Journal of Financial Economics 172 (2025) 104138

Contents lists available at ScienceDirect

Journal of Financial Economics

journal homepage: www.elsevier.com/locate/finec

Machine learning from a "Universe" of signals: The role of feature engineering☆

Bin Li [a], Alberto G. Rossi [b,*], Xuemin (Sterling) Yan [c], Lingling Zheng [d]

[a] Economics and Management School, Wuhan University, PR China
[b] McDonough School of Business, Georgetown University, United States of America
[c] College of Business, Lehigh University, United States of America
[d] School of Business, Renmin University of China, PR China

ARTICLE INFO

Dataset link: Replication_Package_LRYZ_Machine_Learning_Feature_Engineering (Reference data)

Keywords:
Machine learning
Feature engineering
Return predictability
Cross-section of stock returns

ABSTRACT

We construct real-time machine learning strategies based on a "universe" of fundamental signals. The out-of-sample performance of these strategies is economically meaningful and statistically significant, but considerably weaker than those documented by prior studies that use curated sets of signals as predictors. Strategies based on a simple recursive ranking of each signal's past performance also yield substantially better out-of-sample performance. We find qualitatively similar results when examining past-return-based signals. Our results underscore the key role of feature engineering and, more broadly, inductive biases in enhancing the economic benefits of machine learning investment strategies.

1. Introduction

Machine learning methods have received considerable attention in the recent asset pricing literature, particularly in the area of return prediction (see, e.g., Chen et al., 2024c; Freyberger et al., 2020; Gu et al., 2020; Leippold et al., 2022). The general conclusions of the existing studies are remarkably similar— machine-learning models are superior to traditional models in predicting the cross-section of stock returns, and using machine-learning methods leads to large improvements in investment performance. Indeed, a common theme among many existing studies is constructing long–short investment strategies based on machine learning forecasts and demonstrating that these strategies are highly profitable.

While prior studies have clearly established the potential for large economic gains to investors using machine learning forecasts, an important issue that has been overlooked in the literature is the real-time performance of machine learning strategies, and particularly how the choice of input variables affects such performance. Specifically, many existing studies use published anomaly variables as predictors of stock returns and implicitly assume that they are known to investors during the training period, even though most anomalies are discovered years later.[1] While this approach is appropriate if the objective is to measure risk premium or estimate the stochastic discount factor, in which case we can take an econometrician's perspective and analyze data ex-post, such an approach raises the issue of whether investors could have

☆ Nikolai Roussanov was the editor for this article. We are grateful to the editor and an anonymous referee for their feedback and suggestions, which greatly improved the paper. We thank Marcin Kacperczyk, David Solomon, Allan Timmermann, and seminar and conference participants at the Chinese Finance Annual Meeting, Duke Kunshan University, Hunan University, Lehigh University, Nanjing University, Renmin University of China, Sun Yat-Sen University, Taiwan Finance Association Asset Pricing Conference, Tongji University, and Xiamen University for helpful comments and discussions. Bin Li acknowledges financial support from the National Natural Science Foundation of China (No. 72371191 and 71971164) and the Key Program of the National Social Science Fund of China (No. 24AZD020). Lingling Zheng acknowledges financial support from the National Natural Science Foundation of China (No. 72122021 and 72495154). We acknowledge the computational support provided by the Supercomputing Center of Wuhan University. All errors are our own.

* Corresponding author.
E-mail addresses: binli.whu@whu.edu.cn (B. Li), agr60@georgetown.edu (A.G. Rossi), xuy219@lehigh.edu (X. Yan), zhenglingling@rmbs.ruc.edu.cn (L. Zheng).

[1] There are exceptions. For example, Kozak et al. (2020) use shrinkage and selection method to construct a stochastic discount factor (SDF) from a comprehensive set of financial ratios compiled by WRDS.

https://doi.org/10.1016/j.jfineco.2025.104138
Received 15 September 2022; Received in revised form 1 July 2025; Accepted 4 July 2025
Available online 22 July 2025

selected those signals out of a universe of (potentially uninformative) signals in real time. As a consequence, the economic gains from using machine learning forecasts documented by the aforementioned studies are potentially overstated for real-time investors.

In this paper, we examine machine learning strategies based on a "universe" of over 18,000 fundamental signals that are accessible to investors in real time. Because these signals are constructed from financial statement variables using permutational arguments (Yan and Zheng, 2017), our strategies are not based on curated sets of inputs. By comparing machine-learning strategies based on a universe of signals with strategies based on selected sets of signals, our paper can shed light on the importance of feature engineering – i.e. the process of selecting and transforming the predictors used in machine-learning applications – for the performance of machine-learning strategies.[2,3] Such comparisons also provide insights into how human expertise influences machine learning models in predicting returns. Moreover, examining a universe of fundamental signals, rather than selecting a subset of them based on whether they have been published in academic journals, allows us to address the issue of publication bias (Harvey, 2017; Chen and Zimmermann, 2020).[4]

The primary machine learning method we use is boosted regression trees (BRT). We focus on BRT for several reasons. First, previous studies have shown that BRT exhibit strong predictive performance in finance applications. Gu et al. (2020), for example, show that BRT and neural networks are the two best-performing machine learning methods in predicting stock returns. Second, BRT are ideally suited for handling large, high-dimensional data sets because of their computational efficiency. This is important for us because our predictor set, which contains more than 18,000 signals, is much larger than those examined by previous studies. Third, BRT are robust to missing values and outliers. Given the findings in Gu et al. (2020), we also use neural networks as an alternative machine learning method to ensure the robustness of our findings.

We follow Gu et al. (2020) and partition our sample period into a training period, a cross-validation period, and an out-of-sample test period. We form long–short portfolios based on machine learning predicted returns, buying stocks with the highest predicted returns and shorting stocks with the lowest predicted returns. Using boosted regression trees (BRT) forecasts, our equal-weighted long–short portfolio generates an average return of 0.95% per month (t-statistic = 6.63) and an annualized Sharpe ratio of 1.02, while the value-weighted portfolio earns an average return of 0.40% per month (t-statistic = 2.34) and a Sharpe ratio of 0.30. In comparison, Gu et al. (2020) report that their BRT-based equal-weighted portfolios achieve a significantly higher 2.14% per month (Sharpe ratio = 1.73), and their value-weighted portfolios earn 0.99% per month (Sharpe ratio = 0.81), which are more than double the returns and Sharpe ratios we observe.

Neural networks also show weaker performance in our analysis. For equal-weighted portfolios, our strategies generate average returns of 0.80%–1.17% per month with Sharpe ratios of 0.74-1.16, compared to the 3.33% per month (Sharpe ratio = 2.45) documented by Gu et al. (2020). Similarly, for value-weighted portfolios, our neural networks yield returns of 0.21%–0.74% per month (Sharpe ratio = 0.16-0.70), whereas Gu et al. (2020) report 2.26% per month (Sharpe ratio = 1.35). Other studies, such as Chen et al. (2024c) and Freyberger et al. (2020), further emphasize this gap, reporting Sharpe ratios of 2.6 and 2.75, respectively, which significantly exceed our results.

To investigate whether the weaker performance is due to limitations in our machine learning implementation, we replicate our analyses on datasets of published anomalies. Using the Green et al. (2017, GHZ) sample, our BRT and neural network models achieve equal-weighted long–short returns exceeding 3.5% per month with Sharpe ratios of 2.21–2.81, matching the performance reported by Gu et al. (2020). The results are even stronger with the Chen and Zimmermann (2022, CZ) sample: BRT models achieve equal-weighted returns of 5.14% per month with an annualized Sharpe ratio of 3.64. These results indicate that our implementation can deliver performance on par with prior studies when using curated predictors. This suggests that the choice of input predictors, rather than the ML implementation itself, is the key driver of the performance gap between our strategies and those based on published signals.

Thus, compared to the previous literature that uses published – and hence implicitly selected – signals as return predictors, our results indicate that the economic gains to real-time investors from using our machine learning strategies are much more modest. Prior literature, such as Yan and Zheng (2017), however, has shown evidence that investors could obtain large economic gains by learning from a universe of return signals. To explore this possibility, we follow Yan and Zheng (2017) and construct a recursive ranking strategy. In particular, we first construct a long–short strategy based on each fundamental signal in our sample. We then sort all signals each year into deciles based on the t-stat of their past long–short portfolio alphas using a recursive window. Finally, we form an equal-weighted portfolio by going long in those signals ranked in the highest t-stat decile and shorting those signals ranked in the lowest t-stat decile.[5] This procedure can be viewed as a crude machine-learning strategy that selects a subset of predictors to be included in the final investment strategy out of the universe of available ones using the t-statistic of their past univariate performance. The out-of-sample performance of this investment strategy is impressive. The equal- and value-weighted portfolios generate an annualized Sharpe ratio of 1.60 and 1.17, respectively, which are significantly higher than those of our machine learning strategies (i.e., 1.02 and 0.30, respectively). The fact that feeding the universe of predictors to our machine-learning methods delivers a performance that is inferior to a simple recursive ranking strategy suggests that imposing an appropriate structure or "inductive bias" is important to the performance of machine-learning algorithms (Goyal and Bengio, 2022).

Fig. 1 succinctly summarizes our main results. We plot the Sharpe ratios for the following five investment strategies: The real-time machine-learning strategy based on our universe of fundamental signals (FS-ML);[6] the recursive ranking strategy based on the same universe of

[2] We highlight "feature engineering" here for simplicity of exposition. More broadly, our work highlights the importance of the inductive biases associated with implementing machine learning methods. As detailed in Goyal and Bengio (2022), inductive biases encompass preferences or constraints imposed on the hypothesis space to guide the learning and improve the generalization of machine learning methods. Feature engineering can be considered an inductive bias because it imposes a preference over the features to be considered, effectively shaping the learning process by narrowing the focus of machine learning methods to specific predictor variables.

[3] While humans determine the set of inputs to be fed into machine learning algorithms, the machine learning algorithms themselves often include variable selection mechanisms. This built-in variable selection identifies the most relevant features from the human provided inputs based on statistical significance or contribution to the model's accuracy. Throughout this paper, we use the term feature engineering to refer specifically to the human-driven process of selecting and designing the initial set of inputs, distinct from the algorithmic selection of variables during model training.

[4] Although publication bias may overstate the true expected returns of published anomalies, several prior studies have shown that the magnitude of the publication bias is relatively small (McLean and Pontiff, 2016; Chen and Zimmermann, 2020). Chen and Zimmermann (2020), for example, quantify the publication bias to be around 12%.

[5] We thank an anonymous referee for suggesting this analysis, which is motivated by an analysis in Table 3 of Yan and Zheng (2017).

[6] We refer to our machine-learning strategy as real-time machine-learning strategy primarily because our strategy can be implemented using only real-time information, i.e., our universe of signals is accessible to real-time investors. We acknowledge that to implement our machine-learning strategy,

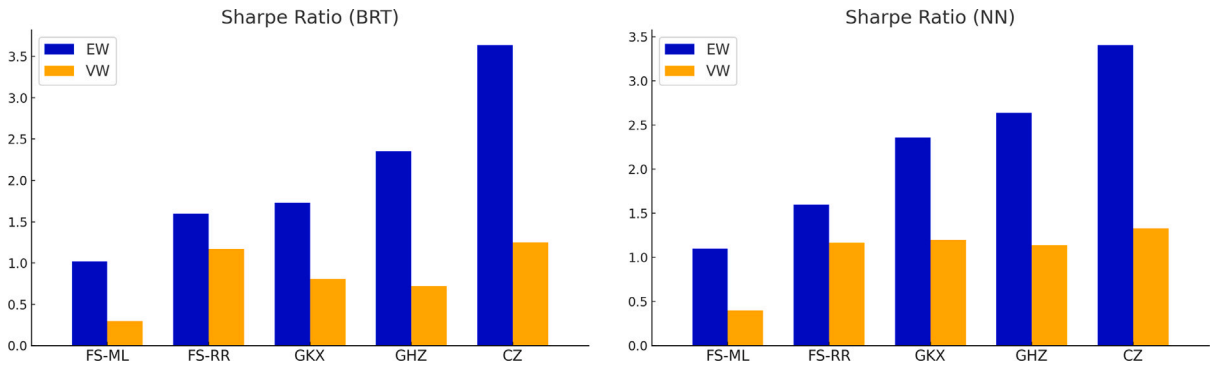

Fig. 1. Comparison of Sharpe ratios across strategies. This figure presents the Sharpe ratios for five investment strategies: **FS-ML**, a machine-learning strategy based on a universe of fundamental signals (1987–2019); **FS-RR**, a recursive-ranking strategy based on the same universe of fundamental signals (1987–2019); **GKX**, the baseline results from Gu et al. (2020) (1987–2016); **GHZ**, a machine-learning strategy based on the Green et al. (2017) signals (1987–2019); and **CZ**, a machine-learning strategy based on the Chen and Zimmermann (2022) signals (1987–2019). The left panel shows results using boosted regression trees, while the right panel presents those based on neural networks. For each approach, we report Sharpe ratios for both equally weighted (EW) and value-weighted (VW) portfolio returns.

fundamental signals (FS-RR); the Gu et al. (2020) strategy (GKX); the machine-learning strategy based on the selected set of signals in Green et al. (2017) (GHZ); and the machine-learning strategy based on the selected set of signals in Chen and Zimmermann (2022) (CZ). Three main takeaways are evident in Fig. 1. First, our real-time machine-learning strategies (FS-ML) deliver economically meaningful out-of-sample Sharpe ratios. Second, machine-learning strategies based on curated sets of signals (GKX, GHZ, and CZ) exhibit significantly higher Sharpe ratios than FS-ML. Third, FS-RR, which can also be implemented using only real-time information, performs significantly better than FS-ML. That is, a simple recursive ranking strategy based on the same universe of fundamental signals yields much higher Sharpe ratios than standard machine-learning strategies. Our main findings are robust across BRT and neural networks and hold for both equal- and value-weighted portfolios.[7] Overall, our results indicate that large economic gains are achievable for real-time investors, and that feature engineering and – more broadly – inductive biases are key to achieving such gains.

Our analyses so far have focused on fundamental signals. The main reason for this focus is that we can construct a "universe" of fundamental signals (Yan and Zheng, 2017). Past return-based signals are another class of predictors for which we can construct an "exhaustive" list of signals. In particular, we follow Martin and Nagel (2022) and use the past 120 months (excluding the most recent month) of stock returns. We also consider an alternative sample that includes the most recent month of stock return. As in our analysis of fundamental signals, we continue to use BRT as the primary machine-learning method. We find that the machine-learning strategy based on past-return signals earns an average return of 1.38% per month (t-statistic = 4.93) and exhibits an annualized Sharpe ratio of 1.04 in equal-weighted portfolios. The value-weighted portfolios deliver an average long–short return of 0.78% per month (t-statistic = 2.41) and a Sharpe ratio of 0.46. The results are stronger when we include the most recent month of stock returns. The strategy earns an average return of 1.81% per month (t-statistic = 6.40) and exhibits an annualized Sharpe ratio of 1.77 in equal-weighted portfolios while earning an average return of 0.98% per month (t-statistic = 3.14) and a Sharpe ratio of 0.66 in value-weighted portfolios.

These results are economically and statistically significant; however, they are weaker than those reported by prior studies that use curated sets of past-return-based predictors. For example, Moritz and Zimmermann (2016) show that machine learning strategies based on the past 24 monthly returns, which could have been plausibly selected by investors in real time, deliver a Sharpe ratio of 2.96 in equal-weighted portfolios. Similarly, Murray et al. (2024) show that machine learning strategies based on past 12 monthly cumulative returns, exhibit a Sharpe ratio of 0.78 in value-weighted portfolios. Comparing these performances with those of our past return strategies suggests, once again, that feature engineering can significantly improve the performance of machine-learning strategies.

As in Fig. 1 for fundamental signals, we summarize our main results for past-return signals in Fig. 2. We plot the Sharpe ratios of the following strategies: (1) Our machine-learning strategy based on 119 past monthly returns excluding the most recent month return (PR119); (2) the machine-learning strategy based on the past 120 monthly returns including the most recent month return (PR120); (3) the Moritz and Zimmermann (2016) strategy (MZ); and (4) the Murray et al. (2024) strategy (MXX). Because Moritz and Zimmermann (2016) report equal-weighted results, while Murray et al. (2024) report value-weighted results, we combine the results of these two strategies in Fig. 2.

