Man vs Machine: Can AI outperform humans in stock-picking?

Paulo Roberto Fonteles Guimarães[1], Herbert Kimura[1]

Abstract

AI's potential to replace human roles is a prominent concern and jobs in financial markets industry may be specially jeopardized. This study compared out-of-sample performance of Machine-Learning models and human managers in equity portfolio management, which involves deciphering the complex, non-linear and time-varying individual stocks return-generating process. Through a walk-forward validation scheme and grid-search, various models and ensembles were used to forecast benchmark-relative returns with 46,312 monthly observations and nearly one hundred factor zoo features. Rules-based, Risk-Parity, and Mean-Tracking-Error portfolios backtested these models ability to generate economic value, producing significant out-of-sample alphas (t-stats above 3.0 or 4.0) and better risk-adjusted returns than equity funds composites, including Best-in-Class. Although not conclusive proof of human replacement, these findings highlight Machine-Learning relevance in equity portfolio management.

Keywords: factor zoo, asset pricing, machine-learning, return prediction, emerging markets, risk-parity, portfolio optimization

1. Introduction

Recently, Artificial Intelligence (AI) popularity has been soaring. In fact, according to Google Trends, the topic has reached unprecedented worldwide levels of interest since the start of 2022, peaking at the beginning of 2023, with ChatGPT's release.

Email addresses: pauloguimaraes871@gmail.com (Paulo Roberto Fonteles Guimarães), herbert.kimura@gmail.com (Herbert Kimura)

Despite the economic benefits usually associated, such as productivity increase, these tools also cause many concerns to the general public, a phenomenon denominated AI anxiety, with job replacement being one of the main factors (Li and Huang, 2020). In a recent survey of Ernst & Young, 75% of respondents fear that AI can impact their income or make their job obsolete, a concept called Fear of Being Obsolete - FOBO (Diasio et al., 2023).

In this sense, the World Economic Forum evaluated the exposure of different industries to AI, by analysing 19,000 individual tasks across 867 occupations, finding that financial services and capital markets have the highest exposure to automation (WEF, 2023).

To assess FOBO in equity portfolio management, this study applied ML models to predict individual stock returns in Brazil. Methods included simple linear models, penalized linear regressions, tree-based methods, neural networks, and ensembles of models. The fitting process involved splitting the sample into three sets: training, validation, and testing, hence simulating out-of-sample behavior. Hyperparameter tuning followed grid-search in a walk-forward scheme. Finally, 93 features were built based on common themes from asset pricing literature, besides sector dummies.

To assess ML models' out-of-sample economic value ML compared to human managers, many long-only portfolio construction methods were applied, covering simple heuristics, risk-parity, Mean-Tracking Error constrained and unconstrained optimization (MTOc and MTOunc), together with three covariance matrix estimators: sample, Principal Component Analysis (PCA) and Shrinkage. The latter two are useful for handling high-dimensional assets collections (Ledoit and Wolf, 2022); (Coqueret and Milhau, 2014); (Hsu et al., 2022). ML portfolios were compared with the average active equity fund manager, top quartile manager (Best-in-Class) and Ibovespa (IBOV), Brazil's most-used cap-weighted equity benchmark.

Results indicate that ML portfolios produced superior risk-adjusted returns compared to benchmarks, as assessed by various metrics. Nevertheless, ML's out-of-sample performance can still face periods of poor predictability. Moreover, being ML fundamentally different, hence possibly low correlated to human managers, combining both approaches could enhance diversification and information ratios.

The paper is structured as follows. In the next section, we briefly discuss the asset pricing literature related to our study. Then, we thoroughly present the methods and data used in the empirical analysis. The main results of the

study are examined, and finally, we conclude the paper with implications for theory and practice.

2. Asset Pricing Literature

Asset pricing is a well-explored topic, dating back to the early 19th century (Jokanovic and Poitras, 2001). The cornerstone model is the Capital Asset Pricing Model (CAPM), which linearly describes equilibrium expected future returns based on exposure to systematic market risk. CAPM and related equilibrium models are based on the fair game hypothesis, according to which strategies based on an information set cannot produce expected returns in excess of equilibrium expected returns conditional to this set (Fama, 1970).

Despite its relevance, the CAPM fails to explain many findings, as subsequent empirical research explored numerous variables and found many statistically significant relationships with expected returns (Green et al., 2013; Maiti, 2019), suggesting that the cross-section of expected returns is more complex than the CAPM posits. These empirical findings, which challenge the assumption that the market portfolio is the sole driver of returns, are often denominated anomalies (Coqueret and Guida, 2020).

In fact, hundreds of anomalies have been documented, collectively defining the "factor zoo" (Cochrane, 2011), raising a debate over the literature's credibility. Some argue that many findings are false, stemming from p-hacking and publication biases, which explain the lower post-publication performance and even irreproducibility of some results (Harvey et al., 2016); (Harvey and Liu, 2019); (Linnainmaa and Roberts, 2018); (Chordia et al., 2020); Hou et al. (2017); (McLean and Pontiff, 2016). However, others believe that p-hacking explanations are insufficient to explain some large t-stats obtained by certain anomalies (Chen, 2022) and also disagree with some irreproducibility claims, attributing differences in results to methodological choices concerning benchmark asset pricing models and weighting schemes (Jacobs and Müller, 2020); (Aghassi et al., 2023); (Jensen et al., 2023).

Traditional asset pricing literature often relies on characteristics-sorted portfolios and Fama-MacBeth regressions to identify anomalies/factors, but those methods have limitations. For instance, they struggle to capture non-linear relationships between characteristics and expected future returns, model interactions between signals and between signals and industries. Finally, they are prone to overfitting in the high-dimensional cross-section of ex-

pected future returns (Kelly and Xiu, 2023); (Cattaneo et al., 2020); (Goyal, 2012); (Patton and Timmermann, 2010); (Romano and Wolf, 2013); (Aghassi et al., 2023); (Piotroski and So, 2012); (Asness et al., 2000).

More flexible methods have been employed to address high-dimensionality and non-linearity in return prediction, aiming to strike a better balance between model complexity and approximation of the highly complex, time-dependent, and non-linear data generating process of asset returns. ML offers a diverse range of flexible alternatives compared to traditional models, better approximating the data generating process of cross-sectional returns while handling high-dimensional feature sets. In simple terms, this is achieved by incorporating non-linearities, interactions among signals, and tuning hyperparameters on validation samples to mitigate overfitting and improve out-of-sample forecasting (Gu et al., 2020; Kelly and Xiu, 2023; Kelly et al., 2022).

Besides aggregate and individual return prediction, ML has notable applications in portfolio optimization, yield and volatility forecasting, option valuation, and mutual fund manager selection (Kelly and Xiu, 2023). In portfolio selection, various studies explore ML techniques aiming at providing better risk-adjusted returns (e.g. Du (2022); Wu et al. (2023); Dai et al. (2024); Zhao et al. (2023); Alzaman (2024); Ma et al. (2021).

3. Methods and Data

The endeavour involves a dependent variable that is either a forward return or risk-adjusted return metric. Instead of raw returns, benchmark-relative active returns were considered, better neutralizing sources of systematic risks. Moreover, unlike absolute returns, active returns are more relevant to the portfolio optimization problem of benchmark-sensitive equity managers (Hsu et al., 2022).

Following Gu et al. (2020), we refer to the flexible functional form $g^*(.)$ to express the generating process of excess returns:

$$EXR_{i,t+h} = E_t(R_{i,t+h} - R_{m,t+h}|\mathbf{z_{i,t}}) + \epsilon_{i,t+h} = g^*(\mathbf{z_{i,t}}) + \epsilon_{i,t+h} \qquad (1)$$

where $EXR_{i,t+h}$ represents h-month return $R_{i,t+h}$ of stock i at time $t+h$ in excess to market-cap weighted benchmark return $R_{m,t+h}$. Conditional on a set of $j = 1, \ldots, J$ firm-level characteristics $\mathbf{z_{i,t}}$, $g^*(\mathbf{z_{i,t}})$ represents a general

flexible function that neither depends on $i = 1, \ldots, I$ or $t = 1, \ldots, T$, making estimates more stable (De Prado, 2018). Following Blitz et al. (2023), h is chosen as 3 and IBOV was chosen as cap-weighted benchmark.

The data used in this study comprises monthly observations from Brazilian listed stocks between March 2001 and September 2023, covering $T = 271$ months and $I = 1,014$ stocks in an unbalanced panel due to stocks being listed and delisted. Therefore, a observation $n = 1, \ldots, N$ consists of a combination of i and t.

Based on raw financial and market metrics from Economatica, $J = 93$ features were constructed, including valuation metrics, past price and return information, market cap, a comprehensive set of accounting indicators covering quality, profitability, leverage, investment and accruals anomalies, growth rate and standardized unexpected realizations of accounting indicators, liquidity indicators, risk measures related to market prices and fundamentals, macroeconomic factors and sector classifications. While the majority of variables were constructed in an absolute fashion, some industry-relative factors were also developed to account for within-industry effects (Asness et al., 2000).

Comparatively, Lewellen (2015), Freyberger et al. (2020), and Gu et al. (2020) used 15, 36, and 94 signals, respectively. In the latter study, the authors interacted original features with 8 aggregate macroeconomic predictors, increasing the total number of covariates to 920. In this work, the impact of macroeconomic factors is modeled through five features, which multiply the stock's sensitivity to a macro risk (beta) by the time-standardized macro factor (e.g., inflation and real rates).

Features were categorized in 15 different clusters/themes: Value, Low Risk, Accruals, Seasonality, Profit Growth, Leverage, Quality, Momentum, Skewness, Profitability, Reversal, Investment, Size, Macro and Sectors (Jensen et al., 2023). Further details are provided in the Appendix. Differences in definitions from original studies may exist because of data availability.

Each month t, characteristics were winsorized to percentiles 97.5% and 0.025% and then normalized to the $[-1, 1]$ interval. This procedure limits the effect of outliers and possible errors in the database. Importantly, at each month t, only data existing at that time was used for normalization, preventing forward-looking biases and data leakage (Coqueret and Guida, 2020; Gu et al., 2020; Hanauer and Kalsbach, 2023).

Another important consideration in terms of data cleansing and preprocessing relates to missing observations. Following Gu et al. (2020), to

leverage the available information, observations with missing data were filled according to pre-specified procedures based on industry or full-sample period-specific means.

Due to their abundance, micro and nanocap stocks tend to heavily influence estimates. However, their generally low liquidity poses real constraints on position sizing, especially for large investors. Therefore, for each period, stocks were ordered by mean 3-month trading volume, and those in the 0.5% percentile were excluded, following Hou et al. (2017), Jensen et al. (2023), and Hanauer and Kalsbach (2023). Despite their minimal participation in total trading volume, this group accounts for close to 60% of total observations.

In the database, stocks with no trading information were excluded. Although nanocaps were eliminated before model estimation, they were still used to rank stocks for each characteristic. After cleaning and pre-processing, a total of $N = 46,312$ observations were used for training, validation, and testing.

3.1. Choosing $g()$

As mentioned, ML literature provides options to approximate $g^*()$. Typically, for a given $g(\mathbf{z_{i,t}}, \theta)$ associated to model $m = 1, \ldots, M$, one must decide on the model's complexity P_m/N, where P_m is the length of the parameters vector $\theta_{\mathbf{m}} = (\theta_{m,1}, \theta_{m,2}, \ldots, \theta_{m,P_m})$.

