

HEC MONTRÉAL

**Forecasting Daily Realized Volatility Using Intraday
Microstructure Indicators and Machine Learning Models**

by

Katayoun Ghollasimoud

Under the supervision of

Tolga Cenesizoglu

Department of Decision Sciences

(Financial Engineering)

In Partial fulfillment of the requirements for
the Degree of Master of Science (M.Sc.)

December 2025

Résumé

Cette thèse examine la prévision hors échantillon de la volatilité réalisée à l'aide d'indicateurs de microstructure de marché à haute fréquence. En utilisant la volatilité réalisée quotidienne et les indicateurs intrajournaliers basés sur les données TAQ à la milliseconde de WRDS (agrégats quotidiens), l'analyse évalue des prévisions à un jour pour 30 actions du Dow Jones Industrial Average (DJIA) dans un cadre de prévision glissante en temps réel.

Les prévisions sont produites à l'aide de différentes classes de modèles, allant des régressions linéaires et régularisées aux ensembles d'arbres et aux réseaux de neurones à propagation avant. Les modèles sont comparés dans un cadre commun avec des ensembles de prédicteurs et des horizons alignés. Les performances sont examinées sous différentes longueurs de fenêtres d'estimation glissantes afin d'évaluer la sensibilité à l'historique disponible, et les choix de réglage sont mis en œuvre de manière à préserver une évaluation hors échantillon.

Les résultats indiquent que la précision prédictive dépend de l'interaction entre la flexibilité du modèle et la longueur de la fenêtre d'estimation. La régularisation améliore les performances par rapport aux modèles linéaires non pénalisés de référence, tandis que les ensembles d'arbres, en particulier les forêts aléatoires (Random Forest), offrent les gains les plus fiables parmi les classes de modèles et les spécifications de référence évaluées dans cette thèse lorsque l'historique d'apprentissage est suffisant. Des fenêtres d'estimation courtes réduisent la précision pour l'ensemble des approches. Les résultats indiquent en outre que les conditions de liquidité et la composition des flux de transactions constituent les principaux moteurs de la prévisibilité à court horizon, tandis que la persistance de la volatilité joue un rôle dominant aux horizons plus longs, l'importance des variables variant selon les classes de modèles.

Abstract

This thesis investigates out-of-sample forecasting of realized volatility using high-frequency market microstructure indicators. Using daily realized volatility and WRDS millisecond TAQ-based Intraday Indicators (daily aggregates), the analysis evaluates one-day-ahead forecasts for 30 Dow Jones Industrial Average (DJIA) stocks in a real-time rolling framework.

Forecasts are produced with a range of model classes, spanning linear and regularized regressions, tree-based ensembles, and feed-forward neural networks. Models are compared under a common design with aligned predictor sets and horizons. Performance is examined under alternative rolling estimation-window lengths to assess sensitivity to available history, and tuning choices are implemented to preserve an out-of-sample evaluation.

The findings indicate that predictive accuracy depends on the interaction between model flexibility and estimation-window length. Regularization improves performance relative to unpenalized linear benchmarks, while tree-based ensembles, especially Random Forest, deliver the most reliable gains among the model classes and baseline specifications evaluated in this thesis when sufficient training history is available. Short estimation windows reduce accuracy across approaches. The results further indicate that liquidity conditions and trading-flow composition are the primary drivers of short-horizon predictability, whereas volatility persistence plays a dominant role at longer horizons, with feature importance varying across model classes.

Acknowledgement

At the conclusion of this master's thesis, I would like to express my sincere appreciation and gratitude to all those who contributed to the completion of this work. I am especially grateful to my research director, Professor Tolga Cenesizoglu, whose invaluable guidance, expertise, and support have profoundly shaped this research. I am truly thankful for the opportunity to work under his mentorship and for his continued support throughout my studies in the Financial Engineering program.

I am deeply thankful to my husband and parents for their unwavering support, encouragement, and sacrifices throughout my academic journey.

I acknowledge the use of generative artificial intelligence tools as assistive resources during the research and writing of this thesis. These tools supported idea clarification, literature review efficiency, formatting in accordance with HEC Montréal's academic requirements, and improvements to the readability and structure of Python code used for data preprocessing and modeling, as well as language clarity during drafting. All AI-generated suggestions were critically reviewed and revised to ensure consistency with my own understanding, analytical reasoning, and academic voice. No confidential, personal, or unpublished data were shared, and the final content of this thesis represents my original work.

Table of Contents

Table of Contents.....	5
Section 1: Introduction.....	8
Section 2: Literature Review.....	11
2.1 Volatility Forecasting and Realized Volatility.....	11
2.2 High-Frequency Microstructure Indicators and Volatility Predictability	12
2.3 Linear Models, Regularization, and Forecasting with High-Dimensional Predictors	12
2.4 Machine Learning Approaches to Volatility Forecasting	13
2.4.1 Tree-Based Methods	13
2.4.2 Neural Networks	14
2.5 Real-Time Forecasting, Rolling Windows, and Evaluation Practices	14
2.6 Interpretability and Feature Importance in Predictive Modeling.....	15
2.7 Gaps in Literature and Contribution of This Study.....	15
Section 3: Methodology	16
3.0 Research Questions and Hypotheses.....	17
3.1. Data & Variables.....	18
3.1.1 Data Description	18
3.1.2 Variable Selection.....	18
3.1.3 Log Transformation of the Target Variable	19
3.2. Data Preprocessing.....	20
3.2.1 Data Cleaning.....	20
3.2.2 Handling Multicollinearity.....	21
3.2.3 Winsorization and Standardization	22
3.2.4 Lagged Volatility Variables	23
3.3. Forecasting Framework	24
3.3.1 Rolling-Window Design	24
3.3.2 Hyperparameter Tuning Strategies	25
3.3.3 Evaluation Metrics	27
3.4. Model Specifications	28
3.4.1 Linear and Penalized Linear Models.....	29
3.4.2 Tree-Based Models	30
3.4.3 Neural Network Models (NN1–NN5).....	31
3.5. Benchmark For Comparison	33
3.5.1 Simple Moving Average (SMA).....	34
3.5.2 Autoregressive Model AR(1).....	34

3.5.3 GARCH(1,1) Model.....	35
3.6. Feature Importance	35
3.6.1 Feature Categorization	36
3.6.2 Model-Specific Importance Definitions.....	37
Section 4: Results.....	39
4.1 Forecasting Performance	39
4.1.1 252-Day Rolling Window	39
4.1.2 21-Day Rolling Window	43
4.1.3 Out-of-Sample Performance in Earlier versus Later Sample Periods (Stability Analysis).....	46
4.2 Feature Importance Results.....	49
4.2.1 Feature Importance: 252-Day Window.....	49
4.2.2 Feature Importance: 21-Day Window.....	51
4.3 Results Relative to the Hypotheses.....	54
4.3.1 H1: Regularization Effect	54
4.3.2 H2: Nonlinearity	54
4.3.3 H3: Model Capacity	55
4.3.4 H4: Estimation-Window Length.....	55
Section 5: Limitations.....	56
Section 6: Discussion and Conclusion.....	59
6.1 Rolling-Window-Length-Dependent Predictability.....	59
6.2 Model Class Comparison.....	59
6.2.1 Tree-Based Models	59
6.2.2 Regularized Linear Models.....	60
6.3 Temporal Stability of Forecasting Performance	61
6.4 Economic Interpretation of Feature Importance	61
6.5 Implications and Concluding Remarks	61
References.....	62
Appendix A — Variable dictionary	67
Appendix B — Feature Importance Results	69
B.1 Feature Importance — 252-Day Rolling Window	69
B.2 Feature Importance — 21-Day Rolling Window	72
Appendix C — Full Results Tables	76

List of Figures

Figure 1: Category-Level Feature Importance for Random Forest (W=252).....	50
Figure 2: Category-Level Feature Importance for XGBoost (W=252)	51
Figure 3: Category-Level Feature Importance for Elastic Net (W=252).....	51
Figure 4: Category-Level Feature Importance for Random Forest (W=21).....	52
Figure 5: Category-Level Feature Importance for XGBoost (W=21)	53
Figure 6: Category-Level Feature Importance for Ridge Regression (W=21).....	53

List of Tables

Table 1: Feature Categories Used for Category-Level Importance Aggregation	36
Table 2: Out-of-Sample R^2 for One-Day-Ahead Volatility Forecasts (W=252)	40
Table 3: Out-of-Sample MSE for One-Day-Ahead Volatility Forecasts (W=252).....	40
Table 4: Out-of-Sample R^2 for One-Day-Ahead Volatility Forecasts (W=21)	43
Table 5: Out-of-Sample MSE for One-Day-Ahead Volatility Forecasts (W=21).....	44
Table 6: Early vs. Late Out-of-Sample Forecasting Performance (W=252)	47
Table 7: Early vs. Late Out-of-Sample Forecasting Performance (W=21)	48

Section 1: Introduction

Volatility forecasting plays a central role in modern financial economics and practice (Poon & Granger, 2003). Accurate volatility estimates underpin portfolio allocation (Markowitz, 1952), derivative pricing (Black & Scholes, 1973), risk management (Jorion, 2006), and regulatory capital requirements (Basel Committee on Banking Supervision, 2019). Short-horizon volatility forecasts are crucial for daily risk control and margining (Christoffersen, 1998; Basel Committee on Banking Supervision & International Organization of Securities Commissions, 2015; CME Group, 2019), and in high-frequency markets rapid changes in market conditions can have material financial consequences (O’Hara, 2015).

The increasing availability of high-frequency financial data has fundamentally changed the empirical study of volatility (Hautsch, 2012). Intraday transaction and quote data enable the construction of realized-volatility measures (Barndorff-Nielsen & Shephard, 2002; Andersen et al., 2003) that provide substantially more information than traditional daily return-based estimators (Andersen et al., 2001). This development has stimulated a large literature examining the predictability of realized volatility (Andersen et al., 2003; Corsi, 2009) using intraday market information, including trade intensity, order flow, and microstructure-based indicators (Hautsch, 2012). While these data offer richer signals, they also introduce high dimensionality, strong collinearity, and nonstationary behavior, which pose challenges for standard econometric models (Hastie, Tibshirani, & Friedman, 2009; Inoue, Jin, & Rossi, 2017).

In parallel, machine learning methods have gained prominence in financial forecasting (Gu, Kelly, & Xiu, 2020). Flexible models such as tree-based ensembles (Breiman, 2001; Friedman, 2001) and neural networks (Hastie et al., 2009) are designed to capture nonlinearities, interaction effects, and complex nonlinear dynamics without imposing restrictive parametric assumptions. These characteristics make them appealing candidates for modeling realized volatility, which is known to exhibit persistence and clustering (Engle, 1982; Bollerslev, 1986; Andersen et al., 2003; Corsi, 2009) and time-varying relationships with market activity (Andersen, 1996; Karpoff, 1987; Tauchen & Pitts, 1983).

Despite this promise, empirical evidence on the out-of-sample performance of machine-learning models in volatility forecasting remains mixed. While some studies report substantial gains relative to traditional models (Christensen, Siggaard, & Veliyev, 2023), others find limited or unstable improvements relative to strong econometric benchmarks (Kılıç, 2025;

Audrino & Chassot, 2024). Reported results are often sensitive to data preprocessing choices, estimation window selection, and hyperparameter re-tuning strategies, particularly in high-dimensional and low signal-to-noise environments (Gu et al., 2020; Audrino & Chassot, 2024). In particular, differences in evaluation design, such as expanding versus rolling windows, in-sample versus real-time tuning, and forecast-horizon alignment, complicate comparisons across studies and raise concerns about overfitting and data snooping in out-of-sample forecast evaluation (Rossi & Inoue, 2012).

A central empirical challenge is therefore to assess whether increased model flexibility delivers robust gains in forecasting accuracy under realistic real-time conditions. High-capacity models may perform well when trained on large samples but can become unstable when estimation windows are short or when parameters must be repeatedly re-estimated (Hastie et al., 2009; Inoue et al., 2017). Conversely, simpler or regularized models may offer greater robustness at the cost of flexibility. Understanding this trade-off is especially important in high-frequency volatility forecasting, where structural changes in market microstructure and liquidity regimes are common (Hautsch, 2012; O’Hara, 2015).

Existing studies differ widely in data construction, evaluation design, estimation-window choice, and hyperparameter-tuning strategies, which limits the direct comparability of reported forecasting results (Gu et al., 2020; Audrino & Chassot, 2024). Consequently, observed performance differences across model classes are often sensitive to empirical design choices, including estimation-window length and parameter-tuning procedures (Audrino & Chassot, 2024; Inoue et al., 2017). This heterogeneity motivates a controlled comparison in which models of differing complexity are evaluated under identical, strictly out-of-sample conditions.

This thesis addresses these issues by providing a disciplined, like-for-like comparison of forecasting models with varying degrees of complexity under a strictly out-of-sample rolling-window framework. Using daily realized measures and microstructure indicators derived from millisecond-level intraday Trades and Quotes (TAQ) data for 30 Dow Jones Industrial Average (DJIA) constituents, the analysis evaluates one-day-ahead realized-volatility forecasts produced by linear regressions, penalized linear models (Hoerl & Kennard, 1970; Tibshirani, 1996), tree-based ensemble methods (Breiman, 2001; Friedman, 2001), and feedforward neural networks. All models are estimated using an identical data pipeline, identical predictor sets, and identical rolling-window procedures, ensuring comparability across model classes.

A central feature of the empirical design is the explicit role of estimation-window length. Forecasts are generated using both a long rolling window of 252 trading days and a short rolling window of 21 trading days. This allows the analysis to examine how model performance depends on sample size, parameter stability, and potential sensitivity to regime changes. Hyperparameters are tuned once using in-sample data and held fixed throughout the out-of-sample forecasting exercise in order to prevent look-ahead bias and reflect operational forecasting constraints.

The contribution of this thesis is empirical rather than methodological. Rather than proposing new volatility estimators or machine-learning architectures, the study offers a transparent and realistic assessment of when model complexity is beneficial and when it becomes detrimental. By isolating model flexibility and window length as the primary dimensions of interest, the analysis clarifies the conditions under which nonlinear machine-learning models outperform simpler alternatives and the circumstances under which their advantages erode.

The empirical results show that forecasting accuracy depends strongly on both model complexity and the length of the rolling estimation window. Unregularized OLS performs poorly in this high-dimensional setting, while penalized linear models (Lasso, Ridge, and Elastic Net) deliver clear and stable improvements, especially under the longer 252-day window. Overall, tree-based ensemble methods, and in particular Random Forest, provide the strongest and most consistent out-of-sample performance among the models evaluated in this thesis when sufficient training history is available. Neural networks offer only limited gains over linear benchmarks in this dataset and rolling-window setting and do not reliably outperform tree-based models, with performance varying more across architectures and window lengths. Taken together, the results suggest that model flexibility improves realized-volatility forecasting only up to a point, and that the benefits of complex nonlinear methods depend critically on having enough data for stable estimation.

The remainder of the thesis is organized as follows. Section 2 reviews the literature on realized-volatility forecasting and the application of machine-learning methods in finance. Section 3 describes the data, model specifications, estimation procedures, and feature-importance diagnostics. Section 4 presents the out-of-sample forecasting results and evaluates them relative to the stated hypotheses. Section 5 addresses the limitations and constraints of the study. Section 6 concludes and discusses implications and directions for future research.

Section 2: Literature Review

2.1 Volatility Forecasting and Realized Volatility

Volatility plays a central role in asset pricing, risk management, derivative valuation, and portfolio allocation (Black & Scholes, 1973; Jorion, 2006; Markowitz, 1952; Poon & Granger, 2003). The academic literature has long documented that financial volatility is not purely random but exhibits persistent and predictable structure (Poon & Granger, 2003). Early contributions, such as Engle (1982) and Bollerslev (1986), introduced ARCH and GARCH models to formally capture conditional heteroskedasticity, highlighting the existence of volatility clustering in conditional variance. Subsequent extensions have demonstrated that volatility exhibits high persistence, with conditional variance responding asymmetrically to positive and negative return innovations (Engle & Ng, 1993).

A major development occurred with the advent of high-frequency data, which enabled the estimation of realized volatility (RV) from intraday returns. Barndorff-Nielsen and Shephard (2002) provide the foundational econometric framework linking realized measures to quadratic variation, while Andersen, Bollerslev, Diebold, and Labys (2003) show how realized measures can be modeled and forecast in practice using high-frequency information. Realized volatility has since become a benchmark target in empirical finance because it incorporates vast amounts of intraday information and provides a far more precise proxy for latent variance than daily squared returns and other coarse return-based estimators (Andersen, Bollerslev, Diebold, & Labys, 2003; Barndorff-Nielsen & Shephard, 2002). Numerous studies show that RV follows highly persistent dynamics that can be captured by both parametric models, such as the heterogeneous autoregressive (HAR) model of Corsi (2009), and by nonparametric or semi-parametric alternatives (Andersen et al., 2003; Corsi, 2009).

Despite the progress of econometric volatility models, realized volatility forecasting remains challenging due to extreme persistence, parameter instability, and the difficulty of separating true price discovery from microstructure noise (Aït-Sahalia, Mykland, & Zhang, 2005; Andersen et al., 2003; Inoue, Jin, & Rossi, 2017). In particular, microstructure noise creates a measurement trade-off: sampling more frequently increases the amount of raw data but can worsen the signal-to-noise ratio in realized measures if noise dominates (Aït-Sahalia et al., 2005). These difficulties motivate the exploration of more flexible modeling approaches capable of leveraging the rich information embedded in high-frequency data.

2.2 High-Frequency Microstructure Indicators and Volatility Predictability

Microstructure theory provides a framework for understanding how trading activity, order flow, liquidity, and information asymmetry shape price dynamics. Foundational work by Kyle (1985), Glosten and Milgrom (1985), and Hasbrouck (1991) shows that price variability is influenced not only by fundamental risk but also by frictions such as asymmetric information, price impact, and order-processing/adverse-selection components embedded in transaction prices and quotes (Glosten & Milgrom, 1985; Hasbrouck, 1991; Kyle, 1985).

Empirical studies confirm that intraday market characteristics are closely related to short-term volatility. Measures derived from trading intensity, bid–ask spreads, order imbalances, trade direction, and depth dynamics are strongly linked to contemporaneous and future volatility (Hautsch, 2012). For example, Hasbrouck and Saar (2013) study low-latency trading and relate it to market quality, including short-horizon volatility outcomes, rather than treating volatility as driven purely by end-of-day fundamentals. Other works highlight the role of liquidity provision and market fragmentation in shaping price variability in modern electronic equity markets (Menkveld, 2013; O’Hara & Ye, 2011).

High-frequency indicators provide a rich representation of market conditions within the trading day, capturing aspects such as: transitory and permanent price impacts, episodic liquidity shocks, clustering of information arrival, microstructure noise in trade and quote dynamics, and the interaction between order flow and volatility. Given this, microstructure features have become increasingly relevant for forecasting realized volatility (Hautsch, 2012). They offer a way to incorporate granular information without imposing parametric structure on how intraday variables influence daily volatility. However, their high dimensionality and strong multicollinearity pose methodological challenges for traditional econometric techniques (Hastie, Tibshirani, & Friedman, 2009).

2.3 Linear Models, Regularization, and Forecasting with High-Dimensional Predictors

Linear models are a cornerstone of financial econometrics due to their interpretability, analytical tractability, and well-understood statistical properties. Classic forecasting approaches, such as OLS regression, AR-type models, and the HAR specification, demonstrate that simple linear relationships often achieve competitive results when volatility is highly persistent (Corsi, 2009).

Yet, standard linear specifications face limitations when the predictor set is expanded to include large and highly correlated feature sets, as is typical with microstructure-derived data. High-dimensional regressions suffer from multicollinearity, unstable coefficients, and inflated variance, which degrade out-of-sample performance (Hastie et al., 2009). These issues motivated the introduction of regularization techniques.

Lasso (Tibshirani, 1996), Ridge (Hoerl & Kennard, 1970), and Elastic Net (Zou & Hastie, 2005) impose penalties that shrink or eliminate coefficients, improving generalization in the presence of noise or redundant predictors. In financial contexts, regularization has proven to be effective for forecasting returns, volatility, and macroeconomic series (Christensen, Siggaard, & Veliyev, 2023; De Mol, Giannone, & Reichlin, 2008; Gu, Kelly, & Xiu, 2020). While regularization offers important improvements over classical OLS, linear specifications remain limited in their ability to represent nonlinear relationships and interaction effects that may play a central role in volatility dynamics (Hastie et al., 2009).

2.4 Machine Learning Approaches to Volatility Forecasting

Machine learning (ML) methods have gained prominence in financial forecasting due to their flexibility and ability to approximate complex functions (Gu et al., 2020). The volatility literature increasingly incorporates ML tools to capture nonlinear dependencies, interactions among predictors, and time-varying predictive relationships (Christensen et al., 2023; Gu et al., 2020). A central question is whether this added flexibility improves genuinely out-of-sample forecasts once tuning choices and window design are disciplined, or whether gains reflect implicit overfitting to a particular evaluation protocol.

2.4.1 Tree-Based Methods

Decision-tree ensembles, particularly Random Forests (Breiman, 2001) and gradient boosting methods (Friedman, 2001) such as XGBoost (Chen & Guestrin, 2016), are widely used for forecasting tasks across economics and finance (Gu et al., 2020). Their strengths include: automatic handling of nonlinearities and interaction effects, robustness in high-dimensional settings, and strong empirical performance in noisy and potentially nonstationary environments (Breiman, 2001; Chen & Guestrin, 2016; Friedman, 2001; Hastie et al., 2009). Recent work in volatility forecasting finds that tree-based ensembles can be competitive with, and sometimes improve upon, strong linear benchmarks under realistic evaluation designs (Christensen et al., 2023). However, because trees effectively “search” over many nonlinear splits, their

performance can depend strongly on sample size and tuning choices (Breiman, 1996), making them a natural candidate for testing window-length sensitivity under rolling estimation.

2.4.2 Neural Networks

Neural networks provide an even more flexible functional form. Their universal approximation property makes them suitable for capturing nonlinear features in financial time series (Cybenko, 1989; Hornik, 1991). Neural networks have been used in forecasting realized volatility and related volatility targets (Christensen et al., 2023). The literature emphasizes both the potential and the challenges of neural networks:

- Potential: ability to capture nonlinearity, complex response surfaces, and latent structures (Cybenko, 1989; Hornik, 1991).
- Challenges: risk of overfitting, sensitivity to hyperparameters, instability under structural breaks, and difficulties with interpretability (Hastie et al., 2009).