The main takeaways from Fig. 2 are as follows. First, our baseline machine-learning strategy (PR119) delivers economically meaningful Sharpe ratios. Second, PR120, which includes the short-term reversal as an additional predictor, yields a higher Sharpe ratio than PR119. Third, strategies based on curated sets of inputs (i.e., MZ and MXX) perform better than both PR119 and PR120. Overall, our analyses based on past-return signals paint a similar picture to those based on fundamental signals. That is, the performance of our real-time machine-learning strategies is economically meaningful and statistically significant. More importantly, larger returns are available to real-time investors, and choosing a curated set of signals is key to realizing these larger returns for real-time investors.

We perform several robustness tests and additional analyses.[8] First, we repeat our analysis using a rolling-window approach instead of a recursive one. If the relations between fundamental signals and future

investors would also need considerable computing power and access to modern machine-learning algorithms. To the extent that these resources may not have been available to investors in the 1980s and 1990s, the performance of our machine-learning strategy shown in Fig. 1 could overestimate the economic gains to real-time investors.

[7] We include the performance of FS-RR in Fig. 1 for comparison, even though it is not a standard machine-learning strategy. Its performance is identical in the left panel (BRT) and the right panel (NN).

[8] Robustness is important in machine learning studies given the large "non-standard errors" found in the recent paper by Chen et al. (2024a).

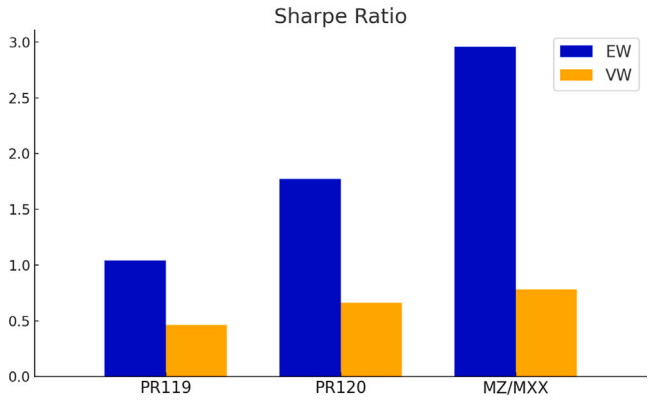

Fig. 2. Comparison of Sharpe ratios across ML strategies based on past-return signals. This figure presents the Sharpe ratios for three investment strategies: **PR119**, a BRT machine-learning strategy based on 119 past monthly returns—excluding the most recent month (1987–2019); **PR120**, a BRT machine-learning strategy based on 120 past monthly returns—including the most recent month (1987–2019); **MZ**, the Moritz and Zimmermann (2016) baseline results (1968–2012); and **MXX**, the Murray et al. (2024) baseline results (1963–2022). We report Sharpe ratios for both equally weighted (EW) and value-weighted (VW) portfolio returns.

stock returns are unstable over time, then the rolling-window approach should perform better. Contrary to this argument, our machine-learning strategies perform slightly worse under the rolling-window approach than under the recursive-window approach. Second, we repeat our analysis using alternative training and validation periods and find our results to be robust. Third, we repeat our analysis for subsamples of stocks sorted by firm size. We find that the out-of-sample performance of our machine-learning strategies is significantly stronger among small stocks than among large stocks. Fourth, we examine the after-trading-cost performance of our machine-learning strategies using Chen and Velikov (2022)'s low-frequency effective spreads as our trading cost measure. We find that the net returns to our machine-learning strategies based on fundamental signals are positive, while the net returns to strategies based on past-return signals are consistently negative. Finally, we explore the issue of time-varying predictability and find some evidence that the performance of our machine-learning strategies varies with the state of the market. However, there is little evidence that the profitability of our strategies varies systematically with investor sentiment, market volatility, market liquidity, or business cycle conditions.

Our paper builds on and contributes to the recent literature employing machine learning methods in empirical asset pricing. Gu et al. (2020) use machine learning methods to measure risk premium and show that machine learning models, particularly trees and neural networks, significantly outperform linear regression models in predicting stock returns. Freyberger et al. (2020) use the adaptive group LASSO for model selection and show that their model exhibits superior out-of-sample performance. Kozak et al. (2020) use shrinkage and selection methods to construct an SDF that summarizes the joint explanatory power of a large cross-section of return predictors. Chen et al. (2024c) estimate the SDF using deep neural networks and show that their model outperforms all other benchmark models.[9] These studies have established the potential for large economic gains to investors using machine learning strategies. We complement the existing studies by taking the perspective of real-time investors. Specifically, we show that using machine learning methods is beneficial for real-time investors and that feature engineering is key to significantly enhancing such benefits.

Our paper is also related to a growing literature examining the performance of data-mined signals (Yan and Zheng, 2017; Chordia et al., 2020; Harvey and Liu, 2020; Zhu, 2023; Chen and Dim, 2024; Chen, 2024; Chen et al., 2024b). In particular, Chen and Dim (2024) show that systematic data-mining leads to significant out-of-sample performance and argue that "high-throughput methods provide a rigorous, unbiased method for documenting asset pricing facts". Chen et al. (2024b) show that mining a universe of accounting ratios yields out-of-sample performance comparable to that of published signals.

Finally, our paper is related to Arnott et al. (2019) and Israel et al. (2020), who caution that machine learning methods may not work as well in finance as in some other disciplines. In particular, machine learning methods face three significant challenges in finance applications: the lack of data (on the time series dimension), the low signal-to-noise ratio, and the adaptive nature of financial markets. While the modest performance of our real-time machine-learning strategies could be a manifestation of these challenges faced by market professionals and investors, we argue that feature engineering holds considerable promise in significantly improving the performance of machine-learning-based investment strategies.

The rest of our paper proceeds as follows. Section 2 describes our data, sample, and methods. Section 3 presents our main empirical results. Section 4 presents the results for additional analyses and robustness tests. Section 5 concludes.

2. Data, sample, and methods

This section describes the stock sample and the fundamental signals we employ in our main analysis. We then describe the cross-sectional prediction problem underlying the portfolio strategies we generate and the main empirical method we use—boosted regression trees (BRT). Finally, we describe how we implement our machine-learning strategy.

2.1. Stock sample and associated fundamental signals

We obtain monthly stock returns, share price, SIC code, and shares outstanding from the Center for Research in Security Prices (CRSP) and annual accounting data from Compustat. Our sample consists of the NYSE, AMEX, and NASDAQ common stocks (with a CRSP share code of 10 or 11) with the necessary data to construct fundamental signals and compute subsequent stock returns. We exclude financial stocks, i.e., those with a one-digit SIC code of 6. We also remove stocks with

[9] For additional studies that use machine learning methods in asset pricing, please also see, e.g., Rapach et al. (2013), Chinco et al. (2019), Feng et al. (2020), Bryzgalova et al. (2020), Bianchi et al. (2021), Dong et al. (2022), Leippold et al. (2022), Avramov et al. (2022), Kelly and Xiu (2023), Geertsema and Lu (2023), Kaniel et al. (2023), Bali et al. (2023) and Chen and McCoy (2024). Several earlier studies (Ou and Penman, 1989; Holthausen and Larcker, 1992; Haugen and Baker, 1996) use machine learning-like methods to predict future stock returns.

a share price lower than $1. To mitigate backfilling biases, we require that a firm be listed on Compustat for two years before it is included in our sample (Fama and French, 1993). We obtain Fama and French (1996, 2015) factors and the momentum factor from Kenneth French's website and Hou et al. (2015) q-factors from Lu Zhang's website.[10] Our sample spans from July 1963 to June 2019, and our sample consists of 15,035 stocks.

We construct the universe of fundamental signals for our sample of stocks following Yan and Zheng (2017).[11] We start with 240 accounting variables (listed in Table B.1) and compute, for each variable, a total of 76 signals (listed in Table C.1). These signals are obtained by taking the original accounting variables and transforming them by computing changes, ratios, and other potentially economically meaningful transformations. The final number of fundamental signals we include in our analysis is 18,113, which is slightly smaller than 18,240 (240 × 76) because not all combinations of the accounting variables result in meaningful signals, and some of the combinations are redundant. For brevity, we refer the readers to Yan and Zheng (2017) for complete details regarding selecting accounting variables and constructing fundamental signals.

2.2. Methodology

2.2.1. Prediction equation

We predict the cross-section of stock returns using the following specification:

$$R_{i,t+1} = f\left(\mathbf{x}_{i,t}|\theta\right) + \epsilon_{i,t+1} \tag{1}$$

where $R_{i,t+1}$ denotes annual excess return for stock i from July of year t to June of year $t+1$, $\mathbf{x}_{i,t}$ denotes a vector of variables used to predict the cross section of returns, and θ denotes the parameters for the prediction function f. Stocks are indexed as $i = 1, \ldots, N$ and years are indexed by $t = 1, \ldots, T$.

The vector of predictive variables includes the 18,113 fundamental signals described earlier. To make sure the accounting information is publicly available to investors, we follow Fama and French (1992) and pair accounting variables in year $t-1$ with stock returns from July of year t to June of year $t+1$. We follow Gu et al. (2020) and transform all fundamental signals as follows. We first rank all non-missing fundamental signals each year and then scale their ranks to the interval $[-1, +1]$. By construction, the cross-sectional median of the transformed fundamental signals is zero.

We predict annual excess returns for two reasons. First, our fundamental signals are constructed from annual financial statements and are updated annually. Second, the number of signals considered in our study is substantially larger than those in prior studies. Predicting annual returns is computationally more efficient than predicting monthly returns.[12]

2.2.2. Machine learning methods vs. linear regressions

Traditionally, it was common in the literature to assume linearity of the f function and estimate Equation (1) using linear regression (LR) methods. More recently, the finance literature has instead started adopting more advanced Machine Learning (ML) methods.

One may expect that ML methods should have an advantage compared to linear regression methods because they feature (1) variable selection, (2) model combination, and (3) regularization/shrinkage, which allow them to handle large sets of conditioning information and stabilize their predictions by making them less sensitive to outliers.

ML methods also allow to capture nonlinearities in the relations between the target variable and the regressors. When viewed through the lenses of the bias–variance trade-off, including nonlinearities allows for a smaller bias at the cost of a higher variance which positively relates to the instability of the predictions. In fact, a growing field in computer science, referred to as "adversarial machine learning", shows that even very small perturbations of the predictor variables can result in large changes in ML predictions.[13]

Similar effects could arise naturally in finance, where the data-generating process relating regressands and regressors constantly evolves. As profitable strategies are arbitraged away by smart money in a Schumpeterian creative destruction cycle, ML methods could potentially overfit certain temporary patterns that exist only in certain periods. This is particularly true for ML models with thousands (millions or even billions) of parameters that have been trained to capture deep, non-linear interactions because such a process makes them less adaptable to changes in the underlying dynamics of the data. These issues are further complicated by the fact that financial datasets are relatively small compared to those used in other fields, and financial research often faces weak signal-to-noise ratios (Kelly and Xiu, 2023). In these contexts, simpler models, like linear regression, could be more robust to changes in the data-generating process and deliver a more robust performance out-of-sample.

An important question is whether we should expect the advantages and disadvantages of ML models compared to LR models to vary depending on whether the researchers use a "universe" versus a "selected set" of predictors in their analysis. The theoretical literature does not provide a definitive answer to this question. Intuitively, on the one hand, we can expect ML methods to have a greater advantage compared to LR methods in the "universe" predictor setting than in the "selected" predictor setting because they feature regularization and variable selection. On the other hand, ML may have a smaller advantage relative to LR when deployed on a "universe" of predictors because nonlinearities and variable interactions may be less important in higher-dimensional settings, and ML methods may be less robust to time variations in the relation between regressand and regressors. We leave an in-depth analysis of these theoretical and empirical issues to further research.

2.2.3. Boosted regression trees

Our baseline specification includes 18,113 fundamental signals. We choose the "off-the-shelf" machine learning tool called Boosted Regression Trees (BRT), in particular, the LightGBM implementation (Ke et al., 2017) for our baseline analysis.

We choose BRT as our primary machine learning method for several reasons. First, BRT routinely rank among the very best machine learning algorithms in both finance and non-finance applications.[14] Second, BRT can handle large data sets with high dimensionality without overfitting because they simultaneously perform subsampling, model

[10] Kenneth French's data library is located at https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html. The q-factors can be downloaded from http://global-q.org/index.html.

[11] To minimize our discretion, we use a pre-existing universe of fundamental signals instead of constructing one specifically for this study. Chordia et al. (2020) extend (Yan and Zheng, 2017) and construct a universe of over 2 million fundamental signals. We choose not to use this universe because real-time investors are unlikely to have the computing power to evaluate these many predictive variables in a machine-learning context.

[12] We conduct our empirical analyses on a high-performance cluster of 45 computing nodes, each of which is equipped with 128 GB, 384 GB, or 4TB of RAM. For neural networks, we have to use nodes with 4TB of RAM.

[13] See https://en.wikipedia.org/wiki/Adversarial_machine_learning for an introduction to the topic and additional details.

[14] See a list of Machine Learning Challenge Winning Solutions on the LightGBM's website at https://github.com/microsoft/LightGBM/tree/master/examples.

combination, and shrinkage. Third, BRT are robust to missing values and outliers (Hastie et al., 2009). In particular, BRT are invariant under all monotone transformations of the individual input variables, making the forecasts generated robust to extreme values. Finally, because BRT are rooted in the CART framework, they possess good interpretability. For example, BRT return the rank and relative importance of all the potential regressors available, known as relative influence measures.[15] This feature distinguishes BRT from harder-to-interpret methods such as neural networks.

Regression Trees

A regression tree is built through a process known as binary recursive partitioning, which is an iterative process that splits the data into partitions or branches. Suppose we have P potential predictor ("state") variables and a single dependent variable over T observations, i.e., $\left(x_t, y_{t+1}\right)$ for $t = 1, 2, \ldots, T$, with $x_t = \left(x_{t1}, x_{t2}, \ldots, x_{tp}\right)$. Fitting a regression tree requires deciding (i) which predictor variables to use to split the sample space and (ii) which split points to use. The regression trees we use employ recursive binary partitions, so the fit of a regression tree can be written as an additive model:

$$f(x) = \sum_{j=1}^{J} c_j I\left\{x \in S_j\right\},$$

where $S_j, j = 1, \ldots, J$ are the regions we split the space spanned by the predictor variables into, $I\{\cdot\}$ is an indicator variable, and c_j is the constant used to model the dependent variable in each region. If the L^2 norm criterion function is adopted, the optimal constant is $\hat{c}_j = mean\left(y_{t+1} | x_t \in S_j\right)$.

The globally optimal splitting point is difficult to determine, particularly in cases where the number of state variables is large. Hence, we use a sequential greedy algorithm. Using the full set of data, the algorithm considers a splitting variable p and a split point s so as to construct half-planes,

$$S_1(p, s) = \left\{X | X_p \leq s\right\} \quad and \quad S_2(p, s) = \left\{X | X_p > s\right\},$$

that minimize the sum of squared residuals:

$$\min_{p,s}\left[\min_{c_1} \sum_{x_t \in S_1(p,s)} \left(y_{t+1} - c_1\right)^2 + \min_{c_2} \sum_{x_t \in S_2(p,s)} \left(y_{t+1} - c_2\right)^2\right].$$

For a given choice of p and s, the fitted values, $\hat{c}_1$ and $\hat{c}_2$, are

$$\hat{c}_1 = \frac{1}{\sum_{t=1}^{T} I\left\{x_t \in S_1(p, s)\right\}} \sum_{t=1}^{T} y_{t+1} I\left\{x_t \in S_1(p, s)\right\},$$

$$\hat{c}_2 = \frac{1}{\sum_{t=1}^{T} I\left\{x_t \in S_2(p, s)\right\}} \sum_{t=1}^{T} y_{t+1} I\left\{x_t \in S_2(p, s)\right\}.$$

The best splitting pair (p, s) in the first iteration can be determined by searching through each of the predictor variables, $p = 1, \ldots, P$. Given the best partition from the first step, the data is then partitioned into two additional states, and the splitting process is repeated for each of the subsequent partitions. Predictor variables that are never used to split the sample space do not influence the fit of the model, so the choice of splitting variable effectively performs variable selection.

Regression trees are ideally suited for handling high-dimensional data sets, incorporating multiway interactions among predictors, and capturing non-linear relations between predictors and the predicted variable. However, the approach is sequential, and successive splits are performed on fewer and fewer observations, increasing the risk of fitting idiosyncratic data patterns. Furthermore, there is no guarantee that the sequential splitting algorithm leads to the globally optimal solution. To deal with these problems, we next consider a regularization method known as boosting.

