Estimating Stock Market Betas via Machine Learning

Wolfgang Drobetz[1], Fabian Hollstein[2], Tizian Otto[3], and Marcel Prokopczuk[4,‡]

September 2022

Abstract

This paper evaluates the predictive performance of machine learning techniques in estimating time-varying market betas of U.S. stocks. Compared to *established* estimators, *machine learning-based* approaches outperform from both a statistical and an economic perspective. They provide the lowest forecast errors and lead to truly ex-post market-neutral portfolios. Among the different techniques, random forests perform the best overall. Moreover, the inherent model complexity is strongly time-varying. Historical betas, as well as turnover and size signals, are the most important predictors. Compared to linear regressions, interactions and nonlinear effects substantially enhance predictive performance.

Keywords: Beta estimation; machine learning; active trading strategy

JEL classification codes: G11, G12, C58, G17

[1] Faculty of Business Administration, University of Hamburg, Moorweidenstr. 18, 20148 Hamburg, Germany.

[2] School of Human and Business Sciences, Saarland University, Campus C3 1, 66123 Saarbruecken, Germany.

[3] Faculty of Business Administration, University of Hamburg, Moorweidenstr. 18, 20148 Hamburg, Germany.

[4] School of Economics and Management, Leibniz University Hannover, Koenigsworther Platz 1, 30167 Hannover, Germany

[‡] *Acknowledgments*: We thank Yakov Amihud, Michael Bauer, Wolfgang Bessler, Axel Cabrol, Mikhail Chernov, Hubert Dichtl, Valentin Haddad, Kay Giesecke, Barney Hartman-Glaser, Lisa Goldberg, Bernard Herskovic, Bryan Kelly, Serhiy Kozak, Markus Leippold, Martin Lettau, Lars A. Lochstoer, Harald Lohre, Tyler Muir, Andreas Neuhierl, Terrance Odean, Stavros Panageas, Markus Pelger, Tatjana Puhan, Carsten Rother, Boris Vallee, Michael Weber, and Ivo Welch for helpful comments. Conflicts of interest: none.

1. Introduction

In single-factor asset pricing models, such as the Capital Asset Pricing Model (CAPM) of Sharpe (1964), Lintner (1965), and Mossin (1966), a stock's expected return is driven solely by its sensitivity to a systematic risk factor, i.e., the market risk. While multi-factor models that include additional systematic risk factors based on firm fundamentals appear to explain the cross-sectional variation in expected returns somewhat better (see, e.g., Fama and French, 2008, and Harvey, Liu, and Zhu, 2016, for comprehensive evidence), the CAPM performs well in explaining time series variation and continues to dominate in the industry. Graham and Harvey (2001), Jacobs and Shivdasani (2012), and Graham (2022) document that the vast majority of chief financial officers of large U.S. companies rely primarily on a one-factor market model for capital-budgeting decisions, i.e., to estimate their equity cost of capital. For this application, firms typically estimate market betas as main ingredients and treat the market risk premium almost as a free parameter (Cochrane, 2011; Jacobs and Shivdasani, 2012). Investors, in turn, utilize market betas for capital-allocation decisions and portfolio risk management (Barber, Huang, and Odean, 2016; Berk and van Binsbergen, 2016; Daniel et al., 2020). However, there are two main problems when using the CAPM, and hence directly relying on market betas, for these applications: Betas 1) cannot be observed directly, underscoring the need for precise estimates, and 2) are time-varying (Campbell et al., 2001). The second problem substantially complicates matters, because forecasts of future betas are needed for the vast majority of applications (e.g., the equity cost of capital over the lifetime of a new project or the future market risk of an investment). Therefore, the main goal of both researchers and practitioners is to find approaches that yield estimates of future betas with minimal forecast errors (see Section 2 for an extensive overview of the previous literature).

Machine learning techniques have been shown to outperform established approaches in various other prediction tasks, such as forecasting stock-level expected returns (e..g, Gu, Kelly, and Xiu, 2020, Drobetz and Otto, 2021, and Leippold, Wang, and Zhou, 2021), bond risk premia (e.g., Bianchi, Büchner, and Tamoni, 2021, and Bali et al., 2022), and earnings expectations (e.g., van Binsbergen, Han, and Lopez-Lira, 2021). Realized betas are less noisy than realized returns. Given the higher signal-to-noise ratio, machine learning-based models potentially work equally well or even better for estimating market betas than they do for the tasks examined in these previous studies, e.g., predicting stock-level expected returns.

Therefore, our main objective is to examine whether machine learning-based beta estimators can outperform established approaches in estimating time-varying market betas and, if yes, why. For our empirical analysis, we use 1) a large universe of U.S. stocks, 2) a long and recent sample period, 3) a broad set of both benchmark and machine learning-based beta estimators, and 4) a comprehensive set of predictor variables. Relative to the extant literature, we dig considerably deeper. That is, as our first contribution, we substantially expand the scope and rigor in each of these four dimensions. Second, and even more importantly, we examine *when* and *how* machine learning-based estimators add value. To the best of our knowledge, we are the first to comprehensively address the black box issue in estimating market betas by investigating the characteristics and functioning scheme of machine learning techniques.

We compare the predictive performance of machine learning-based beta estimators (linear regressions, tree-based models, and neural networks) to that of several established benchmarks (rolling-window approaches, as well as shrinkage-based, portfolio-based, and long-memory forecast models). To fit the machine learning techniques, in line with Gu, Kelly, and Xiu (2020), we use an additive prediction error model, and follow the time series cross-validation approach based on a ten-year rolling window. We consider a comprehensive set of eighty-one predictors (see Table 1 for details). As such, our study greatly extends the set of conditioning variables used in Cosemans et al. (2016), and includes predictors based on accounting information, technical indicators, a macroeconomic indicator, predictors based on sample estimates of beta, and the industry classification of Fama and French (1997).

Our first key result is that machine learning techniques outperform established approaches from both a statistical and an economic perspective. Random forests perform particularly well. They yield the lowest average forecast error, as measured by the average value-weighted mean squared error (MSE), closely followed by gradient boosted regression trees and neural networks. These three approaches yield dramatically lower average forecast errors than any of the established benchmark approaches. These machine learning methods are in the Hansen, Lunde, and Nason (2011) model confidence set (MCS) the vast majority of the time. The corresponding fractions for all benchmark models are substantially smaller. Moreoevr, the prediction errors of all benchmark approaches are significantly higher during most of the sample period, as is evident from significantly positive Diebold and Mariano (1995) test statistics. For example, compared to the most natural benchmark model, one-year daily rolling betas, the average MSE is 19% lower for random forests. They significantly outperform this benchmark model more than 60% of the time and are in the MCS more than twice as often.

Next, we examine the differences in forecast errors across beta estimators, and identify the parts of the sample period and the types of stocks, for which the differences in forecast errors across beta estimators are particularly pronounced. In addition to providing more accurate beta estimates in general, machine learning methods outperform the benchmark models even more in distressed economic environments (during or right after most National Bureau of Economic Research (NBER) recessions). These are the times when it is particularly difficult to accurately predict market betas. According to the Mincer and Zarnowitz (1969) MSE decomposition, the low forecast errors of machine learning-based approaches stem from a well-balanced trade-off between bias and inefficiency. In contrast to established beta estimators, random forests and other machine learning methods produce less extreme and less volatile forecasts. Such properties avoid the systematic underestimation of the betas of stocks in low-beta deciles and systematic overestimation of those in high-beta deciles, a central problem inherent to the task of forecasting time-varying market betas. We further find that the machine learning-based approaches are superior for nearly all types of stocks (sorted into portfolios based on firm characteristics or industry affiliation), and especially beneficial for small and illiquid stocks, value stocks, and loser stocks. Therefore, including the respective firm fundamentals as predictors in the forecast models likely helps generate better forecasts for these stocks.

In addition to the statistical comparison, we analyze the economic value of beta forecasts in portfolio formation exercises. We find that the machine learning methods again outperform all other approaches. They are the only ones that can generate minimum variance portfolios and long–short anomaly portfolios (for all common stock market anomalies) that are truly market neutral ex post. This is because they exhibit the lowest forecast errors for those stocks that typically receive the highest weights within market-neutral minimum variance portfolios, i.e., the firms with the lowest and highest ex-ante beta estimates. Furthermore, the machine learning-based estimators exhibit the lowest forecast errors for those stocks that eventually end up in the long or short portfolios, i.e., those stocks with extreme values for the corresponding firm characteristics.

In a penultimate step, we inspect changes in the inherent model complexity over time and decompose predictions into the contributions of individual variables using relative variable importance metrics. We find that the inherent model complexity positively correlates with the general difficulty in predicting betas, e.g., more complex models are required if betas are difficult to predict. Furthermore, more complex models tend to generate somewhat lower benchmark-adjusted forecast errors. We also find that the historical betas and technical indicators are the first and second most important groups of predictor variables, respectively. However, variable importance varies over time, and also unconditionally less important variables play important roles at times.

Finally, our results underscore the systematic connection between market betas and firm characteristics. One important economic reason that machine learning methods outperform is their ability to distill the information content of a large set of predictor variables. Importantly, however, random forests, gradient boosted regression trees, and neural networks also outperform linear regressions, which incorporate the same set of covariates. We show that this is largely due to their ability to utilize nonlinear and interactive patterns, which provides a second, complementary explanation for the advantageousness of machine learning methods. However, they must be adequately trained and tuned to avoid overfitting.

Overfitting can manifest along two different strands: model overfitting and backtest overfitting. Model overfitting refers to machine learning models with overly high in-sample fit but poor out-of-sample predictive performance. To avoid model overfitting, we control for the degree of model complexity by tuning relevant hyperparameters. These parameters cannot be pre-set, but must be determined adaptively from the sample data. The parameter tuning approach iteratively reduces in-sample fit by searching for the degree of model complexity that will produce reliable out-of-sample predictive performance. Backtest overfitting refers to a researcher's arbitrariness in choosing firm coverage and sample period, predictors, and tuning parameters. If information from the out-of-sample period is used to fit the models, consciously or not (Schorfheide and Wolpin, 2012), this might lead to overstated out-of-sample predictive performance (Bailey et al., 2014, 2017; Harvey and Liu, 2014, 2015; Harvey, Liu, and Zhu, 2016). To avoid backtest overfitting, we use the largest possible firm coverage and sample period.[5] We also use a comprehensive set of eighty-one predictor variables that are motivated by the previous literature (see Table 1 for details), instead of focusing on only those covariates that have been shown to perform best in similar prediction tasks. Finally, we follow common parameter choices to cover a representative range of possible parameter specifications (see Internet Appendix, Section A, Table A1, Panel B for details), from which the hyperparameters are selected.

The remainder of this paper is organized as follows: Section 2 briefly reviews the related literature and Section 3 describes our dataset. Section 4 summarizes the different forecast models (including the machine learning techniques and the key tuning parameters). The empirical results are presented in Section 5, with the estimation results in Section 5.1 and the characteristics and functioning scheme of the machine learning techniques in Section 5.2. Section 6 concludes. The Internet Appendix contains technical details, results of further analyses, and several robustness tests.

2. Literature review

Because the original CAPM is a static one-period model, its most natural application is based on the premise that stocks' market betas are constant over time. However, various studies find evidence for time variation in these betas (Bollerslev, Engle, and Woolridge, 1988; Jagannathan and Wang, 1996; Ferson and Harvey, 1999; Petkova and Zhang, 2005; Ang and Chen, 2007). Jagannathan and Wang (1996) therefore propose a conditional version of the CAPM, and show that it explains the cross-sectional variation in expected returns much better than its static counterpart.

[5] As per Gu, Kelly, and Xiu (2020), using large datasets mitigates sample selection or data snooping biases (Lo and MacKinlay, 1990) and also help avoid model overfitting by increasing the ratio of observation count to parameter count.

To estimate time-varying market betas, traditional approaches focus on historical return information. Black, Jensen, and Scholes (1972) and Fama and MacBeth (1973) use the coefficient estimate in a time series ordinary least squares (OLS) regression of stock-level excess returns on excess returns of the market portfolio. Their five-year rolling window of monthly returns accounts for the time variation in beta estimates. Despite the robustness to misspecification (no predictors needed), rolling-beta estimates face a bias-variance trade-off with regard to window length and data frequency. In addition, such time series estimators are sensitive to outliers in the return history, and often produce extreme and volatile beta forecasts. The literature offers modifications of the baseline rolling-window approach to improve this trade-off. For example, Hollstein, Prokopczuk, and Wese Simen (2019) show that a weighted least squares (WLS) approach with exponential weights performs well, while Welch (2019) suggests winsorizing stock-level returns before running the OLS regressions. Both studies find substantially reduced forecast errors (compared to the baseline rolling-window approach).

Enhancing rolling betas with supplemental cross-sectional information has also been shown to improve beta forecasts. The idea is that a stock's beta estimate should not be too dissimilar from those of other stocks with similar characteristics. Vasicek (1973) and Karolyi (1992) find that shrinking rolling-beta estimates towards a prior regarding the true beta reduces estimation noise. This helps increase the signal-to-noise ratio. In contrast, Cosemans et al. (2016) argue that shrinkage based on such joint priors (identical across firms) dampens only part of the noise in rolling betas. They suggest specifying priors unique to each firm, while incorporating a broad set of firm fundamentals as predictors. Kim, Korajczyk, and Neuhierl (2020) and Kelly, Moskowitz, and Pruitt (2021) emphasize that commonly used firm fundamentals (such as market capitalization, book-to market ratios, etc.) can help improve the prediction of time-varying market betas. Other successful approaches include: 1) assigning portfolio beta estimates to individual stocks (Fama and French, 1992), and 2) exploiting the long-memory properties of beta time series (Becker et al., 2021).

Studies that use machine learning-based approaches abound in the empirical asset pricing literature. While most of them focus on the predictability of return characteristics, there is little research to date on the predictability of risk characteristics. For example, Christensen, Siggaard, and Veliyev (2021) compare various machine learning algorithms (including tree-based models and neural networks) in forecasting stock-level expected volatility. They find substantial outperformance relative to the established heterogeneous autoregressive (HAR) approach.[6] These studies almost exclusively focus on a stock's *total* risk. However, an estimate of *systematic* risk (e.g., the CAPM beta) is at least as important, for both firms and investors.

Several papers model expected returns by using firm characteristics that capture time variation in multi-factor betas (Connor and Linton, 2007; Connor, Hagmann, and Linton, 2012; Fan, Liao, and Wang, 2016; Kelly, Pruitt and Su, 2019). Gu, Kelly, and Xiu (2021) use this approach in a machine learning setting. We add to this literature by explicitly and directly focusing on market betas rather than on expected returns. This focus is targeted for two main applications: 1) equity cost of capital estimation, for which it is industry practice to use estimated market betas rather than expected returns, and 2) portfolio risk management, which requires direct knowledge of stock-level risk characteristics, i.e., market betas.

The study most closely related to ours is Jourovski et al. (2020). The authors use estimates of linear regression approaches and regression tree models to forecast realized betas. To this end, they analyze the MSCI US stock universe (on average, 540 mostly large-cap stocks) during the January 1999–December 2019 sample period. They show that, overall, regression trees outperform rolling-window forecasts and linear regressions. However, they do not confront machine learning methods with the best benchmark models documented in the recent literature. Furthermore, the authors only examine the economic value of machine learning methods for Betting-Against-Beta portfolios. They do not investigate changes in the inherent model complexity over time or explore patterns of nonlinear and interactive effects in the relationship between predictors and expected market betas.

Therefore, in this paper, we dig considerably deeper: we 1) comprehensively compare the performance of machine learning estimators (including neural networks) to the best benchmark models docu-

[6] Other studies that also apply machine learning techniques to forecasting future volatility concentrate on only one specific approach, respectively. For example, Mittnik, Robinzonov, and Spindler (2015) and Luong and Dokuchaev (2018) consider tree-based models, while Donaldson and Kamstra (1997), Hillebrand and Medeiros (2010), Fernandes, Medeiros, and Schadt (2014), Bucci (2020), and Rahimikia and Poon (2020) explore neural networks.

mented in the recent literature, 2) analyze when and how machine learning techniques outperform by disentangling a model's forecast error into its separate components, compare the time series of forecast errors, and analyze forecast errors of cross-sectional portfolio sorts, 3) document the economic value for a minimum variance portfolio and a large set of anomalies, 4) analyze the inherent model complexity as well as nonlinear and interactive effects, and 5) examine a substantially larger and longer sample along with a much more comprehensive set of predictors.[7]

3. Data

Our market and fundamental data come from the Center for Research in Security Prices (CRSP) and Compustat, and consist of daily and monthly returns as well as various firm characteristics.[8] Our sample is free of survivorship bias and includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ. To calculate excess returns, we use the three-month T-bill rate, scaled to a daily or monthly horizon, as the risk-free rate. The value-weighted portfolio of all stocks serves as a proxy for the market portfolio.

We follow Cosemans et al. (2016) in cleaning the initial dataset. We include a stock in the empirical analysis for month t if it satisfies the following criteria: First, its book value of equity (according to Fama and French, 1992) must be non-negative and both its net sales and monthly dollar trading volume must be positive. Second, its return in the current month t and over the previous thirty-six months must be available. Third, it must provide full information on all predictor variables (no missing values). In every month, we require at least 250 firms to be included in the cross section. This limits our sample period to July 1972–December 2020, which consists of 1,500 stocks per month on average.

[7] A comparison with the best benchmark models is important because, as indicated above, the previous literature documents several methods that outperform simple rolling-window estimators. Without further analysis, it remains unclear whether machine learning techniques also outperform more sophisticated, and thus more conservative benchmark models and indeed provide the best beta estimates. In contrast to Jourovski et al. (2020), our sample covers the July 1972–December 2020 period and includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ (on average, 1,500 stocks).

[8] Market data from CRSP are assumed to become public immediately, while fundamental data from Compustat are assumed to be known four months after the end of the fiscal year.

In Table 1, we present in detail our comprehensive set of eighty-one predictors. It extends the set used in Cosemans et al. (2016). Theirs includes five fundamental covariates (*size*, *book-to-market ratio*, *financial leverage*, *operating leverage*, and *momentum*), one macroeconomic covariate (*default spread*), and forty-seven dummies that correspond to the industry classification of Fama and French (1997).[9] To this baseline set, we add twenty-eight variables that have been shown to explain cross-sectional variation in future market betas (Beaver, Kettler, and Scholes, 1970; Amihud and Mendelson, 2000; Jacoby, Fowler, and Gottesman, 2000; Chincarini, Kim, and Moneta, 2020; Kelly, Moskowitz, and Pruitt, 2021).[10] In particular, we incorporate twenty-five additional fundamental covariates (e.g., *age*, *illiquidity*, or *turnover*), which we further classify into eighteen additional *predictors based on accounting information* and seven additional *technical indicators*.[11] To capture time series dynamics in beta, we also include three *predictors based on sample estimates of beta* obtained from rolling regressions. We use three-month and one-year historical windows of daily returns ($ols_{1y,d}$ and $ols_{3m,d}$), as well as a five-year historical window of monthly returns ($ols_{5y,m}$) to obtain information from short-, medium-, and long-term trends in the beta time series. Including historical betas based on three different horizons allows for a heterogeneous autoregressive predictive structure. This may help capture the long-memory properties of the market beta time series, as documented by Becker et al. (2021).

In line with Cosemans et al. (2016), we winsorize outliers in all firm characteristics to the 0.5th and 99.5th percentile values of their cross-sectional distributions, and we correct for skewed distributions by logarithmically transforming relevant predictor variables. In addition, to remove any time trend in their

[9] Cosemans et al. (2016) follow Gulen et al. (2011) in measuring a firm's operating leverage (as the ratio of change in operating income before depreciation to change in net sales). We opt for the Novy-Marx (2011) definition, which is another well-established measurement approach in the literature. This increases consistency across predictors, especially with respect to financial leverage. Note that the main findings of the empirical analysis are qualitatively similar for other $oplev$ definitions.

[10] From Kelly, Moskowitz, and Pruitt's (2021) extensive list of predictors that significantly predict future market betas, we omit only the bid–ask spread. This is because its inclusion, together with the requirement that all predictor variables must be available for all stocks, would shrink our sample substantially (the data are largely unavailable until the mid-1980s), and thus could hamper the empirical analysis. Note that the main findings of the empirical analysis are qualitatively similar when including the bid–ask spread as an additional fundamental covariate.

[11] Chincarini, Kim, and Moneta (2020) follow Jovanovic and Rousseau (2001), Loughran and Ritter (2004), and Fink et al. (2010) in measuring a firm's age (based on incorporation/founding or IPO dates). We opt for the Fama and French (2004) definition, another well-established measurement approach in the literature, as other definitions would shrink our sample substantially, and thus could hamper our empirical analysis. Note that the main findings of the empirical analysis are qualitatively similar for other age definitions.

average values, we standardize *all* firm characteristics by subtracting the cross-sectional mean and dividing by the cross-sectional standard deviation on a monthly basis.

[Insert Table 1 here]

Many of the predictors are constructed similarly, e.g., sample estimates of beta based on different rolling windows, or incorporate similar information, e.g., valuation ratios measured relative to the market value of equity, which leads to high correlations. As discussed in Lewellen (2015), any resulting multicollinearity is not a major concern because we are mostly interested in the overall predictive power of the machine learning-based forecast models, rather than the marginal effects of each single predictor. In addition, the machine learning techniques we use (except simple linear regressions) are perfectly suitable for solving the multicollinearity problem either by nature (tree-based models) or by applying different types of regularization, e.g., a lasso-based penalization of the weights (neural networks).

4. Forecast models

The main objective of our empirical analysis is to examine whether machine learning techniques can outperform established beta estimators in terms of predictive performance and, if yes, why. We are particularly interested in exploring whether incorporating interactions and nonlinearity in the relationship between predictors and future market betas can add incremental predictive power. We therefore run a horse race between established and machine learning-based beta estimators, comparing the predictive performance from both a statistical and an economic perspective. In addition, we investigate the characteristics and functioning scheme of the machine learning techniques that help explain their superior predictive performance.

In line with Cosemans et al. (2016), the estimation setting in our empirical analysis is as follows. Out-of-sample beta estimates are obtained at the firm level and on a monthly basis, following an iterative procedure: In the first iteration step, we utilize data up to the end of month t and obtain forecasts for each stock i's beta during the out-of-sample period (from the beginning of month $t+1$ to the end of month $t+$

k), i.e., $\beta^F_{i,t+k|t}$ (or abbreviated $\beta^F_{i,t}$). We set k equal to 12, focusing on a one-year forecast horizon.[12] In the next iteration step, we utilize data up to the end of month $t+1$, and obtain forecasts of stock-level betas over the subsequent out-of-sample period (from the beginning of month $t+1+1$ to the end of month $t+1+k$). By iterating through the dataset, we obtain time series of overlapping out-of-sample beta estimates, which we then compare to realized betas. Andersen, Bollerslev, and Meddahi (2006) show that a realized beta measure constructed from high-frequency returns is a consistent estimator of the true integrated beta. Therefore, we measure realized betas using daily returns over exactly the one-year forecast intervals, i.e., $\beta^R_{i,t+k} = \frac{Cov^R_{iM,t+k}}{Var^R_{M,t+k}}$, where $Cov^R_{iM,t+k}$ is the realized covariance between stock i and the market portfolio M, and $Var^R_{M,t+k}$ is the realized market variance. Both moments are computed from continuously compounded returns, which, in turn, are obtained from log price changes.

Before presenting the results of our empirical analysis (see Section 5), we briefly introduce the different models to forecast time-varying market betas (see Internet Appendix, Section A, Table A1 for an overview). They differ in their overall approach and complexity, but all aim to minimize the forecast error, which we compute as the value-weighted MSE in each out-of-sample forecast period:[13]

$$MSE_{t+k|t} = \sum_{i=1}^{N_t} w_{i,t}(\beta^R_{i,t+k} - \beta^F_{i,t+k|t})^2 \text{, with } k = 12. \tag{1}$$

where N_t is the number of stocks in the sample at the end of month t, and $w_{i,t}$ is stock i's market capitalization-based weight. Note that realized betas are itself estimates. However, evaluating the forecasts based on future realized betas is an approach that works very well statistically (see, e.g., Hansen and Lunde, 2006,

[12] Alternatively, one-month and five-year forecast horizons are common in the literature ($k = 1$ and $k = 60$, respectively). However, both alternatives have shortcomings, which is why we opt for a one-year forecast horizon. First, realized betas computed from one-month rolling windows of daily returns are very noisy, which hampers the evaluation of forecast errors. Second, forecast horizons much longer than twelve months are less common in the industry due to the underlying nature of fiscal years.

[13] Andersen, Bollerslev, and Meddahi (2005) analyze the problem of measurement errors in the context of volatility forecasting. They find that the true predictive accuracy of forecasts is underestimated due to noise in the realized volatility measures (which serve as a proxy for the true latent volatility). Measurement errors are particularly concerning when using lower-frequency (daily) data. Regardless of the data frequency, however, measurement errors are most pronounced for small stocks, as their returns are more sensitive to microstructure noise. Using a value-weighting scheme in computing forecast errors mitigates the impact of noise in realized betas of small stocks, allowing for a more reliable evaluation of predictive performance than with an equal-weighting scheme. However, in Section C of the Internet Appendix, we show that the results are robust to using the equal-weighted MSE and the value-weighted mean absolute error (MAE) loss function.

for theoretical framework and empirical evidence). Furthermore, Patton (2011) shows that the MSE criterion is robust to mean-zero noise in the evaluation proxy.

4.1. Benchmark estimators

From the voluminous literature on beta estimation, we select a representative set of established forecast models, which we classify into four model families based on methodology (see Internet Appendix, Section A for details, implementation choices, and references). The first model family consists of rolling-window estimators, for which we consider two basic *historical betas* obtained from rolling regressions using a five-year window of monthly returns (ols_5y_m) and a one-year window of daily returns (ols_1y_d), as well as two common modifications, *exponentially-weighted betas* based on short ($ewma\_s$) and long ($ewma\_l$) half-lives, and *slope-winsorized betas* (bsw). The second subcategory consists of shrinkage-based estimators, for which we include three *shrinkage betas* that shrink ols_1y_d towards the average beta within the stock universe ($vasicek$), an industry portfolio ($karolyi$), and a firm-specific beta prior ($hybrid$). The third and fourth model families are portfolio-based and long-memory estimators, for which we include *portfolio betas* that are assigned to individual stocks ($fama\text{-}french$), and *long-memory betas* that exploit the long-memory properties of beta times series ($long\text{-}memo$).