Recent advances in optimization, initialization, and regularization (e.g., Adam, dropout, weight decay, gradient clipping) have improved the empirical reliability of neural networks in noisy financial environments (Kingma & Ba, 2015; Krogh & Hertz, 1992; Pascanu, Mikolov, & Bengio, 2013; Srivastava et al., 2014). Nonetheless, compared to tree ensembles, neural networks often require more extensive tuning and may be more sensitive to limited training samples (Christensen et al., 2023). These characteristics make them especially informative for assessing the interaction between model complexity and rolling-window length.

2.5 Real-Time Forecasting, Rolling Windows, and Evaluation Practices

A central insight from the predictive-econometrics literature is that model evaluation must mimic real-time conditions. Goyal and Welch (2008) and Rapach and Zhou (2013) show that in-sample performance is often a poor indicator of out-of-sample predictive accuracy, particularly in unstable financial environments (Goyal & Welch, 2008; Rapach & Zhou, 2013). Volatility forecasting studies therefore adopt rolling estimation windows to avoid look-ahead bias, allow parameter updates, and reflect the instability of financial relationships (Inoue et al., 2017; Rossi & Inoue, 2012).

Rolling windows have gained traction due to their ability to: track structural shifts in market dynamics, ensure strict temporal causality, allow models to adapt gradually to evolving conditions, and provide feasible (pseudo-out-of-sample) forecasts for evaluation (Inoue et al., 2017). However, rolling frameworks also introduce challenges. Short windows may increase

estimation noise, while long windows may obscure regime changes (Inoue et al., 2017). While Mean Squared Error (MSE) is the standard metric for assessing accuracy, it assumes symmetric loss aversion, and volatility-forecast rankings can depend on the chosen loss function and proxy (Patton, 2011). These evaluation practices provide context for assessing the incremental value of more sophisticated forecasting techniques.

2.6 Interpretability and Feature Importance in Predictive Modeling

As forecasting models increase in complexity, interpretability becomes an important concern in empirical finance. Linear models offer transparent economic interpretation through their coefficients, whereas this transparency diminishes in nonlinear specifications such as tree-based ensembles and neural networks (Hastie et al., 2009).

The interpretability literature distinguishes between model-specific and model-agnostic approaches. Model-specific measures exploit a model's internal structure, for example, coefficient-based summaries in linear regressions and split-based importance measures in tree ensembles (Breiman, 2001). Model-agnostic methods, such as SHAP values, aim to provide a unified explanation framework across model classes (Lundberg & Lee, 2017). In forecasting applications, however, model-agnostic explanations can be costly to compute at scale and may be sensitive to dependence structures among predictors (Aas et al., 2021; Molnar, 2019). These limitations motivate the widespread use of model-consistent, computationally tractable importance diagnostics in large empirical studies.

Consistent with this view, this thesis relies on intrinsic feature-importance measures (standardized coefficients, impurity, and gain) that are computationally feasible within a rolling-window framework. To facilitate comparison across models and stocks despite the heterogeneity of predictors, these importance scores are aggregated into broad economic categories. This approach links predictive performance to interpretable economic information (e.g., liquidity, trading activity, volatility) while respecting the computational and statistical constraints of the high-dimensional setting.

2.7 Gaps in Literature and Contribution of This Study

This thesis does not aim to close gaps in the volatility-forecasting literature. Instead, its contribution is empirical and comparative. It provides a disciplined assessment of how increasingly flexible forecasting models, ranging from linear and penalized regressions to tree-

based ensembles and shallow feed-forward neural networks, perform in short-horizon realized-volatility prediction when evaluated under identical and realistic conditions.

Using a common daily high-frequency-derived microstructure dataset (constructed from millisecond TAQ data) and a strictly one-day-ahead rolling evaluation protocol, the analysis clarifies how model class and estimation-window length jointly affect out-of-sample volatility forecasts across 30 large U.S. equities. All models are trained on the same feature set, including lagged realized volatility and a common set of microstructure indicators, and are evaluated using the same out-of-sample cutoff and rolling-window schedule, ensuring strict temporal causality. The study further benchmarks these models against standard time-series alternatives, including SMA, AR(1), and GARCH(1,1), allowing performance differences to be interpreted relative to commonly used baseline econometric specifications (Bollerslev, 1986). More specialized realized-volatility benchmarks (e.g., HAR-family specifications and related realized-volatility models) are not implemented here; the benchmark set is intentionally kept parsimonious to maintain a controlled like-for-like design centered on microstructure indicators, leakage-safe rolling estimation, and operational tuning constraints.

Relative to existing studies, the distinguishing feature is the controlled “like-for-like” design: all models share the same predictors, the same forecast horizon, and the same strict rolling-window timing, so differences in performance can be attributed more confidently to model flexibility and estimation-window length rather than to evaluation artifacts. To further reduce evaluation-driven differences, hyperparameters are tuned only on the in-sample period and then held fixed throughout the out-of-sample exercise, and preprocessing steps (e.g., winsorization and standardization) are implemented in a leakage-safe manner within each rolling training window where required.

Section 3: Methodology

This thesis adopts a forecasting framework that combines econometric and machine learning approaches to predict realized volatility in financial markets. The analysis relies on high-frequency intraday indicators aggregated to the daily level and employs systematic preprocessing and rolling-window estimation to approximate real-time forecasting conditions. This framework is designed to facilitate a consistent comparison of linear and nonlinear models under identical forecasting horizons.

Methodological choices prioritize internal validity and comparability across models. All data transformations are computed exclusively using the information available in the training sample to prevent information leakage. Model estimation follows a rolling-window scheme to replicate real-time forecasting, and reproducibility is ensured through standardized procedures and fixed random seeds where applicable. Together, these design choices support an evaluation of model performance relative to established volatility benchmarks.

3.0 Research Questions and Hypotheses

This thesis evaluates whether increasingly flexible model classes deliver meaningful improvements in realized-volatility forecasting. In particular, it tests whether added model complexity improves out-of-sample accuracy in environments characterized by persistent volatility, regime variation, and microstructure noise.

The following hypotheses are examined:

H1 (Regularization effect)

Penalized linear models (Lasso, Ridge, Elastic Net) outperform OLS due to severe multicollinearity among intraday predictors.

H2 (Nonlinearity)

Tree-based models outperform linear regressions in out-of-sample forecasting because they can capture nonlinearities and interaction effects among intraday predictors.

H3 (Model capacity)

Neural networks provide additional forecasting gains only if volatility dynamics exhibit nonlinear structure beyond that captured by tree-based models.

H4 (Estimation-window length)

Forecasting performance is sensitive to rolling-window length, with shorter windows leading to less stable parameter estimates and reduced out-of-sample accuracy, particularly for high-capacity nonlinear models.

All hypotheses are evaluated using strictly out-of-sample performance metrics under a rolling, one-step-ahead forecasting design.

3.1. Data & Variables

3.1.1 Data Description

This research utilizes high-frequency financial data sourced from the Wharton Research Data Services (WRDS) platform, specifically the Millisecond Intraday Indicators dataset (Wharton Research Data Services, n.d.). This dataset provides a daily summary of 198 intraday variables, including trade prices, volumes, trade classifications, liquidity, and volatility measures, constructed from millisecond-level TAQ data. These variables are extracted and processed to construct predictive features for forecasting realized volatility. A variety of machine learning models are trained and evaluated on this dataset, with predictive performance assessed using standard out-of-sample accuracy metrics. The dataset is accessible through the HEC Montréal Library.

The empirical analysis focuses on daily observations (business days) for 30 Dow Jones Industrial Average constituents, covering the period from September 2003 to July 2025. This extensive sample spans multiple market regimes, including the 2008 financial crisis, the COVID-19 pandemic, and more recent volatility episodes, thus allowing evaluation across structurally distinct volatility regimes. To ensure that the analysis reflects the price dynamics of the primary common stock, the data was filtered using the *SYM_SUFFIX* identifier; only observations with a blank suffix were retained, thereby excluding secondary issues such as preferred shares, warrants, and 'when-issued' trading lines

To preserve the integrity of the forecasting framework, the sample is split chronologically. All observations up to July 31, 2024, are designated as the in-sample (ITS) period used for model training, validation, and hyperparameter tuning. Observations from August 1, 2024, onward form the out-of-sample (OOS) period, which is reserved strictly for performance evaluation. This separation is fixed at the beginning of the analysis, and the OOS segment is excluded from all preprocessing, feature engineering, and model selection procedures. Doing so avoids information leakage and ensures that predictive accuracy reflects genuine, real-time forecasting performance.

3.1.2 Variable Selection

Given that the objective of this research is to forecast realized volatility, a quantitative time-series econometric design was adopted. Accordingly, only numerical predictors were retained for model development, while categorical and non-numeric variables were excluded.

The WRDS Intraday Indicators dataset provides two candidate realized volatility measures: $ivol_t$, trade-based intraday volatility (variance of high-frequency trade returns), and $ivol_q$, quote-based intraday volatility (variance of high-frequency mid-quotes). Both are derived from second-by-second data aggregated to the daily level and could serve as dependent variables in prediction tasks.

For this study, $ivol_t$ (trade-based realized volatility) was selected as the target variable. The rationale is that the standard definition of realized volatility in the empirical finance literature is based on trade returns, as in Andersen & Bollerslev (1998) and Barndorff-Nielsen & Shephard (2002). Trade-based volatility directly reflects the ex-post variability of transaction prices experienced by investors, making it the more theoretically consistent and widely accepted measure. While $ivol_q$ can be useful for mitigating microstructure noise like bid–ask bounce, it reflects quote dynamics rather than actual transaction prices. Since the aim here is to forecast the realized volatility of returns, the trade-based measure $ivol_t$ provides a more accurate and relevant benchmark.

3.1.3 Log Transformation of the Target Variable

The dependent variable in this study is the trade-based realized volatility measure ($ivol_t$), which in the raw dataset takes extremely small positive values (e.g., 0.000000107, 3.7E-08). As is typical for realized volatility, the distribution is heavily right-skewed and dominated by large spikes during turbulent periods.

These distributional properties are problematic for regression and machine learning models, which typically assume more stable variance and more symmetric error structures. To address these issues, a logarithmic transformation was applied prior to modeling.

$$y_t = \log(ivol_t)$$

The log compresses extreme values, reduces skewness, and stabilizes variance, thereby mitigating heteroskedasticity and limiting the disproportionate influence of outliers. It also improves distributional properties: while raw volatility is strictly positive and highly skewed, log-volatility is closer to symmetric and exhibits behavior more compatible with approximate

Gaussian assumptions, which better aligns with the assumptions underlying many regression-based methods.

The choice of log-volatility also reflects established econometric practice. Canonical models such as AR(1), HAR, and stochastic volatility are commonly specified in log form, where volatility displays more stationary and mean-reverting dynamics. Moreover, the transformation enhances interpretability, as regression coefficients on log-volatility can be interpreted in multiplicative terms, meaning that changes in predictors correspond to proportional, or percentage, changes in volatility.

In summary, transforming $ivol_t$ into log-volatility reduces skewness, stabilizes variance, improves model fit, and ensures consistency with both econometric convention and the statistical requirements of forecasting models.

3.2. Data Preprocessing

3.2.1 Data Cleaning

Data cleaning was performed separately for each of the thirty Dow Jones Industrial Average constituents over the in-sample (ITS) period, which contains up to 5,258 business days between September 2003 and 31 July 2024. The objective was to construct a balanced and internally consistent panel of predictors, while avoiding interpolation noise and ensuring that no information from the out-of-sample (OOS) window influenced model development.

Removal of variables with structural missingness

The procedure began by assessing missingness across all 198 Intraday Indicator variables provided in the WRDS dataset. Variables with more than 2,500 missing observations, approximately half of the ITS window, were excluded. These fields consist primarily of odd-lot statistics (e.g., *n_oddlot_trade*, *oddlot_vol*, *oddlot_dollar*) and symbol-suffix identifiers. As noted in the WRDS Intraday Indicators manual, odd-lot variables populate only when odd-lot trades occur and therefore exhibit structural sparsity rather than measurement error. Their highly intermittent coverage makes them weak time-series predictors and unsuitable for stable model estimation, and they were therefore removed from the predictor set.

Restricting the sample to a homogeneous microstructure environment

A second category of variables exhibited missingness concentrated almost entirely before early 2007. These include the retail order-flow indicators and ISO-related measures, which the WRDS manual documents as being nominally available from October 2006 onward. Empirical inspection across Dow Jones Industrial Average constituents, however, shows that these variables remain sparsely populated throughout late 2006 and only become reliably available from 26 February 2007 onward. Starting the ITS window on this date ensures that all retail and ISO variables are simultaneously present across stocks, while losing the minimum number of valid trading days. Adopting 26 February as the start date therefore maximizes sample length while guaranteeing a homogeneous microstructure environment across the full cross-section.

Final filtering and removal of residual missing values

Once the time window and variable set were finalized, only numeric predictors were retained. Any remaining missing values were sparse and isolated, and corresponding rows were removed rather than imputed, preventing the introduction of artificial variation. This produced a balanced ITS dataset with complete observations for all predictors used in correlation screening, feature preprocessing, and model estimation.

In contrast, OOS observations (from 1 August 2024 onward) were deliberately excluded from all cleaning decisions. They did not inform variable selection, missingness thresholds, or time-window choices. OOS data were subjected solely to deterministic transformations. This strict temporal separation eliminates any possibility of information leakage and preserves the validity of the forecasting evaluation.

3.2.2 Handling Multicollinearity

To mitigate multicollinearity and reduce redundant predictors, a structured feature-reduction procedure was implemented using the in-sample (training) data only. After restricting the dataset to numeric variables and removing observations with missing values, the pairwise correlation matrix was computed using only the predictor variables. No predictive information from $ivol_t$ is used in feature selection, ensuring that feature selection relied solely on relationships among predictors and not on correlations with the dependent variable.

Variables displaying an absolute correlation above 0.95 with at least one other predictor were removed. A threshold of 0.95 is deliberately conservative: it retains economically relevant

collinear predictors while eliminating near-linear dependencies that destabilize estimation. For any pair of highly correlated features, the feature appearing later in the upper-triangle structure of the correlation matrix was removed, providing a systematic and reproducible rule for identifying collinear variables.

The resulting feature set is more compact and statistically stable, which benefits both econometric and machine-learning models. Reducing multicollinearity enhances numerical stability, improves interpretability, and decreases the risk of overfitting. Moreover, it ensures that performance metrics, such as mean squared error (MSE) and R^2 , reflect genuine explanatory power rather than artificial gains arising from duplicated or near-duplicated predictors.

3.2.3 Winsorization and Standardization

To prepare the explanatory variables for model estimation, a two-step preprocessing procedure, winsorization followed by standardization, was applied. High-frequency financial data often contain extreme values arising from unusual trading conditions or reporting irregularities. Such outliers can distort regression coefficients and destabilize machine-learning optimization. To limit these effects while preserving the underlying distributional structure, each predictor was winsorized at the 1st and 99th percentiles.

$$X_{t,j}^* = \min\{\max(X_{t,j}, Q_{0.01,j}), Q_{0.99,j}\}$$

$$\tilde{X}_{t,j} = \frac{X_{t,j}^* - \mu_j}{\sigma_j}$$

Following winsorization, all variables were standardized to zero mean and unit variance. Standardization is preferred to min–max normalization in this setting because financial predictors are heavy-tailed and exhibit substantial scale heterogeneity. Rescaling by variance yields features that are comparable across dimensions and improves the numerical stability of linear, penalized, and nonlinear models.

The preprocessing procedure was applied consistently to both training and test sets, but with an essential safeguard: all parameters were estimated from the training sample only. Specifically, percentile cutoffs, means, and standard deviations were computed within the training window and then stored. For the test set, winsorization was performed by capping values at the training-based thresholds, and standardization was applied using the training-sample mean and standard deviation. This approach guarantees that test data are transformed

in a manner strictly comparable to the training data, while avoiding any look-ahead bias or leakage of future information.

By deriving all preprocessing rules from the ITS window and applying them forward, the evaluation framework closely mimics real-time forecasting conditions. Consequently, the reported predictive performance reflects genuine model generalization rather than artifacts of the data transformation process.

In practice, winsorization and standardization were implemented through a custom scikit-learn transformer (*WSWrapper*) and incorporated directly into the modeling pipeline. This ensures that preprocessing is fitted only on the ITS sample and automatically applied to each fold of cross-validation, preserving the strict training-only estimation of transformation parameters. This behavior is enforced programmatically by fitting the transformer only on training folds and reusing stored statistics during validation and forecasting.

3.2.4 Lagged Volatility Variables

A standard characteristic of financial time series is the presence of volatility clustering, whereby episodes of elevated volatility tend to be followed by similarly high levels, and tranquil periods tend to persist. This persistence structure is well documented in the volatility literature and forms the basis of classical econometric models such as AR(1), GARCH, and HAR. To incorporate this temporal dependence directly into the feature set, lagged versions of the realized volatility measure were constructed.

Specifically, five lagged values of the log-transformed realized volatility (lags 1 through 5) were generated. This choice allows representation of short-memory dynamics while limiting overfitting within relatively narrow rolling windows. The lagged features provide the models with information about recent volatility dynamics, enabling them to capture short-horizon persistence and mean-reversion patterns. Including multiple lags also helps approximate higher-order autoregressive behavior without imposing a parametric time-series structure, which is particularly advantageous for machine-learning algorithms that rely on flexible, data-driven relationships. Five lags additionally strike a balance between capturing short-horizon volatility persistence and avoiding dimensional inflation within rolling windows; extending the lag structure further materially reduces the effective sample size and increases estimation noise without clear gains in forecasting accuracy.

In line with good forecasting practice and to avoid look-ahead bias, lagged volatility features were constructed using only past information. The lags were created within the in-sample window and then applied forward, ensuring that each observation in the out-of-sample period uses only volatility data that would have been available at the time of prediction. This approach maintains the temporal integrity of the dataset and aligns the forecasting design with real-world settings.

These lagged log-volatility variables were integrated directly into the modeling pipeline alongside the winsorized and standardized predictors, ensuring consistent treatment across all models and cross-validation folds.

3.3. Forecasting Framework

3.3.1 Rolling-Window Design

The forecasting procedure is implemented using a fixed rolling-window design, in which models are re-estimated sequentially using only the most recent block of historical observations. This structure ensures a strictly forward-looking forecasting process: each prediction is generated using only information that would have been available at the corresponding point in time, thereby preventing look-ahead bias and aligning the setup with the constraints of real-time decision-making in financial markets.

In this framework, the training window consists of a fixed-length sequence of daily observations ending at time t . For each day in the out-of-sample (OOS) period, beginning on 1 August 2024, the model is fitted on the most recent window and used to produce a one-day-ahead forecast for $t+1$. After the forecast is generated, the window shifts forward by one day, the earliest observation is discarded, the newly realized volatility enters the window, and the entire estimation process is repeated. This yields a sequence of genuinely out-of-sample forecasts and provides a consistent basis for evaluating predictive performance across the full OOS horizon.

$$\hat{y}_{t+1} = f_t(X_t; \theta)$$

The rolling structure ensures strict temporal causality: every forecast is based solely on information available up to day t , and no future data are incorporated at any stage. This is essential in financial forecasting, where forward-looking evaluation must avoid any form of look-ahead bias. The window length is held fixed throughout the OOS period, allowing the

effect of different history lengths ($W = 21$ reflects one trading month; $W = 252$ corresponds to one trading year.) to be examined directly. Short windows adapt more rapidly to recent volatility regimes, while long windows provide smoother parameter estimates. These window lengths correspond to short-memory and medium-memory volatility dynamics commonly studied in the realized-volatility literature.

While rolling estimation does not guarantee uniformly strong performance across all assets or window sizes, it delivers an unbiased and operationally realistic framework for evaluating predictive accuracy. By relying exclusively on backward-looking information at each step, the design provides a credible representation of how these models would behave in real-time forecasting applications.

3.3.2 Hyperparameter Tuning Strategies

Hyperparameters control the structural configuration, regularization intensity, and effective complexity of each forecasting model. Their selection must respect the time-series nature of the data, avoid any use of future information, and remain computationally feasible within the rolling-window forecasting routine. Two hyperparameter selection strategies are considered: a fully adaptive double-rolling approach, and the hybrid approach adopted in this thesis.

In the empirical analysis, only the hybrid strategy is implemented; the fully adaptive double-rolling scheme serves as a conceptual benchmark for discussing the trade-off between flexibility and feasibility.

Fully adaptive Double-rolling approach

The fully adaptive double-rolling strategy recalibrates hyperparameters within every estimation window and treats each forecasting origin as an independent learning problem. For each out-of-sample (OOS) prediction date $t+1$, the preceding W observations $[t-W+1, t]$ are treated as a self-contained training sample, and an inner validation scheme is used to re-select the hyperparameters that minimize validation error:

$$\theta_t^* = \arg \min_{\theta \in \Theta} \text{CV_MSE}_{[t-W+1, t]}(\theta)$$

$$\text{MSE} = \frac{1}{T} \sum_{t \in \mathcal{T}} (\hat{y}_t - y_t)^2$$

Given these updated hyperparameters, the model is then re-estimated on the same window, producing the forecast

$$\hat{y}_{t+1} = \hat{f}_t(x_t; \theta_t^*)$$

$$\hat{f}_t(\cdot; \theta_t^*) \text{ trained on } [t - W + 1, t]$$

This approach yields the most flexible specification: both the structural configuration of the model (captured by θ_t^*) and the model parameters $\hat{f}_t$ are allowed to evolve freely as market conditions change. In principle, such adaptivity could better track shifts in volatility dynamics, microstructure frictions, and regime changes.

In practice, however, double-rolling methods are rarely feasible for high-frequency financial datasets. Hyperparameter tuning is computationally intensive even when performed once; repeating it for every OOS date multiplies the cost by the number of forecasts. For models with large hyperparameter spaces, such as Random Forests, Gradient-Boosting Machines, or Neural Networks, each window may require dozens or hundreds of candidate fits. Moreover, short rolling windows provide limited information for hyperparameter selection, increasing sensitivity to noise and raising the risk of overfitting local fluctuations rather than genuine structure. These issues make fully adaptive double-rolling strategies computationally prohibitive and statistically fragile for the present context.

