precisely the kind of structure that recursive partitioning is designed to identify, consistent with the findings of Yang et al. (2017), Luong and Dokuchaev (2018),

and Mittnik et al. (2015). The shallow neural network architectures benefit from the additional input channels, which allows the network to approximate nonlinear dependencies among predictors, in line with the evidence presented by Miura et al. (2019) and Bucci (2020).

The picture under the QLIKE loss function is more discriminating and reveals that the MSE-based gains are not symmetric across ML classes. Under the $\mathcal{M}_{\text{ALL}}$ feature set, the regularized linear models and the NN family produce relative QLIKE entries against HAR above 1, with LA at 1.090, EN at 1.093, and the NN architectures ranging from 1.103 ($\text{NN}_1^1$) to 1.661 ($\text{NN}_4^{10}$). Reading the corresponding cells in the HAR column, HAR significantly dominates all regularized linear models at the 5% significance level and the corresponding neural network architectures under QLIKE at the 1% significance level. The MSE-based improvements documented above therefore do not translate into improved proportional accuracy across the test sample. The tree-based models, on the other hand, exhibit a different pattern. BG, RF, and GB all improve on HAR under both MSE and QLIKE on the $\mathcal{M}_{\text{ALL}}$ feature set, with RF recording a relative QLIKE of 0.732 against HAR and 0.923 against HARQ, both significant at the 1% level. The improvement of BG and RF on HARQ under QLIKE represents the only instance in our empirical analysis in which any specification statistically dominates HARQ, and constitutes the clearest evidence in our paper that ML methods extract additional incremental information.

The refined answer to the first research question is therefore as follows. The macroeconomic and financial predictors do enable ML models to extract additional signals beyond what is contained in the realized variance components alone, but the channels through which this exploitation occurs vary substantially across model categories and loss functions. Tree-based ensembles extract information that improves performance under both MSE and QLIKE metrics, while regularized linear models and neural networks extract information that improves performance under MSE only. We expand the analysis on the observed heterogeneity of our empirical results in Subsection (6.2).

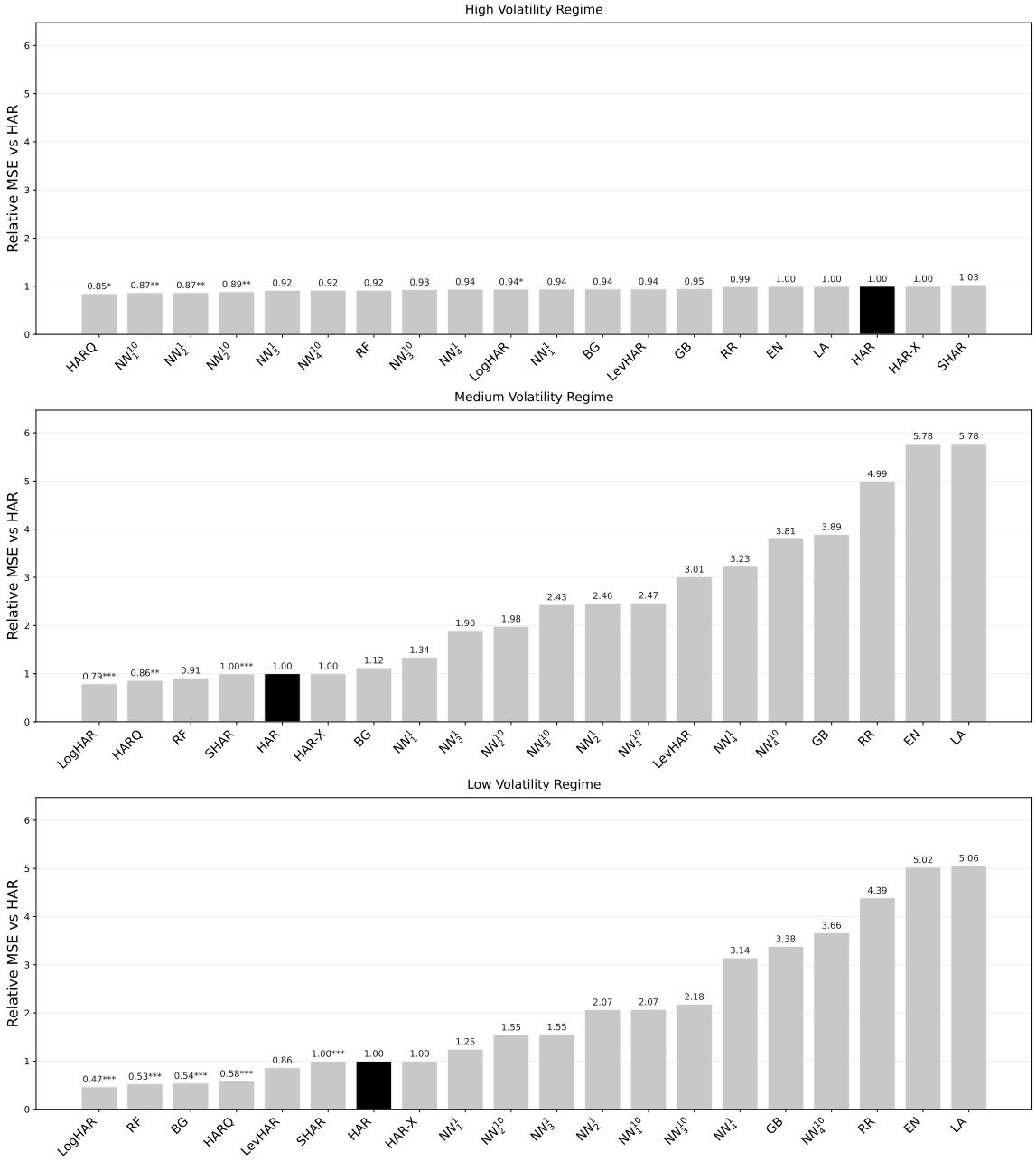
## 6.2 Identifying the Best Performing Model

Identifying the most suitable model for forecasting realized volatility on the EURO STOXX 50 index entails considering both feature sets and loss functions jointly. A specification that records superior performance under MSE but not under QLIKE, or excels on $\mathcal{M}_{\text{HAR}}$ but deteriorates on $\mathcal{M}_{\text{ALL}}$, lack robustness for practitioners seeking consistent forecasting accuracy across evaluation criteria. We find that two specifications meet these requirements. HARQ dominates under the MSE loss function on both feature sets, while also being among the top performers under the QLIKE loss function, with BG and RF serving as the only alternatives that significantly outperforms HARQ under QLIKE on $\mathcal{M}_{\text{ALL}}$. Moreover, RF records material improvements under both loss functions on $\mathcal{M}_{\text{ALL}}$ and constitutes the best alternative when additional information is available.

To better understand why these specifications emerge as robust, we decompose fore-

casting performance by volatility regimes ex-post. Figures (6) and (7) report relative MSE and relative QLIKE against the HAR benchmark across three volatility terciles. The regime decomposition reveals three insights that are not observed in Section (5).

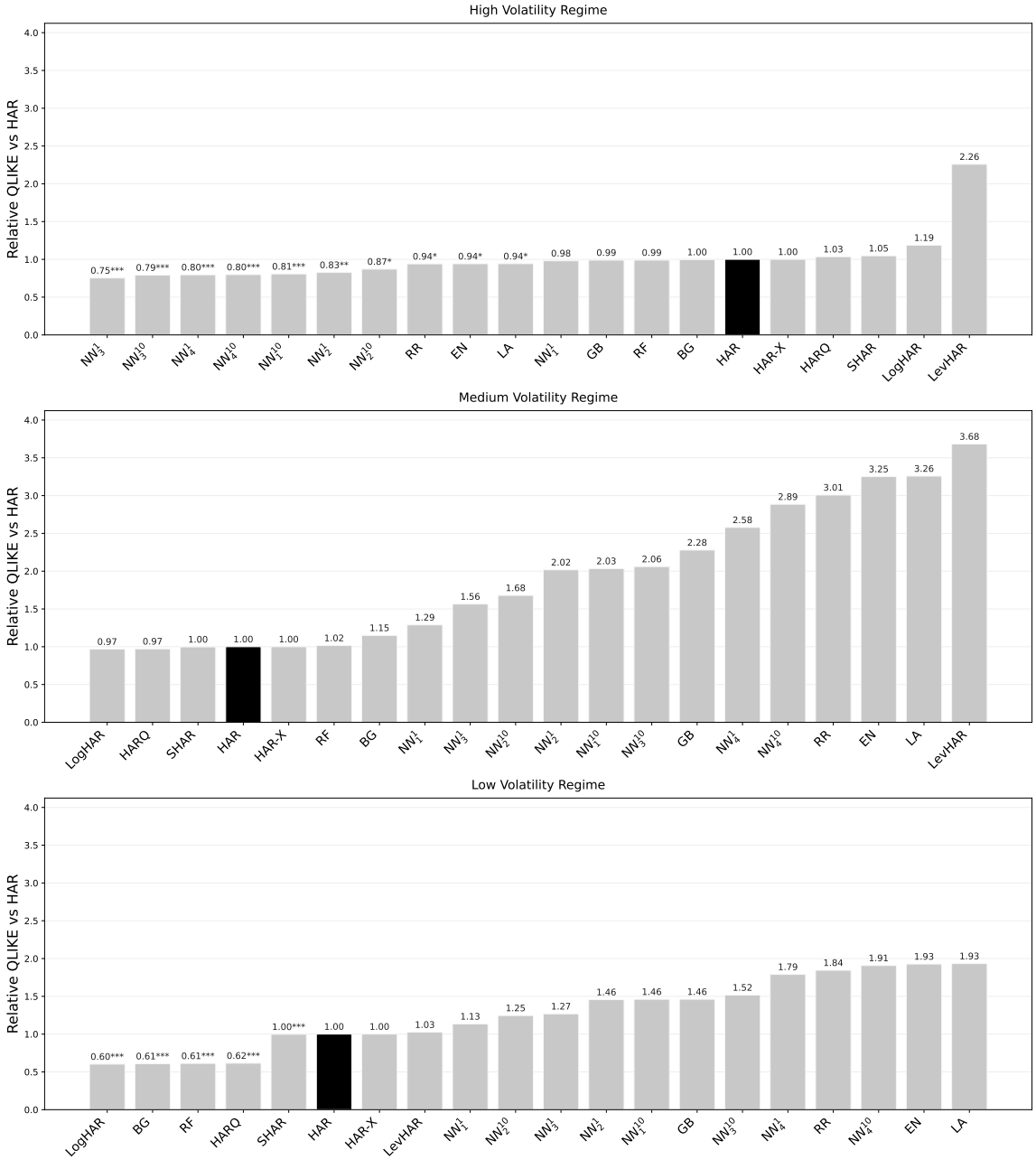
**Figure 6:** Relative MSE under Different Volatility Regimes




