Review of Quantitative Finance and Accounting (2023) 60:195–230
https://doi.org/10.1007/s11156-022-01099-z

ORIGINAL RESEARCH

Enhancing stock market anomalies with machine learning

Vitor Azevedo[1] · **Christopher Hoegner**[2]

Accepted: 24 July 2022 / Published online: 30 August 2022

Abstract
We examine the predictability of 299 capital market anomalies enhanced by 30 machine learning approaches and over 250 models in a dataset with more than 500 million firm-month anomaly observations. We find significant monthly (out-of-sample) returns of around 1.8–2.0%, and over 80% of the models yield returns equal to or larger than our linearly constructed baseline factor. For the best performing models, the risk-adjusted returns are significant across alternative asset pricing models, considering transaction costs with round-trip costs of up to 2% and including only anomalies after publication. Our results indicate that non-linear models can reveal market inefficiencies (mispricing) that are hard to conciliate with risk-based explanations.

Keywords Anomalies · Machine learning models · Efficient market hypothesis · Asset pricing models

JEL Classification G12 · G29 · M41

1 Introduction

Over the last decades, an unprecedented amount of stock market anomalies has been published by researchers in the field of asset pricing theory and factor investing.[1] To summarize the number of factors and anomalies published in journals, as of January 2019, there were over 400 signals documented in academic publications (Harvey and Liu 2019). Described in Cochrane's presidential address as factor zoo, the questions about "which characteristics really provide independent information about average returns" and how to overcome the "multidimensional challenge" remain an ongoing debate (Cochrane 2011).

[1] We follow other studies in using the terms anomalies, signals, characteristics, and factors interchangeably.

✉ Vitor Azevedo
vitor.azevedo@wiwi.uni-kl.de

Christopher Hoegner
christopher.hoegner@tum.de

[1] Department of Business Studies and Economics, Technical University Kaiserslautern, Gottlieb-Daimler-Straße 42, 67663 Kaiserslautern, Germany

[2] McKinsey & Company, Sophienstraße 26, 80333 Munich, Germany

The challenge of navigating the high-dimensional factor zoo is amplified by the issue of data *p*-hacking,[2] non-stationarity[3] and low chronological depth,[4] potential conditioning and biases from former literature,[5] and limited access to out-of-sample data. Traditional linear instruments such as ordinary least square regressions might not be able to overcome these issues. Thus, the problem of either selecting the correct subset of factors with real predictive power or cleverly combining the predictive power of the anomaly set remains an ongoing debate. The tremendous enhancements in machine learning and artificial intelligence, and the ability of smart algorithms to uncover complex relationships in large datasets, has the potential to overcome this issue.

As probably one of the most innovative and fastest developing computer technologies of the last decade, machine learning is predicted to fundamentally transform and disrupt entire industries. Referring to the latest Gartner Hype Cycle for Emerging Technologies (Panetta 2019), many innovation triggers are directly or indirectly linked to machine learning advances. We follow the definition of Murphy (2012, p. 1) and define machine learning as “a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data.” In contrast to conventional algorithms, where the computer receives input data and the specific program logic to calculate the result, machine learning algorithms receive both input and output data in the form of training samples to derive the program logic by themself. This ability to describe complex relationships through autonomous learning from experience (Samuel 1959) without explicitly coding any rules and exceptions is particularly suitable for the field of asset pricing.

Recently, researchers began to explore the potential of these algorithms in the context of stock market anomalies. Among these papers, Gu et al. (2020b) find that machine learning models can be used to create long-short strategies with positive and significant alphas. In the same way, other studies find promising results (e.g., Azevedo et al. 2022; Chen et al. 2020; Tobek and Hronec 2021). However, more recently, Avramov et al. (2022) find that the alphas of long-short strategies of anomalies enhanced by machine learning are attenuated after imposing economic restrictions.

Thus, with the proliferation of machine learning models in financial research, the literature lacks implications of these models in asset pricing literature, as well as evidence on how robust these models are conditional to the assumptions, approaches, and specifications. Furthermore, there is an ongoing debate on whether the results are driven by mispricing, risk, data dredging, or even limits-to-arbitrage. We reassess the predictability of 299 capital market anomalies enhanced by 30 machine learning approaches and over 250 models in a dataset with more than 500 million firm-month anomaly observations.

[2] In the following, we use the terms *p*-hacking, data snooping, data dredging, and data fishing interchangeably, referring to the excessive and negatively associated use of data mining.

[3] Financial time series are of non-stationary and high-dimensional nature (López de Prado 2018), meaning that the relations among variables are subject to change over time. Being additionally dependent on the time period, publication year (McLean and Pontiff 2016), and parameter settings such as portfolio weighting methodology (Fama 1998), empirical findings become less robust and potentially unreliable.

[4] In contrast to natural sciences and their ability to generate data through experiments on demand, financial time series have a low chronological depth, and it needs time-intensive observations to collect them (Harvey et al. 2016; Arnott et al. 2019).

[5] As Conrad et al. (2003) stated, data mining is inevitable in anomaly research, as scholars are already conditioned on former empirical studies’ results, leading to biased studies in the long run. Academics’ false incentive system of discouraging negative results known as file drawer problem amplifies this issue (Fanelli 2012, 2013; Rosenthal 1979; Harvey 2017).

Among the machine learning models tested, we include six different algorithms, seven feature reduction methods, and multiple variations of training approaches. The anomalies are used to predict a stock's next-month return, on which we form decile portfolios with the same standardized portfolio-sort strategy. This approach allows us an accurate comparison of the linear and the machine learning models and a robust estimation of the additional value of non-linear interaction effects.

Among the more than 250 models tested, we find that over 80% of the models tested yield equally good or better returns than our linearly constructed baseline factor, which achieved average monthly returns of 0.92%. For our top-performing models, we see significant monthly returns of 1.8–2.0%, indicating about 1.0% additional return above the linear benchmark. Among the best-performing algorithms are tree-based methods such as the GBM and DRF, as well as neural networks. Using hyperparameter optimization, feature interpretation methods, and the inclusion of transaction costs, data dredging seems not to be the underlying cause. Among the machine learning models that underperformed the linear models, the approaches either use a rolling window of only five years or use shrinkage methods. These results are an indication that, overall, most of the stock anomalies seem to have some predictive power and do not seem to add noise to the model. Furthermore, our results show that a rolling window of only five years is not enough to capture the importance of each anomaly in the model.

The positive and significant alphas from the machine learning are robust against transaction costs with round-trip costs of up to 2.4%, and remain stable across different parameter sets and when including anomalies only after publication, making it unlikely that the findings are merely a consequence of *p*-hacking. Furthermore, the returns are not explainable by common factor models, indicating mispricing effects and market inefficiencies within the stock market, and casting doubt on the current form of standard asset pricing models.

Our paper contributes to the current literature in several ways. First, our empirical analysis provides a replication study for classical anomaly research, reinforcing the current set of meta-studies (Jacobs and Müller 2020; McLean and Pontiff 2016) and replication studies (e.g., Kim and Lee 2014) and confirming the issue of *p*-hacking. In contrast to other scientific areas, in finance and accounting, the publication of replication studies is not encouraged or acknowledged particularly well (Harvey 2017).[6] However, replication studies are an "essential component of scientific methodology" (Dewald et al. 1986, p. 600), with out-of-sample data and modified assumptions being necessary to distinguish true causation from correlation. We contribute to the recent awareness of meta-studies in the field by testing a subset of 299 anomalies and verifying former findings.

Our second contribution lies in the broad assessment of machine learning capabilities in asset pricing and anomaly research (Chen et al. 2020; Gu et al. 2020b). Beyond the empirical analysis and former literature, we applied several tools to test the returns' robustness, with a positive outcome. Our hyperparameter optimization yields on average robust returns, thereby excluding parameter picking in the out-of-sample dataset as a cause for our findings. As an important extension to former literature, our post-publication model

[6] Among the reasons for the scarcity of replication studies, one can argue that the fact that the data is readily available (e.g., CRSP and Compustat), and the likelihood of an outright fraud being minimal make replication studies less interesting, and not commonly published in top journals (e.g., Harvey 2017). Furthermore, replication papers tend to receive not as many citations as the replicated studies. It can generate a back and forth with the authors of the replicated studies, and replication studies tend not to be awarded as a conventional paper in the tenure decisions.

variation only uses past data and methodology available at a specific time. This approach can strictly exclude any forward-looking bias both in terms of data and methodology, decreasing data dredging's likeliness and underlining the existence of real interaction effect causing the additional return.

As a third contribution, our study quantifies the value of non-linear effects among the factor zoo. As the results can neither be traced back to data dredging nor risk components, our findings cast significant doubt on the market's efficiency and current asset pricing models. Our findings support that the market can efficiently erase arbitrage opportunities from linear effects. However, more complex structures remain exposed to investors, which could increase our understanding of essential market mechanisms and the EMH and lead the way to a new generation of asset pricing models.

2 Related literature

Our paper also contributes to the growing field of the use of machine learning in asset pricing. Whereby Snow (2020, 2020a) describes how the overall portfolio construction in asset management benefits from various machine learning approaches, recent studies introduced more concrete application cases and empirical tests specifically in the context of anomaly-based trading strategies. Distinguished by algorithms, researchers tested among others approaches with shrinkage methods[7] (e.g., Han et al. 2018; Chinco et al. 2019; Ban et al. 2018), the class of Support Vector Machines (SVM) (e.g., Cao and Tay 2003; Matías and Reboredo 2012; Dunis et al. 2013; Ren et al. 2019; Huang et al. 2005; Trafalis and Ince 2000), as well as tree-based methods (e.g., Moritz and Zimmermann 2016; Tan et al. 2019; Qin et al. 2013; Basak et al. 2019; Bryzgalova et al. 2019) such as the Gradient Boosting Machine (GBM) or the Distributed Random Forest (DRF). Furthermore, a majority of papers applied various architectures of neural networks to predict future asset prices (e.g., Heaton et al. 2017; Abe and Nakayama 2018; Fischer and Krauss 2018; Feng et al. 2018; Zhang et al. 2020; White 1988; Dunis et al. 2008; Adeodato et al. 2011). Other, less widespread methodologies include Bayesian inference (e.g., Bodnar et al. 2017), autoencoders (e.g., Gu et al. 2020a), and Reinforcement learning (e.g., Moody and Saffell 2001; Zhang et al. 2020; Li et al. 2019).

In their innovative study, Gu et al. (2020b) compare diverse machine learning methods, including generalized linear methods, boosted regression trees, random forests, and neural networks, to estimate expected returns of stocks. The authors use information from 94 firms' characteristics, as well as eight macroeconomic predictors, in a sample from 1957 to 2016, and they find that tress and neural networks have the best performance. For instance, a zero investment long-short portfolio of deciles based on a neural network with three hidden layers (NN3) reports a monthly value-weighted alpha of 1.76% controlled for the Fama and French (2018) six-factor model. In the same vein, Tobek and Hronec (2021) test a similar setting with an international sample from 1963 to 2018. They find that 153 stock market anomalies enhanced by neural networks report a value-weighted alpha controlled for the Fama and French (2015) of 0.843%

[7] Most popular shrinkage approaches include ridge, lasso, and elastic net methods, which aim at reducing the number of coefficients in a regression to prevent overfitting and allow for the selection of the most important variables. Most of these methods use a regularization (i.e., applying a penalty term to the Loss function used in the model to limit the size, shrink, or even set coefficients equal to zero).

(t-statistic of 5.668). Chen et al. (2020) propose an approach that combines four neural networks to take advantage of conditioning information to estimate individual stock returns. The authors use 46 stock anomalies and 178 macroeconomic time series in a sample that spans from 1967 to 2016 as an input to estimate stock returns. Their model reports an annual Sharpe ratio of 2.6 compared to 1.7 for the linear special case of their model.

More recently, Avramov et al. (2022) reassess the results from Gu et al. (2020b) and Chen et al. (2020) and others by applying economic restrictions, such as excluding microcaps, distressed stocks, as well as episodes of high market volatility. In a sample from 1987 to 2017, they find that economic restrictions significantly weakens the profitability of machine learning. For instance, a Fama and French (2018) six-factor value-weighted alpha based on NN3 from Gu et al. (2020b) is 0.312% (t-statistic of 1.51) after excluding microcaps, while the alpha is 2.23% (t-statistic of 8.06) for the full sample.

Our paper sheds light on these results by analyzing the limits-to-arbitrage and different asset pricing models in a large range of machine learning approaches. Our findings are consistent with Gu et al. (2020b) who also explores a wide range of machine learning models with an emphasis on comparative analysis among the models. Our paper diverges from theirs by checking how robust these results are, addressing data dredging concerns, and analyzing the implications of these models in asset pricing. Although our empirical results are in line with the findings of Tobek and Hronec (2021) and Gu et al. (2020b), confirming significant benefits from using non-linear methods, by testing more than 250 models, we find that not all machine learning models outperform a baseline (linear) model. In other words, the superior performance of machine learning models can be conditional to the (ex-post) decisions of the models and parameters. Among alternative models and parameters that can drive the results, we find that, in general, dimensionality reduction models tend to underperform other non-linear models, which is an indication that machine learning models, such as GBM and DRF, can handle well the apparent high dimensionality of (299) anomalies. Furthermore, by analyzing alternative training and validation samples based on static windows, (five-year and ten-year) rolling windows, and expanding windows, we find that adding more recent data in the training and validation samples does not necessarily improve the results, which indicates that the patterns of the relation between anomalies and returns do not seem to change over time.