Hybrid approach (adopted):

The compromise, and the approach adopted in this thesis for all models (Lasso, Ridge, Elastic Net, Random Forest, XGBoost, and the neural network), is a hybrid approach. Hyperparameters are tuned once on the entire ITS sample using time-series cross-validation, a procedure that respects the temporal ordering of the data.

$$\theta_t^* = \arg \min_{\theta \in \Theta} \text{CV_MSE}_{\text{ITS}}(\theta) \quad \text{with } \hat{f}_t(\cdot; \theta_t^*) \text{ trained on } [t - W + 1, t]$$

The resulting hyperparameters θ^* are treated as fixed structural characteristics of each model and carried forward unchanged into the out-of-sample (OOS) period.

Forecasting is still implemented in a rolling manner. For each out-of-sample (OOS) prediction date $t+1$, the model is re-estimated on the most recent W observations $[t-W+1, t]$ using these fixed hyperparameters. The one-step-ahead forecast is given by $\hat{y}_{t+1} = \hat{f}_t(x_t; \theta_t^*)$.

This avoids “double-dipping” into the OOS sample, prevents look-ahead bias, and keeps the computational burden manageable, while still allowing the model coefficients (or trees, in the case of ensembles) to adapt to new information.

Overall, the hybrid strategy offers a disciplined compromise between statistical adaptivity and operational feasibility, ensuring that hyperparameter choices remain exogenous to the forecasting exercise while model coefficients evolve with incoming information.

3.3.3 Evaluation Metrics

Model performance is assessed exclusively on the out-of-sample (OOS) period using strictly forward-looking forecasts generated by the rolling-window procedure. All metrics are computed from realized forecast errors and aggregated across the OOS horizon. This ensures that reported results measure true predictive performance rather than in-sample fit or cross-validation error.

Let y_{t+1} denote the realized (log) volatility at time $t+1$, and let $\hat{y}_{t+1}$ denote the corresponding one-step-ahead forecast produced at time t using only information available up to t .

Mean Squared Error (MSE)

The primary loss function is the Mean Squared Error (MSE), defined as

$$\text{MSE} = \frac{1}{T_{\text{OOS}}} \sum_{t=T_0+1}^T (y_t - \hat{y}_t)^2$$

Where T_{OOS} denotes the number of out-of-sample observations and T_0 is the last index of the in-sample period.

MSE penalizes large forecast errors disproportionately and is therefore particularly appropriate in a volatility forecasting context, where occasional extreme deviations can have material consequences for risk management and derivative pricing. Using squared errors gives greater weight to large predictive failures, aligning the loss function with applications such as Value-at-Risk and portfolio risk control, where underestimation of volatility is especially costly.

Out-of-sample coefficient of determination (R^2)

To evaluate performance relative to a naive benchmark, an out-of-sample coefficient of determination is computed for each model:

$$R^2 = 1 - \frac{\sum_{t=T_0+1}^T (y_t - \hat{y}_t)^2}{\sum_{t=T_0+1}^T (y_t - \bar{y}_t)^2}$$

where $\bar{y}_t$ is the rolling mean of realized volatility over the corresponding training window. This benchmark reflects a simple historical-average forecast and provides a meaningful baseline for assessing whether model complexity translates into genuine predictive improvement. An R^2 greater than zero indicates that a model outperforms the naive predictor, while negative values indicate underperformance. Because this statistic compares forecast errors directly to those of a simple reference model, it allows performance comparisons across modeling classes on a common scale.

The combination of MSE and out-of-sample R^2 provides complementary perspectives on forecast quality. MSE captures absolute predictive accuracy, while R^2 offers a relative measure of improvement over a simple baseline. Models are therefore evaluated not only by minimizing error but by demonstrating forecasting improvement over naive baselines

Stability and robustness of forecasting performance

As a robustness check, the out-of-sample (OOS) evaluation period is split into two equal subperiods, referred to as the early and late OOS samples. For each model and each asset, forecasting performance metrics are computed separately over these two subperiods using the same rolling-window procedure. The resulting early and late performance statistics are then aggregated across assets. This split is intended to assess the temporal stability of predictive performance and to identify potential degradation or improvement over time.

3.4. Model Specifications

This section describes the forecasting models used to predict one-day-ahead realized volatility for each Dow Jones Industrial Average constituent. All models are estimated separately for each stock using only its own historical data. No cross-sectional pooling or multi-asset learning is employed. Each model therefore represents a purely time-series forecasting framework applied at the single-asset level.

All models use the same winsorization and standardization scheme for the predictor variables, and are implemented under a common rolling-window design. Neural networks additionally

standardize and then de-standardize the log-volatility target inside each training window, but the preprocessing of predictors is otherwise identical across models, ensuring comparability in forecasting performance.

For each model class, the objective is to approximate the unknown forecasting function

$$y_{t+1} = f(x_t) + \varepsilon_{t+1}$$

3.4.1 Linear and Penalized Linear Models

Four linear forecasting models are considered: Ordinary Least Squares (OLS), Lasso, Ridge, and Elastic Net regression. These specifications impose different constraints on the coefficient vector to mitigate multicollinearity, reduce dimensionality, and limit overfitting in the high-dimensional intraday indicator set.

Ordinary Least Squares (OLS)

The classical linear regression model is given by

$$y_{t+1} = \alpha + \beta^\top x_t + \varepsilon_{t+1}$$

Where β is the vector of coefficients. Estimation is performed via least squares within each rolling window.

Lasso Regression

Lasso Regression introduces an ℓ_1 penalty that encourages sparsity in the coefficient vector:

$$\min_{\alpha, \beta} \frac{1}{n} \sum_{t=1}^n (y_{t+1} - \alpha - \beta^\top x_t)^2 + \lambda \|\beta\|_1$$

This penalty shrinks some coefficients exactly to zero, allowing automatic feature selection in environments with correlated intraday indicators.

Ridge Regression

Ridge regression applies an ℓ_2 penalty to stabilize coefficient estimates under multicollinearity:

$$\min_{\alpha, \beta} \frac{1}{n} \sum_{t=1}^n (y_{t+1} - \alpha - \beta^\top x_t)^2 + \lambda \|\beta\|_2^2.$$

Unlike Lasso, Ridge retains all predictors but shrinks coefficients continuously.

Elastic Net

Elastic Net combines the two penalties:

$$\min_{\alpha, \beta} \frac{1}{n} \sum_{t=1}^n (y_{t+1} - \alpha - \beta^\top x_t)^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2.$$

This specification balances sparsity and stability and is particularly effective when predictors are both numerous and highly correlated.

All penalized regression hyperparameters are selected using expanding-window time-series cross-validation on the ITS window as described in Section 3.3.2, and then fixed during the out-of-sample forecasting phase

3.4.2 Tree-Based Models

To capture nonlinearities and interactions among predictors, two nonparametric tree-based methods are employed: Random Forests and XGBoost. Both methods operate on the same standardized predictor set as the linear models, enabling a controlled comparison between linear and nonlinear forecasting structures.

Tree-based methods are particularly well-suited to realized volatility modeling, as high-frequency volatility measures are known to exhibit nonlinear dynamics, regime-dependent behavior, and interaction effects across intraday indicators that are difficult to capture with linear specifications. By allowing for flexible, data-driven partitioning of the predictor space, tree ensembles can accommodate nonlinear marginal effects and higher-order interactions without requiring explicit model specification.

Random Forest

Random Forest aggregates predictions across a collection of regression trees:

$$\hat{f}(x_t) = \frac{1}{B} \sum_{b=1}^B T_b(x_t),$$

where each tree T_b is fitted to a bootstrap sample and trained using random subsampling from the pool of standardized predictors at each split.

This design decorrelates individual trees and reduces estimation variance through averaging, making Random Forest particularly well-suited to noisy and collinear predictors, features that are prevalent in realized-volatility data derived from intraday observations.

XGBoost

Extreme Gradient Boosting constructs trees sequentially in a gradient-boosting framework. Each new tree improves the ensemble by fitting residuals from previous models:

$$\hat{f}^{(m)}(x_t) = \hat{f}^{(m-1)}(x_t) + \eta \cdot g_m(x_t),$$

where g_m is a regression tree and η is the learning rate that controls the contribution of each tree.

In contrast to Random Forest, which relies on bagging, XGBoost explicitly minimizes a regularized objective function using gradient-based updates. Regularization penalties on leaf weights, random subsampling of observations and predictors, and shrinkage through learning rate scaling are employed to control model complexity and mitigate overfitting

Both Random Forest and XGBoost hyperparameters are chosen via time-series cross-validation on the ITS and held fixed during rolling forecasts. For XGBoost, two tuning procedures are considered: a randomized search strategy and Bayesian optimization implemented via Optuna.

3.4.3 Neural Network Models (NN1–NN5)

Neural networks are included to capture nonlinear relationships and interaction effects that may not be adequately represented by linear specifications or tree-based ensemble models in forecasting realized volatility. In a financial context, volatility dynamics reflect nonlinear persistence, regime shifts, and complex interactions among intraday indicators that are difficult to model using fixed parametric forms.

All networks share a common feed-forward architecture with SiLU activation functions and differ only in depth, ranging from one to five hidden layers (NN1–NN5). Preprocessing, regularization, and the rolling-window estimation procedure are held constant across specifications, ensuring comparability. This design permits a controlled assessment of how network depth alone affects forecasting performance.

Structural interpretation of depth

Network depth is not treated as a tuning device alone but as a structural dimension of representational capacity. Shallow architectures approximate smooth nonlinear response surfaces, whereas deeper networks allow hierarchical feature transformations that better represent interaction effects across indicators and time. In the context of volatility forecasting, greater depth may capture latent regimes in which liquidity measures, order-flow imbalances, and intraday dispersion interact differently under calm and stressed market conditions. The comparison between NN1 and NN5 therefore tests whether volatility is governed primarily by low-order nonlinearities or by deeper interaction structures.

Data preprocessing

Volatility is modeled in logarithmic form to stabilize variance and reduce skewness. However, the log-transformed volatility series remains highly heterogeneous across assets and exhibits pronounced dispersion over time. When used directly as a training target, this scale variability leads to poorly conditioned optimization and unstable gradient dynamics in neural-network estimation.

To improve numerical stability, the target variable is standardized within each rolling estimation window. Standardization centers the series and rescales it to unit variance, thereby preventing large-magnitude errors from dominating gradient updates and ensuring that network training remains well-behaved across assets with different volatility levels. After prediction, the standardized forecasts are transformed back to the original log-volatility scale to preserve economic interpretability and ensure comparability with other models.

All numeric predictors are winsorized and standardized within each rolling window using training-sample statistics, consistent with the preprocessing applied across all models in the study. This ensures that neural networks operate on numerically well-scaled inputs while preserving the strict separation between in-sample estimation and out-of-sample evaluation. Training parameters (percentile cutoffs, means, and standard deviations) are computed exclusively from historical observations and applied forward, thereby preventing look-ahead bias.

Training procedure and regularization

All neural networks are re-estimated from scratch in each rolling window using the Adam optimizer, which is well suited to the nonstationary and noisy nature of realized-volatility data.

Training on short and overlapping samples introduces numerical instability and elevated overfitting risk, requiring explicit control of optimization dynamics.

Regularization and stabilization are therefore treated as structural components of the modeling framework rather than optional tuning devices. Gradient clipping is applied to prevent divergence under large or erratic gradient updates. Weight decay (ℓ_2 regularization) constrains parameter growth across repeated re-estimation cycles, while early stopping prevents overtraining on locally short samples. Dropout further limits dependence on transient predictor patterns that arise in narrow windows. Xavier initialization ensures stable signal propagation at initialization and reduces sensitivity to layer depth.

In practice, early stopping and weight decay are the primary controls against window-level overfitting, while gradient clipping is essential for numerical stability under heavy-tailed target distributions. Applied jointly, these mechanisms ensure that forecasting performance reflects economically meaningful signal rather than optimization artifacts.

Interpretability

Neural networks are employed in this study primarily as predictive models rather than structural econometric inferential tools. Unlike penalized regressions and tree-based ensembles, neural networks do not produce naturally interpretable coefficient paths. Feature importance is therefore approximated using aggregate statistics derived from first-layer weight magnitudes, used as a heuristic diagnostic rather than a formal structural interpretation; however, interpretability is treated as secondary to forecast performance.

3.5. Benchmark For Comparison

To evaluate the incremental value of the machine-learning models, their forecasts are compared against simple and widely used time-series benchmarks. All benchmarks are implemented in a rolling, one-step-ahead fashion using the same window lengths as the main models ($W \in \{21, 252\}$) and are applied separately to each stock. Performance is assessed using the out-of-sample MSE and the R-squared measure where the reference forecast is the rolling mean of past volatility.

Three benchmark models are considered: a rolling simple moving average (SMA), a rolling autoregressive model of order one, AR(1), and a rolling GARCH(1,1) specification with AR(1) conditional mean.

3.5.1 Simple Moving Average (SMA)

The first benchmark is a rolling simple moving average of past realized volatility. For each forecast origin t , the model uses the preceding W observations as a training window and sets the one-step-ahead forecast equal to the in-window average:

$$\hat{y}_{t+1}^{SMA} = \frac{1}{W} \sum_{i=0}^{W-1} y_{t-i}$$

where y_t denotes the (log) realized volatility at time t .

This approach has the virtue of simplicity and serves as a naive expectation that volatility tomorrow equals the historical mean of the recent past. By construction, the SMA forecast is constant within each rolling window and does not account for more complex dynamics or interactions across features. Because the SMA forecast coincides with the rolling-mean benchmark used in the definition of out-of-sample R^2 , its R^2 is mechanically equal to zero by construction, and only its MSE is reported.

3.5.2 Autoregressive Model AR(1)

The second benchmark is a rolling autoregressive model of order one, AR(1), estimated within each window. At each forecast origin t , the past W observations are treated as a local training sample for the regression

$$y_s = \alpha + \phi y_{s-1} + \varepsilon_s, \quad s = t - W + 1, \dots, t$$

estimated by ordinary least squares. Where $y_s = \log(ivol_s)$, α is an intercept, ϕ captures the persistence of volatility, and ε_s is a white-noise error term.

This rolling AR(1) benchmark allows both the intercept and the persistence parameter to adjust over time, accommodating slow changes in unconditional volatility and in the degree of volatility clustering.

Estimation is performed recursively in a rolling manner: at each out-of-sample date, the AR(1) parameters α and ϕ are estimated using the most recent W observations. The one-step-ahead forecast is then obtained as

$$\hat{y}_{t+1|t}^{AR(1)} = \hat{\alpha}_t + \hat{\phi}_t y_t$$

The AR(1) model incorporates volatility clustering by allowing today's forecast to depend directly on yesterday's realized volatility.

3.5.3 GARCH(1,1) Model

The third benchmark augments the AR(1) mean with a GARCH(1,1) volatility specification, again estimated in a rolling manner. Within each window $[t - W + 1, t]$, the conditional mean follows an AR(1):

$$y_s = \mu_s + \varepsilon_s, \quad \mu_s = \alpha + \phi y_{s-1}$$

while the conditional variance of ε_s is modeled as

$$\sigma_s^2 = \omega + \beta \sigma_{s-1}^2 + \gamma \varepsilon_{s-1}^2$$

The model is estimated by maximum likelihood under a Gaussian error distribution, and a one-step-ahead forecast of the conditional mean $\hat{\mu}_{t+1|t}$ is used as the volatility forecast $\hat{y}_{t+1|t}$

This benchmark represents the canonical low-dimensional econometric approach to volatility modeling, combining autoregressive mean dynamics with persistent conditional variance dynamics. It provides a natural point of comparison for the more flexible machine-learning models, which do not impose a parametric volatility structure.

As in the AR(1) case, out-of-sample performance is summarized by MSE and by R^2 relative to the rolling-mean benchmark. Because all three benchmarks are implemented under the same rolling-window protocol and evaluated on the same OOS horizon, they offer a transparent baseline against which the marginal contribution of penalized regressions, tree-based ensembles, and neural networks can be assessed.

3.6. Feature Importance

To complement the forecasting evaluation, the thesis reports feature-importance measures for selected models. All importance measures are computed under the same preprocessing steps and rolling-window structure as the forecasting exercise, ensuring that only information available at the forecast origin is used.

Due to the large number of high-frequency predictors and stock-specific data-cleaning procedures, feature importance is analyzed using a structured aggregation approach, described below.

3.6.1 Feature Categorization

The data-cleaning procedure removes highly correlated predictors on a stock-by-stock basis. As a result, the exact set of retained features differs across the 30 Dow Jones Industrial Average stocks. This heterogeneity makes direct averaging of feature-importance scores at the individual-variable level inappropriate, as a given predictor may be absent for some assets.

To address this issue, all candidate predictors are mapped ex ante into 11 mutually exclusive and economically motivated feature categories, each representing a distinct dimension of market information (e.g., liquidity, order flow, trading activity). Each feature belongs to exactly one category.

Rather than aggregating raw importance magnitudes across heterogeneous feature sets, global feature importance is assessed using a selection-frequency approach. For each stock, model, and rolling window, the top three most important features, as ranked by the model-specific importance measure explained in 3.6.2, are identified. These top-ranked features are then mapped to their respective categories.

For a given model, category-level importance is measured by the frequency with which features from a given category appear among the top three predictors, aggregated across the 30 stocks.

This approach provides a standardized and interpretable measure of global importance that is robust to stock-specific feature availability while preserving the relative prominence of different information types.

Table 1 summarizes the feature categories used in the analysis. A complete mapping of individual variables to categories is provided in the Appendix A.

Table 1: Feature Categories Used for Category-Level Importance Aggregation

Category	Description
Liquidity and trading costs	Bid-ask spreads, depth, and realized spread measures
Price impact	Measures capturing the price response to order flow
Market efficiency	Variance ratio-based efficiency indicators
Returns and volatility	Past returns and realized volatility measures
Lagged volatility	Lagged values of realized volatility
Trading activity	Volume, dollar volume, and trade counts
Market microstructure	Microstructure-related trade and quote indicators
Order flow (Lee-Ready)	Order flow classified using the Lee-Ready algorithm
Order flow (alternative)	Order flow based on tick and WRDS classifications
Investor-type flow	Order flow segmented by investor type
Control variables	Time-of-day and market state controls

Each predictor is assigned to a single, mutually exclusive economic category. The categorization is used to aggregate feature-importance measures at the category level across stocks.

3.6.2 Model-Specific Importance Definitions

Feature importance is computed separately for each model class, using standard and model-appropriate measures.

Linear models (multivariate OLS, Lasso, Ridge, Elastic Net)

Within each rolling window, all numeric predictors are winsorized at the 1st and 99th percentiles and then standardized using the window mean and standard deviation. The target is log realized volatility, and the linear models are estimated on the one-step-ahead alignment used throughout the thesis: predictors at time t are used to forecast volatility at time $t+1$.

For multivariate OLS, the regression is re-estimated in every rolling window. Let $\beta_{j,t}$ denote the coefficient on feature j estimated in window t (excluding the intercept). Because predictors are standardized within the window, coefficient magnitudes are comparable across predictors. Over the OOS period, each feature's coefficient path is summarized by the mean, median, and standard deviation of $\beta_{j,t}$, and feature importance is defined as the mean of the absolute coefficient:

$$\text{Importance}_j = E_t[|\beta_{j,t}|].$$

For penalized linear models (Lasso, Ridge, Elastic Net), the same rolling estimation procedure is applied using fixed hyperparameters selected on the in-sample period via time-series cross-validation, with preprocessing embedded in the estimation pipeline. Feature importance is computed using the same absolute-coefficient definition and summarized over time using the mean, median, and standard deviation.

Random Forest

For Random Forest, feature importance is computed in each rolling window using the model's impurity-based importance, defined as the total reduction in squared-error loss attributable to splits on a given feature, aggregated across trees and normalized to sum to one. Hyperparameters are selected once on the in-sample period using time-series cross-validation with preprocessing embedded in the tuning pipeline. During the out-of-sample evaluation, the

Random Forest is re-estimated in each rolling window using these fixed hyperparameters, with predictors winsorized and standardized using window-specific training statistics.

XGBoost

For XGBoost, feature importance is based on gain, defined as the improvement in the regularized tree-splitting objective attributable to splits on a given feature. In each rolling window, an XGBoost model is re-estimated using fixed hyperparameters selected on the in-sample period (via Optuna-based tuning or randomized search, depending on the specification), with predictors winsorized and standardized using window-specific training statistics. Gain scores are extracted from the trained booster and aligned to the original feature set; features not used in any split within a given window are assigned zero importance. As with other models, feature-importance paths are summarized over the out-of-sample period using the mean, median, and standard deviation.

Neural networks (NN1–NN5)

Neural-network feature importance is computed using a window-consistent ablation procedure. For each rolling window, the network (NN1 through NN5) is trained on window-preprocessed predictors, and a one-step-ahead forecast is produced for the corresponding test observation. Holding the trained network fixed, each feature is then ablated one at a time by setting its standardized input value to zero, which corresponds to the in-window mean after standardization, and the forecast is recomputed.

For feature j in window t , importance is defined as the increase in squared error induced by ablation:

$$FI_{j,t} = \left(\hat{y}_{t+1}^{(-j)} - y_{t+1} \right)^2 - (\hat{y}_{t+1} - y_{t+1})^2$$

where $\hat{y}_{t+1}$ is the baseline forecast and $\hat{y}_{t+1}^{(-j)}$ is the forecast with feature j set to zero. Positive values indicate that removing the feature increases the squared prediction error at that forecast origin.

Section 4: Results

This section presents the empirical results of the volatility forecasting exercise. The analysis is structured around two main components. Section 4.1 evaluates out-of-sample forecasting performance under different rolling estimation windows, while Section 4.2 examines feature importance patterns across models and rolling-window specifications.

