Predicting Equity Returns with Forecast Combinations of Deep Learning and Ensemble Methods*

Eike-Christian Brinkop† **Emese Lazar**‡ **Marcel Prokopczuk**§

December 11, 2023

ABSTRACT

We analyse forecast combination methods in the context of machine learning to predict equity returns. Whilst individual models lack robustness, forecast combinations over two levels display stability and Sharpe ratios of up to 3.06. We use decision trees in genetic algorithms to analyse the structure of variable influence. The impact of these variables displays inconsistencies and shows variations across different models and data. We propose a new performance measure for risk premium forecasts which leads to more robust evaluations than existing performance measures such as R^2, whilst providing economic interpretability. This measure can be linked to the advantages models offer for portfolio choice.

JEL: C51, C52, C53, C55, G12, G17.
Keywords: Equity Return Prediction, Forecast Combination, Deep Learning

*We thank the participants of the 2023 IAAE conference in Oslo and the 2023 IRMC conference in Florence for their useful comments.

†ICMA Centre, Henley Business School, University of Reading, UK
e.brinkop@pgr.reading.ac.uk
‡ICMA Centre, Henley Business School, University of Reading, UK
e.lazar@icmacentre.ac.uk
§School of Economics and Management, Leibniz University Hannover, Germany
prokopczuk@fcm.uni-hannover.de

1 Introduction

Challenges at forecasting equity returns include the large variety of variables to consider as well as the high level of noise in these returns. Thus, machine learning models built for risk premium predictions in the equity market often suffer from instability and overfitting on pseudo patterns. Sophisticated models lead to improvements of predictive performance and performance of constructed portfolios compared to traditional models struggling to cope with many input features or nonlinear effects. At the same time, such models attempt to limit the computational demand by using regularisation. Nonetheless, their high parameterisation leads to convergence to local optima in the training data, that are likely sub-optimal out-of-sample.

In this paper, we study the advantages of using forecast combinations in the context of machine learning methods for return prediction in the US equity market on two levels. Our primary hypothesis is that forecast combinations increase the stability of models out-of-sample and improve the results, both on a predictive and a portfolio level. We use a forecast combination methodology that stabilises model outputs on two levels, namely within and across model families. Our forecast combinations improve Sharpe ratios at each level, resulting in Sharpe ratios of up to 3.06 after the two levels of forecast combination. We compare naive averaging and a new method of combining forecasts by the inverse standard deviation of proposed point forecasts. We find that this new method excels in combining forecasts across different model families. In addition, we provide and test a new method to interpret machine learning models. Second, we implement an in-depth analysis of the influence of 94 return predictive signals (RPS) and measure their predictive performance for one-month US equity risk premia in depth. We do so by splitting the dataset by multiple RPS at a time and examining the predictability of the resulting groups of returns, using a genetic algorithm.

Our study has three major contributions. Our first contribution is that we perform a detailed analysis of forecast combinations for machine learning methods in the context of forecasting stock returns. Forecast combinations within and across different model architectures accomplish two things: (1) They lead to more stable point forecasts as compared to individual models. This makes them more suitable to construct investing strategies. They also lead to a reduced sensitivity to outliers of in-sample optima and are less reliant on specific patterns. (2) They allow for nonlinear combinations of individual models, improving overall performance. Gu et al. (2020) use naive averaging to stabilise neural networks out-of-sample. We propose a non-naive averaging approach, accounting for the volatility of individual forecasts. Moreover, we extend this by using different architectures within the neural network family to exploit different kernel functions and structures. Rapach et al. (2010) use an ensemble of linear regressions on their input factors and show that the combination of these is more powerful for prediction than a single "kitchen-sink"-type model. We extend both these studies by implementing a multi-level approach. After combining forecasts within a model family, in a first step, we also combine forecasts across stabilised versions of the models. We do so in order to exploit different multidimensional patterns naturally found by the different architectures. Our new method of forecast combination outperforms naive averaging.

Our second contribution is that we conduct a detailed analysis of the influences multiple variables have on forecasts. Increased computing power, advances in statistics and the financial literature have lead to an increasing number of variables being associated with future stock returns. Early factor models, such as Fama and French (1993), only use a few RPS. Harvey et al. (2016) document over 300 RPS that have been used to explain the cross-section of stock returns. Green et al. (2013) study the impact of 330 RPS on equity risk premia. Generally, the interpretation of the effects of these variables on the stock market is challenging. Inspired by the setup of Bryzgalova et al. (2020),

we propose using genetic algorithms on decision trees of our RPS, to detect multidimensional patterns in both marginal effects and importance of the company specific variables. Our approach is different in that we do not use these trees to price the assets, but to identify factor clusters that explain the predictability of equity returns. We identify market liquidity and size factors among the most influential variables. However, these results vary across models and are more noticeable when using a tree-architecture.

Our third contribution is a new measure predictive performance. The traditional R^2 does not provide intuition of portfolio performance advantages that stem from predicting equity returns. It lacks interpretability and economic intuition in highly noise-driven forecasting tasks. Therefore, we construct a predictive performance measure, that works by sorting predictions into quantiles and comparing these with the realised quantiles of the target variable. Our new measure adds robustness to cross-sectional shifts of the target variable, provides intuition for portfolio performance, and has economic interpretability.

The work most closely related to our study is Gu et al. (2020). We extend the research of Gu et al. (2020) by considering forecast combinations of machine learning methods on multiple levels. We select the strongest machine learning models, namely neural networks, tree ensemble methods and elastic net regressions. To illustrate our methodology, we provide a comparison, based on US stock market data between 1962 and 2020. By using over 840 variables and three million monthly company data points, we obtain several novel findings. Our portfolios outperform those of Gu et al. (2020) in terms of out-of-sample Sharpe ratios with values as high as 3.06 compared to values around 2.45 documented in their paper for a comparable portfolio. We attribute these results to the stabilising and performance enhancing effects of our forecast combination approach.

We observe that the excess returns obtained by the models revert towards the market return in the most recent period of our data sample. Through the availability of more computational resources to a wider public and advances in algorithms, trading systems have become more powerful and are able to price more components of individual stock returns. Nonetheless, pricing more factors and/or developing more sophisticated models or systems can lead to higher portfolio returns.

The literature on machine learning for predicting financial markets has grown recently. Heaton et al. (2017) use several methods and applications for neural networks, replicating the S&P500 index with only ten of its components, while Freyberger et al. (2020) use an adaptive group least absolute shrinkage and selection operator (LASSO) to reduce the dimension in a return prediction setting, reporting Sharpe ratios as high as 2.75. Campisi et al. (2023) compare the performance of machine learning methods at predicting the direction of S&P500 index returns. Random forests and bagging are their best classifiers, predicting the direction of the underlying index returns with over 80% accuracy. Bryzgalova et al. (2020) propose asset pricing trees in a stochastic discount factor (SDF) setting using asset baskets based on decision trees. Kelly et al. (2019) price US equity using instrumented PCA, a multi level regression system which can be expressed as a two layer neural network. Lin and Taamouti (2023) use a novel approach to predict stock returns using machine learning quantile regression, achieving Sharpe ratios of up to 0.76. Bianchi et al. (2021) apply several methods of machine learning to bond pricing using a dataset of over 100 macroeconomic variables. Bali et al. (2021) predict option returns using machine learning using cross-sectional predictors, increasing the out of sample prediction accuracy. Ban et al. (2018) use machine learning regularisation and validation methods for portfolio optimisation. Gu et al. (2021) use autoencoder networks to model latent factors of stocks in a no-arbitrage setting with HML portfolios earning

a Sharpe ratio of 1.53. Multiple neural networks are combined for equity pricing by Chen et al. (2021). Their generative adversarial neural network generates an annual Sharpe ratio of up to 2.6.

Cochrane (2011) presumes that a combination of time series portfolio sorts can yield additional information about the cross-section of returns. Welch and Goyal (2007) argue that influences of RPS in regression models vary over time and should be incorporated in a way that allows the regression model to account for that. Instead of a "kitchen-sink" regression model, combining all input factors in one regression, while Rapach et al. (2010) regresses the return on RPS individually and averages the output model, outperforming the "kitchen-sink" model leading to higher consistency and robustness. We adopt this method by averaging forecast results over multiple models. In a general context, LeBlanc and Tibshirani (1996) provide the basis for combined machine learning models. They argue that a combination approach both sidesteps the discussion of the best possible model and adds flexibility and nonlinearity to the model.