Complexity is a very important aspect in return prediction. A simple model with few parameters will poorly approximate $g^*()$, even though its estimates have low variance, making them more stable. Conversely, complex models with more parameters tend to better approximate $g^*()$, but have higher variance. This is known as the bias-variance trade-off, a central concept in ML(Kelly et al., 2022).

In ML, balancing bias and variance and determining the complexity of $g()$ entails tuning model-specific hyperparameters vectors $\lambda_{\mathbf{m}} = (\lambda_{m,1}, \lambda_{m,2}, \lambda_{m,V_m})$, which enables shrinkage of $\theta_{\mathbf{m}}$ estimates and selection of complexity based on simulated out-of-sample behavior.

Kelly et al. (2022) argue that, given optimal shrinkage, expected out-of-sample expected returns are strictly increasing in complexity due to very unstable and non-linear behavior of stock returns requiring more parametrization. Hence, even highly complex models can generate substantial Sharpe ratios.

Hyperparameters were tuned through grid-search, a widely adopted method (Bouthillier and Varoquaux, 2020) that usually involves the following non-parametric approach.

First, a training set $\tau_{training}$ is used to fit the model under a given combination of pre-defined candidate values for each hyperparameter in $\lambda_{\mathbf{m}}$. Then, a validation set $\tau_{validation}$, such that $\tau_{validation} \cap \tau_{training} = \emptyset$, is used to evaluate a loss-function $\mathbf{L}()$, mimicking out-of-sample performance (Kelly and Xiu, 2023).

The process is repeated, exhaustively combining candidate values for hyperparameters and the combination $\lambda^*_{\mathbf{m}}$ that minimizes $\mathbf{L}()$ is selected.

Candidate values for hyperparameters should strike a balance between computational efficiency and coverage of a wide spectrum of values, thus encompassing various degrees of shrinkage for variance control (LaValle et al., 2004). Following related studies, the pool of candidates adopted for grid-search is outlined in Table 11.

Finally, the model is refit using $\tau_{training} \cup \tau_{validation}$ and genuine out-of-sample performance is measured by computing different statistics on $\tau_{testing}$, such that $\tau_{testing} \cap (\tau_{training} \cup \tau_{validation}) = \emptyset$ (Kelly and Xiu, 2023).

Best overall $g()$ is chosen based on an out-of-sample forecasting performance metric computed on $\tau_{testing}$. Following related studies, out-of-sample performance was primarily assessed by Out-of-Sample R^2 (R^2_{OOS}), under an undemeaned denominator (Gu et al., 2020).

Additionally, Root-Mean-Squared-Error (RMSE), Mean-Absolute-Percentage-Error (MAPE) and Accuracy were also calculated.

Each model's most influential features were also identified. In each case, variable importance was normalized to sum to one, focusing on relative importance.

3.1.1. Validation Scheme

In order not to make results dependent on a specific split, $\mathbf{T}$ splits are considered, such that $\tau_{\mathbf{training}}$, $\tau_{\mathbf{validation}}$ and $\tau_{\mathbf{testing}}$ each represent sets of groups of observations. For each specific split, corresponding groups are disjoint sets.

There are different schemes to split data, and the choice depends on distributional behavior and computational constraints (Schnaubelt, 2019).

A popular approach is k-fold Cross-Validation (CV), a resampling approach in which data is randomly divided into k equal-sized folds. A model is trained under $k-1$ folds and the remaining fold is used to compute $\mathbf{L}()$. The

process is repeated and the hyperparameter set $\lambda_{\mathbf{m}}^{*}$ that minimizes validation error across folds is chosen (Geisser, 1975). For the special computationally intensive case $k = N$, the method is called Leave-One-Out CV (Schnaubelt, 2019).

Although widely used, k-fold CV assumes independently and identically distributed (i.i.d.) observations, a condition often violated by financial data. This approach risks using future data to predict past values, overlooking time-dependent structures. Therefore, walk-forward validation is commonly employed, a sequential scheme in which training, validation, and testing subsets progress through time, preserving the temporal ordering of data (Schnaubelt, 2019; Bergmeir et al., 2018).

Two main walk-forward schemes exist: rolling and expanding. In rolling schemes, the training sample size remains fixed, causing the training, validation, and test samples to advance in time at each iteration. Conversely, expanding-windows schemes involve enlarging training sets recursively, while the validation and test sets progress (Kelly and Xiu, 2023; Tashman, 2000). There is an ongoing debate on which scheme is superior.

Except for special circumstances in which CV might be applicable to time series data, theory and empirical evidence are more supportive of walk-forward schemes when data isn't stationary, a condition that heavily impacts CV (Cerqueira et al., 2017; Bergmeir et al., 2018; Schnaubelt, 2019).

Therefore, in this study, a walk-forward expanding scheme, similar to Hanauer and Kalsbach (2023) and Gu et al. (2020), is employed. Therefore, at every refit, the training sample is increased, with a fixed origin, and the validation and testing sets are moved forward in time.

However, instead of refitting the model each month, which would bring a great deal of computational cost, a holding period of 12 months is introduced, although predictions are done monthly. Initial split considers a buffer period of 120 observations for training and 60 for validation.

Data leakage is a significant concern when designing the optimal scheme, as it introduces forward-looking bias. In finance, leakage often results in inflated backtests and disappointing out-of-sample risk-adjusted returns (De Prado, 2018).

In order to prevent leakage, at each fit, a h-sized space is introduced, thereby eliminating data points not known at each period.

3.1.2. Candidates for $g()$

In this paper, seven models were applied: Ordinary Least Squares (OLS), Elastic Net (ENET), Random Forest (RF), eXtreme Gradient Boosting - XGBoost (XGB) and three Artificial Neural Networks (ANN) with different architectures. Additionally, three ensembles were considered: one of all ANN (E_ANN), one with nonlinear models (E_NL), and a final comprising all models (E_ALL).

Model 1: OLS

For OLS ($m = 1$), there are no hyperparameters to tune, which tends to be a problem in large dimensional sets. For OLS, $g()$ assumes the following linear form:

$$g(\mathbf{z_{i,t}}, \theta_\mathbf{1}) = \mathbf{z_{i,t}^T}\theta_\mathbf{1} \tag{2}$$

In this case, $g()$ is a linear combination of signals and P_1 parameters. θ_1 are estimated by minimizing Mean-Squared Error (MSE) $\mathbf{L_{MSE}}()$ based on the pooled panel:

$$\mathbf{O_1}(\theta_\mathbf{1}) = \mathbf{L_{MSE}}(\theta_\mathbf{1}) = \frac{\sum_{i=1}^{I}\sum_{t=1}^{T}(EXR_{i,t+1} - g(\mathbf{z_{i,t}}, \theta_\mathbf{1}))^2}{N} \tag{3}$$

The biggest problem with OLS is overfitting, as the model's coefficients' variance tends to increase as $P/N \to 1$, which tends to make OLS perform poorly in the high-dimensional factor zoo.

Model 2: ENET

Regularization, such as l_1 and l_2 penalties, enhances out-of-sample performance in regression models. These penalties, included in $\mathbf{O_2}(\theta_\mathbf{2})$, shrink $\theta_\mathbf{2}$ estimates, improving the bias-variance trade-off (Gareth, 2013).

l_2 regularization reduces estimates toward zero, curbing large values, whereas l_1 regularization additionally forces certain covariates to zero, conducting feature selection. The combination of both penalties forms ENET:

$$\mathbf{O_2}(\theta_\mathbf{2}; \lambda_\mathbf{2}) = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T}(EXR_{i,t+1} - g(\mathbf{z_{i,t}}, \theta_\mathbf{2}))^2}{N} + \lambda_2^{\psi}(1-\lambda_2^{\rho})\sum_{j=1}^{J}|\theta_{2,j}| + \frac{1}{2}\lambda_2^{\psi}\lambda_2^{\rho}\sum_{j=1}^{J}\theta_{2,j}^2 \tag{4}$$

Setting $\lambda_2^{\psi} = 0$ is equivalent to OLS.

For $m = 1, 2$, variable importance was determined by the absolute value of estimated coefficients.

Models 3 and 4: Ensembled tree-based regressions

While ENET improves the bias-variance trade-off through shrinkage, it still imposes a linear functional form, which tends to underperform flexible models that can better approximate $g^*()$.

Regression trees are nonparametric models that introduce non-linearities and interactions among $\mathbf{z_{i,t}}$, by employing sequential splitting rules to cluster observations into bins.

During tree growth, branches are sequentially formed, splitting observations from the previous step into rectangular partitions based on features, grouping together observations that share a common relationship with the feature. When further splits are not feasible or when reaching a maximum depth $\lambda^D_{m\in(3,4)}$, a leaf l emerges. Predictions are then made by averaging the values of EXR_{t+h} for observations within each partition (Kelly and Xiu, 2023).

$\lambda^D_{m\in(3,4)}$ represents the maximum number of separations along the longest branch, so that a tree of depth $\lambda^D_{m\in(3,4)}$ can have $\lambda^D_{m\in(3,4)} - 1$ interactions. Another tree-specific hyperparameter is number of terminal nodes (leaves) $\lambda^L_{m\in(3,4)}$. Both tend to control tree complexity:

$$g(\mathbf{z_{i,t}}; \theta_{\mathbf{m}\in(\mathbf{3},\mathbf{4})}, \lambda_{\mathbf{m}\in(\mathbf{3},\mathbf{4})}) = \sum_{l=1}^{\lambda^L_m} \theta_{m\in(3,4),l} \mathbf{1}_{[\mathbf{z_{i,t}}\in C_l(\lambda^D_{m\in(3,4)})]} \tag{5}$$

where $C_l(\lambda^D_{m\in(3,4)})$ is the l-th partition. $\mathbf{1}_{[\mathbf{z_{i,t}}\in C_l(\lambda^D_{mm\in(3,4)})]}$ indicates if an observation is in a specific bin and $\theta_{m\in(3,4),l}$ is the average of outcomes relative to that bin:

$$\theta_{m\in(3,4),l} = \frac{1}{N_l} \sum_{\mathbf{zi,t}\in C_l(\lambda^D_{m\in(3,4)})} EXR_{i,t+h} \tag{6}$$

The algorithm to define a region $C_l(\lambda^D_{m\in(3,4)})$ minimizes MSE. Since considering every possible partition is computationally burdensome, a recursive binary splitting approach is commonly used. This top-down greedy algorithm selects the best split at each step of growing a tree, rather than considering potential impacts in subsequent steps. It begins by selecting characteristic

z_j and cut-point z_j^* that lead to the greatest reduction in MSE (Breiman, 1984).

The process is repeated until a stopping criterion is called ($\lambda^L_{m\in(3,4)}$ or $\lambda^D_{m\in(3,4)}$).

While a flexible model, a single tree is also prone to overfitting. Therefore, popular choices for improving out-of-sample performance are pruning and ensembling, the latter consisting of regularization by combining multiple trees.

One such ensembling approach is bagging, which generates $\lambda^T_{m\in(3,4)}$ trees using bootstrap resampling. Each tree is trained on a subset of $\lambda^B_{m\in(3,4)}$ bootstrap samples and then predictions are averaged, reducing variance.