The model set includes linear, machine learning, and time-series benchmarks, consistent with the classification introduced in the methodology section. OLS denotes a multivariate linear regression model. Lasso, Ridge, and Elastic Net (ENet) correspond to penalized linear regressions employing ℓ_1 , ℓ_2 , and mixed regularization schemes, respectively. Among machine learning approaches, RF denotes the Random Forest model, while XGB1 and XGB2 refer to Extreme Gradient Boosting models estimated using Optuna-based hyperparameter optimization and randomized search, respectively. NN1–NN5 denote feed-forward neural networks with one to five hidden layers. Finally, AR1 represents a rolling autoregressive benchmark, and GARCH denotes a rolling GARCH(1,1) volatility benchmark.

Forecast accuracy is assessed using two complementary metrics: the out-of-sample coefficient of determination (R^2), computed relative to a historical mean benchmark, and the mean squared error (MSE). All results are reported across 30 Dow Jones stocks, allowing for cross-sectional comparisons and assessment of performance stability across assets.

4.1 Forecasting Performance

This subsection evaluates the out-of-sample forecasting performance of the proposed models under multiple rolling estimation windows. Section 4.1.1 reports results based on a 252-day rolling window, while Section 4.1.2 focuses on using a 21-day rolling window. Section 4.1.3 further examines the temporal stability of forecasting performance by comparing early and late out-of-sample periods.

4.1.1 252-Day Rolling Window

This subsection evaluates the out-of-sample one-day-ahead volatility forecasting performance of the proposed models using a 252-day rolling estimation window, which corresponds to approximately one trading year. Performance is evaluated using the Out-of-Sample R^2 (R_{OOS}^2) and the Mean Squared Error (MSE). The results represent the aggregate performance across all 30 constituent stocks of the Dow Jones Industrial Average.

Table 2: Out-of-Sample R^2 for One-Day-Ahead Volatility Forecasts (252-Day Rolling Window)

Model	Mean	Median	Std	Min	Max
OLS	-0.892	-0.586	0.919	-3.789	0.148
Lasso	0.235	0.230	0.124	-0.128	0.445
Ridge	0.247	0.238	0.103	0.048	0.429
Enet	0.254	0.238	0.102	0.012	0.446
RF	0.289	0.288	0.076	0.179	0.454
XGB1	0.271	0.274	0.078	0.148	0.424
XGB2	0.256	0.273	0.078	0.135	0.413
NN1	0.097	0.100	0.147	-0.312	0.312
NN2	0.123	0.184	0.186	-0.316	0.409
NN3	0.098	0.088	0.161	-0.207	0.359
NN4	0.045	0.045	0.151	-0.251	0.308
NN5	0.057	0.107	0.164	-0.331	0.301
AR1	0.152	0.132	0.061	0.049	0.278
GARCH	0.154	0.133	0.061	0.051	0.279

The table reports summary statistics of out-of-sample R^2 values across 30 Dow Jones Industrial Average stocks for each model. Mean, median, standard deviation (Std), minimum (Min), and maximum (Max) are computed across stocks. The R^2 is measured relative to a rolling historical-mean benchmark; positive values indicate improvement over the benchmark.

Table 3: Out-of-Sample MSE for One-Day-Ahead Volatility Forecasts (252-Day Rolling Window)

Model	Mean	Median	Std	Min	Max
OLS	1.775	1.514	0.890	0.773	4.999
Lasso	0.731	0.710	0.201	0.488	1.303
Ridge	0.716	0.712	0.168	0.480	1.101
Enet	0.711	0.711	0.179	0.470	1.143
RF	0.678	0.686	0.161	0.448	1.082
XGB1	0.705	0.682	0.157	0.470	1.126
XGB2	0.720	0.708	0.163	0.468	1.136
NN1	0.867	0.836	0.193	0.589	1.278
NN2	0.840	0.822	0.208	0.550	1.417
NN3	0.868	0.829	0.206	0.495	1.454
NN4	0.917	0.871	0.208	0.639	1.438
NN5	0.907	0.919	0.218	0.582	1.582
SMA	0.970	0.981	0.211	0.617	1.441
AR1	0.821	0.811	0.175	0.555	1.219
GARCH	0.819	0.808	0.174	0.553	1.205

The table reports summary statistics of out-of-sample mean squared error (MSE) values across 30 Dow Jones Industrial Average stocks for each model. Mean, median, standard deviation (Std), minimum

(Min), and maximum (Max) are computed across stocks. Lower MSE values indicate higher forecasting accuracy.

Linear Machine Learning Models

The results for linear models, including OLS, Lasso, Ridge, and Elastic Net, clearly indicate substantial heterogeneity in performance within this group.

The OLS model performs poorly, exhibiting a strongly negative mean R^2 (-0.892) and large dispersion across stocks, as reflected by both the high standard deviation and the extremely negative minimum value. This suggests that unconstrained linear regression suffers from overfitting and lacks robustness when forecasting volatility in a high-dimensional setting. Consistent with this finding, OLS also records by far the worst MSE performance, with a mean of 1.775, confirming its inadequacy.

In contrast, the regularized linear models, Lasso, Ridge, and Elastic Net, show a marked improvement. All three achieve positive mean and median R^2 values, indicating systematic gains over the historical mean benchmark. Among them, Elastic Net performs best in terms of average explanatory power (mean R^2 of 0.254), closely followed by Ridge and Lasso. The relatively low dispersion in R^2 and MSE across stocks suggests that regularization effectively stabilizes coefficient estimates and mitigates overfitting. This improvement highlights the importance of shrinkage in volatility forecasting when using large sets of predictors.

Tree-Based Machine Learning Models

Tree-based ensemble models, including Random Forest (RF) and Extreme Gradient Boosting, outperform the linear benchmarks across both evaluation metrics.

Random Forest achieves the highest mean R^2 (0.289) among all models considered, as well as the lowest mean MSE (0.678). Its strong median performance further indicates that the gains are not driven by a small subset of stocks but are broadly distributed across the cross-section. This suggests that Random Forest effectively captures nonlinearities and interaction effects that are important for volatility dynamics.

The two XGBoost specifications also perform very well, with mean R^2 values exceeding those of the regularized linear models and MSE values close to that of Random Forest. While XGB1, which relies on Optuna-based hyperparameter optimization, slightly outperforms XGB2, estimated using randomized search, their overall performance remains comparable. This

suggests that the predictive performance of XGBoost models is relatively robust to the choice of hyperparameter tuning strategy in this setting. The relatively low standard deviations in both R^2 and MSE further point to stable forecasting behavior across assets.

Overall, these results confirm the strong suitability of tree-based ensemble methods under longer rolling-window specifications, particularly in settings characterized by nonlinear relationships and heterogeneous predictor effects within a one-day-ahead forecasting framework.

Neural Network Models

The neural network specifications (NN1–NN5) exhibit weaker and more heterogeneous performance compared to tree-based ensemble models and regularized linear regressions under the 252-day rolling window specification. While some neural networks achieve positive mean R^2 values, their performance remains notably lower than that of the best-performing machine learning models. Several specifications display negative minimum R^2 values, indicating poor performance for certain stocks, and this instability is further reflected in relatively high standard deviations of both R^2 and MSE.

Examining performance across network depth reveals no systematic improvement as the number of hidden layers increases. Shallow architectures generally perform on par with, or slightly better than, deeper networks, while deeper specifications tend to exhibit greater dispersion in performance across assets. This pattern suggests that increasing model complexity through additional hidden layers does not translate into improved predictive accuracy under longer estimation windows.

In terms of MSE, neural networks consistently underperform tree-based models and regularized linear regressions. Taken together, these results indicate that, under a 252-day rolling window, feed-forward neural networks may struggle to generalize effectively, and increasing architectural depth does not yield clear performance gains. While neural networks are flexible by construction, their advantages do not appear to materialize in a forecasting framework based on longer rolling estimation windows.

Time-Series Benchmark Models

Finally, we examine the traditional time-series benchmarks, including AR(1) and GARCH models. Both models perform moderately well, with mean R^2 values around 0.15 and MSE values close to 0.82.

Their performance is clearly superior to OLS and several neural network specifications but inferior to tree-based models and regularized linear regressions. The relatively low dispersion across stocks indicates stable behavior, reflecting the parsimonious nature of these models. However, their limited explanatory power suggests that relying solely on past volatility dynamics is insufficient when richer predictor information is available.

Summary of Long Rolling-Window Results

Taken together, the results from the 252-day rolling window reveal a clear performance ranking across model groups. Tree-based ensemble models dominate in both R^2 and MSE, followed by regularized linear models, then time-series benchmarks, with neural networks and OLS trailing behind. These findings underscore the importance of nonlinear modeling and regularization when forecasting volatility using longer rolling estimation windows.

4.1.2 21-Day Rolling Window

This subsection evaluates the out-of-sample one-day-ahead volatility forecasting performance of the proposed models using a 21-day rolling estimation window, corresponding to approximately one trading month. As in Section 4.1.1, forecasting performance is assessed using the out-of-sample coefficient of determination (R^2) and the mean squared error (MSE). The reported results summarize performance across the 30 constituent stocks of the Dow Jones Industrial Average.

Table 4: Out-of-Sample R^2 for One-Day-Ahead Volatility Forecasts (21-Day Rolling Window)

Model	Mean	Median	Std	Min	Max
OLS	-0.694	-0.639	0.321	-1.587	-0.281
Lasso	-0.685	-0.674	0.280	-1.434	-0.268
Ridge	-0.012	0.002	0.057	-0.128	0.089
Enet	-0.432	-0.394	0.252	-1.216	-0.111
RF	0.017	0.022	0.031	-0.046	0.074
XGB1	-0.005	0	0.034	-0.091	0.097
XGB2	-0.004	0	0.026	-0.141	0.017
NN1	-0.420	-0.400	0.162	-0.851	-0.135
NN2	-0.275	-0.252	0.156	-0.679	-0.043
NN3	-0.235	-0.214	0.133	-0.609	-0.014
NN4	-0.256	-0.239	0.176	-0.871	-0.049
NN5	-0.208	-0.218	0.136	-0.620	0.003
AR1	-0.022	-0.020	0.042	-0.084	0.103
GARCH	-0.026	-0.031	0.040	-0.092	0.094

The table reports summary statistics of out-of-sample R^2 values across 30 Dow Jones Industrial Average stocks for each model. Mean, median, standard deviation (Std), minimum (Min), and maximum (Max) are computed across stocks. The R^2 is measured relative to a rolling historical-mean benchmark; positive values indicate improvement over the benchmark.

Table 5: Out-of-Sample MSE for One-Day-Ahead Volatility Forecasts (21-Day Rolling Window)

Model	Mean	Median	Std	Min	Max
OLS	1.145	1.092	0.271	0.716	1.905
Lasso	1.148	1.092	0.294	0.750	1.820
Ridge	0.691	0.683	0.148	0.462	1.055
Enet	0.977	0.936	0.265	0.610	1.661
RF	0.674	0.655	0.152	0.445	1.051
XGB1	0.694	0.666	0.146	0.475	1.019
XGB2	0.694	0.669	0.147	0.436	1.023
NN1	0.976	0.966	0.205	0.670	1.386
NN2	0.876	0.845	0.186	0.602	1.264
NN3	0.847	0.806	0.167	0.564	1.228
NN4	0.862	0.837	0.194	0.606	1.383
NN5	0.829	0.814	0.166	0.550	1.184
SMA	0.690	0.665	0.147	0.434	1.000
AR1	0.706	0.702	0.149	0.452	1.047
GARCH	0.707	0.700	0.145	0.462	1.028

The table reports summary statistics of out-of-sample mean squared error (MSE) values across 30 Dow Jones Industrial Average stocks for each model. Mean, median, standard deviation (Std), minimum (Min), and maximum (Max) are computed across stocks. Lower MSE values indicate higher forecasting accuracy.

Linear Machine Learning Models

The results for linear benchmark models reveal substantially weaker performance under the 21-day rolling-window specification. Both OLS and Lasso exhibit strongly negative mean R^2 values (-0.694 and -0.685 , respectively), indicating a systematic inability to outperform the historical mean benchmark under this rolling-window length. This poor performance is reinforced by their relatively high MSE values, which rank among the largest across all models.

Among the regularized linear specifications, Ridge regression performs notably better than OLS, Lasso, and Elastic Net. While its mean R^2 remains close to zero, Ridge achieves substantially lower MSE values, suggesting modest gains in predictive accuracy relative to other linear models. Elastic Net, by contrast, exhibits negative mean R^2 values and higher

dispersion, indicating that combining penalties does not yield clear benefits when estimation is based on shorter rolling windows.

Tree-Based Machine Learning Models

Tree-based ensemble models continue to perform relatively well under the 21-day rolling window, although their advantages are less pronounced than in the longer rolling-window specification. Random Forest achieves the highest mean R^2 among all models (0.017) and the lowest mean MSE (0.674), indicating the most consistent performance across stocks under this rolling-window length.

The two XGBoost specifications yield mean R^2 values close to zero and MSE values comparable to Random Forest. While neither specification delivers substantial improvements over the benchmark in terms of explanatory power, their relatively low dispersion across assets suggests stable one-day-ahead forecasting behavior. Overall, tree-based models remain among the best-performing approaches in terms of MSE, even though gains in R^2 are limited at this horizon.

Neural Network Models

Neural network specifications (NN1–NN5) exhibit weak and heterogeneous performance under the 21-day rolling window. All neural network models display negative mean R^2 values, indicating underperformance relative to the historical mean benchmark across most stocks. Although MSE values improve modestly for some architectures, neural networks remain inferior to tree-based models and Ridge regression in terms of predictive accuracy.

Consistent with the results obtained under the 252-day rolling-window specification, no systematic improvement is observed as network depth increases. Deeper architectures tend to exhibit similar or only marginally improved MSE values but do not achieve meaningful gains in R^2 , suggesting that additional model complexity does not translate into improved forecasting performance under shorter rolling estimation windows.

Time-Series Benchmark Models

Traditional time-series benchmarks, including AR(1) and GARCH(1,1) models, perform comparably under the short-horizon rolling window. Both models yield slightly negative mean R^2 values and MSE values close to those of Ridge regression and XGBoost. Their relatively

low dispersion across stocks indicates stable behavior, although their explanatory power remains limited.

The simple moving average (SMA) benchmark performs similarly in terms of MSE but fails to achieve positive R^2 values, confirming that naive averaging strategies offer limited predictive value when estimation is based on shorter rolling windows.

Summary of Short Rolling-Window Results

In contrast to the results obtained under the 252-day rolling-window specification, the 21-day rolling window reveals that volatility forecasting using shorter rolling estimation windows is substantially more challenging. Most models fail to consistently outperform the historical mean benchmark in terms of R^2 . Nevertheless, tree-based ensemble models, particularly Random Forest, achieve the lowest MSE values and exhibit the most stable performance across stocks. Regularized linear models and traditional time-series benchmarks perform moderately well, while neural networks and unregularized linear models consistently underperform under this rolling-window length.

4.1.3 Out-of-Sample Performance in Earlier versus Later Sample Periods (Stability Analysis)

The temporal stability of out-of-sample forecasting performance is examined by comparing model accuracy in the early and late portions of the evaluation period. For each rolling window specification, the out-of-sample period is divided into two equal subsamples, referred to as the early and late OOS periods. Performance is assessed using the out-of-sample coefficient of determination (R^2) and the mean squared error (MSE), aggregated across the 30 Dow Jones stocks.

Table 6: Early vs. Late Out-of-Sample Forecasting Performance (252-Day Rolling Window)

Model	Early R^2	Early MSE	Late R^2	Late MSE
OLS	-1.103	2.210	-0.642	1.340
Lasso	0.233	0.803	0.220	0.659
Ridge	0.251	0.779	0.224	0.652
ENet	0.256	0.778	0.235	0.645
RF	0.314	0.722	0.247	0.634
XGB1	0.295	0.752	0.227	0.658
XGB2	0.282	0.767	0.211	0.673
NN1	0.083	0.965	0.093	0.769
NN2	0.111	0.925	0.115	0.755
NN3	0.093	0.952	0.082	0.783
NN4	0.028	1.016	0.037	0.818
NN5	0.050	1.002	0.040	0.811
SMA	0	1.072	0	0.868
AR1	0.152	0.912	0.147	0.730
GARCH	0.153	0.903	0.149	0.735

The table reports average out-of-sample forecasting performance across 30 Dow Jones Industrial Average stocks. The out-of-sample period is split into two equal subsamples (early and late). R^2 denotes the out-of-sample coefficient of determination computed relative to a historical mean benchmark, and MSE denotes the mean squared error. Higher R^2 and lower MSE indicate better forecasting performance.

Long Rolling-Window Stability (252-Day Window)

Under the 252-day rolling window, model performance is relatively stable across the early and late OOS periods. Tree-based ensemble models retain strong predictive accuracy in both subsamples, with Random Forest achieving the highest R^2 and lowest MSE in each period. Although a modest decline in R^2 is observed in the late subsample for several models, relative performance rankings remain largely unchanged compared to the full-sample results reported in Section 4.1.1.

Regularized linear models, including Lasso, Ridge, and Elastic Net, also exhibit consistent performance across subsamples, with only limited variation in R^2 and MSE. In contrast, OLS continues to perform poorly in both periods, despite a slight improvement in the late subsample. Neural network models display weaker and less stable performance over time; while some specifications show small reductions in MSE in the late period, R^2 values remain low and heterogeneous. Time-series benchmarks, including AR(1) and GARCH, demonstrate stable but moderate performance across both subsamples.

Table 7: Early vs. Late Out-of-Sample Forecasting Performance (21-Day Rolling Window)

Model	Early R^2	Early MSE	Late R^2	Late MSE
OLS	-0.803	1.261	-0.579	1.029
Lasso	-0.782	1.260	-0.566	1.036
Ridge	-0.046	0.746	0.026	0.636
ENet	-0.520	1.074	-0.332	0.881
RF	0.007	0.712	0.028	0.635
XGB1	-0.014	0.732	0.007	0.656
XGB2	-0.007	0.729	0.001	0.659
NN1	-0.501	1.065	-0.347	0.886
NN2	-0.351	0.963	-0.197	0.788
NN3	-0.300	0.927	-0.163	0.767
NN4	-0.336	0.950	-0.174	0.775
NN5	-0.270	0.900	-0.145	0.757
SMA	0	0.718	0	0.662
AR1	-0.025	0.740	-0.019	0.672
GARCH	-0.031	0.738	-0.019	0.676

The table reports average out-of-sample forecasting performance across 30 Dow Jones Industrial Average stocks. The out-of-sample period is split into two equal subsamples (early and late). R^2 denotes the out-of-sample coefficient of determination computed relative to a historical mean benchmark, and MSE denotes the mean squared error. Higher R^2 and lower MSE indicate better forecasting performance.

Short Rolling-Window Stability (21-Day Window)

Under the 21-day rolling window, forecasting performance is generally weaker and more volatile across both subsamples. Most models record negative R^2 values in both the early and late periods, indicating limited explanatory power relative to the historical mean benchmark, consistent with the aggregate results obtained under the 21-day rolling-window specification documented in Section 4.1.2.

Tree-based ensemble models achieve the lowest MSE values and exhibit modest improvements in the late subsample, although gains in R^2 remain limited. Ridge regression and traditional time-series benchmarks also display improved MSE performance in the late period, suggesting relatively stable behavior compared to other linear and neural network models. Neural networks continue to exhibit limited stability, with negative R^2 values in both subsamples and only minor reductions in MSE over time. The simple moving average benchmark performs comparably to time-series models in terms of MSE but does not achieve positive R^2 values in either period.

Summary.

Overall, the early-versus-late analysis confirms that the model performance rankings identified in Sections 4.1.1 and 4.1.2 are broadly stable over time. Tree-based ensemble methods and regularized linear models exhibit the most consistent performance under the 252-day rolling-window specification, while neural networks and unregularized linear models exhibit greater variability in performance. Under the 21-day rolling-window specification, predictive accuracy deteriorates for most models across both subsamples, indicating that the difficulty of volatility forecasting using shorter rolling estimation windows is not driven by a specific subsample.

4.2 Feature Importance Results

This section examines feature-importance patterns for models that exhibit *relatively* strong and stable out-of-sample forecasting performance in Section 4.1 for the 21-day and 252-day rolling windows, considered separately. Feature importance for models with weak or unstable predictive accuracy (such as OLS and neural networks) is reported in the Appendix, as their importance measures are less informative in a forecasting context.

To account for heterogeneous feature sets arising from the data-cleaning procedure, feature importance is analyzed at the category level. As described in Section 3.6, importance is assessed using a selection-frequency approach, whereby features are grouped into eleven predefined economic categories. Category-level importance is measured by the frequency with which features from a given category are selected among the top three important predictors, aggregated across the 30 Dow Jones stocks.

Accordingly, the figures in this section reflect the relative prominence of broad information categories, rather than the magnitude of individual feature effects.

4.2.1 Feature Importance: 252-Day Window

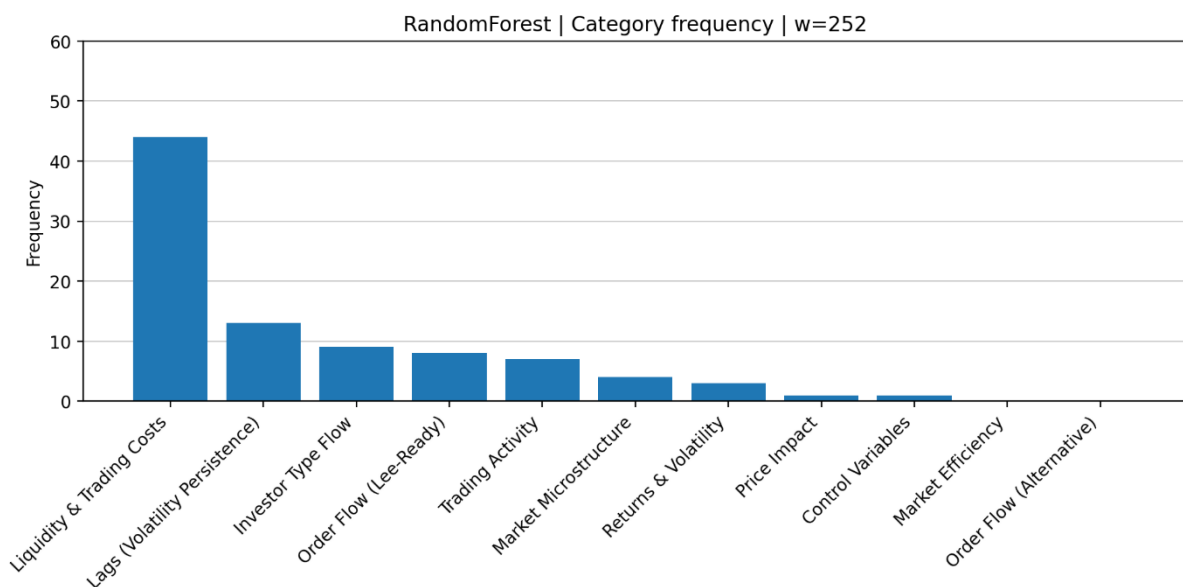
Figures 1-3 present category-level feature importance for the 252-day rolling window for Random Forest, XGBoost (Optuna-tuned), and Elastic Net, respectively.