[15] To conserve space, we provide a description of the relative influence measures in Appendix D. We also implement the relative influence measure on our data and report the results in Appendix D.

Boosting

Boosting is based on the idea that combining a series of simple prediction models can lead to more accurate forecasts than those available from any individual model. Boosting algorithms iteratively re-weight data used in the initial fit by adding new trees in a way that increases the weight on observations modeled poorly by the existing collection of trees. From above, recall that a regression tree can be written as:

$$\mathcal{T}\left(x; \left\{S_j, c_j\right\}_{j=1}^{J}\right) = \sum_{j=1}^{J} c_j I\left\{x \in S_j\right\}.$$

A boosted regression tree is simply the sum of regression trees:

$$f_B(x) = \sum_{b=1}^{B} \mathcal{T}_b\left(x; \left\{S_{b,j}, c_{b,j}\right\}_{j=1}^{J}\right),$$

where $\mathcal{T}_b\left(x; \left\{S_{b,j}, c_{b,j}\right\}_{j=1}^{J}\right)$ is the regression tree used in the bth boosting iteration and B is the number of boosting iterations. Given the model fitted up to the $(b-1)$-th boosting iteration, $f_{b-1}(x)$, the subsequent boosting iteration seeks to find parameters $\left\{S_{j,b}, c_{j,b}\right\}_{j=1}^{J}$ for the next tree to solve a problem of the form

$$\left\{\hat{S}_{j,b}, \hat{c}_{j,b}\right\}_{j=1}^{J} = \min_{\left\{S_{j,b}, c_{j,b}\right\}_{j=1}^{J}} \sum_{t=0}^{T-1}\left[y_{t+1} - \left(f_{b-1}\left(x_t\right) + \mathcal{T}_b\left(x_t; \left\{S_{j,b}, c_{j,b}\right\}_{j=1}^{J}\right)\right)\right]^2.$$

For a given set of state definitions ("splits"), $S_{j,b}$, $j = 1, \ldots, J$, the optimal constants, $c_{j,b}$, in each state are derived iteratively from the solution to the problem

$$\hat{c}_{j,b} = \min_{c_{j,b}} \sum_{x_t \in S_{j,b}} \left[y_{t+1} - \left(f_{b-1}\left(x_t\right) + c_{j,b}\right)\right]^2 = \min_{c_{j,b}} \sum_{x_t \in S_{j,b}} \left[e_{t+1,b-1} - c_{j,b}\right]^2,$$

where $e_{t+1,b-1} = y_{t+1} - f_{b-1}\left(x_t\right)$ is the empirical error after $b-1$ boosting iterations. The solution to this problem is the regression tree that most reduces the average of the squared residuals $\sum_{t=1}^{T} e_{t+1,b-1}^2$, and $\hat{c}_{j,b}$ is the mean of the residuals in the jth state.

Forecasts are simple to generate from this approach. The boosted regression tree is first estimated using data from $t = 1, \ldots, t^*$. Then, the forecast of y_{t^*+1} is based on the model estimates and the value of the predictor variable at time t^*, x_{t^*}. Boosting makes it more attractive to employ small trees (characterized by few terminal nodes) at each boosting iteration, reducing the risk that the regression trees will overfit. Moreover, by summing over a sequence of trees, boosting performs a type of model averaging that increases the stability and accuracy of the forecasts.

2.3. Implementation

We implement our BRT model by following Gu et al. (2020). We divide our sample period (1963–2019) into 12 years of training sample (1963–1974), 12 years of validation sample (1975–1986), and the remaining 33 years (1987–2019) for out-of-sample testing. We begin the out-of-sample period in 1987 in order to align with Gu et al. (2020).

We refit our model every year because our fundamental signals are updated annually. Each time we refit the model, we increase the training sample by one year while maintaining the length of the validation period at 12 years. This recursive window approach allows for the incorporation of all available information in generating forecasts. Every year, we generate return forecasts for all the stocks in our sample. We then construct decile portfolios based on the predicted returns. We hold these portfolios for 12 months and rebalance them every year. Our long–short strategy goes long in the decile portfolio with the highest BRT expected returns and short in the decile portfolio with the lowest BRT predicted returns.

To generate return forecasts, we need to estimate the model's parameters using the training data and specify two key hyper-parameters, i.e., the number of boosting iterations and the BRT shrinkage parameter (also known as the learning rate). To choose these two hyper-parameters, we adopt the commonly used grid search with validation procedure (Hastie et al., 2009; Gu et al., 2020).[16] We leave all other tuning parameters at their LightGBM default values.

Specifically, we first use the training sample to estimate the model under each set of hyper-parameter values. We then use the hyper-parameters that show the best performance during the validation period to re-estimate the final model. For example, suppose we want to forecast the cross-section of stock returns for 1987. We fit models under different hyper-parameter values during the training period 1963–1974 and then use the validation period 1975–1986 to gauge the performance of these trained models. We choose the hyper-parameters that deliver the best performance during the validation period and then use these hyper-parameters to re-estimate the final model for the combined training and validation period 1963–1986. When we move forward and forecast the cross-section of stock returns for 1988, our validation period rolls forward by one year and stays at 12 years, i.e.,1976–1987, while our training period increases by one year and goes from 1963 to 1975 (13 years).[17]

Our fundamental signals contain missing values. Although BRT can handle missing values, we pre-process the missing values to make BRT forecasts comparable to other machine learning methods that cannot handle missing values. Specifically, we follow the approach of Gu et al. (2020) and replace missing values with the cross-sectional median.[18] Recall that we have normalized all non-missing fundamental signals to the $[-1, +1]$ interval by using their cross-sectional ranks. By construction, the cross-sectional median of the transformed signals is zero. We, therefore, assign all missing values as zero.[19]

Performance Evaluation

Each year, we sort all sample stocks into deciles based on BRT predicted returns, construct equal- and value-weighted portfolios, and focus on the long–short strategy that buys stocks in the top decile and shorts stocks in the bottom decile. We estimate CAPM 1-factor, Fama–French 3-factor, Carhart 4-factor, Fama–French 5-factor, Fama–French 5-factor + Momentum factor, and q-factor models by running the following time-series regressions:

$$r_t = \alpha + \beta MKT_t + \epsilon_t$$

$$r_t = \alpha + \beta MKT_t + sSMB_t + hHML_t + \epsilon_t$$

$$r_t = \alpha + \beta MKT_t + sSMB_t + hHML_t + uUMD_t + \epsilon_t$$

$$r_t = \alpha + \beta MKT_t + sSMB_t + hHML_t + rRMW_t + cCMA_t + \epsilon_t$$

$$r_t = \alpha + \beta MKT_t + sSMB_t + hHML_t + rRMW_t + cCMA_t + uUMD_t + \epsilon_t$$

$$r_t = \alpha + \beta MKT_t + sSMB_t + rROE_t + iIA_t + \epsilon_t$$

where r_t is the long–short portfolio return based on BRT-generated forecasts for month t, and MKT, SMB, HML, UMD, RMW, CMA, ROE, and IA are market, size, value, momentum, profitability, investment (FF5), return on equity, and investment (Q) factors (Carhart, 1997; Fama and French, 2015; Hou et al., 2015). We focus on the alpha estimates and their t-statistics estimated using Newey and West (1987) standard errors.

[16] Our grid for the number of boosting iterations is $\{100, 250, 500, 750, 1000\}$, while our grid for the learning rate is $\{0.01, 0.05, 0.10\}$.

[17] We show in Section 4.2 that our main results are robust to alternative training and validation periods.

[18] Chen and McCoy (2024) provide a rigorous justification for the use of mean/median imputation in machine learning studies. Specifically, they show that mean/median and sophisticated imputation methods lead to similar results.

[19] The performance of the BRT portfolios is similar without pre-processing the missing values.

3. Main results

In this section, we report the main results of our paper. We start by reporting in Section 3.1 the baseline results that compute the out-of-sample realized returns for BRT portfolios using our predictor universe. We then report in Section 3.2 the abnormal performance of the BRT portfolios that control for various risk factors. Section 3.3 uses an alternative machine learning method, i.e., neural networks. Section 3.4 examines whether our ML implementation can generate high long–short returns and Sharpe ratios using selected sets of predictors. Section 3.5 examines the performance of a simple recursive ranking strategy applied to the universe of predictors. Finally, Section 3.6 examines the machine learning performance based on a universe of past-return signals.

3.1. Baseline results

Table 1 shows the results of our baseline analysis. As stated earlier, we sort stocks into deciles each year based on one-year-ahead BRT predicted returns constructed using our universe of fundamental signals. We then construct a long–short portfolio that buys stocks with the highest BRT predicted returns and sells stocks with the lowest BRT predicted returns. We track the performance of these portfolios for 12 months. Following Gu et al. (2020), we report in Table 1 the BRT predicted returns (i.e., the sorting variable), the average realized returns, the standard deviation of realized returns, and the annualized Sharpe ratios of BRT-sorted portfolios.

The left panel of Table 1 focuses on equally weighted portfolios. The first column shows the BRT predicted return, which is by construction monotonically increasing from decile 1 (−0.04% per month) to decile 10 (1.69% per month). The second column reports the out-of-sample average realized return for each portfolio: our primary variable of interest. We find that the performance of BRT portfolios increases nearly monotonically from decile 1 (−0.01%) to decile 10 (0.94%). The long–short portfolio earns an average return of 0.95% per month (or 11.4% per year), with a highly significant t-statistic of 6.63.[20]

The standard deviation of the realized returns is U-shaped across the BRT decile portfolios, i.e., the portfolios with the lowest and the highest BRT predicted returns have higher volatilities than the other portfolios. Not surprisingly, we find that the long–short portfolio has a much lower volatility than the long-only portfolios. Finally, the last column of the left panel reports the annualized Sharpe ratio, which ranges from −0.01 to 0.62 across the ten BRT decile portfolios. The Sharpe ratio of the long–short portfolio is much higher at 1.02, which is primarily driven by the lower volatility of the long–short portfolio.

Equally weighted portfolios tend to overweight small-cap stocks that can be harder and more expensive to trade (e.g., Fama and French, 2008; Novy-Marx and Velikov, 2016). To mitigate this issue, we examine in the right panel of Table 1 the value-weighted portfolio returns. The BRT predicted return is again by construction monotonically increasing from decile 1 (0.00%) to decile 10 (1.61%). More importantly, the realized average portfolio return also increases from decile 1 (0.40%) to decile 10 (0.80%), although the relation is not monotonic. The spread between decile 10 and decile 1 is 0.40% per month, or 4.8% per year.[21] Even though this spread is less than half of the spread for equally weighted portfolios, it is nevertheless economically meaningful and statistically significant at the 5% level.

[20] Appendix D reports the top 25 fundamental signals based on an analysis of variable importance. We find that signals constructed using excise tax and minority interest are among the most important predictors.

[21] These returns are before trading costs. We report the before-trading cost performance of our machine learning strategies for ease of comparison with prior literature (e.g., Chen et al., 2024c; Freyberger et al., 2020; Gu et al., 2020). In Section 4.4, we examine the after-trading-cost performance of our machine-learning strategies.

Table 1
Performance of portfolios sorted by BRT predicted returns. This table reports the excess returns of decile portfolios sorted by BRT predicted returns from 1987 to 2019. We predict stock annual excess returns using 18,113 fundamental signals (as described in Section 2.1). We use a recursive window approach and select the optimal hyper-parameters using a cross-validation approach. Our initial estimation period is 1963–1986. The first 12 years is the training period and the second 12 years is the validation period. As we roll forward, the training period expands while the validation period stays at 12 years. The left panel reports equal-weighted portfolio results. In the first column of this panel we report the average predicted monthly returns from the BRT model (*Pred*). The second and third columns report the average realized monthly excess returns (*Avg*) and associated *t*-statistics (*t-stat*), computed using Newey and West (1987) standard errors with 12 lags. Finally, in the fourth and fifth column we report the portfolios' return standard deviations (*SD*) and annualized Sharpe ratios (*SR*), respectively. The right panel reports the same results using value-weighted portfolio returns. All returns are expressed in percent per month.

Rank	Equal weight					Value weight				
	Pred	*Avg*	*t-stat*	*SD*	*SR*	*Pred*	*Avg*	*t-stat*	*SD*	*SR*
1 (Low)	−0.04	−0.01	−0.05	7.51	−0.01	0.00	0.40	1.30	6.18	0.22
2	0.30	0.49	1.58	6.22	0.28	0.30	0.58	2.39	5.53	0.36
3	0.49	0.65	2.12	6.02	0.37	0.50	0.58	2.63	4.73	0.42
4	0.64	0.74	2.65	5.64	0.46	0.64	0.75	3.21	4.60	0.56
5	0.73	0.76	2.72	5.45	0.48	0.73	0.60	2.34	4.61	0.45
6	0.80	0.90	3.23	5.44	0.58	0.80	0.66	2.99	4.51	0.51
7	0.88	0.90	3.18	5.57	0.56	0.88	0.68	2.68	4.93	0.48
8	0.97	0.96	3.17	5.43	0.62	0.97	0.49	1.81	4.82	0.35
9	1.12	0.93	2.84	5.78	0.56	1.11	0.64	2.20	5.15	0.43
10 (High)	1.69	0.94	2.55	6.71	0.48	1.61	0.80	2.51	5.96	0.47
10–1	1.74	0.95	6.63	3.26	1.02	1.61	0.40	2.34	4.68	0.30

The Sharpe ratio exhibits a similar pattern, higher for decile 10 (0.47) than for decile 1 (0.22). The Sharpe ratio for the long–short portfolio is 0.30.

Overall, we show in Table 1 that long–short portfolios formed based on BRT forecasts earn economically and statistically significant returns. The magnitude of the long–short performance, however, is much lower than that documented in the prior literature. For example, the BRT models in Gu et al. (2020) achieve an equally weighted monthly long–short portfolio return of 2.14% per month and a Sharpe ratio of 1.73. The corresponding numbers for value-weighted portfolios are 0.99% per month and a Sharpe ratio of 0.81.[22] The long–short portfolios formed based on neural network forecasts perform even better in Gu et al. (2020), earning an average return of 3.33% per month and an annualized Sharpe ratio of 2.45 in equal-weighted portfolios and an average return of 2.26% per month and a Sharpe ratio of 1.35 in value-weighted portfolios.[23] Similarly, Chen et al. (2024c) report an out-of-sample Sharpe ratio of 2.60, and Freyberger et al. (2020) report that their model delivers an out-of-sample Sharpe ratio of 2.75.

The main difference between our paper and prior studies is that we employ a universe of fundamental signals and do not feed our machine-learning methods a curated set of predictors. Hence, these results provide initial evidence that feature engineering – as a form of inductive bias involving the selection and transformation of predictors in machine-learning applications – may play a key role in determining the economic gains achievable by real-time investors.

3.2. Controlling for common risk factors

The results in Table 1 do not control for risk exposures. It could be that the long–short portfolios based on BRT forecasts have positive and significant returns because they are exposed to well-known sources of risk, such as value or profitability. Table 2 shows the risk-adjusted performance of our BRT portfolios once we control for risk exposures using the six models described in Section 2.3. Irrespective of whether we use the CAPM model (columns 1–2), the Fama–French 3-factor model (columns 3–4), the Carhart 4-factor model (columns 5–6), the Fama–French 5-factor model (columns 7–8), the Fama–French 5-factor model augmented with momentum (columns 9–10) or the *q*-factor model (columns 11–12), we find that portfolios with higher BRT predicted returns have higher average realized risk-adjusted returns. Taking the Carhart 4-factor model as an example, we find that the alpha of decile 1 is negative and significant at −0.71% per month (*t*-statistic = −4.63), while the alpha of decile 10 is 0.37% per month (*t*-statistic = 2.66). The resulting long–short portfolio has a monthly alpha of 1.08% and is statistically significant with a *t*-statistic of 6.43.

The results for value-weighted risk-adjusted returns are weaker than the equal-weighted results—in line with the findings in Table 1. Across the various risk-adjustment models, the monthly abnormal performance ranges from a minimum of 0.46% (5.52% annualized) for the CAPM to a maximum of 0.80% (9.60% annualized) for the Fama–French 5-factor model with momentum. In all cases, the alphas of the long–short portfolios are statistically different from zero.

Consistent with the findings reported in Section 3.1, our results suggest that machine learning tools indeed can help predict stock returns. After adjusting for standard asset pricing factors, the long–short returns are economically meaningful and statistically significant. Still, the degree of predictability is significantly lower than what has been reported in the literature that uses selected signals as return predictors.