Instead of considering all characteristics, one might use only a random subset, comprising a fraction $\lambda^R_{m\in(3,4)}$ of J. This reduces the impact of dominant characteristics and creates less correlated trees, improving overall out-of-sample performance. When $\lambda^R_{m\in(3,4)} \neq 1$, we have RF ($m = 3$). As shown in Table 11, bagging ($\lambda^R_3 = 1$) is covered during grid-search (Breiman, 2001).

Grid-search for $m = 3$ focused solely on λ^R_3 and λ^D_3). λ^T_3 was set at a relatively high number to counterbalance variance reduction, computational costs and achieve convergence (Probst et al., 2019). λ^B_3 is set as one and minimal terminal node sizes are set to 1 (Wright and Ziegler, 2017), hence controlling λ^L_3.

Another ensembling option is boosting, which sequentially grows trees, fitting each new tree to the residuals of the last tree. To prevent overfitting, boosting combines forecasts from individual, usually underfit on their own, shallow trees with depth λ^D_4, denominated weak learners, which implies that the algorithm learns slowly (Friedman, 2001). Therefore, the process is different to RF, in which trees are combined in an independent and agnostic way.

In particular, the scalable and computationally efficient XGB system of Chen and Guestrin (2016) was applied ($m = 4$).

The model seeks to minimize the regularized objective by using λ^F_4 additive functions to predict the output, one step at a time:

$$\mathbf{O_4}(\theta_4, \lambda^F_4, \Omega(T_f)) = \sum_{i=1}^{N}\sum_{t=1}^{T} \mathbf{L}(EXR_{i,t+h}, g(\mathbf{z_{i,t}}, \theta_4)) + \sum_{f=1}^{\lambda^F_4} \Omega(T_f) \quad (7)$$

The first term, measured over all observations, represents a differentiable

convex loss-function and the second one is a penalization term:

$$\Omega(T_f) = \lambda_4^{\gamma}\lambda_4^{L} + \frac{1}{2}\lambda_4^{\Lambda}\sum_{l=1}^{\lambda_4^{L}}\theta_{4,l}^2 \tag{8}$$

For a given tree T, its structure can be defined by $T(\mathbf{z_{i,t}}) = \theta_{q(\mathbf{z_{i,t}})}$, $q : \mathbb{R}^j \to \lambda_4^L$ where q represents the tree structure that maps an input to its final leaf. The penalty term can be decomposed into a penalty λ_4^{γ} on the total number of leaves, controlling depth, and a shrinkage λ_4^{Λ} that penalizes the magnitude of outputs, reducing variance.

Equation 7 is used to train the model in an additive greedy manner. Given a previous structure of $f = 1, \ldots, F-1$ trees, a tree T_F is added in such a way to minimize prevailing error:

$$\mathbf{O_4} = \sum_{i=1}^{I}\sum_{t=1}^{T}\mathbf{L}(EXR_{i,t+h}, m_{f-1}(\mathbf{z_{i,t}}) + T_F(\mathbf{z_{i,t}})) + \sum_{f=1}^{\lambda_4^{F}}\Omega(T_F) \tag{9}$$

where m_{F-1} is the model up to tree T_F. Following the derivation in Coqueret and Guida (2020) and Chen and Guestrin (2016), for a fixed structure $q(\mathbf{z_{i,t}})$ and assuming a quadratic loss function for $\mathbf{L}$, the optimal weight $\theta_{4,l}^*$ of leaf l and corresponding optimal objective can be, respectively, given by Equations 10 and 11:

$$\theta_{4,l}^* = \frac{\sum_{(i,t)\in I_l}(EXR_{i,t+h} - m_{F-1}(\mathbf{z_{i,t}}))}{1 + \frac{\lambda_4^{\Lambda}}{2}\#\{(i,t) \in I_l\}} \tag{10}$$

$$\mathbf{O}_{\mathbf{4},\lambda_4^L}(q) = -\frac{1}{2}\sum_{l=1}^{\lambda_4^{L}}\frac{(EXR_{i,t+h} - m_{F-1}(\mathbf{z_{i,t}}))^2_{(i,t)\in I_l}}{1 + \frac{\lambda_4^{\Lambda}}{2}\#\{(i,t) \in I_l\}} + \lambda_4^{\gamma}\lambda_4^{L} \tag{11}$$

$\#$ counts the items in set I_l, representing observations in leaf l. Equation 11 serves as a scoring function for a q structure.

To build q, the algorithm, at each node, checks if a split is useful according to the objective function. This is a greedy approach to tree growth, as considering all possible q is impractical. The gain from a split is calculated as:

$$\mathbf{O_{split}} = \frac{1}{2}\left[\frac{\sum_{(i,t)\in I_{left}}(EXR_{i,t+h} - m_{F-1}(\mathbf{z_{i,t}}))^2}{1+\frac{\lambda_4^\Lambda}{2}\#\{(i,t)\in I_{left}\}} + \frac{\sum_{(i,t)\in I_{right}}(EXR_{i,t+h} - m_{F-1}(\mathbf{z_{i,t}}))^2}{1+\frac{\lambda_4^\Lambda}{2}\#\{(i,t)\in I_{right}\}} - \frac{\sum_{(i,t)\in I}(EXR_{i,t+h} - m_{F-1}(\mathbf{z_{i,t}}))^2}{1+\frac{\lambda_4^\Lambda}{2}\#\{(i,t)\in I\}}\right] - \lambda_4^\gamma \quad (12)$$

In this case, $I = I_{left} \cup I_{right}$ measures the original gain, while the other two correspond to gains of left and right bins. The penalty term λ_4^γ controls the minimum loss reduction required to make a further partition.

Besides a λ_4^Λ l_2 normalization, a l_1 λ_4^α penalty on weights can also be introduced. This way, the penalty function is as follows:

$$\Omega(T_f) = \lambda_4^\gamma \lambda_4^L + \frac{1}{2}\lambda_4^\Lambda \sum_{l=1}^{\lambda_4^L} \theta_{4,l}^2 + \lambda_4^\alpha \sum_{l=1}^{\lambda_4^L} |\theta_{4,l}| \quad (13)$$

XGB also implements variance reduction through a shrinkage parameter that scales each θ_l by a learning rate $\lambda^\eta 4 \in (0, 1]$, after each step of tree boosting. This reduces the influence of each individual tree, letting future trees improve the overall model and preventing overfitting, as a large number of aggregate optimized trees may introduce significant generalization error. Another technique used to improve variance is feature subsampling, set by $\lambda^R 4$, akin to RF, or row subsampling λ_4^S, which works in a similar manner.

Another crucial resource for training the model is early stopping, which involves halting the learning process when validation error ceases to improve after λ_4^ϕ rounds.

For tree methods, feature importance was calculated based on global overall gain attributed to subsequent splits related to a specific feature, across the nodes for which the characteristic was selected (Ishwaran et al., 2014).

Model 5: Neural Networks

Widely used across various tasks, including financial problems, ANN are the final models used to approximate $g^*()$. These highly flexible models can

arbitrarily approximate continuous functions with at least one hidden layer, by continuously adding new units (Costarelli et al., 2016).

ANN, mainly feed-forward ones, are commonly used in return prediction due to their flexible structures. Information flows from covariates, through hidden layers that apply nonlinear functions, to the output layer, which aggregates predictions into the final outcome. These networks compound together functions, forming a chain, and the chain's length, interpreted as model depth, is associated with the idea of deep learning (Goodfellow et al., 2016). As such:

$$g(\mathbf{z_{i,t}}) = g^{(3)}(g^{(2)}(g^{(1)}(\mathbf{z_{i,t}}))) \tag{14}$$

in which $g^{(1)}$, $g^{(2)}$ and $g^{(3)}$ are first, second and third layers of the network. A simple network with one hidden layer with two units consists of first applying $\mathbf{h} = g^{(1)}(\mathbf{z_{i,t}}, \theta_{\mathbf{5}}^{(\mathbf{1})})$ and then $g^{(2)}(\mathbf{h}, \theta_{\mathbf{5}}^{(\mathbf{2})}, \theta_0^{(2)})$, where subscripts denote layers. For $g^{(1)}$, a common choice is a nonlinear function, often an affine transformation controlled by learned parameters followed by an activation function (Goodfellow et al., 2016). Following Gu et al. (2020) and Hanauer and Kalsbach (2023), the popular rectified linear unit (ReLU) activation function is used (Feng and Lu, 2019).

$$ReLU(x) = \max(0, x) \tag{15}$$

Therefore, the network is:

$$g(\mathbf{z_{i,t}}, \theta_{\mathbf{5}}^{(\mathbf{1})}, \theta_{\mathbf{5}}^{(\mathbf{2})}) = {}^{\mathrm{t}}\theta_{\mathbf{1,5}}^{(\mathbf{2})} \max(0, {}^{\mathrm{t}}\theta_{\mathbf{1,5}}^{(\mathbf{1})}\mathbf{z_{i,t}} + \theta_{0,5}^{(1)}) + \theta_{0,5}^{(2)} \tag{16}$$

There are numerous architectures for an ANN, being Equation 16 a simple choice. By defining $\lambda_5^{U^{(\pi)}}$ as the number of units in each layer $\pi = 1, \ldots, \lambda_5^{\pi}$ and setting the output of unit u in layer π as $o_u^{(\pi)}$, with o^0 being initialized with the features, one has the following iterative equation for the network at each unit in layer $\pi \neq 0$:

$$o_u^{(\pi)} = \max(0, {}^{\mathrm{t}}\theta_{\mathbf{u,5}}^{(\pi-\mathbf{1})} o^{(\pi-1)}) \tag{17}$$

Grid searching all possible structures is demanding, so architectures are usually set in advance.

Given results of Eldan and Shamir (2016), Gu et al. (2020), and Orimoloye et al. (2019) on shallow (up to three layers) networks outperforming deep ones in return prediction, especially in smaller datasets, three architectures

are predefined, with neuron numbers following the geometric pyramid rule of Masters (1993).

Therefore, models consisted of ANN1 - a single hidden layer with 32 neurons, ANN2 - two layers with 32 and 16 neurons, and ANN3 - three hidden layers with 32, 16, and 8 neurons.

In order to optimize, the following objective function is defined:

$$\mathbf{O_5} = \sum_{i=1}^{I} \sum_{t=1}^{T} \mathbf{L}(EXR_{i,t+h}, g(z_{i,t}, \theta_{\mathbf{5}}) + \Omega(\theta_{\mathbf{5}}) \tag{18}$$

Due to their high parametrization and nonlinear nature, ANN are typically trained using Stochastic Gradient Descent (SGD). This is done to reduce computational demands, though it may impact accuracy (Bottou and Bousquet, 2012). SGD's effectiveness can be sensitive to initial values, so it's often recommended to initialize feedforward networks with small random values (Goodfellow et al., 2016).

Gradient descent involves updating weights according to:

$$\theta_{\mathbf{5}} \leftarrow \theta_{\mathbf{5}} - \lambda_5^{\eta} \frac{\partial \mathbf{L}(EXR_{i,t}, g(z_{i,t}, \theta_{\mathbf{5}}))}{\partial \theta_{\mathbf{5}}} \tag{19}$$

where $\theta_{\mathbf{5}}$ denotes all weights of the network. The choice of λ_5^{η} is crucial, as large values may cause the algorithm not to converge.

The finite difference method is commonly used to approximate derivatives. By leveraging the chain rule and recycling mechanism detailed in Coqueret and Guida (2020), computation speed can be improved. During back-propagation, derivatives are computed from the last layer to the first, opposite to the forward pass that evaluates the loss.