Finally, our paper provides insight on the findings from Avramov et al. (2022). By showing positive and significant alphas across eight factor models even using anomalies after publication, as well as by reporting that machine learning methods can be positive significant even with round-trip costs of up to 240 basis points, we find important evidence that limits-to-arbitrage cannot fully explain the strong profitability of machine learning methods.

In the following section, we present the data sources and the underlying methodology of our study. We present our empirical findings in twofold. First, we show the performance of both individual anomalies and the linear baseline factor (Sect. 4). These results serve as a replication study and benchmark for our more advanced machine learning models presented in Sect. 5. In Sect. 6, we discuss the empirical findings, advantages, and pitfalls of our approaches. In particular, we perform a model comparison, review the interpretation and parameter tuning in machine learning models, and test the results against common factor models. Section 7 summarizes the study's main findings, its implications, and an outlook on further research questions.

3 Data and methodology

Our empirical study consists of three steps. First, we calculate the raw signals for each firm-month observation. Based on this dataset, we apply a classical portfolio-sort analysis to examine each anomaly's performance individually and create a consolidated baseline factor as a linear benchmark, merging the original anomalies into one single meta signal. In step three, we use the anomaly dataset as input for our non-linear machine learning models. We test various algorithms, feature reduction methodologies, and training approaches to investigate the models' respective predictive power and their additional profit compared to our linear baseline model.

3.1 Data origin, preprocessing and anomaly construction

For the anomaly calculation, we follow to a great extent the open-source code published by Chen and Zimmermann (2020). We use data provided by the Wharton Research Data Service (WRDS), restricting ourselves to the U.S. market to ensure the highest data quality and make the study more comparable with the anomaly discovery's original publications. In particular, we use CRSP for both the monthly and daily pricing data and COMPUSTAT for the annual and quarterly fundamental data. Thereby, we follow the assumptions of Chen and Zimmermann (2020), namely applying a lag of six and three months for annual, respectively, quarterly accounting data.[8] We are conservative in our assumption of the reporting lag to avoid look-ahead bias. The recommendation and earnings-forecast-related anomalies are constructed with additional data provided by the Institutional Brokers Estimate System (I/B/E/S). A small fraction of anomalies requires more specific data, which includes the Sin Stock classification of Hong and Kacperczyk (2009), the government index from Gompers et al. (2003), and macroeconomic data from the Federal Reserve Bank of St . Louis (2020). We refer to the source code of Chen and Zimmermann (2020) for more details on the data gathering process. While we exclude some of the original data sources due to limited accessibility, we could calculate a set of 299 anomalies in total. Internet Appendix A gives readers a detailed overview of the anomalies evaluated in this study.

While we do not oppose any strict filters for prices or market cap during the data gathering process, we follow Griffin et al. (2010) by including only common equity (i.e., stocks with a WRDS share code of 10, 11, or 12) and excluding any stock that is not listed at the U.S. exchanges NYSE, NASDAQ or AMEX. However, as the individual anomaly evaluation follows the original author's proposal, some anomalies have specific selection filters applied during the portfolio construction process. Internet Appendix A includes a list of applied filters for each anomaly for this sample.

The anomalies we calculated are split into Accounting signals (175), Event signals (13), Analyst-based anomalies (18), Price-related signals (64), Trading (18), and other signals (11). This broad range has the potential to incorporate complex relationships and correlations on future returns. The machine learning algorithms' objective is to exploit these hidden patterns for profitable trading strategies.

[8] We follow Fama and French (1993) with a six-month lag for yearly accounting data. For quarterly data, we choose to use a three-month lag instead of the quarterly earnings announcement dates (rdq) because the coverage of this variable is relatively poor, particularly in the first years of our sample.

We calculate the anomaly set for every firm-month observation available, ranging from 1945 to 2019. However, our main analysis focuses only on the period from 1979 to 2019 (492 months of observations) to reduce the number of missing values in the training set while simultaneously ensuring a large enough and diverse dataset to find profitable patterns. Particularly, analyst recommendations and quarterly-based fundamental data often do not match our quality and quantity requirements before 1979. On average, we build our calculations on 5573 firms per month, with a peak in November 1997 with 7939 firms. Most stocks originate from the NASDAQ exchange, while in terms of market capitalization, the NYSE remains the most important exchange. This pre-filtering leads to a total of 2,742,141 unique firm-month observations, whereby on average, 197 out of 299 signals are available per observation, resulting in 542,346,630 data points or unique firm-month-anomaly observations both the baseline factor and machine learning models are trained. The size of this dataset is comparable with common meta-studies about stock anomalies such as McLean and Pontiff (2016) with 97 signals, Hou et al. (2020) investigating 447 anomalies, Green et al. (2017) calculating 94 anomalies, and Harvey et al. (2016) verifying 315 stock characteristics.

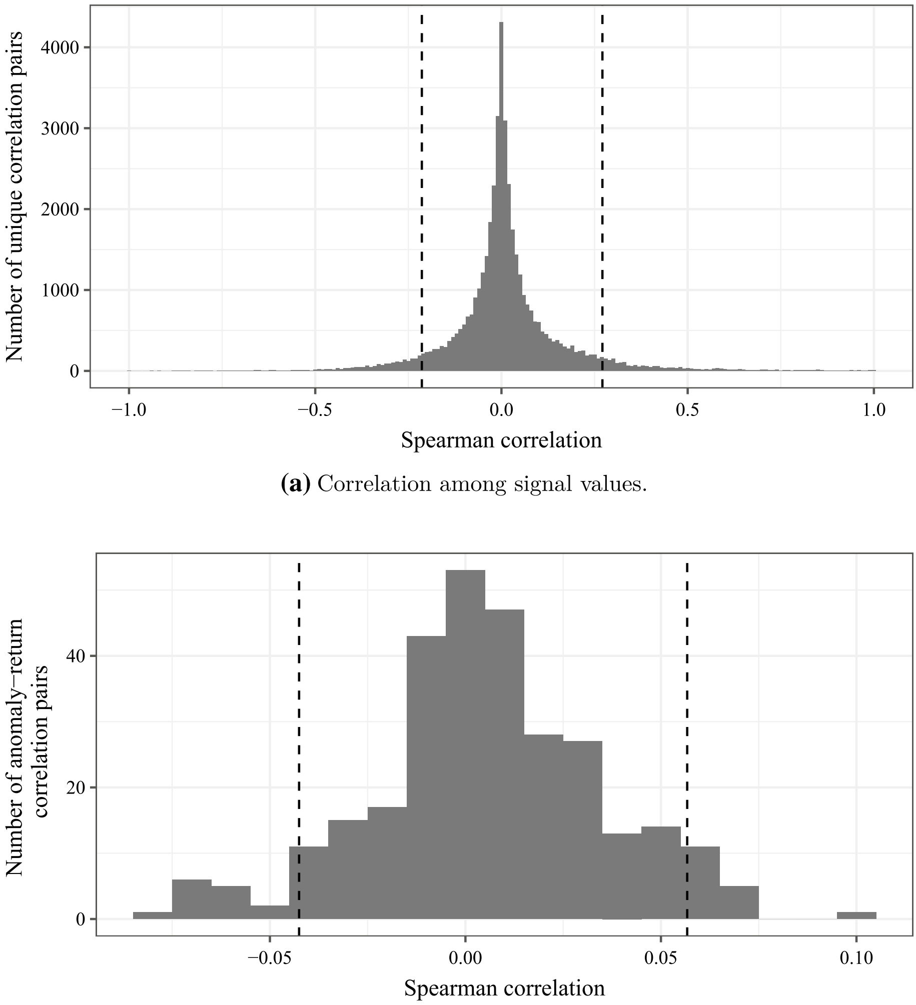
Figure 1a shows the paired Spearman correlation of our anomaly set, consisting of 44,551 unique correlation pairs. The graph demonstrates the high dimensionality of our dataset, as nearly 90% of anomalies are correlated only in the range between −0.21 and 0.27, which is in conformance with other meta-studies of anomalies (Jacobs and Müller 2016). Only a few signals are correlated strongly due to small variations of the same anomalies (e.g., quarterly and annually updated anomalies). We dispense to filter these anomalies as both the machine learning algorithms and the feature selection methods proposed should be able to handle this form of data. Similarly, about 90% of the anomalies have an absolute Spearman correlation with a future return of only 0.05, as depicted in Fig. 1b. While a single signal has limited expressiveness, the inclusion of hidden, non-linear structures with machine learning models could potentially provide significant outperformance.

Although we calculated absolute values for our anomalies, we use only percent-ranked anomaly values for portfolio construction and machine learning training. For each anomaly and month, we sort the values between 0 and 1. Transforming machine learning features in a preprocessing step to a common scale increases the performance of the algorithms (Singh and Singh 2020). While there exists a vast variety of data normalization approaches (Nayak et al. 2004), we followed the percent-ranked approach of Stambaugh and Yuan (2017) as it does not affect the portfolio-sort approach, which only cares about anomalies' absolute monthly rank. Additionally, it allows for a per-month rescaling, which ensures the prevention of any forward-looking bias, and facilitates handling missing values by replacing them with a median of 0.5.

3.2 Portfolio construction methodology and baseline factor composition

Similar to the paper of Chen and Zimmermann (2020), we test anomalies with a simple portfolio-sort strategy. For each month and anomaly, we create portfolios and calculate the spread of the long-short portfolio.

The stock-characteristic portfolio-sort approach is among the most common and dominant instruments to measure an anomalies' potential profitability and determine its statistical significance. Since the 1970s, many empirical studies such as Basu (1977) and Fama and French (1992) applied the portfolio-sort methodology. Multiple reasons explain this popularity: first, the approach's simplicity in terms of construction and interpretation.



(a) Correlation among signal values.

(b) Correlation of signal values with next month's stock return.

Fig. 1 **a** Illustrates the Spearman correlation among the 44,551 unique anomaly pairs, consolidated on the number of anomaly pairs per Spearman correlation value. As indicated by the graph, the dataset is rather symmetrically split between positive and negative correlations, with the 90% interval depicted as a dotted line ranging from −0.21 to +0.27. The low correlation among anomalies underlines the high dimensionality of the dataset. **b** Describes the distribution of Spearman correlations between each anomaly and a stock's next-month return. The graph indicates a relatively low correlation between individual anomalies and future return as well as a comparably symmetric distribution, with the 90% interval depicted as a dotted line ranging from −0.04 to +0.06

Second, its ability to handle a large and varying number of stocks in non-stationary and potentially infinite time series. Third, the capability to deal not only with linear but, more generally, monotonic relations between signals and return. We apply these capabilities by using portfolio sorts for our anomaly values and machine learning results, ranking every stock for each month into a fixed number of portfolios. We then calculate the spread of the long-short portfolio as the monthly return of our strategy and a *t*-statistic along with the time series of returns, which allows for an assessment of the strategy's statistical significance. A popular alternative methodology is the Fama-MacBeth cross-sectional regression approach (Fama and MacBeth 1973), which, due to its regression characteristic, is more vulnerable to outliers and thus microcaps effects. Furthermore, it is limited to linear relationships (Hou et al. 2020), making the non-parametric portfolio sort the preferable approach in our case.

We conduct the portfolio sort following the original authors' methodology as strictly as possible in terms of quantiles (number of portfolios), weighting (value-weight and equally-weight), holding period and rebalance frequency, starting month total examination period, and filtering of minimum prices and exchanges. However, we additionally assess anomalies based on a standardized approach. Thereby, we do not apply any price or exchange filter, adapt the anomalies to a monthly rebalancing and holding frequency, and conduct a decile portfolio sort. This standardized methodology allows a consistent comparison and benchmark with our baseline factor and is also an attempt to minimize *p*-hacking issues due to clever parameter picking. Additionally, a standardized guideline in portfolio construction is a pre-requirement for our machine learning-based portfolios. The standardized environment is calculated both for equally-weighted and value-weighted portfolios. Equally-weighted portfolios typically are hard to outperform, and most of the original publications are based on them. However, as noted by Fama (1998), equally-weighted portfolios give more weight to small stocks and are thus more negatively affected by the bad model problem (i.e., explaining the average return of small stocks). Therefore, the interpretation and decision-making during our empirical study are based on the standardized, value-weighted results. Internet Appendix C includes the results of both weighting methodologies for our machine learning models.

For each portfolio-anomaly-month combination, we not only calculate the return in the form of the long-short spread but also include the number of stocks positioned as long and short and the one-side turnover rate. The latter follows the definition of Hanauer and Windmueller (2019):

$$\text{One-sided turnover rate}_t = \frac{1}{2} \times \sum_{i=1}^{N_t} |w_{i,t} - w_{i,t-1}| \tag{1}$$

where t = Current month; i = Stock identifier; N_t = Number of stocks in dataset in month t; w_{it} = Weight of stock i in portfolio of month t.