The rest of the paper is organised as follows. Section 2 reviews the machine learning methods used and introduces our methodology for hyperparameter tuning, forecast combination and performance measurement. In section 3, we analyse and discuss the results of our analysis regarding forecast combinations and variable influence. Section 4 concludes.

2 Methodology

In this section, we present our overarching approach and the models we use for return prediction. The models are elastic net regressions (ENET), gradient boosted regression trees (GBRT), random forests (RF), simple feed forward neural networks (FFNN), as well as methods of forecast combination both within and across models. Furthermore, we discuss the performance measures used to assess the models, including our own quantile-based measure.

2.1 Return Prediction and Estimation

This subsection specifies our general setup. The overarching objective is to predict one-month equity returns i.e.:

$$r_{i,t} = E_t(r_{i,t}) + \varepsilon_{i,t} \tag{1}$$

$$E_t(r_{i,t}) = g^*(z_{i,t-1}, \theta_t). \tag{2}$$

The return $r_{i,t}$ of asset i over the next month t consists of a model-dependent expected value $E_t(r_{i,t})$ and a noise term $\varepsilon_{i,t}$, which follows a zero mean normal distribution. We assume that the expected return can be expressed by a function $g(\cdot)$ with parameters θ_t and the input vector $z_{i,t-1}$ containing all information of the associated asset available at $t-1$. $g(\cdot)$ with its parameter vector θ_t needs to be estimated by minimising an objective function, in our case the mean squared error (MSE). In comparison to a linear approach, $g(\cdot)$ is extended by nonlinearity and regularisation in the model. This is important since a rich parameterisation often leads to overfitting in-sample; regularisation also controls the complexity of the model in order to reduce computational demand.

In order to update the function $g(\cdot)$ to recent patterns, we use rolling windows of training data. We use rolling windows of 25 years for training and validation with random cross validation, and 12-months testing windows. Validation data is used as an out-of-sample test for models to prevent overfitting and optimise some of the hyperparameters, which are non-weight parameters that contain model settings. The details of this setup are explained in the Supplementary Appendix B.1.

2.2 Elastic Net Regression

The first machine learning model is the elastic net regression (ENET), which is a penalised version of the classic linear regression. It penalises both the absolute value of parameters and the number of

non-zero parameters to make the model applicable for a large set of predictors:

$$L(\theta) = \underbrace{\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}||r_{it+1} - g^*(z_{it},\theta_t)||}_{MSE} + \underbrace{\frac{1}{2}\alpha(1-\lambda)\sum_{j=1}^{P}||\theta_j||}_{Ridge} + \underbrace{\alpha\lambda\sum_{j=1}^{P}|\theta_j|}_{LASSO}; \qquad \lambda \in [0,1], \alpha \geq 0. \tag{3}$$

N and T are the total assets and number of periods.The Ridge penalty introduced by Hoerl and Kennard (1970) expands the linear regression by a penalisation of the coefficients through the shrinkage parameter α. This shrinkage usually regulates the in-sample parameter concentration, which improves the out-of-sample performance, by avoiding over-weighting single variables.

Besides Ridge penalisation, ENET consists of a mechanism to reduce the number of variables, as described by Zou and Hastie (2005) and Friedman et al. (2010). This adds the LASSO penalty to deal with the problem of large sets of predictors. LASSO forces the model to use fewer non-zero predictors, which reduces the number of features that impact the prediction of returns. The ENET is based on the weighted sum of Ridge and LASSO penalisation using the weight λ. ENET is optimised using random cross-validation with α and λ as optimised hyperparameters.

2.3 Gradient Boosted Regression Tree

Second, we use the gradient boosted regression tree (GBRT), which is an ensemble of regression trees.[1] Boosting of trees originates in Schapire (1990) and Freund (1995), and was introduced with the purpose of optimising the out-of-sample performance of weak learners for classification tasks. Friedman et al. (2000) and Friedman (2001) extend this for applications in continuous regression settings. The basic idea of the GBRT is to estimate and stack multiple regression trees declining in

[1] A simplified example is shown in Figure C2 in the Supplementary Appendix.

influence on the overall model:

$$F_M(z,\theta_t) = \sum_{m=1}^{M} \nu^{m-1}\gamma_m h_m(z_m,\theta_m). \tag{4}$$

The GBRT F_M is a set of shallow decision trees $h_m(z_m,\theta_m)$, shrunken by a factor γ_m, which individually limits the influence of trees on the overall model and prevents the ensemble from overfitting. Additionally, the learning rate ν lowers the influence of the newly introduced trees. Every newly added tree h_m has to optimally improve the loss function of the ensemble:

$$h_m = \arg\min_{h} \sum_{i=1}^{n\subset N} L(y_i, F_{m-1}(z_t) + h(z_i), w_i), \tag{5}$$

with loss function L based on the MSE. Here, N is the training sample and n is any subset of N and y is the one month ahead stock return. Every additional tree tries to minimise the remainder of the residuals that the previous tree of weak learners, which are shallow trees, left behind. This minimisation problem is solved with the steepest descent method used to evaluate the decreasing step size γ_m:

$$F_m(z) = F_{m-1}(z) - \gamma_m \sum_{i=1}^{n\subset N} \nabla_{F_{m-1}} L(r_{i,t+1}, F_{m-1}(z_{i,t+1}), w_i), \qquad \text{with} \tag{6}$$

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n\subset N} L\left(r_{i,t+1}, F_{m-1}(z_i)\gamma \frac{\partial L(r_{i,t+1}, F_{m-1}(z_i), w_i)}{\partial F_{m-1}(z_i)}, w_i\right). \tag{7}$$

The steepest descent direction is the negative gradient of the loss function at the last shallow tree with the step size of the gradient γ, where $\nabla_{F_{m-1}}$ is the first derivative of F_{m-1} in (6). γ itself minimises the loss between the GBRT and the gradient at the last estimated shallow tree (7). The hyperparameters of GBRT are the maximum number of trees in the ensemble M, the depth of each of the shallow trees and the learning rate ν. These parameters are optimised during validation and using stochastic gradient descent on the validation scores.

2.4 Random Forest

Random forests (RF) are closely related to GBRT. An RF uses the properties of regression trees and adds randomness by bootstrapping these trees as described by Breiman (2001). For any new tree, the model draws a subset of the available data from the estimation sample. At any node in the tree, the model draws a subset of features for classification. The randomness in these selections regulates the diversity between the trees, so the RF is regularised to avoid loading up on a fraction of features or data points. The ensemble of trees reduces variance, stabilises the output and creates a continuous output space. An RF estimates a number of trees, which are then combined through naive averaging after estimation. This combination of randomness and averaging prevents the RF from overfitting, which is a common drawback of decision trees. The number of trees estimated and averaged and the maximum depth of the trees are optimised during validation.

2.5 Feed Forward Neural Network

The model with the most computational potential, which is most widely applied in the machine learning literature is the neural network. Hornik et al. (1989) describe neural networks as universal approximators for any functional interrelationship in a unified feature space. Neural networks send input data through a network of "thinking" neurons, inspired by natural intelligence findings in neurosciences. The architecture of "thinking" inside a neural network is complex and rarely transparent and therefore the results are hardly interpretable.

An FFNN is a multi-layer system of regressions.[2] It consist of an input layer, multiple hidden layers and an output layer. The input layer feeds the RPS into the model. The hidden layer and

[2]Figure C3 in the Supplementary Appendix presents a simplified architecture of FFNN in comparison to linear regressions.

output layer each process the information through a layer of regressions into a prediction for the target variable after the output layer.

A hidden layer in a neural network contains one or multiple neurons, with each transforming the results of the previous layer, similar to "synapses" in their neurobiological paragon.[3] The synapses are the connections between layer k and layer $k-1$ and contain the predictors θ as well as the selected functional form of the estimation. Each neuron in each layer additionally contains an individual constant.