Across all three models, liquidity and trading cost variables are the most frequently selected category, accounting for the largest share of top-ranked predictors. This indicates that measures related to bid–ask spreads, market depth, and execution costs are repeatedly identified as informative for volatility forecasting under a longer rolling estimation window within a rolling-window framework.

Investor-type flow variables and lagged volatility measures are also selected with high frequency. The prominence of lagged volatility is particularly pronounced for Elastic Net, consistent with the role of volatility persistence in linear specifications. In contrast, tree-based models select investor-type flow and order-flow-related variables more frequently than lagged volatility, suggesting that nonlinear interactions between trading behavior and liquidity conditions play a more prominent role in these models.

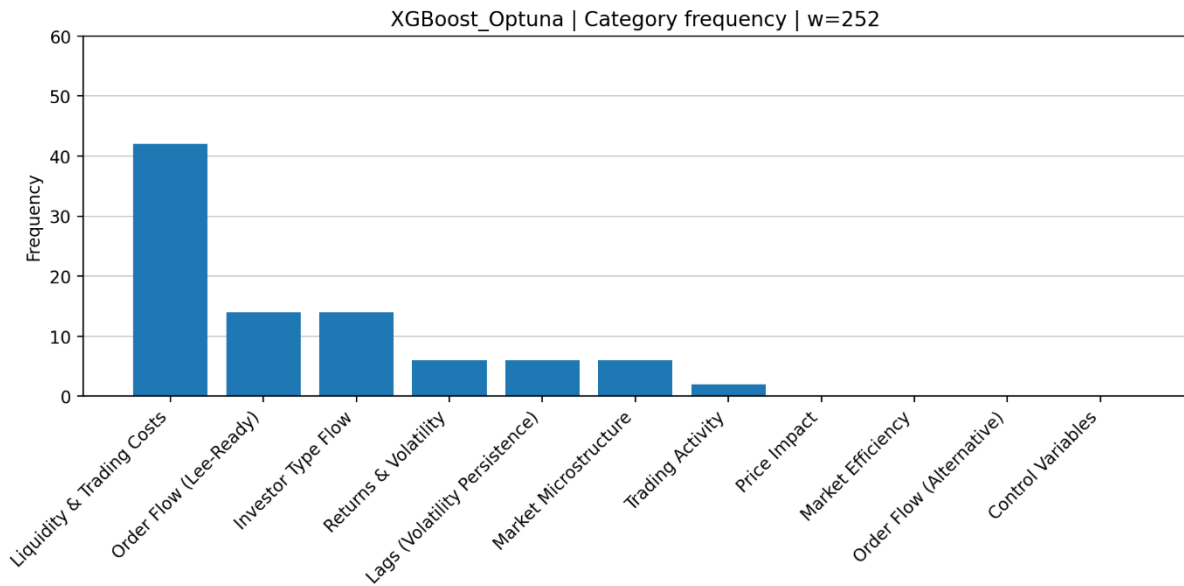
Control variables and Lee–Ready order-flow measures appear with moderate frequency across models, while categories related to trading activity, market microstructure, price impact, and market efficiency are selected less often. Overall, the results under the 252-day rolling-window specification exhibit a relatively stable ranking of information categories, with liquidity-related measures dominating, followed by investor flow and volatility persistence.

Figure 1: Category-Level Feature Importance for Random Forest (252-Day Rolling Window)



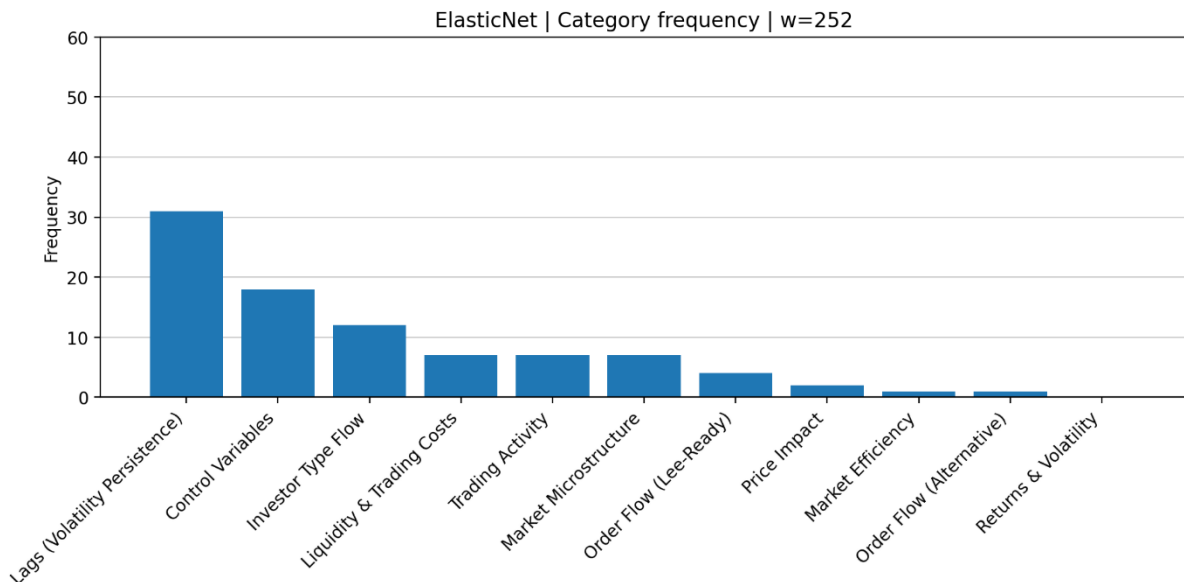
Feature importance is measured using impurity reduction. Bars report the frequency with which features from each category appear among the top three predictors, aggregated across 30 Dow Jones Industrial Average stocks.

Figure 2: Category-Level Feature Importance for XGBoost (252-Day Rolling Window)



Feature importance is measured using gain. Bars report the frequency with which features from each category appear among the top three predictors, aggregated across 30 Dow Jones Industrial Average stocks.

Figure 3: Category-Level Feature Importance for Elastic Net (252-Day Rolling Window)



Feature importance is based on absolute standardized regression coefficients. Bars report the frequency with which features from each category appear among the top three predictors, aggregated across 30 Dow Jones Industrial Average stocks.

4.2.2 Feature Importance: 21-Day Window

Figures 4-6 report category-level feature importance for the 21-day rolling window.

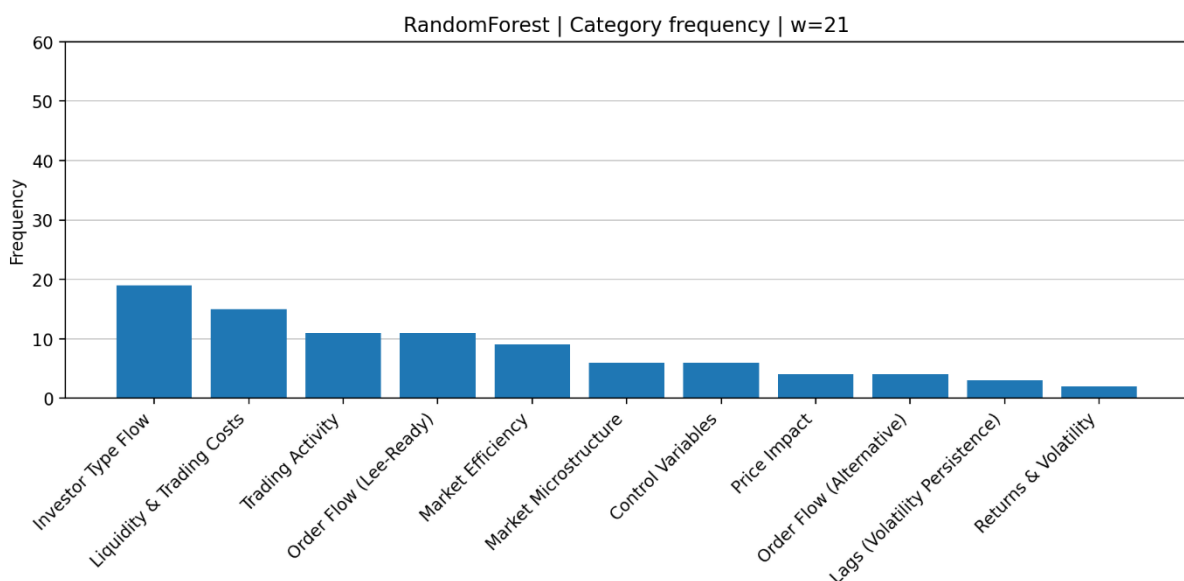
Relative to the 252-day rolling-window specification, feature-importance patterns are more dispersed and less concentrated, consistent with the weaker and less stable forecasting performance documented in Section 4.1.2. In contrast to the 252-day window, control variables emerge as the most frequently selected category, indicating a greater role for short-term timing and market-state information in volatility forecasting under shorter rolling estimation windows.

Investor-type flow and Lee–Ready order-flow measures are also selected frequently across models, highlighting the importance of short-term trading behavior for near-term volatility dynamics. Liquidity and trading cost variables remain relevant but are selected less often than under the longer rolling-window specification.

Lagged volatility variables play a more limited role under the 21-day rolling-window specification, particularly in tree-based models, reflecting the reduced relevance of volatility persistence over shorter estimation windows. Categories related to trading activity, market microstructure, price impact, and market efficiency exhibit moderate and relatively similar selection frequencies, with no single category consistently dominating across models.

Overall, the results obtained under the 21-day rolling-window specification point to greater heterogeneity in information usage and reduced stability of importance rankings, underscoring the inherent difficulty of volatility forecasting when estimation relies on shorter rolling windows.

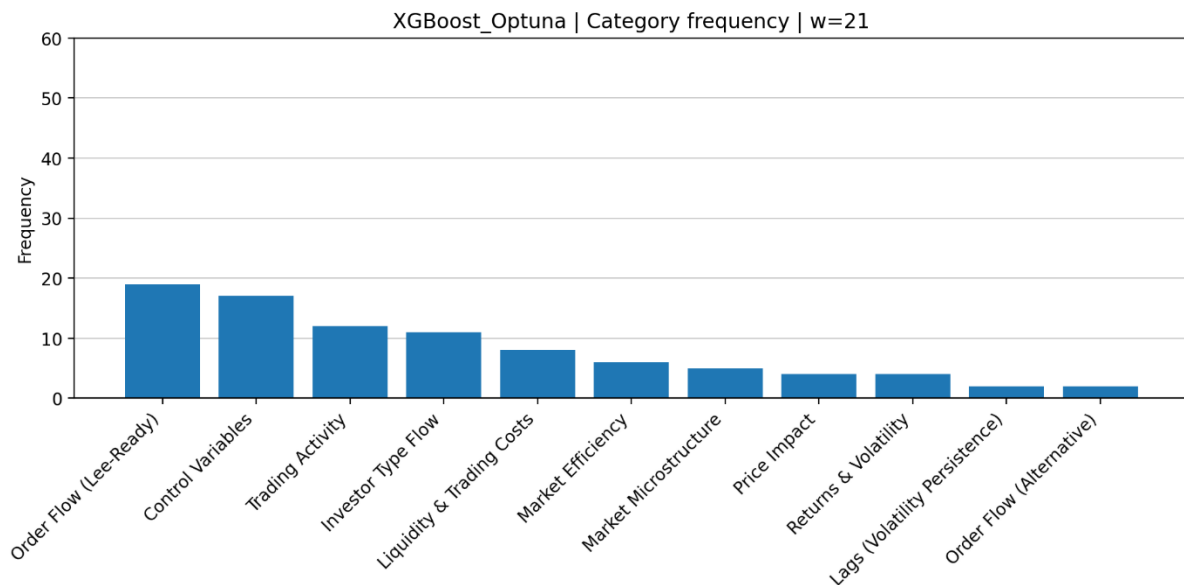
Figure 4: Category-Level Feature Importance for Random Forest (21-Day Rolling Window)



Feature importance is based on impurity reduction. Bars report the frequency with which features from

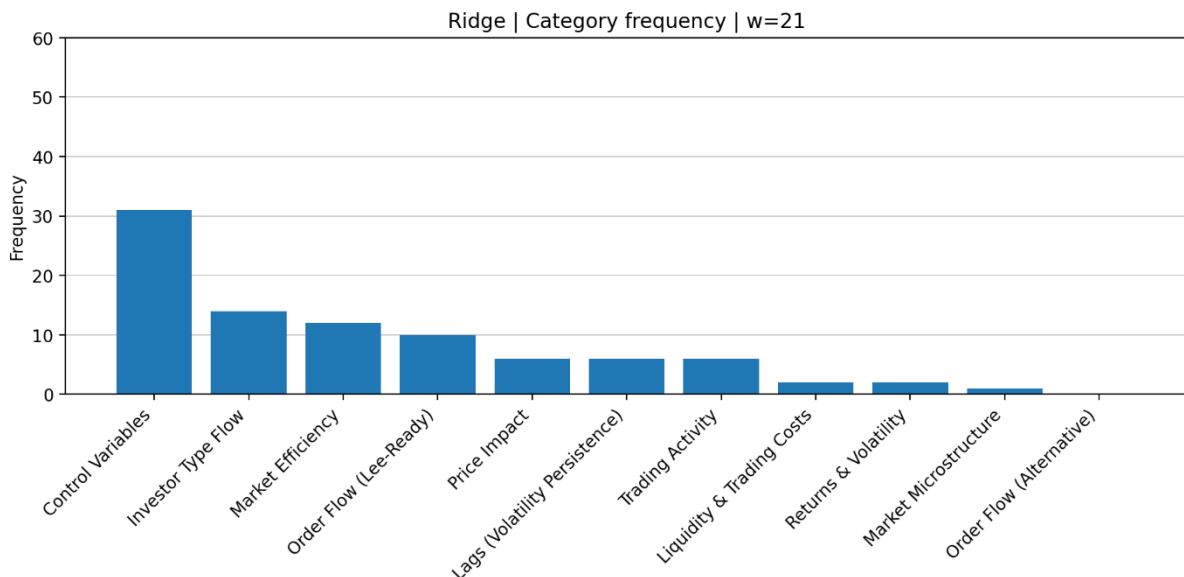
each category appear among the top three predictors, aggregated across the 30 constituent stocks of the Dow Jones.

Figure 5: Category-Level Feature Importance for XGBoost (Optuna-Tuned, 21-Day Rolling Window)



Feature importance is measured using gain. Bars report the frequency with which features from each category appear among the top three predictors, aggregated across the 30 constituent stocks of the Dow Jones.

Figure 6: Category-Level Feature Importance for Ridge Regression (21-Day Rolling Window)



Feature importance is based on absolute standardized regression coefficients. Bars report the frequency with which features from each category appear among the top three predictors, aggregated across the 30 constituent stocks of the Dow Jones.

4.3 Results Relative to the Hypotheses

This section evaluates the empirical findings presented in Sections 4.1 and 4.2 in relation to the research hypotheses stated in Section 3.0. All conclusions are based on strictly out-of-sample performance under the rolling-window forecasting framework.

4.3.1 H1: Regularization Effect

Hypothesis: Penalized linear models outperform OLS due to severe multicollinearity among intraday predictors.

Assessment: Supported

Across both rolling-window specifications, penalized linear models (Lasso, Ridge, and Elastic Net) outperform OLS in terms of both out-of-sample R^2 and MSE. Under the 252-day window, all regularized models achieve positive mean R^2 values, while OLS exhibits strongly negative performance and high cross-sectional dispersion. Under the 21-day window, although overall predictive accuracy deteriorates, Ridge and Elastic Net remain substantially more stable than OLS.

These results confirm that regularization is essential when forecasting volatility using high-dimensional and highly correlated microstructure predictors. Shrinkage improves numerical stability and generalization, particularly under rolling estimation.

4.3.2 H2: Nonlinearity

Hypothesis: Tree-based models outperform linear regressions in out-of-sample forecasting because they capture nonlinearities and interaction effects.

Assessment: Supported, with stronger evidence under longer estimation windows

Tree-based ensemble models (Random Forest and XGBoost) deliver the strongest overall forecasting performance under the 252-day rolling window, achieving the highest mean R^2 values and lowest MSE across assets. Their performance advantage is robust relative to both OLS and regularized linear models, indicating that nonlinear modeling provides meaningful gains when sufficient estimation history is available.

Under the 21-day rolling window, tree-based models continue to rank among the best-performing specifications. Random Forest achieves the highest mean R^2 among all models, while XGBoost performs comparably to Ridge regression and clearly outperforms

unregularized and penalized linear alternatives. However, overall R^2 values are close to zero and performance differences across top models are small, reflecting the difficulty of volatility forecasting when estimation relies on shorter rolling windows.

Taken together, the evidence supports the hypothesis that nonlinear models offer superior out-of-sample forecasting performance. At the same time, the magnitude and stability of this advantage diminish when estimation windows are short, suggesting that the benefits of modeling nonlinearities depend critically on the availability of sufficient historical data.

4.3.3 H3: Model Capacity

Hypothesis: Neural networks provide additional forecasting gains only if volatility dynamics exhibit nonlinear structure beyond that captured by tree-based models.

Assessment: Not supported

Across both rolling-window specifications, neural network models do not consistently outperform tree-based ensembles or regularized linear regressions. While some neural network configurations achieve modest improvements over OLS, their performance remains inferior to Random Forest and XGBoost in terms of both accuracy and stability.

Increasing network depth does not yield systematic gains in forecasting performance, with results varying substantially across assets and specifications. These findings indicate that the nonlinear structure present in the data are largely captured by tree-based models, and that the additional representational capacity offered by neural networks does not translate into robust forecasting improvements in this setting. This pattern is consistent with a bias–variance trade-off, whereby increased model flexibility may lead to higher estimation variance in rolling-window environments with limited effective sample sizes.

4.3.4 H4: Estimation-Window Length

Hypothesis: Forecasting performance is sensitive to rolling-window length, with shorter windows leading to less stable parameter estimates and reduced out-of-sample accuracy.

Assessment: Strongly supported

Forecasting performance deteriorates substantially when the rolling window is reduced from 252 days to 21 days. Across all model classes, mean R^2 values decline and frequently become negative, and performance rankings become less stable across assets.

While tree-based models and Ridge regression retain relatively better MSE performance under the shorter rolling-window specification, overall predictive accuracy is markedly lower. Feature-importance results further show increased dispersion and reduced dominance of economically meaningful categories when estimation relies on shorter rolling windows.

These findings confirm that estimation-window length plays a critical role in volatility forecasting, and that high-capacity models are particularly sensitive to limited training data.

Section 5: Limitations

The empirical design of this study entails several limitations that should be acknowledged when interpreting the findings.

Data-related limitations

The target variable is a trade-based realized volatility measure computed from millisecond TAQ data during regular trading hours. Alternative estimators, such as realized kernels, pre-averaged volatility, bipower variation, or overnight volatility, are not examined. As a result, findings may not generalize volatility definitions that incorporate microstructure-noise corrections or overnight price jumps.

The feature set is restricted to the 198 WRDS Millisecond Intraday Indicators. No technical indicators (e.g., EMA, SMA, ADX, Aroon), oscillators, liquidity–volatility hybrid measures, or pattern-based variables (e.g., day-of-week, month effects, macro announcements) are included. While this deliberate restriction keeps the design focused on exchange microstructure data, it may leave out economically meaningful predictors.

Rolling windows of 21 and 252 days are used uniformly across models, precluding analysis of alternative or adaptive window specifications. Other window lengths, adaptive windows, or expanding-window estimation could produce different results but are not explored.

Finally, structural breaks, such as the introduction of Reg NMS, the rise of high-frequency trading, COVID-19 dislocations, or shifts in liquidity regimes, may affect model stability. Although the rolling-window framework partially mitigates this issue, it does not explicitly account for regime shifts or structural parameter instability.

Model-related limitations

First, the set of forecasting models is restricted to linear and penalized regressions, tree-based ensembles, and fully connected neural networks of limited depth. More specialized architectures, such as LSTM/GRU networks, temporal convolutional networks, attention-based models, or hybrid volatility models, are not considered. These omitted classes may capture temporal dependencies or structural nonlinearities that are not explicitly modeled in the present framework.

Second, hyperparameters are tuned once on the in-sample period and held fixed throughout the rolling out-of-sample procedure. While this approach prevents look-ahead bias and maintains computational feasibility, it limits the ability of models to adjust complexity following structural breaks or shifts in market microstructure.

Third, the forecast horizon is restricted to one-step-ahead predictions. Multi-step volatility forecasts, which are relevant for risk management, margining, and derivatives pricing, are not evaluated. This restricts conclusions to the short-run predictability of daily volatility.

Fourth, despite the rolling-window design, overfitting remains a potential concern, particularly for nonlinear models estimated on short windows and re-initialized repeatedly. Regularization, cross-validated hyperparameter selection, and early stopping reduce, but do not eliminate this risk.

Panel-data limitation

This study adopts a purely time-series forecasting framework in which models are estimated separately for each asset. As a result, potential efficiency gains from exploiting the cross-sectional dimension of realized volatility are not considered. Recent work shows that panel-based machine learning models can improve short-horizon volatility forecasts by pooling information across assets, particularly when time-series samples are short (e.g., Zhu et al., 2023).

Incorporating a panel-data perspective could stabilize parameter estimation, mitigate small-sample issues in shorter rolling windows, and potentially enhance out-of-sample performance. Exploring such cross-sectional pooling strategies is left for future research.