The turnover rate is defined as the portfolio's percentage of stocks necessary to rebalance, indicating the potential trading costs associated with a live implementation. We use this indicator for a more practical evaluation of our models by including round-trip costs per strategy in Sect. 6.4.

Besides the portfolio-sort analysis for each individual anomaly, we calculate another signal, the baseline factor, as a linear combination of the anomaly set. Thereby, we orientate ourselves by the approach of Stambaugh and Yuan (2017), calculating the new signal as the arithmetic average of the percent-ranked 299 anomalies for each firm-month observation from 1979 to 2019. Furthermore, we require at least 100 non-missing anomalies for a firm each month to be included in the investment universe to ensure a diversified enough set of signals for the new factor. The baseline factor assessment is performed in the same way as evaluating the individual anomalies, using the portfolio-sort approach with the standardized methodology described above. We use the baseline factor as a benchmark tool to assess our machine learning models in Sect. 5.

3.3 Introduction into examined machine learning models

After having a baseline benchmark for our dataset, we use the same input data of 299 percent-ranked anomalies from 1979 to 2019 as a foundation for our machine learning algorithms. This subsection describes the overall approach, as well as the working mechanism of the selected models. For better reading comprehension, we only give an overview of the applied techniques here. The more detailed procedure is described, along with the presentation of the empirical findings.

We focus on investigating the additional performance of machine learning algorithms compared to traditional factor construction. In other words, we are interested in adapting the currently linear function $f(x)$ from our baseline factor into a non-linear function $g(x)$ using machine learning-based algorithms. To accomplish this, we restrict ourselves to stringently using only the same input data, namely the 299 anomalies per firm and month. Furthermore, to ensure the best comparability, we apply the same portfolio-sort approach with the standardized construction settings described above. The optimal sorting characteristics in this portfolio construction environment in terms of return would be to create decile portfolios based on a firm's next-month return. Therefore, we train our models to map future stock performance for each firm and month on the anomaly set, using the predicted returns as sorting criteria for the portfolio construction. This new, machine learning-based factor can then be benchmarked against existing literature.

This supervised regression approach can be further distinguished by testing various target variables. In the following, we test both the absolute next-month return[9] and the percent-ranked next-month return, with the latter having the advantage of being scaled and standardized in the same way the input variables are preprocessed. As with every supervised learning approach, we split our data into training and test samples. For the training sample from 1979 to 2002, we apply a 3-fold-cross-validation strategy for more robust metrics estimation.[10] However, the performance measurement was done with models trained on the full training set until 2002 but tested against an out-of-sample environment with data from 2003 onwards.

A common preprocessing step in data science is selecting only the most crucial input signals or applying a feature reduction method to the dataset to reduce any noise. This handling of the data's high-dimensionality might be beneficial to increasing the signal-to-noise ratio. We refer to Sect. 5.2 for a description of the applied algorithms.

All these approaches have in common being strictly static, meaning that they were not updated when new observations became available during backtesting. This training process allowed us a relatively conservative estimation focusing more on stationary patterns and reduced false-positives' risk due to the low number of models. However, particularly for practitioners, it might be interesting to update the model over time to further increase its performance by including the most recent data in the training process. We conduct this rolling training approach by retraining our models on new observations becoming available in the out-of-sample approach, avoiding any

[9] We use the term absolute next-month return to denote the next-month raw return, not compared to other assets' returns or referenced by a benchmark. The values of the absolute next-month return can be positive, negative, or zero.

[10] We use a three-fold validation within the test sample until 2002 to test stability and robustness of the model (i.e., training in rotation with two-thirds of the data and evaluating on the remaining one third and perform hyperparameter optimization).

forward-looking data bias. While the overall approach would even support monthly training, we restrict our study to yearly updates due to the computational effort, which can be considered enough for the overall testing of the hypothesis. Further research and practical implementations might increase training frequency.

As derived from the literature, among some of the best performing algorithms for machine learning, specifically for finance, are tree-based algorithms such as GBM, DRF, and eXtreme Gradient Boosting (XGBoost). We examine most approaches based on these algorithms and add the Generalized Linear Model (GLM) for comparison with a less-complex model. Thereby we enhance the capabilities of the popular open-source machine learning library H2O.ai (2020a). Additionally, as probably the most popular machine learning techniques, Sect. 5.4 shift the focus on neural networks' performance, whose architecture requires adjustments in construction and training processes. We use the popular Tensorflow (2020) framework developed by computer scientists of Google DeepMind. While an in-depth description of each applied machine learning algorithm would be out of this work's scope, we briefly introduce each algorithm in the following. We refer to the original documentation and source code for specific implementation details.

The GLM supports a variety of regression types for different distribution and link types. In its simplest variant, the output is a linear regression model. In our case, we use for both target variables the default identity link. The GBM, first described by Friedman (2001, 2002), is an ensemble method by building multiple decision trees. The boosting technique makes former weak learners, such as decision trees, strong and more robust (Zhou 2012). By weighting the individual learners' predictive power by their performance and focusing future learners on misclassified data, GBM sequentially refines its estimations. In our study, we used the implementation of (Hastie et al. 2001) as described in the H2O library documentation. The XGBoost algorithm originates from the mechanisms of the GBM. However, it has some adaptions, particularly concerning dropout regularization. We use DART, the dropout regularization for regression trees (Rashmi and Gilad-Bachrach 2015). As proposed by Breiman (2001), DRF, similarly to GBM, build on many decision trees using only a random fraction of the available dataset. For prediction, the average of all trees is used. This procedure is called bagging (Breiman 1996).

Neural networks differ from the tree-based algorithm as they aim to imitate the working of human brains by a set of neuron layers. Developed in the 1950s, recent achievements and performance enhancements led to an increasing number of different neural network architectures and application cases. We focus on two forms. First, we apply the Feedforward Neural Network (FNN) as the most basic architecture to attest its suitability in the context of stock market anomalies. We use both a smaller, tunnel-formed and a more extensive, larger architecture illustrated in Internet Appendix D. Another form of neural network particularly effective in time series analysis is the Recurrent Neural Network (RNN). It allows the processing of multiple past observations, creating a form of memory for upcoming predictions. Mainly the latter ability might be promising in the context of stock markets.

4 Creating a baseline: classical portfolio construction

The following section proceeds with the empirical results of our classical anomaly research methodology, replicating most findings of former meta-studies. The outcomes provide us a benchmark to assess our non-linear models in later parts of the study.

4.1 Individual anomaly returns derived from a long-short portfolio-sort strategy

Internet Appendix B lists each of our 299 underlying anomalies' portfolio performance with both the original sample and our standardized sample. As the latter is also used for our machine learning models, only this sample's results should be used for comparative analyses. While the original sample follows whenever possible the original's author construction details and period as outlined by Chen and Zimmermann (2020) (See Internet Appendix A for more details), our standardized sample strictly focuses on the timeframe from 1979 to 2019, using the same construction guideline as described in the previous section. Furthermore, we measure the change after publication as the mean return difference after anomaly publication based on the standardized sample.

While using the author's original sample period and anomaly construction methodology, the average monthly return per anomaly is around 0.53%, with mean *t*-statistics of 3.15. 70% of anomalies have a *t*-statistic of 1.96, and 47% above three, the minimum significance hurdle for new factor discoveries as Harvey et al. (2016) suggested. These results are not surprising, as most published anomalies have significant returns due to the academic journals' incentive system mentioned above.

These figures change once applying the same standardized portfolio construction framework across all anomalies over the full-time period from 1979 to 2019. The mean return of anomalies drops to 0.31% per month, becoming mostly insignificant with an average *t*-statistic of 1.38. Of the 299 anomalies examined, only 33% still overcome the *t*-statistic hurdle of 1.96. With the higher t-hurdle of three, only 46 anomalies remain significant. These results suggest widespread *p*-hacking in previous anomaly research and are in accordance with previous meta-studies findings. Due to the strong dependency of portfolio returns on construction settings such as weighting and rebalancing frequency, many anomaly findings might only be false-positive discoveries, resulting from a particular set of parameters that luckily had significant returns for the examined period. These anomalies weaken and disappear in a standardized environment. However, if genuinely reflecting either mispricing or risk, the returns should be more robust and less dependent on the construction settings.

Among the best performing anomalies in both samples in terms of return and statistical significance are the Earning Announcement Return (Chan et al. 1996), the Industry Return of Big Firms (Hou 2007), and the Firm-Age Momentum (Zhang 2006). Interestingly, all these three top-performing anomalies are members of the data category "Price." Consequently, we examined the average returns per category: while the categories "Accounting," "Event," "Analyst," and others are relatively equally performing in the range of 0.21% to 0.28% return per month, "Trading" underperforms, with only 0.15%. In contrast, "Price" anomalies are significantly exceeding other anomalies, with average monthly returns of 0.49%.

Besides the performance differences between original and standardized portfolio construction, we examined the publication bias of McLean and Pontiff (2016). With a median relative decrease in value-weighted returns of 74% of combined statistical bias and

Table 1 Baseline factor performance metrics

Strategy	Baseline factor			Mean of other signals		
	Return	*t*-statistic	Turnover rate	Return	*t*-statistic	Turnover rate
Full sample period						
Original settings				0.53	3.16	38.60
Equally-weighted	3.26	15.40	63.40	0.54	3.24	46.26
Value-weighted	1.95	9.32	79.15	0.31	1.39	47.66
Before 2003						
Original settings				0.55	3.11	38.70
Equally-weighted	4.14	13.12	62.06	0.71	3.08	47.02
Value-weighted	2.67	8.69	76.81	0.44	1.39	48.22
From 2003 on						
Original settings				0.17	1.00	41.99
Equally-weighted	2.02	9.03	65.31	0.32	1.33	45.20
Value-weighted	0.92	3.87	82.44	0.14	0.44	46.89

The table lists key performance indicators of the Baseline factor. We distinguish between equally-weighted and value-weighted portfolios, and separately analyze the full sample period as well as the periods before and after 2003. The performance measurement is referring to average monthly data, and both Return and Turnover rate are given in %

publication effect, the results are in line with the reported 58% of the original study and the 66% for the value-weighted study of Jacobs and Müller (2020). Out of 228 anomalies for which we could calculate pre-and post-publication-sample statistics, 160 signals faced decreasing returns, with an average absolute decline of 0.47% per month. These findings underline the anomalies' non-stationary character and illustrate how research can influence future anomaly returns. Those signals that previously suggested a profitable arbitrage mostly fade away due to investors' adaption towards exploiting these return spreads.

However, former research is usually focused on single, linear dependencies. By combining multiple firm characteristics into a single signal for portfolio construction, hidden structures might allow further profit opportunities. These patterns might be even more profitable for non-linear combinations machine learning algorithms are capable of uncovering. Therefore, in the next section, we construct a linear factor as a combination of all anomalies. This baseline factor indicates the potential benefits of a multi-anomaly-based strategy and serves as a benchmark for more advanced, non-linear machine learning models in Sect. 5.

4.2 Multi-anomaly-based baseline factor as a linear benchmark

The baseline factor is a linear combination of all available anomalies per firm-month observation. Averaging the percent-ranked values of the anomalies in a standardized sample aims to reduce individual signals' data mining issues, increasing both the returns' stability and reliability. In the case of both value-weighted and equally-weighted settings, we see a significant outperformance of the baseline factor, not only regarding averaged groups of anomalies by data category but also across the full spectrum of individual anomalies. Looking at the average monthly development as depicted in Table 1, we see that over the full period, the return of equally-weighted (value-weighted) is 3.26% (1.95%) per month. The statistical significance is on par with the best-performing individual anomalies, with *t*-statistics of 15.4 and 9.32 for the equally- and value-weighted portfolios. Compared to the value-weighted approach, the equally-weighted portfolio's outperformance is consistent with former research and serves as a further indicator for the bad model problem (Fama 1998).

For a more restrictive analysis of our baseline factor, we separately examined our standardized sample set from 2003 to 2019. In former research, 2003 marks a critical year (Green et al. 2017; Jacobs and Müller 2016). Out-of-sample and particularly post-publication returns of anomalies are significantly lower (McLean and Pontiff 2016), making many anomalies less profitable. This issue was empirically confirmed for the individual anomalies in the previous section. With 2003 as the mean publication year of our data sample, distinguishing a model's performance for the pre- and post-2003 range allows a more conservative and robust estimation of its performance. Particularly for the U.S. datasets, 2003 furthermore marks the first reporting year with the Sarbanes-Oxley act as well as new SEC filing changes in place, increasing the auditing and reporting quality significantly (Green et al. 2017). We follow the approach of two different time frames for evaluating the baseline factor and splitting criteria between training and testing data for our machine learning models.