A neuron in a hidden layer can be written as:

$$z_j^{(d)} = f_j^{(d)}(z^{(d-1)}, \theta_j^{(d)}) = act_j^{(d)} \left(\theta_{0,j}^{(d)} + \sum_{i=1}^{N_{d-1}} \theta_{i,j}^{(d)} * z_i^{(d-1)} \right). \tag{8}$$

The result of neuron j in layer d is determined by a function f_j of the input vector and the predictor vector θ_j of neuron j which determines the input variable $z_j^{(d)}$ for the next layer. N_{d-1} is the number of outputs of neurons in the previous layer. Every neuron can be seen as a regression with an activation function $act_j^{(d)}$ that applies nonlinearity to the output. We argue that different activation functions might be beneficial for different market states and combining them leverages their strengths while levelling out their weaknesses. As activation functions, we use exponential linear units (elu), sigmoid functions, tangens hyperbolicus (tanh) and softsign, respectively, given below:

$$elu(x) = \begin{cases} e^x - 1 & ,x < 0 \\ x & ,else \end{cases} \tag{9}$$

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{10}$$

[3]The main characteristic of feed forward is that a layer can only use results of the previous layer and transmit their own results to the next layer. Other popular architectures, including recurrent neural networks, work differently by connecting layers backward or neurons horizontally. See Rather et al. (2015) for an asset pricing application of recurrent neural networks. See Chen et al. (2021) for an example using LSTM neural networks, a major advancement in recurrent neural networks.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{11}$$

$$softsign(x) = \frac{x}{|x| + 1}. \tag{12}$$

Similarly to the hidden layers, the output layer is written as:

$$g(z^{(D-1)}, \theta^{(D)}) = \theta_0^{(D)} + \sum_j^{N_{D-1}} \theta_j^{(D)} * z_j^{(D-1)}, \tag{13}$$

with only one neuron and no activation and its output is the target variable.

For the structure of our FFNNs, we adapt the findings of Masters (1993), who suggests a so-called "pyramid arrangement" of neurons as an efficient arrangement, halving the number of neurons in each layer. To tackle potential overfitting and model selection bias, we use 4 different setups in our forecast combination FFNN. Each of these employs one of the activation function, a limited number of neurons of 64 for the first hidden layer and 3 layers. Our FFNN are optimised using backpropagation with the adaptive moment estimation method (ADAM) by Kingma and Ba (2017) as optimiser, which is an enhanced stochastic gradient descent. The parameters in a neural network are regularised by the ENET penalisation term for shrinkage and a penalisation for non-zero parameters, as described in section 2.2. We use inverse time decay for the learning rate and early stopping criteria in validation.

2.6 Forecast Combinations

We consider ensemble methods which combine forecasts of multiple learners. LeBlanc and Tibshirani (1996) argue that it might be impossible to find the single best model in a machine learning setting. As reasons, they state the complexity of models, the large number of hyperparameters, and the different computational mechanisms of these models. Forecast combinations offer a solution to

this problem, because they depend less on a single model by combining multiple models and model architectures.

Many machine learning applications suffer from model selection bias. The optimisation of all possible hyperparameters cannot be done during validation because the number of potential models would exceed computational limits by a large margin, which usually leads to a subset of these parameters being selected by trial and error.[4] The decision regarding architecture depends on the overall performance of the models. Therefore, some model selection bias bias cannot be fully eliminated. Additionally, survivorship bias arises from discarding sets of hyperparameters. We partially circumvent this problem by using model combinations to build forecasts.

Gu et al. (2020) use ensemble neural networks to stabilise the highly varying outputs of FFNN within their estimation windows by averaging their output. We also document problems with model stability, visible in Figure 5 and introduce a modified averaging approach. It is conjectured that the multidimensional patterns found by these models are different. Given several machine learning architectures that produce meaningful forecasts and lead to a good portfolio performance, their combination is able to achieve better predictive performance than the individual models. Elliott and Timmermann (2013) and Rapach et al. (2010) have focused on simple averages of model forecasts. Kelly and Pruitt (2013), Rossi (2018) and Rapach and Zhou (2020) show that model forecast combinations also work in the context of machine learning.

The simplest forecast combination is the naive forecast combination of different models, which can be used to reduce the effect of noisy outputs produced by neural networks. A problem of this method is that the variance of the predicted returns varies across different models, which leads to

[4]Usual validation methods include grid search and genetic algorithms over a set of activation functions, layers, depths of layers, LASSO parameters, batch size, dropout ratio, optimiser, normalisation method, loss function, size of rolling window, etc. Because these methods are all computationally expensive, we select the model settings by repeated trial and error over a small subsample of data.

the dominance of high-variation forecasts unfavourable for return prediction. Thus, a correction is required and we extend the simple averaging approach by the standard deviation-adjusted averaging method of the predicted stock returns of the models, leading to forecast combinations as given below:

$$\hat{r}_{i,t} = \mu_t + \sigma_t * \left(\sum_{j=1}^{J} \frac{\hat{r}_{i,j,t} - \mu_{j,t}(\hat{r}_{j,t})}{J\sigma_{j,t}(\hat{r}_{j,t})} \right) \tag{14}$$

$$\sigma_t = \sqrt{\frac{\sum_{j=1}^{J} \sigma_{j,t}^2(\hat{r}_{j,t})}{J}} \tag{15}$$

$$\mu_t = \frac{\sum_{j=1}^{J} \mu_{j,t}(\hat{r}_{j,t})}{J}, \tag{16}$$

where $\hat{r}_{j,t}$ is the vector of return predictions for model j in period t. μ_t and σ_t are the average mean and standard deviation of the individual models' means $\mu_{j,t}$ and standard deviations $\sigma_{j,t}$. Predictions with a high variation carry a lower weight than predictions with a lower variance, accounting for the difference in the amplitude of the different models' forecasts. We call this inverse standard deviation scaling (ISDS).

We use two levels of forecast combinations[5] The first level, within each model family, is primarily used to stabilise the model outputs. The high number of inputs and parameters leads to a high number of potential convergence points for the models in-sample, so their output is naturally unstable. By combining multiple trained models from a model family, this disturbance can be reduced.

The various architectures of machine learning methods have their own strengths and weaknesses. The discrete framework of the GBRT based on decision trees will give patterns that are different from the ones identified by the continuous approach in FFNN. While activation functions and

[5]An illustrative Figure is presented in the Supplementary Appendix C4.

regularisations in FFNN and ENET will likely remove some of the effects of outliers, the GBRT and RF methods are more robust, since they group all values over a certain threshold in one leaf of a tree. On the second level of forecast combinations, we combine the first-level combinations of each of the model families. We find that both levels improve portfolio performance measured by Sharpe ratio and overall return. Two second-level forecast combination models are proposed; (1) 'ALL' combines the forecasts of all four base model families forecast; (2) 'SOPH' combines the sophisticated models' forecasts, which are GBRT and FFNN.

2.7 Performance Measurement

We evaluate the performance of the models from an econometric point of view and in terms of portfolio performance.

2.7.1 Out-of-Sample R^2

The fit of the models can be compared using the out-of-sample R^2 given by:

$$R^2_{oos} = 1 - \frac{\sum_{i,t}\left(r_{i,t} - \hat{r}_{i,t}\right)^2}{\sum_{i,t}\left(r_{i,t}\right)^2}, R^2_{oos} \in (-\infty, 1], \tag{17}$$

for the model's predicted returns $\hat{r}_{i,t}$ in the test sample. The maximum value of this measure is 1, giving a perfect fit for all predicted values. Note that R^2_{oos} can be negative.[6]

Welch and Goyal (2007) argue that R^2_{oos} is not a robust measure of predictive accuracy because in market downturns, predictive power might signal bad explanatory power, despite the fact that

[6]Let ζ_k be the the individual prediction error $(r_k - \hat{r_k})$ for an observation k with $\zeta_k \sim \mathcal{N}(\mu_\zeta, \sigma^2_\zeta)$ and let r_k be its corresponding stock return observation with $r_k \sim \mathcal{N}(\mu_r, \sigma^2_r)$. By assumption, ζ_k and r_k fulfil $Cov(\zeta_k, r_k) = 0 \forall k$. Because $\sigma^2_x = \bar{x^2} - \bar{x}^2$ for a random variable x, we can rewrite R^2_{oos} reducing its fraction by N as $R^2_{oos} = 1 - \frac{\bar{\zeta}^2 + \sigma^2_\zeta}{\bar{r}^2 + \sigma^2_r}$. Then R^2_{oos} has 3 cases:

(1) $\bar{\zeta}^2 + \sigma^2_\zeta < \bar{r}^2 + \sigma^2_r \Rightarrow R^2_{oos} > 0$

(2) $\bar{\zeta}^2 + \sigma^2_\zeta = \bar{r}^2 + \sigma^2_r \Rightarrow R^2_{oos} = 0$

(3) $\bar{\zeta}^2 + \sigma^2_\zeta > \bar{r}^2 + \sigma^2_r \Rightarrow R^2_{oos} < 0.$

It follows that R^2_{oos} will be negative in a setting with mean values or standard deviations from predicted values differing largely from their realised counterparts.

the model distinguishes between winners and losers correctly. Rapach and Zhou (2013) find that forecasts with an out-of-sample predictive R^2_{oos} below 1% can still produce portfolios that perform in excess of the market. The authors also state that the explanatory power and the portfolio returns may lead to different conclusions about model performance.