*Notes:* The figure depicts relative MSE against the HAR benchmark model under three volatility regimes. The volatility regimes are determined by dividing the actual recorded realized variance into terciles, whereby the high volatility period consists of trading days with equal to or higher than 12.18% annualized volatility, the medium volatility consists of trading days with annualized volatility inbetween 8.34% and 12.18%, and the low volatility period consists of trading days with annualized volatility at or below 8.34%. The notations ***, **, and * denote statistical significance at the 10%, 5%, and 1% levels respectively based on a one-sided Diebold and Mariano (1995) test with Newey West standard errors (max lag 5).

First, the dominance of HARQ is not uniform across volatility regimes and loss functions. In the low volatility regime, HARQ achieves a relative MSE against HAR of 0.58 with statistical significance at the 1% level, but this relative score deteriorates to 0.86

**Figure 7:** Relative QLIKE under Different Volatility Regimes




*Notes:* The figure depicts relative QLIKE against the HAR benchmark model under three volatility regimes. The volatility regimes are determined by dividing the actual recorded realized variance into terciles, whereby the high volatility period consists of trading days with equal to or higher than 12.18% annualized volatility, the medium volatility consists of trading days with annualized volatility inbetween 8.34% and 12.18%, and the low volatility period consists of trading days with annualized volatility at or below 8.34%. The notations ***, **, and * denote statistical significance at the 10%, 5%, and 1% levels, respectively, based on a one-sided Diebold and Mariano (1995) test with Newey West standard errors (max lag 5).

in the medium regime, with significance recorded at the 5% level. A similar pattern is observed with regard to the relative QLIKE score. In the high volatility regime, however, HARQ delivers only a marginal improvement over HAR with a relative MSE of 0.85 at the 10% level, while the relative QLIKE score indicates that HARQ performs marginally worse than HAR. This pattern is consistent with the theoretical motivation of Bollerslev et al. (2016), namely that the adjustment for realized quarticity is most valuable when

realized variance is small and the measurement error dominates the signal-to-noise ratio, whereas during turbulent periods the signal in lagged realized variance is sufficiently strong that the adjustment yields diminishing returns. The economic implication is that HARQ stands out as the superior model not by predicting volatility spikes more accurately, but by avoiding overreactions during the prolonged low-volatility expansion between 2016-2019, which constitutes the majority of our test period.

Second, the regime decomposition reveals a striking divergence between MSE and QLIKE in the regularized linear and NN families. Under MSE, RR, LA, EN, and the NN architectures cluster relatively close to HAR under the high volatility regime, with relative scores ranging from 0.87 to 1.00. This indicates that, apart from the shallow NNs, ($\text{NN}_1^{10}$, $\text{NN}_2^1$, and $\text{NN}_2^{10}$) which outperform HAR at the 5% level, none of these models neither outperform nor underperform HAR under high volatility periods. In the low and medium volatility regimes, however, the same models record relative MSE values ranging from 1.25 to 5.78 against HAR, indicating substantial deterioration during calm periods. Under QLIKE, a similar pattern is observed. Regularized linear models and the NNs record relative scores against HAR ranging from 0.75 to 0.94 in the high volatility period. Notably, all NNs except $\text{NN}_1^1$ statistically outperform HAR in this regime. During the low and medium volatility regimes, however, relative QLIKE scores ranges from 1.13 to 3.68, indicative of poor and dispersed performance across model specifications. This pattern resolves the apparent contradiction in Section (5), where these models improved upon HAR under MSE but underperformed according to QLIKE under the $\mathcal{M}_{\text{ALL}}$ feature set. The MSE gains are concentrated in the tail of the volatility distribution, meaning that models which fit a small number of extreme observations reasonably well are rewarded with significant increases in forecast accuracy, whereas the QLIKE penalty accumulates day-by-day over the tranquil portion of the sample, which is much longer. A model that generates forecasts with somewhat decent accuracy on high-volatility days but systematically underestimate volatility during tranquil periods will look favorable under MSE and unfavorable under QLIKE. Furthermore, Patton (2011) and Patton and Sheppard (2009) discuss this property of MSE in the context of noisy volatility proxies, and the regime decomposition provides an empirical illustration of how proportional-error metrics generate model rankings that differ relative to squared-error metrics in forecasting applications.

Third, the regime decomposition reveals that no single model class dominates HAR uniformly across all three volatility regimes and under both loss functions. Instead, different model families excel under different regimes and the global ranking documented in Section (5) reflects a weighted aggregation of these regime specific patterns. Under MSE, RF achieves its strongest performance in the low volatility regime with a relative score of 0.53 against HAR at the 1% significance level, with the relative gain narrowing to 0.91 in the medium regime and 0.92 in the high regime. BG follows a similar pattern, recording 0.54 in the low regime at the 1% significance level, but deteriorates to 1.12 in the medium regime. Under QLIKE, the picture for the bootstrapped-aggregated tree-based ensembles

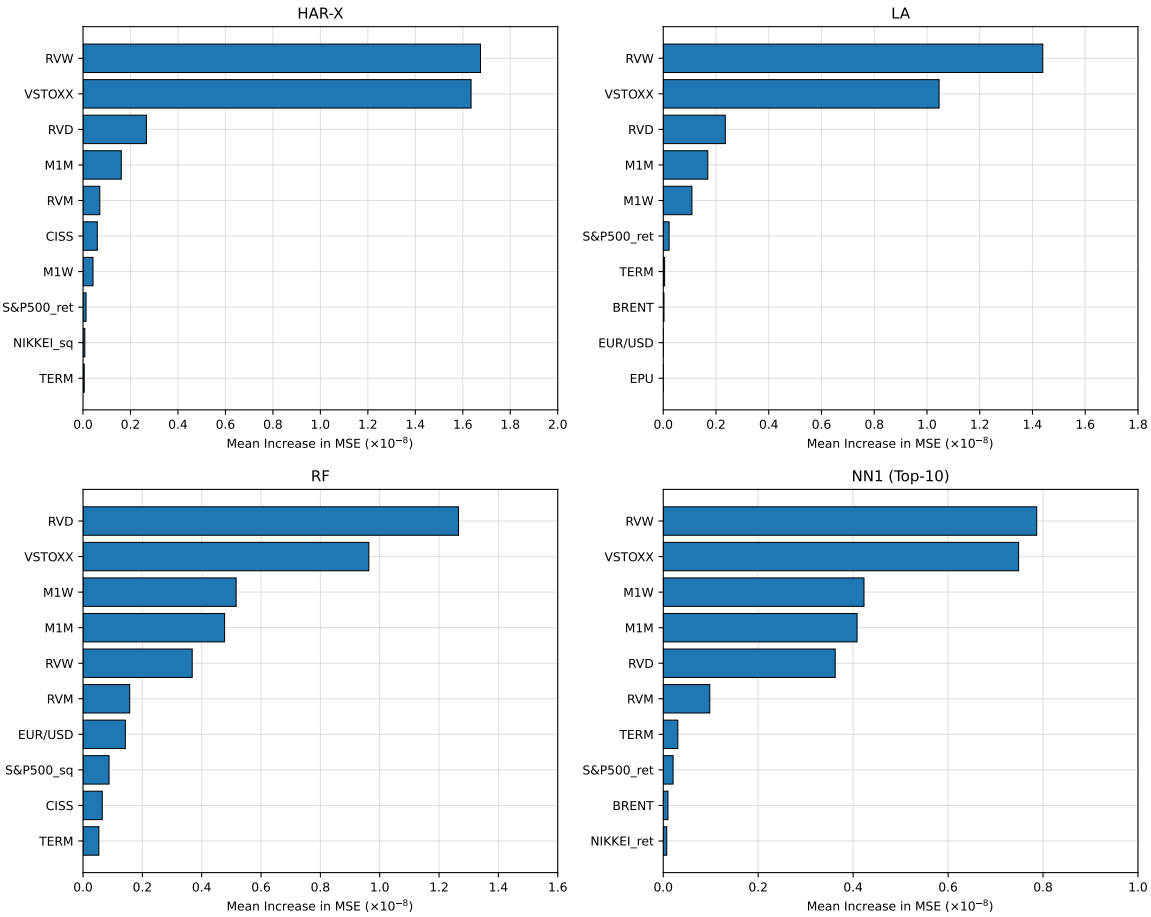
is even more concentrated, where both RF and BG achieve relative scores of 0.61 in the low regime at the 1% level, but converge to approximately 1.00 in the medium and high regimes, where the corresponding Diebold and Mariano (1995) tests fail to reject equal predictive accuracy against HAR. The conclusion is that the QLIKE gains documented for RF and BG in Table (6) are predominantly accumulated during low-volatility periods.