To enhance computational efficiency while leveraging the benefits of SGD, it's common to use intermediate steps between SGD and using the entire training sample. This is achieved by sampling random groups of instances, called batches, with a given size λ_5^b. To cover the full sample, $\frac{N}{\lambda_5^b}$ iterations are needed, and once all batches are used to update $\theta_{\mathbf{5}}$, an epoch is reached. SGD is a special case of this approach when $\lambda_5^b = 1$. In addition to λ_5^b, the number of epochs λ_5^e must also be set.

In this work, we adopted the Adam algorithm, which efficiently estimates first and second-order moments. It's well-suited for highly parametrized models (Kingma and Ba, 2014).

The second part of 18 refers to the penalization term $\Omega(\theta_{\mathbf{5}})$, which incorporates l_1 and l_2 penalizations:

$$\Omega(\theta_{\mathbf{5}}) = \sum_{k=1}^{K} \lambda_5^{\omega} ||\theta_{\mathbf{k,5}}|| + \sum_{j=1}^{J} \lambda_5^{\delta} ||\theta_{\mathbf{j,5}}||^2 \tag{20}$$

where k and j refer to which θ l_1 and l_2 penalizations were applied.

Other regularization methods include early stopping, which halts training before epoch $\lambda^e 5$ if there is no improvement in validation error for more than $\lambda^{\phi} 5$ epochs, and Dropout, which randomly omits λ_5^R of neurons, shrinking the network (Glorot and Bengio, 2014). Lee (2020) found that dropout can efficiently decrease overfitting risk in the context of stock return prediction with deep neural networks.

The batch normalization method of Ioffe and Szegedy (2015) is applied, which adaptively normalizes layer inputs over training to aid in gradient propagation. Even if data is normalized in pre-processing, transformations applied may cause internal covariate shift in data. Finally, an ensemble approach is used, employing λ_5^{ι} random seed weight initializers and averaging all predictions (Dietterich, 2000).

Ensembles

Finally, ensembling was applied, combining predictions from different models. This method, used in studies like Toochaei and Moeini (2023), improves generalization. Combining predictions tends to yield more accurate forecasts, especially when errors are low-correlated (Kuncheva and Whitaker, 2003).

Various methods, including linear combinations and stacked ensembles, can be used. In this work, three linear ensembles were considered: one using only ANN, another with all nonlinear methods, and a final one with all models.

3.2. ML Portfolios

ML expected returns were used as inputs to build portfolios and backtest investment strategies. All portfolios used these forecasts to form investable sets, selecting the quartile of stocks $s = 1, \ldots, S$ with the highest forecast, thus reducing dimensionality for portfolio allocation (Coqueret and Guida, 2020).

Each portfolio started in February 2016, the first out-of-sample month, considering training and validation sample sizes. Rebalancing occurred in March, April, May, August, and November to balance the trade-off between new accounting information and turnover. A 7 bps direct transaction cost was assumed, with market impact costs estimated using Barra's square root model (Barra, 1997). Buffering was applied to reduce turnover, and stocks with benchmark weights greater than 0.03 were also included in the investable set to reduce tracking error.

3.2.1. Heuristic Portfolios

Initial portfolios utilize heuristic methods like Equal-Weighted (EW), Signal-Weighted (SW), Capitalization-Weighted (CW), and Capitalization-Scaled (CS). These methods do not rely on the covariance matrix for weight determination (Ghayur et al., 2019). Weights for EW, SW, CW, and CS are as follows:

$$w_{s,EW} = 1/S \tag{21}$$

$$w_{s,SW} = \frac{Z(\widehat{EXR}_{s,t+h})}{\sum_{s\in S} Z(\widehat{EXR}_{s,t+h})} \tag{22}$$

$$w_{s,CW} = \frac{Z(C_{s,t})}{\sum_{s\in S} Z(C_{s,t})} \tag{23}$$

$$w_{s,CS} = \frac{Z(C_{s,t})Z(\widehat{EXR}_{s,t+h})}{\sum_{s\in S} C(\widehat{EXR}_{s,t})Z(\widehat{EXR}_{s,t+h})} \tag{24}$$

where Z calculates z-score of a variable and $C_{s,t}$ is trading volume. Except for EW, all allocation schemes employed ML returns to determine weights, but neither used risk measures.

3.2.2. Risk-Parity Portfolios

For remaining portfolios, covariance matrix Σ_t was estimated, considering three options: sample ($\widehat{\Sigma}_{SAM,t}$), PCA ($\widehat{\Sigma}_{PCA,t}$), and Shrinkage ($\widehat{\Sigma}_{S,t}$) toward the constant correlation target matrix. Linear shrinkage intensity followed Ledoit and Wolf (2003, 2004). In case of $\widehat{\Sigma}_{PCA,t}$, number of factors was set to $log(S)$ (Coqueret and Milhau, 2014).

$\widehat{\Sigma}_{SAM,t}$ is prone to significant estimation error due to high dimensionality, rendering it unstable. Increasing sample size can be detrimental to out-of-sample performance due to parameter shifting, hence making $\widehat{\Sigma}_{PCA,t}$ and $\widehat{\Sigma}_{S,t}$ better alternatives (Ledoit and Wolf, 2022; Coqueret and Milhau, 2014). This way, at each rebalancing, the sample for estimating Σ_t comprised 720 trading days.

Risk-parity (RP) portfolios frequently outperform optimal mean-variance portfolios in a long-only context, raising their popularity (Chaves et al., 2011). Given estimated portfolio volatility $\widehat{\sigma}(\mathbf{w_t}) = \sqrt{\mathbf{w_t}^T \widehat{\Sigma_t} \mathbf{w_t}}$, risk-parity equalizes risk contribution RC_s of stock s across S (Spinu, 2013).

$$RC_{s,t} = \frac{w_{s,t}(\widehat{\Sigma}_t \mathbf{w_t})_s}{\sqrt{\mathbf{w_t^T} \widehat{\Sigma_t} \mathbf{w_t}}} = \frac{1}{S} \widehat{\sigma}(\mathbf{w_t}) \tag{25}$$

With long-only and full-investment constraints, weights are given by the vanilla convex formulation:

$$\underset{\mathbf{x_t} \geq 0}{\arg\min} \frac{1}{2} \mathbf{x_t}^T \widehat{\Sigma}_t \mathbf{x_t} - \mathbf{w_t}^T log(\mathbf{x_t}) \tag{26}$$

where $\mathbf{x_t} = \frac{\mathbf{w_t}}{\sqrt{\mathbf{w_t^T} \widehat{\Sigma_t} \mathbf{w_t}}}$.

The risk-parity portfolio was computed, one to each $\widehat{\Sigma_t}$. While heuristic portfolios don't incorporate risk in setting weights, risk-parity portfolios don't consider ML expected returns, except for investable set formulation.

3.2.3. Optimal Mean-Tracking Error Portfolios

Adapting Modern Portfolio Theory (MPT) framework to relative risk and return inputs gives the following Information Ratio (IR) maximization problem:

$$\underset{\mathbf{w_t}}{\arg\max} \frac{\mathbf{w_t^T} \widehat{EXR}_{t+h}}{\sqrt{\mathbf{w_t^T} \widehat{\Sigma_t} \mathbf{w_t}}} \tag{27}$$

This gives the tangency portfolio in Mean Tracking-Error (MTO) domain.

However, an unrestricted MTO_UNC approach (except for long-only and full-investment constraints) might lead to significant concentration and corner solutions, a recognized limitation of MPT. Additionally, it fails to incorporate many real-life conditions faced by equity managers.

Therefore, a constrained MTO_CON version was also explored, with box, turnover, and sector constraints chosen to simulate performance under more realistic risk and trading control rules:

1. **Upper Box Constraints:**
   (a) Regular stocks:
   $w_{s,t} \leq w_{s,b,t} + 0.03$
   (b) Stocks included only because $w_{s,b,t} \geq 0.03$:
   $w_{s,t} \leq w_{s,b,t} + 0.02$
   (c) Small-caps ($Z(C_{s,t}) \leq Q_{10}(Z(C_t))$:
   $w_{s,t} \leq w_{s,b} + 0.025$
   (d) Micro-caps ($Z(C_{s,t}) \leq Q_3(Z(C_t))$:
   $w_{s,t} \leq w_{s,b,t} + 0.0075$
2. **Lower Box Constraints:**
   (a) Regular stocks:
   $w_{s,t} \geq max(w_{s,b,t} - 0.02, 0)$
   (b) Stocks included only because of $w_{s,b,t} \geq 0.03$, but that belong to bottom forecast quintile:
   $w_{s,t} \geq max(w_{s,b,t} - 0.03, 0)$
3. **Turnover Constraints:**
   (a) Small-caps:
   $|w_{s,t} - w_{s,t-1}| \leq 0.015$
   (b) Micro-caps:
   $|w_{s,t} - w_{s,t-1}| \leq 0.005$
4. **Sector Constraint:**
   (a) $\left|\sum_{s \in SectorS} w_{s,t} - w_{s,b,t}\right| \leq 0.05$

Additionally, for stocks with volatility in 95, 97,5 and 99 quantiles, max active weights were multiplied by 0.75, 0.50 or 0, respectively.

4. Results

In this section, results on ML models' out-of-sample predictive performance and portfolio performance are presented.

4.1. Expected Returns Model

Table 1 presents summary statistics for out-of-sample forecasting errors. While OLS exhibited the highest RMSE, ML methods generally yielded superior results due to the introduction of shrinkage and non-linearities. Specifically, E_A had the lowest RMSE overall, with RF emerging as the best individual performer.

Table 2 presents R^2_{OOS}, Accuracy, and MAPE. OLS performed poorly in terms of R^2_{OOS} and MAPE, with negative value for R^2_{OOS}, thus aligning with results of Gu et al. (2020) and Hanauer and Kalsbach (2023). The introduction of regularization methods and flexible functional forms leads to significant improvement in this metric.

Particularly, ANN showed good performance for the first two architecture choices, declining for ANN3. This also aligns with literature's findings of shallow outperforming deep learning in return prediction (Kelly and Xiu, 2023).

Even simple ensembling methods yielded favorable results, with E_A displaying best R^2_{OOS} overall, making it the preferred choice for portfolios.

Figure 1 illustrates hyperparameter choice for each rebalancing. For certain models, like ENET, there was minimal variation, while for others, there was more heterogeneity.

Finally, variable importance plots are displayed in Figure 2. Each subfigure represents the relative importance of the 25 most influential variables to that specific method.

Some variables were rather consensual choices among models, such as mom_res_12m, assets_yield, sales_yield, roe_3m, net_mrg_12m and vol_12m.

Particularly, important features for nonlinear models are: mkt_cap, tobin_q, turnover_3m, beta_mrkt_36m, curr_ratio, prc_highprice_12m, max_ret_1m, book_ lev and book_yield. Reversal and Size themes only appeared in non-linear models.

Overall, among 39 most important features, there were 8 in Value (47%), 6 in Momentum (67%), 5 in Quality (71%), 4 in Profitability (44%), 3 in Low Risk (100%), 3 in Size (43%), 3 in Profit Growth (27%), 2 in Leverage (50%), 1 in Reversal (33%), 2 in Macro (50%), 1 in Investment (25%) and 1 in Accruals (50%). Skewness, Seasonality and Sectors had no appearances.