2.7.2 Quantile Assignment Accuracy

To avoid these drawbacks, we propose new measures of predictive performance, which are more robust than the R^2_{oos} and also have economic interpretability. The starting point of these measures is to sort the actual returns and predicted returns of the model into quantiles and then compare the sortings. The sorting advantage has direct implications on the potential gains that can be obtained with an HML portfolio setting and it leads to 2 different measures. First, we obtain a measure of accuracy in terms of sorting predictions into the actual returns' quantile, and second, the deviation of the predicted quantile from the actual observation's quantile. Finally, these measures are transformed into sorting benefits that can be obtained in excess of sorting by chance. For this, we subtract from our measures the values that can be obtained via random assignments.

We call the first measure the Quantile Assignment Accuracy and we denote it by $QAA(Q')$; this the relative advantage in quantile sorting relative to a random assignment:

$$QAA_{Q'}(\hat{r}_t) = \begin{cases} \frac{qaa_{Q'}(\hat{r}_t)}{1-(\frac{1}{N_Q})} & ,if\ qaa_{Q'} \geq 0, \\ qaa_{Q'}(\hat{r}_t) * (1-N_Q) & ,\text{otherwise} \end{cases}, \quad QAA_{Q'} \in [-1,1], \text{with} \tag{18}$$

$$qaa_{Q'}(\hat{r}_t) = \sum_{\hat{r}_{i,t} \in Q'} \left(\frac{\tilde{a}_{i,t}}{N_{\hat{r}_{i,t} \in Q'}} \right) - \frac{1}{N_Q} \tag{19}$$

$$\tilde{a}_{i,t} = \begin{cases} 1 & ,if\ q(r_{i,t}) = q(\hat{r}_{i,t}), \\ 0 & ,\text{otherwise} \end{cases}, \tag{20}$$

where Q denotes the set of quantiles used and Q' is any subset of it with N_Q as the number of quantiles in Q. $N_{\hat{r}_{i,t}\in Q'}$ denotes the total number of returns in set Q'. $QAA_{Q'}(\hat{r}_t)$ measures the relative sorting advantage in the following way: A model forecasts $\hat{r}_{i,t}$. We compare the quantiles of the predictions $\hat{r}_{i,t}$ with the quantiles of $r_{i,t}$ in (20). The resulting measures $\tilde{a}_{i,t}$ for any subset of quantiles are summed up and divided by the number of returns in the quantiles of Q'. Next, we subtract the value that can be obtained by randomly sorting the predictions into quantiles, which is $\frac{1}{N_Q}$. In (18), because $qaa_{Q'}(\hat{r}_t) \in [-\frac{1}{N_Q}, (1-\frac{1}{N_Q})]$, we correct the value dependent on its sign. The result is a relative advantage of sorting compared to a random assignment of returns into quantiles. A value of one means that the predicted returns lead to a perfect sorting. A value of zero reflects the market return, implying that the prediction is no better than the market average.

The second of these measures we call Quantile Assignment Deviation and we denote it by $QAD_{Q'}$. This measures the deviation of the predicted and the actual quantiles:

$$QAD_{Q'}(\hat{r}_t) = \begin{cases} \frac{qad_{Q'}(\hat{r}_t)}{RAD(Q')} & ,if\ qad_{Q'} \geq 0, \\ \frac{qad_{Q'}(\hat{r}_t)}{R_{max}(Q')-RAD(Q')} & ,\text{otherwise} \end{cases}, \quad QAD_{Q'} \in [-1,1], \text{with} \tag{21}$$

$$RAD(Q') = \frac{1}{N_Q * N_{Q'}} \sum_{i\in Q} \sum_{j\in Q'} |i-j|, \tag{22}$$

$$qad_{Q'}(\hat{r}_t) = RAD(Q') - \frac{1}{N_{\hat{r}_{i,t}\in Q'}} \sum_{\hat{r}_{i,t}\in Q'} \left(q(r_{i,t}) - q(\hat{r_{i,t}})\right), \tag{23}$$

$$R_{max}(Q') = \frac{1}{N_{Q'}} \sum_{i\in Q'} |2i-(Q+1)|. \tag{24}$$

The advantage in assignment deviation $QAD_{Q'}$ is given by the average deviation by the predicted quantiles relative to the expected deviation under random assignment $RAD(Q')$. Since $qad_{Q'} \in [RAD(Q'), R_{max}(Q') - RAD(Q')]$, the standardisation depends on the sign of $qad_{Q'}$. The value of the deviation $RAD(Q')$ is the expected difference of quantiles in random sorting of assets to the proposed

quantiles. $qad_{Q'}$ is the mean deviation of quantiles for observations demeaned by $RAD(Q')$. The raw value (without demeaning) of $qad_{Q'}$ therefore lies between the value obtained by perfect sorting and the value obtained by the worst possible sorting $R_{max}(Q')$ as the minimum value. This value can be calculated by iterating over all existing quantiles and all considered quantiles, calculating their average distance. A value of –1 indicates the maximum quantile distance of predictions to their actual quantile. 0 is the expected value, whilst 1 indicates perfect sorting.

2.8 Variable Influence

We analyse the structure of factor influence for return prediction in the machine learning models. Following Green et al. (2013) and Harvey et al. (2016), we identify the most relevant factors in a machine learning setting. For this, we use two measures of variable influence.

The first measure of influence, called feature importance, is based on the differences of predictive performance based on the deciles of the explanatory variables. For this, we sort the companies into monthly deciles according to the 94 RPS. Next, we calculate the R^2_{oos} for the deciles across the dataset. For each variable, we subtract the lowest R^2_{oos} of all deciles from the highest. This difference describes the maximum of information difference that can be obtained by sorting the data by a specific variable. The higher the value, the higher is the information "left on the table" by the model.

Our second measure is based on a genetic algorithm similar to Bryzgalova et al. (2020) that sorts by multiple variables. We create a population of 100 sets of 3 RPS each, where in each set we successively sort the out-of-sample results into 50-50 baskets of predicted returns based on the RPS, resulting in 8 baskets for each set. Then, we compute the score differences in R^2_{oos} as described above and sort the sets from highest to lowest. A fraction (35) of theses sets survives to the next generation and mutates by changing one of its RPS to generate close relatives with a similar score.

New sets are drafted to fill up the initial population size of 100. After 35 generations, the set of survivors converges by not changing by more then 5%. Details of this algorithm are provided in the Supplementary Appendix B.2.

3 Empirical Analysis

In this section, we present the empirical analysis of stock prediction using the machine learning methods and ensemble combinations of these.

3.1 Data

The variables we forecast are the one-month total excess returns of US equity. We obtain US stock market data from CRSP for all companies available between July 1962 and December 2020.[7] Additionally, we obtain other stock market key figures from CRSP.[8] To compute excess returns, we use the three months US-treasury bill rates as the risk-free rate.