3.3. Neural networks

In our baseline analysis, we use BRT, which is one of the most powerful machine learning methods for stock return predictions. Nevertheless, one might be concerned that our main results are specific to BRT and may not extend to other machine-learning methods. To ensure this is not the case, we extend our analysis to neural networks (NNs) mainly because – together with boosted regression trees – NNs are among the top performers when it comes to return prediction (Gu et al., 2020; Bianchi et al., 2021). We follow Gu et al. (2020) and conduct our analysis using NNs with 1 to 5 hidden layers. Appendix A describes our NNs implementation in detail.

Our results, reported in Table 3, reveal several important findings. First, the equal-weighted long–short returns based on NNs are highly significant, while the value-weighted long–short returns are generally (but not always) significant. Second, among both equal- and value-weighted portfolios, we find that shallow NNs perform better than deep NNs. For example, NNs with 1 hidden layer achieve long–short

[22] We note that we implement our BRT model using LightGBM, while Gu et al. (2020) implement using scikit-learn. When we implement our model using scikit-learn in conjunction with our fundamental signals, we obtain even less significant results than what we currently report in the paper.

[23] We implement our strategies using neural networks in Section 3.3.

Table 2
Risk-adjusted performance of portfolios sorted by BRT predicted returns. This table shows the risk-adjusted performance of the BRT portfolios based on the CAPM model, the Fama–French 3-factor model, the Carhart 4-factor model, the Fama–French 5-factor model, the Fama–French 5-factor model augmented with momentum factor, and the q-factor model. The BRT model specifications are the same as that in Table 1. The top panel reports results for equal-weighted portfolios, and the bottom panel reports results for value-weighted portfolios. All returns are expressed in percent per month.

Equal Weight												
Rank	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	alpha	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
L(ow)	−0.83	−3.71	−0.75	−4.89	−0.71	−4.63	−0.44	−3.39	−0.43	−3.30	−0.29	−1.76
2	−0.21	−0.98	−0.19	−1.47	−0.17	−1.45	−0.09	−0.72	−0.08	−0.70	0.03	0.19
3	−0.06	−0.32	−0.06	−0.60	−0.05	−0.59	−0.03	−0.32	−0.03	−0.30	0.07	0.73
4	0.05	0.30	0.04	0.51	0.03	0.49	0.02	0.24	0.02	0.25	0.09	1.30
5	0.11	0.60	0.08	1.08	0.08	1.12	0.03	0.38	0.03	0.44	0.09	1.21
6	0.24	1.22	0.18	2.37	0.23	2.80	0.11	1.44	0.15	1.91	0.19	2.04
7	0.23	1.27	0.21	3.08	0.20	2.77	0.22	2.81	0.21	2.67	0.26	3.76
8	0.30	1.61	0.28	3.21	0.29	3.37	0.29	3.26	0.30	3.37	0.35	3.85
9	0.22	1.34	0.22	2.25	0.29	3.10	0.35	3.26	0.39	3.75	0.45	4.88
H(igh)	0.18	0.85	0.25	1.62	0.37	2.66	0.59	3.88	0.65	4.44	0.69	4.68
H-L	1.01	6.29	1.01	6.34	1.08	6.43	1.03	5.40	1.08	5.58	0.98	5.10
Value Weight												
Rank	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	alpha	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
L(ow)	−0.36	−2.40	−0.26	−2.29	−0.31	−2.74	−0.13	−1.12	−0.18	−1.57	−0.09	−0.59
2	−0.11	−0.76	−0.03	−0.19	−0.04	−0.28	0.01	0.08	0.00	0.00	0.08	0.54
3	−0.05	−0.53	−0.02	−0.23	−0.03	−0.33	−0.08	−0.82	−0.08	−0.84	−0.02	−0.15
4	0.12	1.61	0.15	2.00	0.10	1.27	0.05	0.68	0.02	0.30	0.08	0.88
5	−0.01	−0.14	−0.01	−0.10	0.01	0.11	−0.12	−1.47	−0.10	−1.37	−0.07	−0.98
6	0.08	0.80	0.04	0.47	0.06	0.71	−0.17	−2.03	−0.13	−1.51	−0.12	−1.69
7	0.04	0.41	0.00	0.03	0.03	0.35	−0.13	−1.37	−0.10	−1.01	−0.11	−1.13
8	−0.13	−1.10	−0.13	−1.02	−0.07	−0.60	−0.17	−1.26	−0.12	−0.99	−0.09	−0.61
9	−0.02	−0.10	0.06	0.43	0.11	1.00	0.14	1.25	0.17	1.60	0.25	1.92
H(igh)	0.10	0.46	0.20	1.23	0.34	2.37	0.54	3.80	0.62	4.24	0.58	4.03
H-L	0.46	2.17	0.46	2.57	0.65	3.66	0.67	3.18	0.80	3.79	0.68	2.94

returns of 1.08% (t-statistic = 6.09) for equal-weighted portfolios and 0.74% per month (t-statistic = 4.58) for value-weighted portfolios. The corresponding long–short returns for NNs with 5 hidden layers are much lower at 0.80% (t-statistic = 3.79) and 0.21% per month (t-statistic = 1.00), respectively. This finding is consistent with Gu et al. (2020), who show that shallow learning performs better than deep learning. Third, the performance of long–short portfolios based on neural network forecasts is much weaker than those documented by prior machine learning studies. Gu et al. (2020), for example, show that the long–short portfolios formed based on neural network forecasts earn an average return of 3.33% per month in equal-weighted portfolios and an average return of 2.26% in value-weighted portfolios. Overall, similar to BRT, our results based on neural networks suggest that the real-time performance of machine learning strategies based on a universe of predictors is more modest than that obtained by using a selected set of predictors, highlighting, once again, the importance of the choice of input variables when implementing machine learning investment strategies.

3.4. ML implementation

One might be concerned that the relatively weak performance of our machine learning strategies is perhaps due to our ML implementation not being as powerful as those employed in previous studies. To evaluate this possibility, we replicate our ML results – both boosted regression trees (BRT) and neural networks (NN) – on samples of published anomalies.

For ease of comparison with GKX, we use the sample of 94 anomalies listed in Green et al. (2017, GHZ).[24] We downloaded the SAS code that generates the 94 predictors from Jeremiah Green's website at https://sites.google.com/site/jeremiahrgreenacctg/home. The second sample comprises the Chen and Zimmermann (2022, CZ) predictors. We downloaded the data from https://www.openassetpricing.com/ and used the March 2022 data release, which includes 207 anomaly predictors.[25] The out-of-sample testing period for this analysis is 1987–2019, the same as that for our main analyses based on fundamental signals and past-return signals.[26]

In Table 4, we report the results based on the GHZ sample of anomalies. We find that both BRT and neural networks (NN1 through NN5) deliver an out-of-sample long–short return in excess of 3.5% per month for equally weighted portfolios and over 1.5% per month for value-weighted portfolios. We also find that NNs outperform BRT, in line with the results in Gu et al. (2020). We obtain similar findings when we focus on risk-adjusted returns, as shown in the remaining columns of Table 4. BRT and neural networks generate a Sharpe ratio between 2.21 and 2.81 in equal-weighted portfolios, demonstrating that our ML implementation could generate similar Sharpe ratios to those in prior literature (e.g., Gu et al., 2020) when using published predictors.[27]

[24] GKX construct their data set based on GHZ's 94 characteristics. See footnote 30 in Gu et al. (2020) for more details.

[25] The definitions of these variables are available at https://www.openassetpricing.com/march-2022-data-release/.

[26] We also consider three alternative out-of-sample testing periods, namely 1987–2016, 1991–2004, and 1991–2014 in Tables IA.8 and IA.9. The performance of machine-learning strategies during these alternative sample periods is qualitatively similar to and quantitatively stronger than that for 1987–2019.

[27] Note that, while rather similar, our results for BRT and NN1-NN5 in Table 4 do not replicate exactly those in Table 7 and Table A.9 of GKX. Three implementation differences explain the results. First, the GKX predictions are generated by interacting the original 94 predictors with 8 macroeconomic predictors (such as the aggregate dividend-price ratio) from the Welch and Goyal (2008) dataset, as well as 74 industry dummies. Second, the gradient boosted regression trees (GBRT) in GKX is implemented using the scikit-learn

Table 3
Performance of portfolios sorted by NN predicted returns. This table shows the performance of long–short portfolios sorted by neural network (NN) predicted returns. We consider NN models with hidden layers that range from 1 through 5. The first three columns report average monthly returns for the long–short portfolios as well as the associated annualized Sharpe ratios. The remaining columns report risk-adjusted returns—see Table 2 for details. The top panel reports results for equal-weighted returns. The bottom panel reports results for value-weighted returns. All returns are expressed in percent per month.

Equal Weight															
Method	Returns		SR	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	Avg	*t-stat*		*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
NN1	1.08	6.09	1.16	1.17	6.51	1.11	6.17	1.14	6.78	0.97	5.71	1.01	6.18	1.01	5.59
NN2	1.03	4.10	0.75	1.17	4.83	1.03	4.89	1.11	6.31	0.81	2.93	0.89	3.70	0.73	2.34
NN3	1.17	5.32	1.10	1.30	5.55	1.17	6.21	1.15	6.40	0.89	5.35	0.90	5.41	0.86	4.53
NN4	0.99	5.53	0.98	1.10	5.97	1.02	5.84	1.10	7.04	0.93	4.49	0.99	5.28	0.92	4.08
NN5	0.80	3.79	0.74	0.89	4.17	0.80	3.94	0.84	4.77	0.56	2.79	0.61	3.30	0.54	2.37
Value Weight															
Method	Returns		SR	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	Avg	*t-stat*		*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
NN1	0.74	4.58	0.70	0.74	4.51	0.68	4.22	0.72	3.84	0.62	3.55	0.66	3.61	0.67	3.65
NN2	0.32	1.42	0.23	0.48	2.07	0.36	1.85	0.43	2.17	0.12	0.52	0.19	0.83	0.12	0.48
NN3	0.51	2.05	0.40	0.65	2.62	0.52	2.32	0.56	2.62	0.13	0.78	0.20	1.20	0.22	1.19
NN4	0.42	2.47	0.36	0.51	3.15	0.47	2.93	0.58	3.07	0.44	2.42	0.53	2.75	0.55	2.77
NN5	0.21	1.00	0.16	0.33	1.50	0.29	1.38	0.28	1.39	0.03	0.15	0.05	0.23	0.03	0.13

Table 4
Performance of portfolios sorted by ML predicted returns on the GHZ sample. This table reports the returns and risk-adjusted performance for the long–short portfolios sorted by ML-predicted returns on the GHZ sample from 1987 to 2019. We predict stock monthly excess returns using the 94 signals collected by Green et al. (2017). We use a recursive window approach and select the optimal hyper-parameters using a cross-validation approach. Our initial estimation period is 1963–1986. The first 12 years is the training period and the second 12 years is the validation period. As we roll forward, the training period expands while the validation period stays at 12 years. The risk-adjusted performance are calculated based on the CAPM model, the Fama–French 3-factor model, the Carhart 4-factor model, the Fama–French 5-factor model, the Fama–French 5-factor model augmented with momentum factor, and the q-factor model. The top panel reports results for equal-weighted portfolios. The bottom panel reports results for value-weighted portfolios. All returns are expressed in percent per month.

Equal Weight															
Method	Returns		SR	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	Avg	*t-stat*		*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
BRT	3.57	8.95	2.35	3.67	9.40	3.69	9.15	3.37	8.62	3.67	7.86	3.43	8.05	3.36	6.74
NN1	3.64	8.37	2.62	3.69	8.18	3.67	8.47	3.42	7.78	3.48	8.13	3.32	7.53	3.40	7.87
NN2	4.21	8.80	2.77	4.29	8.86	4.30	8.94	4.04	8.41	4.17	8.48	3.99	8.11	4.07	8.07
NN3	4.16	8.65	2.64	4.25	8.71	4.26	8.75	3.99	8.39	4.15	8.31	3.96	8.18	4.03	7.92
NN4	4.19	8.77	2.81	4.26	8.73	4.28	8.73	3.98	8.29	4.10	8.12	3.89	7.98	3.97	7.74
NN5	3.73	7.99	2.21	3.83	8.18	3.86	8.29	3.45	6.97	3.76	7.18	3.47	6.55	3.55	6.35
Value Weight															
Method	Returns		SR	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	Avg	*t-stat*		*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
BRT	1.52	4.56	0.72	1.67	5.54	1.71	5.17	1.13	4.05	1.56	3.27	1.15	3.35	1.11	2.24
NN1	1.62	4.60	0.88	1.83	4.72	1.79	5.83	1.12	4.48	1.32	4.20	0.87	3.56	1.00	3.36
NN2	2.67	6.27	1.25	2.91	6.59	2.89	7.47	2.30	7.11	2.52	6.21	2.13	6.32	2.23	5.44
NN3	2.42	5.64	1.14	2.68	5.79	2.66	6.35	2.00	6.33	2.20	4.84	1.75	5.48	1.90	4.23
NN4	2.45	6.69	1.19	2.62	6.50	2.62	7.00	1.86	5.96	2.21	5.34	1.68	5.28	1.80	4.28
NN5	2.02	4.67	1.04	2.27	5.00	2.30	5.34	1.66	4.98	2.02	4.25	1.58	4.57	1.76	3.51

In Table 5, we report the results that use the Chen and Zimmermann (2022) covariates. The results for this set of covariates are even more impressive. For example, BRT generate an equal-weighted long–short return of 5.14% per month and a VW long–short return of 2.32% per month. Adjusting for risk using standard models reveals very similar findings. Furthermore, BRT deliver an equally-weighted Sharpe ratio of 3.64. The results for shallow neural networks are somewhat lower than those of BRT but still very strong.

Taken together, these results indicate that our ML implementation is capable of generating rather strong performance when we use published predictors. The fact that we are able to replicate the strong ML performance of previous studies when we use published predictors indicates that our ML implementation is not the reason why the performance of ML strategies based on our universe of fundamental signals is relatively weak. Overall, this analysis confirms that the performance differences between our machine-learning strategies and those in recent studies are primarily driven by the choice of input features rather than the specific machine-learning implementations used.

3.5. A simple recursive ranking strategy

Our findings suggest that the economic benefits for real-time investors from applying our machine-learning strategies are relatively modest. However, Yan and Zheng (2017, Table 3) have shown evidence that investors could obtain large economic gains by learning from a universe of return signals. In this section, we construct a simple recursive ranking strategy following Yan and Zheng (2017). Specifically, we

package in Python. We instead use the LightGBM implementation in Python. Third, despite our best efforts to replicate GKX's implementation of neural networks, it is possible that some differences remain.

Table 5
Performance of portfolios sorted by ML predicted returns on the CZ sample. This table reports the returns and risk-adjusted performance for the long–short portfolios sorted by ML-predicted returns on the CZ sample from 1987 to 2019. We predict stock monthly excess returns using the March 2022 data release from https://www.openassetpricing.com/data/, which contains the 207 signals collected by Chen and Zimmermann (2022). We use a recursive window approach and select the optimal hyper-parameters using a cross-validation approach. Our initial estimation period is 1963–1986. The first 12 years is the training period and the second 12 years is the validation period. As we roll forward, the training period expands while the validation period stays at 12 years. The risk-adjusted performance are calculated based on the CAPM model, the Fama–French 3-factor model, the Carhart 4-factor model, the Fama–French 5-factor model, the Fama–French 5-factor model augmented with momentum factor, and the q-factor model. The top panel reports results for equal-weighted portfolios. The bottom panel reports results for value-weighted portfolios. All returns are expressed in percent per month.