We note that the one-sided turnover rate is higher than the average turnover rate for individual anomalies across different settings, potentially leading to higher transaction costs in practical implementation and lower profitability. This tendency is a consequence of the composition of many signals having a more varying influence on stock rankings, thus causing more volatile portfolio assignments. We examine in Sect. 6 the effect on potential transaction cost.

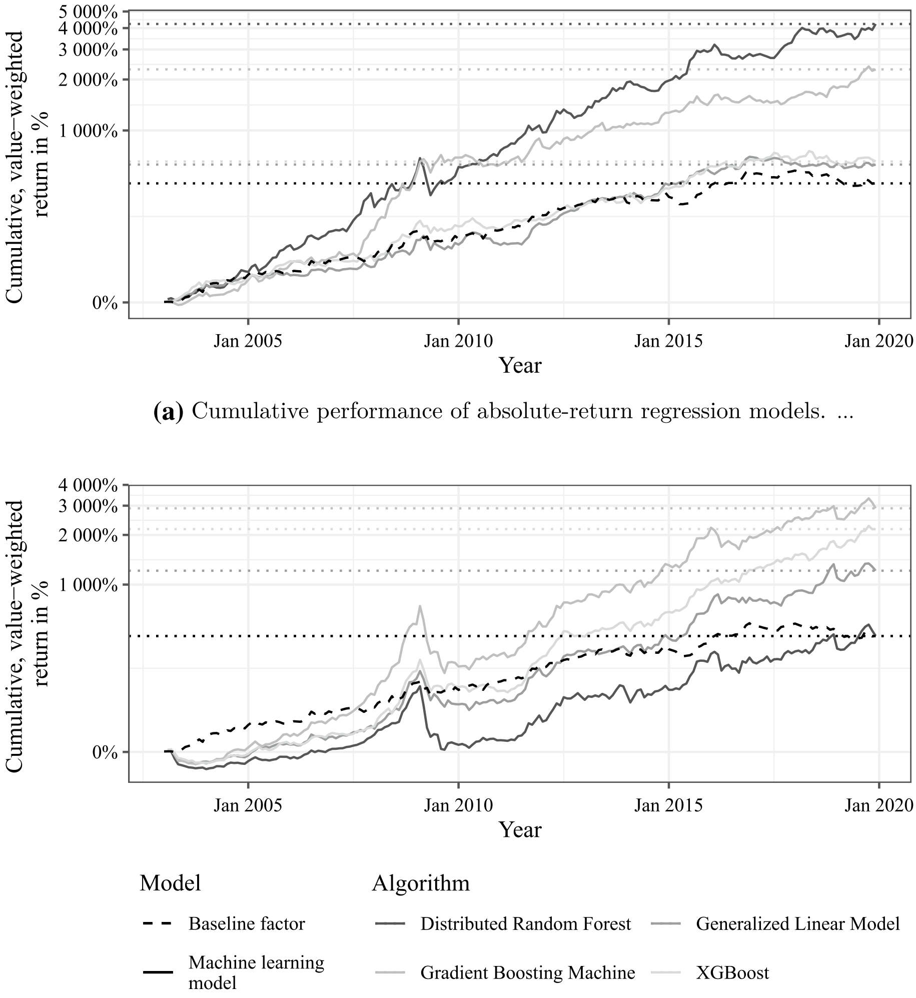
The significant returns of our baseline factor support the existence of potentially profitable relationships among anomalies. While the individual anomalies are vulnerable to data snooping and non-stationarity, the baseline factor reaches more robust returns in the pre- and post-2003 areas by leveraging the versatility of the full anomaly set. This outperformance is true for both the equally- and value-weighted portfolios.

In summary, we can conclude that the empirical results of the individual anomalies suggest data mining issues in former research and underline the strong non-stationary characteristics of financial time series data. However, when combining the anomalies by averaging the percent-ranked values in a standardized environment, we see significant performance improvements. We use these findings, particularly the value-weighted 0.92% [3.87] average monthly return of the baseline factor in the post-2003 period, as a benchmark and evaluation tool for our machine learning models constructed in the following section.

5 Portfolio construction with machine learning algorithms

5.1 Constructing portfolios based on forecasted future returns

As probably the most straightforward attempt to model the anomaly-return relations with machine learning algorithms, we train a set of different algorithms on the absolute



(a) Cumulative performance of absolute-return regression models. ...

(b) Cumulative performance of percent-ranked return regression models ...

Fig. 2 The graphs illustrate the cumulative performance of the four different machine learning algorithms in comparison to the Baseline factor during the post-2003 out-of-sample period. **a** Shows the value-weighted return for the regression approach based on the stocks' absolute next-month return, while **b** refers to the approach based on percent-ranked next-month returns

next-month return of a stock. For each firm-month observation, we thus have the formula $g(anomalies_{t,i}) \rightarrow r_{t+1,i}$. Thereby, $r_{t+1,i}$ is the absolute next-month return of a firm. To reduce the risk of p-hacking, we use the algorithms' default parameter without any hyperparameter tuning.[11] The number of trees for the DRF and GBM model is set to 100 as a balance between generalization ability and computational effort. According to former

[11] See H2O.ai (2020a); The documentation of the H2O.ai library provides further information about the default parameters.

Table 2 Model and portfolio metrics for absolute-return regression models

Model	Baseline	Generalized Linear Model (GLM)		Gradient Boosting Machine (GBM)		Distributed Random Forest (DRF)		XGBoost (XGBOOST)	
Sample		Cross-validation	Out-of-sample	Cross-validation	Out-of-sample	Cross-validation	Out-of-sample	Cross-validation	Out-of-sample
Model metrics									
Mean squared error		0.0390	0.0292	0.0376	0.0296	0.0388	0.0295	0.0371	0.0310
Residual mean squared error		0.1974	0.1709	0.1938	0.1720	0.1969	0.1719	0.1927	0.1760
R^2		0.0115	−0.0020	0.0473	−0.0146	0.0170	−0.0131	0.0582	−0.0625
Mean residual deviance		0.0390	0.0292	0.0376	0.0296	0.0388	0.0295	0.0371	0.0310
Mean absolute error		0.1172	0.1014	0.1152	0.1018	0.1169	0.1017	0.1143	0.1033
Equally-weighted									
Post-2003 performance	2.0182		3.1630		3.5897		3.0377		3.7410
Post-2003 significance	9.0263		12.7024		13.0326		11.1558		14.4962
Average turnover rate	32.6526		64.9735		63.8393		60.2210		61.8980
Annualized return	26.3450		44.3297		51.4271		42.0367		54.2657
Sharpe ratio	2.3814		3.6070		3.7829		3.1279		4.2604
Value-weighted									
Post-2003 performance	0.9182		1.0357		1.6803		2.0062		1.0535
Post-2003 significance	3.8663		4.3415		5.1740		5.5049		4.4637
Average turnover rate	41.2220		74.5310		78.1614		67.1744		70.2809
Annualized return	10.8447		12.4053		20.6437		24.9442		12.6517
Sharpe ratio	0.9229		1.0536		1.2879		1.3868		1.0861

The table above lists both model metrics and portfolio metrics for the training sample (e.g., cross-validation) and the test sample (e.g., out-of-sample) for all our H2O algorithms and the Baseline factor. We distinguish between equally-weighted and value-weighted portfolios. The target value the models are trained on is the absolute next-month return of a stock. Post-2003 performance and significance are referring to average monthly yields. Post-2003 performance, Average turnover rate, and Annualized return are given in %

research, this number seems to propose the biggest gains in performance for these types of tree-based algorithms (Probst and Boulesteix 2017).

Displayed in Fig. 2a, each model's performance over the out-of-sample period exceeds the returns of the baseline factor. Particularly noteworthy are the GBM and DRF, which significantly outperform every model, with 1.68% and respectively 2.01% of average monthly returns. Furthermore, the two models' Sharpe ratios of 1.29 and 1.39 are above the baseline benchmark ratio of 0.92. More performance indicators for both the equally-weighted and value-weighted construction settings are listed in Table 2.

As an alternative approach, we train the same algorithms on a different target value, namely the percent-ranked next-month return. For each firm-month observation, we thus have the formula $g(anomalies_{t,i}) \rightarrow rp_{t+1,i}$, with $rp_{t+1,i}$ being the monthly-ranked future return with values between 0 and 1. Through this approach, we train our models only on each stock's relative performance on which the portfolio-sort algorithm relies. A perfect forecast of both absolute and relative returns would thus yield the same portfolio returns. However, with a target value similarly scaled as the input anomalies, the algorithm might improve overall relationship modeling as it only has to predict the monthly distribution of returns across the stock universe, not the absolute values. That is particularly important for the application within neural networks we examine in Sect. 5.4, for which we follow the same procedure.

Again, all of our four different machine learning algorithms perform at least equal to the baseline factor. While the DRF performs relatively poorly, particularly compared with the previous approach, both the GLM and the XGBoost algorithm perform better than their respective counterpart in the absolute-return regressions. Particularly promising is the GBM, having average monthly returns of 1.89% with a Sharpe ratio of 1.01. More details are given in Table 3.

Noteworthy, we see a potentially systematic difference in the algorithms' working mechanisms, namely their ability to handle scaled and non-scaled values. Where the GLM and XGBoost algorithms performed particularly poorly in absolute-return-based regressions, the performance of both was significantly better in percent-ranked target values. Conversely, the random forest failed in the latter variant. The GBM seems to have the capability to handle both approaches sufficiently.

Besides the portfolio metrics, Table 2 and Table 3 furthermore list the most common model metrics. In contrast to the significant out-of-sample returns, the machine learning metrics are only mediocre. Focusing on the mean absolute error of the percent-ranked regressions, we see only slight improvements in contrast to a random algorithm, which would, by chance, achieve 0.25. Additionally, the best out-of-sample model metrics do not produce the highest returns in the same period. As we construct the portfolios on deciles of forecasted returns, the common metrics might be poorly suitable in our context. The algorithms' final performance in terms of strategy returns is dependent on an accurate assignment of stocks to the lowest and highest deciles and not necessarily on the most precise prediction of future returns. In the following, we give stronger attention to the out-of-sample portfolio metrics as an evaluation instrument. Furthermore, we would like to highlight that no single performance metric can fully assess a model's comprehensiveness. Instead, one should examine the overall picture with multiple indicators for a more robust estimation of the goodness of a models' predictions.

In summary, each of the machine learning algorithms applied performed at least on par with the traditionally constructed baseline factor. As the performance is significantly positive across different algorithms, target values, and portfolio metrics, it seems unlikely to be only a result of *p*-hacking but rather a consequence of non-linear effects within the

Table 3 Model and portfolio metrics for percent-ranked return regression models

Model	Baseline	Generalized linear model (GLM)		Gradient boosting machine (GBM)		Distributed random forest (DRF)		XGBoost (XGBOOST)	
Sample		Cross- validation	Out-of-sample	Cross-validation	Out-of-sample	Cross-validation	Out-of-sample	Cross-validation	Out-of-sample
Model metrics									
Mean squared error		0.0797	0.0819	0.0791	0.0817	0.0798	0.0816	0.0790	0.0822
Residual mean squared error		0.2823	0.2862	0.2813	0.2858	0.2825	0.2857	0.2811	0.2867
R^2		0.0299	0.0064	0.0365	0.0094	0.0283	0.0100	0.0382	0.0032
Mean residual deviance		0.0797	0.0819	0.0791	0.0817	0.0798	0.0816	0.0790	0.0822
Mean absolute error		0.2429	0.2464	0.2421	0.2462	0.2437	0.2465	0.2413	0.2465
Equally-weighted									
Post-2003 performance	2.0182		2.2567		3.2212		2.1036		3.1895
Post-2003 significance	9.0263		6.2871		9.3238		5.1187		11.1431
Average turnover rate	32.6526		61.3383		65.7040		56.5802		68.2377
Annualized return	26.3450		28.6967		44.2628		25.7671		44.3822
Sharpe ratio	2.3814		1.6199		2.5958		1.2703		3.1416
Value-weighted									
Post-2003 performance	0.9182		1.4175		1.8882		1.1097		1.6647
Post-2003 significance	3.8663		3.8060		4.2508		2.3311		4.9158
Average turnover rate	41.2220		65.8816		70.5010		58.6056		75.3800
Annualized return	10.8447		16.4503		22.2494		10.9132		20.2678
Sharpe ratio	0.9229		0.8949		1.0149		0.4645		1.2126

The table above lists both model metrics and portfolio metrics for the training sample (e.g., cross-validation) and the test sample (e.g., out-of-sample) for all our H2O algorithms and the Baseline factor. We distinguish between equally-weighted and value-weighted portfolios. The target value the models are trained on is the percent-ranked next-month return of a stock. Post-2003 performance and significance are referring to average monthly yields. Post-2003 performance, Average turnover rate, and Annualized return are given in %

anomaly set. In the following sections, we use the best-performing models of the two approaches, namely the absolute return-based DRF and the percent-ranked GBM, as our reference models for various training approaches, including different feature reduction and shrinkage methods, as well as for rolling training.