4.2. ML Portfolios Performance

As mentioned, portfolios used E_A as an expected returns model, due to its superiority in R^2_{OOS}. At total, 13 portfolio schemes were used: 4 heuristics (EW, CW, CS, SW), 3 RP, 3 MTO_UNC and 3 MTO_CON, each using $\widehat{\Sigma}_{SAM,t}$, $\widehat{\Sigma}_{PCA,t}$ or $\widehat{\Sigma}_{S,t}$.

To assess out-of-sample ML performance, three benchmarks were used. The first was the cap-weighted IBOV index, the most prominent Brazilian equities benchmark.

The other two were constructed from a sample of equity funds: the average fund (FUNDS_MEAN) and the average Best-in-Class fund (FUNDS_Q75), the latter comprising funds whose returns ranked in the top quartile during the testing subsample. Even though FUNDS_Q75 isn't identifiable ex-ante, it serves as a performance reference.

This sample encompassed 818 domestic and non-exclusive funds. Non-longer active funds were included to mitigate survivorship bias.

To clarify, results present metrics computed on the average return from the two fund groups, rather than the average of individual funds' metrics.

Finally, the sample median fee of 2.0% was deducted from ML portfolios to make comparisons fair.

Table 3 indicates that ML portfolios exhibited significantly higher annualized returns compared to all three benchmarks, averaging two to three times the return of FUND_MEAN and nearly double that of IBOV. While the mean active equity manager underperformed IBOV, Best-in-Class managers outperformed it, albeit unable to surpass any ML portfolio.

Skewness calculations corroborate ML outperformance. While figures were mostly negative, likely influenced by the pandemic, they were less negative than IBOV and active managers, with some even being positive, reflecting a higher proportion of above-median returns.

Conversely, ML portfolios exhibited higher risk, on average, as indicated by standard deviation. However, due to higher returns, they delivered higher risk-adjusted returns, as measured by Sharpe ratios. It's notable that heuristic portfolios yielded higher returns than risk-parity, with the latter mitigating risk accordingly. Additionally, it's worth mentioning that $\widehat{\Sigma}_{t,PCA}$ and $\widehat{\Sigma}_{t,S}$ were associated to increased Sharpe ratios across all frameworks.

Tail measures are crucial in finance, stemming from return distributions that usually deviate from normality and standard deviation's inability to distinguish between upside and downside variation. Following Prospect Theory,

investors are actually loss-averse, making downside measures more pertinent.

Concerning tail risk, Table 4 displays that RP and MTO portfolios showed less kurtosis than IBOV and FUNDS_MEAN, but not FUNDS_Q75, while heuristics showed higher figures. For ETL, ML portfolios showed in-line results with IBOV and FUNDS_MEAN, but again worse than FUNDS_Q75.

Despite exhibiting more downside risk than FUNDS_Q75, risk-adjusted return measures based on downside risk, such as Sortino and Rachev Ratios, markedly favored ML portfolios when compared to all benchmarks.

Results regarding risk-return trade-off are captured in risk-return scatter-plot of Figure 3. Additionally, box-plots summarising the distribution of ML portfolios and benchmarks are displayed in Figure 4.

The discussion between active and passive management is an old topic in finance. With the widespread increase of passive ETFs, the average investor has a very accessible instrument to replicate cap-weighted benchmark. Therefore, for equity managers, it is very important to measure relative performance with the cap-weighted benchmark.

The debate between active and passive management has long been an important topic in finance. With the widespread proliferation of passive ETFs, the average Brazilian investor has an accessible vehicle to replicate cap-weighted benchmark. Consequently, it becomes crucial for equity managers to measure relative performance.

Figure 5 illustrates cumulative net relative performance for ML Portfolios, FUNDS_MEAN and FUND_Q75, showing a clear outperformance for the former. On the other hand, Table 5 revealed a higher tracking error. Nevertheless, higher level of active returns contributed to favoring ML Portfolios concerning the active risk and return trade-off, as measured by IR.

Despite critics, CAPM remains a useful risk model benchmark in literature. Relatedly to the above, Table 5 highlights positive alphas for ML strategies, with t-stats exceeding 2.0 and remarkably above 3.0 or 4.0 for all but MTO_UNC, thus above Harvey et al. (2016) threshold. The average manager displayed non-significant CAPM alphas, while Best-in-Class achieved a t-stat of 2.68.

As expected, CW allocation heuristic showed less active risk than other heuristics and MTOs, by factoring in capitalization when setting weights. However, risk-parity schemes yielded even lower tracking error. Finally, it's noticeable the benchmark-risk reduction and IR/T-Stat increase because of MTO_CON's risk controls. This is also usually true for $\widehat{\Sigma}_{PCA,t}$ and $\widehat{\Sigma}_{S,t}$.

Figure 6 reveals that benchmark outperformance stemmed from a combination of an upside capture ratio above 1 or even 1.2, along with a downside capture ratio below 1 or even 0.9, resulting in superior performance compared to FUND_MEAN. In contrast, while downside capture was similar to FUND_Q75, upside capture was notably better. Hence, although similar in downside protection, ML strategies demonstrate greater proficiency in capturing higher returns during bull markets.

ML portfolios out-of-sample performance relates to results of Hanauer and Kalsbach (2023) and Gu et al. (2020). Nevertheless, despite obvious differences in sample and feature sets, the present work employed benchmark-relative 3-month returns, instead of raw 1-month returns, and also different portfolio-construction schemes.

5. Conclusion

In this study, various ML methods were employed to forecast market-relative cross-sectional returns over a 3-month horizon. This involved fitting OLS, ENET, RF, XGB and ANN to nearly one hundred signals from the factor zoo literature, covering the most common anomalies and factor themes, and a sample of 1,014 Brazilian stocks spanning 271 months.

Contrary to other areas where ML has been successful, financial data exhibit non-stationarity and structural breaks, calling for the adoption of a Walk-Forward validation-scheme.

Regularized and nonlinear models demonstrated higher predictive performance than OLS, with positive R^2_{OOS}, being E_A the most accurate method. Therefore, using its predictions as expected return inputs, diverse long-only portfolio construction methods were applied, encompassing heuristics, risk-parity and MTO.

Those exhibited out-of-sample significant alphas and higher risk-adjusted return metrics than the average equity manager, which underperformed the market, and the average return of Best-in-Class managers. While these results don't imply the replacement of humans in stock picking and equity portfolio management, they highlight the importance of systematic ML strategies grounded in asset pricing literature, deserving greater attention from Brazilian investors. Future works can explore possible diversification benefits stemming from combining ML and human skill in equity portfolios.

References

Aghassi, M., Asness, C., Fattouche, C., Moskowitz, T., 2023. Fact, fiction and factor investing. The Journal of Portfolio Management 49, 1–38.

Alzaman, C., 2024. Deep learning in stock portfolio selection and predictions. Expert Systems with Applications 237, 121404. URL: https://www.sciencedirect.com/science/article/pii/S0957417423019061, doi:https://doi.org/10.1016/j.eswa.2023.121404.

Asness, C., Porter, B., Stevens, R., 2000. Predicting stock returns using industry-relative firm characteristics. Available at SSRN: https://ssrn.com/abstract=213872 .

Barra, 1997. Barra market impact model handbook. Berkeley.

Bergmeir, C., Hyndman, R., Koo, B., 2018. A note on the validity of cross-validation for evaluating autoregressive time series prediction. Computational Statistics & Data Analysis 120, 70–83.

Blitz, D., Hanauer, M., Hoogteijling, T., Howard, C., 2023. The term structure of machine learning alpha. Available at SSRN: https://ssrn.com/abstract=4474637 .

Bottou, L., Bousquet, O., 2012. The trade-offs of large scale learning. MIT Press.

Bouthillier, X., Varoquaux, G., 2020. Survey of machine-learning experimental methods at NeurIPS2019 and ICLR2020. Technical Report. Inria Saclay Ile de France.

Breiman, L., 1984. Classification and regression trees. Routledge.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Cattaneo, M., Crump, R., Farrell, M., Schaumburg, E., 2020. Characteristic-sorted portfolios: Estimation and inference. The Review of Economics and Statistics 102, 531–551.

Cerqueira, V., Torgo, L., Smailovic, J., Mozetic, I., 2017. A comparative study of performance estimation methods for time series forecasting. 2017 IEEE International Conference on Data Science and Advanced Analytics .

Chaves, D., Hsu, J., Li, F., Sharkernia, O., 2011. Risk parity portfolios vs. other asset allocation heuristic portfolios. The Journal of Investing Spring 20, 108–118.

Chen, A., 2022. Do t-statistic hurdles need to be raised. arXiv.org 2204.10275.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining , 785–794.

Chordia, T., Goyal, A., Saretto, A., 2020. Anomalies and false rejections. The Review of Financial Studies 33, 2134–2179.

Cochrane, J., 2011. Discount rates. NBER Working Papers 16972.

Coqueret, G., Guida, T., 2020. Machine learning for factor investing: R version. Chapman & Hall.

Coqueret, G., Milhau, V., 2014. Estimating covariance matrices for portfolio optimization. ERI Scientific Beta White Paper .

Costarelli, D., Spigler, R., Vinti, G., 2016. A survey on approximation by means of neural network operators. Journal of NeuroTechnology , 1–24.

Dai, H.L., Lai, F.T., Huang, C.Y., Lv, X.T., Zaidi, F.S., 2024. Novel online portfolio selection algorithm using deep sequence features and reversal information. Expert Systems with Applications 255, 124565. URL: https://www.sciencedirect.com/science/article/pii/S0957417424014325, doi:https://doi.org/10.1016/j.eswa.2024.124565.

De Prado, M.L., 2018. Advances in financial machine learning. John Wiley & Sons.

Diasio, D., Barrington, M., Kapoor, S., 2023. Ai anxiety in business survey. https://www.ey.com/en_us/consulting/businesses-can-stop-rising-ai-use-from-fueling-anxiety.

Dietterich, T., 2000. Ensemble methods in machine learning. International workshop on multiple classifier systems , 1–15.

Du, J., 2022. Mean–variance portfolio optimization with deep learning based-forecasts for cointegrated stocks. Expert Systems with Applications 201, 117005. URL: https://www.sciencedirect.com/science/article/pii/S0957417422004237, doi:https://doi.org/10.1016/j.eswa.2022.117005.

Eldan, R., Shamir, O., 2016. The power of depth for feedforward neural networks. 29th Annual Conference on Learning Theory 49, 907–940.

Fama, E., 1970. Efficient capital markets: A review of theory and empirical work. The Journal of Finance 25, 383–417.

Feng, J., Lu, S., 2019. Performance analysis of various activation functions in artificial neural networks. Journal of Physics: Conf. Ser. , 1237 022030.

Freyberger, J., Neuhierl, A., Weber, M., 2020. Dissecting characteristics nonparametrically. The Review of Financial Studies 33.

Friedman, J., 2001. Greedy function approximation: a gradient boosting machine. Annals of Statistics , 1189–1232.

Gareth, J.e.a., 2013. An introduction to statistical learning. Springer.

Geisser, S., 1975. The predictive sample reuse method with applications. Journal of the American Statistical Association 70, 320–328.

Ghayur, K., Heaney, R., Platt, S., 2019. Equity smart beta and factor investing for practitioners. John Wiley & Sons.

Glorot, X., Bengio, Y., 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.