Taken together, the regime decomposition supports a more nuanced answer to the second research question. The practical recommendation for practitioners targeting one-day-ahead realized volatility on the EURO STOXX 50 index depends on both the quality of the information embedded in the predictor set, in line with the findings of Díaz et al. (2024), and the applied loss function criteria. When the feature set is restricted to lagged realized variance components, the HARQ model is the preferred forecasting model in our setting. When the information set is expanded to include macroeconomic and financial variables, RF emerges as a potent specification to improve forecasting performance. This finding aligns broadly with Christensen et al. (2023), who report that RF, along with the NN family, are the best performing models. However, in contrast to their finding, we do not observe that NNs consistently outperform across the full test sample. Our results are instead more consistent with Fernandes et al. (2014) and Vortelinos (2017), who find that the long-memory structure of the HAR framework is hard to beat, even for more complex NN specifications. Nevertheless, the volatility regime analysis supports the argument of Bucci (2020) that NN architectures, estimated on the $\mathcal{M}_{\text{ALL}}$ feature set, deliver superior proportional accuracy relative to other model specifications during heightened volatility periods. This suggests that an ensemble or regime-switching approach combining HARQ and RF for normal conditions with a NN for high volatility episodes may provide further improvements in forecast accuracy. We leave the formal investigation of such regime-shifting model configurations to future research.

## 6.3 Analysis of Variable Importance

To investigate which predictors are most important for forecasting realized variance, we examine the best performing specification from each model category in terms of relative MSE against the HAR benchmark under the $\mathcal{M}_{\text{ALL}}$ feature set. These are HAR-X, LA, RF, and $\text{NN}_1^{10}$. Although HARQ is the strongest benchmark specification overall, we analyze the HAR-X model from the HAR family to facilitate a comparable analysis since it incorporates same set of predictors as the other models. Variable importance is assessed using the Permutation Feature Importance (PFI) method, shown in Figure (8), where the relative importance of a predictor is measured by the average increase in out-of-sample MSE following a random permutation of the corresponding variable.

**Figure 8:** Variable Importance of Realized Variance Forecasts




*Notes:* This figure reports the permutation variable importance for the best performing models from each model family. For a given predictor, importance is measured as the average increase in out-of-sample MSE when the observations of that variable are randomly permuted, thereby corrupting its relationship with the target variable. All importance measures are computed on the test set and are based on the most recent rolling estimation window. Results are averaged over 30 random permutations, where larger values imply that the model is relying heavily on that feature to forecast realized variance, whereas lower values indicate that the specific variable bears little predictive power. Each panel shows the 10 most important variables, see Tables (9) and (10) in Appendix (8.6) for additional information.

Based on the importance rankings across the four models in Figure (8), it is evident that the lagged realized variance measures, particularly weekly realized variance ($RVW$), daily realized variance ($RVD$), and to some extent monthly realized variance ($RVM$), are the dominant drivers of forecasting performance across all model classes. This finding is consistent with the observation of Fernandes et al. (2014), that volatility persistence becomes the dominant feature at short forecasting horizons due to the strong dependence structure of volatility-related measures across horizons, which is further evident from the autocorrelation patterns shown in Figure (10). In addition to the lagged realized variance measures, the $VSTOXX$ index emerges as one of the most important predictors across all specifications. As a proxy for implied volatility, the crucial role of the VSTOXX index for volatility forecasting is in line with the findings of Díaz et al. (2024), who identify the analogous VIX index as one of the most important predictors for all regularized, tree-based, and NN specifications. Similarly, Mittnik et al. (2015) document the VIX index as a key determinant of realized variance within the RF framework, while

Christensen et al. (2023) demonstrate that implied volatility indices contain substantial forward-looking information that enhances forecasting performance.

Beyond these common elements, there are also notable differences across model classes. The HAR-X and LA specifications place overwhelming emphasis on a small subset of predictors, primarily $RVW$, $RVD$, and $VSTOXX$, with only limited incremental contribution from the remaining variables. This pattern is consistent with earlier discussions on the shrinkage properties of LA and economically intuitive given the correlation structure within the predictor set, which causes the LA specification to concentrate explanatory power to a relatively small subset of dominant predictors under the $L_1$ penalty structure. This is also consistent with the findings of Audrino and Knaus (2016), who show that the LA is able to recover the lag structure of the HAR framework when it is assumed to be the true underlying model but fails to do so when estimated on real data. As shown in the PFI results, LA extracts almost no predictive power from additional variables, such as $EPU$, $EUR/USD$, $BRENT$, and $TERM$.

The RF and $\text{NN}_1^{10}$ models, in contrast to the linear specifications, distribute feature importance to a larger subset of predictors. In addition to the dominant lagged realized variance measures and $VSTOXX$, these models are able to extract incremental predictive information from momentum variables ($M1W$ and $M1M$), the EUR/USD exchange rate ($EUR/USD$), the CISS index ($CISS$), term spread ($TERM$), and the S&P 500 return and volatility proxies ($S\&P500_{ret}$ and $S\&P500_{sq}$). This broader allocation of importance is economically intuitive and it confirms the interpretation that more flexible models, such as tree-based models and NNs, are able to exploit nonlinear dependencies from large information sets. This is also supported by the findings in Yang et al. (2017), Luong and Dokuchaev (2018), Miura et al. (2019), Gupta and Pierdzioch (2022), and Bucci (2020).

Several of the macroeconomic and financial variables featured at the top of the importance ranking also carry strong support in the broader volatility forecasting literature. The importance assigned to weekly and monthly momentum supports the argument of Díaz et al. (2024), who show that momentum-based predictors contain incremental information that improves forecasting performance particularly for ML-based models. Similarly, the importance of the S&P 500 variance proxy and the CISS index is consistent with the literature on how global risk sentiment and cross-country volatility spillover effects influence stock market volatility, including Audrino et al. (2020) and Choudhry et al. (2016). Furthermore, identifying the term spread as a key driver of forecast accuracy aligns, in broad terms, with the findings of Christiansen et al. (2012) and Fernandes et al. (2014), whereas the lack of significance of the EURIBOR-OIS credit spread ($CREDIT$) stands in contrast to the findings of Mittnik et al. (2015) and Paye (2012). Finally, the trivial role of the Brent oil price across all four models contrasts Wang et al. (2018), who show that crude oil carries information useful for forecasting volatility over shorter horizons.

To answer our third research question, the variable importance analysis highlights two main findings. First, lagged realized variance measures and expectations about im-

plied volatility from the VSTOXX index remain the dominant drivers of forecasting performance across all model classes, indicating that the realized variance components are highly persistent even in richer ML environments. Second, the extent to which models exploit information from macroeconomic and financial predictors differs across model classes. While the linear HAR-X and LA specification concentrates almost all predictive power to a small subset of volatility-related variables, the nonparametric RF and NN models extract relatively more information from the exogenous variables.

# 7 Concluding Remarks

This paper examines the out-of-sample one-day-ahead forecasting performance of a comprehensive set of model specifications in predicting realized variance on the EURO STOXX 50 index from January 1999 to August 2020. We compare a broad suite of HAR-type benchmark models against regularized linear methods, tree-based ensembles, and feed-forward neural networks, under two distinct feature sets. The first feature set is restricted to lagged realized variance components, while the second augments this feature set with a broader array of macroeconomic and financial predictors motivated by our European setting. Forecast accuracy is evaluated using both the MSE and QLIKE loss functions, with statistical significance assessed through the Diebold and Mariano (1995) test.

Our empirical analysis yield three important findings. First, the inclusion of macroeconomic and financial predictors materially improves forecasting performance of several ML specifications, indicating that these models are able to extract predictive signals beyond what is contained in the lagged realized variance measures alone. In particular, we find that the expanded feature set increases the number of specifications that significantly outperform HAR from two to nine models. However, these gains differ substantially across model classes and loss functions. While the tree-based models improve forecast accuracy under both the MSE and QLIKE loss functions, the gains for the regularized models and neural networks are only documented under MSE. This asymmetry can be attributed to the regime-dependent informativeness of the European macroeconomic predictors, which move sharply during episodes of heightened volatility but remain relatively stable during otherwise. This suggests that the regularized and NN specifications primarily improve fit during periods of elevated market stress, whereas the tree-based ensembles generate more stable forecast accuracy across volatility regimes.

Second, we find that two model specifications emerge as particularly robust for volatility forecasting on the EURO STOXX 50. Under the restricted $\mathcal{M}_{\text{HAR}}$ feature set, the HARQ model consistently delivers the best forecasting performance across both loss functions, while BG and RF stands out as the only specifications that statistically outperforms HARQ under the QLIKE loss function in the richer $\mathcal{M}_{\text{ALL}}$ feature set. Given the improvement of RF on both loss functions under the $\mathcal{M}_{\text{ALL}}$ feature set, it emerges as a robust alternative to HARQ when additional information is provided.