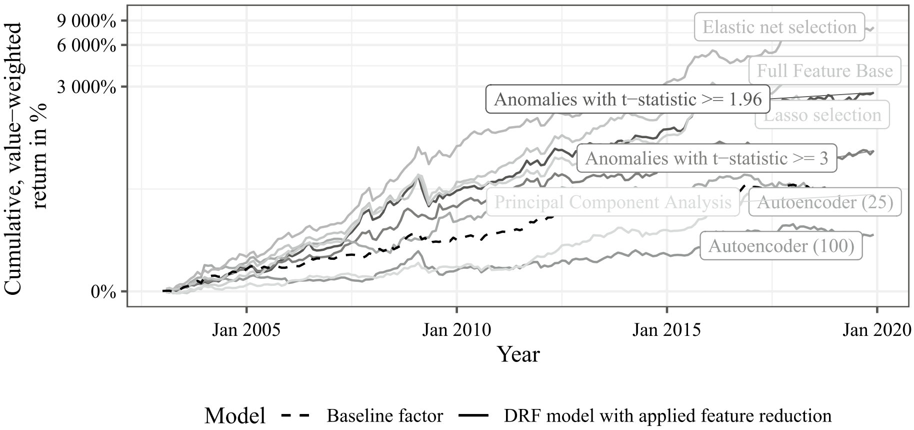
5.2 Reducing the high-dimensionality of the factor zoo with unsupervised learning and feature reduction algorithms

Currently, our models are trained on the full set of 299 percent-ranked signals. This high-dimensional data set may contain redundant data and strongly correlating values due to similarly constructed anomalies. A sophisticated reduction or combination of features into a lower-dimensional dataset could filter out unnecessary noise, further improving our algorithms' performance. In the following, we introduce a variety of common reduction methods and examine their performance impact on our models.

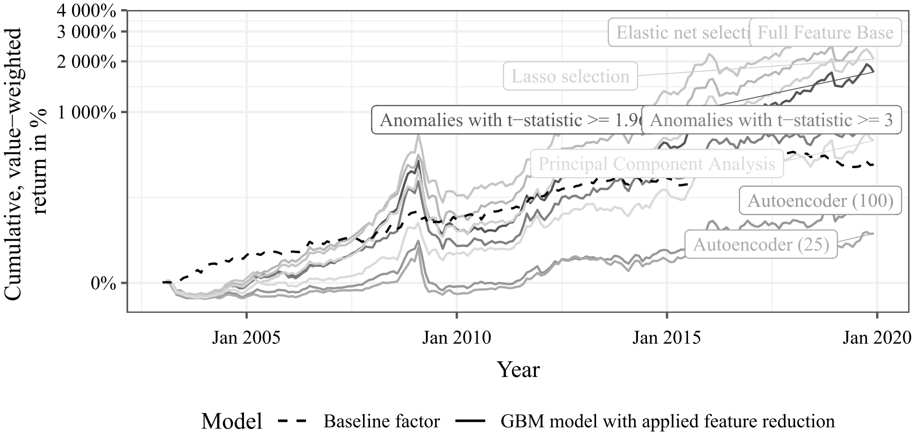
A Principal Component Analysis (PCA) belongs to the best-known feature reduction methodologies and aims to produce (a lower number of) linearly independent components representing the majority of variance of the original feature set. Autoencoders are a special case of Convolutional Neural Networks. Autoencoders, which are invented in the 1980s (Baldi 2012; Rumelhart et al. 1987), can reduce dimensionality by learning the internal representation of the dataset and compressing the input data into a lower dimension. The so-called bottleneck-layer we use in our two autoencoder experiments has 100 and 25 neurons, thus shrinking the feature's dimensionality by over 60% respectively 90% (see Internet Appendix D for more details). In contrast to the PCA, the autoencoders' results are non-linear combinations of the basic feature inputs, enabling the representation of more complex data structures that can be leveraged by our machine learning models. The lasso regression and elastic net selection follow common practice. The theory-derived selection of anomalies uses only past anomalies with *t*-statistics above 1.96 and 3, aiming to reduce the noise of non-important and insignificant signals.

The empirical results of our two reference models from the previous chapters with the inclusion of feature reduction techniques are depicted in Figure 3a and b and are relatively modest. Except for the elastic net, all feature reduction methods reduce our models' overall performance, in some cases, even below the baseline factor. The autoencoders and the PCA shrink average monthly results to values insignificant different from zero. Only the elastic net in the case of the DRF can increase the post-2003 performance of the static model. However, we do not discern a specific cause explaining these results and suspect it to be a false-positive, as we tested many different feature reduction approaches simultaneously. It seems that any feature reduction method tested fails in filtering out only unnecessary interferences. The poor results of the feature reduction based on past significance further indicate that, while an anomaly might individually be statistically not significant, it can contribute to the overall predictions through hidden joint effects. Consequently, removing any anomaly from the dataset can lead to a significant drop in performance.

In total, our approaches indicate that feature reduction might be less potent in the context of anomalies than suggested. While reducing the noise and dimensionality of the dataset, feature reduction might also weaken or eliminate significant signals, decreasing the model's overall performance. Since many machine learning models have in-built capabilities to handle high-dimensional datasets, feature reduction might be less critical than in classical regression approaches of former literature.



(a) Performance comparison of feature reduction methods on the absolute-return DRF regression ...



(b) Performance comparison of feature reduction methods on the percent-ranked GBM regression ...

Fig. 3 The graphs above illustrate the cumulative value-weighted return of our static trained reference models for various feature reduction methods. **a** Shows the performance of the absolute-return-based DRF model, while **b** Shows the same indicators for the percent-ranked-based GBM algorithm

5.3 Boosting performance with a dynamic, rolling machine learning model

So far, we have only used data from before 2003 to train our machine learning models. While that allowed a very critical and lower-bound-oriented estimation of the portfolio returns due to the weakening returns of the individual anomalies after this point in time, this approach also neglected the information value of more recent data available within the backtesting period. In particular, our models cannot yet exploit temporary relationships within the time series and the accompanying profit opportunities.

We adapt our current approach into a rolling training with interim updates of the algorithms' parameter to encounter this issue. We start again with 2003 as an out-of-sample period but retrain our models with the updated dataset every year. Although a monthly update of the model would be possible, it would exceed the available computing resources of this study, and an annual retrain frequency should be sufficient to estimate the performance potential for rolling machine learning models. However, it might be interesting for practitioners to optimize their models to the highest retraining frequency possible. Furthermore, we test multiple windows, namely a 5- and 10-years back-looking static frame, as well as a dynamically extending window across the full, up-to-prediction-date dataset. While more training data is generally positively correlated with a model's performance and ability to generalize, the relevance of older observations decreases due to the non-stationary character of the financial time series. Thus, a shorter time frame with less but more relevant data might increase performance.

We apply the different rolling training approaches to our two reference models, the DRF model for the absolute-return regression and the GBM for the percent-ranked return regression. With two reference models, 17 years of data, and three rolling training variations, this amounts to 102 trained machine learning models.

For the DRF model, a rolling training approach seems to decrease absolute performance. While remaining constant at about 2% per month for the extending rolling learning technique, for the shorter, static time frames, the model fails to exploit the anomalies' predictive power for profitable trading opportunities. In contrast, the GBM seems, at first sight, to enhance with a rolling 10-years window (2.12% per month). However, while having a higher cumulative return at the end of the out-of-sample period, most of it contributes to a peak in performance in 2018/2019. Because we train a total of 17 machine learning models for each model and training approach, the strong performance might be a false positive and should be treated with caution.

These ambiguous findings are confirmed when using the average monthly return and a paired *t*-test as an evaluation instrument. The DRF model is less robust on the training window than the GBM, with performance averaging between 0.35% and 2.02% per month. The 10-year and 5-year rolling window frame differences towards the static variant are highly significant with *t*-statistics above 2.56, whereby the extending window is not significantly different from the static one. Therefore, the rolling training seems to add no value to the case of the DRF. The GBM's rolling performance partly improves, lying between 1.58% and 2.11%, but the differences towards the static model in a paired *t*-test remain insignificant.

In summary, the findings of the rolling training approach are mixed. The DRF algorithm fails in applying a rolling strategy. While the inclusion of more recent data in the extending data frame seems not to harm the GBM's performance significantly, it also does not add significant value in terms of average monthly return. Furthermore, with the higher number of models, the risk of including false positives grows, potentially explaining the peak of the GBM in 2018/2019. The current approach seems not able to exploit significant returns from temporary structures as hypothesized. The fixed rolling window limits the amount of data per training, which might provide insufficient data for the algorithms to learn profitable patterns, particularly for the 5-year window. For the extending window, while including more data, the models might weight recent data not accurately and put too much focus on outdated observations.

5.4 Artificial neural network approaches

Thus far, we have focused on tree-based machine learning models. While they perform best-in-class for many applications, neural networks remain the most popular approach in machine learning. This section outlines the performance of both the standard FNN and a form of RNN in the context of stock market anomalies.

The FNN is among the most intuitive architectures, with one-way connected input and output layers and a variable amount of hidden layers and neurons. We test two different configurations: one smaller neural network with five hidden layers but a decreasing number of neurons per layer (110.821 parameters in total) and a larger variation with only three hidden layers but a higher number of neurons in total (256.400 parameters). More detailed empirical results and a comprehensive description of the architectures are attached in Internet Appendix C and D.

In total, we see a significant and continuous outperformance of the baseline factor for the static trained variant, with average monthly returns of 1.29% for the smaller and 1.68% for the larger model. These figures indicate that the larger model truly benefits from an increased number of neurons. In terms of the rolling models, performance has to be evaluated separately for the two models. While the smaller models seem to benefit from both the 10-year (2.01% per month) and extending window training (1.83% per month), the larger models' performance does not improve significantly. In contrast, the rolling 10-year window significantly reduces overall performance (1.26% per month). Due to the increasing need for observations to estimate the more extensive set of model parameters, a 10-year subset might not be sufficient for the parameter estimation in the training process. Future research might explore alternative approaches such as transfer learning to reduce the necessary data for rolling training.

Particularly successful in time- and order-dependent data such as time series analysis and natural language understanding are models with an RNN architecture. In contrast to FNNs, an RNN uses former time steps of observation in the prediction process, thereby creating a form of short-term memory to improve performance (Abiodun et al. 2018). In our case, we include twelve timesteps (e.g., each prediction is based on a 2D matrix with the twelve past observations of the past year of all 299 anomalies). As RNN suffers from vanishing or exploding backpropagated errors, we test a variant of RNN, namely the Long short-term memory (LSTM), which includes a memory cell to improve the models' capability in terms of long-term memory and efficient learning by holding errors constant (Hochreiter and Schmidhuber 1997).

Our findings indicate modest returns for models trained in the same environment as the FNNs. During most of the out-of-sample time, performance is below the other models, with average monthly returns of 1.48%. While, in theory, the model should be able to handle time-series data better, our test results contradict this hypothesis. Concerning the LSTM results, the poor performance of this approach compared to other machine learning approaches could be driven by the architecture chosen and the standard parameters of the model.[12] Furthermore, with the non-stationary character of our dataset, the high number of factors (299) relative to incorporated backward timesteps (12) (e.g., high dimensionality) might in our configuration not fully exploit the potential for backpropagation, requiring

[12] The architecture of our RNN model has five hidden layers, two LSTM layers, and two dropout layers. We use 32-128 neurons per layer and a total of 270657 parameters.

further finetuning. In short, the high dimensionality of data makes LSTM training more complex.

6 Discussion of findings

6.1 Performance comparison of machine learning models

In the previous chapter, we tested four different machine learning algorithms on two different target variables. We also used the two best approaches as references to test seven different feature reduction methods and three different rolling learning scenarios. Additionally, we calculated three different neural networks with static and rolling training variations. In total, Table 4 lists all 35 different models according to their overall performance and returns above the baseline factor. Other key performance indicators for each model are given in Internet Appendix C.

Although the best performing approach with monthly average returns of 2.33% [6.22] is the combination of static DRF with absolute return target and elastic net feature reduction, the result of a single model must be treated with caution, particularly in this case as no other feature reduction achieved any improvement of the overall outcomes. Since we have tested a large number of model combinations, there is the possibility of a false-positive despite high *t*-statistics due to multiple testing. That is particularly true for the rolling models, with each one consisting of 17 retrained models.

It is more beneficial to analyze the algorithms' overall distribution and approaches to get an idea of the models' potential and their range of returns. The best performing algorithms in our context are the GBM, the DRF as well as FNN. These findings are consistent with former literature (Gu et al. 2020b), identifying tree-based algorithms and neural networks as top-performers. While autoencoders and PCA lower the overall performance, the elastic net seems to add value in a single case; however, these findings appear less apparent and robust in this context. It seems that the algorithms can handle the high dimensionality directly by themselves, and any pre-processing reduction methods weaken essential signals. Rolling learning techniques seem to add value in the case of the GBM, while for other algorithms, an updated model seems to be defeated by static models. The GBM architecture might handle the different amounts of observations attributed to the rolling update better than other approaches.