Goyal, A., 2012. Empirical cross-sectional asset pricing: A survey. Financial Markets and Portfolio Management 26, 3–38.

Green, J., Hand, J., Zhang, X.F., 2013. The supraview of return predictive signals. Review of Accounting Studies 18, 692–730.

Gu, S., Kelly, B., Dacheng, X., 2020. Empirical asset pricing via machine learning. The Review of Financial Studies 33, 2223–2273.

Hanauer, M., Kalsbach, T., 2023. Machine learning and the cross-section of emerging market stock returns. Emerging Markets Review 55.

Harvey, C., Liu, Y., 2019. A census of the factor zoo. Available at SSRN: https://ssrn.com/abstract=3341728 .

Harvey, C., Liu, Y., Zhu, H., 2016. ...and the cross-section of expected returns. The Review of Financial Studies 29, 5–68.

Hou, K., Xue, C., Zhang, L., 2017. Replicating anomalies. NBER Working Papers 23394.

Hsu, J., Liu, X., Viswanathan, V., Xia, Y., 2022. When smart beta meets machine learning and portfolio optimization. Journal of Beta Investment Strategies 8.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv (1502.03167).

Ishwaran, H., Gerds, T., Kogalur, U., Moore, R., 2014. Random survival forests for competing risks. Biostatistics 15, 757–773.

Jacobs, H., Müller, S., 2020. Anomalies across the globe: Once public, no longer existent? Journal of Financial Economics 135, 213–230.

Jensen, T., Kelly, B., Pedersen, L., 2023. Is there a replication crisis in finance? The Journal of Finance 78, 2465–2518.

Jokanovic, F., Poitras, G., 2001. Does god practive a random walk? the "financial physics" of a 19th century forerunner, jules regnault. European Journal of the History of Economic Thought 8, 332–362.

Kelly, B., Malamud, S., Zhou, K., 2022. The virtue of complexity in return prediction. NBER Working Papers 20217.

Kelly, B., Xiu, D., 2023. Financial machine learning. Available at SSRN: https://ssrn.com/abstract=4501707 .

Kingma, D., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv 1412.6980.

Kuncheva, L., Whitaker, C., 2003. Measures of diversity in classifier ensembles. Machine Learning , 181–207.

LaValle, S.M., Branicky, M.S., Lindemann, S.R., 2004. On the relationship between classical grid search and probabilistic roadmaps. The International Journal of Robotics Research 23, 673–692.

Ledoit, O., Wolf, M., 2003. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. Journal of Empirical Finance 10, 603–621.

Ledoit, O., Wolf, M., 2004. A well-conditioned estimator for large-dimensional covariance matrices. Journal of Multivariate Analysis 88, 365–411.

Ledoit, O., Wolf, M., 2022. The power of (non-)linear shrinking: A review and guide to covariance matrix estimation. Journal of Financial Econometrics 20, 187–218.

Lee, S., 2020. Hyperparameter optimization for forecasting stock returns. arXiv Preprint (2001.10278).

Lewellen, J., 2015. The cross-section of expected stock returns. Critical Finance Review 4, 1–44.

Li, J., Huang, J.S., 2020. Dimensions of artificial intelligence based on the integrated fear acquisition theory. Technology in Society 63, 101410.

Linnainmaa, J., Roberts, M., 2018. The history of the cross-section of stock returns. The Review of Financial Studies 31, 2606–2649.

Ma, Y., Han, R., Wang, W., 2021. Portfolio optimization with return prediction using deep learning and machine learning. Expert Systems with Applications 165, 113973. URL: https://www.sciencedirect.com/science/article/pii/S0957417420307521, doi:https://doi.org/10.1016/j.eswa.2020.113973.

Maiti, M., 2019. A critical review on evolution of risk factors and factor models. Journal of Economic Surveys 34, 175–184.

Masters, T., 1993. Practical neural network recipes in c+. Academic Press .

McLean, D., Pontiff, J., 2016. Does academic research destroy stock return predictability? The Journal of Finance 71, 5–32.

Orimoloye, L., Sung, M.-C., M.T., Johnson, J., 2019. Comparing the effectiveness of deep feedforward neural networks and shallow architectures for predicting stock price indices. Expert Systems with Applications 139, 112828.

Patton, A., Timmermann, A., 2010. Monotonicity in asset returns: New tests with applications to the term structure, the capm, and portfolio sorts. Journal of Financial Economics 98, 605–625.

Piotroski, J., So, E., 2012. Identifying expectation errors in value/glamour strategies: A fundamental analysis approach. The Review of Financial Studies 25, 2841–2875.

Probst, P., Wright, M., Boulesteix, A.L., 2019. Hyperparameters and tuning strategies for Random Forests. arXiv:1804.03515 , 1–19.

Romano, J., Wolf, M., 2013. Testing for monotonicity in expected asset returns. Journal of Empirical Finance 23, 93–116.

Schnaubelt, M., 2019. A comparison of machine learning model validation schemes for non-stationary time series data. Technical Report. FAU Discussion Papers in Economics.

Spinu, F., 2013. An algorithm for computing risk parity weights. Available at SSRN: https://ssrn.com/abstract=2297383 .

Tashman, L., 2000. Out-of-sample tests of forecasting accuracy: An analysis and review. International Journal of Forecasting 16, 437–450.

Toochaei, M., Moeini, F., 2023. Evaluating the performance of ensemble classifiers in stock returns prediction using effective features. Expert Systems with Applications 213.

WEF, 2023. Jobs of Tomorrow: Large Language Models and Jobs. Technical Report. World Economic Forum.

Wright, M., Ziegler, A., 2017. Ranger: A fast implementation of random forests for high dimensional data in C++ and R. Journal of Statistical Software 77, 1–17.

Wu, Z., Yang, L., Fei, Y., Wang, X., 2023. Regularization methods for sparse esg-valued multi-period portfolio optimization with return prediction using machine learning. Expert Systems with Applications 232, 120850. URL: https://www.sciencedirect.com/science/article/pii/S0957417423013520, doi:https://doi.org/10.1016/j.eswa.2023.120850.

Zhao, T., Ma, X., Li, X., Zhang, C., 2023. Asset correlation based deep reinforcement learning for the portfolio selection. Expert Systems with Applications 221, 119707. URL: https://www.sciencedirect.com/science/article/pii/S0957417423002087, doi:https://doi.org/10.1016/j.eswa.2023.119707.

Appendix A Tables and Figures

Table 1
Summary statistics of out-of-sample errors. This table displays the following summary statistics of out-of-sample errors: Root Mean Squared Error (RMSE), Quantile (0.05), Kurtosis (Kurt) and Skewness (Skew). Metrics closer (further away) to (from) zero are in **bold** (underlined).

Model	RMSE	Q05	Kurt	Skew
OLS	22.15	-29.85	12.70	**1.78**
ENET	22.13	-29.85	12.74	1.81
RF	22.03	-30.92	12.87	1.81
XGB	22.06	-29.46	13.11	1.79
ANN1	22.09	**-29.03**	**12.66**	1.79
ANN2	22.08	-30.01	12.83	1.80
ANN3	22.12	-29.76	12.77	1.80
E_ANN	22.07	-29.55	12.80	1.81
E_NL	22.03	-29.76	12.93	1.83
E_A	**21.95**	-29.67	13.10	1.83

Table 2: Out-of-sample predictive performance. This table represents performance metrics for each model. Best (worst) R^2_{OOS}, Accuracy and MAPE are in **bold** (underlined).

Model	R^2_{OOS}	Accuracy	MAPE
OLS	-0.12	54.41	11.83
ENET	-0.01	54.38	11.74
RF	0.95	50.72	3.09
XGB	0.62	51.50	3.30
ANN1	0.38	52.75	1.82
ANN2	0.47	51.04	1.74
ANN3	0.14	48.48	2.83
E_ANN	0.56	50.97	**1.72**
E_NL	0.97	50.24	2.19
E_A	**1.61**	**53.23**	3.66

Table 3: Out-of-sample sample net summary statistics. This table represents out-of-sample net summary statistics calculated for ML portfolios and benchmarks. Annualized returns, standard deviation, Sharpe Ratio, Skewness and Excess Kurtosis are displayed.

Portfolio	Ann. Return	Std. Dev.	Sharpe	Skewness	Kurtosis
EW	0.294	0.282	1.04	-0.038	6.193
CW	0.245	0.271	0.91	-0.241	5.999
CS	0.327	0.294	1.11	0.165	6.873
SW	0.350	0.300	1.17	0.138	6.208
RP_SAM	0.214	0.252	0.85	-0.260	5.362
RP_PCA	0.274	0.262	1.05	-0.265	5.744
RP_S	0.242	0.258	0.94	-0.246	5.780
MTO_UNC_SAM	0.270	0.280	0.98	-0.161	5.010
MTO_UNC_PCA	0.324	0.309	1.05	0.149	6.208
MTO_UNC_S	0.315	0.293	1.08	-0.163	5.586
MTO_CON_SAM	0.257	0.260	1.05	-0.259	5.437
MTO_CON_PCA	0.294	0.270	1.09	-0.360	5.260
MTO_CON_S	0.285	0.270	1.05	-0.116	5.644
IBOV	0.151	0.247	0.61	-0.499	5.700
FUNDS_MEAN	0.116	0.226	0.52	-0.517	6.016
FUNDS_Q75	0.197	0.239	0.83	-0.303	5.144

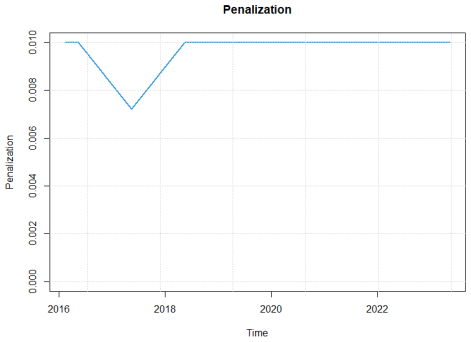
Table 4: Out-of-sample downside metrics. This table represents out-of-sample downside-risk metrics for ML portfolios and benchmarks. Expected tail loss, Semi-Deviation, Sortino and Rachev Ratios are displayed.

Portfolio	ETL	Semi-Dev.	Sortino	Rachev
EW	-0.169	0.056	0.562	1.513
CW	-0.179	0.055	0.479	1.400
CS	-0.155	0.057	0.614	1.622
SW	-0.159	0.058	0.645	1.659
RP_SAM	-0.164	0.052	0.451	1.367
RP_PCA	-0.171	0.053	0.552	1.494
RP_S	-0.169	0.052	0.497	1.404
MTO_UNC_SAM	-0.167	0.056	0.521	1.440
MTO_UNC_PCA	-0.165	0.060	0.586	1.583
MTO_UNC_S	-0.181	0.059	0.571	1.576
MTO_CON_SAM	-0.167	0.053	0.523	1.444
MTO_CON_PCA	-0.176	0.056	0.564	1.421
MTO_CON_S	-0.165	0.054	0.567	1.576
IBOV	-0.180	0.051	0.323	1.216
FUNDS_MEAN	-0.171	0.047	0.272	1.101
FUNDS_Q75	-0.156	0.049	0.433	1.337

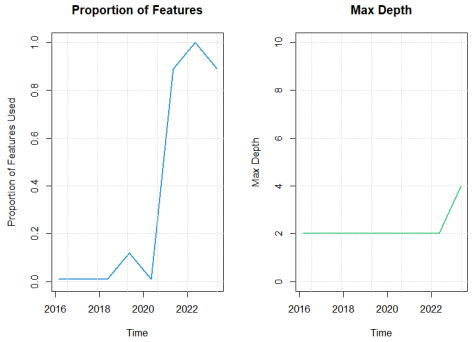
Table 5: Out-of-sample benchmark-relative net performance. This table represents out-of-sample net active performance metrics for ML portfolios and benchmarks. Annualized active returns, tracking errors, Information Ratios and CAPM alphas and t-stats are displayed.