Third, the variable importance analysis reveals broad agreement across model classes regarding the dominant sources of predictive information. The lagged realized variance components and the VSTOXX index consistently emerge as the most important predictors across all specifications, highlighting the central role of volatility persistence and forward-looking market uncertainty as main drivers of future stock market volatility. At the same time, the tree-based and NN specifications extract relatively more information from the broader macro-financial predictor set compared to their linear counterparts, including momentum measures, global spillover proxies, and systemic risk factors. This pattern further supports the notion that more flexible model classes, which do not rely on a prespecified functional form, are better at capturing nonlinearities and interactions within large information sets, even if the predictors are correlated.

Our findings contribute to the comparatively limited strand of literature on volatility forecasting in European equity markets and provide evidence that richer information sets augmented by macroeconomic and financial variables can improve model forecasting performance, especially when combined with nonlinear ML specifications. The results also suggest that the choice of loss function materially affects the interpretation of model performance, particularly during periods of elevated market uncertainty. Finally, we offer practical guidance for selecting forecasting models in a single-index setting, characterized by pronounced volatility regimes.

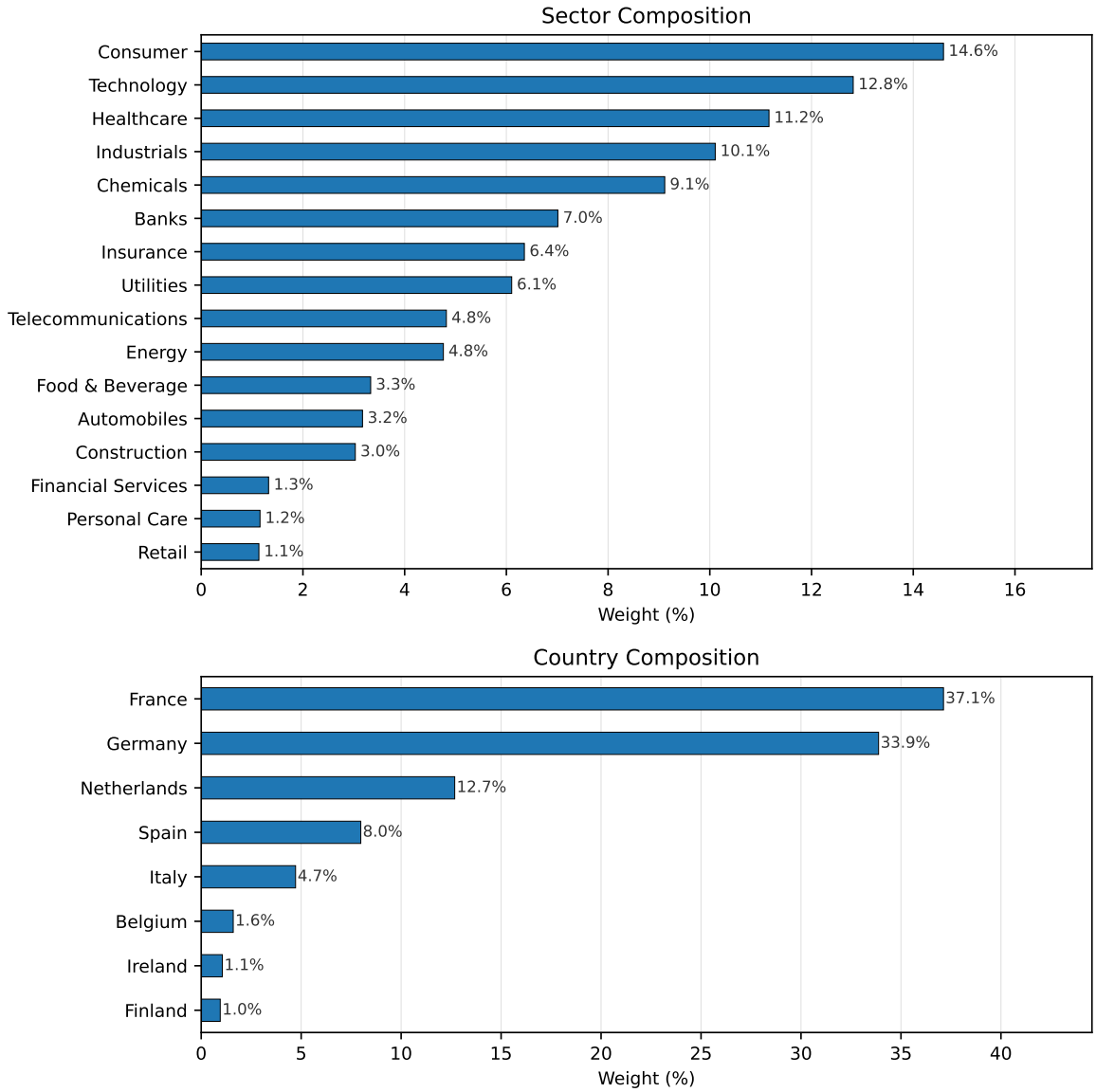
We recognize that our study is subject to limitations. First, our analysis is based on a single-index setting as opposed to the individual constituents of the EURO STOXX 50, which may limit the generalizability of our results. While a cross-sectional analysis could potentially yield more robust conclusions, such an extension would not be feasible given the time-frame and computational costs involved. Second, given that the analysis also focuses exclusively on one-day-ahead forecasts of realized variance, the findings may not extend directly to alternative markets or longer forecast horizons. To compensate for this, we instead focus our analysis on other avenues, such as performance under different loss functions and regime-dependency, which we regard as more meaningful contributions. Finally, due to computational costs, the weights of the feed-forward NN architectures are only set once, whereas the other models are re-estimated over time. Although this approach is consistent with related literature, it remains a limitation of the study.

Avenues for future research include cross-sectional analysis of the individual constituents, longer forecasting horizons, incorporating intraday jumps and overnight returns, exploring ensemble and regime-shifting modeling techniques, and extending the analysis to additional European or international equity markets. Future studies could also investigate more advanced neural network architectures, such as recurrent neural networks, as well as the application of these models within a Value-at-Risk (VaR) framework.

# 8 Appendix

## 8.1 EURO STOXX 50 Composition

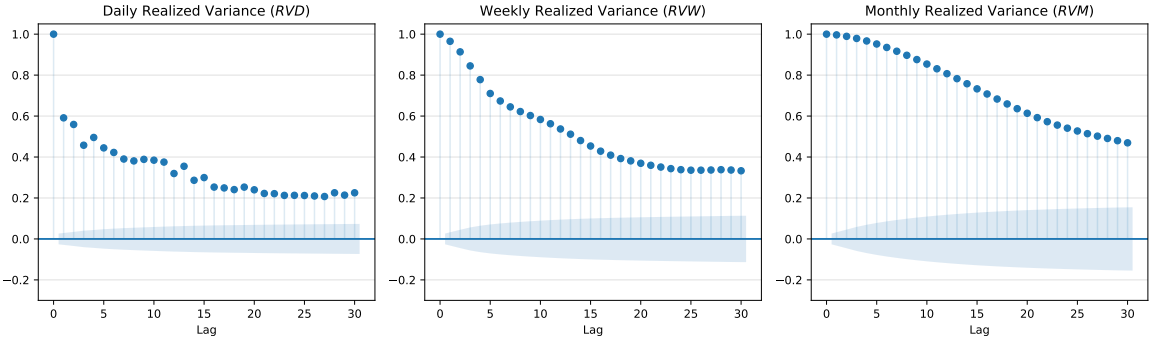
**Figure 9:** EURO STOXX 50 Composition by Sector and Country (2020)




*Notes:* The figure plots the sector and country composition of the EURO STOXX 50 index as of June 30, 2020. The top panel shows the distribution across industry sectors, while the bottom panel presents the distribution across countries. Please note that this is only a snapshot in time, as composition data is only available on a quarterly basis from 2019 onwards. As such, the weights are not representative of the entire sample period, but provide a general overview of the index, with weights expected to remain relatively stable. Index constituents, sector classifications, and weights are sourced from the STOXX website.