4.2. Machine learning-based estimators

The machine learning-based approach follows a different, more rigid path in capturing cross-sectional variation in future betas. For example, shrinkage-based approaches derive prior beliefs and sample estimates of beta separately, before aggregating these two sources of information into shrinkage betas. Rather than taking this “detour”, the machine learning techniques focus explicitly on the objective of forecasting market betas. Realized betas directly enter the regressive setting as dependent variables, while firm characteristics, sample estimates of beta, etc. contribute as predictors. This helps keep the forecast objective in mind while simultaneously using multiple sources of (prior) information, which potentially leads to incremental predictive power. We adapt the additive prediction error model outlined in Gu, Kelly, and Xiu (2020) to describe a stock’s beta:

$$\beta_{i,t+k}^{R} = E_t(\beta_{i,t+k}^{R}) + \varepsilon_{i,t+k}, \tag{2}$$

where $\beta_{i,t+k}^{R}$ is stock i's realized beta over the one-year forecast horizon starting at the beginning of month $t+1$. The expected beta is estimated as a function of predictor variables, and described by the "true" model $g^*(z_{i,t})$, where $z_{i,t}$ represents the P-dimensional set of predictors:

$$E_t(\beta_{i,t+k}^{R}) = g^*(z_{i,t}). \tag{3}$$

Although the machine learning-based forecast models used in our empirical analysis belong to different families (e.g., linear regressions, regression trees, and neural networks), they are all designed to approximate the true forecast model by minimizing the out-of-sample MSE. Approximations of conditional expectations $g^*(z_{i,t})$ are flexible and family-specific. Approximation functions $g(.)$ can be linear or nonlinear, as well as parametric, $g(z_{i,t}, \theta)$, where θ is the set of true parameters, or non-parametric $g(z_{i,t})$.

4.2.1. Sample splitting

Machine learning methods are devised 1) to incorporate a comprehensive set of variables simultaneously and 2) to consider both nonlinearity and interactions in the relationship between predictors and expected market betas. However, they are prone to overfitting, which is why we must control for the degree of model complexity by tuning the relevant hyperparameters. Tuning parameters are, e.g., the number and/or depth of trees in tree-based models, or the number of layers and/or nodes in neural networks. To avoid overfitting and maximize out-of-sample predictive power, hyperparameters should not be pre-set, but rather must be determined adaptively from the sample data. In particular, they are selected from a comprehensive set of parameter specifications (see Internet Appendix, Section A, Table A1, Panel B for details). The parameter tuning approach iteratively reduces in-sample fit by searching for the degree of model complexity that will produce reliable out-of-sample predictive performance. To this end, in line with Gu, Kelly, and Xiu (2020), we use the time series cross-validation approach, which maintains the temporal ordering of the data, and splits the sample into three distinct subsamples: a training sample, a validation sample, and a test sample.

We use the training sample to estimate the model for multiple parameter specifications, while the purpose of the subsequent validation sample is to tune the parameters.[14] That is, based on the models estimated from the training sample, we calculate the time series mean of the monthly *value-weighted* MSEs within the validation sample for each parameter specification. The model with the parameter specification that minimizes the validation error is used for out-of-sample testing. Note that, because we choose the tuning parameters from the validation sample, it is not truly out-of-sample. The test sample, however, is used for neither model estimation nor parameter tuning. This is why it is truly out-of-sample, and appropriate for evaluating a model's out-of-sample predictive power.

In an asset management context, where new data emerge over time, a sample-splitting scheme that periodically includes more recent data should be applied (see, e.g., West, 2006, for an extensive overview). This is why the "rolling window" and "recursive window" methods gradually shift the training and validation samples forward in time. The former method holds the length of training and validation samples constant; the latter increases them progressively. Moreover, because the recursive window approach always incorporates the entire history of data, it is computationally more intensive than the rolling-window approach. Because of this, and because machine learning algorithms are generally computationally intensive, Gu, Kelly, and Xiu (2020) avoid recursively refitting models each month. Instead, they refit once every year, as most of the fundamental firm characteristics are only updated annually anyway. To allow for time variability in the relation between predictors and future market betas, we follow the rolling-window approach implemented by Drobetz and Otto (2021). Each year, we roll forward the training and validation sample one year, while holding constant the length of the respective samples.[15] In our empirical analysis, we select ten years for training and validation, i.e., nine years for training and one year for validation, and

[14] While simple linear regressions do not require parameter tuning (based on the validation sample), we also estimate this model from only the training sample. This enhances the comparability with the machine learning-based models (penalized linear regressions, tree-based models, and neural networks). Note that the main findings of the empirical analysis are qualitatively similar if we pool the training and validation samples together to estimate the simple linear regressions model at each re-estimation date.

[15] Gu, Kelly, and Xiu (2020) follow the recursive-window approach. Each year, they increase the training sample by one year, while holding the length of the validation sample constant but rolling it forward one year. For sample periods as large as ours, this might overweight past observations, hindering the informativeness of future betas. The rolling-window approach allows to omit those uninformative observations. Importantly, the main findings of the empirical analysis are qualitatively similar for other lengths of the training and validation samples (including the sample splitting scheme used in Gu, Kelly, and Xiu, 2020).

one year for testing from the data. Beginning in December 1979, we obtain the last beta estimates in December 2019, using ten years of data for training and validation (2009:01–2017:12 and 2018:01–2018:12, respectively), which we compare to realized betas over the next year.[16] In total, we use forty years and one month of data for testing.

4.2.2. Machine learning techniques

We consider a comprehensive set of eighty-one predictors (consisting of fundamental and macroeconomic covariates, as well as predictors based on sample estimates of beta, see Table 1 for details) to fit the machine learning techniques. Throughout our empirical analysis, we analyze three different forecast model families, which differ in their overall approach and complexity (see Internet Appendix, Section B for details, implementation choices, and references).

The first model family consists of *linear regressions*, for which we use the training sample to run a pooled regression of future realized betas $\beta_{i,t+k}^{R}$ on the set of eighty-one predictors at each re-estimation date. In particular, we either use the ordinary least squares loss function (lm) or modify it by adding a penalty term, i.e., we apply an elastic net penalization ($elanet$).[17] The latter is the most common machine learning technique to overcome the overfitting problem in a high-dimensional regression setting (e.g., when the number of predictors becomes very large relative to the number of observations). It is important to note that, if not explicitly included as pre-determined terms, pooled regressions cannot capture any interactions or nonlinear effects (neither simple nor penalized approaches). We thus utilize linear regressions as a benchmark to identify whether such effects, on top of the two-way interaction between firm characteristics and the default spread, lead to incremental predictive power.

The second model family consists of *tree-based models*, for which we include random forests (rf) and gradient boosted regression trees ($gbrt$), the most common representatives within this subcategory.

[16] Because we focus on a one-year forecast horizon, there is a one-year gap between the end of the sample used for training and validation (2018:12) and the estimation date (2019:12).

[17] Note that the main findings of the empirical analysis are qualitatively similar when running weighted least squares (WLS) regressions (using the stocks' market capitalization-based weights).

The third model family are *neural networks*, for which we consider specifications with up to five hidden layers and up to thirty-two neurons (nn_1 to nn_5).[18] Both tree-based models and neural networks incorporate multi-way interactions and nonlinearity inherently, without the need to add new predictors to capture these effects in advance.

5. Empirical results

Having introduced the established and machine learning-based estimation approaches, we now focus on applying these models to forecast out-of-sample market betas. In the first part of our empirical analysis, we run a horse race between all beta estimators (see Internet Appendix, Section A, Table A1 for an overview), comparing the predictive performance from both a statistical and an economic perspective. In the second part, we address the characteristics and functioning scheme of the machine learning techniques that help explain their superior predictive performance.

5.1. Estimation results

We start with studying the forecast models' ability to predict out-of-sample market betas. Our main objective is to examine whether machine learning-based beta estimators can outperform established approaches in terms of predictive performance and, if yes, why. We thus assess the cross-sectional and time series properties of all beta estimates. We also compare and decompose the resulting forecast errors. We further investigate the underlying causes of differences in predictive performance by contrasting the time series of forecast errors and analyzing the forecast errors of cross-sectional portfolio sorts. Finally, we evaluate whether differences in statistical predictive performance translate into a superior economic predictive performance in portfolio formation exercises.

[18] Neural networks are computationally intensive and can be specified in an innumerable number of different architectures. This is why we retreat from tuning parameters (e.g., the size of batches or the number of epochs) and instead pre-specify five different models. We assume that our nn_1 to nn_5 architectures serve as a conservative lower bound for the predictive performance of neural network models in general. Empirically, as shown in Section C of the Internet Appendix, the predictive performance for the neural network models deteriorates slightly in the number of hidden layers. In the main part of the paper, we thus only present and discuss the results for the simplest nn_1 architecture.

5.1.1. Cross-sectional and time series properties of beta estimates

Initially, we investigate the properties of out-of-sample beta estimates obtained from the different forecast models. Panel A of Table 2 focuses on cross-sectional properties, presenting time series means of 1) the *value-weighted* cross-sectional average of estimated betas, 2) the *value-weighted* cross-sectional standard deviation, and 3) the cross-sectional minimum, median, and maximum values. Following the procedure outlined in Pastor and Stambaugh (1999), we also report the implied cross-sectional standard deviation of true betas, i.e., $\widehat{Std}(\beta^R) = \left[\overline{Var(\beta^F)} - \overline{\widehat{Var}_{\beta_i^R}}\right]^{1/2}$, which helps measure a beta estimator's precision. The minuend ($\overline{Var(\beta^F)}$) is the time series average of the observed *value-weighted* cross-sectional sample variance, and the subtrahend ($\overline{\widehat{Var}_{\beta_i^R}}$) is the *value-weighted* cross-sectional average of each firm's sampling variance. Small gaps between observed and implied standard deviations thus indicate small estimation errors, which point towards a precise measurement of true betas.

The cross-sectional mean is close to one for all beta estimators, while the cross-sectional dispersion varies widely across forecast models. The cross-sectional standard deviations are largest for the rolling-window estimators, regardless of the underlying weighting scheme. Note that running separate firm-level time series regressions ignores any cross-sectional information and appears to lead to extreme and volatile rolling-beta estimates. Winsorizing or shrinking betas towards a well-defined prior, assigning portfolio beta estimates to individual stocks, or exploiting the long-memory properties of beta time series can reduce the cross-sectional spread in estimated betas substantially. Overall, the cross-sectional summary statistics of the benchmark models are consistent with those reported by Cosemans et al. (2016). The machine learning-based beta estimators exhibit the smallest cross-sectional standard deviations and lead to the least extreme beta estimates (according to their cross-sectional minimum and maximum values). Especially the inclusion of slow-moving firm fundamentals as predictors in an additive prediction error model (on top of sample estimates of beta) appears to result in less volatile and less extreme estimates.[19] Among others, firm fundamentals are intended to pick up long-run movements in betas driven by changes in these variables.[20] The observed cross-sectional standard deviation of beta forecasts, however, is most informative when compared to the implied cross-sectional standard deviation of true betas. It reveals that true betas are measured with the lowest precision (large gaps) by the rolling-window estimators, and with the highest precision (small gaps) by the machine learning-based models and the long-memory approach, respectively.

Panel B of Table 2 focuses on time series properties and shows that the different estimation procedures also yield different time series dynamics.[21] It presents *value-weighted* cross-sectional means of 1) the time series average of the estimated betas, 2) the time series standard deviation, 3) the time series minimum, median, and maximum values, and 4) the first-order autocorrelation.

The time series summary statistics are consistent with the findings obtained from the cross-sectional metrics. The average beta estimates are close to one for all estimators, but their time series variabilities differ. The machine learning-based estimators, along with the long-memory model, yield the lowest time series standard deviations, while the high volatility of the historical betas likely reflects measurement noise. Although they incorporate slow-moving firm fundamentals as predictors, the average time series autocorrelations of the machine learning-based models are rather low when compared to those of the benchmark models. Nevertheless, they generally exceed 0.90.

[Insert Table 2 here]

5.1.2. Forecast errors

We now inspect the statistical predictive performance of the different forecast models by comparing their forecast errors (based on a one-year forecast horizon). Panel A of Table 3 reports the time series means

[19] The hybrid beta model also uses slow-moving fundamental predictors to specify a prior regarding the true beta. The slightly larger cross-sectional and time series standard deviation (compared to that from the machine learning techniques) likely stems from rolling-window beta estimates, which are the second component of the hybrid beta estimate.

[20] This is in line with the findings in Cochrane (1998), who reports this phenomenon while using slow-moving dividend ratios to predict long-run dividend growth.

[21] Following Becker et al. (2021), we omit firms with less than fifty beta estimates to allow for valid inference.

for monthly value-weighted MSEs, calculated as in Equation (1). Ignoring any cross-sectional information, beta estimates obtained from rolling regressions lead to sizable forecast errors, ranging from 18.37% for the ols_5y_m model to 8.97% for the $ewma\_l$ model. Winsorizing or shrinking rolling-beta estimates towards a well-defined prior, assigning portfolio beta estimates to individual stocks, or exploiting the long-memory properties of beta time series can reduce the average MSE substantially. In line with Welch (2019), Cosemans et al. (2016), and Becker et al. (2021), the best-performing estimators among our benchmark approaches are slope-winsorized betas (8.31%), hybrid betas (8.13%), and long-memory betas (7.83%).

Turning to the machine learning methods, we observe that tree-based models and neural networks can reduce the average forecast error relative to established beta estimators even further (average MSEs between 7.49% and 7.70%). In contrast, linear regressions (both simple, 9.27%, and penalized, 8.84%) face notably higher average MSEs.[22] Therefore, utilizing information from a comprehensive set of predictors, taken in isolation, is not sufficient for superior beta estimates. Consequently, a large part of the rf, $gbrt$, and nn_1 models' outperformance appears to be due to their ability to capture nonlinearity and interactions, which helps improve the predictive performance.[23] Random forests perform the best overall, yielding an average MSE of 7.49%. They are able to decrease the average forecast error relative to the most commonly used estimation techniques, the ols_5y_m and ols_1y_d model, by 59% and 19%, respectively. Even relative to the best-performing benchmark approach, the $long\text{-}memo$ model, random forests still reduce the forecast error by more than 4%.[24]

Since, by construction, these figures only reflect a forecast model's average predictive performance, we also investigate forecast errors over time. First, we visually examine the differences in MSEs across forecast models over the sample period. For the sake of brevity, we focus on the comparison of random

[22] Our finding that tree-based models perform particularly well in forecasting market betas is consistent with Jourovski et al. (2020), although they do not analyze any of the top three performing benchmark models or neural networks.

[23] In Section 5.2.3, we explore in detail how the predictive performance of random forests is superior to that of simple linear regressions. We show that their ability to capture nonlinearity and interactions in the relationship between predictors and future market betas is an important driver of this outperformance.

[24] Relative to the $hybrid$ model, which also utilizes firm characteristics, the average forecast error of the rf model is nearly 8% lower.

forests with one-year rolling betas and simple linear regressions.[25] Figure 1 depicts the forecast errors of random forests at each re-estimation date relative to those achieved by one-year rolling betas (ols_1y_d; Panel A) and simple linear regressions (lm; Panel B). The relative forecast errors are computed as the percentage difference in monthly MSEs ($mse\_monthly\_pd$) and time series means for monthly MSEs within each test sample (mse_test_pd). These percentage differences are calculated as one minus the MSE of the random forests divided by the MSE of the respective benchmark model. We follow the convention that positive differences indicate superior predictive performance, i.e., reduced forecast errors, of the random forest relative to the ols_1y_d and lm models, respectively. To contextualize these forecast errors, we also show the NBER recession periods (shaded grey).

The visualizations suggest that random forests are able to reduce the forecast errors relative to the ols_1y_d and lm models most of the time during the sample period. This implies that higher average MSEs for one-year rolling betas and simple linear regressions (see Table 3, Panel A) are not predominantly driven by just a few severe outliers in their MSE time series. It rather emphasizes that random forests are generally able to provide more accurate beta estimates. In addition, larger-than-average positive differences during or right after most recessions (MSEs are computed based on a one-year forecast horizon) indicate that the rf model outperforms the two benchmark models even more strongly in distressed economic environments, during which it is particularly difficult to accurately predict market betas.

Second, to assess those differences statistically, Panel B of Table 3 reports the fraction of months during the out-of-sample period for which the column model is 1) in the Hansen, Lunde, and Nason (2011) model confidence set (MCS), and 2) significantly better than the row model in a pairwise comparison (according to the Diebold and Mariano (1995) test (DM test) statistics). The MCS incorporates an adjustment for multiple testing and is designed as the set of models which contains the best model based on a certain

[25] We select random forests (the forecast approach with the lowest average forecast error; see Table 3, Panel A), one-year rolling betas (the most commonly used estimation technique), and simple linear regressions (a counterpart to random forests, using the same forecast model setting, i.e., the additive prediction error model with annual re-estimation, but without the ability to capture nonlinearity and interactions). This selection allows us to evaluate whether the rf model's outperformance is attributable to its ability to include firm fundamentals as predictors and/or to utilize nonlinear and interactive effects.

level of confidence.[26] The DM tests of equal predictive ability inspect differences in stock-level squared forecast errors (SEs):

$$SE_{i,t+k|t} = (\beta^R_{i,t+k} - \beta^F_{i,t+k|t})^2, \text{ with } k = 12. \tag{7}$$

The DM test statistic in month t for comparing the model under investigation j with a competing model i is $DM_{ij,t} = \frac{\bar{d}_{ij,t}}{\hat{\sigma}_{\bar{d}_{ij,t}}}$, where $d_{ij,t} = SE_{i,t+k|t} - SE_{j,t+k|t}$ is the difference in SEs, $\bar{d}_{ij,t} = \sum_{i=1}^{N_t} w_{i,t} d_{ij,t}$ is the value-weighted cross-sectional average of these differences, and $\hat{\sigma}_{\bar{d}_{ij,t}}$ is the heteroskedasticity- and autocorrelation-consistent (HAC) standard error of $d_{ij,t}$. We use the Newey and West (1987) estimator with four lags to compute HAC standard errors, and follow the convention that positive signs of $DM_{ij,t}$ indicate superior predictive performance of model j relative to model i in month t, i.e., that model j yields, on average, lower forecast errors than model i.[27]

We find that regression trees and neural networks are in the model confidence set in most of the 481 months during the out-of-sample period, with fractions ranging from 76.92% for the $gbrt$ model to 85.24% for the rf model. Thus, only about 15% of the time, we can reject the null hypothesis that random forests are the best model. This figure is only slightly higher than the share of false positives to be expected at the 10% significance level. This overwhelmingly suggests that particularly the rf model, but also other machine learning techniques, provide very good forecasts for market betas. The fractions at which the machine learning-based methods are in the model confidence set are nearly twice as large as that of the one-year rolling betas (40.96%). Therefore, for more than 59% of the months, we can reject the null hypothesis that the one-year rolling betas, which perform best among those commonly used in the industry, provide the best forecasts. The shares with which the machine learning estimators are in the MCS are also notably larger than those of the best-performing benchmark approaches, e.g., slope-winsorized betas (56.13%), hybrid betas (59.46%), and long-memory betas (71.73%). This observation holds as well when compared with

[26] We follow Becker et al. (2021) throughout this study in examining statistical significance towards the 10% level. For the MCS approach, this translates to a 90% model confidence set.

[27] According to Gu, Kelly, and Xiu (2020), DM test statistics are asymptotically $N(0,1)$-distributed and test the null hypothesis that the divergence between two forecast models is zero, so they map to *p*-values in the same way as regression *t*-statistics.

linear regressions (53.85% for the lm model and 60.71% for the $elanet$ model). Consequently, the machine learning-based models are predominantly among the best-performing approaches.

Supporting this view, the results from the monthly DM tests show that tree-based models and neural networks dominate most established approaches and linear regressions. In up to 95% of the months, they yield a significantly lower MSE than the respective benchmark models. For nearly all benchmark approaches, this share is well over or close to one half, at least for the rf and $gbrt$ models. The machine learning-based estimators significantly outpredict even the best-performing benchmark model, long-memory betas, during at least 35.14% of the sample period. In turn, the benchmark approaches rarely yield significantly lower MSEs than the machine learning-based approaches (as indicated by low shares for all). The best benchmark, the $long\text{-}memo$ model, significantly outperforms the machine learning-based estimators only 22.04% of the time.

Taken together, this points towards a sizable outperformance of regression trees and neural networks over established beta estimators. Comparing the machine learning techniques, we find that the rf model is slightly superior to the $gbrt$ and nn_1 models. Random forests possess the largest MCS fraction, and also outpredict the gradient boosted regression trees and neural networks more often that they are dominated by them.

[Insert Table 3 and Figure 1 here]

5.1.3. Decomposition of forecast errors

We next investigate the underlying causes of differences in predictive performance across forecast models. In line with Becker et al. (2021), we divide the overall MSE into its components: *bias*, *inefficiency*, and *random error*. Bias indicates a model’s degree of misspecification, i.e., whether beta estimates and realized betas are, on average, different. Inefficiency reveals a model’s tendency to systematically yield positive forecast errors for low values (underestimation) and negative forecast errors for high values (overestimation), or vice versa. Random error represents the remaining forecast error unrelated to predictions and realizations. Adapting the Mincer and Zarnowitz (1969) approach, we first decompose the overall MSE in each out-of-sample forecast period:

$$MSE_{t+k|t} = \underbrace{(\bar{\beta}^R_{t+k} - \bar{\beta}^F_{t+k|t})^2}_{\text{Bias}} + \underbrace{(1-b_t)^2\sigma^2(\beta^F_{t+k|t})}_{\text{Inefficiency}} + \underbrace{(1-\rho^2_t)\sigma^2(\beta^R_{t+k})}_{\text{Random error}}, \tag{8}$$

where $\bar{\beta}$ and $\sigma^2(\beta)$ refer to the value-weighted mean and variance, respectively. b_i is the coefficient estimate in the WLS regression $\beta^R_{i,t+k} = a_t + b_t\beta^F_{i,t+k|t} + e_{i,t+k}$ (using the stocks' market capitalization-based weights), and ρ^2_t is the coefficient of determination of this regression. We then compute the aggregate numbers as the time series averages of monthly MSE components.

Table 4 presents the results of the MSE decomposition. For all machine learning techniques, the bias component is slightly larger than zero, while the inefficiency component is close to 1%. Therefore, their forecasts are, on average, only moderately biased and only slightly inefficient. Among these models, linear regressions exhibit the worst metrics (0.67% and 1.16% vs. 0.70% and 1.01% for the lm and $elanet$ models, respectively). While their forecasts are similarly biased, adding a penalty term to the least squares loss function helps reduce inefficiency. Compared to linear regressions, both bias and inefficiency are substantially lower for regression trees and neural networks. These results support the view that their ability to capture nonlinearity and interactions helps improve the predictive performance. Interestingly, some established approaches are even slightly less biased than tree-based models and neural networks, with numbers ranging from 0.19% for the $fama\text{-}french$ model to 0.42% for the $hybrid$ model. This is because, by construction, the rolling-beta estimates that contribute to these models yield a value-weighted average beta close to one. Importantly, however, nearly all traditional forecast models (except the $long\text{-}memo$ model) yield much larger numbers for the inefficiency component (between 1.17% for the $hybrid$ model and 8.04% for the ols_5y_m model). Accordingly, particularly for low- and high-beta stocks, such beta estimators systematically generate sizable measurement errors.

Adding both MSE components together, it becomes clear that the bias–inefficiency trade-off is notably worse for these forecast models. These results are in accordance with those of Becker et al. (2021) and help explain the outperformance of tree-based models and neural networks over established approaches (as indicated by lower total MSEs).[28] Random forests achieve the best bias–inefficiency trade-off, with 0.32% for the former and 0.90% for the latter. This also leads to the lowest total MSE of 7.49% across all forecast models (see Table 3, Panel A).

[Insert Table 4 here]

5.1.4. Forecast errors of cross-sectional portfolio sorts

We now aim to gain further insights into how machine learning methods outperform traditional predictors. To this end, we want to identify the types of stocks, e.g., high- or low-beta stocks, large or small stocks, etc., for which differences in forecast errors across beta estimators are particularly pronounced. Following the procedure outlined by Cosemans et al. (2016), we begin by examining to what extent the differing abilities to predict future market betas can be traced back to underestimating the betas of low-beta stocks and overestimating those of high-beta stocks. We therefore sort stocks into decile portfolios based on their predicted betas at the end of each month t. Monthly forecast errors in this empirical test are defined as the value-weighted MSE between beta forecasts and realized betas over the next year within each portfolio.[29] However, we are not only interested in the size of forecast errors, but also in their direction. To this end, at each re-estimation date, we compute the fraction of stocks within each decile portfolio for which the difference between beta forecasts and realized betas is positive. Metrics below 0.5 would indicate that an estimator on average underestimates realized betas, while numbers above 0.5 would point towards overestimation on average.

[28] The random error component is the largest part of the disaggregated MSEs. In general, the differences in this component among different models are smaller than for the bias and inefficiency parts. However, tree-based models and neural networks also yield random error components that are smaller than those of all benchmark approaches.

[29] Our procedure is qualitatively similar to that used in Cosemans et al. (2016). They define monthly forecast errors as the mean squared deviation between ex-post and ex-ante portfolio betas within each portfolio. They then measure ex-ante portfolio betas as the value-weighted averages of beta forecasts within each decile portfolio and ex-post portfolio betas as the value-weighted averages of realized betas over the next year. Our approach is more consistent with the forecast error definition used for the unconditional empirical tests (see Table 3). More importantly, it avoids the cancelling out of positive and negative forecast errors (over- and underestimations) within each decile portfolio.

Figure 2 plots the time series averages of monthly forecast errors within each decile portfolio (grey bars).[30] We find that, for all approaches, the extreme portfolios yield the largest average forecast errors. Rolling-beta estimates perform the worst, winsorizing or shrinking them towards a well-defined prior, assigning portfolio beta estimates to individual stocks, or exploiting the long-memory properties of beta time series lowers the average forecast error in the extreme beta deciles.[31] The machine learning approaches further reduce the level of forecast errors for (almost) all portfolios. Compared to the classical predictors, the forecast error distributions across the decile portfolios are more even.

To these visualizations, we add the average under- and overestimation fractions (black unfilled squares). We find that the problem of under- and overestimation (i.e., underestimating the betas of stocks in low-beta deciles, and overestimating those in high-beta deciles) is apparent for all established approaches (albeit to varying degrees). While it is smaller for some of the better-performing approaches (e.g., the $fama\text{-}french$ and $long\text{-}memo$ models, respectively), even those cannot avoid this problem entirely. In contrast to the aforementioned beta estimators, the machine learning techniques do not show signs of systematic underestimation in low-beta deciles and systematic overestimation in high-beta deciles. This pattern stems from less extreme beta estimates, as indicated by the low cross-sectional forecast dispersion (see Table 2, Panel A), and is well explained by the nature of their forecasts (see Section 5.1.1).