Our dataset comprises the same 94 variables that have been used by Gu et al. (2020) and originate in Green et al. (2013). We replace some of the calculations from Green et al. (2013) with alternative approaches, increasing their update frequency. Some RPS in the set of 94 variables require daily stock market data which we obtain from CRSP. In addition to this, we obtain financial statement data from Compustat. Additionally, we use the GICS industry groups to perform industrial demeaning. Of the 94 RPS, 6 are based on daily stock market data and 19 RPS are computed using monthly stock market data, each resulting in a monthly update. 52 RPS are computed using quarterly statements from Compustat, mostly with a quarterly update.[9] 5 RPS are based on annual financial statements and firm reports data and are updated annually. The sin-stock dummy SIN is based on

[7] We select July 1962 because it is the first month when data is available for more than 1000 companies in CRSP. Before then only very few companies have stock price data available.

[8] A full list of the variables can be seen in the Supplementary Appendix A.

[9] Market value related ratios are updated monthly with respect to the monthly market capitalisation.

the companies' industry classification and is not updated. The remaining 11 RPS are differences to industry averages of other variables.

In comparison to Gu et al. (2020), we increase the frequency of many of the variables from annually to a four-quarter rolling system to get as fast and actual updates as possible. To account for the actual report dates, we use the month end after the quarterly report date as given in Compustat, if it is available. Otherwise, or if the report date exceeds the regulatory range of 90 days, we use information from quarterly statements with a lag of three months, which is in line with the SEC's regulations for annual 10-K filing deadlines. Following the suggestions of Welch and Goyal (2007) and Gu et al. (2020), we also use 8 macroeconomic variables in our analysis. These are available on Amit Goyal's website and are listed in the Supplementary Appendix A.

In the following, we construct two different setups.[10] The first setup uses the target variable and the 94 RPS, we call this the *uninteracted* setup and denote it by (u). The second setup is the *interacted* setup which extends the 94 RPS by interactions with the macroeconomic variables and is denoted by (i). As interactions, we multiply the macroeconomic terms with the company specific variables and add the company specific variables as self representation, resulting in $(8+1)*94 = 846$ variables in one input data point $z_{i,t}$ for setup (i). The formulation of this model is closely related to the overarching models of Ferson and Harvey (1999) and Rosenberg (1974), which are also used by Gu et al. (2020).

We start the empirical analysis with the year 1987 as a testing sample. From there, we roll forward year by year until 2020, the last year of the sample. We estimate our 6 models using the uninteracted setup, and also estimate them using the interacted setup. These consist of the 4 models first-level

[10] All portfolios presented are equally weighted. The results for value weighted portfolios can be found in the Supplementary Appendix, and these are qualitatively consistent.

ISDS forecast combinations ENET, RF, GBRT, and FFNN as well as the 2 second-level forecast combinations ALL and SOPH from section 2.6.

3.2 Predictive Performance

Figure 1 shows the results for R^2_{oos} for the entire testing period between 1987 and 2020 for all of our models. We show the total sample R^2_{oos} as well as conditional the R^2_{oos} after ranking the assets by market capitalisation. The latter measures the differences in explanatory power by size. A lighter shade of the colour indicates that the respective model has been estimated using setup (u) while a darker colour indicates estimation based on setup (i).

All models estimated on setup (u) have a positive coefficient of determination out-of-sample. For (i), only the FFNN has a positive R^2_{oos}. Both forecast combination approaches ISDS and naive have very similar results by this comparison, with a slight advantage of naive averaging for neural networks. Generally, using the interacted setup produces worse results than the uninteracted setup. For all other models including the forecast combination models, the estimation on the sparser setup (u) benefits the explained variation. R^2_{oos} for smaller companies is generally better than for larger companies. Considering that in 2020, the largest 1000 companies on average account for over 90% of market capitalisation while only representing a sixth of the firms, this result does not surprise in a setting where each observation is equally weighted within training. The strongest models by R^2_{oos} are RF(u), GBRT(u), ALL(u) and SOPH(u) with slight differences between large and small companies. Second level forecast combinations provide improvements and have an R^2_{oos} above the average of the individual models independently of size, and outperform all their components for the full sample. ALL(u) has the highest score on explained variation.

Second, we put the predictive performances into context using the test proposed by Diebold and Mariano (2002) and report the results in Table 1. We find clear distinctions between the models. Notably, the forecast combination ALL(u) significantly outperforms all its components in terms of predictive performance and is generally the best performing model overall. SOPH(u), which removes ENET and RF, the less complex models, from the forecast combination, underperforms ALL(u) in this test. For the ALL(i), the Diebold-Mariano test only shows outperformance of its tree-architecture-based components. In both setups, removing the less sophisticated structures from the forecast combination costs predictive power. Of the base models, GBRT(u) has a comparative predictive advantage to all other models except for the forecast combinations.

Third, we do a predictive performance analysis for the QAA and QAD. Since we use an HML portfolio setting with decile portfolios, we compute this performance measurement for 10 deciles in the "Sample" quantiles setting and for just the lowest and the highest deciles, that is the HML quantiles. Results are shown in the top of Figure 2. All models give a general advantage in portfolio sorting by providing a 2% sorting advantage for all stocks on a monthly basis. Looking at the highest and lowest decile alone, the advantage gained is between 6% and 7% for HML portfolios. The deviation measure QAD improves by about 4% over all deciles and by 6% for the HML deciles.

The two forecast combination methods ISDS and naive perform very similarly again. The results obtained with the (u) setup generally outperform those obtained with the (i) setup, indicating that including interactive terms with macro variables does not provide prediction improvements over the cross section of returns. The only exception to this is FFNN. GBRT(u) leads to the best results overall in terms of predictive power as measured by quantiles. The two forecast combination approaches follow closely in second and third place. They outperform 3 of the 4 models in this setting and perform very close to their top component. RF(i) has by far the weakest performance

for sorting advantage, which is in line with the results on portfolio performance presented in section 3.4. Generally, these sorting measures are intuitive and are able to differentiate between different levels of portfolio performance achieved by the various models.

We compute our sorting measures QAA and QAD separately for the largest and smallest 1000 companies, with results presented in the bottom of Figure 2. The size effect is visible in this figure, showing that the sorting advantage is smaller for large companies than for small companies. The effect of the sorting is particularly strong for GBRT, ALL and SOPH. As before, the forecast combination performs best or second best compared with the corresponding individual models.

3.3 Predictive Performance of Forecast Combinations

Next, we provide selected results that highlight the advantages of forecast combinations for predictive performance within the 4 base models. Figure 3 presents the predictive performance of submodels within a model family compared with the forecast combination of models on setup (u).

Regarding R^2_{oos}, forecast combinations show an immediate advantage for ENET and GBRT, increasing their out-of-sample predictive accuracy (Figure 3 (a), (c)). For RF, the forecast combination underperforms all single submodels in the sample (Figure 3 (b)). One explanation is that RFs are essentially a naive averaging over hundreds of trees. For FFNN, the part of the architecture running on the sigmoid activation function (submodels 9–12) generally underperforms the other submodels (Figure 3 (d)). The first-level ISDS forecast combination offsets this behaviour. In this setting, the forecast combination stabilises the results over a range of submodels, and its performance is closer to the best model than to the worst performing model. Naive averaging produces a better R^2 for neural networks, although this advantage vanishes when looking at the sorting. Otherwise, the predictive performance of the ISDS and naive averaging is very similar.

The advantages of forecast combinations on the sorting of stocks are evident in Figure 3. For all models, the forecast combination of submodels outperforms either all submodels or performs second best in every category. This result is clearer for regression-based ENET and FFNN, and less evident for the ensembles of GBRT and RF. One explanation is that the ensemble architecture within each submodel of the classifiers GBRT and especially RF already removes most of the variation. The highly parameterised FFNNs particularly benefit from the forecast combination approach, as the combination not only stabilises the overall sorting, but also clearly outperforms all of the submodels for the deciles used to construct our long–short portfolios.

3.4 Portfolio Results

Next, we discuss the results regarding the performance of the machine learning portfolios. In line with our decile approach for QAA and QAD, we sort the returns into deciles based on the predicted return in each month in the out-of-sample period. Table 2 shows statistics on the performance of the ten decile portfolios and the HML portfolio results for our six methods in setup (u) for ISDS and naive averaging.[11] We report the monthly prediction average, mean return, standard deviation and annualised Sharpe ratio for the ten decile portfolios for each of the models.