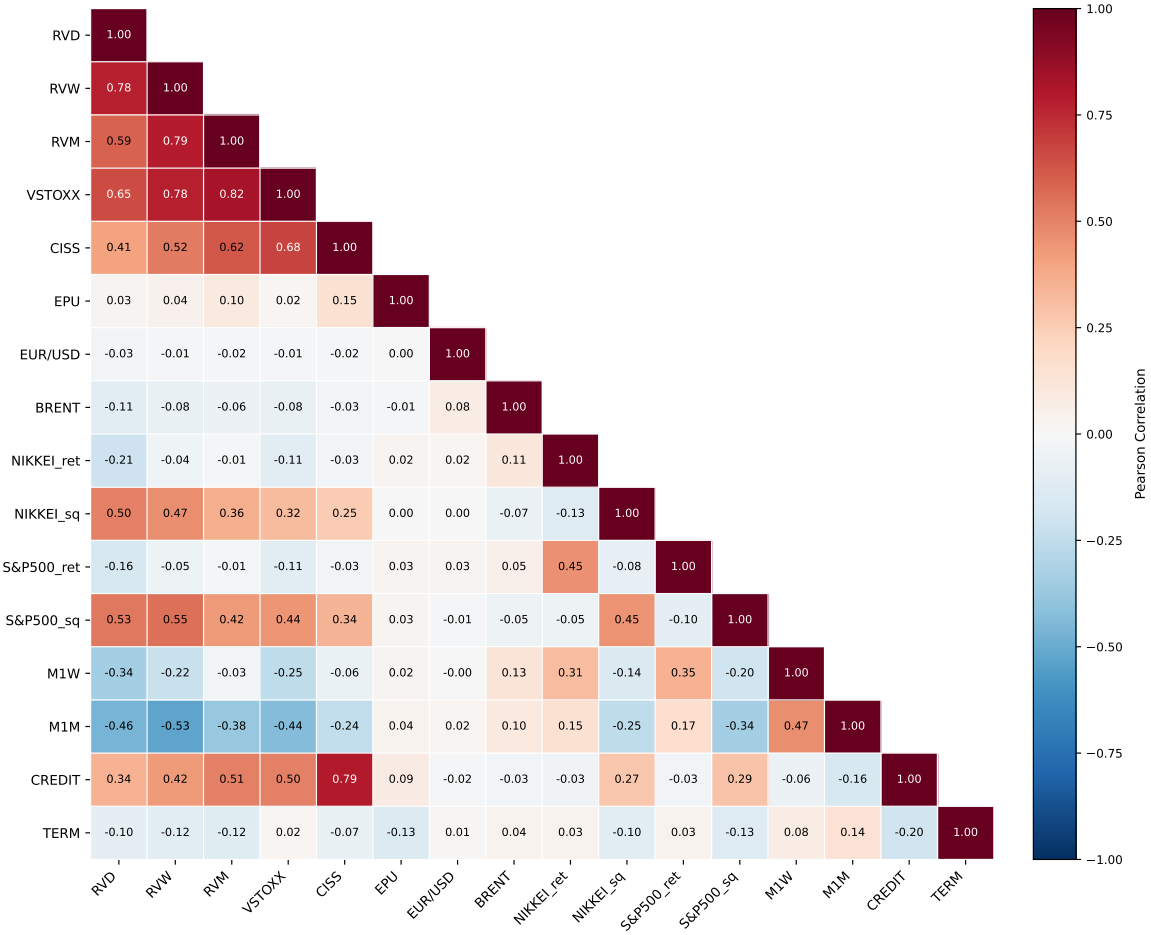
## 8.2 Realized Variance Dynamics and Predictor Correlations

**Figure 10:** Autocorrelation of Realized Variance Components




*Notes:* This figure plots the sample autocorrelation functions (ACFs) of the daily ($RVD$), weekly ($RVW$), and monthly ($RVM$) realized variance components over 30 lags. The shaded areas represent the 95% confidence intervals under the null hypothesis of zero autocorrelation.

**Figure 11:** Correlation Analysis of Predictors in $\mathcal{M}_{\text{ALL}}$ Feature Set




*Notes:* This figure plots the pairwise Pearson correlation matrix for all macroeconomic and financial variables included in the $\mathcal{M}_{\text{ALL}}$ feature set. Coefficients closer to one are shaded in red and indicate strong positive correlations, whereas values close to zero or below one are shaded in blue and reflect weak or negative linear relationships.

## 8.3 Description of Volatile Events

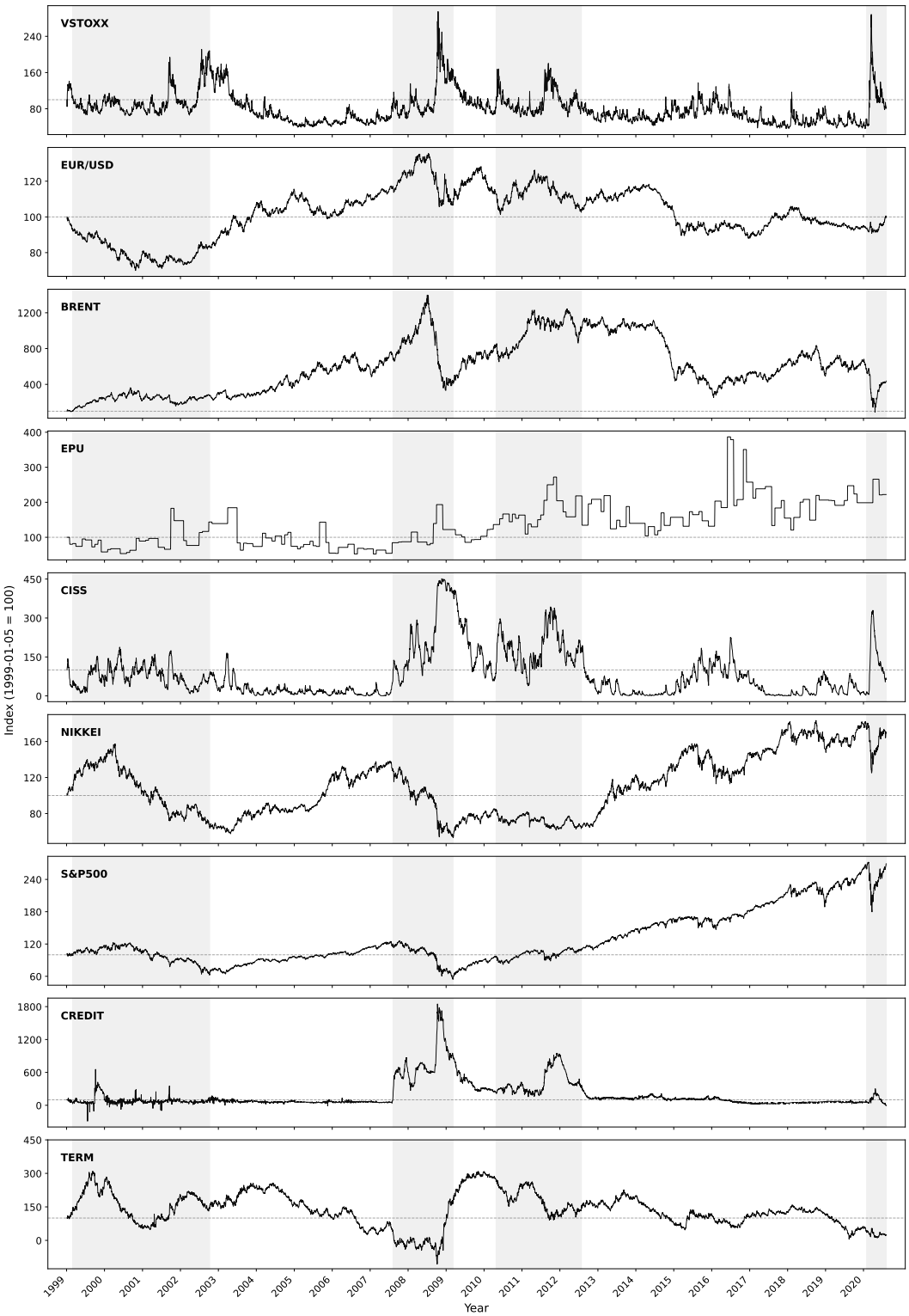
**Table 7:** Description of Annual Volatile Event Outliers on the EURO STOXX 50

| Year | Date | Description of Volatile Event |
|---|---|---|
| 1999 | Jan. 15 | Brazil floated the real, abandoning its dollar peg and triggering concerns. |
| 2000 | Apr. 17 | Dot-com bubble burst and Nasdaq closed its worst week on record. |
| 2001 | Sep. 11 | Al-Qaeda attacks on the World Trade Center and the Pentagon. |
| 2002 | Jul. 25 | Poor market sentiment amid widespread U.S. accounting scandals. |
| 2003 | Mar. 17 | US-led coalition issued its 48-hour ultimatum ahead of the Iraq invasion. |
| 2004 | Mar. 11 | Coordinated Al-Qaeda bombings on Madrid's commuter rail network. |
| 2005 | Jul. 07 | Four coordinated suicide bombings struck London's transport network. |
| 2006 | May 18 | Sharp correction driven by U.S. inflation and Fed rate hike expectations. |
| 2007 | Aug. 17 | Fed announced an emergency 50bp discount rate cut as credit markets froze. |
| 2008 | Oct. 10 | Post-Lehman crisis peak; equities' worst weekly performance since 1929. |
| 2009 | Jan. 23 | UK confirmed its first recession since 1991, deepening recession fears. |
| 2010 | May 07 | US stock market collapsed on May 6th causing a flash crash. |
| 2011 | Aug. 11 | Belgium, France, Italy and Spain imposed short sale bans on stocks. |
| 2012 | Aug. 02 | Aftermath of Draghi's *"Whatever it takes"* pledge on July 26th. |
| 2013 | Feb. 25 | Inconclusive Italian elections produced a hung parliament. |
| 2014 | Oct. 16 | IMF growth downgrades, falling oil prices, and Ebola fears triggered a sell-off. |
| 2015 | Aug. 24 | China's Shanghai Composite plunged over 8% on *"Black Monday"*. |
| 2016 | Jun. 24 | Announcement of Brexit referendum results triggered a record one-day drop. |
| 2017 | Apr. 24 | First-round French election eased Le Pen fears, sparking a relief rally. |
| 2018 | Feb. 06 | The *"Volmageddon"*, the VSTOXX index surged by over 62%. |
| 2019 | Aug. 15 | US-China trade escalation and the yuan's break past 7 per dollar. |
| 2020 | Mar. 16 | *"Black Thursday"* aftermath as Covid-19 lockdowns spread across Europe. |

*Notes:* This table provides an event description for each of the volatile event outliers presented in the box plot in Figure (3) for the entire sample period from January 5, 1999 to August 6, 2020. Note that extremely volatile trading days are often caused by several interacting factors, global events, and market sentiment that has accumulated over time.

## 8.4 Indexed Evolution of the Macroeconomic Predictors

**Figure 12:** Macroeconomic Variables Panel Included in $\mathcal{M}_{\text{ALL}}$ Feature Set




*Notes:* This figure presents the indexed values macroeconomic variables included in the $\mathcal{M}_{\text{ALL}}$ feature set. The index for each variable is set at 100 at the start of the sample period, on January 5, 1999. The sample period runs from January 5, 1999 to the August 6, 2020. Shaded regions indicate significant and persistent volatility regimes, previously described in Figure (2).