[Insert Figure 2 here]

In a next step, we examine how the differences in forecast errors across beta estimators are related to other firm characteristics or the industry classification. For the sake of brevity, we again focus on the comparison of random forests (rf) with one-year rolling betas (ols_1y_d) and simple linear regressions (lm). We replicate the procedure outlined for Figure 2, but now sort stocks into decile portfolios based on a firm's size (me), book-to-market ratio (bm), momentum (mom), illiquidity ($illiq$), and industry

[30] For the sake of brevity, we omit the ols_5y_m, $ewma\_s$, $ewma\_l$, $vasicek$, and $karolyi$ models, for which the results are (slightly) worse than those for the competing forecast model(s) of the same family, which we present in Figure 2 (i.e., the ols_1y_d, bsw and $hybrid$ models, respectively).

[31] Winsorizing or shrinking rolling-window beta estimates, but also portfolio-based und long-memory betas, avoid extreme beta estimates by construction. To this end, Cosemans et al. (2016) argue that shrinkage is most beneficial for stocks with extreme sample estimates of beta, which, in turn, are likely attributable to large measurement errors.

classification (ind).[32] Figure 3 plots the time series averages of monthly forecast errors within each decile portfolio for the respective benchmark model (red bars) and the rf model (grey bars). To these visualizations, we add the percentage differences in average forecast errors relative to the respective benchmark model (black unfilled triangles), calculated as one minus the MSE of the random forests divided by the MSE of the respective benchmark model.

Overall, the graphs suggest that random forests can reduce the forecast errors relative to the ols_1y_d (left-hand column) and lm (right-hand column) models for nearly all decile portfolios, respectively. This is indicated by percentage differences larger than zero (triangles above the dashed line). In line with Figure 2, this implies that random forests generally provide more accurate beta estimates. It also emphasizes that the higher average MSEs for one-year rolling betas and simple linear regressions (see Table 3, Panel A) are not predominantly driven by high forecast errors for only a few stocks with specific firm characteristics. However, compared to the ols_1y_d model, random forests can reduce the forecast errors within extreme decile portfolios even starker, both in absolute and relative terms.

For example, especially for small and illiquid stocks, the rf model can provide more accurate beta estimates than those obtained from the ols_1y_d model. Moreover, despite the patterns being less strongly pronounced, we observe larger-than-average benefits of random forests for value and loser stocks, respectively. Because the outperformance of random forests over simple linear regressions is mostly marginal for these stocks, we can attribute this observation to the inclusion of firm fundamentals as predictors, rather than the rf model's ability to capture nonlinearity and interactions. Finally, we find that random forests outperform the two benchmark models in each single industry. Compared to the ols_1y_d model, their value-added is largest for "healthcare, medical equipment, and drugs" (HMD), "utilities" (U), and "wholesale, retail, and some services" (WRS), while the rf model is most beneficial for "telephone and television transmission" (TTT) when compared to simple linear regressions.

[32] Note that, for these visualizations, we consider ten rather than forty-seven dummies that correspond to the industry classification of Fama and French (1997).

In summary, for traditional approaches, very small and large beta forecasts, as well as beta forecasts for stocks with extreme firm characteristics or within specific industries, should be used with caution. However, machine learning-based beta forecasts appear to be uniformly useful for all types of stocks.

[Insert Figure 3 here]

5.1.5. Market-neutral minimum variance portfolios

Because accurate beta estimates are crucial for various applications in academia and industry, we now investigate whether statistically more accurate forecasts translate into economic gains in portfolio formation.[33] We follow Cosemans et al. (2016), and apply a framework that ensures a straightforward comparison of relative performance across forecast models. In line with Ghysels and Jacquier (2006), our objective is to construct a market-neutral portfolio with a minimum return variance. This setup mimics the strategy of a hedge fund that aims for a portfolio with a minimum return variance, while neutralizing its market risk. When constructing minimum variance portfolios (MVPs), expected returns do not enter the optimization.[34] Therefore, differences in weights of each stock within the optimized portfolio result solely from differences in estimated covariance matrices. Beta estimates, in turn, can be used to impose a factor structure. Since the covariance matrix of stocks is high-dimensional, this approach helps obtain more precise estimates by reducing the number of parameters necessary to be estimated.

In particular, the portfolio optimization follows the procedure outlined in Ghysels and Jacquier (2006).[35] Separately for each forecast model, we first obtain the out-of-sample beta forecasts, which we then utilize within a single-factor model framework to predict the out-of-sample covariance matrix $\Omega_{t+1|t}$,

[33] The tests conducted thus far are statistical in nature. Leitch and Tanner (1991) suggest that there may be only a weak association between statistical measures and economic profitability. We therefore explore whether differences in statistical predictive performance lead to differences in predictive power from an economic perspective.

[34] Note that we are interested in a comparison across beta estimators, rather than contrasting various optimization procedures for minimum variance portfolios (although we acknowledge that some of those techniques might lead to optimized portfolios with even smaller variance). To this end, we do not follow previous studies that, e.g., aim to improve minimum variance portfolios using shrinkage-based estimators of the covariance matrix or impose weight constraints (Jagannathan and Ma, 2003; Ledoit and Wolf, 2003). Instead, we follow Cosemans et al. (2016) and opt for the construction of simple MVPs based on a factor structure for the covariance matrix.

[35] In line with Chan, Karceski, and Lakonishok (1999) and Cosemans et al. (2016), we exclude microcaps and focus on stocks with market capitalizations above the 20th percentile of NYSE stocks.

i.e., $\Omega_{t+1|t} = s^2_{M,t+1|t} B_{t+1|t} B'_{t+1|t} + D_{t+1|t}$, where $B_{t+1|t}$ represents the $N_t \times 1$ vector of out-of-sample beta forecasts, $s^2_{M,t+1|t}$ the out-of-sample forecast of market variance (variance of excess market return), and $D_{t+1|t}$ the diagonal matrix containing the out-of-sample forecasts of idiosyncratic variances $d^2_{i,t+1|t}$. We compute residuals as the differences between realized and estimated stock returns, i.e., $R_{i,t} - \beta^F_{i,t} R_{M,t}$. We obtain the market and idiosyncratic variances from monthly returns over the last three years ending in month t, taking these historic values as predictions for month $t+1$. We then utilize the covariance forecasts to construct a market-neutral minimum variance portfolio by selecting portfolio weights that solve the following problem, separately for each beta estimator:

$$\min_{w_t} \quad w'_t \Omega_{t+1|t} w_t \ \text{ s.t.} \tag{9}$$

$$\sum_i w_{i,t} = 1$$

$$\sum_i w_{i,t} \beta^F_{i,t+1|t} = 0$$

The first constraint implies that the portfolio is fully invested, while the second states that the ex-ante predicted portfolio beta should be zero. Because hedge funds usually have short investment horizons, we rebalance the portfolio on a monthly basis and record the realized return over the next month. To assess the performance of the optimized portfolio, following Cosemans et al. (2016), we obtain the ex-post portfolio beta from a regression of monthly portfolio returns on market returns over the sample period.[36] From this market model, we also extract the ratio of the remaining systematic variance to the total variance of the resulting minimum variance portfolio. In theory, perfect beta estimates lead to a truly market-neutral portfolio. In this case, the ex-post beta is zero and total variance depends only on idiosyncratic, not systematic, risk. Consequently, an ex-post beta close to zero and a small fraction of systematic to total variance would indicate highly accurate ex-ante beta forecasts.

[36] As a robustness check, we obtain monthly ex-post portfolio betas as the weighted average of realized betas over the next year, using the weights of each stock within the optimized portfolio (similarly to the procedure outlined for Figures 2 to 4). This setup mimics the strategy of a hedge fund that, on a monthly basis, aims to neutralize the market risk of its portfolio *over the next year* (instead of focusing on the full sample period). Importantly, the machine learning techniques continue to outperform established estimation approaches, while still leading to portfolios that are truly market-neutral ex post (with portfolio betas that are very close to zero, together with insignificant *t*-statistics testing the null hypothesis that the respective betas are zero).

Panel A of Table 5 reports the ex-post portfolio betas (β), the t-statistics (in parentheses) testing the null hypothesis that these betas are zero, and the ratio of systematic to total variance. For the t-tests, we use Newey and West (1987) standard errors with four lags. We find that the machine learning models are the only approaches leading to portfolios that are truly market neutral ex post. They yield realized portfolio betas between 0.00 for the rf model and 0.12 for the lm model. According to the respective Newey and West (1987) t-statistics between 0.02 and 1.21, these betas are insignificantly different from zero. In addition, the systematic component of the total portfolio risk is very close to zero (between 0.00% of the total variance for the rf model and 1.62% for the lm model). Thus, the remaining variation in returns of the machine learning-based portfolios is indeed independent of market movements.

Importantly, all the portfolios based on traditional forecast models realize ex-post betas that are statistically significant at least at the 10% level. They are also economically large, ranging from 0.18 (t-statistic of 1.90) for the $hybrid$ model to 0.36 (t-statistic of 6.18) for the ols_5y_m model.[37] Furthermore, the ratios of systematic to total variance are substantially larger than zero (between 4.12% for the $fama\text{-}french$ model and 22.48% for the ols_5y_m model).

While random forests again perform the best overall, even linear regression models (the lm and $elanet$ models, respectively) lead to market-neutral portfolios. This suggests that utilizing information from a comprehensive set of predictors and avoiding the problem of under- and overestimation, which results in the most evenly distributed and lowest forecast errors (see Figure 2), seems to be beneficial for constructing market-neutral MVPs, regardless of the forecast models' ability to capture nonlinear and interactive effects.

Panel B of Table 5 adds the time series averages of minimum and maximum portfolio weights assigned through the optimization process and the percentage of negative weights within the optimized port-

[37] In our empirical setting, with a substantially longer sample period and a one-year rolling window to forecast betas (instead of half-year rolling windows as in Cosemans et al., 2016), we find that their $hybrid$ model yields a significant ex-post portfolio beta. Following their specification more closely (with the same sample period and rolling windows for the calculation of market and idiosyncratic variances), our results are qualitatively similar to theirs.

folios. None of the forecast models leads, on average, to extreme negative or positive weights. Any exposure to systematic market risk (ex-post beta that is significantly different from zero) is therefore not driven by extreme positions in just a few stocks within the portfolio. However, we find that the forecast models that produce less extreme and less volatile beta estimates (according to the results from Table 2) lead to more pronounced portfolio weights on average (-2.28% to 5.76% for the $elanet$ model vs. -1.04% to 3.54% for the ols_5y_m model). Hence, a higher confidence in beta estimates also appears to allow the optimizer to place more weight on single stocks within the portfolio.

To gain a better understanding of the results in Table 5, we relate the weights of stocks in the optimized portfolios to their beta forecasts in Figure C1 (see Internet Appendix, Section C for details). Typically, based on the goal of creating a market-neutral portfolio, the optimizer assigns positive weights to low-beta stocks, and negative weights to high-beta stocks. Figure C1 also shows how the machine learning-based approaches outperform other estimators in the portfolio optimization. Compared to established forecast models, they reduce forecast errors for those stocks that receive the highest weights during the optimization process, i.e., firms with the lowest and highest ex-ante beta estimates. Therefore, machine learning-based approaches benefit twice from any reduction in forecast errors within the extreme-beta deciles. Given the results from Table 2, they provide the least extreme and also least volatile beta estimates, which additionally enables the optimizer to place more weight on stocks within these decile portfolios.

[Insert Table 5 here]

5.1.6. Market-neutral anomaly portfolios

Another approach to evaluating the economic gains of statistically more accurate forecasts is to investigate the average ex-post realized betas of ex-ante market-neutral long–short anomaly portfolios. This setup mimics the strategy of another hedge fund that aims for significant exposure to a specific stock market anomaly, e.g., to exploit the size effect, while taking on zero market risk. We consider commonly used anomaly variables, i.e., size (me), book-to-market ratio (bm), momentum (mom), and illiquidity ($illiq$).[38]

[38] Note that the patterns identified and their implications are qualitatively similar for most other anomaly variables.

We also inspect the anomaly based on each forecast model's predicted beta (β^F), i.e., the Betting-Against-Beta (BAB) factor of Frazzini and Pedersen (2014).

In particular, the portfolio optimization extends the procedure outlined in Hollstein, Prokopczuk, and Wese Simen (2019). In a first step, at the end of each month t, we sort stocks into quintile portfolios based on the respective anomaly variables.[39] We then use the out-of-sample beta forecasts of the stocks within the top (H) and bottom (L) quintiles to construct the long and short portfolios, respectively. We require the ex-ante predicted portfolio beta for both quintile portfolios to be one. To this end, we select portfolio weights that solve the following problem, separately for each beta estimator:

$$\min_{w_t} \ \sum_i \left(w_{i,t} - w_{i,t}^*\right)^2 \ \text{s.t.} \tag{10}$$

$$w_{i,t} \geq 0$$

$$\sum_i w_{i,t}\beta_{i,t+1|t}^F = 1$$

The optimizer aims to minimize the sum of squared deviations from the original market capitalization-based weights $w_{i,t}^*$ within the respective portfolios.[40] The first constraint implies that the quintile portfolios are each long-only, while the second states that the ex-ante predicted portfolio betas should be one. Combining the long and short portfolios (by multiplying the calculated weights for the short portfolio by –1) results, by construction, in a long–short anomaly portfolio (HML) that is ex-ante market neutral.[41] In line

[39] The next step in the original approach is to compute the ex-ante portfolio betas for the long and short portfolios, separately for each beta estimator. Ex-ante portfolio betas are measured as the value-weighted averages of beta forecasts within the respective portfolio. The final step is to create the long–short anomaly portfolios that are ex-ante market neutral, i.e., to solve for the weight w_t that fulfills the equation $w_t\bar{\beta}_{t+k|t}^{F,long} - \bar{\beta}_{t+k|t}^{F,short} = 0$. Ex-post portfolio betas are the value-weighted averages of realized betas over the next year. The advantage of our approach is that we can separately examine the performance of the long-only and long–short portfolios separately. This allows us to evaluate further strategies of long-only investors. The original approach focuses only on the long–short portfolio, where forecast errors may cancel out. Note that our results for the long–short portfolios are qualitatively similar when following Hollstein, Prokopczuk, and Wese Simen's (2019) approach.

[40] This approach helps ensure that the resulting long and short portfolios are indeed investable for a hedge fund. In other words, it leads to portfolio weights, especially for (very) small stocks, that are not too large, while avoiding straight out excluding microcaps from the stock universe.

[41] Alternatively, investors can use the long-only portfolios. These can then be made market neutral by taking a short position in exchange-traded funds or futures based on some market proxy.

with ex-ante portfolio betas, ex-post portfolio betas are the weighted averages of realized betas over the next year.

Table 6 reports the time series averages of ex-post portfolio betas (β) for the long–short anomaly portfolios and the *t*-statistics (in parentheses) testing the null hypothesis that these betas are zero. For the *t*-tests, we use Newey and West (1987) standard errors with eleven lags. It also presents the numbers for the long and short anomaly portfolios separately. In this case, the null hypotheses are that these betas are one. We find that the machine learning models are the only approaches leading to long–short portfolios that are truly market neutral ex post for all stock market anomalies. For example, the rf model yields realized portfolio betas between 0.00 for the value–growth anomaly and 0.07 for the momentum anomaly. According to the respective Newey and West (1987) t-statistics between -0.10 and 1.63, these betas are insignificantly different from zero.

Moreover, tree-based models and neural networks seem to slightly outperform linear regressions (in terms of lower realized portfolio betas and *t*-statistics for most stock market anomalies). Importantly, the long–short portfolios based on all established forecast models realize ex-post betas that are statistically significant at least at the 10% level for multiple stock market anomalies, and are also economically large. Considering the size anomaly as an illustrative example, unlike each machine learning model, nearly all traditional approaches realize statistically significant ex-post betas, ranging from 0.18 (*t*-statistic of 4.04) for the $long\text{-}memo$ model to 0.58 (*t*-statistic of 9.26) for the ols_5y_m model.

However, statistically insignificant long–short portfolio betas might be misleading if the realized betas for the underlying long and short portfolios are biased in the same direction, i.e., if they are larger or smaller than one by a similar amount. Therefore, to gain a better understanding of the long–short portfolio betas, we now inspect them separately. Overall, they confirm the aforementioned findings. The machine learning models are the only ones leading to long and short portfolios with realized betas that are insignificantly different from one for nearly all stock market anomalies and both the long and short side. Tree-based models and neural networks seem to slightly outperform linear regressions. The realized betas for the long and short portfolios based on traditional forecast models show that the significant exposure to market risk of the long–short portfolios is mostly driven by *either* the long *or* short side (but not both).

The traditional approaches mostly fail to construct portfolios with realized betas insignificantly different from one for small and illiquid stocks. In particular, the beta of the portfolio consisting of winner stocks appears to be the hardest to predict accurately, even for machine learning-based estimators (in line with the only marginal value-added of random forests for this type of stocks; see Figure 3). In addition, the portfolios that consist of stocks with high or low beta estimates realize ex-post betas that are statistically significant and economically large for all established approaches. This is because all of them are subject to the problem of under- and overestimation for low- and high-beta stocks (see Figure 2) and fail to produce accurate forecasts for stocks with extreme firm characteristics (including small and illiquid stocks; see Figure 3).

In summary, the results from the portfolio allocation exercises highlight the practical consequences of inaccurate beta estimates. An investment strategy that is meant to be market neutral ex ante may still exhibit a significant exposure to market risk ex post. We find that established estimation techniques fail to produce truly ex-post market-neutral portfolios due to systematic errors in beta forecasts. The machine learning-based approaches, however, exhibit the lowest forecast errors for those stocks that are most relevant during both optimization processes. This allows for portfolios that are truly market neutral ex post.

[Insert Table 6 here]

5.2. Characteristics and functioning scheme of machine learning-based estimators

Studying the forecast models' ability to predict out-of-sample market betas, we find that regression trees and neural networks outperform established beta estimators and linear regressions in terms of predictive performance. We next focus on determining *how* these techniques, which are often referred to as "black boxes", achieve such superior predictive performance. In the following subsections, we therefore address the black box issue in estimating market betas by investigating the characteristics and functioning scheme of the random forests. Compared to the gradient boosted regression trees and neural networks, they exhibit the lowest forecast errors (both on average and over time, see Table 3) and yield the best bias–inefficiency trade-off (see Table 4). At the same time, they avoid the under- and overestimation problem (see Figure 2), produce accurate forecasts for stocks with extreme firm characteristics (see Figure 3), and allow for portfolios that are truly market neutral ex post (see Tables 5 and 6). This is why we focus on the random forests (the rf model) throughout our subsequent empirical analysis.[42] In particular, we inspect changes in the inherent model complexity over time, decompose predictions into the contributions of individual variables using relative variable importance metrics, and explore patterns of nonlinear and interactive effects in the relationship between predictor variables and expected market betas.

5.2.1. Model complexity

Since we re-estimate the random forests on an annual basis, it is interesting to gauge whether model complexity changes over time or rather remains stable. The rf model is non-parametric and tree-based, and thus we take the number of trees added to the ensemble prediction (mc) to measure model complexity at each re-estimation date. For example, a large number of trees points towards high model complexity, i.e., the respective random forest needs information from multiple different bootstrap-replicated trees to optimally explain the cross-sectional variation in realized betas within the validation sample. A smaller number of trees, in turn, indicates that a less complex model is sufficient to meet the objective of minimizing the validation error. To contextualize the model complexity measure, we compute the time series means for monthly MSEs within each validation sample (mse_vali) and each test sample (mse_test). We also relate the rf model's mse_test metrics to those obtained for the ols_1y_d model. In particular, we compute the percentage difference in test-sample MSEs (mse_test_pd) as one minus the MSE of the random forests divided by that of the respective benchmark model. Again, we follow the convention that positive differences indicate superior predictive performance of the random forests relative to the ols_1y_d model.

Figure 4 illustrates the rf model's complexity at each re-estimation date and its association with forecast errors by plotting mc over time, together with mse_vali (Panel A), mse_test (Panel B), and

[42] Note that the patterns identified and their implications are qualitatively similar for both the $gbrt$ model and neural networks.

mse_test_pd (Panel C), respectively. It also visualizes the NBER recession periods (shaded grey). In accordance with Gu, Kelly, and Xiu (2020) and Drobetz and Otto (2021), the graphs suggest that model complexity varies substantially over time. For example, many trees are required at multiple re-estimation dates during the 1993–2002 period (with a global peak in December 2001). In contrast, model complexity is much lower during the subsequent years, i.e., the 2002–2019 period.

We identify a co-movement between mc and mse_vali, which is implied by a time series correlation of 0.92. A t-test significantly rejects the null hypothesis that the correlation coefficient is zero (unreported). Therefore, the model complexity varies substantially over time in the context of time-varying stock market conditions. In particular, we find that the rf model's complexity is high during difficult-to-predict periods (large validation errors). When stock markets are easier to predict, much fewer trees are necessary for the ensemble prediction to minimize the validation error.

It is interesting to note that we do not find synchronicity between the mc and mse_test metrics (time series correlation of -0.04). According to an insignificant t-statistic, we cannot reject the null hypothesis that this correlation is zero (unreported) and conclude that neither high- nor low-complexity forecasts systematically coincide with high nor low forecast errors within the test samples. This insignificance highlights the need to determine the hyperparameters governing the model complexity adaptively from the sample data, rather than forcing it to remain constant by pre-setting them.

We also identify a co-movement between mc and mse_test_pd, according to a time series correlation of 0.35 and a significant t-statistic (unreported). The random forests can reduce the test-sample forecast error relative to the ols_1y_d model most of the time during the sample period (as discussed for Figure 1). However, the relative outperformance appears to be substantially stronger when the rf model's complexity is high. In the context of the aforementioned co-movement between mc and mse_vali, we conclude that the more sophisticated machine learning-based approach is particularly helpful for providing accurate beta estimates when stock markets are difficult to predict, e.g., during or right after the NBER recessions.

[Insert Figure 4 here]

5.2.2. Variable importance

Next, since the inherent model complexity is time-varying, it is instructive to explore whether each predictor's contribution to the overall forecast ability of the random forests also changes over time.[43] To this end, we calculate the variable importance matrix in a two-step approach, separately for each re-estimation date. First, we compute the absolute variable importance as the increase in value-weighted MSE from setting all values of a given predictor to its uninformative median value within the training sample. Second, we normalize the absolute variable importance measures to sum to 1, signaling the relative contribution of each variable to the rf model.

Panel A of Figure 5 depicts the time series average of relative variable importance measures for the predictor categories introduced in Section 3.[44] We find that historical betas are, on average, most informative, followed by technical indicators and accounting-based predictors, which, in turn, appear to be much more important than macroeconomic indicators. Historical betas in total account for more than 60% of the total variable importance. This is as expected because realized betas are highly persistent and have long-memory properties (as described in Section 4.1). Technical indicators, however, are also important at roughly 25%. This suggests that the time variation in market betas might be driven more strongly by changes in the firm fundamentals than by changes in the underlying economic conditions.

Panel B of Figure 5 presents the time series average of relative variable importance measures for the ten most influential predictors. We observe that random forests place most of their weights on five variables, which, ignoring industry classifications, leaves twenty-nine variables with notably lower importance. Consistent with the above, the most influential predictors are the three sample estimates of beta, with the largest weight placed on the one-year daily estimates, followed by the three-month daily and five-year monthly

[43] For the sake of brevity, we exclude the industry classifiers throughout the variable importance tests because they are ultimately among the least informative predictors. Note that the patterns identified and their implications are qualitatively similar when including them.

[44] To this end, we simultaneously set all values of all predictors within each category to their uninformative median value within the training sample before computing the absolute and relative variable importance metrics as described above.

counterparts. In addition, a firm's turnover (to) and size (me) are also highly important. In total, the average relative contribution of the top five variables to the rf model sums up to 84.57%.[45]

[Insert Figure 5 here]

Because the overall relative variable importance measures only mirror a predictor's mean contribution to the random forests' predictive performance, we also investigate relative variable importance metrics over time. Volatile metrics would indicate that all covariates in the predictor set are essential; stable figures would mean we should remove uninformative predictors permanently, as they may decrease the rf model's signal-to-noise ratio. We focus on the twenty-nine least important predictors for which removal is a consideration.

To identify time variability in relative variable importance measures for the set of these covariates, we omit the remaining variables prior to normalizing the absolute variable importance measures to sum to 1 at each re-estimation date. Panel C of Figure 5 presents these relative variable importance metrics over the sample period. Although we still observe sizable differences, the graph indicates that the relative variable importance metrics change substantially over time. Therefore, each predictor is an important contributor to the random forests (albeit to varying degrees).

Moreover, we must assess the aggregate contribution of the twenty-nine least important covariates to the rf model's predictive performance. At each re-estimation date, we therefore compute the fraction of aggregate absolute variable importance (sum of increases in value-weighted MSE across all variables) that is attributed to this subset of predictors. Panel D of Figure 5 visualizes this fraction over the sample period, together with the NBER recession periods (shaded grey). While the aggregate contribution is exceptionally low during the *Great Recession* (2008:01–2009:06), it peaks at 27.70% shortly afterward, in December 2010. Accordingly, even these covariates appear to be material for the random forests, at least for some periods. The findings from Figure 5 do *not* indicate that we should remove specific predictors. On the

[45] Using a smaller and slightly different set of predictor variables, Jourovski et al. (2020) also find that a lagged beta is the most influential variable. However, they do not include betas at several lags, which turns out to be very important in our empirical analysis.

contrary, these results justify the use of our comprehensive set of predictors.[46] While using information from multiple different sources seems to be essential for the predictive performance of the machine learning-based forecast models, it makes them prone to potential misspecification. To this end, we only choose to include those covariates that have been shown, theoretically or empirically, to explain and forecast time-varying market betas. This delivers a partial explanation for the result that we should *not* remove specific predictors: we simply refrain from including too many likely irrelevant ones in the first place.

5.2.3. Nonlinearity and interactions

Our results thus far suggest that regression trees and neural networks are superior to established beta estimators. Both machine learning-based model families are designed to capture nonlinearity and interactions in the relationship between predictors and future market betas. Importantly, they also outperform linear regressions, which incorporate the same set of covariates. Consequently, a large part of this outperformance may be attributable to their ability to utilize nonlinear and interactive patterns. We thus investigate whether the best-performing machine learning approach, random forests (rf), indeed captures nonlinearity and interactions. For comparison, we contrast the results with beta estimates obtained from simple linear regressions (lm).[47]

We first examine the marginal association between a single predictor and its beta estimates ($\beta^{F}_{i,t+k|t}$, with $k = 12$). To illustrate, we select a firm's sample estimate of beta from rolling regressions using a one-year window of daily returns ($ols_{1y,d}$). It is the most influential predictor in our empirical analysis (see Figure 5, Panel B), and helps address the problem of under- and overestimation inherent to estimating time-

[46] To be conservative, we compare the statistical and economic predictive performance of the original rf model with versions that only consider the top five predictors in terms of their overall relative variable importance. Out-of-sample tests (unreported) are identical to the tests shown in Section 5.1. We find that this version does not exhibit substantial outperformance relative to the full-fledged rf model in any of the tests. Hence, we choose not to remove unconditionally less informative variables from the predictor set, and instead consider each predictor as informative (albeit to varying degrees). Additionally, we caution that the pre-estimation variable selection based on relative importance metrics derived from the entire sample period could lead to foresight bias, undermining the credibility of any out-of-sample tests.