Equal Weight

Method	Returns		SR	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	Avg	*t-stat*		*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
BRT	5.14	10.35	3.64	5.28	10.40	5.23	10.79	4.91	10.85	5.02	10.00	4.81	10.32	4.84	9.53
NN1	4.66	9.43	3.46	4.78	9.33	4.75	9.74	4.62	9.49	4.62	9.39	4.55	9.39	4.60	8.83
NN2	4.91	10.07	3.57	5.05	10.03	5.00	10.48	4.85	10.21	4.78	10.01	4.69	10.03	4.73	9.67
NN3	4.65	10.06	3.41	4.78	9.98	4.74	10.21	4.57	9.99	4.57	9.64	4.47	9.72	4.49	9.30
NN4	4.62	9.40	3.37	4.73	9.25	4.69	9.73	4.59	9.33	4.53	9.45	4.47	9.25	4.51	9.00
NN5	4.57	9.50	3.29	4.67	9.45	4.66	9.66	4.53	9.24	4.58	9.38	4.49	9.24	4.54	8.65

Value Weight

Method	Returns		SR	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	Avg	*t-stat*		*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
BRT	2.32	7.21	1.25	2.59	8.41	2.59	8.09	1.92	9.01	2.30	5.67	1.83	7.49	1.96	4.60
NN1	2.30	6.65	1.40	2.43	6.66	2.39	7.02	1.89	6.41	2.23	5.51	1.87	5.95	1.95	4.87
NN2	2.84	8.69	1.75	3.08	9.33	3.02	9.93	2.51	9.04	2.71	8.08	2.36	8.47	2.40	7.57
NN3	2.25	6.39	1.33	2.46	6.75	2.40	6.95	1.95	6.63	2.06	5.61	1.76	5.92	1.86	5.22
NN4	2.34	6.37	1.29	2.49	6.49	2.45	7.22	1.99	6.56	2.17	7.57	1.86	6.93	1.93	6.51
NN5	2.27	7.70	1.41	2.43	8.08	2.38	8.02	1.99	7.40	2.21	6.54	1.93	6.90	1.99	5.74

follow Yan and Zheng (2017) and first construct a long–short strategy of stocks based on each fundamental signal in our sample. We then sort all signals each year into deciles based on the t-stat of their past long–short portfolio alphas using a recursive window. Finally, we form an equal-weighted portfolio of signals by going long in those signals ranked in the highest t-stat decile and shorting those signals ranked in the lowest t-stat decile. Unlike Yan and Zheng (2017), we apply a recursive-window approach instead of dividing the sample period into two halves. These portfolios are held for one year and rebalanced annually. To align with our machine-learning strategies, the out-of-sample evaluation begins in 1987. This strategy would have been accessible to real-time investors since it relies solely on historical data to form the portfolios. Importantly, this procedure can be viewed as a crude machine-learning strategy that selects a subset of predictors to be included in the final investment strategy out of the universe of available ones using the t-statistic of their past univariate performance.

Table 6 presents the out-of-sample performance of the recursive-ranking strategy. As in earlier tables, we provide both raw returns and a range of risk-adjusted returns, along with the Sharpe ratios. Panel A displays results for equal-weighted portfolios, while Panel B reports the results for value-weighted portfolios. Overall, the results highlight the strong performance of this investment strategy. For instance, the equal-weighted long–short portfolio generates an average monthly return of 0.87% with a t-statistic of 9.05 and a Sharpe ratio of 1.60. Notably, this Sharpe ratio exceeds those of our machine-learning strategies (1.02 for BRT and 1.16 for NN1). Similarly, the value-weighted long–short portfolio delivers an average monthly return of 0.80%, with a t-statistic of 6.60 and a Sharpe ratio of 1.17, again outperforming our machine-learning strategies, which had Sharpe ratios of 0.30 for BRT and 0.70 for NN1.

The fact that feeding the universe of predictors to our machine-learning methods results in performance that is not only inferior to the same methods that use curated sets of predictors but also worse than a simple recursive ranking strategy that incorporates feature engineering underscores the importance of imposing an appropriate structure or "inductive bias" on machine-learning algorithms (Goyal and Bengio, 2022). Inductive biases encompass preferences or constraints imposed on the hypothesis space to guide learning and improve generalization in machine learning methods. This effectively shapes the learning process by narrowing the focus of machine learning methods to specific predictor variables, thereby enhancing their performance and leading to greater economic gains for real-time investors.

3.6. Past-return signals

Our analyses so far have focused on fundamental signals. The main reason for this focus is that we can construct a "universe" of fundamental signals (Yan and Zheng, 2017). Past return-based signals represent another class of signals for which we can construct an "exhaustive" list. In this section, we follow Martin and Nagel (2022) and construct a universe of past return-based signals and then repeat our main analyses.[28] Specifically, we include in our universe the monthly returns during the past 120 months, excluding the most recent month. Martin and Nagel (2022) exclude the most recent month to avoid microstructure effects. Therefore, we have 119 past return-based signals in our baseline analysis. To gauge the impact of short-term reversal, we also repeat our analysis by adding the most recent month's return to the predictor set.

Our stock sample for this analysis consists of the NYSE, AMEX, and NASDAQ common stocks (with a CRSP share code of 10 or 11) with valid past return data. We exclude those stocks with a share price lower than \$1 at the end of month $t-1$. For ease of comparison with our analysis of fundamental signals and previous machine learning studies, the sample period of our past-return analysis spans from July 1963 to December 2019. We employ the same training, cross-validation, and out-of-sample testing periods as in our study of fundamental signals.

We continue to use BRT as the primary machine-learning method but also examine neural networks with 1 to 5 hidden layers. As in our analysis of fundamental signals, we form long–short portfolios of stocks based on the machine learning predicted returns. Specifically, we go long in the stocks with the highest predicted returns and short in the stocks with the lowest predicted returns. We track the performance of these portfolios for one month and compute the return spread between the long and short portfolios. For performance evaluation, we report alphas for the long–short portfolio using the CAPM, the

[28] Moritz and Zimmermann (2016) and Murray et al. (2024) also examine machine learning strategies based on past-return signals.

Table 6
Performance of portfolios sorted using a simple recursive ranking strategy. This table reports the returns and risk-adjusted performance for decile and long–short portfolios constructed using a simple recursive ranking strategy from 1987 to 2019. We use the baseline samples in Table 1, and conduct the analysis using a recursive window specification. We first construct a long–short strategy based on each fundamental signal and then perform a simple recursive ranking of all signals according to their past strategy performance. Finally, we form decile and long–short portfolios based on this ranking. The first three columns report average monthly returns for the long–short portfolios as well as the associated annualized Sharpe ratios. The risk-adjusted performance is calculated based on the CAPM model, the Fama–French 3-factor model, the Carhart 4-factor model, the Fama–French 5-factor model, the Fama–French 5-factor model augmented with momentum factor, and the q-factor model. The top panel reports results for equal-weighted portfolios. The bottom panel reports results for value-weighted portfolios. All returns are expressed in percent per month.

Equal Weight

Rank	Returns		SR	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	Avg	*t-stat*		*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
L(ow)	−0.48	−8.34	−1.47	−0.54	−6.18	−0.51	−6.45	−0.44	−6.67	−0.41	−5.49	−0.36	−6.02	−0.38	−5.11
2	−0.25	−9.66	−1.71	−0.26	−6.56	−0.26	−7.02	−0.24	−7.43	−0.24	−7.45	−0.23	−7.90	−0.24	−6.21
3	−0.15	−6.89	−1.22	−0.14	−5.12	−0.16	−6.68	−0.16	−6.86	−0.17	−7.45	−0.16	−7.38	−0.17	−5.78
4	−0.07	−2.12	−0.37	−0.04	−1.12	−0.06	−2.15	−0.08	−2.98	−0.11	−3.60	−0.12	−3.96	−0.13	−3.44
5	−0.01	−0.40	−0.07	0.01	0.41	−0.01	−0.33	−0.03	−1.34	−0.07	−2.60	−0.08	−2.96	−0.09	−2.69
6	0.06	1.51	0.27	0.10	2.52	0.07	2.79	0.03	1.26	0.00	−0.12	−0.03	−0.85	−0.04	−1.06
7	0.14	4.11	0.73	0.16	4.75	0.15	6.30	0.11	4.96	0.08	3.25	0.05	2.24	0.05	1.68
8	0.22	4.45	0.79	0.26	5.23	0.24	6.76	0.18	5.84	0.13	4.09	0.10	3.12	0.09	2.27
9	0.29	7.96	1.41	0.32	7.86	0.30	9.61	0.26	9.99	0.22	7.38	0.20	7.30	0.20	6.00
H(igh)	0.39	8.88	1.57	0.44	6.75	0.41	8.42	0.35	9.15	0.29	7.02	0.26	8.07	0.28	7.27
H-L	0.87	9.05	1.60	0.97	6.64	0.91	7.47	0.79	7.96	0.70	6.25	0.63	7.17	0.66	6.15

Value Weight

Rank	Returns		SR	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	Avg	*t-stat*		*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
L(ow)	−0.37	−5.65	−1.00	−0.45	−4.20	−0.40	−5.19	−0.34	−4.88	−0.26	−5.17	−0.22	−4.94	−0.24	−4.44
2	−0.20	−7.27	−1.29	−0.22	−4.45	−0.21	−5.25	−0.18	−4.90	−0.16	−5.31	−0.15	−4.91	−0.16	−4.92
3	−0.13	−5.24	−0.93	−0.11	−3.95	−0.12	−4.74	−0.11	−4.45	−0.11	−4.78	−0.11	−4.48	−0.12	−4.29
4	−0.07	−2.90	−0.51	−0.04	−1.89	−0.05	−2.79	−0.06	−3.04	−0.08	−3.55	−0.08	−3.73	−0.08	−3.35
5	−0.03	−0.96	−0.17	0.01	0.25	−0.01	−0.63	−0.02	−1.47	−0.05	−2.89	−0.06	−3.34	−0.06	−3.00
6	0.04	1.20	0.21	0.08	2.34	0.06	2.61	0.04	1.73	0.00	−0.18	−0.02	−1.15	−0.02	−0.83
7	0.10	2.77	0.49	0.15	3.45	0.12	4.34	0.09	3.60	0.05	2.25	0.03	1.40	0.03	1.24
8	0.18	3.61	0.64	0.25	3.97	0.22	5.02	0.17	4.56	0.10	4.14	0.07	3.70	0.08	2.63
9	0.28	5.36	0.95	0.35	4.55	0.32	5.63	0.27	5.54	0.20	5.73	0.17	6.00	0.18	4.97
H(igh)	0.43	7.00	1.24	0.50	4.53	0.47	5.55	0.40	5.54	0.32	5.66	0.29	5.96	0.30	5.44
H-L	0.80	6.60	1.17	0.95	4.44	0.87	5.51	0.74	5.39	0.58	5.70	0.51	5.84	0.54	5.21

Fama–French three-factor model, and the Carhart four-factor model, the Fama–French five-factor alphas, Fama–French five-factor plus momentum factor alphas, and q-factor alphas. We report results for both equal-weighted and value-weighted portfolios.

Table 7 report the results. Panel A reports the results for our baseline sample that excludes the most recent month, i.e., 119 past return signals, while Panel B reports the results for 120 past return signals that include the most recent month. In each panel, we report the results for BRT as well as neural networks with 1–5 hidden layers. As in previous tables, we report raw returns, risk-adjusted returns, and Sharpe ratios.[29]

In Panel A, we find that the BRT strategy based on past-return signals earns an average return of 1.38% per month (t-statistic = 4.93) and exhibits an annualized Sharpe ratio of 1.04 in equal-weighted portfolios.[30] The performance of value-weighted portfolios is significantly weaker. The average long–short return is 0.78% per month (t-statistic = 2.41), while the Sharpe ratio is 0.46. The results based on neural network forecasts are qualitatively similar, with shallow networks (NN1 and NN2) performing similarly to BRT and deep networks (NN3 through NN5) performing worse than BRT.

Risk-adjusted returns indicate that the performance is significantly reduced when we control for the momentum factor. For example, the Carhart alpha is 1.09% (t-statistic = 6.62) for equal-weighted portfolios and 0.63% (t-statistic = 3.05) for value-weighted portfolios. The FF5+MOM alpha is even lower, at 0.78% (t-statistic = 5.83) for equal-weighted portfolios and 0.28% (t-statistic = 1.55) for value-weighted portfolios. The smaller Carhart alpha and the smaller FF5+MOM alpha are not surprising because much of the predictive ability of past returns is related to the momentum effect of Jegadeesh and Titman (1993).

The results reported in Panel B, which includes the short-term reversal, are measurably higher than those reported in Panel A. Specifically, we find that the BRT strategy earns an average return of 1.81% per month (t-statistic = 6.40) and exhibits an annualized Sharpe ratio of 1.77 in equal-weighted portfolios. The average long–short return for value-weighted portfolios is 0.98% per month (t-statistic = 3.14), while the Sharpe ratio is 0.66. The results based on neural network forecasts are qualitatively similar. We again find that shallow networks (NN1 and NN2) perform better than deep networks. Risk-adjusted returns continue to indicate that the performance is significantly reduced when we control for the momentum factor.

Overall, our results based on past-return signals are broadly consistent with those based on fundamental signals. Specifically, we find significant long–short returns for our machine learning strategies, suggesting that real-time investors benefit from machine learning forecasts. However, the performance of these real-time machine-learning strategies is weaker or significantly weaker than those reported in the prior literature. For example, Moritz and Zimmermann (2016) and Murray et al. (2024) use past 12 or 24 monthly returns to construct predictive signals and find Sharpe ratios of 2.96 (equal-weighted) and 0.78 (value-weighted), respectively. In comparison, our past-return-based machine learning strategies deliver a Sharpe ratio of 1.04 for EW portfolios and 0.46 for VW portfolios when we exclude short-term reversal, and a Sharpe ratio of 1.77 for EW portfolios and 0.66 for VW portfolios when we include short-term reversal. It is important to note that the higher Sharpe ratios of Moritz and Zimmermann (2016) and Murray et al. (2024), like those of our machine-learning strategies, are available to real-time investors. Comparing our Sharpe ratios with those

[29] For brevity, we only report the long–short portfolio returns in this table.
[30] Appendix D reports the top 25 past-return signals. Return during month $t-24$ is the most important predictor, followed by return during month $t-12$. Overall, the list is dominated by past returns during the past two years.

Table 7
Performance of portfolios constructed using past-return signals. This table reports the returns and risk-adjusted performance for the long–short portfolios sorted by ML-predicted returns based on past-return signals from 1987 to 2019. We predict stock monthly excess returns using 119 or 120 past-return signals (PR119 and PR120 as described in Section 3.6). Our sample starts in 1963, and the out-of-sample periods begin in 1987, which is consistent with our baseline specifications on fundamental signals. We use a recursive window approach and select the optimal hyper-parameters using a cross-validation approach. Our initial estimation period is 1963–1986. The first 12 years is the training period, and the second 12 years is the validation period. As we roll forward, the training period expands while the validation period stays at 12 years. The risk-adjusted performance is calculated based on the CAPM model, the Fama–French 3-factor model, the Carhart 4-factor model, the Fama–French 5-factor model, the Fama–French 5-factor model augmented with momentum factor, and the q-factor model. The top panel reports results for equal-weighted portfolios. The bottom panel reports results for value-weighted portfolios. All returns are expressed in percent per month.