In summary, out of our 35 models, 30 approaches show at least equal average monthly returns as our baseline factor for the period from 2003 to 2019. Moreover, 15 models show significantly higher returns above the 95% confidence interval. Excluding the poorly performing feature reduction methods, out of 21 models, over 90% are equal to or outperform the baseline factor with a mean return of 1.59%. Our best-performing models show both very high *t*-statistics and monthly returns of around 2%, more than twice the performance the baseline factor yields. These findings agree with the recent study of Gu et al. (2020b), who doubled the Sharpe ratio of standard linear models to 1.35 with neural networks. Similarly, in terms of Sharpe ratios, our results are within the range of 1.0 and 1.3.

It seems unlikely that these yields are merely the result of data dredging. First, the *t*-statistics are highly significant, both in terms of absolute returns and baseline improvement.

Table 4 Performance comparison and added-value of machine learning techniques

Model specifications				Performance		Baseline factor improvement	
Algorithm	Return target	Feature set	Rolling learning	Return in %	*t*-stat.	Add. return	*t*-stat.
DRF	Absolute	Elastic net	Static	2.33	6.22	1.42	3.86
GBM	Percent-ranked	Full	10y-rolling	2.12	4.23	1.20	2.49
DRF	Absolute	Full	Extending	2.02	4.93	1.11	2.79
DRF	Absolute	Full	Static	2.01	5.50	1.09	2.88
FNN	Percent-ranked	Full	10y-rolling	2.01	4.54	1.09	2.34
GBM	Percent-ranked	Full	Extending	1.97	4.13	1.06	2.30
GBM	Percent-ranked	Full	Static	1.89	4.25	0.97	2.33
FNN	Percent-ranked	Full	Extending	1.83	4.27	0.91	2.15
GBM	Percent-ranked	Elastic net	Static	1.80	4.22	0.88	2.15
DRF	Absolute	*t*-stat. > 3	Static	1.78	5.09	0.87	2.50
FNN (larger)	Percent-ranked	Full	Extending	1.76	4.13	0.85	2.00
GBM	Percent-ranked	Lasso	Static	1.72	3.93	0.81	1.95
GBM	Absolute	Full	Static	1.68	5.17	0.76	2.31
FNN (larger)	Percent-ranked	Full	Static	1.68	3.68	0.76	1.68
DRF	Absolute	Lasso	Static	1.67	4.65	0.76	2.13
XGBoost	Percent-ranked	Full	Static	1.66	4.92	0.75	2.08
GBM	Percent-ranked	*t*-stat. > 3	Static	1.64	3.74	0.72	1.74
GBM	Percent-ranked	Full	5y-rolling	1.58	3.23	0.66	1.33
RNN	Percent-ranked	Full	Static	1.48	3.19	0.57	1.25
GLM	Percent-ranked	Full	Static	1.42	3.81	0.50	1.41
GBM	Percent-ranked	*t*-stat. > 1.96	Static	1.31	2.85	0.40	0.92
DRF	Absolute	*t*-stat. > 1.96	Static	1.30	4.09	0.38	1.24
FNN	Percent-ranked	Full	Static	1.29	3.08	0.37	0.90
FNN (larger)	Percent-ranked	Full	10y-rolling	1.26	2.58	0.35	0.72
GBM	Percent-ranked	PCA	Static	1.19	2.83	0.27	0.67
DRF	Percent-ranked	Full	Static	1.11	2.33	0.19	0.43
XGBoost	Absolute	Full	Static	1.05	4.46	0.14	0.50
GLM	Absolute	Full	Static	1.04	4.34	0.12	0.43
DRF	Absolute	Full	10y-rolling	0.95	2.84	0.03	0.10
DRF	Absolute	PCA	Static	0.93	3.67	0.02	0.08
DRF	Absolute	Autoen. 25	Static	0.89	2.70	−0.02	−0.07
GBM	Percent-ranked	Autoen. 100	Static	0.80	1.81	−0.11	−0.26
GBM	Percent-ranked	Autoen. 25	Static	0.67	1.45	−0.24	−0.56
DRF	Absolute	Autoen. 100	Static	0.65	2.33	−0.26	−1.02
DRF	Absolute	Full	5y-rolling	0.35	1.00	−0.57	−1.62

The table above shows all tested models, differentiating between the type of model, the target value trained on, the feature input, and the rolling learning methodology. We separately state the algorithm's absolute performance and the additional value above our baseline factor. The latter is defined as the mean difference in monthly returns, with the *t*-statistic indicating the statistical significance that the difference is not zero, i.e., means of the time series predictions are different

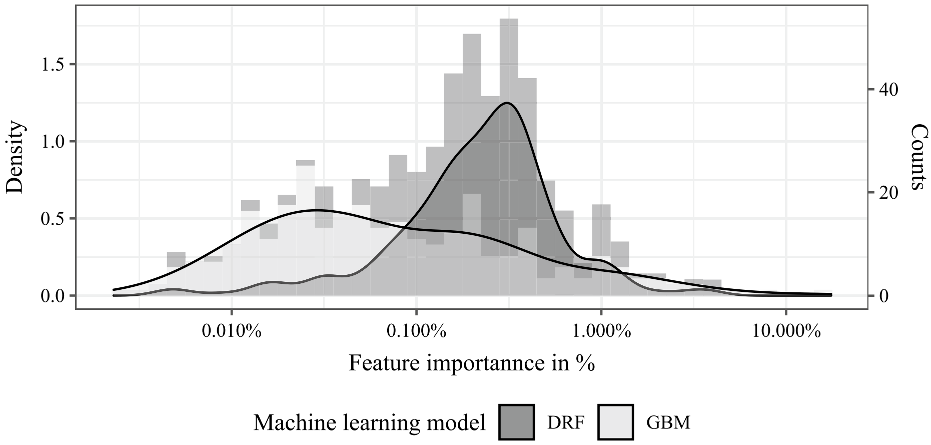


Fig. 4 Distribution of feature importance The figure shows the histogram and density plot with the distribution of variables according to their relative importance. We distinguish between our two reference models, the static DRF trained on absolute returns and the static, percent-ranked-trained GBM. While the curve illustrates the density of the feature importance, the histogram depicts the absolute count

Second, all of these approaches can handle the high-dimensional and non-linear data structures but differ in the specific underlying algorithm. Even if we face single false positives, as most models show significant gains, we can conclude that there are most likely arbitrage opportunities in the market or hidden risk components within the factor zoo that our models can exploit.

6.2 Model interpretation and feature importance

The results so far attest to a strong performance of the machine learning-based approaches concerning individual anomalies and the linearly constructed baseline factor. However, previous research about stock market anomalies was mostly concerned with linear models, as they appeal with ease of interpretation and testing. Highly complex models such as random forests with thousands of individual decision trees or neural networks with tens of thousands of parameters cannot keep up with this simplicity. This issue is a consequence of the model size and inevitably follows from its ability to learn complicated and nonlinear interactions within data structures that go beyond superficial if-else relationships. Researchers refer to this issue as the black box problem of Artificial Intelligence (Zednik 2019).

However, with the rise of machine learning, computer scientists began to develop some mechanisms to weaken this issue. This section focuses on the interpretation of tree-based algorithms by applying the relative importance of variables. The importance is determined by the variables selected for a split in a decision tree, as well as how they affect the squared error of the predictions.[13] Figure 4 depicts the distribution of variable importance across

[13] See H2O.ai (2020b); The documentation of the H2O.ai library provides further information about the exact implementation used for the feature importance calculation.

our two reference models, the static- and absolute-return-trained DRF and the percent-ranked-trained GBM.

As a consequence of the different boosting and bagging mechanisms inherent in the two algorithms, the distribution of feature importance varies substantially. GBM builds the trees sequentially, gradually weighting them to capture step-by-step all the subtleties of the data structure. This method leads to a higher weighting of a few variables, whereas the DRF uses averages, giving equal weight to the individual trees. Correspondingly, the weighting of the features is much more balanced across the factor zoo.

Examining the five most important anomalies for the predictions of the DRF and GBM, we see more similarities. Both the Short-term Reversal (MOM1M) and the Industry Return of Big Firms (INDRETBIG) seem rather important in the algorithms' return prediction. However, we see that the Idiosyncratic Risk (IDIOVOLAHT) is the most important variable for the GBM-approach, making the algorithm potentially less robust. These results are in accordance with the overall distribution of the importance depicted in Fig. 4.

It is noticeable that the most critical features regularly fall into the data category "price." Examining the overall distribution of the share of each data category on the feature importance, the results reveal that accounting and price features are by far the most essential components for our models' predictions. This circumstance naturally follows from the dataset, consisting mainly of accounting (more than 50%) and price (around 25%) anomalies. The DRF follows this distribution, slightly overweighting the importance of price anomalies. This behavior stands in contrast to the GBM, which weights price signals more than twice as high, and reduces the proportion of accounting signals to the same extent. The difference in the assessment of feature importance is perhaps a major driver of GBM's strong performance, as price signals were among the most stable and profitable anomalies, as demonstrated in Sect. 4. The algorithm seems to identify this circumstance correctly and adapted its weights accordingly.

In summary, we see significant differences in how the model weight features. While we can analyze remarkable characteristics to trace some of the working mechanisms behind the training process, interpretability is limited due to the "black box" characteristics of current algorithms. This issue is not only limited to finance but is a general challenge in machine learning-related tasks.

6.3 The impact of hyperparameter tuning on machine learning performance

A common task in a data science pipeline and particularly in machine learning models is estimating parameters belonging to the specific algorithm. These parameters include the number of trees and learning rates for tree-based algorithms and the number of neurons and hidden layers in neural network architectures. Depending on the model, there exists a wide range of possible parameter combinations, and purely analytical estimation of the best combination based on the underlying dataset is usually not possible. A common way to tune these parameters is to sample different combinations, train them via cross-validation, and select the best-performing one.

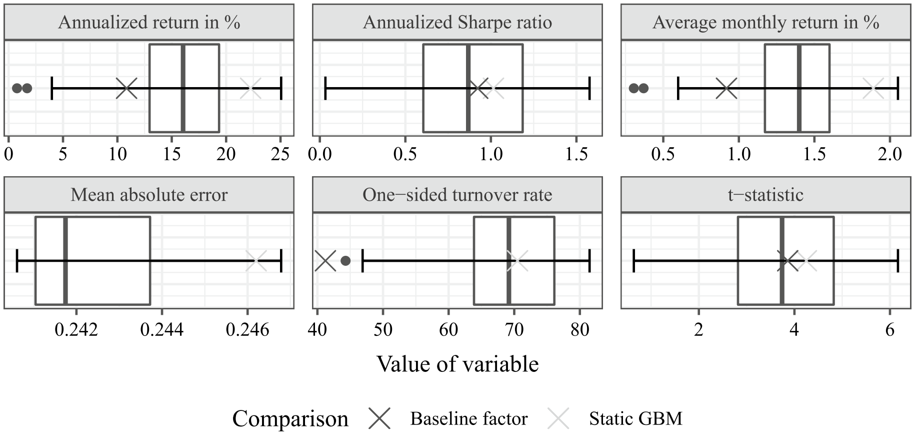


Fig. 5 Hyperparameter tuning for GBM model The figure illustrates the distribution of the key performance indicators of the 64 GBM models involved in the process of hyperparameter optimization. We focus on the value-weighted returns of the respective portfolios for model assessment. Besides, the performance of both the original static GBM model and the baseline factor is plotted (excluding the mean absolute error for the latter one as the metric does not apply to the linearly constructed factor)

Thus far, we use the same default parameters for our models, which offers a favorable combination regarding resource consumption and model complexity. However, this approach may pose a higher false-positive risk only attributed to a luckily, nevertheless, randomly selected parameter set. For a more robust estimation of our machine learning algorithms' profit span, we optimize the percent-ranked GBM model through hyperparameter tuning. We train the algorithm with 64 different combinations of essential parameters: the number of trees, the learning rate, the maximum depth, the sample rate of rows, and the column sample rate.[14] The boxplots in Fig. 5 illustrate the range of different key performance indicators achieved by the varying combinations.

The empirical findings suggest significant differences depending on the chosen parameters. Depending on the parameter set, the 90% confidence interval of returns ranges from 0.67% to 2.01%, with *t*-statistics between 1.36 and 5.89. However, apart from some rare outliers, the median consistently ranks above the baseline factor over the set of performance indicators, suggesting an overall superiority of the GBM. While we conducted this analysis exemplary for the GBM algorithm, we do not expect significant differences for hyperparameter optimization of other algorithms.

In conclusion, hyperparameter tuning underlines the value of machine learning models in the factor zoo. First, the probability of being only a result of data dredging further decreases, as it seems that the additional profit through non-linear algorithms is not a consequence of cleverly chosen parameters but universally applicable. Second, the upper-bound limit of monthly returns might be higher than estimated since it is possible to further tune an algorithms' parameter or a neural networks' architecture for optimal performance.

[14] See H2O.ai (2020a); The documentation of the H2O.ai library provides further information about available tuning parameters per algorithm and their effect on the predictions.