Feature-importance limitations

Because predictors are selected separately for each stock after removing highly correlated variables, the resulting feature sets are not identical across assets. This prevents meaningful cross-stock averaging of feature importance measures at the individual variable level. To partially address this limitation, feature importance is aggregated and reported at the category level (as detailed in Section 3.6). This approach allows for a comparable global assessment of information drivers across the 30 Dow Jones stocks despite heterogeneous feature sets.

Evaluation-metric limitations

Forecast evaluation is based solely on Mean Squared Error (MSE) and out-of-sample R^2 relative to a rolling-mean benchmark. These metrics assess statistical accuracy but do not capture economic value, such as:

- risk-based utility gains,
- Value-at-Risk or Expected Shortfall accuracy,
- trading profitability,
- hedging efficiency.

Thus, conclusions pertain strictly to statistical forecast performance rather than economic optimality.

Advanced econometric benchmark limitations

The set of benchmark models is limited to simple time-series specifications, including SMA, AR(1), and GARCH(1,1). More advanced econometric volatility models, such as HAR-type models, realized GARCH specifications, or long-memory processes, are not included. While these models have demonstrated strong performance in prior studies, incorporating them would introduce additional modeling and tuning choices beyond the scope of this comparative analysis. Consequently, the results should be interpreted relative to standard baseline benchmarks rather than to the full class of state-of-the-art econometric volatility models.

Overall assessment

Overall, the limitations discussed above delineate the scope of inference of this study. The analysis evaluates the relative statistical forecasting performance of several model classes under a strictly out-of-sample, real-time protocol using a specific high-frequency realized-volatility measure. The results should therefore be interpreted as evidence on comparative

predictive accuracy within this framework, rather than as structural statements about volatility dynamics, causality, or economic value.

Section 6: Discussion and Conclusion

This section interprets the empirical findings of the realized-volatility forecasting exercise and situates them within a broader economic and methodological context. Building on the strictly out-of-sample results reported in Section 4, the discussion focuses on how predictability varies across rolling-window lengths, why certain model classes outperform others, and how feature-importance patterns help explain observed performance differences. Particular attention is paid to the interaction between rolling-window length, model complexity, and market microstructure information. The section concludes by outlining practical implications and directions for future research.

6.1 Rolling-Window-Length-Dependent Predictability

A central finding of this study is the strong dependence of realized-volatility predictability on the length of the estimation window. Forecasts based on the 252-day rolling window consistently outperform those based on the 21-day window across all model classes. Longer windows smooth idiosyncratic shocks and microstructure noise inherent in high-frequency volatility measures, allowing persistent components of volatility to be estimated more reliably.

In contrast, volatility estimated using shorter rolling windows is dominated by transient shocks and measurement noise, substantially reducing the signal-to-noise ratio available for forecasting. This is reflected in the empirical results: under the 252-day window, several models achieve positive and stable out-of-sample R^2 values, whereas under the 21-day window, predictive performance deteriorates sharply, with R^2 values close to zero or negative for most specifications. Volatility forecasting based on shorter rolling estimation windows therefore appears inherently more difficult, even when flexible nonlinear models are employed.

6.2 Model Class Comparison

6.2.1 Tree-Based Models

Tree-based ensemble models, particularly Random Forest and XGBoost, deliver the strongest and most robust forecasting performance in this study. Their advantage is most pronounced under the 252-day rolling window, where they consistently achieve the highest out-of-sample R^2 values and lowest MSE across assets.

This performance reflects the ability of tree-based models to capture nonlinearities and interaction effects among high-dimensional intraday predictors while remaining robust in a rolling-window estimation framework. Although gradient boosting is often expected to outperform bagging-based methods, Random Forest slightly dominates XGBoost in this application. A plausible explanation is that Random Forest's stronger averaging mechanism provides greater robustness to noise and regime variation, whereas boosting methods may be more sensitive to short-lived patterns in high-frequency financial data.

6.2.2 Regularized Linear Models

Regularized linear models, particularly Ridge and Elastic Net, perform consistently well across both estimation horizons and substantially outperform unregularized OLS. Shrinkage acts as an effective noise-filtering mechanism in the presence of severe multicollinearity among intraday predictors, stabilizing coefficient estimates and improving generalization. These results indicate that a substantial portion of the predictive signal in realized volatility can be captured through linear relationships once parameter uncertainty is appropriately controlled.

6.2.3 Neural Networks

Feed-forward neural networks do not deliver additional forecasting gains relative to tree-based models or regularized linear regressions. Across both rolling-window specifications, their performance remains inferior in terms of both accuracy and stability, and increasing network depth does not yield systematic improvements.

The results are consistent with a bias–variance tradeoff: neural networks require larger samples to perform reliably, and in short rolling-window settings, frequent retraining increases estimation variance. As a result, deeper networks do not yield robust out-of-sample gains, as tree-based models already capture most of the relevant nonlinear structure in the data.

6.2.4 Time-Series Benchmarks

Traditional time-series benchmarks such as AR(1) and GARCH remain competitive relative to naive baselines, reflecting their ability to capture volatility persistence. However, their inferior

performance compared to machine-learning models indicates that they fail to exploit richer nonlinear and microstructure-based information embedded in high-frequency data.

6.3 Temporal Stability of Forecasting Performance

The early-versus-late out-of-sample analysis indicates that model performance rankings are broadly stable over time, particularly under the 252-day rolling window. Tree-based models maintain their relative advantage across subsamples, suggesting that their superior performance reflects persistent predictive relationships rather than transient market conditions. Neural networks, by contrast, exhibit greater instability, reinforcing concerns about sensitivity to regime variation and limited effective sample sizes.

6.4 Economic Interpretation of Feature Importance

Feature-importance analysis provides insight into the economic mechanisms underlying model performance. Under longer rolling-window specifications, liquidity and trading-cost variables dominate across models, indicating that persistent market frictions, such as bid–ask spreads, market depth, and execution costs, play a central role in shaping realized-volatility dynamics. Under shorter rolling-window specifications, importance shifts toward control variables and order-flow measures, reflecting the influence of transient trading behavior and short-term market-state conditions. The dispersion of importance rankings under the 21-day window mirrors the weaker and less stable forecasting performance observed under this rolling-window length, reinforcing the view that short-term realized volatility is driven by noisy and rapidly evolving factors.

6.5 Implications and Concluding Remarks

From a practical perspective, the results suggest that tree-based ensemble models, in particular Random Forest, are well suited among the approaches evaluated here for realized-volatility forecasting when sufficient historical data are available, while regularized linear models offer a robust and interpretable alternative. Volatility forecasts based on shorter rolling estimation windows should be interpreted with caution, as predictive gains over simple benchmarks remain limited.

Overall, this thesis demonstrates that the effectiveness of machine-learning models in realized-volatility forecasting depends critically on the interaction between model complexity, rolling-window length, and data structure. The results should therefore be interpreted as evidence on

comparative predictive accuracy within the study’s benchmark set and strictly out-of-sample rolling-window framework, rather than as structural statements about volatility dynamics, causality, economic value, or dominance over the broader realized-volatility forecasting literature. Future research may explore alternative neural-network architectures, regime-adaptive estimation schemes, or panel-based approaches to further improve volatility forecasting under shorter rolling estimation windows.

References

- Aas, K., Jullum, M., & Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, 298, 103502.
- Aït-Sahalia, Y., Mykland, P. A., & Zhang, L. (2005). How often to sample a continuous-time process in the presence of market microstructure noise. *Review of Financial Studies*, 18(2), 351–416.
- Andersen, T. G. (1996). Return volatility and trading volume: An information flow interpretation of stochastic volatility. *Journal of Finance*, 51(1), 169–204.
- Andersen, T. G., & Bollerslev, T. (1998). Answering the skeptics: Yes, standard volatility models do provide accurate forecasts. *International Economic Review*, 39(4), 885–905.
- Andersen, T. G., Bollerslev, T., Diebold, F. X., & Ebens, H. (2001). The distribution of realized stock return volatility. *Journal of Financial Economics*, 61(1), 43–76.
- Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica*, 71(2), 579–625.
- Audrino, F., & Chassot, L. (2024). HARd to beat: The overlooked impact of rolling windows in the era of machine learning (arXiv:2406.08041). *arXiv*.
- Barndorff-Nielsen, O. E., & Shephard, N. (2002). Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(2), 253–280.
- Basel Committee on Banking Supervision. (2019). *Minimum capital requirements for market risk*. Bank for International Settlements.

Basel Committee on Banking Supervision, & International Organization of Securities Commissions. (2015). *Margin requirements for non-centrally cleared derivatives*. Bank for International Settlements / IOSCO.

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3), 637–654.

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

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

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

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM.

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

Christoffersen, P. F. (1998). Evaluating interval forecasts. *International Economic Review*, 39(4), 841–862.

CME Group. (2019, March 1). *CME SPAN® – Standard Portfolio Analysis of Risk: Methodology*. CME Group.

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

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4), 303–314.

De Mol, C., Giannone, D., & Reichlin, L. (2008). Forecasting using a large number of predictors: Is Bayesian regression a valid alternative to principal components? *Journal of Econometrics*, 146(2), 318–328.

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4), 987–1008.

Engle, R. F., & Ng, V. K. (1993). Measuring and testing the impact of news on volatility. *Journal of Finance*, 48(5), 1749–1778.

- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232.
- Glosten, L. R., & Milgrom, P. R. (1985). Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of Financial Economics*, 14(1), 71–100.
- Goyal, A., & Welch, I. (2008). A comprehensive look at the empirical performance of equity premium prediction. *Review of Financial Studies*, 21(4), 1455–1508.
- Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies*, 33(5), 2223–2273.
- Hasbrouck, J. (1991). Measuring the information content of stock trades. *Journal of Finance*, 46(1), 179–207.
- Hasbrouck, J., & Saar, G. (2013). Low-latency trading. *Journal of Financial Markets*, 16(4), 646–679.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.
- Hautsch, N. (2012). *Econometrics of financial high-frequency data*. Springer.
- Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55–67.
- Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2), 251–257.
- Inoue, A., Jin, L., & Rossi, B. (2017). Rolling window selection for out-of-sample forecasting with time-varying parameters. *Journal of Econometrics*, 196(1), 55–67.
- Jorion, P. (2006). *Value at risk: The new benchmark for managing financial risk* (3rd ed.). McGraw-Hill.
- Karpoff, J. M. (1987). The relation between price changes and trading volume: A survey. *Journal of Financial and Quantitative Analysis*, 22(1), 109–126.
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

- Kılıç, R. (2025). *Linear and nonlinear econometric models against machine learning models: Realized volatility prediction* (Finance and Economics Discussion Series). Board of Governors of the Federal Reserve System.
- Krogh, A., & Hertz, J. A. (1992). A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems* (pp. 950–957).
- Kyle, A. S. (1985). Continuous auctions and insider trading. *Econometrica*, 53(6), 1315–1335.
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (pp. 4765–4774).
- Markowitz, H. (1952). Portfolio selection. *Journal of Finance*, 7(1), 77–91.
- Menkveld, A. J. (2013). High frequency trading and the new market makers. *Journal of Financial Markets*, 16(4), 712–740.
- Molnar, C. (2019). *Interpretable machine learning: A guide for making black box models explainable*. Retrieved from <https://christophm.github.io/interpretable-ml-book/>
- O’Hara, M. (2015). High frequency market microstructure. *Journal of Financial Economics*, 116(2), 257–270.
- O’Hara, M., & Ye, M. (2011). Is market fragmentation harming market quality? *Journal of Financial Economics*, 100(3), 459–474.
- Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning* (pp. 1310–1318).
- Patton, A. J. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, 160(1), 246–256.
- Poon, S.-H., & Granger, C. W. J. (2003). Forecasting volatility in financial markets: A review. *Journal of Economic Literature*, 41(2), 478–539.
- Rapach, D. E., & Zhou, G. (2013). Forecasting stock returns. In G. Elliott & A. Timmermann (Eds.), *Handbook of economic forecasting* (Vol. 2A, pp. 328–383). Elsevier.
- Rossi, B., & Inoue, A. (2012). Out-of-sample forecast tests robust to the choice of window size. *Journal of Business & Economic Statistics*, 30(3), 432–453.

- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.
- Tauchen, G. E., & Pitts, M. (1983). The price variability–volume relationship on speculative markets. *Econometrica*, 51(2), 485–505.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288.
- Wharton Research Data Services. (n.d.). *WRDS DTAQ IID manual (Version 2.0)*. https://wrds-www.wharton.upenn.edu/documents/1038/WRDS_DTAQ_IID_Manual_2.0_vzJomVO.pdf
- Zou, H., & 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.
- Zhu, H., Bai, L., He, L., & Liu, Z. (2023). Forecasting realized volatility with machine learning: Panel data perspective. *Journal of Empirical Finance*, 73, 251–271.

Appendix A — Variable dictionary

This appendix lists all variables used in the empirical analysis, grouped by feature category as described in Section 3.6. Variable definitions, construction procedures, and units of measurement are standardized and follow the WRDS Intraday Indicators (IID) documentation referenced in the main text. To avoid redundancy and maintain consistency with established market microstructure measures, individual variable definitions are not repeated in this appendix.

1. Liquidity & Trading Costs:

QuotedSpread_Dollar_tw, QuotedSpread_Percent_tw, BestOfrDepth_Dollar_tw,
BestBidDepth_Dollar_tw, BestOfrDepth_Share_tw, BestBidDepth_Share_tw,
EffectiveSpread_Dollar_Ave, EffectiveSpread_Percent_Ave,
EffectiveSpread_Dollar_DW, EffectiveSpread_Dollar_SW,
EffectiveSpread_Percent_DW, EffectiveSpread_Percent_SW,
DollarRealizedSpread_LR_Ave, PercentRealizedSpread_LR_Ave,
DollarRealizedSpread_LR_SW, DollarRealizedSpread_LR_DW,
PercentRealizedSpread_LR_SW, PercentRealizedSpread_LR_DW.

2. Price Impact:

DollarPriceImpact_LR_Ave, PercentPriceImpact_LR_Ave,
DollarPriceImpact_LR_SW, DollarPriceImpact_LR_DW,
PercentPriceImpact_LR_SW, PercentPriceImpact_LR_DW, TSignSqrtDVol1,
TSignSqrtDVol2.

3. Market Efficiency:

var_ratio1, var_ratio2, var_ratio3, var_ratio4, var_ratio5.

4. Returns & Volatility:

ivol_t, ivol_q, ret_mkt_m, price_low_m, price_high_m, price_low_b, price_high_b,
price_low_a, price_high_a.

5. Volume & Trading Activity:

avg_price_m, vw_price_m, total_vol_m, total_dollar_m, total_n_trades_m,
avg_price_b, vw_price_b, total_vol_b, total_dollar_b, total_n_trades_b, avg_price_a,
vw_price_a, total_vol_a, total_dollar_a, total_n_trades_a, n5_pos, n30_pos, n_obs.

6. Market Microstructure:

HIndex, n_iso_trade, iso_vol, iso_dollar, n_oddlot_trade, oddlot_vol, oddlot_dollar,
n_outside_nbbo_trade.

7. Order Flow (Lee–Ready):

BuyNumTrades_LR, SellNumTrades_LR, total_trade, BuyVol_LR, SellVol_LR,
total_vol, buy_dv_LR, sell_dv_LR, total_dv_LR, avg_buy_price_LR,
avg_sell_price_LR, vwavg_buy_price_LR, vwavg_sell_price_LR.

8. Order Flow (Alternative Methods):

BuyNumTrades_tick, SellNumTrades_tick, total_trade_tick, BuyVol_tick,
SellVol_tick, total_vol_tick, buy_dv_tick, sell_dv_tick, total_dv_tick,
avg_buy_price_tick, avg_sell_price_tick, vwavg_buy_price_tick,
vwavg_sell_price_tick, BuyNumTrades_wrds, SellNumTrades_wrds,
total_trade_wrds, BuyVol_wrds, SellVol_wrds, total_vol_wrds, buy_dv_wrds,
sell_dv_wrds, total_dv_wrds, avg_buy_price_wrds, avg_sell_price_wrds,
vwavg_buy_price_wrds, vwavg_sell_price_wrds.

9. Investor Type Flow:

BuyNumTrades_Retail, SellNumTrades_Retail, total_trade_retail, BuyVol_Retail,
SellVol_Retail, total_vol_retail, buy_dv_Retail, sell_dv_Retail, total_dv_retail,
avg_buy_price_Retail, avg_sell_price_Retail, bs_ratio_retail_num,
bs_ratio_retail_vol, BuyNumTrades_Inst20k, SellNumTrades_Inst20k,
total_trade_Inst20k, BuyVol_Inst20k, SellVol_Inst20k, total_vol_Inst20k,
buy_dv_Inst20k, sell_dv_Inst20k, total_dv_Inst20k, avg_buy_price_Inst20k,
avg_sell_price_Inst20k, bs_ratio_Inst20k_num, bs_ratio_Inst20k_vol,
BuyNumTrades_Inst50k, SellNumTrades_Inst50k, total_trade_Inst50k,
BuyVol_Inst50k, SellVol_Inst50k, total_vol_Inst50k, buy_dv_Inst50k,
sell_dv_Inst50k, total_dv_Inst50k, avg_buy_price_Inst50k, avg_sell_price_Inst50k,
bs_ratio_Inst50k_num, bs_ratio_Inst50k_vol, bs_ratio_num, bs_ratio_vol.

10. Control Variables:

SYM_ROOT, SYM_SUFFIX, symbol, OTime, OEX, OSize, OPrc,
TR_SEQNUM_open, O_official, CTime, CEX, CSize, CPrc, TR_SEQNUM_close,
C_official, size_1pm, ttime_1pm, ptime_1pm, size_4pm, ttime_4pm, ptime_4pm,
stime_open, ttime_open, ptime_open, stime_close, ttime_close, ptime_close,
nbbot_1pm, nbo_1pm, nbb_1pm, nboqty_1pm, nbbqty_1pm, mid_1pm, nbbot_4pm,
nbo_4pm, nbb_4pm, nboqty_4pm, nbbqty_4pm, mid_4pm, nbbot_after_open,
nbo_after_open, nbb_after_open, nboqty_after_open, nbbqty_after_open,
mid_after_open, nbbot_before_close, nbo_before_close, nbb_before_close,
nboqty_before_close, nbbqty_before_close, mid_before_close.

11. Lags:

ivol_lag1, ivol_lag2, ivol_lag3, ivol_lag4, ivol_lag5.

Appendix B — Feature Importance Results

This appendix reports category-level feature importance for models that exhibit weak or unstable out-of-sample forecasting performance and are therefore excluded from the main results section. As discussed in Section 4.2, these models include linear benchmarks (OLS, Lasso, Elastic Net) and neural network specifications.

B.1 Feature Importance — 252-Day Rolling Window

Figure B1: Category-level feature importance for XGBoost (randomized-search tuned, 252-day rolling window). Feature importance is measured using gain.

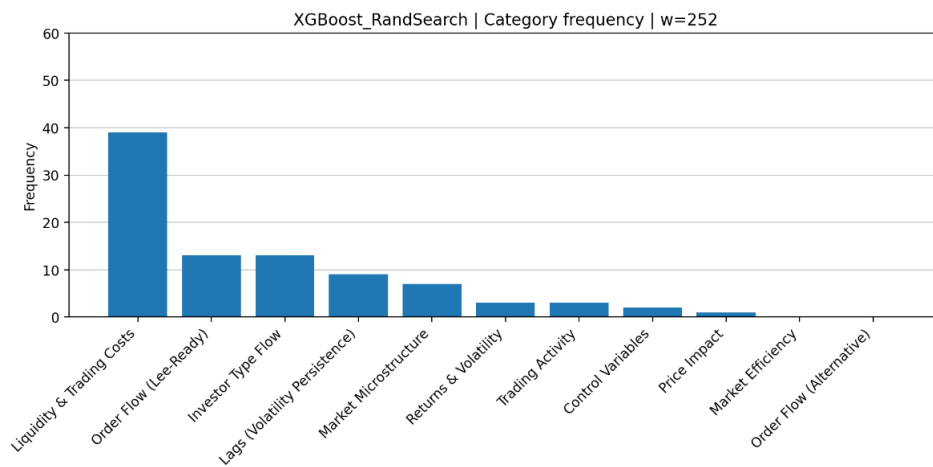


Figure B2: Category-level feature importance for OLS (252-day rolling window). Feature importance is based on absolute standardized coefficients.

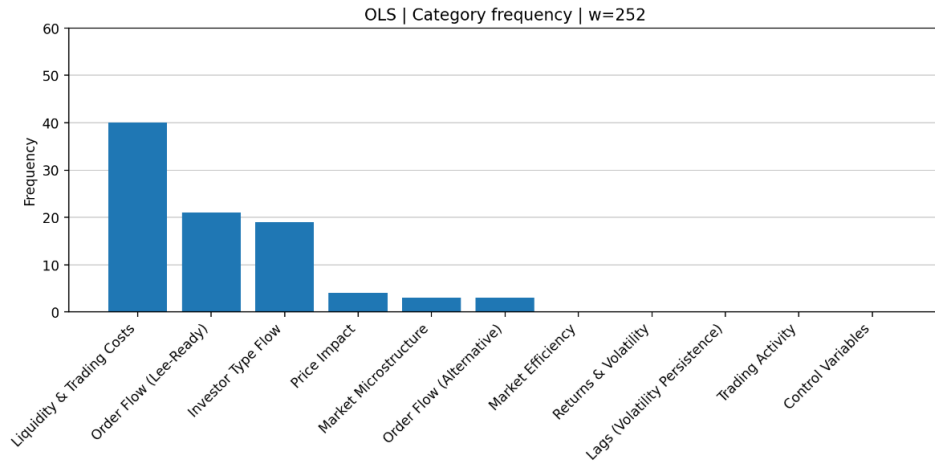


Figure B3: Category-level feature importance for Lasso (252-day rolling window). Feature importance is based on absolute standardized coefficients.