Panel A: PR119

Method	Returns		SR	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	Avg	*t-stat*		*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
Equal Weight															
BRT	1.38	4.93	1.04	1.63	5.86	1.56	6.90	1.09	6.62	1.09	4.82	0.78	5.83	0.78	3.89
NN1	1.37	5.49	1.01	1.51	6.51	1.55	6.82	0.89	6.41	1.27	4.58	0.81	5.20	0.84	2.87
NN2	1.52	6.07	1.09	1.66	7.54	1.71	7.75	1.02	8.66	1.43	4.95	0.95	6.65	0.99	3.12
NN3	1.06	4.20	0.86	1.17	4.97	1.19	5.41	0.63	4.53	0.93	3.74	0.54	3.51	0.59	2.13
NN4	1.19	6.54	1.09	1.29	6.86	1.33	7.44	0.87	8.20	1.10	5.85	0.77	7.29	0.87	4.82
NN5	0.66	4.23	0.68	0.76	5.28	0.77	5.26	0.44	3.11	0.58	2.84	0.35	2.32	0.35	1.60
Value Weight															
BRT	0.78	2.41	0.46	1.17	4.00	1.07	4.23	0.63	3.05	0.56	2.37	0.28	1.55	0.28	1.36
NN1	0.99	3.34	0.58	1.12	3.96	1.22	4.28	0.42	2.18	1.00	2.77	0.44	2.13	0.60	1.55
NN2	1.11	3.97	0.63	1.28	4.75	1.39	5.24	0.58	3.27	1.19	3.67	0.61	3.53	0.82	2.26
NN3	0.74	2.28	0.48	0.83	2.72	0.90	2.91	0.26	1.12	0.80	2.24	0.34	1.34	0.44	1.09
NN4	0.79	3.97	0.56	0.91	4.17	1.01	5.13	0.52	2.62	0.91	4.37	0.56	3.01	0.72	3.39
NN5	0.56	3.17	0.46	0.70	4.17	0.72	4.51	0.33	1.99	0.53	2.85	0.26	1.58	0.39	1.73

Panel B: PR120

Method	Returns		SR	CAPM		FF3		Carhart		FF5		FF5+MOM		Q	
	Avg	*t-stat*		*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*	*alpha*	*t-stat*
Equal Weight															
BRT	1.81	6.40	1.77	1.83	6.21	1.76	6.76	1.67	6.08	1.57	6.67	1.52	5.98	1.47	5.13
NN1	1.83	8.43	1.77	1.77	8.06	1.80	8.22	1.49	7.48	1.71	8.47	1.49	7.22	1.50	6.98
NN2	1.87	7.85	1.83	1.84	7.89	1.88	7.91	1.56	6.84	1.80	7.97	1.58	6.68	1.58	6.58
NN3	1.46	6.43	1.41	1.44	6.43	1.46	6.68	1.13	5.70	1.36	6.85	1.13	5.45	1.10	5.49
NN4	1.41	5.80	1.46	1.39	5.75	1.41	5.88	1.16	4.92	1.32	5.87	1.14	4.85	1.15	4.59
NN5	1.46	6.53	1.45	1.46	6.15	1.46	6.45	1.22	5.55	1.33	6.29	1.16	5.18	1.15	4.88
Value Weight															
BRT	0.98	3.14	0.66	1.13	3.61	1.04	3.77	0.74	2.76	0.72	3.17	0.52	2.14	0.42	1.50
NN1	1.13	4.96	0.79	1.13	4.78	1.21	5.31	0.69	3.59	1.15	5.29	0.77	3.55	0.84	3.56
NN2	1.32	5.32	0.94	1.35	5.48	1.42	5.91	0.86	4.65	1.34	5.46	0.94	4.47	0.98	3.81
NN3	1.04	3.69	0.72	1.03	3.65	1.09	3.99	0.57	3.03	0.97	3.41	0.60	3.12	0.66	2.40
NN4	1.00	3.97	0.74	1.04	4.06	1.12	4.47	0.69	3.15	1.07	4.28	0.75	3.23	0.84	3.23
NN5	0.75	3.56	0.58	0.74	3.36	0.80	3.69	0.37	2.10	0.76	3.14	0.45	2.15	0.54	2.16

documented by Moritz and Zimmermann (2016) and Murray et al. (2024) suggests that using a curated set of inputs – a form of feature engineering – is critical for the performance of ML strategies.

4. Additional results

In this section, we provide several extensions and robustness tests of our baseline analysis. Section 4.1 employs rolling windows instead of recursive windows in estimating the BRT model. Section 4.2 studies whether our results are robust to alternative training and validation periods. Section 4.3 examines the performance of BRT long–short portfolios separately for large and small stocks. Section 4.4 examines the after-trading-cost performance of our machine-learning strategies. Finally, Section 4.5 investigates whether the performance of BRT portfolios varies with economic and market conditions. In all cases, we use our universe of fundamental signals as input for our machine-learning methods. For brevity, we report the results of these additional analyses in the Internet Appendix.

4.1. Rolling windows

We use recursive windows in our baseline specification to align ourselves with the majority of the literature (e.g., Gu et al., 2020). Recursive windows allow for incorporating all available information in generating forecasts, but they can lead to poor forecasts if the data-generating process changes over time. An alternative to recursive windows is rolling windows that generate forecasts based on less information and hence are potentially less precise but are more robust to time variations in the relation between fundamental signals and returns. If the relation between the fundamental signals and stock returns is time-varying, rolling windows may improve the predictive power of machine learning algorithms. To assess this possibility, we repeat our main analysis using the rolling window approach described below.

We set the initial estimation period to 24 years so that our out-of-sample test period starts from 1987, the same as in the recursive window approach. To select the optimal hyper-parameters, we split the 24 years into training and validation periods following our baseline specification. In particular, our training period is 12 years, and the validation period is 12 years.[31] After obtaining the optimal hyper-parameters, we re-estimate the final model using the 24-year window. Each year we refit the model by moving the 24-year window forward

[31] We have considered several alternative training and validation periods and find our results to be qualitatively similar.

by one year. The estimation period is fixed at 24 years under the rolling window approach. In comparison, under the recursive window approach, the estimation period expands as we roll forward.

Table IA.1 presents the performance of BRT portfolios for the rolling window approach. We find that the equally weighted portfolios achieve a long–short return of 0.83% per month (t-statistic = 4.27) and a Sharpe ratio of 0.77. These numbers are lower than their counterparts for the recursive window approach. Specifically, in Table 1 we report that the equal-weighted portfolios exhibit a long–short return of 0.95% (t-statistic = 6.63) and a Sharpe ratio of 1.02. The risk-adjusted returns for the rolling window approach are also correspondingly lower than those for the recursive window approach. The results for value-weighted portfolios paint a similar picture. For example, the average long–short return is 0.33% (t-statistic = 1.35) under the rolling window approach, compared to the 0.40% (t-statistic = 2.34) under the recursive window approach. Overall, we find that the performance of BRT portfolios based on a universe of predictors is somewhat weaker for the rolling window approach than for the recursive window approach.

4.2. Alternative training and validation periods

In our baseline specification, we use an initial training period of 12 years and a validation period of 12 years. In comparison, Gu et al. (2020) employ an initial training period of 18 years and a validation period of 12 years. As explained earlier, we choose an initial training period of 12 years because we want to start our out-of-sample test period in 1987, the same as in Gu et al. (2020). In this section, we examine whether our results are robust to our choices of the initial training period and validation period. Specifically, we consider nine alternative specifications in which the initial training period varies from 10 to 18 years, while the validation period varies from 10 to 14 years. We examine the performance of BRT portfolios under each of these alternative specifications.

Table IA.2 presents the results. The top panel reports the results for equal-weighted portfolios, while the bottom panel reports the results for value-weighted portfolios. For convenience, we reproduce the results for our baseline specification in the first row of each panel. Our baseline specification is denoted as "12+12", meaning 12 years of initial training period and 12 years of validation period. We denote the alternative specifications similarly. For example, "18+12" means 18 years of initial training and 12 years of validation period.

Overall, our results are highly robust across all alternative specifications. For example, the equal-weighted long–short returns range from 0.87% to 1.02% across the alternative specifications, compared to 0.95% for the baseline specification. Similarly, the value-weighted long–short returns range from 0.37% to 0.55% across the alternative specifications, compared to 0.40% for the baseline specification. The level of statistical significance for the long–short returns is also similar between the baseline and alternative specifications. Finally, the results on risk-adjusted returns are also robust to alternative specifications of initial training and validation periods.

4.3. Focusing on stocks with different market capitalizations

To evaluate whether the performance of our machine learning strategies varies across stocks with different capitalizations, each year we divide our sample stocks into two groups based on the median market capitalization: those above the median are large stocks and those below the median are small stocks. We then repeat our baseline analysis for each of these two groups of stocks and report the results in Table IA.3.

The top panel of Table IA.3 reports the results for equal-weighted portfolios. We find that the raw and risk-adjusted long–short returns are positive and significant for both large and small stocks. More importantly, the long–short performance is significantly higher for small stocks than for large stocks. Specifically, the long–short return is 0.63% per month (t-statistic = 2.93) for large stocks and is 1.13% (t-statistic = 6.14) for small stocks. The lower predictive performance for large stocks is not surprising. These stocks are likely to incorporate new information more quickly and are hence less likely to be predictable using machine learning algorithms.

The results for value-weighted portfolios are qualitatively similar. The average long–short return for large stocks is only 0.27% (t-statistic = 1.23). The long–short returns for large stocks do become marginally significant when we control for risks using the Carhart 4-factor model, the Fama–French 5-factor model, the Fama–French 5-factor augmented with momentum, and the q-factor model. In comparison, the average long–short return for small stocks is economically and statistically significant whether we examine raw or risk-adjusted returns. For example, the average long–short return for small stocks is 1.16% (t-statistic = 5.50).

Overall, the results in Table IA.3 indicate that the long–short performance of BRT portfolios is weaker for large stocks than for small stocks. This finding suggests that machine learning methods based on a universe of predictors are better at predicting the returns of smaller stocks, for which news is incorporated more slowly into asset prices.

4.4. After-trading-cost performance

For ease of comparison with prior literature (e.g., Gu et al., 2020; Freyberger et al., 2020; Chen et al., 2024c), we focus on the gross performance of our machine learning strategies in this paper. There is, however, growing attention to trading costs in the anomaly literature and ML literature (e.g., Novy-Marx and Velikov, 2016; Chen and Velikov, 2022; Jensen et al., 2022). In this section, we provide a simple analysis of the net performance (after-trading-cost returns) of our machine-learning strategies based on a universe of predictors.

We follow the general approach of Chen and Velikov (2022) to calculate turnover, trading costs, and net returns to long–short trading strategies. We also use their low-frequency (LF) measures of effective spreads as our trading cost measure.[32] These four LF measures are (i) (Hasbrouck, 2009)'s Gibbs sampler estimate, (ii) (Corwin and Schultz, 2012)'s high-low measure, (iii) (Kyle and Obizhaeva, 2016)'s volume-over-volatility measure, and (iv) (Abdi and Ranaldo, 2017)'s close-high-low measure. Following Chen and Velikov (2022), we use the average of the four low-frequency (LF) measures of effective spreads.

In Table IA.4, we show that the turnover rate for our BRT strategy based on fundamental signals is fairly low, with a two-sided turnover of 14% per month for both EW portfolios and VW portfolios. These relatively low turnover rates are not surprising because most of the fundamental signals are updated annually and we rebalance our portfolios once a year. We find that trading costs account for significantly less than half of the gross returns to our ML strategy. The net returns to the BRT strategy remain positive, at 0.73% per month for EW portfolios and 0.25% for VW portfolios. The net returns of NN strategies are also positive and of similar magnitude. We note that the gross returns reported here are slightly different from those of our baseline analysis. This is because the trading cost data is available only up to 2017, so the sample period for this analysis is slightly shorter than our baseline analysis.

Table IA.5 reports the corresponding results for our past-return-based machine-learning strategies. In contrast to those for fundamental signals, we find that the turnover rate for past-return-based machine-learning strategies is extremely high, well over 100% in both equal- and value-weighted portfolios. As a consequence, we find that net returns to

[32] Due to the data availability issue, we do not adopt their high-frequency (HF) measures of effective spreads. We download the LF data from Andrew Chen's website at https://sites.google.com/site/chenandrewy/. We note their data is available up to 2017, so our analysis ends in 2017.

machine learning strategies are consistently negative. For example, the net return for BRT strategies is −0.97% per month for equal-weighted portfolios and −0.29% for value-weighted portfolios. Adding short-term reversal to the predictor set improves the gross returns but makes the net returns even worse. Specifically, the net return is −1.48% per month for equal-weighted portfolios and −0.40% for value-weighted portfolios after including the short-term reversal. The results for NN strategies are similar to those for BRT strategies.

Chen and Velikov (2022) note that LF spreads are biased upward by 25–50 basis points (compared to HF effective spreads) post decimalization. As such, the net returns to our machine learning strategies reported in Table IA.4 and Table IA.5 may be too low. We decided not to make an ad-hoc adjustment related to this bias because despite their upward bias relative to HF spreads, the LF spreads may underestimate the total trading costs because they do not include other important components of trading costs, such as the cost of short selling and price impact. The shorting cost is particularly important for us because our machine learning strategies are long–short strategies.

Overall, we show that the net performance of ML strategies based on a universe of predictors is positive for fundamental signals and negative for past-return signals. We acknowledge that our analysis is exploratory and preliminary. An in-depth trading cost analysis that incorporates HF spreads, shorting cost, and price impact is a promising area of future research in the machine learning literature.[33]

4.5. Testing for time-varying predictability

In Table IA.7, we examine whether the profitability of BRT strategies varies with economic and market conditions. Specifically, we split our sample period based on investor sentiment,[34] the VIX index also known as the "fear-gauge", market liquidity (Pástor and Stambaugh, 2003), business cycle indicators as published by NBER, and market state—proxied by the cumulative market returns over the previous 24 months. We also divide our sample period into two halves (1987–2003 and 2003–2019) to examine whether the predictability declines over time.

Panel A shows the long–short portfolio returns for high- and low-sentiment periods. When examining equal-weighted returns, we find significant predictability during both high- and low-sentiment periods. In contrast, value-weighted returns are only marginally significant during low-sentiment periods and insignificant during high-sentiment periods. Whether we look at equal- or value-weighted returns, the difference in long–short returns between high- and low-sentiment periods is statistically insignificant. We find similar results in Panel B, where we divide the sample period into high- and low-VIX periods, and in Panel C, where we divide periods into high- and low-liquidity periods. In each panel, we find significant equal-weighted returns across both subperiods. The value-weighted returns, however, are either insignificant or marginally significant. As in Panel A, we find little significant evidence of differential predictability across subperiods. We also find little difference in predictability between recession and expansion periods in Panel D.

In Panel E, we split the sample period into UP and DOWN market states based on previous 24-month cumulative market returns. We find that the long–short return is higher during UP states than during DOWN states. Specifically, the equal-weighted long–short return is 1.24% during UP states and 0.67% during DOWN states. Similarly, the value-weighted long–short return is 0.79% during UP states and 0% during DOWN states. The differences in long–short returns between the UP and DOWN states are economically large and statistically marginally significant. In Panel F, we divide our sample period into two halves and find no statistically significant difference in predictability during the first and second half of our sample period.

Overall, the results in Table IA.7 indicate that the return predictability implied by our machine learning strategies based on a universe of predictors does not change significantly with investor sentiment, market volatility, market liquidity, or business cycle. However, there is some evidence that the profitability of our BRT strategies varies systematically with the state of the market. Finally, we find no evidence that the return predictability differs significantly across the two halves of our sample period.

5. Conclusions

We develop real-time machine-learning strategies based on a broad universe of fundamental signals. These strategies exhibit out-of-sample performance that is both economically meaningful and statistically significant; however, their long–short returns and Sharpe ratios are considerably lower than those reported in earlier studies that use curated sets of signals as return predictors. Our findings suggest that the difference in performance is driven by the differences in input data rather than differences in the implementation of the machine learning algorithms. We also show that strategies employing a simple recursive ranking based on each signal's past performance achieve substantially better out-of-sample results. The fact that feeding the universe of predictors to our machine-learning methods results in performance that is not only inferior to the same methods that use curated sets of predictors but also worse than a simple recursive ranking strategy that incorporates feature engineering underscores the importance of imposing an appropriate structure or "inductive bias" for machine learning algorithms to perform effectively in cross-sectional prediction tasks. As Domingos (2012) puts it, "feature engineering is the key" in machine learning applications. Our analyses using past-return signals yield similar conclusions to those based on fundamental signals. Specifically, while our real-time machine learning strategies produce economically meaningful and statistically significant performance, using curated sets of inputs can significantly enhance this performance for real-time investors. In summary, our results suggest that employing machine learning methods is beneficial for real-time investors, and that feature engineering plays a vital role in substantially elevating these benefits.

CRediT authorship contribution statement

Bin Li: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Alberto G. Rossi:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Xuemin (Sterling) Yan:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Lingling Zheng:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

[33] We also examine the after-trading cost performance of machine-learning strategies based on the GHZ and CZ samples of published anomalies. For brevity, we report the results in Table IA.6 in the Internet Appendix.

[34] We obtain the investor sentiment's data from Wurgler's website at http://people.stern.nyu.edu/jwurgler/.

Table A.1
Grids of hyper-parameters for cross validation and implementation details for BRT and NN. This table shows the grids of hyper-parameters used in the cross validation of boosted regression trees (BRT) and neural networks (NN). We follow Gu et al. (2020) to select the grids of hyper-parameters.

BRT	NN
# of iteration ∈ {100, 250, 500, 750, 1000}	L1 penalty $\lambda_1 \in \{10^{-5}, 10^{-3}\}$
learning rate ∈ {0.01, 0.05, 0.1}	Learning Rate LR ∈ {0.001, 0.01}
	Batch Size = 10000
	Epochs = 100
	Patience = 5
	Ensemble = 10
	Adam Para. = Default

Appendix A

We implement BRT using LightGBM package in Python (version: 3.1.1) using the hyper-parameters' grid reported in Table A.1 and minimizing the standard L2 objective function.