[47] Note that the patterns identified and their implications are qualitatively similar when comparing regression trees and neural networks to estimates obtained from penalized linear regressions ($elanet$). This emphasizes that indeed the ability to utilize nonlinear and interactive patterns leads to the outperformance of regression trees and neural networks.

varying market betas (see Figure 2).[48] We apply the following procedure to visualize the average effect of $ols_{1y,d}$ on $\beta^F_{i,t+k|t}$. At each re-estimation date, we set all predictors to their uninformative median values (within the training sample), and the industry dummies to the value of zero. We then vary $ols_{1y,d}$ across the $(-1,+3)$ interval and compute beta forecasts for this artificial test sample. Finally, we average beta forecasts across re-estimation dates.

Panel A of Figure 6 illustrates the marginal association between $ols_{1y,d}$ and $\beta^F_{i,t+k|t}$. We add a histogram to the visualization that depicts the historical distribution of $ols_{1y,d}$. This allows us to assess the relevance that differences in predictions obtained from the lm and rf models have on the overall forecast results. As expected, higher values for one-year rolling betas lead to higher beta estimates for both model families. By construction, we identify an increasing linear relationship between $ols_{1y,d}$ and $\beta^F_{i,t+k|t}$ for the lm model. At the center of the distribution, approximately within the (+0.3, +1.5) interval, the marginal association between $ols_{1y,d}$ and the beta forecasts of the rf model is also close to linear. However, outside this interval, the rf model provides nearly constant predictions, leading to an overall S-shaped relationship. In contrast, the lm model, by construction, must stick with the increasing linear relationship. This leads to less extreme beta estimates for random forests (compared to simple linear regressions) when $ols_{1y,d}$, ceteris paribus, becomes small or large. Because a material share of observations lie within these outer areas of the historical distribution, differences in predictions are highly relevant. This highlights the need to allow for nonlinear impacts of the predictor variables. The S-shaped relationships are also observed for other predictors (unreported), for example, a firm's turnover (to) and size (me). Taken together, these findings provide an explanation for our previous finding that random forests provide less extreme beta forecasts in general, while also avoiding the systematic underestimation of low-beta stocks and systematic overestimation of high-beta stocks (see Figure 2).[49] These results also help explain the outperformance of random forests over established and linear approaches in the sense of lower forecast errors (see Tables 3 and 4) and their dominance over established forecast models in constructing market-neutral MVPs (see Table 5).

[48] Note that the patterns identified and their implications are qualitatively similar for other predictor variables.

[49] Note that the under- and overestimating pattern for low- and high-beta stocks is also visible for the lm model, although it is somewhat less pronounced.

Next, we investigate between-predictor interactions in estimating future market betas, referring again to $ols_{1y,d}$ as our baseline covariate. We select me, another highly influential predictor in our empirical analysis (see Figure 5, Panel B), as our interactive counterpart, and replicate the procedure outlined above. But, in this case, we compute estimated betas for four different levels of me (across the $(-2, +2)$ interval). The interactive effect of $ols_{1y,d}$ and me on $\beta^F_{i,t+k|t}$ is illustrated in Panel B of Figure 6. Low and high levels for me are marked with red and green lines, respectively. Conceptually, if there is no interaction or if the model is unable to capture such interactions, computing estimated betas for different levels of me would simply shift the lines from Panel A of Figure 6 up- or downward in a parallel fashion. The distance between the lines would then be identical for any given value of $ols_{1y,d}$. This pattern is apparent for simple linear regressions, as no pre-specified interaction term, e.g., $ols_{1y,d} \times me$ for the interaction between $ols_{1y,d}$ and me is included as a predictor in the OLS-based framework. The lines are shifted upward with increasing me, which indicates that an increase in me also increases $\beta^F_{i,t+k|t}$, but independently of $ols_{1y,d}$. Unlike the lm model, random forests uncover the interactive effect between a firm's historical beta and size in estimating future betas.[50] While the lines are also shifted upward for larger levels of me, the strength of the shift is much more pronounced for larger values of $ols_{1y,d}$. Therefore, the effect of a firm's size on its future beta estimate appears to be much stronger if this firm was more sensitive to systematic market risk over the past year.

As this example demonstrates, incorporating the nonlinearity of single predictors and between-predictor interactions are both imperative and foundation for the superior predictive performance of random forests. These effects provide an explanation for the advantageousness of machine learning methods over established and linear benchmark models.

[Insert Figure 6 here]

[50] Despite being slightly less pronounced, random forests also reveal the interactive effects between other firm characteristics in estimating future betas, e.g., between a firm's historical beta and turnover.

6. **Conclusion**

Using a large universe of U.S. stocks and a long and recent sample period, we compare the predictive performance of machine learning-based beta estimators (linear regressions, tree-based models, and neural networks) to that of several established benchmark beta estimators. We find that machine learning techniques outperform established approaches from both a statistical and an economic perspective. Random forests perform the best overall, but gradient boosted regression trees and neural networks also work well. In particular, machine learning methods yield the lowest forecast errors. On top of that, they are the only forecast model family able to generate truly ex-post market-neutral portfolios.

One important economic reason for the outperformance of machine learning methods is that they can capture the information content from a large set of firm characteristics that appear to impact betas. Importantly, however, random forests, gradient boosted regression trees, and neural networks also outperform linear regressions, which incorporate the same set of covariates. We show that a large part of this outperformance is due to their ability to utilize nonlinear and interactive patterns, which provides a second, complementary explanation for the advantageousness of machine learning methods for beta estimation.

References

Amihud, Y., and Mendelson, H. (2000). The Liquidity Route to a Lower Cost of Capital. *Journal of Applied Corporate Finance, 12*(4), 8-25.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Wu, G. (2006). Realized Beta: Persistence and Predictability. In T. Fomby, and D. Terrel, *Advances in Econometrics: Econometric Analysis of Economic and Financial Time Series.* Amsterdam, Netherlands: Elsevier.

Andersen, T. G., Bollerslev, T., and Meddahi, N. (2005). Correcting the Errors: Volatility Forecast Evaluation Using High-Frequency Data and Realized Volatilities. *Econometrica*, *73*(1), 279-296.

Ang, A., and Chen, J. (2007). CAPM Over the Long Run: 1926–2001. *Journal of Empirical Finance*, *14*(1), 1-40.

Bailey, D. H., Borwein, J. M., de Prado, M. L., and Zhu, Q. J. (2014). Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance. *Notices of the American Mathematical Society, 61*(5), 458-471.

Bailey, D. H., Borwein, J. M., de Prado, M. L., and Zhu, Q. J. (2017). The Probability of Backtest Overfitting. *Journal of Computational Finance, 20*(4), 1460-1559.

Bali, T. G., Goyal, A., Huang, D., Jiang, F., and Wen, Q. (2022). The Cross-Sectional Pricing of Corporate Bonds Using Big Data and Machine Learning. *Working Paper.*

Barber, B. M., Huang, X., and Odean, T. (2016). Which Factors Matter to Investors? Evidence From Mutual Fund Flows. *The Review of Financial Studies*, *29*(10), 2600-2642.

Beaver, W., Kettler, P., and Scholes, M. (1970). The Association Between Market Determined and Accounting Determined Risk Measures. *The Accounting Review, 45*(4), 654-682.

Becker, J., Hollstein, F., Prokopczuk, M., and Sibbertsen, P. (2021). The Memory of Beta. *Journal of Banking & Finance, 124*(1), 106026.

Berk, J. B., and van Binsbergen, J. H. (2016). Assessing Asset Pricing Models Using Revealed Preference. *Journal of Financial Economics, 119*(1), 1-23.

Bianchi, D., Büchner, M., and Tamoni, A. (2021). Bond Risk Premiums With Machine Learning. *The Review of Financial Studies*, *34*(2), 1046-1089.

Black, F., Jensen, M. C., and Scholes, M. S. (1972). The Capital Asset Pricing Model: Some Empirical Tests. In M. C. Jensen, *Studies in the Theory of Capital Markets.* New York (NY), U.S.: Praeger.

Blume, M. E. (1975). Betas and Their Regression Tendencies. *The Journal of Finance*, *30*(3), 785–795.

Bollerslev, T., Engle, R. F., and Wooldridge, J. M. (1988). A Capital Asset Pricing Model With Time-Varying Covariances. *Journal of Political Economy, 96*(1), 116-131.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32.

Bucci, A. (2020). Realized Volatility Forecasting With Neural Networks. *Journal of Financial Econometrics*, *18*(3), 502-531.

Campbell, J. Y., Lettau, M., Malkiel, B. G., and Xu, Y. (2001). Have Individual Stocks Become More Volatile? An Empirical Exploration of Idiosyncratic Risk. *The Journal of Finance, 56*(1), 1-43.

Chan, L. K., Karceski, J., and Lakonishok, J. (1999). On Portfolio Optimization: Forecasting Covariances and Choosing the Risk Model. *The Review of Financial Studies, 12*(5), 937-974.

Chincarini, L. B., Kim, D., and Moneta, F. (2020). Beta and Firm Age. *Journal of Empirical Finance, 58*(C), 50-74.

Christensen, K., Siggaard, M., and Veliyev, B. (2021). A Machine Learning Approach to Volatility Forecasting. *CREATES Research Paper*, 1-46.

Cochrane, J. H. (1998). Where is the Market Going? Uncertain Facts and Novel Theories. *NBER Working Paper No. 6207*.

Cochrane, J. H. (2011). Presidential Address: Discount Rates. *The Journal of Finance, 66*(4), 1047-1108.

Connor, G., Hagmann, M., and Linton, O. (2012). Efficient Semiparametric Estimation of the Fama–French Model and Extensions. *Econometrica, 80*(2), 713-754.

Connor, G., and Linton, O. (2007). Semiparametric Estimation of a Characteristic-Based Factor Model of Common Stock Returns. *Journal of Empirical Finance*, *14*(5), 694-717.

Cosemans, M., Frehen, R., Schotman, P. C., and Bauer, R. (2016). Estimating Market betas Using Prior Information Based on Firm Fundamentals. *The Review of Financial Studies, 29*(4), 1072-1112.

Daniel, K., Mota, L., Rottke, S., and Santos, T. (2020). The Cross-Section of Risk and Returns. *The Review of Financial Studies, 33*(5), 1927-1979.

Dichtl, H., Drobetz, W., and Otto, T. (2021). Forecasting Market Crashes via Machine Learning: Evidence from European Stock Markets. *Working Paper*.

Diebold, F., and Mariano, R. (1995). Comparing Predictive Accuracy. *Journal of Business & Economic Statistics, 13*(3), 253-263.

Dietterich, T. (2000). Ensemble Methods in Machine Learning. *Lecture Notes in Computer Science: Multiple Classifier Systems*. Berlin, Germany: Springer.

Donaldson, R. G., and Kamstra, M. (1997). An Artificial Neural Network-GARCH Model For International Stock Return Volatility. *Journal of Empirical Finance, 4*(1), 17-46.

Drobetz, W., Haller, R., Jasperneite, C., and Otto, T. (2019). Predictability and the Cross Section of Expected Returns: Evidence from the European Stock Market. *Journal of Asset Management, 20*(7), 508-533.

Drobetz, W., and Otto, T. (2021). Empirical Asset Pricing via Machine Learning: Evidence from the European Stock Market. *Journal of Asset Management*, *22*(7), 507-538.

Fan, J., Liao, Y., and Wang, W. (2016). Projected Principal Component Analysis in Factor Models. *Annals of Statistics*, *44*(1), 219-254.

Fama, E. F., and French, K. R. (1992). The Cross-Section of Expected Stock Returns. *The Journal of Finance, 47*(2), 427-465.

Fama, E. F., and French, K. R. (1997). Industry Costs of Equity. *Journal of Financial Economics, 43*(2), 153-193.

Fama, E. F., and French, K. R. (2004). New Lists: Fundamentals and Survival Rates. *Journal of Financial Economics, 73*(2), 229-269.

Fama, E. F., and French, K. R. (2008). Dissecting Anomalies. *The Journal of Finance, 63*(4), 1653-1678.

Fama, E. F., and MacBeth, J. D. (1973). Risk, Return, and Equilibrium: Empirical Tests. *Journal of Political Economy, 81*(3), 607-636.

Fernandes, M., Medeiros, M. C., and Scharth, M. (2014). Modeling and Predicting the CBOE Market Volatility Index. *Journal of Banking & Finance, 40*(1), 1-10.

Ferson, W. E., and Harvey, C. R. (1999). Conditioning Variables and the Cross Section of Stock Returns. *The Journal of Finance, 54*(4), 1325-1360.

Fink, J., Fink, K. E., Grullon, G., and Weston, J. P. (2010). What Drove the Increase in Idiosyncratic Volatility During the Internet Boom? *Journal of Financial and Quantitative Analysis, 45*(5), 1253-1278.

Frazzini, A., and Pedersen, L. H. (2014). Betting Against Beta. *Journal of Financial Economics, 111*(1), 1-25.

Ghysels, E., and Jacquier, E. (2006). Market Beta Dynamics and Portfolio Efficiency. *Working Paper*.

Graham, J. R. (2022). Presidential Address: Corporate Finance and Reality. *The Journal of Finance, 77*(4), 1975-2049.

Graham, J. R., and Harvey, C. R. (2001). The Theory and Practice of Corporate Finance: Evidence From the Field. *Journal of Financial Economics, 60*(2-3), 187-243.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies, 33*(5), 2223-2273.

Gu, S., Kelly, B., and Xiu, D. (2021). Autoencoder Asset Pricing Models. *Journal of Econometrics, 222*(1), 429-450.

Gulen, H., Xing, Y., and Zhang, L. (2011). Value Versus Growth: Time-Varying Expected Stock Returns. *Financial Management, 40*(2), 381-407.

Hansen, L., and Salamon, P. (1990). Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(10), 993-1001.

Hansen, P. R., and Lunde, A. (2006). Consistent Ranking of Volatility Models. *Journal of Econometrics, 131*(1-2), 97-121.

Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The Model Confidence Set. *Econometrica*, 79(2), 453-497.

Harvey, C. R., and Liu, Y. (2014). Evaluating Trading Strategies. *The Journal of Portfolio Management, 40*(5), 108-118.

Harvey, C. R., and Liu, Y. (2015). Backtesting. *The Journal of Portfolio Management, 42*(1), 13-28.

Harvey, C. R., Liu, Y., and Zhu, H. (2016). ... and the Cross-Section of Expected Returns. *The Review of Financial Studies, 29*(1), 5-68.

Hillebrand, E., and Medeiros, M. C. (2010). The Benefits of Bagging for Forecast Models of Realized Volatility. *Econometric Reviews, 29*(5-6), 571-593.

Hollstein, F., and Prokopczuk, M. (2016). Estimating Beta. *Journal of Financial and Quantitative Analysis, 51*(4), 1437-1466.

Hollstein, F., Prokopczuk, M., and Wese Simen, C. (2019). Estimating Beta: Forecast Adjustments and the Impact of Stock Characteristics for a Broad Cross-Section. *Journal of Financial Markets, 44*(1), 91-118.

Ioffe, S., and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning*, 448-456.

Jacobs, M. T., and Shivdasani, A. (2012). Do You Know Your Cost of Capital? *Harvard Business Review, 90*(7), 118–124.

Jacoby, G., Fowler, D. J., and Gottesman, A. A. (2000). The Capital Asset Pricing Model and the Liquidity Effect: A Theoretical Approach. *Journal of Financial Markets, 3*(1), 69-81.

Jagannathan, R., and Ma, T. (2003). Risk Reduction in Large Portfolios: Why Imposing the Wrong Constraints Helps. *The Journal of Finance, 58*(4), 1651-1683.

Jagannathan, R., and Wang, Z. (1996). The Conditional CAPM and the Cross-Section of Expected Returns. *The Journal of Finance, 51*(1), 3-53.

Jourovski, A., Dubikovskyy, V., Adell, P., Ramakrishnan, R., and Kosowski, R. (2020). Forecasting Beta Using Machine Learning and Equity Sentiment Variables. In E. Jurczenko, *Machine Learning for Asset Management: New Developments and Financial Applications*. London, U.K.: Wiley-ISTE.

Jovanovic, B., and Rousseau, P. L. (2001). Why Wait? A Century of Life Before IPO. *American Economic Review, 91*(2), 336-341.

Karolyi, G. A. (1992). Predicting Risk: Some New Generalizations. *Management Science, 38*(1), 57-74.

Kelly, B., Moskowitz, T. J., and Pruitt, S. (2021). Understanding Momentum and Reversal. *Journal of Financial Economics*, *140*(3), 726-743.

Kelly, B. T., Pruitt, S., and Su, Y. (2019). Characteristics are Covariances: A Unified Model of Risk and Return. *Journal of Financial Economics, 134*(3), 501-524.

Kim, S., Korajczyk, R. A., and Neuhierl, A. (2020). Arbitrage Portfolios. *The Review of Financial Studies, 34*(6), 2813-2856.

Ledoit, O., and Wolf, M. (2003). Improved Estimation of the Covariance Matrix of Sock Returns with an Aapplication to Portfolio Selection. *Journal of Empirical Finance, 10*(5), 603-621.

Leippold, M., Wang, Q., and Zhou, W. (2021). Machine Learning in the Chinese Stock Market. *Journal of Financial Economics*, *145*(2), 64-82.

Leitch, G., and Tanner, J. E. (1991). Economic Forecast Evaluation: Profits Versus the Conventional Error Measures. *The American Economic Review, 81*(3), 580-590.

Levi, Y., and Welch, I. (2017). Best Practice for Cost-of-Capital Estimates. *Journal of Financial and Quantitative Analysis, 52*(2), 427-463.

Lewellen, J. (2015). The Cross-Section of Expected Stock Returns. *Critical Finance Review, 4*(1), 1-44.

Lintner, J. (1965). Security Prices, Risk, and Maximal Gains From Diversification. *The Journal of Finance, 20*(4), 587-615.

Lo, A. W., and MacKinlay, A. C. (1990). Data-Snooping Biases in Tests of Financial Asset Pricing Models. *The Review of Financial Studies, 3*(3), 431-467.

Loughran, T., and Ritter, J. (2004). Why Has IPO Underpricing Changed Over Time? *Financial Management, 33*(3), 5-37.

Luong, C., and Dokuchaev, N. (2018). Forecasting of Realised Volatility With the Random Forests Algorithm. *Journal of Risk and Financial Management, 11*(4), 61.

Masters, T. (1993). *Practical Neural Network Recipes in C++*. Burlington (MA), U.S.: Morgan Kaufmann Publishers.

Mincer, J. A., and Zarnowitz, V. (1969). The Evaluation of Economic Forecasts. In J. A. Mincer, *Economic Forecasts and Expectations: Analysis of Forecasting Behavior and Performance.* Cambridge (MA), U.S.: NBER.

Mittnik, S., Robinzonov, N., and Spindler, M. (2015). Stock Market Volatility: Identifying Major Drivers and the Nature of Their Impact. *Journal of Banking & Finance, 58*(1), 1-14.

Mossin, J. (1966). Equilibrium in a Capital Asset Market. *Econometrica, 34*(2), 768-783.

Newey, W. K., and West, K. D. (1987). Hypothesis Testing With Efficient Method of Moments Estimation. *International Economic Review, 28*(3), 777-787.

Novy-Marx, R. (2011). Operating Leverage. *Review of Finance, 15*(1), 103-134.

Pastor, L., and Stambaugh, R. F. (1999). Costs of Equity Capital and Model Mispricing. *The Journal of* Finance, 54(1), 67-121.

Patton, A. J. (2011). Volatility Forecast Comparison Using Imperfect Volatility Proxies. *Journal of Econometrics, 160*(1), 246-256.

Petkova, R., and Zhang, L. (2005). Is Value Riskier Than Growth? *Journal of Financial Economics, 78*(1), 187-202.

Rahimikia, E., and Poon, S. H. (2020). Machine Learning for Realised Volatility Forecasting. *Working Paper*.

Schorfheide, F., and Wolpin, K. I. (2012). On the Use of Holdout Samples for Model Selection. *American Economic Review*, *102*(3), 477-481.

Sharpe, W. F. (1964). Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk. *The Journal of Finance*, *19*(3), 425-442.

Vasicek, O. (1973). A Note on Using Cross-Sectional Information in Bayesian Estimation of Market betas. *The Journal of Finance, 28*(5), 1233-1239.

Van Binsbergen, J. H., Han, X., and Lopez-Lira, A. (2021). Man vs. Machine Learning: The Term Structure of Earnings Expectations and Conditional Biases. *NBER Working Paper No. 27843.*

Welch, I. (2019). Simpler Better Market Betas. *NBER Working Paper No. 26105*.

Welch, I., and Goyal, A. (2008). A Comprehensive Look at the Empirical Performance of Equity Premium Prediction. *The Review of Financial Studies*, *21*(4), 1455-1508.

West, K. D. (2006). Forecast Evaluation. In G. Elliott, C. Granger, and A. Timmermann, *Handbook of Economic Forecasting*. Amsterdam, Netherlands: Elsevier.

Tables

Table 1
Variable descriptions and definitions

This table presents the descriptions and definitions for each of the eighty-one predictors used in the empirical analysis. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period. Market data from CRSP are assumed to become public immediately, while fundamental data from Compustat are assumed to be known four months after the end of the fiscal year.

#	Predictor	Description	Definition
1) Predictors based on accounting information			
1	age	Age	*Log* number of years since first inclusion in CRSP
2	at	Total assets	*Log* book value of total assets
3	bm	Book-to-market ratio	*Log* ratio of book and market value of equity
4	capturn	Capital turnover	*Log* ratio of net sales to lagged book value of total assets
5	divpay	Dividend payout ratio	Ratio of dividends paid during the last fiscal year to net income
6	ep_covar	Covariability in earnings	Coefficient estimate in the regression of monthly earnings-to-price ratios on the market's monthly earnings-to-price ratio (i.e., the value-weighted average of all stocks' monthly earnings-to-price ratios) over the last three years
7	ep_var	Variability in earnings	*Log* standard deviation of monthly earnings-to-price ratios over the last three years
8	finlev	Financial leverage	*Log* ratio of book value of total assets to market value of equity
9	fxdcos	Fixed cost of sales	*Log* ratio of selling, general, and administrative expenses plus research and development expenses plus advertising expenses to net sales
10	invest	Investment	Year-on-year growth of book value of total assets
11	noa	Net operating assets	Ratio of operating assets minus operating liabilities to book value of total assets
12	opaccr	Operating accruals	Ratio of changes in non-cash working capital minus depreciation to book value of total assets
13	oplev	Operating leverage	*Log* ratio of operating costs (i.e., the sum of costs of goods sold and selling, general, and administrative expense) to market value of total assets
14	ppe	PPE change-to-assets ratio	Ratio of changes in property, plants, and equipment to lagged book value of total assets
15	prof	Profitability	Ratio of gross profits to book value of equity
16	roa	Return on assets	Ratio of income before extraordinary items to book value of total assets
17	roe	Return on equity	Ratio of income before extraordinary items to book value of equity
18	ron	Return on net operating assets	Ratio of operating income after depreciation to lagged net operating assets
19	salestoassets	Sales-to-assets ratio	*Log* ratio of net sales to book value of total assets
20	salestoprice	Sales-to-price ratio	*Log* ratio of net sales to market value of equity
21	SGAtosales	SGA-to-sales ratio	*Log* ratio of selling, general, and administrative expenses to net sales
2) Technical indicators			
22	illiq	Illiquidity	Ratio of monthly absolute return to monthly dollar trading volume
23	intermom	Intermediate momentum	Excess return from month -12 to month -7
24	ivol	Idiosyncratic volatility	*Log* standard deviation of daily residuals from the Fama and French (1992) three-factor model within the current month
25	ltrev	Long-term reversal	Excess return from month -36 to month -13
26	me	Size	*Log* market value of equity
27	mom	Momentum	Excess return from month -12 to month -2
28	relprc	Relative price	Ratio of end-of-month price to its highest daily price during the last year
29	strev	Short-term reversal	Excess return from the current month
30	to	Turnover	*Log* monthly dollar trading volume
3) Macroeconomic indicators			
31	dfy	Default spread	Yield differential between Moody's Baa- and Aaa-rated corporate bonds
4) Predictors based on sample estimates of beta			
32	ols_3m_d	Short-term beta	Sample estimate of beta obtained from rolling regressions using a three-month window of daily returns
33	ols_1y_d	Medium-term beta	Sample estimate of beta obtained from rolling regressions using a one-year window of daily returns
34	ols_5y_m	Long-term beta	Sample estimate of beta obtained from rolling regressions using a five-year window of monthly returns
5) Industry classifiers			
35-81	ind	Industry classification	Fama and French (1997) industry classification, resulting in $48 - 1 = 47$ industry dummies

Table 2
Cross-sectional and time series properties of beta estimates

This table reports the properties of out-of-sample beta estimates obtained from the different forecast models introduced in Section 4. Panel A focuses on cross-sectional properties, presenting time series means of 1) the *value-weighted* cross-sectional average of estimated betas, 2) the cross-sectional standard deviation, and 3) the cross-sectional minimum, median, and maximum value. Following the procedure outlined in Pastor and Stambaugh (1999), it also reports the implied cross-sectional standard deviation of true betas, i.e., $\widehat{Std}(\beta^R) = \left[\overline{Var(\beta^F)} - \overline{\widehat{Var}}_{\beta_i^R}\right]^{1/2}$. Panel B focuses on time series properties, presenting *value-weighted* cross-sectional means of 1) the time series average of estimated betas, 2) the time series standard deviation, 3) the time series minimum, median, and maximum value, and 4) the first-order autocorrelation. Following Becker et al. (2021), firms with less than fifty beta estimates are omitted for the summary statistics in Panel B. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.