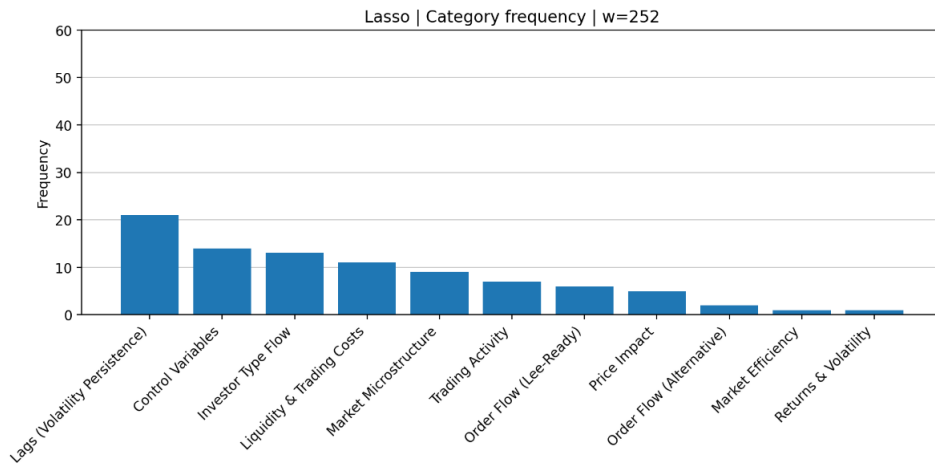


Figure B4: Category-level feature importance for Ridge (252-day rolling window). Feature importance is based on absolute standardized coefficients.

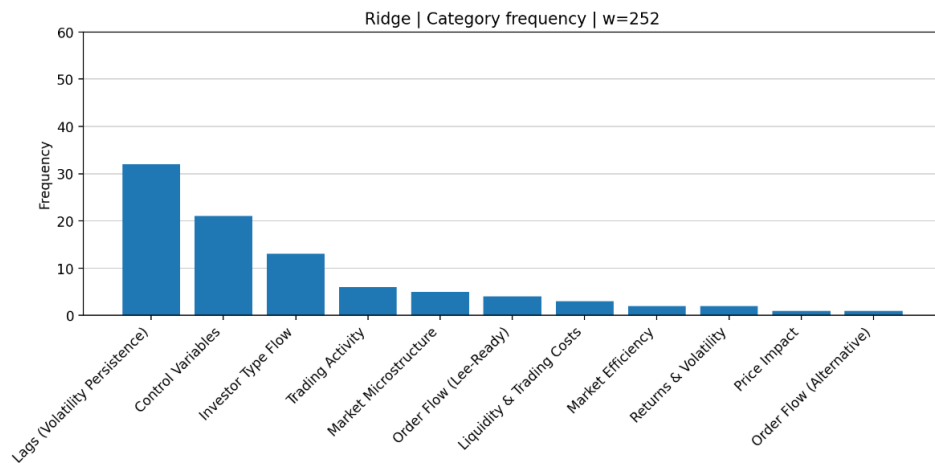


Figure B5: Category-level feature importance for Neural Network 1 (252-day rolling window). Feature importance is based on ablation-induced changes in squared forecast error.

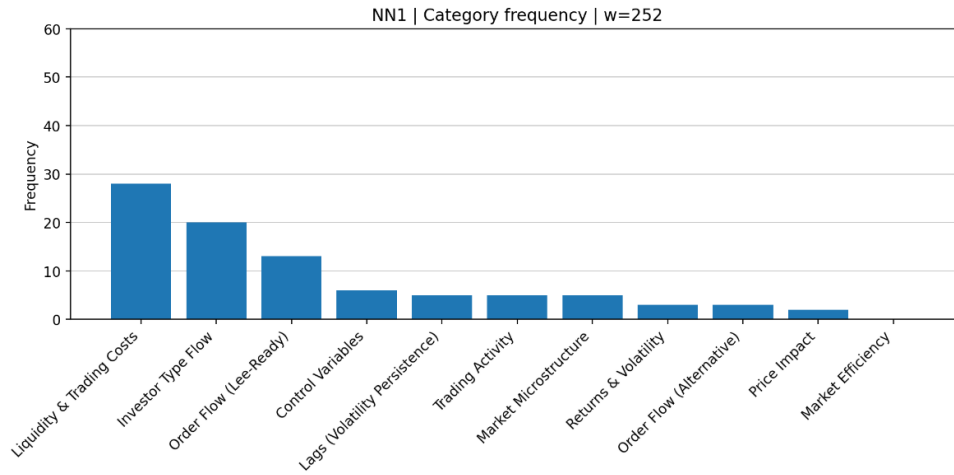


Figure B6: Category-level feature importance for Neural Network 2 (252-day rolling window). Feature importance is based on ablation-induced changes in squared forecast error.

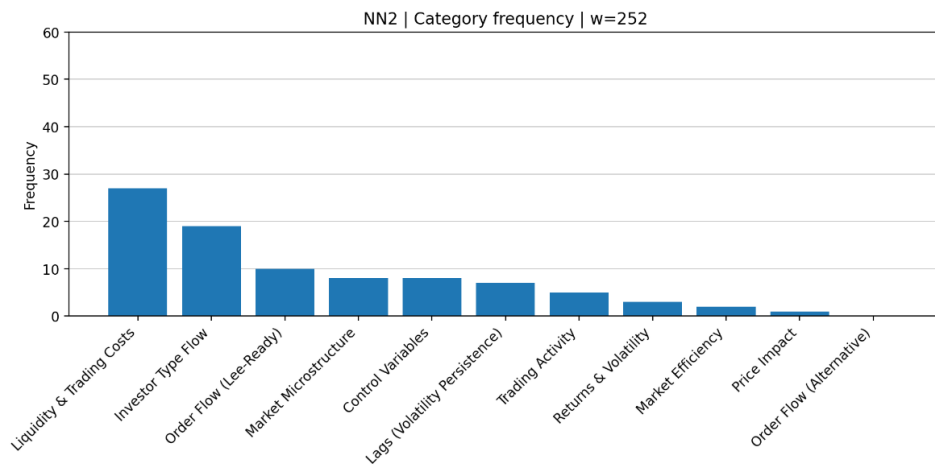


Figure B7: Category-level feature importance for Neural Network 3 (252-day rolling window). Feature importance is based on ablation-induced changes in squared forecast error.

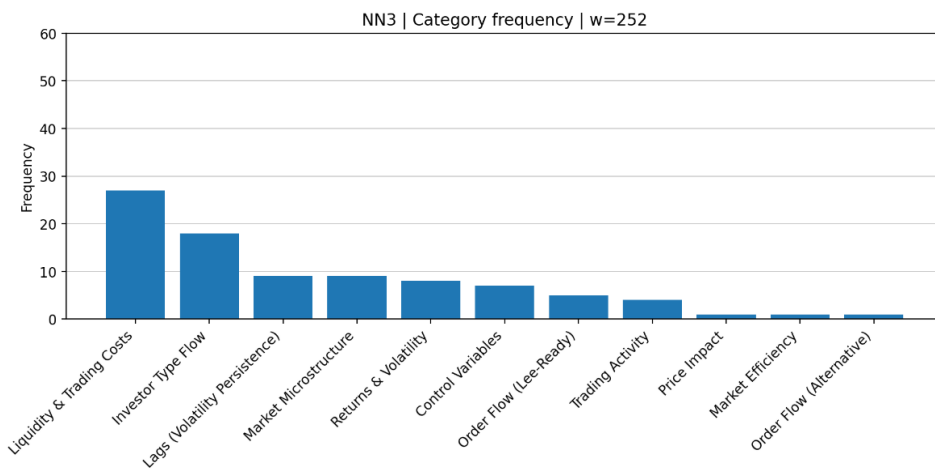


Figure B8: Category-level feature importance for Neural Network 4 (252-day rolling window). Feature importance is based on ablation-induced changes in squared forecast error.

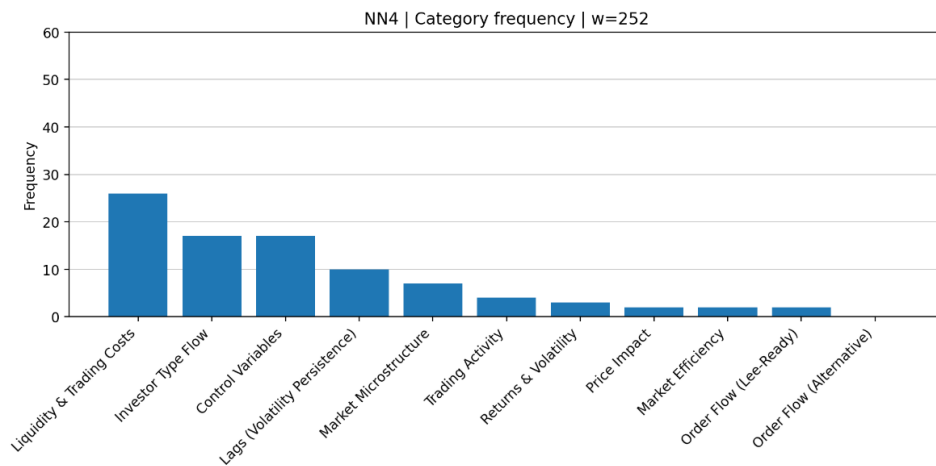
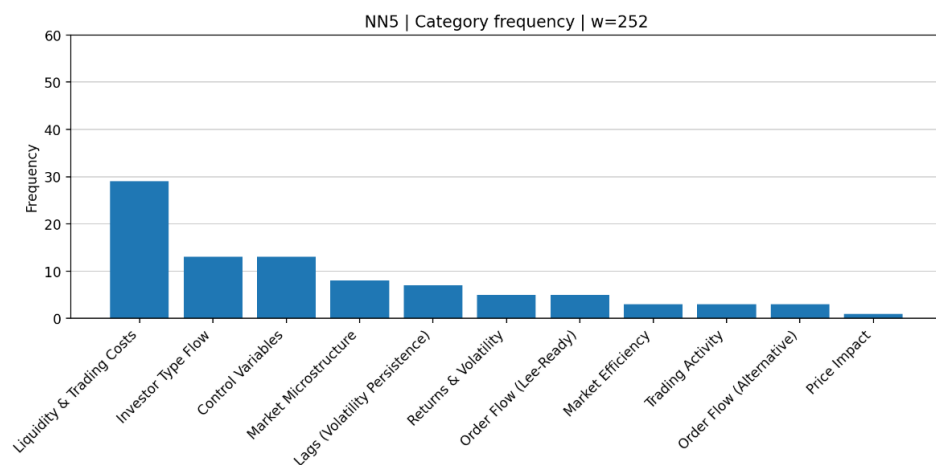


Figure B9: Category-level feature importance for Neural Network 5 (252-day rolling window). Feature importance is based on ablation-induced changes in squared forecast error.



B.2 Feature Importance — 21-Day Rolling Window

Figure B10: Category-level feature importance for XGBoost (randomized-search tuned, 21-day rolling window). Feature importance is measured using gain.

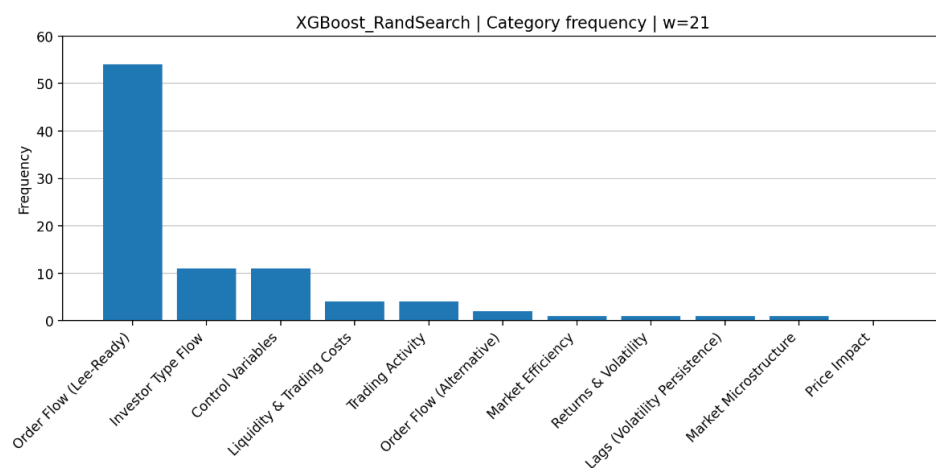


Figure B11: Category-level feature importance for OLS (21-day rolling window). Feature importance is based on absolute standardized coefficients.

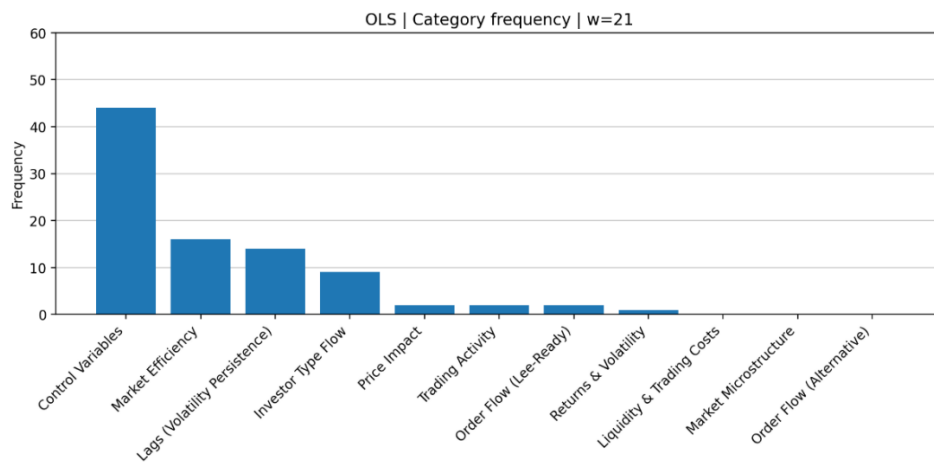


Figure B12: Category-level feature importance for Lasso (21-day rolling window). Feature importance is based on absolute standardized coefficients.

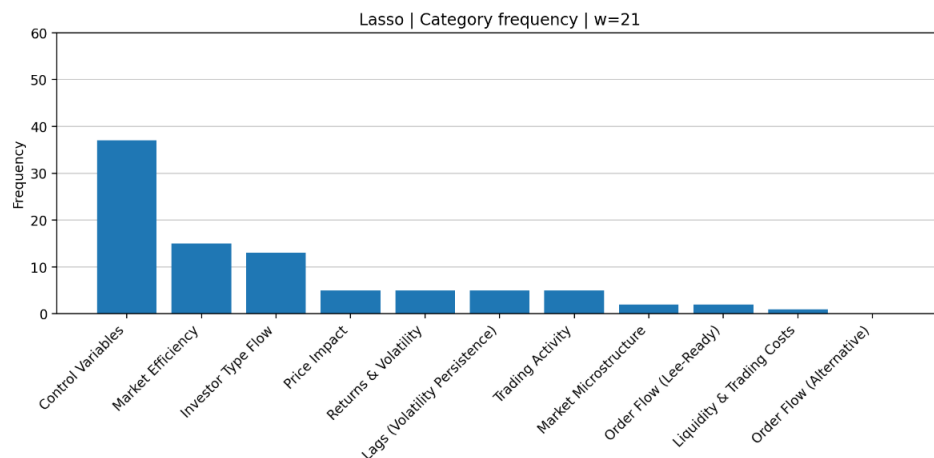


Figure B13: Category-level feature importance for Elastic Net (21-day rolling window). Feature importance is based on absolute standardized coefficients.

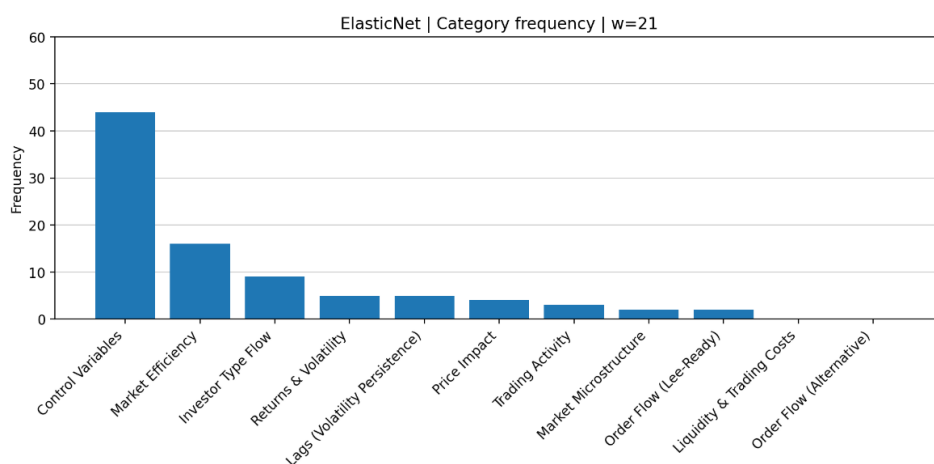


Figure B14: Category-level feature importance for Neural Network 1 (21-day rolling window). Feature importance is based on ablation-induced changes in squared forecast error.

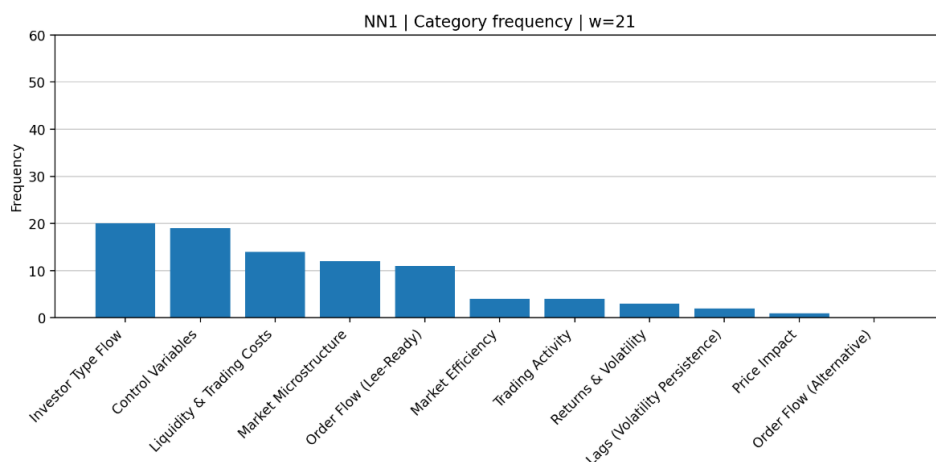


Figure B15: Category-level feature importance for Neural Network 2 (21-day rolling window). Feature importance is based on ablation-induced changes in squared forecast error.

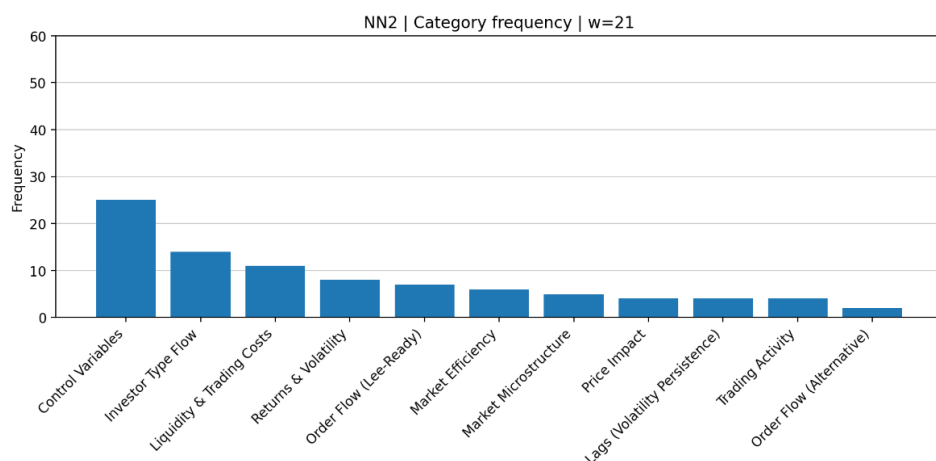


Figure B16: Category-level feature importance for Neural Network 3 (21-day rolling window). Feature importance is based on ablation-induced changes in squared forecast error.

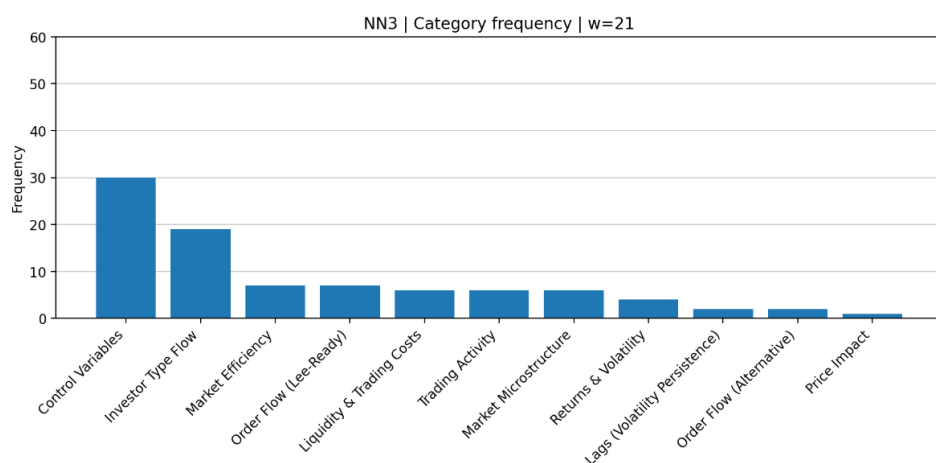


Figure B17: Category-level feature importance for Neural Network 4 (21-day rolling window). Feature importance is based on ablation-induced changes in squared forecast error.