Table 5 Top-10 performing models and Baseline factor comparison with regard to round-trip costs

Model name	Return [*t*-stat.]	Turnover rate	Round-trip costs
DRF.RETURN.ELASTICNET.STATIC	2.33% [6.22]	66.62 %	2.4 %
FNN.PERCENTRANK.FULL.ROLLING10Y	2.01% [4.54]	58.62 %	1.94 %
DRF.RETURN.FULL.STATIC	2.01% [5.5]	67.17 %	1.92 %
GBM.PERCENTRANK.FULL.ROLLING10Y	2.12% [4.23]	60.94 %	1.87 %
DRF.RETURN.FULL.ROLLINGEXT	2.02% [4.93]	67.07 %	1.82 %
DRF.RETURN.LIKELY.STATIC	1.78% [5.09]	61.98 %	1.77 %
FNN.PERCENTRANK.FULL.ROLLINGEXT	1.83% [4.27]	62.53 %	1.58 %
GBM.PERCENTRANK.FULL.ROLLINGEXT	1.97% [4.13]	67.08 %	1.54 %
FNN_WIDE.PERCENTRANK.FULL.ROLLINGEXT	1.76% [4.13]	63.48 %	1.46 %
GBM.PERCENTRANK.FULL.STATIC	1.89% [4.25]	70.5 %	1.44 %
Baseline factor	0.92 [3.87]	41.22 %	1.1 %

The table above lists the top performing machine learning models according to their respective round-trip costs. Additionally, the Baseline factor's key performance indicators are given as lower benchmark. Return in % and *t*-statistic refer to average monthly portfolio yields, and the one-sided turnover rate indicates the amount of rebalancing per month

6.4 Turnover rate and break-even transaction cost considerations

Our current machine learning models are all optimized to predict the next-month stock returns, which would lead to maximum long-short spreads. However, our model's true profitability furthermore depends on the transaction costs associated with it when being executed. These costs are related to the relative amount of rebalancing per month, referred to as the one-sided turnover rate.

The empirical results indicate a significant (*t*-statistics > 5) positive relationship between monthly returns and average turnover rate. As we optimize for return predictability with monthly portfolio realignment, the long-short portfolios are always constructed according to the predicted maximum spread, regardless of the previous portfolio state and potential transaction costs. While we thereby maximize mean absolute return, the increased turnover rate might lead to an overall decreasing portfolio return, as indicated by the findings of Novy-Marx and Velikov (2016), who associated lower profitability with higher turnover rates.

To address this issue and gain a more comprehensive understanding of the strategies' profitability in a real implementation, we calculate the round-trip costs as an indicator for the upper bound of transaction costs. An estimation of the maximum amount of allowable transaction costs for a profitable strategy allows us to analyze whether the higher return of our high-turnover strategies compensates for higher transaction costs. We calculate these round-trip costs as in Grundy and Martin (2001), Barroso and Santa-Clara (2015), and Hanauer and Windmueller (2019) using a Z-score at the 5% significance level:

$$\text{Round-trip costs}_{\alpha=5\%} = \left(1 - \frac{1.96}{T_S}\right) \times \frac{\bar{\mu}_S}{T\bar{O}_S} \tag{2}$$

where $S =$ Portfolio strategy S; $T_S = t$-statistic of strategy S; $\mu_S =$ Average monthly return of strategy S; $TO_S =$ One-sided turnover rate of strategy S.

The findings outlined in Table 5 indicate that our best-performing machine learning models, as well as our two static and most conservative models with the full feature set, outperform the baseline factor not only in terms of absolute monthly return but additionally compensate their higher turnover rate. With round-trip costs between 1.4% and 2.4%, these strategies allow realistic transaction costs while remaining profitable. These findings are a robust indicator that no *p*-hacking took place and strengthen our hypothesis that non-linear patterns in the factor zoo might offer rich profit opportunities.

6.5 Avoiding methodological forward-looking bias with post-publication feature inclusion

Until now, our models have always used the complete set of 299 anomalies available. The feature reduction methods also reduced the amount of input data based on the complete observation. However, our anomalies' average publication date is 2003 (i.e., our models currently use anomalies whose underlying calculation methodology has not yet been published at the point in time of prediction). Although there was no direct forward-looking bias concerning the observations (which was always used ex-ante), there might be a form of forward-looking methodological bias in current research. To counteract this potential data-mining bias and to observe the size of this effect, we train our percent-ranked GBM model in a rolling training fashion based on post-publication anomalies. For each yearly retrained model from 2003 to 2019, we use only those anomalies that have already been published.

The first models used 106 anomalies starting in 2003, exceeding 250 features in 2012. We see strong growth in accounting anomalies, tripling in total over the post-2003 period. In terms of performance figures, we compare it with the baseline and static reference model and the extending rolling learning GBM from Sect. 5.3 due to its training similarities. In contrast to the reference variants of the GBM, performance only slightly decreases, and the difference is not statistically significant. With average monthly returns of 1.85% and a *t*-statistic above four, we can further reduce the risk of data snooping within our empirical research. The strict pre-filtering of unpublished anomalies prevents both data and methodological forward-looking in our training process. This approach bolsters our previous findings and underlines that our returns truly originate from either a risk component or a mispricing effect that caused the spread of our models' long-short portfolios.

6.6 Risk or return? Testing machine learning returns against common factor models

With the probability of emerging purely from data snooping being relatively low, the question arises whether our models' average monthly returns of 1.8–2% are a consequence of a hidden risk component or an indicator of market inefficiencies and irrational investor behavior. A typical instrument to test the risk component hypothesis is to test the models' return against common factor models. If these factor models with their respective loadings can satisfactorily explain the models' return (i.e., only have insignificant alphas in linear regressions), then the models' performance is fully contributable to these risk components.

To ensure a robust assessment of whether underlying risk components are attributable to the models' return, we test our two reference model (DRF and GBM) and the post-publication GBM against the most common factor models: the CAPM (Sharpe 1964; Lintner 1965; Mossin 1966), the Carhart (1997) Four-Factor model, the Fama-French Three- and

Table 6 Factor model comparison

Factor model	Value-weighted			Equally-weighted		
	DRF (absolute, static, full)	GBM (percent-ranked, static, full)	GBM (percent-ranked, rolling, post-publication)	DRF (absolute, static, full)	GBM (percent-ranked, static, full)	GBM (percent-ranked, rolling, post-publication)
Carhart four-factor model	1.96 %	2.32 %	2.16 %	3 %	3.52 %	3.44 %
	[6.19]	[7.59]	[6.76]	[10.95]	[12.06]	[11.38]
Capital asset pricing model	2.15 %	2.53 %	2.43 %	2.99 %	3.6 %	3.55 %
	[5.94]	[6.75]	[6.08]	[10.71]	[11.52]	[10.9]
DHS	1.82 %	1.52 %	1.39 %	3.08 %	3.01 %	2.94 %
	[5.21]	[4.33]	[3.75]	[10.88]	[10.55]	[10.11]
DMRS	1.68 %	1.55 %	1.35 %	3.22 %	3.09 %	2.91 %
	[4.24]	[3.37]	[2.91]	[10.61]	[8.35]	[7.71]
Fama-French five-factor model	2.06 %	2.06 %	1.9 %	3.06 %	3.29 %	3.18 %
	[5.62]	[5.93]	[5.16]	[10.89]	[10.94]	[10.2]
Fama-French three-factor model	2.09 %	2.48 %	2.33 %	2.97 %	3.6 %	3.53 %
	[5.91]	[6.84]	[6.14]	[10.8]	[11.67]	[10.98]
Q Factor	1.68 %	1.51 %	1.35 %	3.03 %	2.89 %	2.77 %
	[4.49]	[5.12]	[4.26]	[10.22]	[10.31]	[9.71]
Stambaugh and Yuan mispricing factor	1.79 %	1.86 %	1.48 %	3.37 %	3.18 %	2.93 %
	[4.69]	[5.39]	[4.22]	[10.74]	[10.49]	[9.6]

The table above lists both alpha values and t-statistics when applying our best-performing machine learning models against the most common factor models. We focus on our two reference models and the post-publication GBM. The regressions are calculated within the out-of-sample period from 2003 to 2019. However, for some factor models, we faced missing public factor loadings for recent years. Namely, the mispricing factor is limited to December 2016, the DMRS model is limited to June 2019, and the DHS model is limited to December 2018

Five-Factor models (Fama and French 1993, 2015), the mispricing factor of Stambaugh and Yuan (2017), as well as against the more recent Q-Factor model (Hou et al. 2015), the Behavioral Factor (DHS) of Daniel et al. (2020a) and the Daniel et al. (2020b) (DMRS) Factor. We utilize the respective factor loadings published as time series by the original authors. The empirical results are depicted in Table 6.

The results underline that no factor model can satisfactorily explain the results of the machine learning models. Both the equally-weighted and value-weighted portfolios have alphas between 2.9% and 3.6% respectively, 1.4% and 2.5%. Significant values in the form of *t*-statistics are consistently greater than 3. It can also be observed that the alphas are more pronounced for the GBM model than for the DRF.

Derived from these results, it seems that the risk components of standard factor models cannot explain our machine learning models' returns. Consequently, and underlined by the rather unlikely case of *p*-hacking, any attempt to explain the returns will inevitably point to potential market inefficiencies and mispricing issues or shortcomings in asset pricing models (Joint Hypothesis). Arbitrage opportunities usually disappear through investors' trading adaptions. In the case of machine learning algorithms, these relationships might have been too complex and hidden in the factor zoo such that investors were not yet able to exploit them. That could also explain why the profits are relatively non-stationary, as our best-performing models were statically trained with pre-2003 data, trading up to 2019 without updates. In the future, and with a more widespread application of machine learning algorithms, rolling techniques might become increasingly important to retain constant profits by exploiting temporary limited, non-linear patterns.

7 Conclusion

Our study replicated many findings of former meta-studies. It showed that most anomaly returns mitigate and disappear when using a standardized framework across the full factor zoo instead of the authors' original construction settings. This tendency underlines the widespread issue of data dredging in anomaly research. The empirically confirmed post-publication effect of McLean and Pontiff (2016), combined with the non-stationarity of the time series, makes replication studies across different timeframes particularly important. When using a combined baseline factor as the average of each firm-month observation, monthly long-short spreads of 0.92% with high significance are achievable. These findings lead to the hypothesis that while individual anomalies often do not provide significant returns, a combined approach might yield robust earnings.

As the primary insight of our study, using machine learning algorithms as an advancement of the baseline factor indicates significant potential by leveraging non-linear structures of the factor zoo. Compared to our baseline factor, most machine learning models show clear superiority in performance. With monthly returns of up to 2.0% and *t*-statistics greater than three, these models are strong indicators that there are anomalies in the market challenging the EMH and that the effects on returns are not only linear and can rely on the influence of multiple anomalies. We estimate the effect of the non-linear components and interaction effects to be up to 1.0%, but they are conditional to the machine learning models and the parameters used.

In addition to the recent study of Gu et al. (2020b), we encountered the issue of data dredging through conservative testing periods (i.e., the post-2003 period), the inclusion of transaction costs in the form of round-trip costs, as well as by hyperparameter optimization to estimate the effect of parameter-picking. Notably, we test a rolling model variation, including only features after publication, to encounter any methodological forward-looking bias. By showing the robustness of the model superiority across different parameter sets, training variations, and algorithms, the probability of the findings being merely a result of data dredging is low. Common factor models seem not able to explain the models' returns. These findings lead to the presumption that the returns are actually due to market inefficiencies and mispricing. As the exploited patterns are less transparent to investors, the returns are less likely to be arbitraged away by professional investors, casting doubt on the semi-strong form of the EMH and current asset pricing models.

For researchers, the algorithms' results show that linear models might not be able to handle the high-dimensionality of the factor zoo sufficiently. While standard linear regression is attractive due to its straightforward interpretation, machine learning models seem to outperform them using statistical significances and returns. Interaction effects among anomalies might provide further insights into the working mechanism of the stock market. It might enable a broader understanding of the Joint Hypothesis and EMH, which are challenged by our trading strategy's significant returns. For practitioners active in the quantitative asset management industry, our machine learning models' empirical results on anomalies might provide new opportunities for profitable trading strategies. With linear relations mostly arbitraged away by investors, non-linear relations and interaction effects might offer new profit opportunities. It also casts a new light on robot advisor services that emerged in recent years.

Due to the enormous diversity of data, algorithms, and training variations in the finance and computer science field, our study cannot test every possible approach. There are thousands of variations on how to train the models, and parameter-tuning exponentially increases this number. However, by using 299 anomalies covering a significant part of the factor zoo and testing only the most widespread algorithms, we believe that we have made an accurate estimate of the potential of the current state of machine learning in finance.

A rather general issue of machine learning is its interpretability and black box character, partly owed to its complexity. Explainable AI is among the top research areas of computer scientists and needs significantly more attention to develop new approaches to evaluating a model's performance and goodness. We encountered this issue in our study by evaluating many different algorithms to increase robustness and use feature importance as an interpretation tool for tree-based algorithms. For future studies, alternative methods to reduce the black-box issue could be applied, such as Local Interpretable Model-Agnostic Explanations (LIME).