		Panel A: Cross-sectional properties						*Panel B: Time series properties*					
	Model	Mean	SD	Min	Median	Max	Impl. SD	Mean	SD	Min	Median	Max	Auto-corr.
Established estimators	ols_5y_m	0.98	0.45	-1.00	1.02	4.35	0.31	1.07	0.31	0.49	1.04	1.77	0.96
	ols_1y_d	0.99	0.39	-1.26	0.80	3.11	0.26	1.04	0.28	0.44	1.02	1.77	0.95
	ewma_s	0.99	0.40	-1.53	0.80	3.29	0.26	1.04	0.30	0.38	1.02	1.86	0.92
	ewma_l	0.99	0.39	-1.34	0.80	3.14	0.26	1.04	0.29	0.43	1.02	1.79	0.94
	bsw	0.98	0.35	-0.16	0.82	2.23	0.24	1.03	0.25	0.49	1.01	1.63	0.95
	vasicek	0.98	0.34	-0.12	0.85	2.23	0.24	1.03	0.25	0.49	1.01	1.65	0.95
	karolyi	0.99	0.35	-0.14	0.85	2.38	0.25	1.03	0.25	0.49	1.01	1.68	0.95
	hybrid	1.00	0.33	-0.14	0.90	2.35	0.24	1.04	0.23	0.53	1.03	1.60	0.95
	fama-french	0.99	0.33	0.22	0.78	1.93	0.23	1.04	0.25	0.50	1.02	1.73	0.90
	long-memo	1.00	0.33	-0.52	0.80	2.41	0.26	1.06	0.18	0.68	1.05	1.49	0.84
ML estimators	lm	1.01	0.28	-0.45	0.80	2.13	0.18	1.06	0.19	0.65	1.05	1.60	0.92
	elanet	1.02	0.24	-0.34	0.80	2.02	0.16	1.06	0.18	0.69	1.05	1.49	0.92
	rf	0.99	0.26	0.07	0.80	1.89	0.18	1.04	0.18	0.68	1.03	1.47	0.91
	gbrt	0.98	0.26	0.02	0.78	1.90	0.18	1.02	0.19	0.61	1.01	1.48	0.90
	nn_1	0.98	0.28	-0.12	0.78	2.13	0.20	1.02	0.19	0.65	1.01	1.51	0.91

Table 3
Forecast errors

This table examines the differences in forecast errors produced by the forecast models introduced in Section 4. Panel A reports the time series means for monthly value-weighted MSEs, i.e., $MSE_{t+k|t} = \sum_{i=1}^{N_t} w_{i,t}(\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F})^2$, with $k = 12$, where $w_{i,t}$ is stock i's market capitalization-based weight. Panel B reports the fraction of months during the out-of-sample period for which the column model is 1) in the Hansen, Lunde, and Nason (2011) model confidence set (MCS), and 2) significantly better than the row model in a pairwise comparison (according to the Diebold and Mariano (1995) test (DM test) statistics). The DM tests of equal predictive ability inspect differences in stock-level squared forecast errors (SEs), i.e., $SE_{i,t+k|t} = (\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F})^2$, with $k = 12$. The DM test statistic in month t for comparing the model under investigation j with a competing model i is $DM_{ij,t} = \frac{\bar{d}_{ij,t}}{\hat{\sigma}_{\bar{d}_{ij,t}}}$, where $d_{ij,t} = SE_{i,t+k|t} - SE_{j,t+k|t}$ is the difference in SEs, $\bar{d}_{ij,t} = \sum_{i=1}^{N_t} w_{i,t} d_{ij,t}$ is the value-weighted cross-sectional average of these differences, and $\hat{\sigma}_{\bar{d}_{ij,t}}$ is the Newey and West (1987) standard error of $d_{ij,t}$ (with four lags to account for possible heteroskedasticity and autocorrelation). Positive signs of $DM_{ij,t}$ indicate superior predictive performance of model j relative to model i in month t, i.e., that model j yields, on average, lower forecast errors than model i. All statistical tests are based on the 10% significance level. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.

		Established estimators										ML estimators				
		ols_5y_m	ols_1y_d	ewma_s	ewma_l	bsw	vasicek	karolyi	hybrid	fama-french	long-memo	lm	elanet	**rf**	gbrt	nn_1
Panel A: Average forecast errors																
	MSE, v.w. [%]	18.37	9.21	9.10	8.97	8.31	8.40	8.47	8.13	8.71	7.83	9.27	8.84	**7.49**	7.65	7.70
Panel B: Forecast errors over time																
	In MCS	4.57	40.96	49.90	48.65	56.13	56.34	55.51	59.46	53.85	71.73	53.85	60.71	**85.24**	76.92	79.63
Established estimators	vs. ols_5y_m		87.53	87.11	87.53	91.48	90.64	89.81	93.97	90.23	95.01	90.44	89.81	**93.76**	93.56	94.59
	vs. ols_1y_d	0.83		31.19	46.15	70.48	75.47	79.83	64.24	47.82	54.89	37.21	43.87	**61.12**	59.25	57.80
	vs. ewma_s	0.83	26.20		38.05	49.48	46.78	48.02	50.31	37.21	53.22	34.51	42.00	**55.93**	56.76	55.09
	vs. ewma_l	0.62	18.09	21.41		53.01	51.56	50.73	51.14	36.80	52.39	34.30	40.12	**56.13**	54.68	53.64
	vs. bsw	0.42	6.03	12.68	13.31		25.78	21.21	38.88	16.01	44.91	25.57	34.10	**51.35**	52.39	48.65
	vs. vasicek	0.83	7.90	12.89	12.06	23.70		15.38	42.00	12.06	46.15	24.53	33.47	**49.90**	54.05	49.06
	vs. karolyi	0.62	5.20	13.31	11.23	24.95	31.19		41.79	18.30	45.74	27.44	36.17	**50.94**	53.01	49.69
	vs. hybrid	0.00	6.65	15.38	17.26	24.12	27.03	28.07		16.01	37.21	18.09	29.73	**45.32**	45.95	44.91
	vs. fama-french	1.66	14.14	17.26	18.71	28.07	28.48	27.86	33.89		44.91	23.08	33.89	**53.01**	54.05	51.35
	vs. long-memo	0.00	17.26	19.13	19.33	22.04	22.04	23.08	23.28	20.79		12.68	20.37	**35.76**	35.14	37.63
ML estimators	vs. lm	5.20	34.30	33.47	35.14	40.33	39.50	39.50	39.71	37.01	48.02		37.84	**65.07**	61.54	66.53
	vs. elanet	4.99	31.19	31.39	32.64	37.21	35.97	37.01	34.10	31.81	39.92	9.98		**55.93**	51.98	55.09
	vs. **rf**	**1.25**	**8.11**	**7.90**	**8.32**	**13.93**	**12.89**	**13.31**	**10.81**	**6.86**	**20.79**	**3.33**	**8.11**		**23.49**	**28.69**
	vs. gbrt	1.04	11.02	12.27	12.68	16.01	16.22	16.84	15.59	9.98	22.04	4.57	8.73	**31.39**		25.78
	vs. nn_1	1.04	14.35	14.55	16.01	17.26	17.88	18.92	18.09	14.55	21.00	3.95	9.56	**29.73**	21.21	
	T	481	481	481	481	481	481	481	481	481	481	481	481	**481**	481	481

Table 4
Forecast error decomposition

This table presents the results of the Mincer and Zarnowitz (1969) MSE decomposition for the forecast models introduced in Section 4. It follows a two-step approach. First, the overall MSE in each out-of-sample forecast period is decomposed to disentangle the *bias*, *inefficiency*, and *random error* components, i.e.,

$$MSE_{t+k|t} = \underbrace{(\bar{\beta}^{R}_{t+k} - \bar{\beta}^{F}_{t+k|t})^2}_{\text{Bias}} + \underbrace{(1-b_t)^2\sigma^2(\beta^{F}_{t+k|t})}_{\text{Inefficiency}} + \underbrace{(1-\rho_t^2)\sigma^2(\beta^{R}_{t+k})}_{\text{Random error}},$$

where $\bar{\beta}$ and $\sigma^2(\beta)$ refer to the value-weighted mean and variance, respectively. b_i is the coefficient estimate in the WLS regression $\beta^{R}_{i,t+k} = a_i + b_i\beta^{F}_{i,t+k|t} + e_{i,t+k}$ (using the stocks' market capitalization-based weights), and ρ_t^2 is the coefficient of determination of this regression. Second, the aggregate numbers are computed as the time series averages of monthly MSE components. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.

	Model	Bias [%]	Inefficiency [%]	Random Error [%]	Overall [%]
Established estimators	ols_5y_m	0.41	8.04	9.93	18.37
	ols_1y_d	0.37	2.13	6.71	9.21
	ewma_s	0.31	2.22	6.58	9.10
	ewma_l	0.34	2.08	6.56	8.97
	bsw	0.38	1.24	6.69	8.31
	vasicek	0.36	1.30	6.74	8.40
	karolyi	0.37	1.47	6.64	8.47
	hybrid	0.42	1.17	6.55	8.13
	fama-french	0.19	1.49	7.03	8.71
	long-memo	0.32	0.99	6.52	7.83
ML estimators	lm	0.67	1.16	7.44	9.27
	elanet	0.70	1.01	7.13	8.84
	rf	**0.32**	**0.90**	**6.27**	**7.49**
	gbrt	0.48	0.81	6.36	7.65
	nn_1	0.44	0.96	6.30	7.70

Table 5
Market-neutral minimum variance portfolios

This table reports the properties of market-neutral minimum variance portfolios constructed based on out-of-sample beta estimates obtained from the forecast models introduced in Section 4. The portfolio optimization follows the procedure outlined in Ghysels and Jacquier (2006) and is described in Section 5.1.5. Panel A reports the ex-post portfolio betas (β) and the *t*-statistics (in parentheses) testing the null hypothesis that these betas are zero, based on Newey and West (1987) standard errors (with four lags to account for possible heteroskedasticity and autocorrelation). It further presents the ratio of systematic to total variance. Panel B adds the time series averages of minimum and maximum portfolio weights assigned through the optimization process, and the percentage of negative weights within the optimized portfolios. In line with Chan, Karceski, and Lakonishok (1999) and Cosemans et al. (2016), microcaps (stocks with market capitalizations equal to or below the 20th percentile of NYSE stocks) are excluded. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.

		Panel A: Risk metrics		*Panel B: Weights*		
	Model	β	$\sigma^2_{syst}/\sigma^2_{total}$ [%]	Min w_i	Max w_i	w_i<0
Established estimators	ols_5y_m	0.36	22.48	-1.04	3.54	40.36
		(6.18)				
	ols_1y_d	0.29	11.89	-1.36	3.49	40.39
		(3.14)				
	ewma_s	0.30	12.87	-1.35	3.39	39.68
		(3.22)				
	ewma_l	0.29	11.82	-1.36	3.46	40.13
		(3.10)				
	bsw	0.25	7.97	-1.57	3.95	42.50
		(2.51)				
	vasicek	0.23	6.77	-1.66	4.15	43.00
		(2.33)				
	karolyi	0.23	6.48	-1.70	4.14	41.91
		(2.30)				
	hybrid	0.18	4.26	-1.69	4.69	45.30
		(1.90)				
	fama-french	0.19	4.12	-1.89	4.10	43.26
		(1.83)				
	long-memo	0.20	4.64	-1.50	4.09	43.84
		(2.09)				
ML estimators	lm	0.12	1.62	-1.87	5.31	45.47
		(1.21)				
	elanet	0.02	0.03	-2.28	5.76	46.14
		(0.16)				
	rf	0.00	0.00	-1.92	5.27	47.20
		(0.02)				
	gbrt	0.07	0.44	-1.76	4.87	46.09
		(0.60)				
	nn_1	0.10	1.00	-1.70	4.92	45.17
		(0.92)				

Table 6
Market-neutral anomaly portfolios

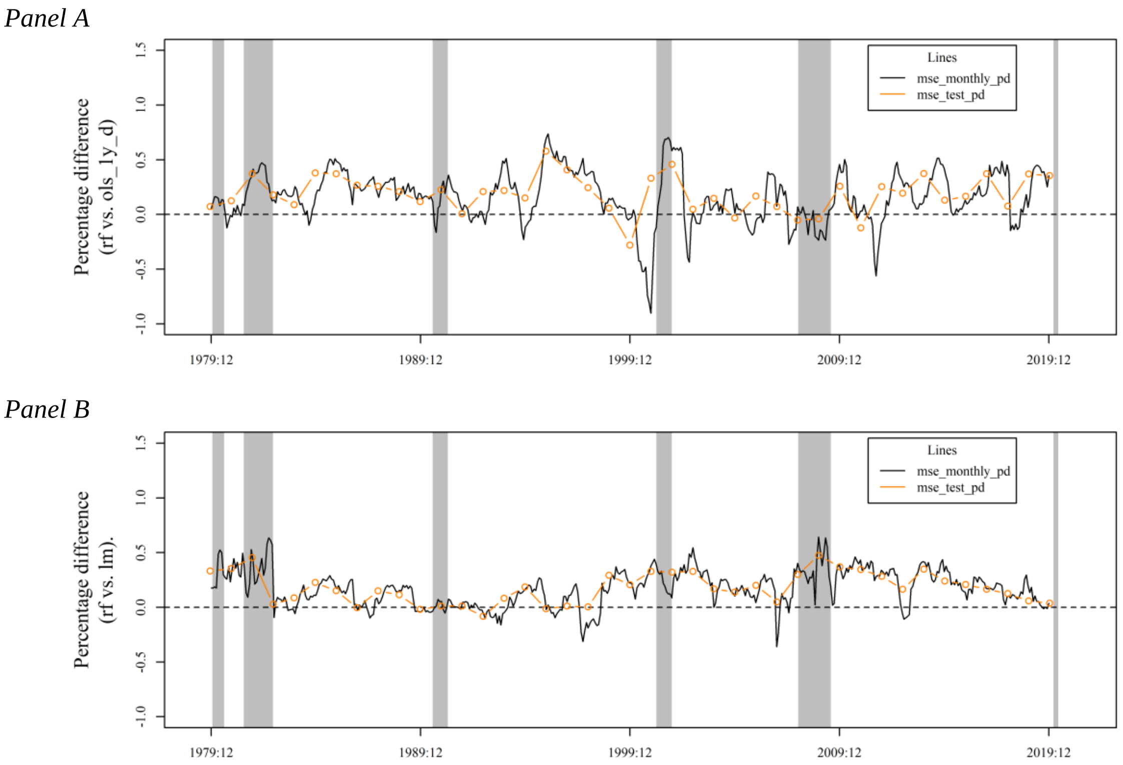
This table reports the properties of market-neutral anomaly portfolios, which are constructed based on out-of-sample beta estimates obtained from the forecast models introduced in Section 4. The anomaly variables are size (me), book-to-market ratio (bm), momentum (mom), illiquidity ($illiq$), and each forecast model's predicted beta (β^F). The portfolio optimization is described in Section 5.1.6. Briefly, the stocks are sorted into quintile portfolios and then the weights are optimized within the high and low portfolios, respectively. The objective is to minimize the deviation of the weights from those implied by the stocks' relative market capitalizations in that portfolio subject to the weights being non-negative and the expected portfolio beta being equal to one. In particular, this table reports the time series averages of ex-post portfolio betas (β) for the long–short anomaly portfolios (HML) and the t-statistics (in parentheses) testing the null hypothesis that these betas are zero, based on Newey and West (1987) standard errors (with eleven lags to account for possible heteroskedasticity and autocorrelation). It also presents the numbers for the long (H) and short (L) anomaly portfolios separately. In this case, the null hypotheses are that these betas are one. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.

		me			bm			mom			illiq			β^F		
	Model	β_{HML}	β_H	β_L	β_{HML}	β_H	β_L	β_{HML}	β_H	β_L	β_{HML}	β_H	β_L	β_{HML}	β_H	β_L
Established estimators	ols_5y_m	0.58	1.02	0.44	-0.18	0.86	1.03	0.17	1.06	0.90	-0.52	0.51	1.03	-0.73	0.77	1.50
		(9.26)	(0.33)	(-12.44)	(-5.15)	(-4.44)	(1.15)	(4.30)	(2.73)	(-4.75)	(-6.83)	(-9.32)	(0.64)	(-2.08)	(-3.64)	(2.00)
	ols_1y_d	0.26	0.99	0.73	-0.02	0.97	1.00	0.09	1.04	0.95	-0.21	0.78	0.99	-0.92	0.89	1.81
		(2.41)	(-0.20)	(-2.85)	(-0.58)	(-0.81)	(-0.10)	(2.81)	(2.75)	(-2.13)	(-2.12)	(-2.62)	(-0.26)	(-5.89)	(-3.07)	(5.31)
	ewma_s	0.28	0.99	0.71	-0.03	0.97	0.99	0.07	1.03	0.96	-0.23	0.76	0.99	-1.08	0.89	1.96
		(3.36)	(-0.27)	(-3.82)	(-0.81)	(-1.15)	(-0.20)	(2.62)	(2.40)	(-2.03)	(-2.84)	(-3.32)	(-0.34)	(-5.58)	(-4.45)	(5.05)
	ewma_l	0.26	0.99	0.73	-0.02	0.97	1.00	0.08	1.04	0.96	-0.21	0.78	0.99	-0.94	0.89	1.83
		(2.79)	(-0.23)	(-3.24)	(-0.66)	(-0.95)	(-0.15)	(2.70)	(2.53)	(-2.09)	(-2.40)	(-2.90)	(-0.30)	(-6.18)	(-3.37)	(5.53)
	bsw	0.26	1.00	0.74	-0.03	0.98	1.01	0.08	1.06	0.98	-0.21	0.79	1.00	-0.33	0.93	1.25
		(2.39)	(0.05)	(-2.86)	(-0.66)	(-0.48)	(0.31)	(2.40)	(3.37)	(-1.02)	(-2.06)	(-2.49)	(0.04)	(-3.93)	(-2.31)	(3.32)
	vasicek	0.33	1.00	0.67	-0.04	0.97	1.01	0.10	1.06	0.96	-0.28	0.72	1.00	-0.24	0.93	1.16
		(3.18)	(-0.03)	(-3.54)	(-0.93)	(-0.85)	(0.25)	(2.72)	(3.48)	(-1.55)	(-2.74)	(-3.17)	(-0.05)	(-1.82)	(-2.25)	(1.27)
	karolyi	0.37	1.00	0.62	-0.04	0.97	1.01	0.10	1.05	0.96	-0.31	0.69	1.00	-0.19	0.92	1.11
		(4.29)	(-0.06)	(-4.84)	(-0.95)	(-0.91)	(0.20)	(2.73)	(3.29)	(-1.70)	(-3.38)	(-3.94)	(-0.09)	(-1.77)	(-2.48)	(1.03)
	hybrid	0.39	0.99	0.61	-0.06	0.94	1.00	0.10	1.04	0.94	-0.32	0.67	0.99	-0.11	0.92	1.03
		(4.25)	(-0.12)	(-5.27)	(-1.70)	(-1.80)	(0.13)	(3.18)	(2.97)	(-2.46)	(-3.58)	(-4.40)	(-0.14)	(-0.78)	(-2.09)	(0.26)
	fama-french	0.11	0.99	0.88	-0.01	1.00	1.00	0.09	1.06	0.97	-0.08	0.91	0.99	-0.25	0.94	1.19
		(1.31)	(-0.83)	(-1.63)	(-0.20)	(-0.10)	(0.23)	(2.27)	(4.10)	(-0.98)	(-1.14)	(-1.52)	(-0.89)	(-3.79)	(-2.39)	(3.38)
	long-memo	0.18	0.98	0.80	-0.02	0.97	0.99	0.11	1.06	0.94	-0.14	0.84	0.98	-0.30	0.93	1.22
		(4.04)	(-0.57)	(-5.36)	(-0.58)	(-1.01)	(-0.28)	(3.04)	(3.58)	(-2.21)	(-3.26)	(-4.29)	(-0.80)	(-5.15)	(-3.70)	(4.62)
ML estimators	lm	0.10	0.97	0.87	-0.03	0.95	0.98	0.05	1.02	0.97	-0.04	0.93	0.97	-0.34	0.94	1.28
		(1.43)	(-1.52)	(-2.25)	(-1.08)	(-2.15)	(-0.83)	(1.20)	(0.91)	(-1.21)	(-0.61)	(-1.30)	(-1.63)	(-1.72)	(-3.18)	(1.36)
	elanet	0.11	0.96	0.84	0.02	0.98	0.97	0.04	1.02	0.98	-0.05	0.91	0.96	-0.20	0.96	1.16
		(1.57)	(-1.40)	(-2.78)	(0.39)	(-0.53)	(-1.08)	(0.75)	(0.76)	(-0.66)	(-0.71)	(-1.70)	(-1.55)	(-1.04)	(-1.93)	(0.82)
	rf	**0.01**	**0.99**	**0.98**	**0.00**	**1.00**	**1.01**	**0.07**	**1.05**	**0.98**	**0.02**	**1.01**	**0.99**	**0.01**	**1.00**	**0.99**
		(0.11)	**(-0.29)**	**(-0.22)**	**(-0.10)**	**(0.05)**	**(0.23)**	**(1.63)**	**(2.58)**	**(-0.63)**	**(0.24)**	**(0.18)**	**(-0.31)**	**(0.10)**	**(0.04)**	**(-0.11)**
	gbrt	0.02	1.01	0.98	0.01	1.02	1.02	0.05	1.05	1.00	0.00	1.00	1.01	-0.11	1.00	1.11
		(0.25)	(0.31)	(-0.22)	(0.13)	(0.46)	(0.76)	(1.20)	(2.39)	(-0.09)	(-0.06)	(0.02)	(0.29)	(-0.98)	(0.05)	(0.99)
	nn_1	0.01	1.00	0.99	0.00	1.01	1.01	0.03	1.03	1.00	0.01	1.01	1.00	-0.12	0.98	1.10
		(0.09)	(0.08)	(-0.08)	(-0.08)	(0.28)	(0.57)	(0.74)	(1.75)	(-0.14)	(0.16)	(0.19)	(0.05)	(-0.89)	(-0.64)	(0.90)

Figures

Figure 1
Relative forecast errors over time

This figure depicts the forecast errors of random forests (rf), introduced in Section 4.2, at each re-estimation date relative to those achieved by one-year rolling betas (ols_1y_d, Panel A) and betas obtained from simple linear regressions (lm, Panel B), introduced in Sections 4.1 and 4.2, respectively. The relative forecast errors are computed as the percentage difference in monthly MSEs (mse_monthly_pd) and time series means for monthly MSEs within each test sample (mse_test_pd). These percentage differences are calculated as one minus the MSE of the random forests divided by the MSE of the respective benchmark model. The points are assigned to the re-estimation dates, i.e., the dates at which forecasts of stock-level betas (over the next year) are obtained. It also visualizes the National Bureau of Economic Research (NBER) recession periods (shaded grey). The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.


Panel A

Panel B

Figure 2
Average forecast errors for decile portfolios based on beta forecasts

This figure plots the time series averages of monthly mean squared forecast errors (grey bars) for decile portfolios based on beta forecasts. To this end, stocks are sorted into decile portfolios based on their predicted betas at the end of each month t, separately for each of the selected forecast models introduced in Section 4. Monthly forecast errors in this empirical test are defined as the value-weighted MSE between beta forecasts and realized betas over the next year within each portfolio. To these visualizations, the average under- and overestimation fractions (black unfilled squares) are added. They are computed as the fraction of stocks within each decile portfolio for which the difference between beta forecasts and realized betas is positive. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.

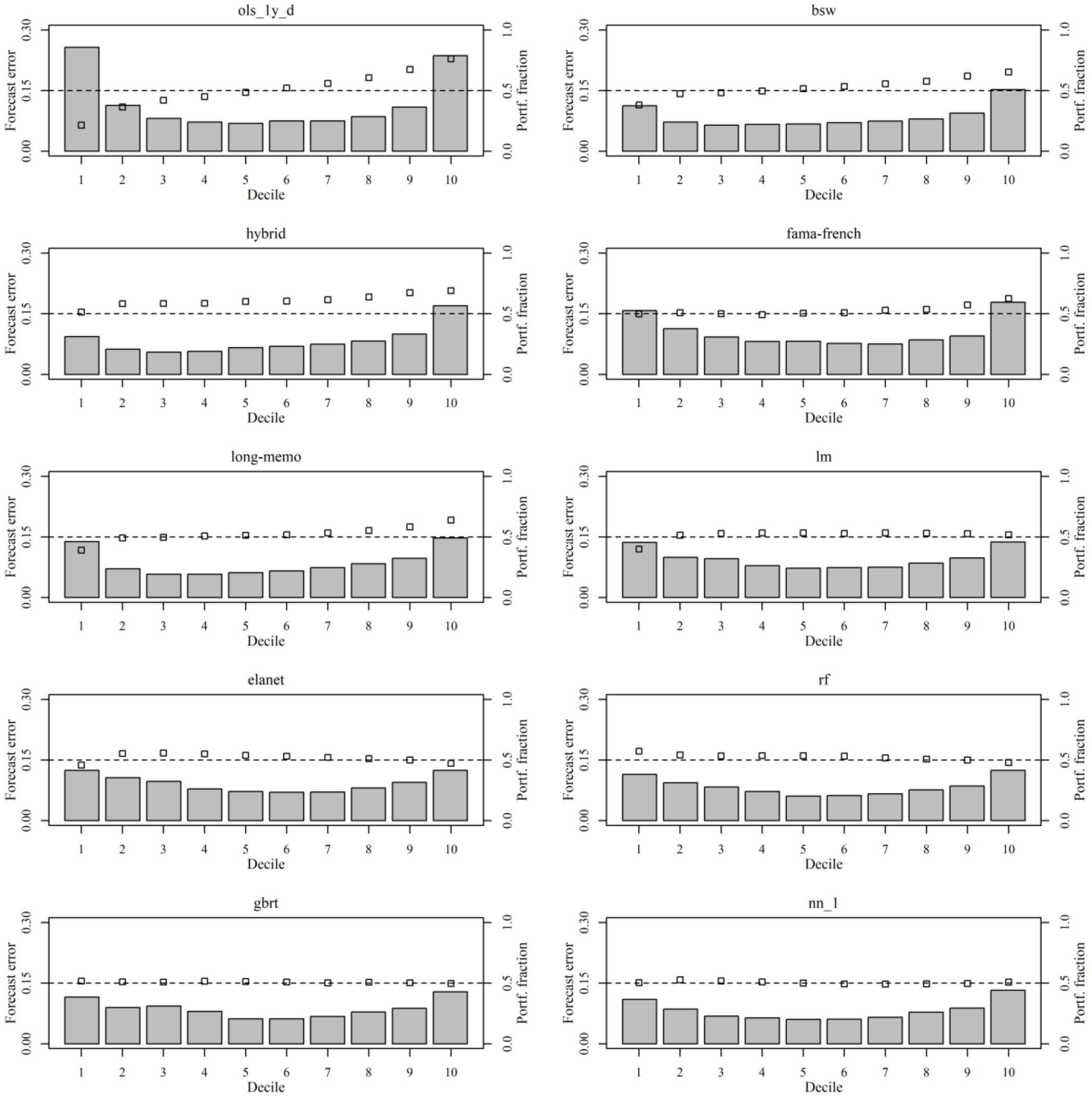


Figure 3
Average forecast errors for decile portfolios based on firm characteristics and industry classification

This figure plots the time series averages of monthly mean squared forecast errors (grey bars) for decile portfolios based on firm characteristics and industry classifications. To this end, the procedure outlined for Figure 2 is replicated, but stocks are sorted into decile portfolios based on a firm's size, valuation, momentum, illiquidity, and the industry classification of Fama and French (1997), i.e., Consumer Nondurables (CND), Consumer Durables (CD), Manufacturing (M), Oil, Gas, and Coal Extraction and Products (OGC), Business Equipment (B), Telephone and Television Transmission (TTT), Wholesale, Retail, and Some Services (WRS), Healthcare, Medical Equipment, and Drugs (HMD), Utilities (U), and Other (O). In particular, this figure plots the time series averages of monthly forecast errors within each decile portfolio for the rf model (grey bars), introduced in Section 4.2, and the respective benchmark model (red bars), i.e., the ols_1y_d (left-hand column) and lm (right-hand column) models. Monthly forecast errors in this empirical test are defined as the value-weighted MSE between beta forecasts and realized betas over the next year within each portfolio. To these visualizations, the percentage differences in average forecast errors relative to the respective benchmark model (black unfilled triangles; calculated as one minus the MSE of the random forests divided by the MSE of the respective benchmark) are added. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.