The first objective for all models is to obtain a clear ordering of the decile portfolios for all models where a higher decile is associated with a higher average return. We also consider the Sharpe ratios of the decile portfolios, and we observe that the highest return deciles typically also have the highest Sharpe ratios. Generally, all models lead to decile portfolios with the ordering of the predicted decile returns corresponding to the ordering of the actual decile returns. The Sharpe ratio for the ISDS forecast combination HML portfolio is better than the naive counterpart for all models.

[11] We only present these for setup (u), because the performance is better. Setup (i), although underperforming in general, aligns qualitatively with the results of setup (u).

This forecast combination method leads to below average standard deviation, above average mean returns and the highest Sharpe ratios compared to their components. This is not the same for naive averaging. The two second layer naive forecast combinations outperform only FFNN in terms of Sharpe ratio for the high-minus low portfolios. The reason for this is that they produce more fluctuating portfolios, as expressed by their high standard deviations. ALL ISDS has a Sharpe ratio of 2.72 compared to 2.06 for ENET, 1.48 for RF, 2.31 for GBRT and 2.8 for FFNN. The highest value is 3.06 for the ISDS forecast combination of GBRT and FFNN, SOPH. Generally, using ISDS forecast combination makes it possible to improve on the Sharpe ratio, which is not possible for naive averaging. This is attributable mostly to the variance reducing effects of ISDS forecast combination. The weighting structure leads to assets being selected from its lower standard deviation components, while maintaining a high average return. These results emphasise the benefits of ISDS forecast combinations on a portfolio level and show that they are economically significant.

Using the decile portfolios, we construct HML portfolios. These are zero net investment portfolios short-selling the 10% stocks with the lowest predicted returns and buying stocks with the 10% highest predicted returns. Table 3 shows the performance measures for these HML portfolios out-of-sample. We present the results for setup (u) and the naive averaging compared to ISDS forecast combinations as well as the market performance. The monthly market return in this case is an equally weighted market portfolio of all available stocks.

All portfolios have a higher mean return than the market. ALL(u) and SOPH(u) exhibit a high monthly average return. The standard deviation of portfolios formed with ISDS is lower than the naive averaging of all models. As a result, the models with the highest monthly Sharpe ratios are the FFNN, GBRT and SOPH using ISDS forecast combinations. For naive averaging, FFNN has the

highest Sharpe ratio, because its standard deviation is lower than the one obtained with the forecast combinations.

The maximum drawdown (DDmax) measures the highest realised loss from any high-water mark. This is a measure of maximum loss at any point in time. Considering this, ENET(u) ISDS shows the best performance, since it has the lowest loss at only 12.24%. Again, ISDS performs better than the naive averaging, and the difference is economically large for the second level forecast combination methods ALL and SOPH. For the drawdown, ALL performs very close to ENET using the ISDS forecast combination. The ISDS forecast combinations also stabilise the risk profile of their components. The same goes for the other statistics such as the minimum and maximum one-month return, where forecast combinations have statistics that are very close to their best component. The same doesn't apply for the naive averaging, where the drawdowns and minimum returns are closer to their worst components. All portfolios produce very high monthly turnover ratios, which is taken into consideration when computing the monthly performance. Essentially, if this rate is multiplied with a transaction cost factor, for example 0.2%, then it can be deducted from the mean return, which will decrease the overall performance.

The systematic risk of the resulting portfolios, measured as the correlation with the market portfolio, is the only statistic for which the forecast combinations do not offer a clear advantage for ISDS. According to this measure, the best performing models are GBRT(u) and SOPH(u) while ALL(u) leads to a higher systematic risk. All portfolios have a statistically significant α, which validates their outperformance of the market with respect to the CAPM. We also regress the models' returns against common factors and find that all portfolios except for RF(i) earn α against both the Fama and French (1993) 3-factor model and against the Fama and French (2015) 5-factor model augmented by

Jegadeesh and Titman (1993)'s momentum factor. The returns of our portfolios can not be explained by common factor models.

Figure 4 shows the overall return charts for the highest and lowest decile portfolios of all models, both setups, and for ISDS and naive averaging. All machine learning long portfolios outperform the market, although there are differences in their overall performance. For (u), GBRT performs best among the first-level forecast combinations. For (i), the predictions made by FFNN form a better HML portfolio than all other base models. In both scenarios, the top two long portfolios are ISDS second-level forecast combinations. In setup (u), these are followed by the naive averaging second-level forecast combinations. In setup (i), FFNN outperforms the naive averaging forecast combinations for ALL and SOPH. For the short portfolios, only GBRT(u) has a similar performance to the forecast combinations. The long portfolio constructed with RF(i) is very close to the market portfolio, while the corresponding short portfolio fails to identify losing stocks.

There is a strong co-movement overall. The financial crisis of 2008 is visible across the board in terms of portfolio performance. In addition, the COVID-19 drop on the capital markets is also visible for all models and forecast combinations. Most of the portfolios fluctuate much stronger than the market in the early 2000s. This effect is stronger for the methods that are based on setup (i). Generally, the short positions are not consistent and do not have a visible trend since 2003, while the long portfolios still follow a long-term upward trend during the same time period. Only the forecast combinations based on setup (u) and GBRT are able to identify the worst 10% of the stocks post 2003.

The long positions show a weaker performance after 2003, gaining about 2/3 of their total return before the end of 2003. This indicates that the market is generally getting more efficient with respect to available information. It can be argued that the models used in this paper would not have been

realistically available to market players before 2003. The strong early over-performance can be explained by the fact that these models can cleverly process a lot of information in a timely manner, which was not possible to achieve in the 1990s and early 2000s.

All methods perform better using setup (u). This also indicates that macroeconomic data do not contain information about the cross-section of stock returns out-of-sample. More elaborate trading strategies based on, for example, a flexible long–short ratio based on macroeconomic variables might be able to extract more relevant data regarding the interaction between macroeconomic and stock return data. However, based on our results, it can be concluded that individual stock returns are not cross-sectionally affected by macroeconomic movements.

3.5 Portfolio Results for Forecast Combinations

We also analyse the effectiveness of forecast combinations on the resulting portfolios. Figure 5 shows the portfolio performance for long and short positions of the first-level forecast combinations. Forecast combinations inside the model family do not provide a benefit for random forests as shown in Figure5(b). One possible explanation is that the naive averaging approach already balances the variation. GBRT(u) provides a slight advantage through combining its individual models. The base model for GBRT, which is the one that comes from our cross-validation hyperparameter optimisation, is shown as sm1 in panel (c); this model proves to be the second-weakest submodel for long and short portfolios. The forecast combination outperforms all submodels by a small margin.

This effect becomes much stronger for the regression-based approaches ENET and FFNN. ENET shows an advantage of about 0.4 log returns over the dataset for the long portfolio, compared to its strongest submodel component sm1 as shown in panel (a). For neural networks, forecast combinations significantly improve the results of the long and short portfolios. The log return for

the HML portfolio increases by 2 for the entire test dataset. The forecast combination performs better than all of the submodels by a large margin. FFNNs are highly parameterised models and depending on the submodel, they use between 5000 and 8000 parameters. In such a parameter space, many local in-sample optima exist which may not be the global optimum out-of-sample. Forecast combinations are able to lower the negative effects of local optima. We find that ISDS and naive averaging are very similar in this comparison.

3.6 Influence of Return Predictive Signals

In this section we address the question of selecting the most significant RPS from the 94 RPS for the prediction of US equity returns using ISDS forecast combination. We split this question into two parts. The first part of this section discusses the selection of the variables which are most influential for the prediction of stock returns. The second part of the section addresses the selection of combinations of variables that are most informative for stock return prediction.

3.6.1 Direct Variable Influence

Using the methodology from Section 2.8, we obtain Figure 6 which shows the influence of the individual variables. For this analysis, we only use the uninteracted setup, since their R^2_{oos}s are better than the corresponding values obtained from the interacted setup. Each cell represents the potential increase in R^2_{oos} that could be gained by pre-sorting observations in deciles using the row variable and the column model. These values are generally higher for tree-based architectures (GBRT, RF) than for the regression methods (ENET, FFNN). The forecast combinations shift these values closer to FFNN than to any other model.