The factor zoo remains challenging to handle. The non-stationary character of financial and economic time series, combined with the data's limited chronological depth, complicates any analysis. However, the results of our machine learning models underline the capabilities of smart algorithms in this field. While being difficult to interpret, the ability to go beyond linear relationships enables new insights for researchers and tangible profit opportunities for practitioners. With further advancements in the algorithms, higher computing capacity, and a larger set of literature and research, machine learning might answer how to handle the factor zoo, broaden our understanding of the EMH, and trigger a new generation of asset pricing models.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11156-022-01099-z.

Acknowledgements We acknowledge the anonymous referees, as well as participants from the French Finance Association (AFFI) annual meeting for helpful comments. Part of this research was undertaken while Vitor Azevedo worked at Technical University Munich. The authors also thank the National Council for Scientific and Technological Development (CNPq) and the Science without Borders Program for financial support, including a research scholarship.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Abe M, Nakayama H (2018) Deep learning for forecasting stock returns in the cross-section. arXiv q-fin. ST:1–12

Abiodun OI, Jantan A, Omolara AE, Dada KV, Mohamed NA, Arshad H (2018) State-of-the-art in artificial neural network applications: a survey. Heliyon 4(11):1–41

Adeodato PJ, Arnaud AL, Vasconcelos GC, Cunha RC, Monteiro DS (2011) MLP ensembles improve long term prediction accuracy over single networks. Int J Forecast 27(3):661–671

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

Avramov D, Cheng S, Metzker L (2022) Machine learning versus economic restrictions: evidence from stock return predictability. Manag Sci 1–89 (Forthcoming)

Azevedo V, Kaiser S, Müller S (2022) Stock market anomalies and machine learning across the globe. SSRN Electron J 1–48

Baldi P (2012) Autoencoders, unsupervised learning, and deep architectures. In JMLR: Workshop and Conference Proceedings 27:37–50

Ban G-Y, El Karoui N, Lim AEB (2018) Machine learning and portfolio optimization. Manag Sci 64(3):1136–1154

Barroso P, Santa-Clara P (2015) Momentum has its moments. J Financ Econ 116(1):111–120

Basak S, Kar S, Saha S, Khaidem L, Dey SR (2019) Predicting the direction of stock market prices using tree-based classifiers. North Am J Econ Financ 47:552–567

Basu S (1977) Investment performance of common stocks in relation to their price-earnings ratios: a test of the efficient market hypothesis. J Financ 32(3):663–682

Bodnar T, Mazur S, Okhrin Y (2017) Bayesian estimation of the global minimum variance portfolio. Eur J Oper Res 256(1):292–307

Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Bryzgalova S, Pelger M, Zhu J (2019) Forest through the trees: building cross-sections of stock returns. SSRN Electron J 1–62

Cao L, Tay F (2003) Support vector machine with adaptive parameters in financial time series forecasting. IEEE Trans Neural Networks 14(6):1506–1518

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

Chan LKC, Jegadeesh N, Lakonishok J (1996) Momentum strategies. J Financ 51(5):1681–1713

Chen AY, Zimmermann T (2020) Open source cross-sectional asset pricing. SSRN Electron J 1–44

Chen L, Pelger M, Zhu J (2020) Deep learning in asset pricing. arXiv q-fin.ST:1–63

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

Cochrane J (2011) Presidential address: discount rates. J Financ 66(4):1047–1108 (**Publisher: American Finance Association**)

Conrad J, Cooper M, Kaul G (2003) Value versus glamour. J Financ 58(5):1969–1995

Daniel K, Hirshleifer D, Sun L (2020a) Short- and long-horizon behavioral factors. Rev Financ Stud 33(4):1673–1736

Daniel K, Mota L, Rottke S, Santos T (2020b) The cross-section of risk and returns. Rev Financ Stud 33(5):1927–1979

Dewald WG, Thursby JG, Anderson RG (1986) Replication in empirical economics: the Journal of money, credit and banking project. Am Econ Rev 76(4):587–603

Dunis CL, Laws J, Evans B (2008) Trading futures spread portfolios: applications of higher order and recurrent networks. Eur J Financ 14(6):503–521

Dunis CL, Likothanassis SD, Karathanasopoulos AS, Sermpinis GS, Theofilatos KA (2013) A hybrid genetic algorithm-support vector machine approach in the task of forecasting and trading. J Asset Manag 14(1):52–71

Fama EF (1998) Market efficiency, long-term returns, and behavioral finance. J Financ Econ 283–306

Fama EF, French KR (1992) The cross-section of expected stock returns. J Financ 47(2):427–465

Fama EF, French KR (1993) Common risk factors in the returns on stocks and bonds. J Financ Econ 33(1):3–56

Fama EF, French KR (2015) A five-factor asset pricing model. J Financ Econ 116(1):1–22

Fama EF, French KR (2018) Choosing factors. J Financ Econ 128(2):234–252

Fama EF, MacBeth JD (1973) Risk, return, and equilibrium: empirical tests. J Polit Econ 81(3):607–636 (**Publisher: University of Chicago Press**)

Fanelli D (2012) Negative results are disappearing from most disciplines and countries. Scientometrics 90(3):891–904

Fanelli D (2013) Positive results receive more citations, but only in some disciplines. Scientometrics 94(2):701–709

Federal Reserve Bank of St . Louis (2020) Economic research

Feng G, Polson N, Xu J (2018) Deep learning factor alpha. SSRN Electron J 1–44

Fischer T, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. Eur J Oper Res 270(2):654–669

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232

Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378

Gompers P, Ishii J, Metrick A (2003) Corporate governance and equity prices. Q J Econ 118(1):107–156

Green J, Hand JRM, Zhang XF (2017) The characteristics that provide independent information about average U.S. monthly stock returns. Rev Financ Stud 30(12):4389–4436

Griffin JM, Kelly PJ, Nardari F (2010) Do market efficiency measures yield correct inferences? A comparison of developed and emerging markets. Rev Financ Stud 23(8):3225–3277 (**Publisher: Oxford Academic**)

Grundy BD, Martin JS (2001) Understanding the nature of the risks and the source of the rewards to momentum investing. Rev Financ Stud 14(1):29–78

Gu S, Kelly B, Xiu D (2020a) Autoencoder asset pricing models. J Econom 1–22

Gu S, Kelly B, Xiu D (2020b) Empirical asset pricing via machine learning. Rev Financ Stud 33(5):2223–2273

H2O.ai (2020a) H2O.ai programming library

H2O.ai (2020b) Variable importance - H2o.ai documentation

Han Y, He A, Rapach D, Zhou G (2018) How many firm characteristics drive US stock returns? SSRN Electron J 1–36

Hanauer MX, Windmueller S (2019) Enhanced momentum strategies. SSRN Electron J 1–73

Harvey CR (2017) Presidential address: the scientific outlook in financial economics: Scientific outlook in finance. J Financ 72(4):1399–1440

Harvey CR, Liu Y (2019) A census of the factor zoo. SSRN Electron J 1–7

Harvey CR, Liu Y, Zhu H (2016) ... and the cross-section of expected returns. Rev Financ Stud 29(1):5–68 (**Publisher: Oxford Academic**)

Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning, 12th edn. Springer Series in Statistics, Springer

Heaton JB, Polson NG, Witte JH (2017) Deep learning for finance: deep portfolios. Appl Stoch Model Bus Ind 33(1):3–12

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Hong H, Kacperczyk M (2009) The price of sin: the effects of social norms on markets. J Financ Econ 93(1):15–36

Hou K (2007) Industry information diffusion and the lead-lag effect in stock returns. Rev Financ Stud 20(4):1113–1138

Hou K, Xue C, Zhang L (2015) Digesting anomalies: an investment approach. Rev Financ Stud 28(3):650–705

Hou K, Xue C, Zhang L (2020) Replicating anomalies. Rev Financ Stud 33(5):2019–2133

Huang W, Nakamori Y, Wang S-Y (2005) Forecasting stock market movement direction with support vector machine. Comput Oper Res 32(10):2513–2522

Jacobs H, Müller S (2016) ...and nothing else matters? On the dimensionality and predictability of international stock returns. SSRN Electron J 1–44

Jacobs H, Müller S (2020) Anomalies across the globe: Once public, no longer existent? J Financ Econ 135(1):213–230

Kim S, Lee C (2014) Implementability of trading strategies based on accounting information: Piotroski (2000) revisited. Eur Account Rev 23(4):553–558

Li Y, Zheng W, Zheng Z (2019) Deep robust reinforcement learning for practical algorithmic trading. IEEE Access 7:108014–108022

Lintner J (1965) The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. Rev Econ Stat 47(1):13–37

López de Prado MM (2018) Advances in financial machine learning. Wiley, New Jersey

Matías JM, Reboredo JC (2012) Forecasting performance of nonlinear models for intraday stock returns. J Forecast 31(2):172–188

McLean RD, Pontiff J (2016) Does academic research destroy stock return predictability? J Financ 71(1):5–32

Moody J, Saffell M (2001) Learning to trade via direct reinforcement. IEEE Trans Neural Networks 12(4):875–889

Moritz B, Zimmermann T (2016) Tree-based conditional portfolio sorts: the relation between past and future stock returns. SSRN Electron J 1–81

Mossin J (1966) Equilibrium in a capital asset market. Econometrica 34(4):768–783

Murphy KP (2012) Machine learning: a probabilistic perspective. MIT Press, Cambridge, MA (**Adaptive computation and machine learning series**)

Nayak SC, Misra BB, Behera HS (2004) Impact of data normalization on stock index forecasting. Int J Comput Inf Syst Ind Manag Appl 6:257–269

Novy-Marx R, Velikov M (2016) A taxonomy of anomalies and their trading costs. Rev Financ Stud 29(1):104–147

Panetta K (2019) 5 trends appear on the gartner hype cycle for emerging technologies, 2019

Probst P, Boulesteix A-L (2017) To tune or not to tune the number of trees in random forest. J Mach Learn Res 18(1):6673–6690

Qin Q, Wang Q-G, Li J, Ge SS (2013) Linear and nonlinear trading models with gradient boosted random forests and application to Singapore stock market. J Intell Learn Syst Appl 05(01):1–10

Rashmi KV, Gilad-Bachrach R (2015) DART: Dropouts meet multiple additive regression trees. In JMLR: Workshop and Conference Proceedings 38:489–497

Ren R, Wu DD, Liu T (2019) Forecasting stock market movement direction using sentiment analysis and support vector machine. IEEE Syst J 13(1):760–770

Rosenthal R (1979) The file drawer problem and tolerance for null results. Psychol Bull 86(3):638–641

Rumelhart DE, Hinton GE, Williams RJ (1987) Learning internal representations by error propagation. In: Parallel distributed processing: explorations in the microstructure of cognition: foundations. MIT Press, pp 318–362

Samuel AL (1959) Some studies in machine learning using the game of checkers. IBM J Res Dev 3(3):210–229 (**Conference Name: IBM Journal of Research and Development**)

Sharpe WF (1964) Capital asset prices: a theory of market equilibrium under conditions of risk. J Financ 19(3):425–442

Singh D, Singh B (2020) Investigating the impact of data normalization on classification performance. Appl Soft Comput 97(Part B):105524

Snow D (2020a) Machine learning in asset management: part 2: portfolio construction-weight optimization. J Financ Data Sci. Publisher: Institutional Investor Journals Umbrella, pp 17–24

Snow D (2020) Machine learning in asset management-Part 1: Portfolio construction-Trading strategies. J Financ Data Sci 2(1):10–23 (**Publisher: Institutional Investor Journals Umbrella**)

Stambaugh RF, Yuan Y (2017) Mispricing factors. Rev Financ Stud 30(4):1270–1315

Tan Z, Yan Z, Zhu G (2019) Stock selection with random forest: an exploitation of excess return in the Chinese stock market. Heliyon 5(8):e02310

Tensorflow (2020) TensorFlow. Library Catalog: www.tensorflow.org

Tobek O, Hronec M (2021) Does it pay to follow anomalies research? Machine learning approach with international evidence. J Financ Markets 56:1–47

Trafalis T, Ince H (2000) Support vector machine for regression and applications to financial forecasting. In: Neural computing: new challenges and perspectives for the new millennium, vol 6. IEEE, Como, Italy, pp 348–353

White (1988) Economic prediction using neural networks: the case of IBM daily stock returns. In: Proceedings of 1993 IEEE international conference on neural networks (ICNN '93), vol 2. IEEE, San Diego, CA, USA, pp 451–458

Zednik C (2019) Solving the black box problem: a normative framework for explainable artificial intelligence. Philos Technol 1–24

Zhang XF (2006) Information uncertainty and stock returns. J Financ 61(1):105–137

Zhang Z, Zohren S, Stephen R (2020) Deep reinforcement learning for trading. J Financ Data Sci 2(2):25–40 (**Publisher: Institutional Investor Journals Umbrella**)

Zhou Z-H (2012) Ensemble methods: foundations and algorithms. Chapman & Hall/CRC machine learning & pattern recognition series. Taylor & Francis, Boca Raton, FL