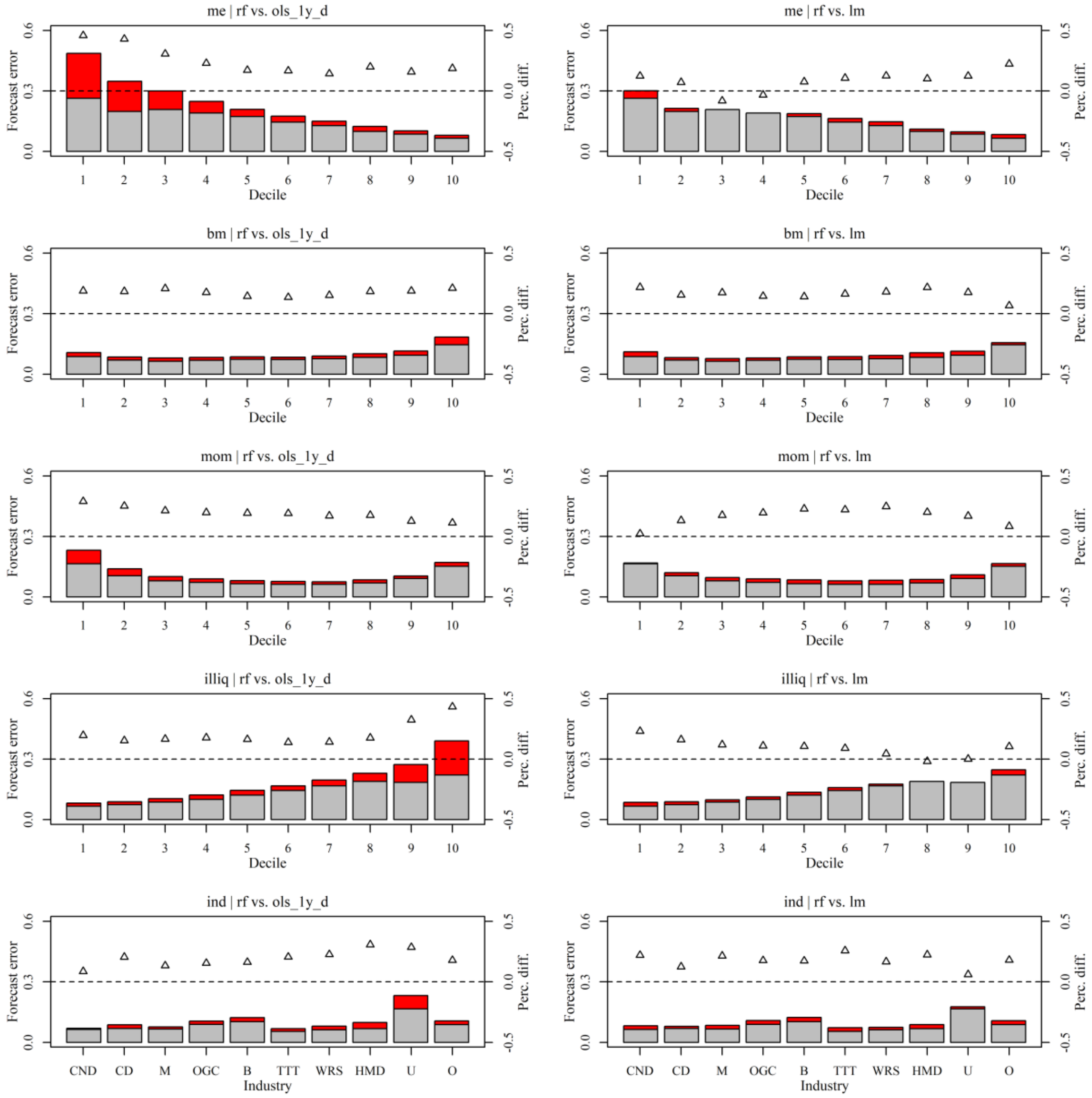
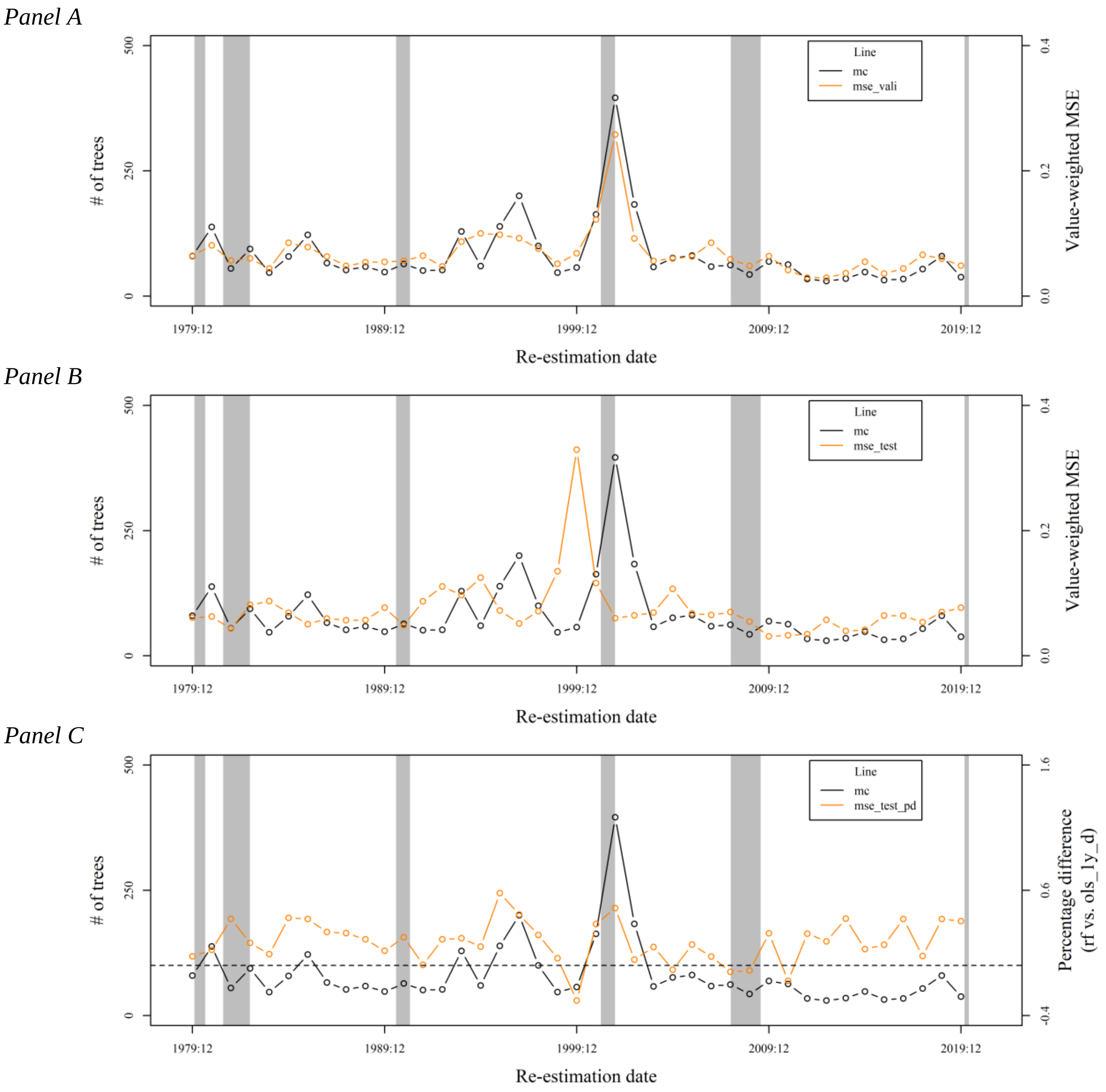


Figure 4
Model complexity over time

This figure illustrates the model complexity of random forests (rf), introduced in Section 4.2, at each re-estimation date and its association with forecast errors. The rf model is non-parametric and tree-based, so the number of trees added to the ensemble prediction (mc) is taken to measure model complexity. The forecast errors are computed as time series means for monthly MSEs within each validation sample (mse_vali) and each test sample (mse_test), respectively. The relative forecast errors (between random forests and one-year rolling betas (ols_1y_d), introduced in Section 4.1) are the percentage difference in the test-sample MSEs (mse_test_pd). These differences are calculated as one minus the MSE of the random forests divided by the MSE of the respective benchmark model. In particular, this figure plots mc over time, together with mse_vali (Panel A), mse_test (Panel B), and mse_test_pd (Panel C), respectively. The points are assigned to the re-estimation dates, i.e., the dates at which forecasts of stock-level betas (over the next year) are obtained. It also visualizes the National Bureau of Economic Research (NBER) recession periods (shaded grey). The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.


Panel A

Panel B

Panel C

Figure 5
Relative variable importance in aggregate and over time

This figure depicts the relative importance of variables incorporated as predictors in random forests (rf), which are introduced in Section 4.5, both in aggregate and over time. To this end, the variable importance matrix is calculated in a two-step approach, separately for each re-estimation date. First, the absolute variable importance is computed as the increase in value-weighted MSE from setting all values of a given predictor to its uninformative median value within the training sample. Second, the absolute variable importance measures are normalized to sum to 1, signaling the relative contribution of each variable to the rf model. Panels A and B depict the time series average of relative variable importance measures for the predictor categories introduced in Section 3 and for the ten most influential predictors. Panel C presents these relative variable importance metrics over the sample period, focusing on the twenty-nine least important predictors. To this end, the remaining variables are omitted prior to normalizing the absolute variable importance measures to sum to 1 at each re-estimation date. Panel D visualizes the fraction of aggregate absolute variable importance (sum of increases in value-weighted MSE across all variables) that is attributed to the twenty-nine least important predictors over the sample period, together with the National Bureau of Economic Research (NBER) recession periods (shaded grey). The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 period, while the first beta estimates are obtained in December 1979.

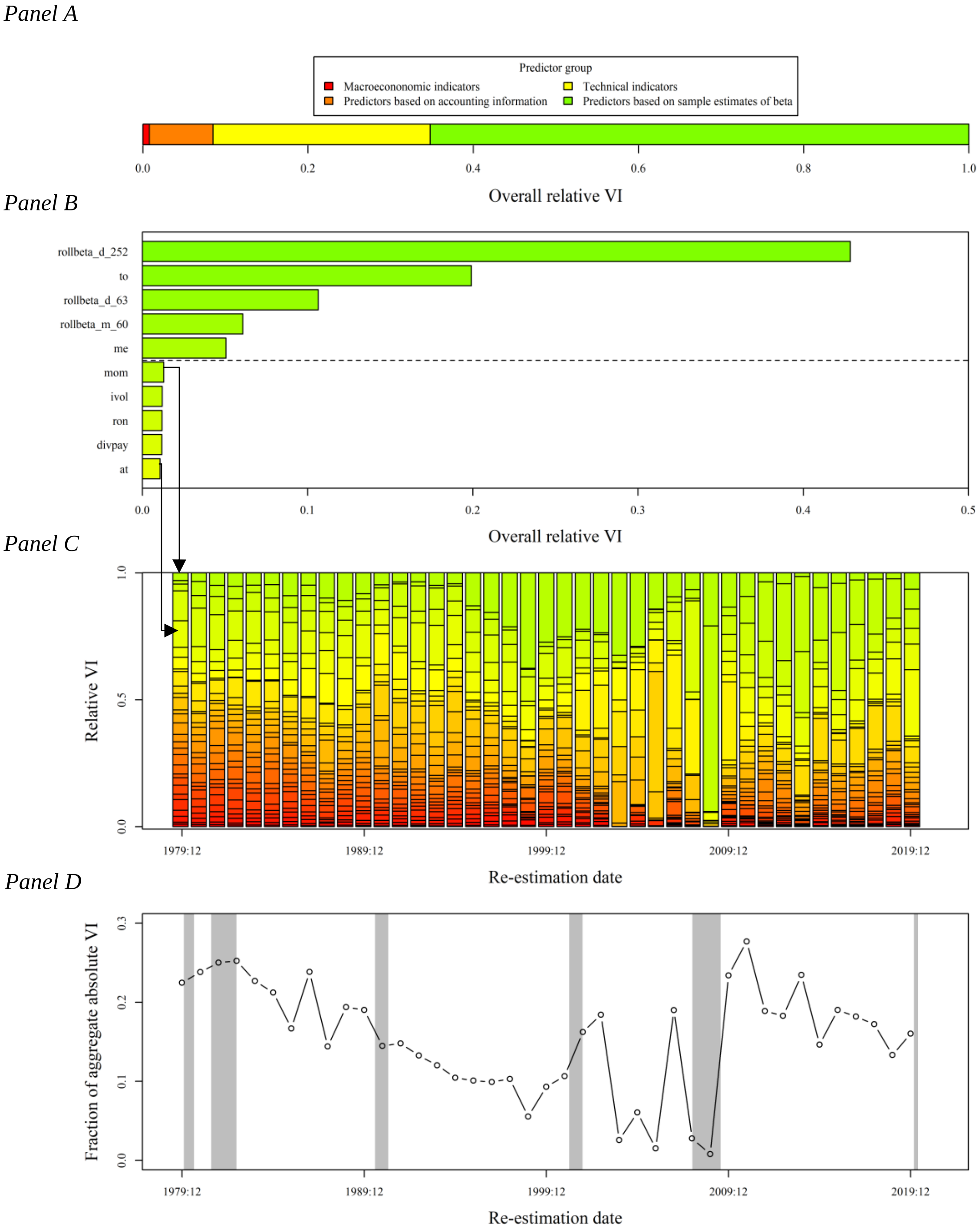
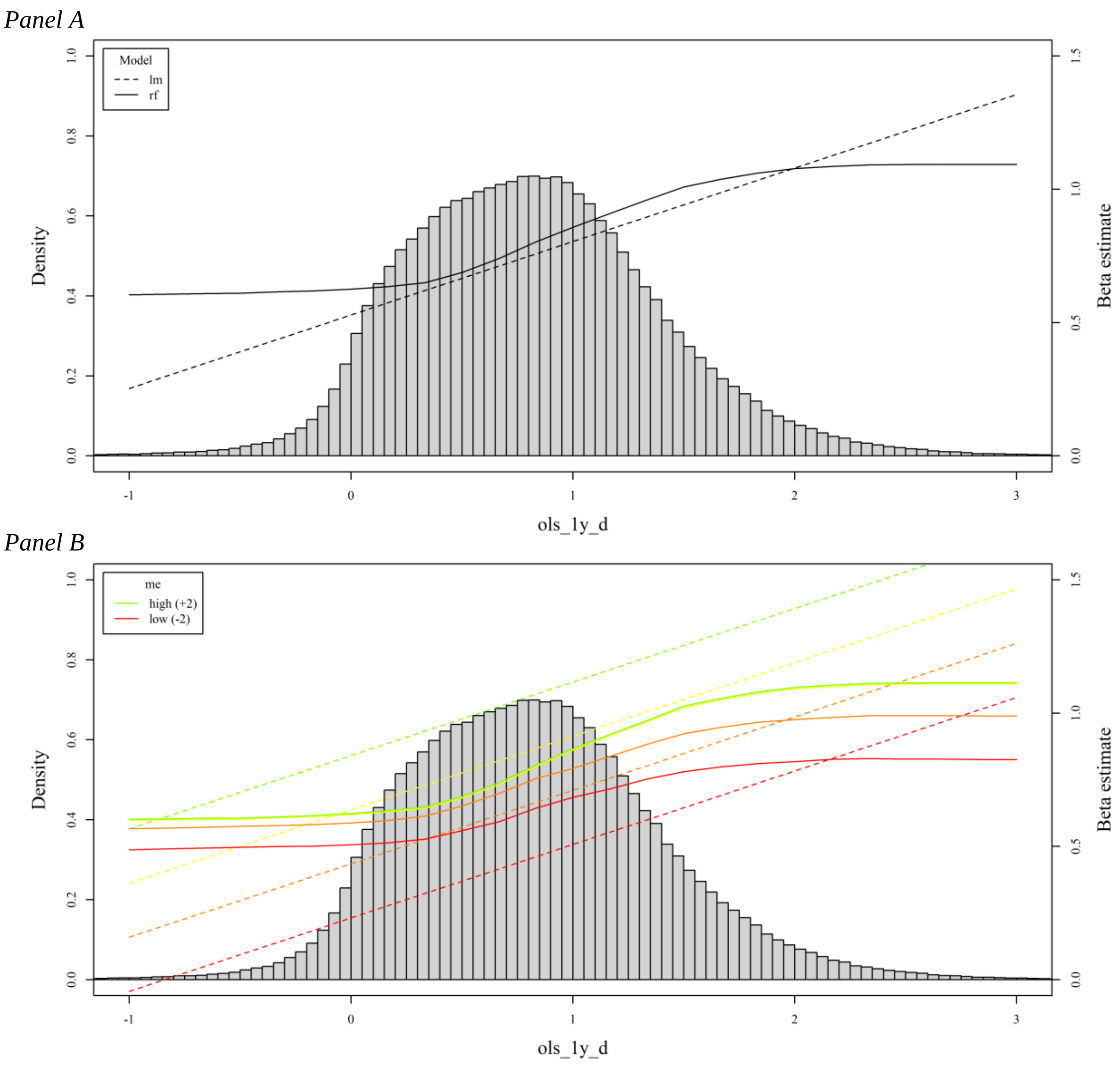


Figure 6
Nonlinear and interactive effects in estimating future market betas

This figure examines the ability to capture nonlinear and interactive effects in estimating future market betas of random forests (rf) and simple linear regressions (lm). Both forecast models are introduced in Section 4.5. Panel A illustrates the marginal association between a firm's sample estimate of beta from rolling regressions using a one-year window of daily returns ($\mathrm{ols}_{1y,d}$) and its beta estimates ($\beta^{F}_{it+k|t}$, with $k = 12$). To visualize the average effect of $\mathrm{ols}_{1y,d}$ on $\beta^{F}_{it+k|t}$, all predictors are set to their uninformative median values (within the training sample) at each re-estimation date, and the industry dummies to the value of zero. $\mathrm{ols}_{1y,d}$ is then varied across the $(-1, +3)$ interval and estimated betas are computed for this artificial test sample. Finally, beta estimates are averaged across re-estimation dates. A histogram that depicts the historical distribution of $\mathrm{ols}_{1y,d}$ is added to the visualization. Panel B shows the interactive effect of $\mathrm{ols}_{1y,d}$ and a firm's size according to its market value of equity (me) on $\beta^{F}_{it+k|t}$. To this end, the procedure outlined above is replicated. However, in this case, estimated betas are computed for different levels of me (across the $(-2, +2)$ interval), while low and high levels for me are marked with red and green lines, respectively. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.



Panel A

Panel B

Internet Appendix

Internet Appendix A: Benchmark estimators

In this appendix, we briefly introduce the representative set of established forecast models listed in Section 4.1. We utilize them as benchmarks to identify whether the machine-learning based estimators introduced in Section 4.2 lead to incremental predictive power. We begin with simple rolling-window estimators, continue with shrinkage-based approaches, and end up with a portfolio-based and a long-memory model. We also note the major differences among the model families. In Table A1, we summarize (in Panel A) the definitions and descriptions of the different models to forecast time-varying market betas.

Rolling-window estimators

In the baseline rolling-window approach, we obtain *historical betas* by running time series OLS regressions, i.e.,

$$r_{i,ts} = \alpha_{i,t}^{H} + \beta_{i,t}^{H} r_{M,ts} + \varepsilon_{i,ts},$$

where $r_{i,ts}$ and $r_{M,ts}$ are excess returns on stock i and the market portfolio M, respectively. The subscript t indicates that we estimate historical alphas and betas ($\alpha_{i,t}^{H}$ and $\beta_{i,t}^{H}$, respectively) for each month t using a rolling window of daily or monthly excess returns. The subscript $s = (1,2,\ldots,\tau)$ indicates returns before the end of month t, while τ refers to the length of the rolling window. The intercept $\alpha_{i,t}^{R}$ is the risk-adjusted return, while the slope $\beta_{i,t}^{R}$ is our parameter of interest. The error term $\varepsilon_{\mathrm{it,s}}$ is an idiosyncratic return shock.

Rolling-window estimates of beta are based only on historical return information. Because there is no need to specify a set of conditioning variables (neither fundamental nor macroeconomic), these estimators are less prone to misspecification. However, they implicitly assume that betas are constant within the rolling window (and going forward), which leads to an important bias–variance trade-off, i.e., picking up short-term fluctuations in betas (conditionality) versus precisely capturing long-term averages. Shorter rolling windows increase the ability to use short-term information, which reduces estimation bias. The resulting smaller rolling samples, however, are more strongly affected by microstructure noise, which increases both estimation variance and measurement errors. Because of this trade-off, we consider two sets of rolling betas estimated from different window lengths and data frequencies. The first utilizes a five-year window of monthly returns (ols_5y_m), as proposed by Black, Jensen, and Scholes (1972) and Fama and MacBeth (1973). The second is obtained from rolling regressions using a one-year window of daily returns (ols_1y_d), as in Andersen, Bollerslev, and Meddahi (2006). Modifications of the baseline rolling-window approach that help improve the bias–variance trade-off are numerous. In our empirical analysis, we focus on the two most commonly used approaches.

First, within a standard OLS regression setting, observations that enter the market model are equally weighted. However, from a conceptual perspective, altering the underlying weighting scheme allows to place higher weights on more recent observations. Thus, the estimates are "conditional", while still incorporating sufficiently large rolling samples. In line with Hollstein, Prokopczuk, and Wese Simen (2019), we utilize rolling-window estimators with an exponentially-weighted moving average structure. To obtain *exponentially-weighted betas*, we run WLS rolling regressions using a one-year window of daily returns with exponential weights. These are defined by the time required for the weights to fall below half of their initial value, i.e., their half-life. A longer time span hereby reflects slower exponential decay. As an example, Figure A1 provides a visualization of exponentially decaying weights based on long and short half-lives. To be conservative, we consider two sets of rolling betas estimated from different horizons: 1) one-third ($ewma\_s$) and 2) two-thirds ($ewma\_l$) of the number of observations of the (initial) rolling window.

Second, the standard OLS regression setting is particularly prone to outliers in a stock's return history, resulting in extreme and volatile beta estimates. To moderate the influence of single outlier returns, Welch (2019) assumes that market betas to lie within the $(-2, +4)$ interval. In theory, this assumption increases the signal-to-noise ratio, which translates into improved predictive power. To this end, stock-level returns must first be winsorized at market return-based bounds, i.e., $r_{i,ts} \in \left(-2r_{Mt,s}, +4r_{Mt,s}\right)$. A one-year window of these daily winsorized returns is then used within the basic OLS rolling regression approach to obtain *slope-winsorized betas* (bsw).

Shrinkage-based estimators

Another approach that aims for refined beta forecasts is to enhance rolling-beta estimates with supplemental cross-sectional information. The idea is that a stock's beta forecast should not be too dissimilar from those of other stocks with similar characteristics. Therefore, a prior regarding the true beta can be specified, towards which the sample estimate of beta obtained from rolling regressions is shrunk. The established approach for obtaining *shrinkage betas* is to compute the weighted average of the prior belief $\overline{\beta}_{i,t}$ and the sample estimate of beta $\beta_{i,t}^{H}$, i.e.,

$$\tilde{\beta}_{i,t} = \varphi_{i,t}\bar{\beta}_{i,t} + (1 - \varphi_{i,t})\beta_{i,t}^{H}.$$

The shrinkage weight $\varphi_{i,t}$ is given by $\varphi_{i,t} = \frac{s^2_{\beta^H_{i,t}}}{\sigma^2_{\bar{\beta}_{i,t}} + s^2_{\beta^H_{i,t}}}$, where $\sigma^2_{\bar{\beta}_{i,t}}$ is the variance of the prior, and $s^2_{\beta^H_{i,t}}$ is the sampling variance of the rolling-beta estimates. The degree of shrinkage is proportional to the relative precision of the rolling-beta estimate and the prior. The lower the relative precision of the rolling-beta estimate (i.e., the larger $s^2_{\beta^H_{i,t}}$ is relative to $\sigma^2_{\bar{\beta}_{i,t}}$), the more weight is given to the prior. Conceptually, shrinking towards a well-defined prior reduces estimation noise, helping improve the accuracy of rolling-beta estimates. Prior beliefs hereby can be specified in various ways.

Vasicek (1973) suggests that, if no other information about a stock is known except that it comes from a broad universe, the optimal prior density for the true underlying beta is based on the cross-sectional distribution of beta. Thus, the value-weighted mean and variance of rolling betas within the cross section is used as prior information. Karolyi (1992) proposes grouping stocks into portfolios based on firm fundamentals and shrinking a firm's rolling-beta estimate is shrunk towards its portfolio beta. In particular, the value-weighted mean and variance within each industry portfolio are used as prior information.

Cosemans et al. (2016), however, argue that shrinkage based on Vasicek (1973) and Karolyi (1992) dampens only part of the noise in rolling-beta estimates. This is because the prior does not use the cross-sectional information embedded in firm fundamentals at all (Vasicek, 1973) or may be hampered by large intra-portfolio dispersion in betas (Karolyi, 1992). They suggest specifying priors unique to each firm, while incorporating a comprehensive set of firm fundamentals as predictors. In particular, they outline a complex Bayesian framework (which they refer to as "hybrid model") to compute firm-specific priors.[51]

Cosemans et al. (2016) also argue that the Bayesian approach yields a better bias–variance trade-off than standard frequentist methods. First, it pools the loadings on the conditioning variables (which reduces variance), while letting other coefficients vary across firms to capture variation in betas unrelated to the firm fundamentals included in the model (which reduces bias). Furthermore, they note that the panel structure is superior to a traditional OLS-based framework. At the one extreme, running separate OLS regressions for each firm also allows for firm-level parameter heterogeneity, but it leads to large measurement errors because it does not exploit cross-sectional information. At the other extreme, estimating the prior using a pooled OLS regression leads to precise beta estimates (as it exploits cross-sectional information). The estimates, however, are biased as this approach does not allow for unobserved cross-sectional heterogeneity in betas. Applying Cosemans et al.'s (2016) hybrid model would allow for both.