The top ranking variable by this measure is illiquidity, which might give an advantage in R^2_{oos} of about 0.6% on average. Illiquidity measures the change in the stock price relative to its dollar

trading volume. In the same group of variables TV, which measures the relative monthly trading volume to market capitalisation and ZTD, the number of zero trading days in the previous month are also in the top 10. This shows that market liquidity is an important indicator of explanatory power for individual stocks.

Five of the top ten factors use information on the size by market capitalisation. These consist of BM and IBM which are the book-to-market ratio and its industry-adjusted variable, MCAP and IMCAP which are the market capitalisation and its industry adjustment, as well as RDMCAP, which measures the research and development expenses relative to the market capitalisation. Size is an important factor to determine the explanatory power for individual stock returns. This is also in line with the results from Section 3.2 and Figure 1, which indicate that returns of small firms are easier to predict.

3.6.2 Deep Variable Influence

Figure 7 shows the results of the deep variable importance algorithm, computed over 25 generations, as described in Section 2.8. We report the difference in terms of explanatory power between the baskets of returns sorted by these variables. Sorting by variable combinations leads to gains in terms of R^2_{oos} which are higher than the gains obtained when sorting by individual variables. This information can be used to form portfolios similar to Fama and French, 1993.

We find that, for FFNN(u), the benefits of using grouped factors is smaller than for the other models, which indicates that it is an efficient model regardless of the depth of the RPS structure analysed. This suggests that FFNN generally identify a deeper structure in the dataset, even in a simple feed forward architecture. To a lesser extent, this also holds for ENET(u). The tree-based architectures can benefit from selecting RPS based on this measure for portfolio construction.

Sorting by the financial statement score (FSS), zero trading days (ZTD) and book-to-market ratio (BM) is beneficial to indicate predictability of stock returns. On average, the methods based on this variable combination lead to an increase in the R^2 of about 1.1%. This is followed very closely by the variable combination of ZTD, BM and industry adjusted market capitalisation (IMCAP), as well as the combination of the return on assets (ROA), industry adjusted book-to-market ratio (IBM) and ZTD variables. The weakest survivor after 25 generations still leads to a gap of 0.7% in R^2_{oos}.

Figure 8 shows the occurrences of single-variable and two-variable combinations in the 35 survivors of generation 25. The two-variable combinations BM_ZTD and IBM_ZTD occur 7 times each in the survivors, which is the maximum for any two-variable combination imposed by the limit to ensure diversity. This means that in a deeper structure of variable influence, a mix with book-to-market value or liquidity can yield additional information about stock return predictability. The next three combinations all include volatility of share turnover, which is a measure for market activity.

Illiquidity (ILL), previously identified as the top factor for variable influence when sorting by one variable, only ranks joint 6th. Whilst trading volume (TV) does not appear as an influential variable in our analysis based on 3 factors combinations, it is ranked second when only one factor is used. The top 4 single factors are the two book-to-market RPS and two market liquidity measures, specifically the volatility of share turnover (VST) and the number of zero trading days (ZTD). Out of the variables that capture quarterly firm performance, the earnings-to-price ratio (EP) is the highest-ranking RPS. All this highlights that monthly stock returns are driven mostly by stock market activity and size, and depend less on their business activity.

4 Conclusion

We study the effectiveness of forecast combinations based on machine learning methods in the context of return forecasting. The main advantages of forecast combinations are stability in the prediction and increased accuracy due to the reliance on different architectures. Forecast combinations in a machine learning setting are beneficial, when combining within a model family and across model families. These results are particularly strong for regression-based architectures, where first level forecast combinations stabilise and improve the performance drastically. The ISDS forecast combination proposed here outperforms naive averaging in almost every portfolio statistic.The Sharpe ratio increases by about to 3.06 on the second level of forecast combination, compared to the strongest component FFNN at 2.8.

Additionally, we introduce a new performance measure in a quantile portfolio setup. The main advantage of the new measure is that it adds robustness in a noise-driven prediction setting and is arguably a better indicator of portfolio performance than the R^2_{oos} measure.

We also introduce a new measure of variable influence for return prediction using genetic algorithms on decision tree-based combinations. This provides an interpretation of complex machine learning models in terms of multi-level variable influence. We conclude that market liquidity and firm size contribute to return predictability more than business activity.

We find that the advantage of machine learning portfolios has declined in the last years of the test sample. The market seems to be able to price an increasing amount of information through advancements in information technology. Our results show no benefits to using macroeconomic variables for equity return prediction in a cross-product interaction term setting.

Declaration

During the preparation of this work the authors didn't use any AI tools to improve writing.

References

Bali, T. G., Beckmeyer, H., Moerke, M., & Weigert, F. (2021). Option return predictability with machine learning and big data. *Working Paper*.

Ban, G.-Y., El Karoui, N., & Lim, A. E. (2018). Machine learning and portfolio optimization. *Management Science*, *64*(3), 1136–1154.

Bianchi, D., Büchner, M., & Tamoni, A. (2021). Bond risk premiums with machine learning. *The Review of Financial Studies*, *34*(2), 1046–1089.

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

Bryzgalova, S., Pelger, M., & Zhu, J. (2020). Forest through the trees: Building cross-sections of stock returns. *Working Paper*.

Campisi, G., Muzzioli, S., & De Baets, B. (2023). A comparison of machine learning methods for predicting the direction of the US stock market on the basis of volatility indices. *International Journal of Forecasting*.

Chen, L., Pelger, M., & Zhu, J. (2021). Deep learning in asset pricing. *Working Paper*.

Cochrane, J. H. (2011). Presidential address: Discount rates. *The Journal of Finance*, *66*(4), 1047–1108.

Diebold, F. X., & Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *20*(1), 134–144.

Elliott, G., & Timmermann, A. (2013). *Handbook of economic forecasting; Chapter 4: Forecast combinations*. Elsevier, Amsterdam, Netherlands.

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, *33*(1), 3–56.

Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, *116*(1), 1–22.

Ferson, W. E., & Harvey, C. R. (1999). Conditioning variables and the cross section of stock returns. *The Journal of Finance*, *54*(4), 1325–1360.

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation*, *121*(2), 256–285.

Freyberger, J., Neuhierl, A., & Weber, M. (2020). Dissecting characteristics nonparametrically. *The Review of Financial Studies*, *33*(5), 2326–2377.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, *28*(2), 337–407.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1–22.

Green, J., Hand, J. R., & Zhang, X. F. (2013). The supraview of return predictive signals. *Review of Accounting Studies*, *18*(3), 692–730.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*(5), 2223–2273.

Gu, S., Kelly, B., & Xiu, D. (2021). Autoencoder asset pricing models. *Journal of Econometrics*, *222*(1), 429–450.

Harvey, C. R., Liu, Y., & Zhu, H. (2016). . . . and the cross-section of expected returns. *Review of Financial Studies*, *29*(1), 5–68.

Heaton, J., Polson, N., & Witte, J. H. (2017). Deep learning for finance: Deep portfolios. *Applied Stochastic Models in Business and Industry*, *33*(1), 3–12.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, *48*(1), 65–91.

Kelly, B., & Pruitt, S. (2013). Market expectations in the cross-section of present values. *The Journal of Finance*, *68*(5), 1721–1756.

Kelly, B., Pruitt, S., & Su, Y. (2019). Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, *134*(3), 501–524.

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization. *Working Paper*.

LeBlanc, M., & Tibshirani, R. (1996). Combining estimates in regression and classification. *Journal of the American Statistical Association*, *91*(436), 1641–1650.

Lin, W., & Taamouti, A. (2023). Portfolio selection under non-gaussianity and systemic risk: A machine learning based forecasting approach. *International Journal of Forecasting*.

Masters, T. (1993). *Practical neural network recipes in C++*. Academic, San Diego, California.

Rapach, D., Strauss, J., & Zhou, G. (2010). Out-of-sample equity premium prediction: Combination forecasts and links to the real economy. *The Review of Financial Studies*, *23*(2), 821–862.

Rapach, D., & Zhou, G. (2013). Forecasting stock returns. *Handbook of Economic Forecasting*.