For the implementation of neural networks, we follow Gu et al. (2020) and Chen and McCoy (2024) and build 5 neural networks, including NN1 to NN5. NN1 has hidden layers with 32 neurons, NN2 has hidden layers with 32 and 16 neurons, NN3 has hidden layers with 32, 16, and 8 neurons, NN4 has hidden layers with 32, 16, 8, and 4 neurons, and NN5 has hidden layers with 32, 16, 8, 4, and 2 neurons. All layers are connected with the ReLU activation function. The objective is L2 with an L1 penalty to weight parameters and the minimization is performed using the Adam extension of the Stochastic Gradient Descent under early stopping with a patience parameter 5 and batches of 10,000, for 100 epochs. We also include batch normalization. Finally, we construct the final forecasts as the ensemble average of 10 neural network forecasts.

Appendix B

See Table B.1.

Appendix C

See Table C.1.

Appendix D. Relative influence measures

One criticism of machine learning algorithms is that they are "Black Boxes" that do not provide a lot of intuition to the researcher and the reader. This criticism hardly applies to BRT that feature very useful and intuitive visualization tools. The first commonly used measure is referred to as the "relative influence" measure. Consider the reduction in the empirical error every time one of the covariates $x_{l,\cdot}$ is used to split the tree. Summing the reductions in empirical errors (or improvements in fit) across the nodes in the tree gives a measure of the variable's influence (Breiman et al., 1984):

$$I_l(\mathcal{T}) = \sum_{j=2}^{J} \Delta e(j)^2 \, I(x(j) = l),$$

where $\Delta e(j)^2 = T^{-1} \sum_{t=1}^{T} \left(e_t(j-1)^2 - e_t(j)^2 \right)$ is the reduction in the squared empirical error at the jth node and $x(j)$ is the regressor chosen at this node, so $I(x(j) = l)$ equals 1 if regressor l is chosen, and 0 otherwise. The sum is computed across all observations, $t = 1, \ldots, T$, and over the $J - 1$ internal nodes of the tree.

The rationale for this measure is that at each node, one of the regressors gets selected to partition the sample space into two substates. The particular regressor at node j achieves the greatest reduction in the empirical risk of the model fitted up to node $j-1$. The importance of each regressor, $x_{l,\cdot}$, is the sum of the reductions in the empirical errors computed over all internal nodes for which it was chosen as the splitting variable. If a regressor never gets chosen to conduct the splits, its influence is zero. Conversely, the more frequently a regressor is used for splitting, and the bigger its effect on reducing the model's empirical risk, the larger its influence.

This measure of influence can be generalized by averaging over the number of boosting iterations, B, which generally provides a more reliable measure of influence:

$$\bar{I}_l = \frac{1}{B} \sum_{b=1}^{B} I_l(\mathcal{T}_b).$$

This is best interpreted as a measure of relative influence that can be compared across regressors. We therefore report the following measure of relative influence, $\overline{RI}_l$, which sums to 1:

$$\overline{RI}_l = \frac{\bar{I}_l}{\sum_{l=1}^{L} \bar{I}_l}.$$

Figure IA.1 shows the relative influence of the top 25 signals in the baseline BRT model estimated in the paper. We first compute the signals' relative influence in each year of the test period,1987–2019, and average their values across all test years. Note that the relative importance measure across all signals sums to one every year. We then rank and plot the signals according to their average relative influence. The Y-axis reports the 25 most important signals, while the X-axis presents each signal's average relative influence measure.

Figure IA.2 shows the relative influence of the top 25 signals in the baseline BRT model on past return signals. We first compute the signals' relative influence in each month of the test period,1987–2019, and average their values across all test months. Note that the relative importance measure across all signals sums to one every month. We then rank and plot the signals according to their average relative influence. The Y-axis reports the 25 most important signals in terms of lags, while the X-axis presents each signal's average relative influence measure.

Appendix E. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jfineco.2025.104138.

Data availability

Replication_Package_LRYZ_Machine_Learning_Feature_Engineering (Reference data) (Mendeley Data)

Table B.1
List of accounting variables. This table lists the 240 accounting variables used in this study and their descriptions. Our sample period is 1963–2019. We begin with all accounting variables on the balance sheet, income statement, and cash flow statement included in the annual Compustat database. We exclude all variables with fewer than 20 years of data or fewer than 1,000 firms with non-missing data on average per year. We exclude per-share-based variables such as book value per share and earnings per share. We remove LSE (total liabilities and equity), REVT (total revenue), OIBDP (operating income before depreciation), and XDP (depreciation expense) because they are identical to TA (total assets), SALE (total sale), EBITDA (earnings before interest) and DFXA (depreciation of tangible fixed assets) respectively. Please refer to Yan and Zheng (2017) for more details.

#	*Variable*	*Description*	*Missing rate*	*Start year*
1	ACCHG	Accounting changes - cumulative effect	39.29%	1988
2	ACO	Current assets other total	0.76%	1963
3	ACOX	Current assets other sundry	2.20%	1963
4	ACT	Current assets - total	2.13%	1963
5	AM	Amortization of intangibles	33.03%	1965
6	AO	Assets – other	0.06%	1963
7	AOLOCH	Assets and liabilities other net change	38.36%	1988
8	AOX	Assets – other - sundry	2.22%	1963
9	AP	Accounts payable – trade	4.88%	1963
10	APALCH	Accounts payable & accrued liabilities increase/decrease	53.14%	1988
11	AQC	Acquisitions	12.98%	1972
12	AQI	Acquisitions income contribution	32.50%	1975
13	AQS	Acquisitions sales contribution	32.26%	1975
14	AT	Assets – total	0.01%	1963
15	BAST	Average short-term borrowing	74.28%	1978
16	CAPS	Capital surplus/share premium reserve	2.08%	1963
17	CAPX	Capital expenditure	2.18%	1963
18	CAPXV	Capital expenditure PPE Schedule V	1.39%	1963
19	CEQ	Common/ordinary equity - total	1.54%	1963
20	CEQL	Common equity liquidation value	1.62%	1963
21	CEQT	Common equity tangible	1.64%	1963
22	CH	Cash	12.33%	1963
23	CHE	Cash and short-term investments	0.72%	1963
24	CHECH	Cash and cash equivalents increase/decrease	28.77%	1972
25	CLD2	Capitalized leases - due in 2nd year	46.55%	1985
26	CLD3	Capitalized leases - due in 3rdyear	46.44%	1985
27	CLD4	Capitalized leases - due in 4thyear	46.18%	1985
28	CLD5	Capitalized leases - due in 5thyear	46.15%	1985
29	COGS	Cost of goods sold	0.09%	1963
30	CSTK	Common/ordinary stock (capital)	1.96%	1963
31	CSTKCV	Common stock-carrying value	28.31%	1963
32	CSTKE	Common stock equivalents – dollar savings	0.06%	1963
33	DC	Deferred charges	28.45%	1965
34	DCLO	Debt capitalized lease obligations	10.08%	1965
35	DCOM	Deferred compensation	72.02%	1980
36	DCPSTK	Convertible debt and stock	2.85%	1963
37	DCVSR	Debt senior convertible	9.89%	1970
38	DCVSUB	Debt subordinated convertible	11.96%	1970
39	DCVT	Debt – convertible	5.80%	1963
40	DD	Debt debentures	10.55%	1965
41	DD1	Long-term debt due in one year	5.05%	1963
42	DD2	Debt Due in 2nd Year	23.27%	1974
43	DD3	Debt Due in 3rd Year	23.32%	1974
44	DD4	Debt Due in 4th Year	23.16%	1974
45	DD5	Debt Due in 5th Year	24.04%	1974
46	DFS	Debt finance subsidiary	79.68%	1992
47	DFXA	Depreciation of tangible fixed assets	65.07%	1970
48	DILADJ	Dilution adjustment	62.54%	1994
49	DILAVX	Dilution available excluding extraordinary items	62.54%	1994
50	DLC	Debt in current liabilities - total	0.72%	1963
51	DLCCH	Current debt changes	60.86%	1974
52	DLTIS	Long-term debt issuance	10.50%	1972
53	DLTO	Other long-term debt	9.96%	1965
54	DLTP	Long-term debt tied to prime	38.66%	1975
55	DLTR	Long-term debt reduction	9.84%	1972
56	DLTT	Long-term debt - total	0.20%	1963
57	DM	Debt mortgages &other secured	33.76%	1981
58	DN	Debt notes	10.56%	1965
59	DO	Income (loss) from discontinued operations	3.66%	1963
60	DONR	Nonrecurring discontinued operations	71.10%	1994
61	DP	Depreciation and amortization	0.24%	1963
62	DPACT	Depreciation , depletion and amortization	0.44%	1963
63	DPC	Depreciation and amortization (cash flow)	8.59%	1972
64	DPVIEB	Depreciation ending balance (schedule VI)	19.34%	1970
65	DPVIO	Depreciation other changes (schedule VI)	65.12%	1970
66	DPVIR	Depreciation retirements (schedule VI)	65.14%	1970
67	DRC	Deferred revenue current	73.42%	1994

(*continued on next page*)

Table B.1 (*continued*).

#	Variable	Description	Missing rate	Start year
68	DS	Debt-subordinated	9.93%	1965
69	DUDD	Debt unamortized debt discount and other	29.51%	1963
70	DV	Cash dividends (cash flow)	8.55%	1972
71	DVC	Dividends common/ordinary	0.11%	1963
72	DVP	Dividends - preferred/preference	0.06%	1963
73	DVPA	Preferred dividends in arrears	17.95%	1964
74	DVPIBB	Depreciation beginning balance (schedule VI)	60.82%	1970
75	DVT	Dividends – total	0.11%	1963
76	DXD2	Debt (excl capitalized leases) due in 2nd year	49.31%	1985
77	DXD3	Debt (excl capitalized leases) due in 3rd year	49.25%	1985
78	DXD4	Debt (excl capitalized leases) due in 4thyear	48.96%	1985
79	DXD5	Debt (excl capitalized leases) due in 5thyear	49.36%	1985
80	EBIT	Earnings before interest and taxes	1.36%	1963
81	EBITDA	Earnings before interest	0.21%	1963
82	ESOPCT	ESOP obligation (common) - total	40.69%	1980
83	ESOPDLT	ESOP debt - long term	49.09%	1990
84	ESOPT	Preferred ESOP obligation - total	41.01%	1964
85	ESUB	Equity in earnings -unconsolidated subsidiaries	12.33%	1963
86	ESUBC	Equity in net loss earnings	22.05%	1972
87	EXRE	Exchange rate effect	38.46%	1988
88	FATB	Property, plant, and equipment buildings	51.33%	1985
89	FATC	Property, plant and equipment construction in progress	47.36%	1985
90	FATE	Property, plant, equipment and machinery equipment	53.32%	1985
91	FATL	Property, plant, and equipment leases	57.58%	1985
92	FATN	Property, plant, equipment and natural resources	47.37%	1985
93	FATO	Property, plant, and equipment other	52.84%	1985
94	FATP	Property, plant, equipment and land improvements	51.25%	1985
95	FIAO	Financing activities other	38.35%	1988
96	FINCF	Financing activities net cash flow	38.35%	1988
97	FOPO	Funds from operations other	7.83%	1972
98	FOPOX	Funds from operations - Other excl option tax benefit	76.37%	1992
99	FOPT	Funds from operations total	69.42%	1972
100	FSRCO	Sources of funds other	70.81%	1972
101	FSRCT	Sources of funds total	71.27%	1972
102	FUSEO	Uses of funds other	70.81%	1972
103	FUSET	Uses of funds total	71.61%	1972
104	GDWL	Goodwill	47.13%	1989
105	GP	Gross profit (loss)	0.09%	1963
106	IB	Income before extraordinary items	0.05%	1963
107	IBADJ	IB adjusted for common stock equivalents	0.05%	1963
108	IBC	Income before extraordinary items (cash flow)	7.82%	1972
109	IBCOM	Income before extraordinary items available for common	0.05%	1963
110	ICAPT	Invested capital – total	1.54%	1963
111	IDIT	Interest and related income - total	42.18%	1965
112	INTAN	Intangible assets – total	10.02%	1963
113	INTC	Interest capitalized	16.78%	1963
114	INTPN	Interest paid net	43.82%	1988
115	INVCH	Inventory decrease (increase)	43.46%	1988
116	INVFG	Inventories finished goods	41.28%	1970
117	INVO	Inventories other	52.52%	1984
118	INVRM	Inventories raw materials	40.27%	1969
119	INVT	Inventories – total	1.43%	1963
120	INVWIP	Inventories work in progress	43.69%	1970
121	ITCB	Investment tax credit (balance sheet)	3.20%	1963
122	ITCI	Investment tax credit (income account)	37.65%	1963
123	IVACO	Investing activities other	38.35%	1988
124	IVAEQ	Investment and advances – equity	9.07%	1963
125	IVAO	Investment and advances other	7.07%	1963
126	IVCH	Increase in investments	13.68%	1972
127	IVNCF	Investing activities net cash flow	38.35%	1988
128	IVST	Short-term investments – total	12.35%	1963
129	IVSTCH	Short-term investments change	48.38%	1988
130	LCO	Current liabilities other total	4.76%	1963
131	LCOX	Current liabilities other sundry	6.10%	1963
132	LCOXDR	Current liabilities-other-excl deferred revenue	72.40%	1994
133	LCT	Current liabilities – total	1.69%	1963
134	LIFR	LIFO reserve	22.04%	1976
135	LO	Liabilities – other – total	0.72%	1963
136	LT	Liabilities – total	0.50%	1963
137	MIB	Minority interest (balance sheet)	6.37%	1963
138	MII	Minority interest (income account)	10.24%	1963
139	MRC1	Rental commitments minimum 1styear	27.85%	1975

(*continued on next page*)

Table B.1 (*continued*).

#	Variable	Description	Missing rate	Start year
140	MRC2	Rental commitments minimum 2ndyear	28.34%	1975
141	MRC3	Rental commitments minimum 3rdyear	28.46%	1975
142	MRC4	Rental commitments minimum 4th year	28.61%	1975
143	MRC5	Rental commitments minimum 5th year	30.38%	1975
144	MRCT	Rental commitments minimum 5 year total	29.51%	1975
145	MSA	Marketable securities adjustment	18.18%	1976
146	NI	Net income (loss)	0.06%	1963
147	NIADJ	Net income adjusted for common stock equiv.	2.24%	1963
148	NIECI	Net income effect capitalized interest	59.92%	1976
149	NOPI	Non-operating income (expense)	0.10%	1963
150	NOPIO	Non-operating income (expense) other	0.10%	1963
151	NP	Notes payable short-term borrowings	0.80%	1963
152	OANCF	Operating activities net cash flow	38.36%	1988
153	OB	Order backlog	64.22%	1971
154	OIADP	Operating income after depreciation	0.07%	1963
155	PI	Pre-tax income	0.06%	1963
156	PIDOM	Pretax income domestic	74.94%	1981
157	PIFO	Pretax income foreign	75.36%	1981
158	PPEGT	Property, plant, and equipment – total (gross)	0.45%	1963
159	PPENB	Property, plant, and equipment buildings (net)	70.38%	1970
160	PPENC	Property plant equipment construction in progress (net)	65.66%	1970
161	PPENLI	Property plant equipment land and improvements (net)	70.26%	1970
162	PPENME	Property plant equipment machinery and equipment (net)	69.73%	1970
163	PPENNR	Property plant equipment natural resources (net)	69.31%	1970
164	PPENO	Property plant and equipment other (net)	69.26%	1970
165	PPENT	Property, plant, and equipment – total (net)	0.11%	1963
166	PPEVBB	Property plant equipment beginning balance (schedule V)	57.03%	1963
167	PPEVEB	Property, plant, and equipment ending balance	15.25%	1963
168	PPEVO	Property, plant, and equipment other changes (schedule V)	62.50%	1963
169	PPEVR	Property, plant and equipment retirements (schedule V)	62.50%	1963
170	PRSTKC	Purchase of common and preferred stock	12.98%	1972
171	PSTK	Preferred/preference stock (capital) – total	0.24%	1963
172	PSTKC	Preferred stock convertible	4.96%	1963
173	PSTKL	Preferred stock liquidating value	0.05%	1963
174	PSTKN	Preferred/preference stock – non-redeemable	1.48%	1963
175	PSTKR	Preferred/preference stock - redeemable	20.89%	1964
176	PSTKRV	Preferred stock redemption value	0.06%	1963
177	RDIP	In process R&D expense	65.68%	1994
178	RE	Retained earnings	2.04%	1963
179	REA	Retained earnings restatement	10.33%	1970
180	REAJO	Retained earnings other adjustments	30.06%	1983
181	RECCH	Accounts receivable decrease (increase)	41.58%	1988
182	RECCO	Receivables – current – other	3.21%	1963
183	RECD	Receivables – estimated doubtful	29.03%	1970
184	RECT	Receivables – total	1.45%	1963
185	RECTA	Retained earnings cumulative translation adjustment	30.39%	1983
186	RECTR	Receivables – trade	17.96%	1967
187	REUNA	Retained earnings unadjusted	29.89%	1983
188	SALE	Sales/turnover (net)	0.05%	1963
189	SEQ	Stockholders' equity – total	2.24%	1963
190	SIV	Sale of investments	16.24%	1972
191	SPI	Special items	3.93%	1963
192	SPPE	Sale of property	28.92%	1972
193	SPPIV	Sale of property plant equipment investments gain (loss)	38.36%	1988
194	SSTK	Sale of common and preferred stock	9.55%	1972
195	TLCF	Tax loss carry forward	23.48%	1963
196	TSTK	Treasury stock – total (all capital)	16.37%	1970
197	TSTKC	Treasury stock – common	26.38%	1974
198	TSTKP	Treasury stock – preferred	25.51%	1963
199	TXACH	Income taxes accrued increase/decrease	56.69%	1988
200	TXBCO	Excess tax benefit stock options -cash flow	76.43%	1992
201	TXC	Income tax – current	16.78%	1963
202	TXDB	Deferred taxes (balance sheet)	3.34%	1963
203	TXDBA	Deferred tax asset - long term	73.84%	1993
204	TXDBCA	Deferred tax asset - current	73.11%	1994
205	TXDBCL	Deferred tax liability - current	74.46%	1994
206	TXDC	Deferred taxes (cash flow)	10.38%	1972
207	TXDFED	Deferred taxes-federal	48.37%	1985
208	TXDFO	Deferred taxes-foreign	45.98%	1985
209	TXDI	Income tax – deferred	6.99%	1963
210	TXDITC	Deferred taxes and investment tax credit	3.34%	1963
211	TXDS	Deferred taxes-state	48.91%	1985
212	TXFED	Income tax federal	16.78%	1963
213	TXFO	Income tax foreign	19.02%	1970

(*continued on next page*)

Table B.1 (*continued*).