In our empirical analysis, we implement the shrinkage approach as follows. We obtain the rolling-beta estimates that contribute to each of the three shrinkage beta from rolling regressions using a one-year window of daily returns.[52] Prior information for the Vasicek (1973) and Karolyi (1992) beta estimates are obtained by considering the entire cross section ($vasicek$) and, analogously to Cosemans et al. (2016), by creating forty-seven industry portfolios ($karolyi$) according to the classification of Fama and French (1997). In line with Cosemans et al. (2016), prior information for the hybrid beta estimates are based on the conditioning variables *size*, *book-to-market ratio*, *financial leverage*, *operating leverage*, *momentum*, and *default spread* ($hybrid$).[53]

[51] The parameters of the hybrid model are estimated via Markov chain Monte Carlo methods (Cosemans et al., 2016).

[52] While Cosemans et al. (2016) obtain the sample betas that contribute to the hybrid estimates of beta from a rolling regression using a *half-year* window of daily returns, we opt for a *one-year* window. First, although not the main objective of our empirical analysis, using the same rolling-window length for each (shrinkage) beta is the only way to compare their predictive performance consistently. It allows to assess whether differences in predictive performance truly stem from differences in prior information, rather than from differences in rolling-window beta estimates. Second, and more importantly, in our empirical setting, we find that the predictive performance of the hybrid beta estimates is better for the one-year window (compared to the half-year window). This makes it an even more conservative benchmark for identifying the value of the machine learning techniques.

[53] Our implementation differs slightly from the original Cosemans et al. (2016) shrinkage approach in two ways: First, we opt for the Novy-Marx (2011) definition of operating leverage (see Footnote 5), and for a one-year window of daily returns to compute sample estimates of beta obtained from rolling regressions (see Footnote 48).

Portfolio-based estimators

In estimating time-varying market betas, Fama and French (1992) take a different approach. They first sort individual stocks into portfolios, then estimate rolling betas for each portfolio, and, finally, assign portfolio betas to individual stocks. In our empirical analysis, we implement their approach as follows. On a monthly basis, we sort stocks into size deciles based on NYSE breakpoints. Each size decile is subdivided into ten portfolios based on sample estimates of beta obtained from rolling regressions using a one-year window of daily returns. Equal-weighted daily returns are computed for each of the resulting 100 size–beta portfolios over the next month. Finally, we obtain the *portfolio betas* from rolling regressions using a one-year window of daily post-ranking portfolio returns. These beta estimates are assigned to the individual stocks in each of the 100 size–beta portfolios ($fama\text{-}french$).[54]

Long-memory estimators

Instead of shrinking rolling-beta estimates to prior beliefs or assigning rolling portfolio betas to individual stocks, Becker et al. (2021) focus on the time series properties of realized betas to obtain beta forecasts. They state that the degree of memory within a beta time series, i.e., the order of integration d (typically with $0 \leq d \leq 1$), is the key determinant to modelling beta dynamics. A larger d hereby indicates a longer memory, and vice versa. Much of the literature concentrates on extreme cases (Blume, 1975; Ang and Chen, 2007; Hollstein and Prokopczuk, 2016; Levi and Welch, 2017), i.e., $d = 0$ ("no" memory) or $d = 1$ ("infinite" memory). But Becker et al. (2021) find that beta time series clearly show long-memory properties ($0 < d < 1$). In this case, the current value of a variable depends on past shocks, but the less so the further in the past these shocks are. In other words, past shocks neither die out quickly nor persist infinitely, but exhibit a hyperbolically decaying impact. In our empirical analysis, we adapt their long-

[54] Fama and French (1992) estimate the pre- and post-ranking betas using monthly returns, and construct the size–beta portfolios on an annual basis. However, Cosemans et al. (2016) and Hollstein, Prokopczuk, and Wese Simen (2019) find that rolling-window beta estimates computed from daily returns are more accurate predictors of future betas than rolling-window beta estimates computed from monthly returns. This is another reason why we estimate these betas using a one-year window of daily returns. Furthermore, to always incorporate the most recent data, we construct the size–beta portfolios on a monthly basis.

memory approach ($long$-$memo$) by implementing a $FI(0.4)$ model, i.e., a fractionally integrated time series process with $= 0.4$.

Table A1
Details on forecast models

This table summarizes (in Panel A) the definitions and descriptions of each established beta estimator introduced in Section 4.1, and provides (in Panel B) the definitions and specifications of hyperparameters of each machine learning-based beta estimator introduced in Section 4.2.

Panel A: Established estimators

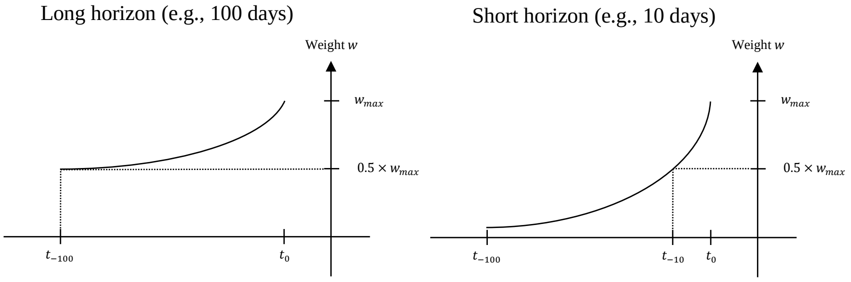
	Description	Definition
ols_5y_m	Historical beta	Rolling regressions using a five-year window of monthly returns
ols_1y_d	Historical beta	Rolling regressions using a one-year window of daily returns
ewma_s	Exponentially-weighted beta	Rolling regressions using a one-year window of daily returns with exponentially decaying weights (short half-life)
ewma_l	Exponentially-weighted beta	Rolling regressions using a one-year window of daily returns with exponentially decaying weights (long half-life)
bsw	Slope-winsorized beta	Rolling regressions using a one-year window of daily *winsorized* returns
vasicek	Shrinkage beta	Shrinkage of *ols_1y_d* towards average beta within stock universe
karolyi	Shrinkage beta	Shrinkage of *ols_1y_d* towards average beta within industry portfolio
hybrid	Shrinkage beta	Shrinkage of *ols_1y_d* towards firm-specific beta prior
fama-french	Portfolio beta	Assignment of portfolio betas (rolling regressions using a one-year window of daily post-ranking portfolio returns) to individual stocks
long-memo	Long-memory beta	Application of fractionally integrated long-memory time series process

Panel B: ML estimators

	Hyperparameter	Specification	Definition
lm			
	None		
elanet			
	λ	(0,1)	General strength of the penalization
	p	{0,0.5,1}	Weight on the lasso and ridge penalization
rf			
	L	(1,10)	Depth of the single regression trees
	M	{20,25,30,35,40}	Number of predictors randomly considered as potential split variables
	B	(10,500)	Number of trees added to the ensemble prediction
gbrt			
	L	(1,5)	Depth of the single regression trees
	ν	{0.01,0.05,0.1}	Weight for the learning rate shrinkage
	B	(10,500)	Number of trees added to the ensemble prediction
nn_1 – nn_5			
	$size_{batch}$	1000	Batch size
	$number_{epochs}$	100	Number of epochs
	patience	25	Number of iterations during which the value-weighted MSE is allowed to increase in the validation sample
	dropout rate	0.1	Fractional rate of input variables that are randomly set to zero at each iteration
	ensemble	10	Number of independent seeds used for each specification family

Figure A1
Stylized visualizations | Exponential decaying weights

This figure depicts a stylized visualization that helps explain the concept of exponential decaying weights (all observations’ weights sum to 1).


Long horizon (e.g., 100 days)

Short horizon (e.g., 10 days)

Internet Appendix B: Machine learning-based estimators

In this appendix, we briefly introduce the representative set of machine learning techniques listed in Section 4.2. We begin with linear regressions, continue with regression trees, and end up with neural networks. We also note the major differences among the model families. Table A1 provides (in Panel B) the definitions and specifications of hyperparameters of each machine learning-based beta estimator.

Linear regressions

OLS regressions *(lm)* constitute the least complex approach in our empirical analysis. At each re-estimation date, we use the training sample to run a pooled OLS regression of future realized betas $\beta_{i,t+k}^{R}$ on the set of eighty-one predictors.[55] In line with Petkova and Zhang (2005), who argue that the relationship between firm characteristics and beta varies over the business cycle, we divide this set of predictors into firm characteristics (including industry dummies), $z_{i,t}$, and the default spread, x_t, which we choose as an indicator of the state of the economy (Jagannathan and Wang, 1996). We follow Cosemans et al. (2016), and include interactions between firm characteristics and the default spread in the regression model, i.e.,[56]

$$\beta_{i,t+k}^{R} = \delta_0 + \delta_1 x_t + \delta_2' z_{i,t} + \delta_3' x_t z_{i,t} + \varepsilon_{i,t+k}. \tag{4}$$

Incorporating interactions between covariates might add incremental explanatory power, but it further increases the number of predictors. However, a high dimensionality problem arises when the number of predictors becomes very large relative to the number of observations. Particularly in a prediction context, because the convexity of the traditional least squares objective tends to emphasize heavy-tailed observations, with an increasing number of predictors simple linear models begin to overfit noise rather than extract signals, thereby undermining the stability of predictions.

[55] Focusing on a similar type of forecast objective (predicting future stock-level excess returns), Lewellen (2015) and Drobetz et al. (2019) show that this approach is promising in capturing cross-sectional variation in the dependent variable, from both a statistical and an economic perspective. Although Drobetz and Otto (2021) use Fama and MacBeth (1973) regressions (FM regressions) that are re-estimated on a monthly basis, they show that the OLS-based model provides nearly identical predictions in a sample-splitting and re-estimation setting similar to ours. To ensure comparability with the machine learning models that cannot be re-estimated on a monthly basis (due to computational limitations), we use pooled OLS regressions as a proxy for the FM model.

[56] For the sake of parsimony, we do not include interactions between industry dummies and the default spread.

The most common machine learning technique to overcome the overfitting problem in a high-dimensional regression setting is the ***penalized least squares approach***. It modifies the least squares loss function by adding a penalty term, denoted as $\Phi(\theta)$, to favor more parsimonious model specifications, i.e.,

$$l(\theta) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \left((\beta_{i,t+k}^{R} - g\left(z_{i,t};\theta\right) \right)^2 + \Phi(\theta). \quad (5)$$

This helps identify which predictors are informative and omit those that are not (see, e.g., Gu, Kelly, and Xiu, 2020, for details). We use the *elastic net* approach *(elanet)*, which combines the lasso and ridge methodologies.[57] It computes the weighted sum of both penalties to increase flexibility, i.e.,

$$\Phi(\theta) = \lambda(1-p) \sum_{j=1}^{P} \left|\theta_j\right| + \lambda p \sum_{j=1}^{P} \left(\theta_j\right)^2. \quad (6)$$

The tuning parameters in this forecast model are $\lambda \in (0,1)$ and $p \in (0,1)$. λ indicates the general strength of the penalty (in particular, how strongly the regression coefficients are forced to zero); p denotes the relative weights of the lasso and the ridge approaches. $p = 0$ corresponds to lasso; and $p = 1$ corresponds to ridge.

Importantly, if not explicitly included as *pre-determined* terms, pooled regressions cannot capture any interactions or nonlinear effects (neither simple nor penalized approaches). We thus utilize linear regressions as a benchmark to identify whether such effects, on top of the two-way interaction between firm characteristics and the default spread, can lead to incremental predictive power. Note that both regression trees and neural networks incorporate multi-way interactions and nonlinearity inherently, without the necessity to add new predictors capturing these effects in advance.

[57] The lasso approach penalizes the sum of absolute coefficients, thereby setting regression coefficients of a subset of predictors to exactly zero (variable selection). The ridge approach penalizes the sum of squared regression coefficients, thereby only pushing coefficients close to zero (variable shrinkage). We also test the lasso and ridge methodologies separately. We find no improvement in predictive performance relative to the elastic net approach, so we do not present those results here.

Tree-based models

The idea behind ***regression trees*** is that they adaptively split the dataset into groups of observations that behave in similar manners. They follow an iteration process that is inspired by the growing behavior of real trees in nature (see Figure B1): First, the process begins with one initial node, the root, in order to find the optimal split variable and the optimal split value for it by minimizing the value-weighted MSE within each partition. This results in two nodes with minimized impurity. Second, to further disentangle the dataset, the algorithm determines optimal split variables and values on the subsamples left over from the preceding step(s) to iteratively grow the regression tree. This results in multiple final nodes with minimized impurity, the leaves. The predicted beta for each leaf reflects the simple average of the historically realized betas of the firms sorted into this leaf. Regression trees are invariant to monotonic transformations of predictors, able to incorporate categorical and numerical data in the same forecast models, and designed to inherently capture interactions and nonlinearity. However, they are prone to overfitting and must be strongly regularized. To accomplish this, we use the ensemble forecast approach that aggregates forecasts from many different regression trees into a single one. There are two common methods: 1) bagging, and 2) boosting.

Random forests (rf) modify Breiman's (2001) traditional bagging approach. The idea is to draw multiple bootstrap samples of the original dataset, fit deep trees independently, and then average their predictions into an ensemble forecast, creating a single strong learner. Because dominant predictors are always more likely to become split variables at low levels, which can lead to large correlations between bootstrap-replicated trees, random forests apply the so-called "dropout" method. At each potential branch, they randomly drop out predictors, leaving only a subsample of predictors to be selected as potential split variables. The tuning parameters in this forecast model are the depth of trees L, the number of predictors M randomly considered as potential split variables, and the number of trees B added to the ensemble prediction.

In contrast, *gradient boosted regression trees (*$gbrt$*)* follow the boosting approach, which is based on the idea that combining multiple shallow trees creates a single strong learner, stronger even than a single deep tree. The iterative procedure is as follows: It computes a first shallow tree to fit the realized betas. This oversimplified tree exhibits a high forecast error. Next, it computes a second shallow tree, fitting the forecast residuals from the first tree. The forecasts from these two trees are then added together to form an ensemble prediction. To avoid overfitting the forecast residuals, one has to shrink the forecast component from the second tree by a factor $\nu \epsilon (0{,}1)$. Each additional shallow tree fits the forecast residual from the preceding ensemble prediction, and its shrunk forecast component is added accordingly to the ensemble forecast. The tuning parameters in this forecast model are the depth of the trees L, the shrinkage weight ν, and the number of trees B added to the ensemble prediction.

Neural networks

Neural networks (nn) are the most complex method in our empirical analysis. They are highly parameterized, which makes them suitable for solving very complicated machine learning problems. However, they are opaque and can be difficult to interpret. In general, they map inputs (predictors) to outputs (realized betas). Inspired by the functioning of the human brain, they are composed of many interconnected computational units, called “neurons”. Each neuron on its own provides very little predictive power, but a network of multiple neurons functions cohesively and improves the predictive performance. We use feed-forward neural networks, where each node has a connection to all the nodes in the previous layer and the connections follow a one-way direction, from input to output layer. The input layer contains the predictor variables (e.g., lagged firm characteristics), while the output layer contains a prediction for the dependent variable (realized betas). The simplest neural network (without any hidden layer) equals the OLS regression model. Adding hidden layers leads from shallow to deep architectures, which are able to capture interactions and nonlinear effects (see Figure B2, Panel A).

Neural networks predict the output y as the weighted average of inputs x. In the simplest model, the OLS regression coefficients are taken as weights. In more complex architectures, the weights must be computed iteratively by using the “backpropagation” algorithm. As an initialization, this algorithm assigns random weights to each connection. It also calculates the initial value-weighted MSE based on the predictions derived from the inputs of the (last) hidden layer. It then proceeds iteratively as follows: First, it recursively (from output to input layer) computes the gradient of the value-weighted MSE with regard to the weights. Second, it adjusts the weights slightly in the *opposite* direction of the computed gradients, because the objective is to *minimize* the value-weighted MSE. Third, based on the adjusted weights, it recalculates the value-weighted MSE.

The iteration process, known as “gradient descent”, stops when the value-weighted MSE is ultimately minimized. Thus far, we assume that each node in the hidden layer creates a signal (i.e., it is incorporated into the computation of the weighted average). In the human brain, however, neural networks work somewhat differently. To avoid noise, a specific node transforms each of the preceding signals it transmits (if at all). For example, it may amplify or condense the preceding signals, or only create a signal if the accumulated impulse is sufficiently strong. Thus, at each node, the weighted average of the preceding signals (x, which stems either from the input or the preceding layer) is subject to an activation function (see Figure B2, Panel B).[58]

Following Gu, Kelly, and Xiu (2020), we choose the rectified linear unit (ReLU) activation function and apply it to each node in the hidden layers. To encourage sparsity in the number of active neurons, it only provides a signal as an output if the information from the preceding layer accumulates beyond a threshold: $ReLU(x) = \begin{cases} 0 \; if \; x < 0 \\ x \; otherwise \end{cases}$.

In our empirical analysis, we consider neural networks with up to five hidden layers (HL) and up to thirty-two neurons (N), which we choose according to the geometric pyramid rule (Masters, 1993).[59] In line with Gu, Kelly, and Xiu (2020) and Drobetz and Otto (2021), we simultaneously apply different types of regularization to ensure computational feasibility and avoid overfitting. In addition to the ReLU activation and a lasso-based penalization of the weights, we use the *stochastic gradient descent* (SGD) approach to train the neural networks. During the iteration process, the algorithm cuts the training sample into small random subsamples, so-called “batches”, and uses one at each iteration. This leads to strong improvements in computational speed. The algorithm still sees the entire training sample (consecutively, not contemporaneously, and at least once but usually multiple times), which helps incorporate all available information

[58] While all weight transformations in the different nodes are purely linear, it is the activation function that allows neural networks to capture nonlinearity.

[59] The neural network architectures are: nn_1 ($HL = 1; N = \{32\}$), nn_2 ($HL = 2; N = \{32,16\}$), nn_3 ($HL = 3; N = \{32,16,8\}$), nn_4 ($HL = 4; N = \{32,16,8,4\}$), and nn_5 ($HL = 5; N = \{32,16,8,4,2\}$).

and thus avoids impairing the predictive performance. Consequently, the number of iterations depends on the size of the batches and the number of epochs (i.e., the number of times the algorithm sees the entire training sample).

Second, we adopt the batch normalization algorithm introduced by Ioffe and Szegedy (2015). It is intended to mitigate the internal covariate shift that occurs as the distribution of each hidden layer's inputs changes during the training (as the parameters of the preceding layers change) and slows down the learning process. To this end, within each batch, it cross-sectionally normalizes the input to each hidden layer.

Third, we apply *learning rate shrinkage* (see Figure B2, Panel C). The learning rate determines the size of the incremental steps in the gradient, while iteratively minimizing the value-weighted MSE. It faces a trade-off between finding the global minimum instead of the local counterpart (smaller learning rate) and computational speed (larger learning rate). This regularization procedure begins with a larger learning rate to speed up computation. As the gradient approaches zero, it shrinks the learning rate towards zero to overcome a potential local minimum.

Fourth, we implement *early stopping* (see Figure B2, Panel D), as neural networks aim to minimize the value-weighted MSE in the training sample. This regularization terminates the SGD iteration process when the value-weighted MSE in the validation sample increases for a pre-specified number of iterations (so-called “patience”), which also speeds up computation.

Fifth, we adopt the *ensemble* approach proposed by Hansen and Salamon (1990) and Dietterich (2000). We compute ten neural networks from the same specification family at each re-estimation date, using independent seeds.[60] We then average over the predictions to increase the signal-to-noise ratio, since the stochastic nature of the SDG approach leads to different forecasts for different seeds.

[60] Seeds are numbers used to initialize random processes. This approach ensures different but reproducible predictions.

Lastly, in addition to the regularization applied by Gu, Kelly, and Xiu (2020), we employ the *dropout* method. It randomly sets a fraction rate of input variables to exactly zero at each iteration, and thus is one of the most effective methods in the neural network framework to prevent overfitting.

Neural networks are computationally intensive and can be specified in an innumerable number of different architectures. This is why we retreat from tuning parameters (e.g., the size of batches or the number of epochs) and instead pre-specify five different models. We assume that our nn_1 to nn_5 architectures serve as a conservative lower bound for the predictive performance of neural network models in general. Empirically, as shown in Section C of the Internet Appendix, the predictive performance for the neural network models deteriorates slightly in the number of hidden layers. In the main part of the paper, we thus only present and discuss the results for the simplest nn_1 architecture.

Figure B1
Stylized visualizations | Regression trees

This figure depicts a stylized visualization that helps explain the structure and functioning of regression trees (adapted from Gu, Kelly, and Xiu, 2020).

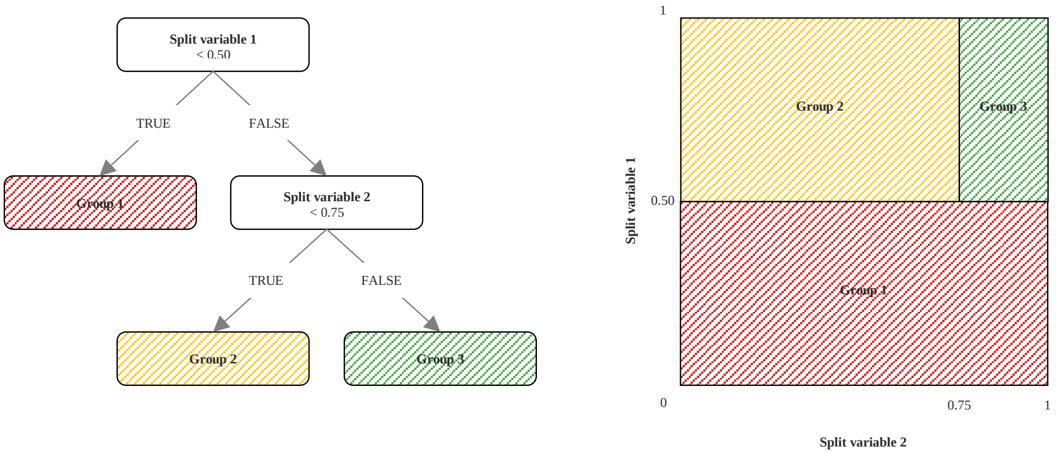
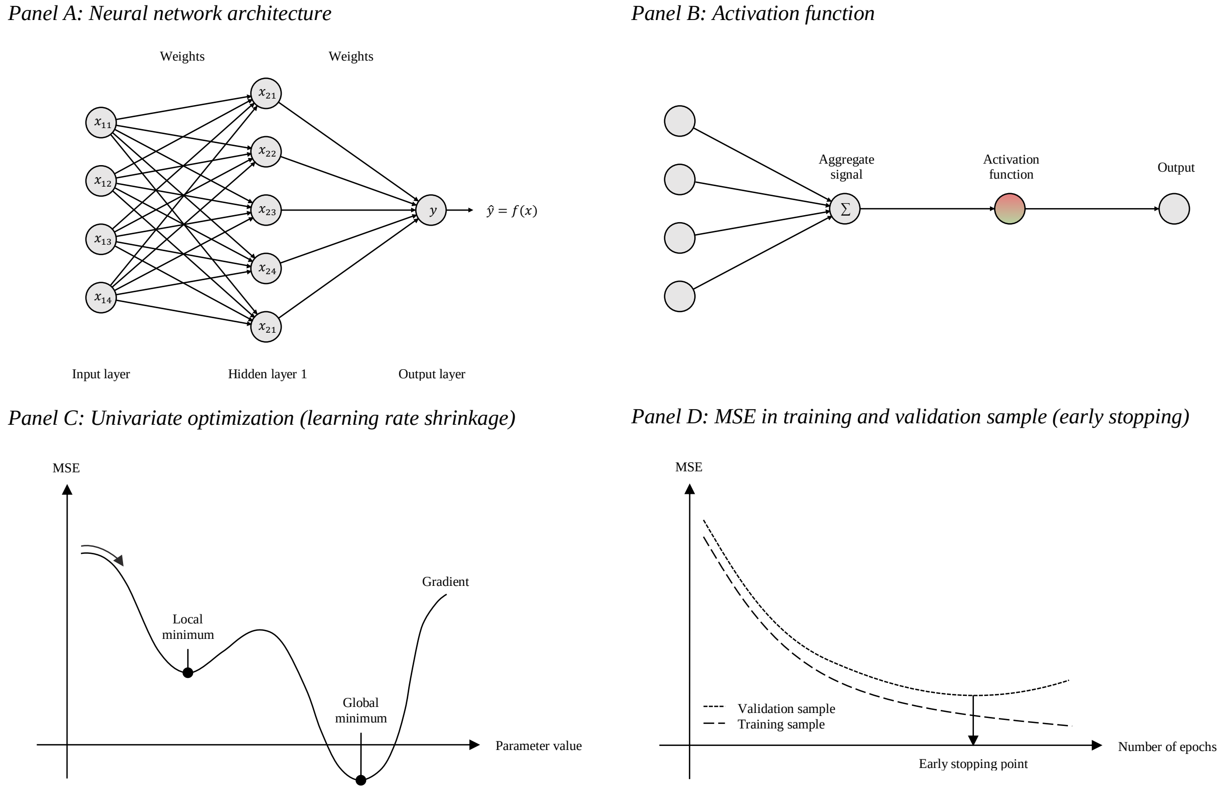


Figure B2
Stylized visualizations | Neural networks

This figure depicts four stylized visualizations that help explain the structure, functioning, and regularization of neural networks (adapted from Drobetz and Otto, 2021).



Panel A: Neural network architecture

Panel B: Activation function

Panel C: Univariate optimization (learning rate shrinkage)

Panel D: MSE in training and validation sample (early stopping)

Internet Appendix C: Further analyses and robustness tests

In this appendix, we present results of further analyses and conduct several robustness tests. In Figure C1, we further investigate the characteristics of the market-neutral minimum variance portfolios introduced in Section 5.1.5. In particular, we relate the weights of the stocks in the optimized portfolios to their beta forecasts to gain a better understanding of the results in Table 5. To this end, we replicate the procedure outlined for Figure 2, which presents the average forecast error (grey bars) for decile portfolios formed by sorting stocks on their predicted beta. To these visualizations, we add the average summed minimum variance weights (black unfilled circles) obtained from the construction of market-neutral MVPs. Figure C1 reveals that the average forecast errors for all estimation approaches are (slightly) larger for extreme beta deciles, and particularly large for the top decile portfolios.[61]

In Table C1, we examine the robustness of our main results to changes in the specifications of the machine learning-based estimators considered in the empirical analysis. First, we present additional results for the neural network architectures with two to five hidden layers. We find that these more complex models perform comparably well as the main neural network architecture with only one hidden layer. However, the average forecast errors are slightly larger, indicating that the less complex models generally perform somewhat better. This result is consistent with findings in Gu, Kelly, and Xiu (2020) and Drobetz and Otto (2021), and is due to overfitting. Deeper neural networks seem to be too complex for the relatively small dataset and parsimonious set of predictors, and/or the monthly forecast model setting we use in our empirical analysis.