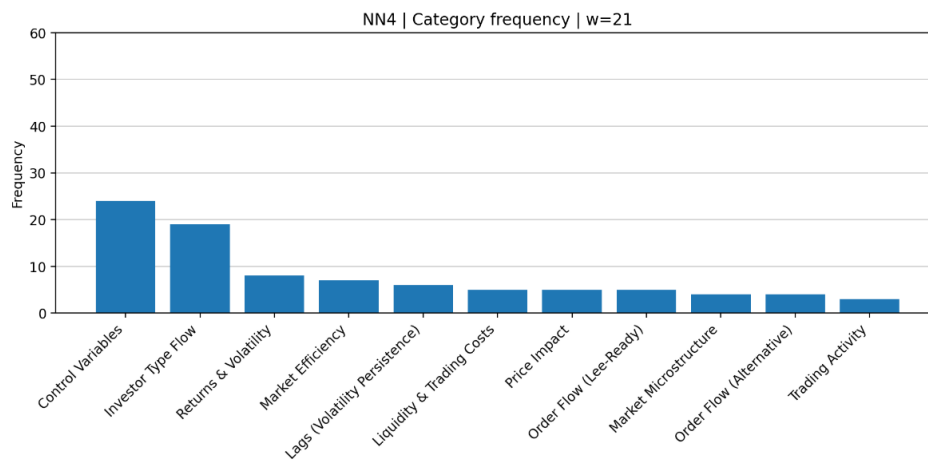
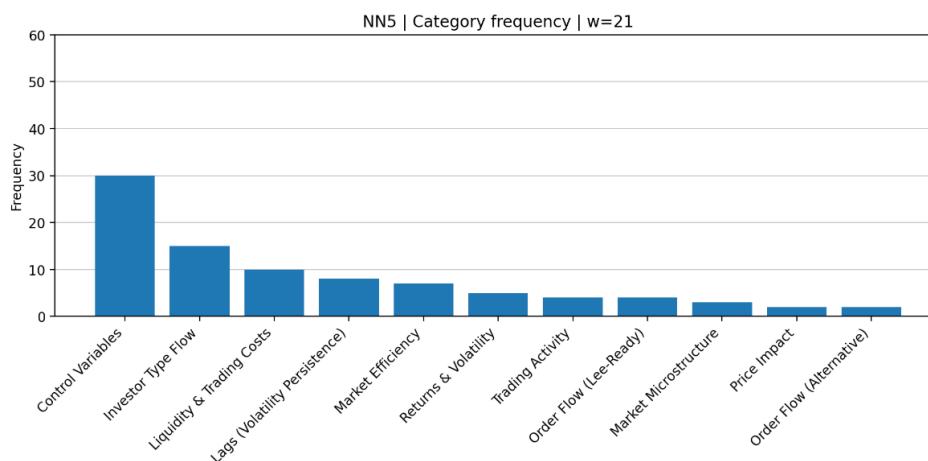


Figure B18: Category-level feature importance for Neural Network 5 (21-day rolling window). Feature importance is based on ablation-induced changes in squared forecast error.



Appendix C — Full Results Tables

Figure C1: Out-of-sample R^2 for one-day-ahead volatility forecasts using a 252-day rolling window over 30 stocks of Dow Jones Industrial Average (DJIA).

Ticker	OLS	Lasso	Ridge	Enet	RF	XGB1	XGB2	NN1	NN2	NN3	NN4	NN5	AR1	GARCH
AAPL	-0.132	0.300	0.270	0.309	0.339	0.348	0.294	0.261	0.242	0.217	0.098	0.301	0.261	0.261
AMGN	-3.073	0.129	0.051	0.146	0.223	0.169	0.147	-0.312	-0.316	-0.165	-0.173	-0.331	0.100	0.105
AMZN	0.148	0.417	0.421	0.422	0.368	0.368	0.343	0.312	0.278	0.359	0.197	0.277	0.130	0.131
AXP	-0.301	0.422	0.427	0.427	0.454	0.424	0.393	0.290	0.409	0.269	0.220	0.237	0.203	0.207
BA	-0.256	0.315	0.322	0.325	0.356	0.293	0.333	0.096	0.240	0.319	-0.043	0.025	0.222	0.225
CAT	-0.688	0.305	0.351	0.340	0.372	0.326	0.317	0.282	0.194	0.342	0.222	0.239	0.167	0.169
CRM	-0.717	0.248	0.237	0.248	0.288	0.286	0.285	0.140	0.230	0.089	0.185	0.040	0.172	0.171
CSCO	-0.616	0.145	0.168	0.137	0.182	0.149	0.142	0.035	0.143	0.171	0.095	0.132	0.079	0.083
CVX	-0.128	0.222	0.238	0.238	0.218	0.215	0.185	0.054	0.185	0.243	0.079	0.060	0.118	0.124
DIS	-0.542	0.115	0.195	0.176	0.209	0.217	0.200	0.080	0.127	-0.028	0.166	-0.060	0.131	0.133
GS	-0.247	0.230	0.207	0.203	0.256	0.197	0.185	0.117	-0.173	-0.098	-0.251	-0.029	0.094	0.097
HD	-1.458	-0.128	0.048	0.012	0.179	0.204	0.176	-0.164	-0.239	-0.207	-0.022	-0.076	0.049	0.051
HON	-2.066	0.225	0.188	0.191	0.247	0.245	0.231	0.104	-0.030	0.065	-0.027	0.115	0.133	0.134
IBM	-0.806	0.224	0.254	0.234	0.328	0.264	0.216	-0.051	0.241	0.204	-0.129	-0.045	0.129	0.132
JNJ	-0.484	0.094	0.135	0.138	0.218	0.189	0.181	0.064	0.184	0.020	0.018	-0.065	0.095	0.093
JPM	-0.352	0.300	0.299	0.309	0.322	0.304	0.305	0.030	0.274	0.088	0.308	0.132	0.147	0.150
KO	-2.215	0.342	0.327	0.343	0.310	0.313	0.313	0.214	0.232	0.068	0.274	0.124	0.174	0.172
MCD	-0.665	0.230	0.254	0.269	0.303	0.269	0.273	-0.121	-0.068	-0.093	-0.067	-0.065	0.061	0.063
MMM	-0.765	0.371	0.372	0.385	0.354	0.332	0.305	0.227	0.278	0.284	0.010	0.235	0.207	0.207
MRK	-1.178	0.249	0.231	0.238	0.267	0.268	0.251	0.114	0.057	0.121	0.027	0.110	0.125	0.126
MSFT	-0.204	0.283	0.278	0.288	0.287	0.290	0.273	0.117	0.248	0.051	0.170	0.144	0.160	0.161
NKE	-0.261	0.214	0.227	0.223	0.303	0.280	0.280	0.069	-0.011	0.175	-0.120	-0.075	0.195	0.198
NVDA	-0.188	0.160	0.171	0.169	0.193	0.153	0.135	0.041	0.155	-0.034	-0.008	0.106	0.100	0.098
PG	-1.517	0.263	0.276	0.285	0.305	0.282	0.282	0.134	0.143	0.043	0.063	0.108	0.121	0.123
SHW	-0.556	0.182	0.178	0.196	0.240	0.197	0.183	0.013	0.158	0.010	-0.144	-0.216	0.131	0.130
TRV	-3.789	0.445	0.429	0.446	0.452	0.413	0.413	0.273	0.377	0.224	0.267	0.141	0.278	0.279
UNH	-2.022	0.207	0.191	0.218	0.240	0.348	0.294	0.261	0.242	0.217	0.098	0.301	0.261	0.261
V	-0.494	0.380	0.374	0.385	0.404	0.392	0.379	0.249	0.217	0.242	0.131	0.232	0.257	0.259
VZ	-0.787	0.143	0.077	0.138	0.187	0.148	0.145	-0.108	-0.246	-0.157	-0.190	-0.172	0.099	0.099
WMT	-0.390	0.005	0.209	0.180	0.275	0.233	0.220	0.091	-0.082	-0.111	-0.099	-0.209	0.165	0.166

Figure C2: Out-of-sample MSE for one-day-ahead volatility forecasts using a 252-day rolling window over 30 stocks of Dow Jones Industrial Average (DJIA).

Ticker	OLS	Lasso	Ridge	Enet	RF	XGB1	XGB2	NN1	NN2	NN3	NN4	NN5	SMA	ARI	GARCH
AAPL	1.185	0.733	0.764	0.724	0.692	0.682	0.739	0.773	0.793	0.819	0.944	0.732	1.043	0.774	0.773
AMGN	2.512	0.537	0.585	0.526	0.479	0.512	0.526	0.809	0.811	0.718	0.723	0.821	0.617	0.555	0.553
AMZN	0.918	0.628	0.623	0.623	0.680	0.680	0.708	0.741	0.778	0.691	0.865	0.779	1.073	0.938	0.936
AXP	1.321	0.587	0.582	0.581	0.554	0.585	0.617	0.722	0.600	0.742	0.792	0.775	1.016	0.810	0.806
BA	0.913	0.498	0.493	0.490	0.468	0.514	0.485	0.657	0.552	0.495	0.758	0.709	0.724	0.565	0.563
CAT	1.385	0.571	0.533	0.542	0.516	0.553	0.561	0.589	0.662	0.540	0.639	0.625	0.822	0.684	0.685
CRM	1.683	0.737	0.748	0.737	0.698	0.700	0.701	0.843	0.755	0.894	0.799	0.941	0.977	0.812	0.810
CSCO	2.139	1.131	1.101	1.143	1.082	1.126	1.136	1.278	1.135	1.098	1.198	1.150	1.319	1.219	1.205
CVX	1.305	0.899	0.881	0.882	0.904	0.907	0.942	1.094	0.943	0.875	1.065	1.087	1.153	1.020	1.008
DIS	1.471	0.844	0.767	0.786	0.755	0.747	0.763	0.878	0.833	0.980	0.795	1.011	0.950	0.828	0.827
GS	1.233	0.762	0.784	0.788	0.736	0.794	0.805	0.873	1.160	1.085	1.237	1.018	0.985	0.896	0.887
HD	2.177	0.999	0.843	0.875	0.728	0.705	0.730	1.031	1.097	1.069	0.906	0.953	0.883	0.842	0.841
HON	3.280	0.829	0.869	0.865	0.806	0.808	0.822	0.959	1.102	1.001	1.098	0.947	1.066	0.928	0.926
IBM	2.096	0.900	0.866	0.889	0.780	0.855	0.910	1.219	0.881	0.924	1.310	1.213	1.156	1.011	1.007
JNJ	1.558	0.951	0.908	0.905	0.821	0.851	0.860	0.983	0.857	1.028	1.031	1.118	1.046	0.950	0.940
JPM	1.715	0.888	0.890	0.876	0.860	0.883	0.882	1.231	0.922	1.157	0.877	1.101	1.263	1.081	1.078
KO	3.540	0.725	0.741	0.723	0.760	0.757	0.756	0.865	0.845	1.026	0.800	0.964	1.105	0.910	0.918
MCD	1.071	0.495	0.480	0.470	0.448	0.470	0.468	0.721	0.687	0.703	0.686	0.685	0.641	0.604	0.603
MMM	1.370	0.488	0.488	0.478	0.501	0.519	0.540	0.600	0.561	0.556	0.769	0.594	0.774	0.616	0.616
MRK	2.055	0.709	0.726	0.719	0.692	0.691	0.707	0.836	0.890	0.830	0.918	0.840	0.940	0.826	0.825
MSFT	1.052	0.626	0.631	0.622	0.623	0.620	0.635	0.772	0.657	0.829	0.725	0.748	0.870	0.734	0.732
NKE	1.140	0.711	0.699	0.703	0.630	0.651	0.651	0.842	0.914	0.746	1.012	0.972	0.901	0.728	0.724
NVDA	0.773	0.546	0.540	0.541	0.525	0.551	0.563	0.624	0.550	0.673	0.656	0.582	0.648	0.586	0.580
PG	1.973	0.578	0.568	0.561	0.545	0.563	0.563	0.679	0.672	0.750	0.735	0.699	0.782	0.689	0.688
SHW	1.318	0.693	0.696	0.681	0.644	0.681	0.692	0.836	0.713	0.839	0.969	1.030	0.844	0.736	0.736
TRV	4.999	0.579	0.596	0.578	0.573	0.613	0.613	0.759	0.651	0.811	0.765	0.897	1.041	0.754	0.753
UNH	1.891	0.496	0.506	0.489	0.475	0.682	0.739	0.773	0.793	0.819	0.944	0.732	1.043	0.774	0.773
V	2.161	0.897	0.905	0.889	0.862	0.879	0.898	1.087	1.133	1.097	1.258	1.111	1.441	1.075	1.072
VZ	1.202	0.576	0.621	0.580	0.547	0.573	0.575	0.745	0.838	0.778	0.800	0.788	0.670	0.606	0.606
WMT	1.818	1.303	1.036	1.073	0.949	1.003	1.021	1.190	1.417	1.454	1.438	1.582	1.304	1.093	1.093

Figure C3: Out-of-sample R^2 for one-day-ahead volatility forecasts using a 21-day rolling window over 30 stocks of Dow Jones Industrial Average (DJIA).

Ticker	OLS	Lasso	Ridge	Enet	RF	XGB1	XGB2	NN1	NN2	NN3	NN4	NN5	AR1	GARCH
AAPL	-0.798	-0.449	-0.024	-0.339	0.028	0.025	0	-0.493	-0.318	-0.214	-0.295	-0.234	-0.062	-0.052
AMGN	-0.929	-0.657	-0.050	-0.395	0.023	-0.001	0	-0.530	-0.504	-0.424	-0.388	-0.352	-0.040	-0.034
AMZN	-0.332	-0.312	0.089	-0.118	0.060	0.097	0.017	-0.285	-0.057	-0.035	-0.081	0.003	-0.002	-0.013
AXP	-1.100	-1.106	-0.090	-0.672	-0.007	-0.060	0	-0.581	-0.415	-0.409	-0.410	-0.436	-0.047	-0.063
BA	-0.674	-0.674	0.016	-0.345	0.049	0.035	0	-0.409	-0.245	-0.185	-0.292	-0.220	0.060	0.050
CAT	-0.384	-0.394	0.026	-0.234	0.034	-0.001	0.006	-0.228	-0.116	-0.014	-0.090	-0.110	0.031	0.029
CRM	-0.918	-1.073	-0.078	-0.985	0.002	0.002	0.006	-0.682	-0.408	-0.389	-0.414	-0.298	-0.029	-0.035
CSCO	-0.628	-0.663	-0.051	-0.654	-0.046	0.001	-0.019	-0.298	-0.221	-0.121	-0.269	-0.116	-0.043	-0.042
CVX	-0.304	-0.268	0.056	-0.111	0.029	0.011	0.010	-0.268	-0.105	-0.094	-0.053	-0.083	-0.028	-0.036
DIS	-0.398	-0.547	0.066	-0.286	0.062	0.002	0	-0.275	-0.043	-0.135	-0.099	-0.141	0.013	0.021
GS	-1.576	-1.434	-0.075	-1.216	-0.016	-0.005	0	-0.851	-0.679	-0.419	-0.871	-0.417	-0.070	-0.073
HD	-0.494	-0.742	0.001	-0.328	0.007	0	0	-0.386	-0.168	-0.109	-0.076	-0.117	-0.021	-0.028
HON	-0.569	-0.757	0.016	-0.392	0.001	-0.002	-0.001	-0.253	-0.235	-0.221	-0.282	-0.216	-0.013	-0.019
IBM	-0.713	-0.816	0.004	-0.490	0.032	-0.006	0.004	-0.529	-0.261	-0.147	-0.274	-0.245	-0.013	-0.006
JNJ	-0.400	-0.674	0.023	-0.234	0.029	-0.004	0	-0.471	-0.182	-0.171	-0.206	-0.138	-0.009	-0.014
JPM	-0.281	-0.278	0.034	-0.116	0.022	-0.002	0	-0.339	-0.187	-0.165	-0.049	-0.090	-0.084	-0.072
KO	-0.831	-0.841	-0.063	-0.592	0.016	0	0	-0.340	-0.258	-0.317	-0.227	-0.338	-0.012	-0.018
MCD	-0.643	-0.721	-0.062	-0.401	-0.022	-0.091	0	-0.537	-0.382	-0.294	-0.392	-0.263	-0.037	-0.060
MMM	-0.652	-0.478	0.010	-0.253	0.049	-0.001	0	-0.391	-0.246	-0.187	-0.168	-0.102	-0.019	-0.016
MRK	-0.881	-0.574	-0.093	-0.530	-0.020	-0.001	0	-0.608	-0.552	-0.389	-0.404	-0.257	-0.068	-0.057
MSFT	-0.469	-0.437	0.053	-0.281	0.042	0.002	0	-0.332	-0.078	-0.110	-0.096	-0.098	0.028	0.015
NKE	-0.745	-0.743	-0.062	-0.490	-0.018	0	0	-0.418	-0.324	-0.233	-0.322	-0.275	-0.055	-0.064
NVDA	-0.624	-0.741	0.066	-0.578	0.074	0.006	0	-0.222	-0.190	-0.245	-0.140	-0.023	-0.082	-0.092
PG	-0.785	-0.801	-0.076	-0.426	-0.027	-0.067	-0.003	-0.429	-0.301	-0.338	-0.251	-0.229	-0.080	-0.072
SHW	-1.587	-1.249	-0.128	-0.756	-0.027	-0.011	-0.005	-0.717	-0.573	-0.609	-0.582	-0.620	0.003	-0.030
TRV	-0.635	-0.782	-0.025	-0.608	-0.014	-0.032	-0.141	-0.267	-0.152	-0.214	-0.205	-0.172	-0.009	-0.032
UNH	-0.970	-0.653	0.010	-0.338	0.027	0.025	0	-0.493	-0.318	-0.214	-0.295	-0.234	-0.062	-0.052
V	-0.404	-0.301	0.058	-0.142	0.055	0.004	-0.001	-0.135	-0.158	-0.095	-0.107	-0.011	0.103	0.094
VZ	-0.470	-0.502	0.026	-0.232	0.045	0	0	-0.379	-0.257	-0.258	-0.128	-0.172	-0.011	-0.010
WMT	-0.614	-0.898	-0.030	-0.419	0.011	-0.063	0	-0.445	-0.318	-0.281	-0.204	-0.235	-0.014	-0.009

Figure C4: Out-of-sample MSE for one-day-ahead volatility forecasts using a 21-day rolling window over 30 stocks of Dow Jones Industrial Average (DJIA).

Ticker	OLS	Lasso	Ridge	Enet	RF	XGB1	XGB2	NN1	NN2	NN3	NN4	NN5	SMA	ARI	GARCH
AAPL	1.194	0.963	0.680	0.889	0.645	0.648	0.664	0.992	0.876	0.807	0.860	0.820	0.665	0.706	0.701
AMGN	1.008	0.866	0.549	0.729	0.511	0.523	0.523	0.800	0.786	0.745	0.726	0.707	0.521	0.544	0.538
AMZN	1.002	0.987	0.685	0.841	0.707	0.679	0.739	0.967	0.795	0.778	0.813	0.750	0.749	0.754	0.761
AXP	1.165	1.168	0.605	0.927	0.558	0.588	0.555	0.877	0.785	0.782	0.782	0.797	0.553	0.581	0.588
BA	0.853	0.853	0.501	0.686	0.485	0.492	0.510	0.718	0.635	0.604	0.659	0.622	0.508	0.479	0.484
CAT	0.805	0.810	0.566	0.717	0.562	0.582	0.578	0.714	0.649	0.590	0.634	0.645	0.580	0.563	0.566
CRM	1.294	1.399	0.727	1.340	0.673	0.674	0.671	1.135	0.950	0.937	0.954	0.876	0.673	0.695	0.699
CSCO	1.635	1.670	1.055	1.661	1.051	1.003	1.023	1.304	1.226	1.126	1.274	1.121	1.000	1.047	1.028
CVX	1.180	1.147	0.854	1.005	0.879	0.895	0.896	1.147	1.000	0.990	0.952	0.980	0.901	0.930	0.928
DIS	0.991	1.096	0.662	0.911	0.665	0.707	0.709	0.903	0.739	0.805	0.779	0.809	0.706	0.699	0.694
GS	1.905	1.800	0.795	1.638	0.751	0.743	0.739	1.368	1.241	1.049	1.383	1.047	0.737	0.791	0.782
HD	1.042	1.215	0.696	0.925	0.692	0.697	0.697	0.966	0.814	0.773	0.750	0.779	0.695	0.712	0.714
HON	1.264	1.416	0.793	1.121	0.805	0.807	0.807	1.010	0.995	0.984	1.033	0.980	0.803	0.816	0.821
IBM	1.331	1.410	0.773	1.157	0.751	0.781	0.774	1.187	0.980	0.891	0.989	0.967	0.775	0.787	0.783
JNJ	1.208	1.445	0.843	1.065	0.838	0.866	0.863	1.269	1.020	1.010	1.040	0.982	0.862	0.870	0.865
JPM	1.127	1.123	0.850	0.981	0.860	0.881	0.879	1.177	1.044	1.025	0.923	0.958	0.877	0.953	0.944
KO	1.316	1.324	0.765	1.144	0.707	0.719	0.719	0.963	0.904	0.947	0.882	0.962	0.723	0.728	0.738
MCD	0.716	0.750	0.462	0.610	0.445	0.475	0.436	0.670	0.602	0.564	0.606	0.550	0.434	0.452	0.462
MMM	0.870	0.778	0.521	0.660	0.500	0.527	0.526	0.732	0.656	0.625	0.615	0.580	0.525	0.536	0.535
MRK	1.252	1.048	0.728	1.018	0.679	0.667	0.666	1.070	1.033	0.925	0.935	0.837	0.663	0.711	0.704
MSFT	0.979	0.957	0.631	0.853	0.638	0.665	0.666	0.888	0.719	0.740	0.730	0.732	0.665	0.647	0.657
NKE	1.056	1.055	0.643	0.902	0.616	0.605	0.605	0.858	0.801	0.746	0.800	0.772	0.603	0.639	0.643
NVDA	1.016	1.089	0.584	0.987	0.579	0.622	0.626	0.765	0.744	0.779	0.713	0.640	0.623	0.677	0.679
PG	0.927	0.935	0.558	0.740	0.533	0.554	0.520	0.742	0.675	0.695	0.650	0.638	0.518	0.560	0.557
SHW	1.594	1.385	0.695	1.082	0.633	0.623	0.619	1.058	0.969	0.991	0.975	0.998	0.614	0.614	0.634
TRV	0.961	1.047	0.603	0.945	0.596	0.607	0.671	0.745	0.677	0.714	0.708	0.689	0.586	0.593	0.607
UNH	0.939	0.788	0.472	0.638	0.464	0.648	0.664	0.992	0.876	0.807	0.860	0.820	0.665	0.706	0.701
V	1.317	1.220	0.883	1.071	0.887	0.934	0.939	1.065	1.086	1.028	1.039	0.948	0.937	0.841	0.851
VZ	0.851	0.870	0.564	0.713	0.553	0.579	0.579	0.798	0.728	0.728	0.653	0.679	0.577	0.585	0.585
WMT	1.548	1.820	0.988	1.361	0.949	1.019	0.959	1.386	1.264	1.228	1.155	1.184	0.958	0.973	0.969