Portfolio	Act. Ret.	TE	Info Ratio	Alpha	T-Stat
EW	0.142	0.105	1.35	0.001	2.98
CW	0.094	0.066	1.43	0.006	3.10
CS	0.176	0.114	1.55	0.012	3.33
SW	0.199	0.134	1.49	0.013	3.24
RP_SAM	0.064	0.039	1.63	0.005	3.67
RP_PCA	0.123	0.055	2.22	0.008	4.88
RP_S	0.091	0.054	1.68	0.006	3.75
MTO_UNC_SAM	0.119	0.105	1.13	0.008	2.58
MTO_UNC_PCA	0.173	0.171	1.01	0.013	2.36
MTO_UNC_S	0.164	0.128	1.28	0.011	2.85
MTO_CON_SAM	0.106	0.056	1.90	0.007	4.21
MTO_CON_PCA	0.143	0.069	2.07	0.010	4.52
MTO_CON_S	0.134	0.069	1.94	0.009	4.23
FUNDS_MEAN	-0.035	0.056	-0.62	-0.002	-0.94
FUNDS_Q75	0.046	0.048	0.97	0.004	2.68

Figure 1
Hyperparameter choice for models. The figure represents hyperparameter choices at each rebalancing.

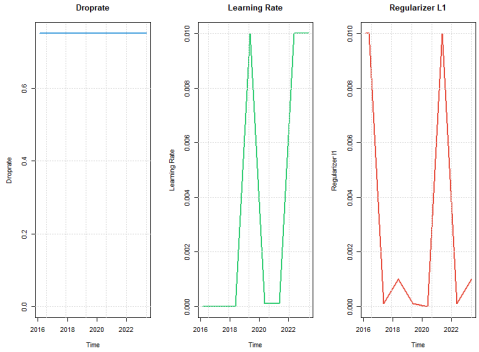


(a) ENET

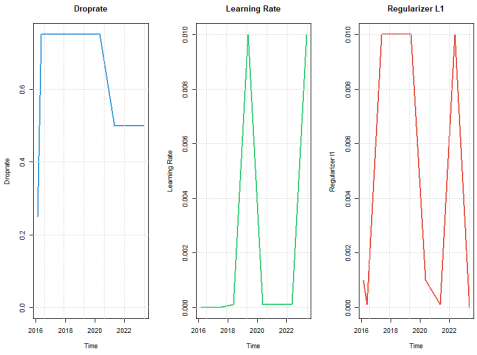


(b) RF

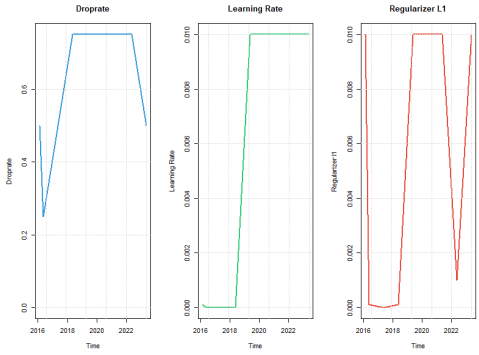
(c) XGB



(d) NN1

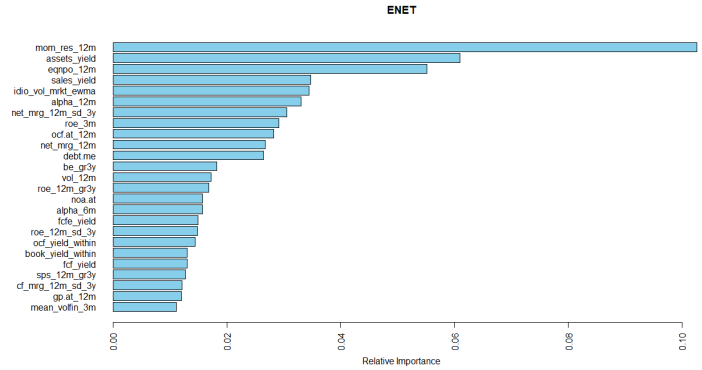


(e) NN2

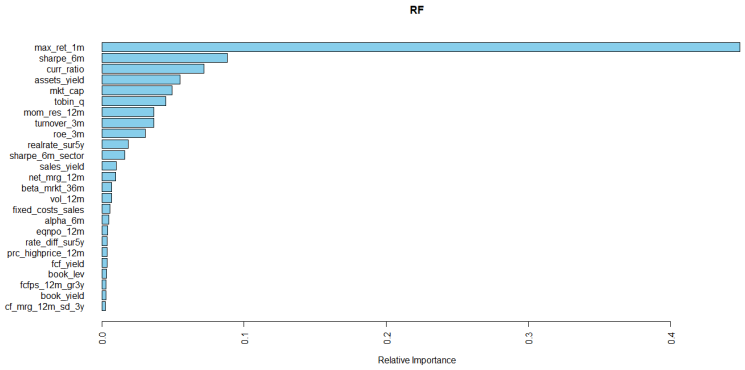


(f) NN3

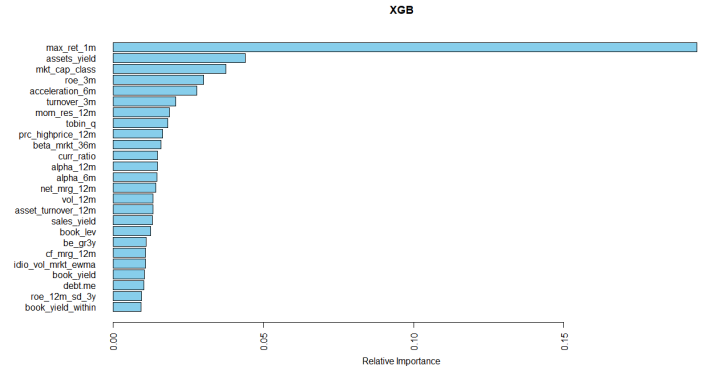
Figure 2
Feature importance. Each subfigure represents relative variable importance in each model.



(a) ENET



(b) RF



(c) XGB

Figure 3
Risk-return. The figure represents risk-return scatter-plot.

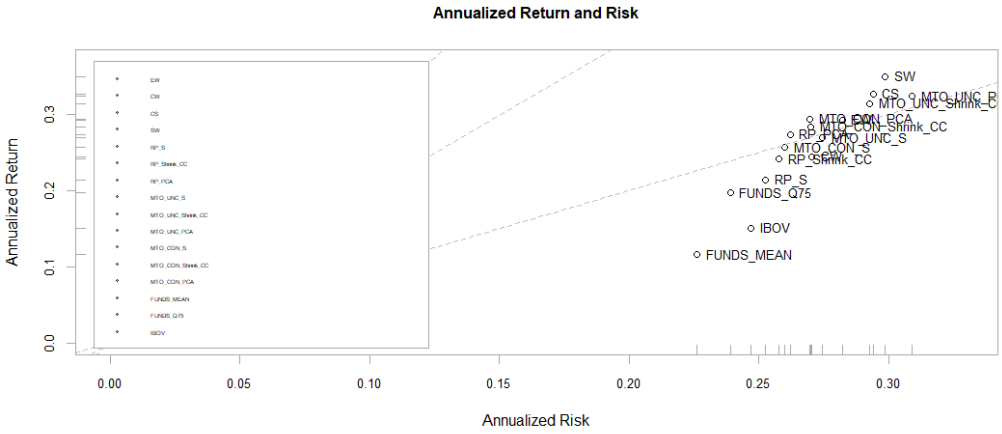


Figure 4
Box-plots. The figure represents box-plots of ML portfolios and benchmarks.

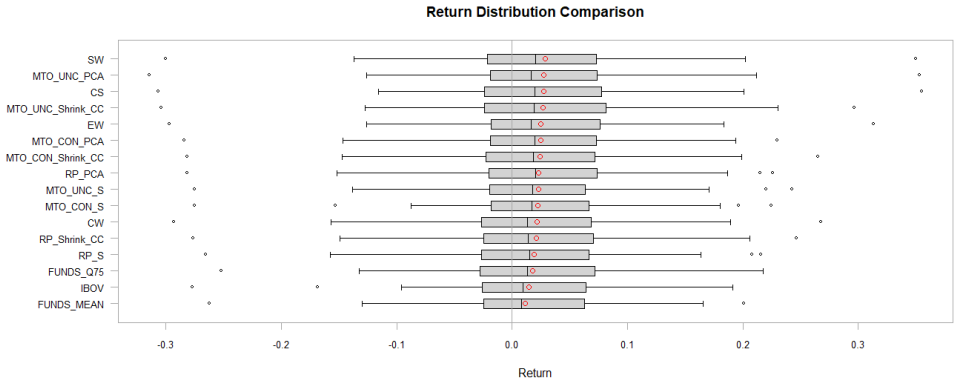


Figure 5
Cumulative Net Relative Returns. The figure represents cumulative net benchmark-relative performance of ML portfolios and equity funds.

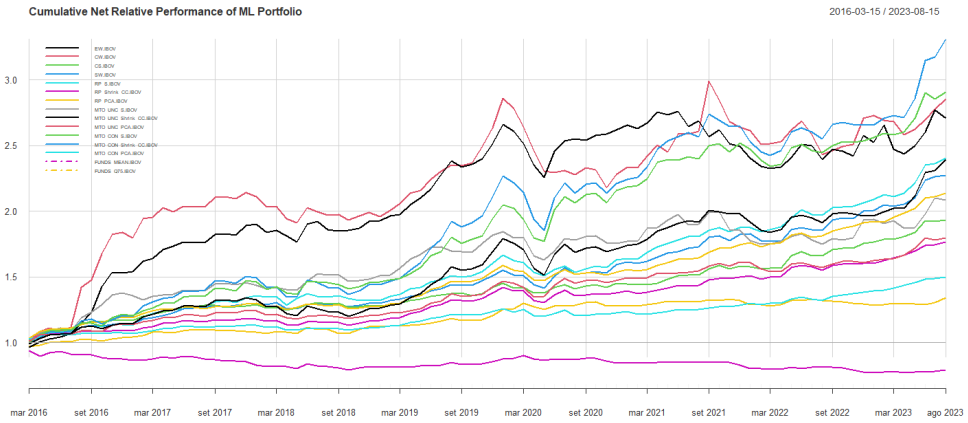


Figure 6
Capture Ratios of Active Strategies. The figure represents upside and downside benchmark-relative capture of ML portfolios and active managers.