Second, we test the robustness to including additional macroeconomic variables on top of just the default spread (dfy). That is, following Welch and Goyal (2008), we add the *treasury-bill rate* (tbl), the *treasury-bill rate volatility* (tbl_sd), the *term spread* (tms), the *stock variance* ($svar$), both the *earnings-to-price ratio* (ep) and the *dividend payout ratio* (dp) at a market level, and the *consumption, wealth and*

[61] The average forecast errors for the bottom decile portfolios are notably lower in Figure A1 than in Figure 2. This is well explained by the exclusion of microcaps, which tend to be assigned to low-beta deciles (Fama and French, 1992), from the stock universe in this empirical test. As documented by Andersen, Bollerslev, and Meddahi (2005), microcaps also tend to exhibit the most pronounced measurement errors due to microstructure noise.

income ratio (cay). In addition, we include measures for industrial production, inflation, and unemployment as provided by the Federal Reserve Bank of St. Louis. Given our findings of the low variable importance for dfy, we expect additional macroeconomic variables to make little difference. Indeed, the results in Table C1 for the random forests with the additional macroeconomic variables are qualitatively similar. Thus, the additional macroeconomic variables do not help improve the predictive performance, but their inclusion does not hurt it either.

Third, we present the results for an ensemble prediction for which we average the predictions of the lm, $elanet$, rf, $gbrt$, and nn_1 models (ens_1), the $elanet$, rf, $gbrt$, and nn_1 models (ens_2), and the rf, $gbrt$, and nn_1 models (ens_3), respectively. Indeed, all ensemble approaches work slightly better than any single machine learning technique. We also find that including only those machine learning techniques that work best in isolation, i.e., the rf, $gbrt$, and nn_1 models, also leads to the best ensemble predictions. One caveat is that applying this ensemble approach in practice is computationally intensive as one first needs to estimate each of the three to five forecast models, respectively.

In a further step, we examine the robustness of our main results to changes in the forecast error measure. In particular, we test the robustness to using equal-weighted MSEs in Table C2, i.e.,

$$MSE_{t+k|t} = \sum_{i=1}^{N_t} (\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F})^2 \text{, with } k = 12, \tag{7}$$

and to using value-weighted mean absolute errors (MAEs) in Table C3, i.e.,

$$MSE_{t+k|t} = \sum_{i=1}^{N_t} w_{i,t} |\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F}| \text{, with } k = 12. \tag{8}$$

We find that the equal-weighted MSEs are notably higher for all approaches than for the value-weighted examination. This is consistent with previous results showing that it is considerably more difficult to estimate the betas of small stocks than it is for large stocks. Moreover, the machine learning-based approaches outperform the benchmark models when weighting equally on an even larger scale. Thus, they appear to be even more beneficial for small than large stocks. In the MAE framework, all forecast errors are penalized in the same way. Consequently, large forecast errors are less influential than in the MSE framework. Nevertheless, machine learning-based approaches still outperform the benchmark models, which indicates that differences in predictive performance are not predominantly driven by severe outliers in the forecast errors for just a few stocks.

Finally, we use Mincer and Zarnowitz (1969) regressions in Table C4 to test for the unbiasedness of the different estimation approaches. Following Fama and MacBeth (1973), we run either a WLS regression (using the stocks' market capitalization-based weights) or an OLS regression (using equal weights) of realized betas on the beta forecasts obtained from the different models each month, i.e., $\beta_{i,t+k}^{R} = a_t + b_t \beta_{i,t+k|t}^{F} + e_{i,t+k}$. Table C4 reports the time series averages of monthly intercepts (a) and slopes (b) and the *t*-statistics (in parentheses) testing the null hypotheses that $a = 0$ and $b = 1$, respectively. For the *t*-tests, we use Newey and West (1987) standard errors with eleven lags. Consistent with our previous results, we find that the machine learning models that performed best thus far are also the least biased. For all machine learning techniques, the average intercept is closer to zero and the slope is closer to one (with *t*-statistics being mostly insignificant). In contrast, in the vast majority of cases, we have to reject the unbiasedness hypotheses for the established models (according to the large *t*-statistics).

Table C1
Forecast errors (additional forecast models)

This table examines the differences in forecast errors produced by the forecast models introduced in Sections 4 and C of the Internet Appendix. Panel A reports the time series means for monthly value-weighted MSEs, i.e., $MSE_{t+k|t} = \sum_{i=1}^{N_t} w_{i,t}(\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F})^2$, with $k = 12$, where $w_{i,t}$ is stock i's market capitalization-based weight. Panel B reports the fraction of months during the out-of-sample period for which the column model is 1) in the Hansen, Lunde, and Nason (2011) model confidence set (MCS), and 2) significantly better than the row model in a pairwise comparison (according to the Diebold and Mariano (1995) test (DM test) statistics). The DM tests of equal predictive ability inspect differences in stock-level squared forecast errors (SEs), i.e., $SE_{i,t+k|t} = (\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F})^2$, with $k = 12$. The DM test statistic in month t for comparing the model under investigation j with a competing model i is $DM_{ij,t} = \frac{\bar{d}_{ij,t}}{\hat{\sigma}_{\bar{d}_{ij,t}}}$, where $d_{ij,t} = SE_{i,t+k|t} - SE_{j,t+k|t}$ is the difference in SEs, $\bar{d}_{ij,t} = \sum_{i=1}^{N_t} w_{i,t} d_{ij,t}$ is the value-weighted cross-sectional average of these differences, and $\hat{\sigma}_{\bar{d}_{ij,t}}$ is the Newey and West (1987) standard error of $d_{ij,t}$ (with four lags to account for possible heteroskedasticity and autocorrelation). Positive signs of $DM_{ij,t}$ indicate superior predictive performance of model j relative to model i in month t, i.e., that model j yields, on average, lower forecast errors than model i. All statistical tests are based on the 10% significance level. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.

		Baseline specifications							Alternative specifications							
		ols_1y_d	long-memo	lm	elanet	rf	gbrt	nn_1	nn_2	nn_3	nn_4	nn_5	rf_amv	ens_1	ens_2	ens_3
Panel A: Average forecast errors																
	MSE, v.w. [%]	9.21	7.83	9.27	8.84	7.49	7.65	7.70	7.76	7.85	7.86	7.82	7.67	7.61	7.49	7.35
Panel B: Forecast errors over time																
	In MCS	43.45	64.03	39.71	45.53	75.05	67.78	73.39	67.36	67.78	67.57	69.85	69.44	76.92	78.59	81.70
Baseline specifications	vs. ols_1y_d		54.89	37.21	43.87	61.12	59.25	57.80	55.30	55.30	54.47	54.68	59.88	55.93	59.25	62.79
	vs. long-memo	17.26		12.68	20.37	35.76	35.14	37.63	35.97	35.76	35.55	33.68	33.89	40.75	43.04	42.83
	vs. lm	34.30	48.02		37.84	65.07	61.54	66.53	60.91	60.29	61.12	60.91	59.25	83.78	77.34	74.43
	vs. elanet	31.19	39.92	9.98		55.93	51.98	55.09	51.98	51.56	49.90	51.14	54.89	75.68	75.26	64.66
	vs. rf	8.11	20.79	3.33	8.11		23.49	28.69	28.48	27.65	28.27	22.66	14.55	31.19	36.80	47.40
	vs. gbrt	11.02	22.04	4.57	8.73	31.39		25.78	23.70	24.95	21.41	21.62	30.98	31.60	41.16	53.85
	vs. nn_1	14.35	21.00	3.95	9.56	29.73	21.21		15.38	19.33	12.06	14.55	28.07	28.48	32.22	40.12
Alternative specifications	vs. nn_2	17.46	22.87	3.74	10.40	38.25	27.86	33.47		21.83	19.33	24.74	34.10	36.38	40.75	48.02
	vs. nn_3	16.22	23.49	4.78	11.02	37.42	29.11	38.67	29.11		25.99	30.15	34.30	36.17	40.33	47.82
	vs. nn_4	16.22	24.53	4.78	9.77	36.38	27.44	34.93	30.98	25.57		26.20	35.14	33.26	40.33	46.99
	vs. nn_5	16.84	23.28	3.74	9.56	37.21	27.86	28.69	29.31	31.60	26.40		33.06	31.81	40.54	49.69
	vs. rf_amv	11.02	23.49	3.74	11.02	37.21	27.03	32.02	31.60	30.77	32.02	30.35		35.14	42.62	48.86
	vs. ens_1	10.81	19.54	0.00	1.46	25.78	14.76	27.44	22.87	21.00	17.67	13.93	23.08		39.71	38.88
	vs. ens_2	9.56	18.09	0.62	1.66	23.08	11.23	21.21	18.50	15.80	12.47	8.94	22.66	8.73		36.59
	vs. ens_3	9.36	15.80	1.66	3.95	16.22	7.07	16.63	14.35	14.14	9.56	8.32	15.59	11.23	19.75	
	T	481	481	481	481	481	481	481	481	481	481	481	481	481	481	481

Table C2
Forecast errors (equal-weighted MSEs)

This table examines the differences in forecast errors produced by the forecast models introduced in Section 4. Panel A reports the time series means for monthly equal-weighted MSEs, i.e., $MSE_{t+k|t} = \sum_{i=1}^{N_t}(\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F})^2$, with $k = 12$. Panel B reports the fraction of months during the out-of-sample period for which the column model is 1) in the Hansen et al. (2011) model confidence set (MCS), and 2) significantly better than the row model in a pairwise comparison (according to the Diebold and Mariano (1995) test (DM test) statistics). The DM tests of equal predictive ability inspect differences in stock-level squared forecast errors (SEs), i.e., $SE_{i,t+k|t} = (\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F})^2$, with $k = 12$. The DM test statistic in month t for comparing the model under investigation j with a competing model i is $DM_{ij,t} = \frac{\bar{d}_{ij,t}}{\hat{\sigma}_{\bar{d}_{ij,t}}}$, where $d_{ij,t} = SE_{i,t+k|t} - SE_{j,t+k|t}$ is the difference in SEs, $\bar{d}_{ij,t} = \frac{1}{\mathrm{N_t}}\sum_{i=1}^{N_t} d_{ij,t}$ is the equal-weighted cross-sectional average of these differences, and $\hat{\sigma}_{\bar{d}_{ij,t}}$ is the Newey and West (1987) standard error of $d_{ij,t}$ (with four lags to account for possible heteroskedasticity and autocorrelation). Positive signs of $DM_{ij,t}$ indicate superior predictive performance of model j relative to model i in month t, i.e., that model j yields, on average, lower forecast errors than model i. All statistical tests are based on the 10% significance level. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.

		Established estimators										ML estimators				
		ols_5y_m	ols_1y_d	ewma_s	ewma_l	bsw	vasicek	karolyi	hybrid	fama-french	long-memo	lm	elanet	**rf**	gbrt	nn_1
Panel A: Average forecast errors																
	MSE, v.w. [%]	52.73	22.90	24.30	22.95	17.63	18.45	19.27	19.75	17.31	16.69	17.21	17.02	**15.94**	16.26	15.77
Panel B: Forecast errors over time																
	In MCS	0.00	2.91	4.99	3.95	18.92	13.72	9.77	8.11	26.61	36.17	19.33	24.32	**44.70**	42.00	67.36
Established estimators	vs. ols_5y_m		97.51	92.72	96.05	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	**100.00**	100.00	100.00
	vs. ols_1y_d	0.00		21.41	35.97	91.48	83.58	72.14	55.93	89.60	87.11	76.92	82.33	**86.49**	86.69	86.07
	vs. ewma_s	0.00	44.91		63.62	88.15	82.54	71.52	56.34	82.54	84.20	76.51	81.50	**84.82**	85.45	85.24
	vs. ewma_l	0.00	27.65	9.36		89.19	82.12	66.94	50.10	84.82	82.95	74.64	80.25	**85.03**	85.45	83.99
	vs. bsw	0.00	1.66	3.95	3.33		12.27	6.44	17.05	34.93	46.78	53.01	54.26	**71.52**	70.27	72.77
	vs. vasicek	0.00	2.70	4.16	4.37	55.30		5.82	24.32	48.02	54.89	61.12	64.24	**75.05**	75.88	77.55
	vs. karolyi	0.00	5.41	5.82	6.24	67.78	59.04		36.80	55.30	63.62	63.62	68.61	**77.96**	80.04	79.83
	vs. hybrid	0.00	20.17	18.30	21.83	64.03	58.00	49.06		64.24	69.23	70.27	72.14	**80.87**	81.50	82.12
	vs. fama-french	0.00	1.87	4.16	3.33	28.27	21.41	11.43	17.67		43.87	48.44	52.18	**63.83**	64.66	69.02
	vs. long-memo	0.00	4.16	3.33	4.37	23.08	18.09	12.89	9.15	26.82		33.06	35.97	**50.73**	51.77	57.17
ML estimators	vs. lm	0.00	10.60	11.64	12.06	28.27	20.58	17.26	12.47	32.43	40.12		47.61	**64.03**	61.95	77.55
	vs. elanet	0.00	8.94	8.52	9.15	24.95	17.88	14.35	12.47	29.73	37.84	16.22		**65.07**	59.67	72.77
	vs. rf	**0.00**	**8.73**	**8.94**	**9.56**	**16.01**	**15.18**	**11.64**	**8.52**	**20.37**	**22.66**	**8.11**	**11.02**		**21.62**	**46.99**
	vs. gbrt	0.00	9.36	8.94	9.98	16.84	14.35	12.06	8.52	20.79	25.57	12.06	13.93	**34.72**		53.01
	vs. nn_1	0.00	7.48	7.48	7.69	15.59	13.10	10.60	8.11	16.42	21.41	2.29	5.41	**25.16**	18.09	
	T	481	481	481	481	481	481	481	481	481	481	481	481	**481**	481	481

Table C3
Forecast errors (value-weighted MAEs)

This table examines the differences in forecast errors produced by the forecast models introduced in Section 4. Panel A reports the time series means for monthly value-weighted MAEs, i.e., $MAE_{t+k|t} = \sum_{i=1}^{N_t} w_{i,t} |\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F}|$, with $k = 12$, where $w_{i,t}$ is stock i's market capitalization-based weight. Panel B reports the fraction of months during the out-of-sample period for which the column model is 1) in the Hansen et al. (2011) model confidence set (MCS), and 2) significantly better than the row model in a pairwise comparison (according to the Diebold and Mariano (1995) test (DM test) statistics). The DM tests of equal predictive ability inspect differences in stock-level absolute forecast errors (AEs), i.e., $AE_{i,t+k|t} = |\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F}|$, with $k = 12$. The DM test statistic in month t for comparing the model under investigation j with a competing model i is $DM_{ij,t} = \frac{\bar{d}_{ij,t}}{\hat{\sigma}_{\bar{d}_{ij,t}}}$, where $d_{ij,t} = AE_{i,t+k|t} - AE_{j,t+k|t}$ is the difference in AEs, $\bar{d}_{ij,t} = \sum_{i=1}^{N_t} w_{i,t} d_{ij,t}$ is the value-weighted cross-sectional average of these differences, and $\hat{\sigma}_{\bar{d}_{ij,t}}$ is the Newey and West (1987) standard error of $d_{ij,t}$ (with four lags to account for possible heteroskedasticity and autocorrelation). Positive signs of $DM_{ij,t}$ indicate superior predictive performance of model j relative to model i in month t, i.e., that model j yields, on average, lower forecast errors than model i. All statistical tests are based on the 10% significance level. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.

		Established estimators										ML estimators				
		ols_5y_m	ols_1y_d	ewma_s	ewma_l	bsw	vasicek	karolyi	hybrid	fama-french	long-memo	lm	elanet	**rf**	gbrt	nn_1
Panel A: Average forecast errors																
	MSE, v.w. [%]	32.27	22.58	22.40	22.27	21.60	21.70	21.75	21.40	21.91	20.89	23.07	22.41	**20.41**	20.61	20.71
Panel B: Forecast errors over time																
	In MCS	7.07	45.95	55.30	55.93	62.99	62.58	62.37	61.95	60.29	76.72	56.13	62.58	**88.36**	81.08	80.46
Established estimators	vs. ols_5y_m		82.74	82.54	83.58	86.69	85.65	85.03	91.06	87.11	93.76	87.94	88.77	**93.56**	93.76	93.35
	vs. ols_1y_d	1.25		29.73	42.00	65.49	73.80	75.47	60.91	38.67	50.73	31.19	40.33	**54.47**	53.01	52.60
	vs. ewma_s	0.83	23.08		32.02	44.49	42.83	43.04	42.62	30.56	48.44	30.15	38.88	**51.56**	51.35	50.10
	vs. ewma_l	0.62	17.05	20.37		46.78	46.15	44.91	45.32	32.22	47.61	30.56	38.67	**50.52**	50.52	49.06
	vs. bsw	0.83	5.20	13.72	12.89		22.87	17.67	34.51	18.71	41.16	22.25	31.19	**46.57**	44.91	42.41
	vs. vasicek	1.04	6.24	13.93	12.89	20.17		16.01	39.29	16.42	40.96	21.83	30.35	**46.57**	45.95	42.83
	vs. karolyi	1.25	2.08	13.31	11.85	20.17	27.65		37.01	19.54	41.16	21.83	32.02	**47.61**	45.32	44.49
	vs. hybrid	0.21	8.94	16.42	17.05	24.12	26.82	26.82		16.22	35.34	16.22	24.95	**41.37**	41.58	38.88
	vs. fama-french	2.29	13.51	16.01	16.22	26.82	22.45	24.95	28.48		40.75	17.26	29.31	**48.65**	48.86	43.04
	vs. long-memo	0.83	17.26	19.33	19.75	21.41	21.41	22.45	21.00	19.33		8.32	15.80	**30.77**	30.77	33.26
ML estimators	vs. lm	6.65	33.68	34.51	35.14	40.33	38.46	39.50	40.54	37.01	48.44		35.55	**61.75**	58.21	65.07
	vs. elanet	6.65	32.85	31.81	33.26	36.80	35.97	36.38	35.14	32.43	39.09	12.47		**55.09**	50.73	52.81
	vs. **rf**	**2.29**	**9.15**	**9.77**	**10.81**	**14.55**	**12.06**	**14.35**	**12.27**	**7.69**	**21.62**	**2.70**	**6.65**		**25.16**	**26.82**
	vs. gbrt	1.87	9.56	11.43	10.60	15.18	13.72	16.01	14.55	9.36	20.79	4.37	7.28	**26.82**		23.08
	vs. nn_1	1.87	15.38	14.97	15.80	18.30	18.09	19.54	17.67	15.59	21.83	3.53	7.48	**26.82**	21.41	
	T	481	481	481	481	481	481	481	481	481	481	481	481	**481**	481	481

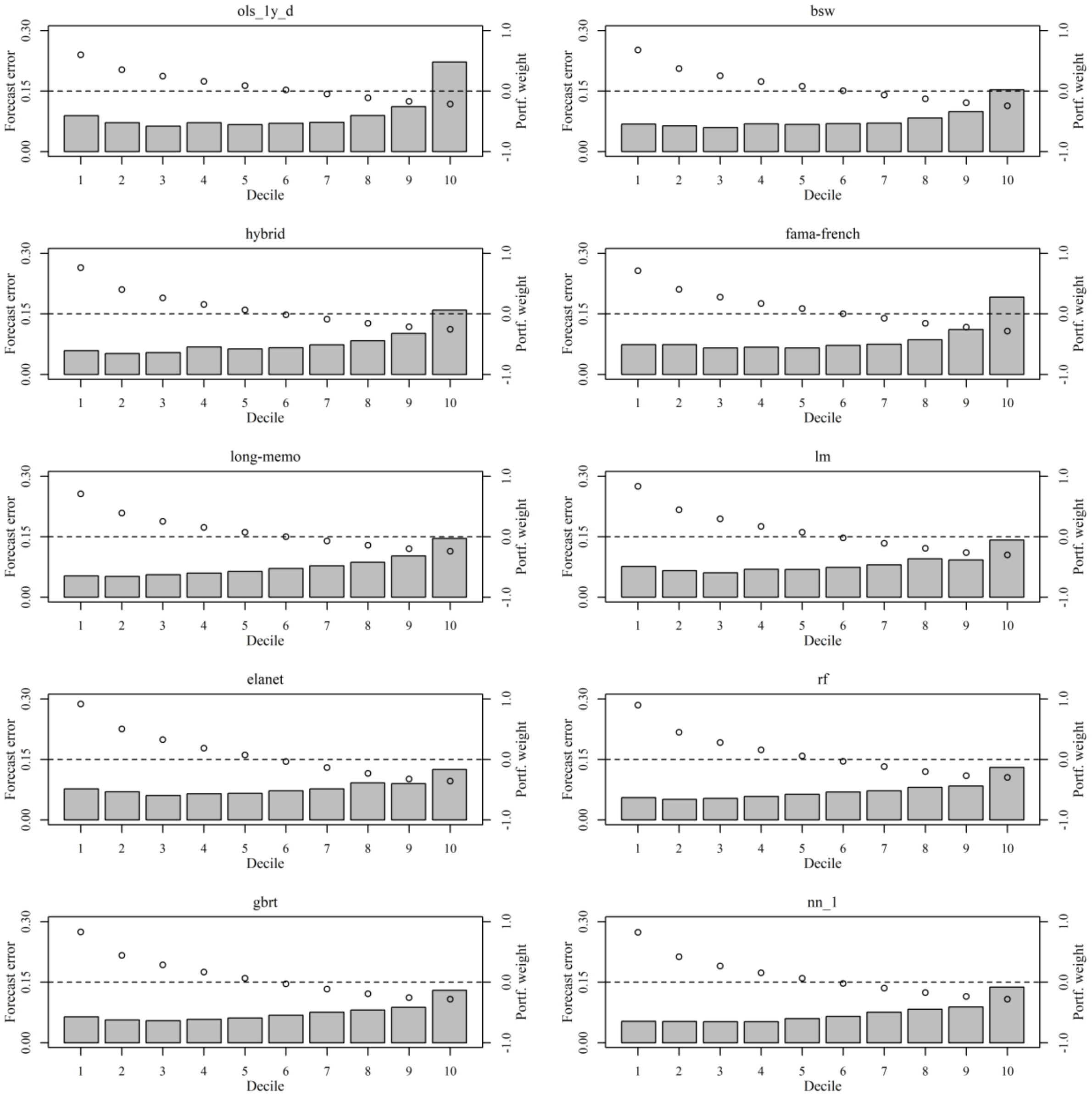
Table C4
Mincer and Zarnowitz (1969) unbiasedness test

This table reports the results of Mincer and Zarnowitz (1969) regressions to test for the unbiasedness of the forecast models introduced in Section 4. Following Fama and MacBeth (1973), either a WLS regression (using the stocks' market capitalization-based weights) or an OLS regression (using equal weights) of realized betas on the beta forecasts obtained from the different models is run each month, i.e., $\beta^{R}_{i,t+k} = a_t + b_t\beta^{F}_{i,t+k|t} + e_{i,t+k}$. In particular, this table reports the time series averages of monthly intercepts (a) and slopes (b) and the t-statistics (in parentheses) testing the null hypotheses that $a = 0$ and $b = 1$, respectively. The t-tests are based on Newey and West (1987) standard errors (with eleven lags to account for possible heteroskedasticity and autocorrelation). Panel A presents the value-weighted results (based on WLS regressions), while Panel B adds the results for equal weights (based on OLS regressions). The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.

		Panel A: Value-weighted		*Panel B: Equal-weighted*	
	Model	α	β	α	β
Established estimators	ols_5y_m	0.54	0.45	0.44	0.34
		(3.63)	(-4.66)	(1.72)	(-10.42)
	ols_1y_d	0.26	0.73	0.29	0.63
		(2.69)	(-3.15)	(7.92)	(-5.45)
	ewma_s	0.27	0.71	0.31	0.60
		(3.91)	(-4.54)	(9.30)	(-6.68)
	ewma_l	0.26	0.73	0.30	0.62
		(3.08)	(-3.61)	(8.37)	(-5.87)
	bsw	0.19	0.81	0.13	0.82
		(1.78)	(-1.97)	(3.46)	(-2.98)
	vasicek	0.17	0.83	0.06	0.87
		(1.65)	(-1.89)	(1.06)	(-1.69)
	karolyi	0.19	0.81	0.04	0.86
		(1.81)	(-2.08)	(0.77)	(-1.46)
	hybrid	0.11	0.87	0.04	0.85
		(0.84)	(-1.17)	(0.46)	(-2.12)
	fama-french	0.13	0.86	0.10	0.89
		(1.24)	(-1.46)	(2.88)	(-2.24)
	long-memo	0.11	0.87	0.14	0.83
		(1.43)	(-2.17)	(6.88)	(-7.15)
ML estimators	lm	-0.02	0.99	0.08	0.93
		(-0.09)	(-0.07)	(1.82)	(-1.79)
	elanet	-0.19	1.14	0.00	1.02
		(-1.26)	(1.18)	(-0.01)	(0.35)
	rf	**-0.12**	**1.12**	**0.00**	**1.04**
		(-1.01)	**(1.06)**	**(-0.01)**	**(0.60)**
	gbrt	-0.07	1.07	0.04	1.00
		(-0.61)	(0.68)	(0.71)	(0.06)
	nn_1	-0.03	1.02	0.04	1.00
		(-0.19)	(0.22)	(0.65)	(-0.00)

Figure C1
Average forecast errors and market-neutral minimum variance portfolio weights for decile portfolios based on beta forecasts

This figure plots the time series averages of monthly mean squared forecast errors (grey bars) for decile portfolios based on beta forecasts. To this end, the procedure outlined for Figure 2 is replicated but for a slightly different dataset. In particular, for the construction of market-neutral minimum variance portfolios (MVPs), the first five years of the sample period are omitted (due to the computation of market and idiosyncratic variances), and microcaps are excluded from the stock universe. Monthly forecast errors in this empirical test are defined as the value-weighted MSE between beta forecasts and realized betas over the next year within each portfolio. To these visualizations, the average summed minimum variance weights (black unfilled circles) are added. They are obtained from the construction of market-neutral MVPs. The baseline sample includes all firms that were or are publicly listed on the NYSE, AMEX, or NASDAQ in any given month during the July 1972–December 2020 sample period, while the first beta estimates are obtained in December 1979.


ols_1y_d
bsw
hybrid
fama-french
long-memo
lm
elanet
rf
gbrt
nn_1
Forecast error
Portf. weight
Decile