## 8.5 Hyperparameter Tuning

**Table 8:** Tuning of Hyperparameters

| **Panel A: Regularized Linear Models (RR, LA, EN)** | | | |
|---|---|---|---|
| | **RR** | **LA** | **EN** |
| Penalty term ($\lambda$) | $[10^{-5}, 10^{5}]^{*}$ | $[10^{-10}, 10^{1}]^{*}$ | $[10^{-5}, 10^{2}]^{*}$ |
| Mixing parameter ($\alpha$) | – | – | $\{0.01, \ldots, 1.00\}^{*}$ |
| **Panel B: Tree-Based Models (BG, RF, GB)** | | | |
| | **BG** | **RF** | **GB** |
| Minimum node size | 5 | 5 | – |
| Number of trees | 500 | 500 | $\{50, 100, \ldots, 500\}$ |
| Feature split ($p$) | – | $p/3$ | – |
| Tree depth | – | – | $\{1, 2\}$ |
| Learning rate | – | – | $\{0.01, 0.1\}$ |
| **Panel C: Neural Networks (NN)** | | | |
| Learning rate | 0.001 | | |
| Epochs | 500 | | |
| Patience | 100 | | |
| Drop-out | 0.8 | | |
| Initializer | Glorot normal | | |
| Optimizer | Adam (default) | | |
| Ensemble size | $\{1, 10\}$ | | |

*Notes:* This table reports the hyperparameter choices and tuning grids used across model classes. Parameters marked with an asterisk are selected via validation-set tuning. RR, LA, and EN denote ridge, lasso, and elastic net regression, respectively. BG, RF, and GB denote bagging, random forest, and gradient boosting models. $p$ denotes the number of predictors in the corresponding feature set. The full mixing-parameter grid for EN is $\{0.01, 0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.70, 0.90, 1.00\}$.

## 8.6 Description of Exogenous Variables

**Table 9:** Description of Predictors in the $\mathcal{M}_{\text{HAR}}$ and $\mathcal{M}_{\text{ALL}}$ Feature Sets

| Variable | Abbreviation | Description |
|---|---|---|
| **A. HAR Model** | | |
| Daily Realized Variance | $RVD$ | Previous-day realized variance |
| Weekly Realized Variance | $RVW$ | Trailing 5-day average of realized variance |
| Monthly Realized Variance | $RVM$ | Trailing 22-day average of realized variance |
| **B. LevHAR Extensions** | | |
| Daily Negative Return | $RD^{-}$ | Negative daily return component |
| Weekly Negative Return | $RW^{-}$ | Trailing 5-day average of negative returns |
| Monthly Negative Return | $RM^{-}$ | Trailing 22-day average of negative returns |
| **C. SHAR Extensions** | | |
| Positive Semivariance | $RVD^{+}$ | Previous-day variation from positive returns |
| Negative Semivariance | $RVD^{-}$ | Previous-day variation from negative returns |
| **D. HARQ Extension** | | |
| Realized Quarticity | $RQ$ | Time-varying proxy for noise in RVD |
| **E. Macroeconomic Variables** | | |
| VSTOXX Index | $VSTOXX$ | Implied volatility (EURO STOXX 50) |
| Economic Policy Uncertainty | $EPU$ | Policy uncertainty index |
| EUR/USD Exchange Rate | $EUR/USD$ | Daily log return of the EUR/USD rate |
| Composite Indicator of Systematic Stress | $CISS$ | ECB financial stress indicator |
| Brent Crude Oil Price | $BRENT$ | Europe Brent crude oil spot price |
| Nikkei 225 Return | $NIKKEI_{ret}$ | Daily return of Japanese equity index |
| Nikkei 225 Squared Return | $NIKKEI_{sq}$ | Proxy for volatility of Japanese equity index |
| S&P 500 Return | $SP\&500_{ret}$ | Daily lagged return of US equity index |
| S&P 500 Squared Return | $SP\&500_{sq}$ | Proxy for volatility of US equity index |
| One-week Momentum | $M1W$ | 1-week log return of EURO STOXX 50 |
| One-month Momentum | $M1M$ | 1-month log return of EURO STOXX 50 |
| Term Spread | $TERM$ | 10-year Euro swap vs. 3-month Euribor spread |
| Credit Spread | $CREDIT$ | Corporate bond yield spread |

*Notes:* This table provides a description of the variables used in the $\mathcal{M}_{\text{HAR}}$ and $\mathcal{M}_{\text{ALL}}$ feature sets. Variables are grouped according to model extensions. Panel A corresponds to the standard HAR model. Panel B contains additional variables used in the LevHAR specification, such that the LevHAR model is constructed using Panels A and B. Panel C contains the semivariance components used in the SHAR model, which combines Panels A and C. Panel D includes realized quarticity, used in the HARQ model, which combines Panels A and D. The LogHAR specification only relies on the variables in Panel A, with a logarithmic transformation applied to each realized variance measure. Panel E contains the set of macroeconomic and financial variables included in the extended $\mathcal{M}_{\text{ALL}}$ feature set. These variables are incorporated in the HAR-X model, as well as in the regularized and machine learning methods, which are all estimated using the combined set of predictors from Panels A and E.

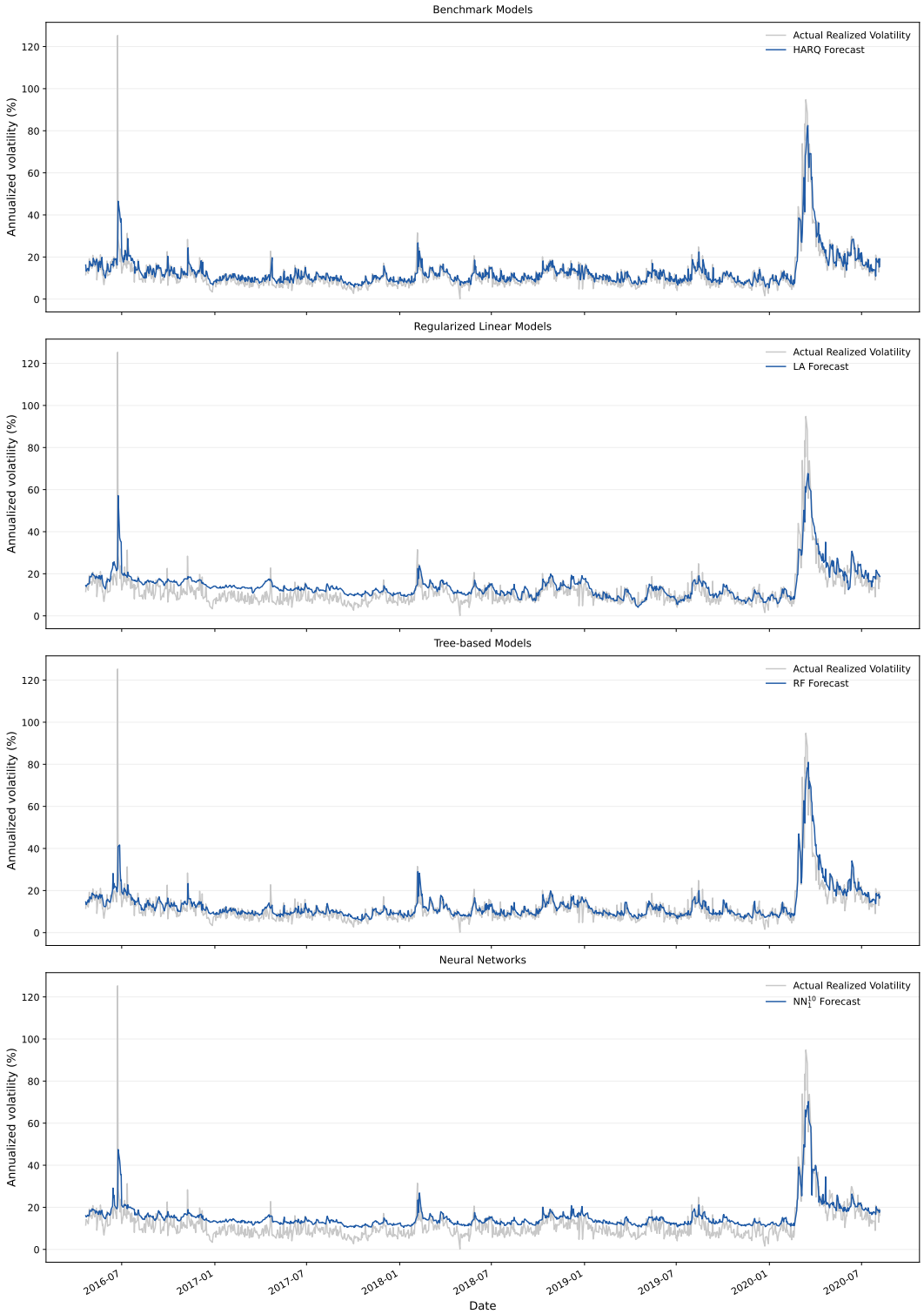
**Table 10:** Description of Predictors in the $\mathcal{M}_{\text{HAR}}$ and $\mathcal{M}_{\text{ALL}}$ Feature Sets