#	Variable	Description	Missing rate	Start year
214	TXNDB	Net deferred tax asset (liab) - total	69.95%	1994
215	TXNDBA	Net deferred tax asset	72.66%	1994
216	TXNDBL	Net deferred tax liability	72.67%	1994
217	TXNDBR	Deferred tax residual	72.05%	1994
218	TXO	Income taxes - other	33.11%	1963
219	TXP	Income tax payable	5.93%	1963
220	TXPD	Income taxes paid	45.36%	1988
221	TXR	Income tax refund	10.40%	1963
222	TXS	Income tax state	17.76%	1963
223	TXT	Income tax total	0.06%	1963
224	TXW	Excise taxes	24.39%	1976
225	WCAP	Working capital (balance sheet)	2.15%	1963
226	WCAPC	Working capital change other increase/decrease	72.51%	1972
227	WCAPCH	Working capital change total	74.62%	1972
228	XACC	Accrued expenses	19.16%	1963
229	XAD	Advertising expense	64.98%	1963
230	XDEPL	Depletion expense (schedule VI)	68.80%	1970
231	XI	Extraordinary items	1.60%	1963
232	XIDO	Extra. items and discontinued operations	0.06%	1963
233	XIDOC	Extra. items and disc. operations (cash flow)	9.44%	1972
234	XINT	Interest and related expenses – total	5.05%	1963
235	XOPR	Operating expenses – total	0.09%	1963
236	XPP	Prepaid expenses	43.96%	1963
237	XPR	Pension and retirement expense	25.03%	1963
238	XRD	Research and development expense	47.01%	1963
239	XRENT	Rental expense	14.34%	1963
240	XSGA	Selling, general and administrative expense	12.13%	1963

Table C.1
List of financial ratios and configurations. This table lists the 76 financial ratios and configurations used in this study. Our sample period is 1963–2019. We begin with all accounting variables on the balance sheet, income statement, and cash flow statement included in the annual Compustat database. We exclude all variables with fewer than 20 years of data or fewer than 1,000 firms with non-missing data on average per year. We exclude per-share-based variables such as book value per share and earnings per share. "X" represents the 240 accounting variables listed in Table B.1. "Y" represents the fifteen base variables, including AT (total assets), ACT (total current assets), INVT (inventory), PPENT (property, plant, and equipment), LT (total liabilities), LCT (total current liabilities), DLTT (long-term debt), CEQ (total common equity), SEQ (stockholders' equity), ICAPT (total invested capital), SALE (total sale), COGS (cost of goods sold), XSGA (selling, general, and administrative cost), EMP (number of employees), and MKTCAP (market capitalization). Please refer to Yan and Zheng (2017) for more details.

#	Description	#	Description	#	Description	#	Description	#	Description
1	X/AT	16	Δ in X/AT	31	%Δ in X/AT	46	ΔX/LAGAT	61	%Δ in X − %Δ in AT
2	X/ACT	17	Δ in X/ACT	32	%Δ in X/ACT	47	ΔX/LAGACT	62	%Δ in X − %Δ in ACT
3	X/INVT	18	Δ in X/INVT	33	%Δ in X/INVT	48	ΔX/LAGINVT	63	%Δ in X − %Δ in INVT
4	X/PPENT	19	Δ in X/PPENT	34	%Δ in X/PPENT	49	ΔX/LAGPPENT	64	%Δ in X − %Δ in PPENT
5	X/LT	20	Δ in X/LT	35	%Δ in X/LT	50	ΔX/LAGLT	65	%Δ in X − %Δ in LT
6	X/LCT	21	Δ in X/LCT	36	%Δ in X/LCT	51	ΔX/LAGLCT	66	%Δ in X − %Δ in LCT
7	X/DLTT	22	Δ in X/DLTT	37	%Δ in X/DLTT	52	ΔX/LAGDLTT	67	%Δ in X − %Δ in DLTT
8	X/CEQ	23	Δ in X/CEQ	38	%Δ in X/CEQ	53	ΔX/LAGCEQ	68	%Δ in X − %Δ in CEQ
9	X/SEQ	24	Δ in X/SEQ	39	%Δ in X/SEQ	54	ΔX/LAGSEQ	69	%Δ in X − %Δ in SEQ
10	X/ICAPT	25	Δ in X/ICAPT	40	%Δ in X/ICAPT	55	ΔX/LAGICAPT	70	%Δ in X − %Δ in ICAPT
11	X/SALE	26	Δ in X/SALE	41	%Δ in X/SALE	56	ΔX/LAGSALE	71	%Δ in X − %Δ in SALE
12	X/COGS	27	Δ in X/COGS	42	%Δ in X/COGS	57	ΔX/LAGCOGS	72	%Δ in X − %Δ in COGS
13	X/XSGA	28	Δ in X/XSGA	43	%Δ in X/XSGA	58	ΔX/LAGXSGA	73	%Δ in X − %Δ in XSGA
14	X/EMP	29	Δ in X/EMP	44	%Δ in X/EMP	59	ΔX/LAGEMP	74	%Δ in X − %Δ in EMP
15	X/MKTCAP	30	Δ in X/MKTCAP	45	%Δ in X/MKTCAP	60	ΔX/LAGMKTCAP	75	%Δ in X − %Δ in MKTCAP
								76	%Δ in X

References

Abdi, F., Ranaldo, A., 2017. A simple estimation of bid-ask spreads from daily close, high, and low prices. Rev. Financ. Stud. 30 (12), 4437–4480.

Arnott, R., Harvey, C.R., Markowitz, H., 2019. A backtesting protocol in the era of machine learning. J. Financ. Data Sci. 1 (1), 64–74.

Avramov, D., Kaplanski, G., Subrahmanyam, A., 2022. Postfundamentals price drift in capital markets: A regression regularization perspective. Manag. Sci. 68 (10), 7065–7791.

Bali, T.G., Beckmeyer, H., Mörke, M., Weigert, F., 2023. Option return predictability with machine learning and big data. Rev. Financ. Stud. 36 (9), 3548–3602.

Bianchi, D., Büchner, M., Tamoni, A., 2021. Bond risk premiums with machine learning. Rev. Financ. Stud. 34 (2), 1046–1089.

Breiman, L., Friedman, J., Stone, C.J., Olshen, R., 1984. Classification and Regression Trees. Chapman and Hall/CRC.

Bryzgalova, S., Pelger, M., Zhu, J., 2020. Forest through the Trees: Building Cross-Sections of Stock Returns. Tech. Rep., London Business School.

Carhart, M.M., 1997. On persistence in mutual fund performance. J. Financ. 52 (1), 57–82.

Chen, A.Y., 2024. Most Claimed Statistical Findings in Cross-Sectional Return Predictability are Likely True. Tech. Rep., arXiv.

Chen, A.Y., Dim, C., 2024. High-Throughput Asset Pricing. Tech. Rep., arXiv.

Chen, M., Hanauer, M.X., Kalsbach, T., 2024a. Design Choices, Machine Learning, and the Cross-Section of Stock Returns. SSRN Scholarly Paper 5031755, Social Science Research Network, Rochester, NY.

Chen, A.Y., Lopez-Lira, A., Zimmermann, T., 2024b. Peer-Reviewed Theory Does Not Help Predict the Cross-Section of Stock Returns. Tech. Rep., arXiv.

Chen, A.Y., McCoy, J., 2024. Missing values handling for machine learning portfolios. J. Financ. Econ. 155, 103815.

Chen, L., Pelger, M., Zhu, J., 2024c. Deep learning in asset pricing. Manag. Sci. 70 (2), 714–750.

Chen, A.Y., Velikov, M., 2022. Zeroing in on the expected returns of anomalies. J. Financ. Quant. Anal. 1–83.

Chen, A.Y., Zimmermann, T., 2020. Publication bias and the cross-section of stock returns. Rev. Asset Pricing Stud. 10 (2), 249–289.

Chen, A.Y., Zimmermann, T., 2022. Open source cross-sectional asset pricing. Crit. Financ. Rev. 11 (2), 207–264.

Chinco, A., Clark-Joseph, A.D., Ye, M., 2019. Sparse signals in the cross-section of returns. J. Financ. 74 (1), 449–492.

Chordia, T., Goyal, A., Saretto, A., 2020. Anomalies and false rejections. Rev. Financ. Stud. 33 (5), 2134–2179.
Corwin, S.A., Schultz, P., 2012. A simple way to estimate bid-ask spreads from daily high and low prices. J. Financ. 67 (2), 719–760.
Domingos, P., 2012. A few useful things to know about machine learning. Commun. ACM 55 (10), 78–87.
Dong, X., Li, Y., Rapach, D., Zhou, G., 2022. Anomalies and the expected market return. J. Financ. 77 (1), 639–681.
Fama, E.F., French, K.R., 1992. The cross-section of expected stock returns. J. Financ. 47 (2), 427–465.
Fama, E.F., French, K.R., 1993. Common risk factors in the returns on stocks and bonds. J. Financ. Econ. 33 (1), 3–56.
Fama, E.F., French, K.R., 1996. Multifactor explanations of asset pricing anomalies. J. Financ. 51 (1), 55–84.
Fama, E.F., French, K.R., 2008. Dissecting anomalies. J. Financ. 63 (4), 1653–1678.
Fama, E.F., French, K.R., 2015. A five-factor asset pricing model. J. Financ. Econ. 116 (1), 1–22.
Feng, G., Polson, N., Xu, J., 2020. Deep Learning in Characteristics-Sorted Factor Models. SSRN Scholarly Paper ID 3243683.
Freyberger, J., Neuhierl, A., Weber, M., 2020. Dissecting characteristics nonparametrically. Rev. Financ. Stud. 33 (5), 2326–2377.
Geertsema, P., Lu, H., 2023. Relative valuation with machine learning. J. Account. Res. 61 (1), 329–376.
Goyal, A., Bengio, Y., 2022. Inductive biases for deep learning of higher-level cognition. Proc. R. Soc. A 478 (2266), 20210068.
Green, J., Hand, J.R.M., Zhang, X.F., 2017. The characteristics that provide independent information about average U.S. Monthly stock returns. Rev. Financ. Stud. 30 (12), 4389–4436.
Gu, S., Kelly, B.T., Xiu, D., 2020. Empirical asset pricing via machine learning. Rev. Financ. Stud. 33 (5), 2223–2273.
Harvey, C.R., 2017. Presidential address: The scientific outlook in financial economics. J. Financ. 72 (4), 1399–1440.
Harvey, C.R., Liu, Y., 2020. False (and missed) discoveries in financial economics. J. Financ. 75 (5), 2503–2553.
Hasbrouck, J., 2009. Trading costs and returns for US equities: Estimating effective costs from daily data. J. Financ. 64 (3), 1445–1477.
Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
Haugen, R.A.R.A., Baker, N.L., 1996. Commonality in the determinants of expected stock returns. J. Financ. Econ. 41 (3), 401–439.
Holthausen, R.W., Larcker, D.F., 1992. The prediction of stock returns using financial statement information. J. Account. Econ. 15 (2), 373–411.
Hou, K., Xue, C., Zhang, L., 2015. Digesting anomalies: An investment approach. Rev. Financ. Stud. 28 (3), 650–705.
Israel, R., Kelly, B.T., Moskowitz, T.J., 2020. Can machines "learn" finance? J. Invest. Manag. 18 (2), 23–36.
Jegadeesh, N., Titman, S., 1993. Returns to buying winners and selling losers: Implications for stock market efficiency. J. Financ. 48 (1), 65–91.
Jensen, T.I., Kelly, B.T., Malamud, S., Pedersen, L.H., 2022. Machine Learning and the Implementable Efficient Frontier. Tech. Rep., Swiss Finance Institute Research Paper No. 22-63.
Kaniel, R., Lin, Z., Pelger, M., Van Nieuwerburgh, S., 2023. Machine-learning the skill of mutual fund managers. J. Financ. Econ. 150 (1), 94–138.
Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., 2017. LightGBM: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems, vol. 30, pp. 3146–3154.
Kelly, B.T., Xiu, D., 2023. Financial machine learning. Found. Trends textregistered Financ. 13 (3–4), 205–363.
Kozak, S., Nagel, S., Santosh, S., 2020. Shrinking the cross-section. J. Financ. Econ. 135 (2), 271–292.
Kyle, A.S., Obizhaeva, A.A., 2016. Market microstructure invariance: Empirical hypotheses. Econometrica 84 (4), 1345–1404.
Leippold, M., Wang, Q., Zhou, W., 2022. Machine learning in the Chinese stock market. J. Financ. Econ. 145 (2), 64–82.
Martin, I., Nagel, S., 2022. Market efficiency in the age of big data. J. Financ. Econ. 145 (1), 154–177.
McLean, R.D., Pontiff, J., 2016. Does academic research destroy stock return predictability? J. Financ. 71 (1), 5–32.
Moritz, B., Zimmermann, T., 2016. Tree-Based Conditional Portfolio Sorts: The Relation between Past and Future Stock Returns. Tech. rep., SSRN Working Paper.
Murray, S., Xiao, H., Xia, Y., 2024. Charting by machines. J. Financ. Econ. 153, 103791.
Newey, W.K., West, K.D., 1987. A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. Econometrica 55 (3), 703–708.
Novy-Marx, R., Velikov, M., 2016. A taxonomy of anomalies and their trading costs. Rev. Financ. Stud. 29 (1), 104–147.
Ou, J.A., Penman, S.H., 1989. Financial statement analysis and the prediction of stock returns. J. Account. Econ. 11 (4), 295–329.
Pástor, L., Stambaugh, R.F., 2003. Liquidity risk and expected stock returns. J. Political Econ. 111 (3), 642–685.
Rapach, D.E., Strauss, J.K., Zhou, G., 2013. International stock return predictability: What is the role of the United States? J. Financ. 68 (4), 1633–1662.
Welch, I., Goyal, A., 2008. A comprehensive look at the empirical performance of equity premium prediction. Rev. Financ. Stud. 21 (4), 1455–1508.
Yan, X.S., Zheng, L., 2017. Fundamental analysis and the cross-section of stock returns: A data-mining approach. Rev. Financ. Stud. 30 (4), 1382–1423.
Zhu, M., 2023. Evaluating the Efficacy of Multiple Testing Adjustments in Empirical Asset Pricing. SSRN Scholarly Paper.