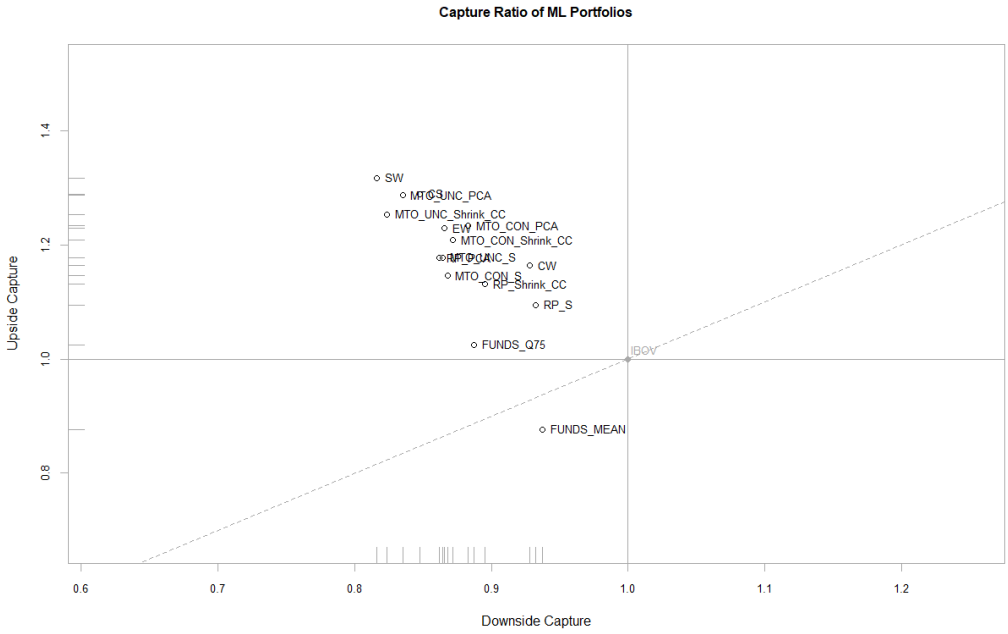


Table 6
Variables Definitions - Part 1. This table shows predictors, their respective theme and related study.

Definition	Name	Citation	Cluster
6m Acceleration	*acceleration_6m*	Gettleman and Marks (2006)	Momentum
6m Alpha	*alpha_6m*	Hünh and Scholz (2018)	Momentum
12m Alpha	*alpha_12m*	As above	Momentum
Asset Growth	*assets_gr3y*	Cooper et al. (2008)	Investment
Asset Turnover	*asset_turnover_12m*	Haugen and Baker (1996)	Quality
Assets Yield	*assets_yield*	Fama and French (1992)	Value
Book Growth	*be_gr3y*	Richardson et al. (2005)	Investment
Book Leverage	*book_lev*	Fama and French (1992)	Leverage
Beta - Market	*beta_mrkt_36m*	Fama and MacBeth (1973)	Low Risk
Book Yield	*book_yield*	Rosenberg et al. (1985)	Value
CAPEX Growth	*capex_gr3y*	Anderson and Garcia-Feijoo (2006)	Investment
Cash Flow Margin	*cf_mrg_12m*	Huang (2009)	Profitability
Cash Flow Volatility	*cf_mrg_12m_sd_3y*	As above	Quality
Commodities Price Surprise[1]	*commodities_sur_3y*	Brooks et al. (2016)	Macro
Correlation - Market	*corr_mrkt_36m*	Asness et al. (2020)	Low Risk
Current Ratio	*curr_ratio*	Ou and Penman (1989)	Quality
Debt-to-Market Equity	*debt.me*	Penman et al. (2007)	Leverage
Dividend Yield	*dps_yield*	Litzenberger and Ramaswamy (1982)	Value
Dividend Stability	*dy_med_36m*	Owain et al. (2000)	Value
Dividend Growth	*dps_12m_gr3y*	Maio and Santa-Clara (2015)	Profit Growth

Table 7
Variables Definitions - Part 2. This table shows predictors, their respective theme and related study.

Definition	Name	Citation	Cluster
Dividend Surprise	dps_12m_sur3y	Guo et al. (2023)	Profit Growth
EBITDA Yield	$ebitda\_yield$	Loughran and Wellman (2011)	Value
Earnings Yield	eps_yield	Basu (1983)	Value
Earnings Growth	eps_12m_gr3y	Asness et al. (2018)	Profit Growth
Earnings Surprise	eps_12m_sur3y	Foster et al. (1984)	Profit Growth
Equity Ney-Payout	$eqnpo\_12m$	Daniel and Titman (2006)	Value
Fixed Costs to Sales	$fixed\_costs.sales$	Gorodnichenko and Weber (2016)	Leverage
Free-Cash Flow to Invested Capital	$fcf.invcap\_12m$	Bouchard et al. (2019)	Profitability
Free Cash-Flow to Invested Capital Growth	$fcf.invcap\_12m\_gr3y$	Bouchard et al. (2019)	Profit Growth
Free-Cash Flow Yield	$fcf.yield$	Lakonishok et al. (1994)	Value
Free-Cash Flow Growth	$fcfps\_12m\_gr3y$	Jansen (2021)	Profit Growth
Free-Cash Flow Surprise	$fcfps\_12m\_sur3y$	Mao and Wei (2016)	Profit Growth
Free-Cash Flow to Equity to Book	$fcfe.be\_12m$	Bouchard et al. (2019)	Profitability
Free-Cash Flow to Equity Yield	$fcfe\_yield$	Lakonishok et al. (1994)	Value
Free-Cash Flow to Firm Yield	$fcff\_yield$	As above	Value
F-Score	$fscore$	Piotroski (2000)	Quality
Gross Profits to Assets	$gp.at\_12m$	Novy-Marx (2013)	Profitability
Idiosyncratic Volatility	$idio\_vol\_mrkt\_ewma$	Ali et al. (2013)	Profitability

Table 8
Variables Definitions - Part 3. This table shows predictors, their respective theme and related study.

Definition	Name	Citation	Cluster
Illiquidity	$mean\_volfin\_3m$	Dino (2023)	Size
Interest Rate Differential Surprise	$rate\_diff\_sur5y$	Lioui and Maio (2014)	Macro
Interest Rate Surprise	$realrate\_sur5y$	As above	Macro
Long-term Reversal	ret_60m_l1y	DeBondt and Thaler (1985)	Reversal
Market Cap	mkt_cap	Banz (1981)	Size
Max Return 1m	max_ret_1m	Bali et al. (2011)	Momentum
Monthly Inflation Surprise	$ipca\_monthly\_sur\_5y$	Duarte (2013)	Macro
Net Margin	net_mrg_12m	Soliman (2008)	Profitability
Net Margin Growth	$net\_mrg\_12m\_gr3y$	Asness et al. (2018)	Profit Growth
Net Margin Volatility	$net\_mrg\_12m\_sd\_3y$	Francis et al. (2004)	Quality
Net Operating Assets Growth	noa_gr3y	Richardson et al. (2005)	Investment
Net Operating Assets to Assets	$noa.at$	Hirschleifer et al. (2004)	Accruals
Operating Accruals	$oaccruals.at$	Sloan (1996)	Accruals
Operating Cash-Flow Yield	ocf_yield	Lakonishok et al. (1994)	Value
Operating Cash-Flow to Assets	ocf_at_12m	Bouchard et al. (2019)	Profitability
Operating Leverage	ope_lev	Novy-Marx (2013)	Quality
Payout	$payout\_12m$	Asness et al. (2018)	Quality

Table 9
Variables Definitions - Part 4. This table shows predictors, their respective theme and related study.

Definition	Name	Citation	Cluster
Price to Max Price 12m	$prc\_highprice\_12m$	George and Hwang (2004)	Reversal
Residual Momentum	mom_res_12m	Blitz, Huij and Martens (2011)	Momentum
Return on Equity	roe_12m	Haugen and Baker (1996)	Profitability
Return on Equity - Quarterly	roe_3m	As above	Profitability
Return on Equity Growth	roe_12m_gr3y	Asness et al.	Profit Growth
Return on Equity - Volatility	$roe\_12m\_sd\_3y$	Francis et al. (2004)	Profitability
Sales Yield	$sales\_yield$	Barbee et al. (1996)	Value
Sales Growth	sps_12m_gr3y	Abarbanell and Bushee (1998)	Profit Growth
Sales Surprise	$sps\_12m\_sur\_3y$	Jegadeesh and Livnat (2006)	Profit Growth
Seasonality Years 1-7	$seas\_1\_7$	Heston and Sadka (2008)	Seasonality
Seasonality Years 2-5	$seas\_2\_5$	As above	Seasonality
Sector Book Yield	$book\_yield\_sector$	Asness et al. (2000)	Value
Sector-Adjusted Book Yield	$book\_yield\_within$	As above	Value
Sector Dummies	$'sector\_name'$	Gu et al. (2020)	Sector
Sector Operating Cash-Flow Yield	ocf_yield_sector	Asness et al. (2000)	Value
Sector-Adjusted Operating Cash-Flow Yield	ocf_yield_within	As above	Value

Table 10
Variables Definitions - Part 5. This table shows predictors, their respective theme and related study.

Definition	Name	Citation	Cluster
Sector-Adjusted Size	$size\_within$	Asness et al. (2000)	Size
Sector Sharpe 6m	$sharpe\_6m\_sector$	Moskowitz and Grinblatt (1999)	Momentum
Sharpe 12m	$sharpe\_12m$	Daniel and Moskowitz (2016)	Momentum
Sharpe 6m	$sharpe\_6m$	As above	Momentum
Sharpe Ewma	$sharpe\_ewma$	As above	Momentum
Skewness 1y	$skew\_1y$	Bali et al. (2016)	Skewness
Skewness 3y	$skew\_3y$	As above	Skewness
Short-term Reversal	ret_1m	Jegadeesh (1990)	Reversal
Volatility of Liquidity	sd_volfin_6m	Chordia et al. (2001)	Size
Tobin Q	$tobin\_q$	Freyberger (2020)	Value
Total Debt Growth	$total\_debt\_gr3y$	Lyandres et al. (2008)	Leverage
Turnover	$turnover\_3m$	Datar et al. (1998)	Size
Trading Volume	$mean\_volfin\_3m$	Chordia et al. (2001)	Size
Unexplained Volume	$qtt\_1m\_sur\_3y$	Garfinkel (2009)	Size
Volatility	vol_12m	Ang et al. (2006)	Low Risk
Yearly Inflation Surprise	$ipca\_yearly\_sur\_5y$	Duarte (2013)	Macro

Table 11
Candidate values for hyperparameters. This table describes candidate values for grid-search.

λ^{v}	ENET	RF	XGBoost	ANN1-ANN3
λ^{ψ}	$10^{-[4,2]}$	-	-	-
λ^{ρ}	0.5	-	-	-
λ^{D}	-	(2, 4, 8, 10)	(2, 4, 8, 10)	-
$\lambda^{T}\lambda^{F}$	-	500	500	-
λ^{R}	-	(0.01, 0.1, ..., 1.0)	(0.50, 0.75)	(0.25, 0.50, 0.75)
$\lambda^{B}\lambda^{S}$	-	1	1	-
λ^{γ}	-	-	(0, 2, 5)	-
λ^{Λ}	-	-	(2, 5)	-
λ^{η}	-	-	(0.001, 0.005, ..., 0.02)	$10^{-(2,3,4,5)}$
λ^{ϕ}	-	-	50	5
λ^{ω}	-	-	-	$10^{-(2,3,4,5)}$
λ^{δ}	-	-	-	0
λ^{b}	-	-	-	512
λ^{e}	-	-	-	100
λ^{ι}	-	-	-	10