| Variable | Abbreviation | Transformation | Source |
|---|---|---|---|
| **A. MHAR Feature Set** | | | |
| Daily Realized Variance | $RVD$ | - | Own calculations |
| Weekly Realized Variance | $RVW$ | - | Own calculations |
| Monthly Realized Variance | $RVM$ | - | Own calculations |
| **B. LevHAR Extensions** | | | |
| Daily Negative Return | $RD^{-}$ | - | Own calculations |
| Weekly Negative Return | $RW^{-}$ | - | Own calculations |
| Monthly Negative Return | $RM^{-}$ | - | Own calculations |
| **C. SHAR Extensions** | | | |
| Positive Semivariance | $RVD^{+}$ | - | Own calculations |
| Negative Semivariance | $RVD^{-}$ | - | Own calculations |
| **D. HARQ Extension** | | | |
| Realized Quarticity | $RQ$ | - | Own calculations |
| **E. Macroeconomic Variables** | | | |
| VSTOXX Index | $VSTOXX$ | Levels | Eurex / Datastream |
| Economic Policy Uncertainty | $EPU$ | Logarithm | Baker et al. (2016) |
| EUR/USD Exchange Rate | $EUR/USD$ | Daily % Change | Datastream |
| Systemic Stress Index | $CISS$ | Levels | ECB |
| Brent Oil Price | $BRENT$ | Daily % Change | Datastream |
| Nikkei 225 Return | $NIKKEI_{ret}$ | Daily % Change | Datastream |
| Nikkei 225 Squared Return | $NIKKEI_{sq}$ | Squared Returns | Own calculations |
| S&P 500 Return (lag) | $SP\&500_{ret}$ | Daily % Change | Datastream |
| S&P 500 Squared Return (lag) | $SP\&500_{sq}$ | Squared Returns | Own calculations |
| One-week Momentum | $M1W$ | Log Return | Own calculations |
| One-month Momentum | $M1M$ | Log Return | Own calculations |
| Credit Spread | $CREDIT$ | First-difference | Datastream |
| Term Spread | $TERM$ | First-difference | Datastream |

*Notes:* This table provides a description of the variables used in the $\mathcal{M}_{\text{HAR}}$ and $\mathcal{M}_{\text{ALL}}$ feature sets. Variables are grouped according to model extensions. The transformation column reports the construction of each variable, including levels, logarithmic transformations, returns, squared returns, and first-differences. The label *Own calculations* refers to variables constructed from raw high-frequency intraday data obtained from the Stockholm School of Economics, which has been processed and subsequently transformed into each respective measures used for volatility forecasting.

## 8.7 Best Performing Model per Model Category on $\mathcal{M}_{\text{ALL}}$

**Figure 13:** Visual Representation of Model Forecast Performance




*Notes:* This panel provides a visual illustration of the best performing model per model category, in terms of MSE, on the $\mathcal{M}_{\text{ALL}}$ feature set. The time series plot depicts daily annualized volatility, spanning the out-of-sample test period in our study from 2016-04-21 to 2020-08-06. The grey line denotes the actual realized volatility while the blue line indicates the one-day-ahead out-of-sample forecast provided by each of the best performing models per model category. To annualize the numbers, we assume an average of 252 trading days per year. For additional information on model performance, see Tables (11) and (12) in Appendix (8.8).

## 8.8 Summary of Out-of-Sample Forecast Performance

**Table 11:** Loss Function Summary for $\mathcal{M}_{\text{HAR}}$ Feature Set

| | Loss Functions | | | |
|---|---|---|---|---|
| | **MSE** ($10^{-8}$) | **QLIKE** | **MAE** ($10^{-4}$) | **RMSE** ($10^{-4}$) |
| **HAR** | 5.66 | 0.24 | 0.53 | 2.38 |
| **HAR-X** | 5.66 | 0.24 | 0.53 | 2.38 |
| **LogHAR** | 5.30 | 0.20 | 0.46 | 2.30 |
| **LevHAR** | 5.39 | 0.42 | 0.55 | 2.32 |
| **SHAR** | 5.84 | 0.24 | 0.53 | 2.42 |
| **HARQ** | 4.79 | 0.19 | 0.44 | 2.19 |
| **RR** | 5.82 | 0.40 | 0.73 | 2.41 |
| **LA** | 5.89 | 0.42 | 0.77 | 2.43 |
| **EN** | 5.88 | 0.42 | 0.76 | 2.43 |
| **BG** | 5.33 | 0.19 | 0.48 | 2.31 |
| **RF** | 5.19 | 0.19 | 0.47 | 2.28 |
| **GB** | 5.51 | 0.34 | 0.63 | 2.35 |
| $\mathbf{NN}_1^1$ | 5.47 | 0.33 | 0.59 | 2.34 |
| $\mathbf{NN}_1^{10}$ | 5.71 | 0.29 | 0.57 | 2.39 |
| $\mathbf{NN}_2^1$ | 5.63 | 0.29 | 0.56 | 2.37 |
| $\mathbf{NN}_2^{10}$ | 5.65 | 0.33 | 0.60 | 2.38 |
| $\mathbf{NN}_3^1$ | 5.69 | 0.39 | 0.67 | 2.39 |
| $\mathbf{NN}_3^{10}$ | 5.68 | 0.35 | 0.62 | 2.38 |
| $\mathbf{NN}_4^1$ | 5.70 | 0.28 | 0.55 | 2.39 |
| $\mathbf{NN}_4^{10}$ | 5.92 | 0.36 | 0.64 | 2.43 |

*Notes:* This table reports the out-of-sample forecast performance of all 20 models estimated using the $\mathcal{M}_{\text{HAR}}$ feature set. Forecast accuracy is evaluated using Mean Squared Error (MSE), Quasi-Likelihood (QLIKE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). These loss functions are detailed further in Section (4.6). All measures are based on one-day-ahead forecasts of realized variance over the test period from 2016-04-21 to 2020-08-06 and rounded to two decimals, where lower values indicate better predictive performance for all four loss functions. The MSE and QLIKE loss functions serve as the primary sources of performance evaluation in Section (5), whereas MAE and RMSE provide supplementary information.

**Table 12:** Loss Function Summary for $\mathcal{M}_{\text{ALL}}$ Feature Set

| | Loss Functions | | | |
|---|---|---|---|---|
| | **MSE** ($10^{-8}$) | **QLIKE** | **MAE** ($10^{-4}$) | **RMSE** ($10^{-4}$) |
| **HAR** | 5.66 | 0.24 | 0.53 | 2.38 |
| **HAR-X** | 5.26 | $403.88^{(*)}$ | 0.54 | 2.29 |
| **LogHAR** | 5.30 | 0.20 | 0.46 | 2.30 |
| **LevHAR** | 5.39 | 0.42 | 0.55 | 2.32 |
| **SHAR** | 5.84 | 0.24 | 0.53 | 2.42 |
| **HARQ** | 4.79 | 0.19 | 0.44 | 2.19 |
| **RR** | 5.22 | $10.76^{(*)}$ | 0.63 | 2.29 |
| **LA** | 5.14 | 0.26 | 0.54 | 2.27 |
| **EN** | 5.14 | 0.26 | 0.54 | 2.27 |
| **BG** | 5.07 | 0.18 | 0.46 | 2.25 |
| **RF** | 4.82 | 0.18 | 0.44 | 2.20 |
| **GB** | 5.22 | 0.21 | 0.46 | 2.28 |
| $\mathbf{NN}_1^1$ | 5.35 | 0.26 | 0.53 | 2.31 |
| $\mathbf{NN}_1^{10}$ | 4.98 | 0.32 | 0.58 | 2.23 |
| $\mathbf{NN}_2^1$ | 4.99 | 0.32 | 0.58 | 2.23 |
| $\mathbf{NN}_2^{10}$ | 5.07 | 0.28 | 0.54 | 2.25 |
| $\mathbf{NN}_3^1$ | 5.23 | 0.27 | 0.55 | 2.29 |
| $\mathbf{NN}_3^{10}$ | 5.34 | 0.32 | 0.60 | 2.31 |
| $\mathbf{NN}_4^1$ | 5.44 | 0.37 | 0.65 | 2.33 |
| $\mathbf{NN}_4^{10}$ | 5.36 | 0.40 | 0.69 | 2.31 |

*Notes:* This table reports the out-of-sample forecast performance of all 20 models estimated using the $\mathcal{M}_{\text{ALL}}$ feature set. Forecast accuracy is evaluated using Mean Squared Error (MSE), Quasi-Likelihood (QLIKE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). These loss functions are detailed further in Section (4.6). All measures are based on one-day-ahead forecasts of realized variance over the test period from 2016-04-21 to 2020-08-06 and rounded to two decimals, where lower values indicate better predictive performance for all four loss functions. The MSE and QLIKE loss functions serve as the primary sources of performance evaluation in Section (5), whereas MAE and RMSE provide supplementary information. Values denoted with superscript $(*)$ are extreme outliers with respect to their QLIKE value. Both HAR-X and RR are both extreme outliers in this regard. We investigate this issue and confirm that the behavior stems from the fact that these models generate many negative forecasts during the test period, causing the negative forecasts to subsequently be replaced via the insanity filter and converted to the corresponding minimum realized variance value in the training set. The problem is that when such values are so small, such that they approach zero, and appear frequently in the forecast window, it causes the QLIKE score to explode, reflecting the loss function's strong asymmetric penalization of underestimations of next-day realized variance.

## 8.9 AI Disclosure

Throughout the process of writing of this paper, three main AI tools have been used across three main domains to improve the overall quality of our empirical research.

First and foremost, ChatGPT and Claude AI have been applied with the main purpose of improving the structure and functionality of the code used to produce the empirical results and findings of our paper. Our main insight from this process is that, although such AI tools have materially improved in terms of code generation, it is still integral to have a good understanding of the underlying programming language, in this case Python, in order to work efficiently with such tools. We observed that poor outputs from these AI tools stemmed primarily from poor and imprecise prompt engineering as opposed to AI hallucinations. Having prior experience of the underlying programming language allowed us to work efficiently with such tools by being able to easily spot why and in what way the applied AI tool may have misinterpreted the prompt. Moreover, to reduce the risk of wrongful conduct, complementary tools such as YouTube, research papers, books, and in some cases StackOverflow, have been used to verify concepts and coding implementations.

Second, AI tools have served as a complementary purpose of assisting with formatting of tables and illustrations in LaTeX. Again, poor prompt engineering and design was acknowledged as the main culprit of deficiencies in the output rather than pure AI hallucinations. Moreover, AI tools have been used in some instances to refine formulations and wording of the text to improve the overall flow and experience for the audience.

Finally, for research on the topic of volatility forecasting, Perplexity has been used as a hybrid search engine for finding relevant articles and sources. In some instances, while using ChatGPT, we noticed that the AI tool had a tendency to hallucinate and come up with completely false claims. Therefore, we have made sure to review all papers referenced in this paper in detail to ensure quality and consistency with established research ethics.

In summary, we recognize that generative AI has served as a supporting tool in terms of code generation and debugging as well as language refinement and formatting of visuals, thus contributing to the overall quality of this thesis. However, the analysis and integrity of this paper remains entirely our own contribution.

# References


Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. (2001). The distribution of realized exchange rate volatility. *Journal of the American statistical association*, 96(453):42–55.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica*, 71(2):579–625.

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

Audrino, F., Sigrist, F., and Ballinari, D. (2020). The impact of sentiment and attention measures on stock market volatility. *International Journal of Forecasting*, 36(2):334–357.

Baillie, R. T., Bollerslev, T., and Mikkelsen, H. O. (1996). Fractionally integrated generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 74(1):3–30.

Baker, S. R., Bloom, N., and Davis, S. J. (2016). Measuring economic policy uncertainty. *The quarterly journal of economics*, 131(4):1593–1636.

Bohl, M. T., Siklos, P. L., and Sondermann, D. (2008). European stock markets and the ecb's monetary policy surprises. *International Finance*, 11(2):117–130.

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

Bollerslev, T. and Mikkelsen, H. O. (1996). Modeling and pricing long memory in stock market volatility. *Journal of econometrics*, 73(1):151–184.

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

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

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

Choudhry, T., Papadimitriou, F. I., and Shabi, S. (2016). Stock market volatility and business cycle: Evidence from linear and nonlinear causality tests. *Journal of Banking & Finance*, 66:89–101.

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

Christiansen, C., Schmeling, M., and Schrimpf, A. (2012). A comprehensive look at financial volatility prediction by economic variables. *Journal of Applied Econometrics*, 27(6):956–977.

Cipollini, F. and Gallo, G. M. (2019). Modeling euro stoxx 50 volatility with common and market-specific components. *Econometrics and Statistics*, 11:22–42.

Corsetti, G., Kuester, K., Meier, A., and Müller, G. J. (2013). Sovereign risk, fiscal policy, and macroeconomic stability. *The Economic Journal*, 123(566):F99–F132.

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

Corsi, F., Audrino, F., and Renó, R. (2012). Har modeling for realized volatility forecasting. *Handbook of volatility models and their applications*, pages 363–382.

Corsi, F. and Renò, R. (2012). Discrete-time volatility forecasting with persistent leverage effect and the link with continuous-time volatility modeling. *Journal of Business & Economic Statistics*, 30(3):368–380.

Díaz, J. D., Hansen, E., and Cabrera, G. (2024). Machine-learning stock market volatility: Predictability, drivers, and economic value. *International Review of Financial Analysis*, 94:103286.

Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business and Economic Statistics*, 13(3):253–263.

Ding, Y., Kambouroudis, D., and McMillan, D. G. (2021). Forecasting realised volatility: Does the lasso approach outperform har? *Journal of International Financial Markets, Institutions and Money*, 74:101386.

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

Engle, R. F., Ghysels, E., and Sohn, B. (2013). Stock market volatility and macroeconomic fundamentals. *Review of Economics and Statistics*, 95(3):776–797.

Engle, R. F. and Ng, V. K. (1993). Measuring and testing the impact of news on volatility. *The journal of finance*, 48(5):1749–1778.

Fernandes, M., Medeiros, M. C., and Scharth, M. (2014). Modeling and predicting the cboe market volatility index. *Journal of Banking & Finance*, 40:1–10.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Glosten, L. R., Jagannathan, R., and Runkle, D. E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *The journal of finance*, 48(5):1779–1801.

Golosnoy, V., Gribisch, B., and Liesenfeld, R. (2015). Intra-daily volatility spillovers in international stock markets. *Journal of International Money and Finance*, 53:95–114.

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

Gunnarsson, E. S., Isern, H. R., Kaloudis, A., Risstad, M., Vigdel, B., and Westgaard, S. (2024). Prediction of realized volatility and implied volatility indices using ai and machine learning: A review. *International review of financial analysis*, 93:103221.

Gupta, R. and Pierdzioch, C. (2022). Forecasting the realized variance of oil-price returns: a disaggregated analysis of the role of uncertainty and geopolitical risk. *Environmental Science and Pollution Research*, 29(34):52070–52082.

Harrison, B. and Moore, W. (2012). Forecasting stock market volatility in central and eastern european countries. *Journal of forecasting*, 31(6):490–503.

Hastie, T. (2009). The elements of statistical learning: data mining, inference, and prediction.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hollo, D., Kremer, M., and Lo Duca, M. (2012). Ciss-a composite indicator of systemic stress in the financial system.

Katsampoxakis, I., Christopoulos, A., Kalantonis, P., and Nastas, V. (2022). Crude oil price shocks and european stock markets during the covid-19 period. *Energies*, 15(11):4090.

Lane, P. R. (2012). The european sovereign debt crisis. *Journal of economic perspectives*, 26(3):49–68.

Leushuis, R. M. and Petkov, N. (2026). Advances in forecasting realized volatility: a review of methodologies. *Financial Innovation*, 12(1):14.

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

Maas, A. L., Hannun, A. Y., Ng, A. Y., et al. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA.

Mansilla-Lopez, J., Mauricio, D., and Narváez, A. (2025). Factors, forecasts, and simulations of volatility in the stock market using machine learning. *Journal of Risk and Financial Management*, 18(5):227.

Masters, T. (1993). *Practical neural network recipes in C++*. Morgan Kaufmann.

Mittnik, S., Robinzonov, N., and Spindler, M. (2015). Stock market volatility: Identifying major drivers and the nature of their impact. *Journal of banking & Finance*, 58:1–14.

Miura, R., Pichl, L., and Kaizoji, T. (2019). Artificial neural networks for realized volatility prediction in cryptocurrency time series. In *International Symposium on Neural Networks*, pages 165–172. Springer.

Müller, U. A., Dacorogna, M. M., Davé, R. D., Olsen, R. B., Pictet, O. V., and Von Weizsäcker, J. E. (1997). Volatilities of different time resolutions—analyzing the dynamics of market components. *Journal of Empirical Finance*, 4(2-3):213–239.

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica: Journal of the econometric society*, pages 347–370.

Nonejad, N. (2017). Forecasting aggregate stock market volatility using financial and macroeconomic predictors: Which models forecast best, when and why? *Journal of Empirical Finance*, 42:131–154.

Parra-Domínguez, J. and Sanz-Martín, L. (2024). Artificial intelligence in the new era of decision-making: A case study of the euro stoxx 50. *Mathematics*, 12(24):3918.

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

Patton, A. J. and Sheppard, K. (2009). Optimal combinations of realised volatility estimators. *International Journal of Forecasting*, 25(2):218–238.

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

Paye, B. S. (2012). 'déja vol': Predictive regressions for aggregate stock market volatility using macroeconomic variables. *Journal of Financial Economics*, 106(3):527–546.

STOXX Ltd. (2025). When do returns come from? an analysis of the overnight effect in equities trading. Accessed: 2026-03-30.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives*, 28(2):3–28.

Vortelinos, D. I. (2017). Forecasting realized volatility: Har against principal components combining, neural networks and garch. *Research in international business and finance*, 39:824–839.

Wang, J., Han, X., Huang, E. J., and Yost-Bremm, C. (2020). Predictability in international stock returns using currency fluctuations and forward rate forecasts. *The North American Journal of Economics and Finance*, 52:101108.

Wang, Y., Wei, Y., Wu, C., and Yin, L. (2018). Oil and the short-term predictability of stock return volatility. *Journal of Empirical Finance*, 47:90–104.

Wei, Y., Liang, C., Li, Y., Zhang, X., and Wei, G. (2020). Can cboe gold and silver implied volatility help to forecast gold futures volatility in china? evidence based on har and ridge regression models. *Finance Research Letters*, 35:101287.

Yang, K., Tian, F., Chen, L., and Li, S. (2017). Realized volatility forecast of agricultural futures using the har models with bagging and combination approaches. *International Review of Economics & Finance*, 49:276–291.

Zakoian, J.-M. (1994). Threshold heteroskedastic models. *Journal of Economic Dynamics and control*, 18(5):931–955.

Zhang, Y., Wei, Y., Zhang, Y., and Jin, D. (2019). Forecasting oil price volatility: Forecast combination versus shrinkage method. *Energy Economics*, 80:423–433.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320.