Rapach, D., & Zhou, G. (2020). Time-series and cross-sectional stock return forecasting: New machine learning methods. *Machine Learning for Asset Management: New Developments and Financial Applications*, 1–33.

Rather, A. M., Agarwal, A., & Sastry, V. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications*, *42*(6), 3234–3241.

Rosenberg, B. (1974). Extra-market components of covariance in security returns. *Journal of Financial and Quantitative Analysis*, *9*(2), 263–274.

Rossi, A. G. (2018). Predicting stock market returns with machine learning. *Working paper*.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*(2), 197–227.

Welch, I., & Goyal, A. (2007). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, *21*(4), 1455–1508.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301–320.

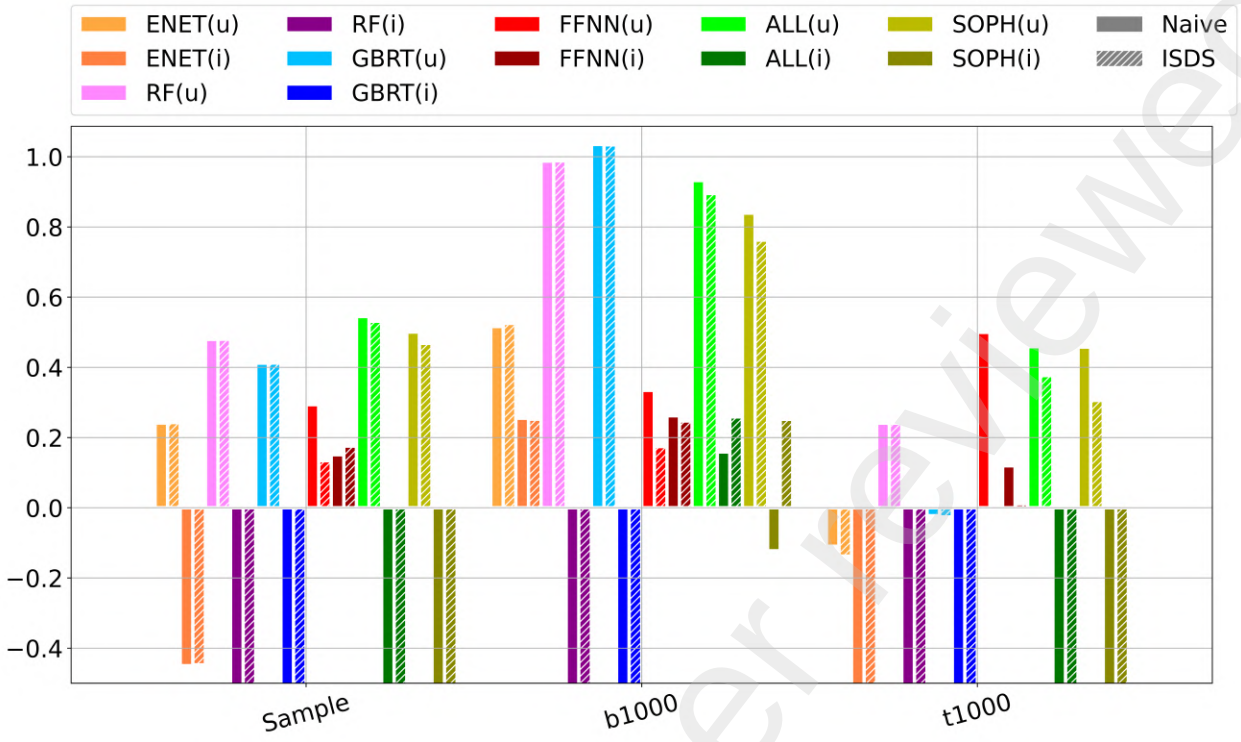


Figure 1: R^2_{oos} results as percentages. We show bars for the entire sample as well as the bottom and top 1000 stocks (denoted by b1000 and t1000) by market capitalisation, per month. (u) denotes the models estimated using the uninteracted setup, and (i) denotes the models estimated using the interacted setup. We cap off highly negative values.

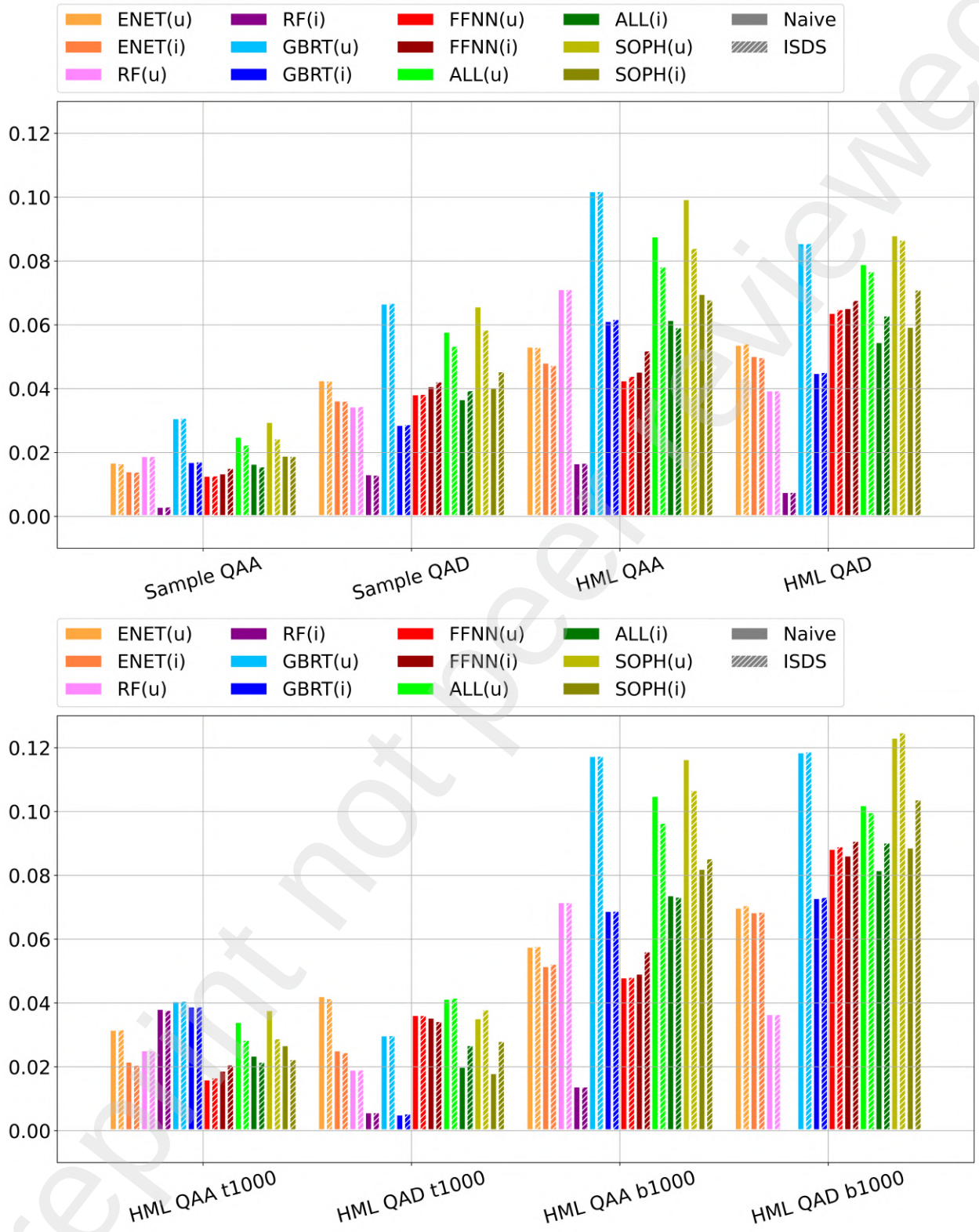


Figure 2: QAA/QAD on decile predictions. The figures show out-of-sample QAA and QAD for the the test samples. We show values for the entire sample as well as for the lowest and highest sorted 10% of stocks out-of-sample in the top figure, and HML decile results for the bottom and top 1000 (denoted as b1000 and t1000) stocks by market capitalisation per month in the bottom figure. (u) denotes models estimated using the uninteracted setup, and (i) denotes the models estimated using the interacted setup.

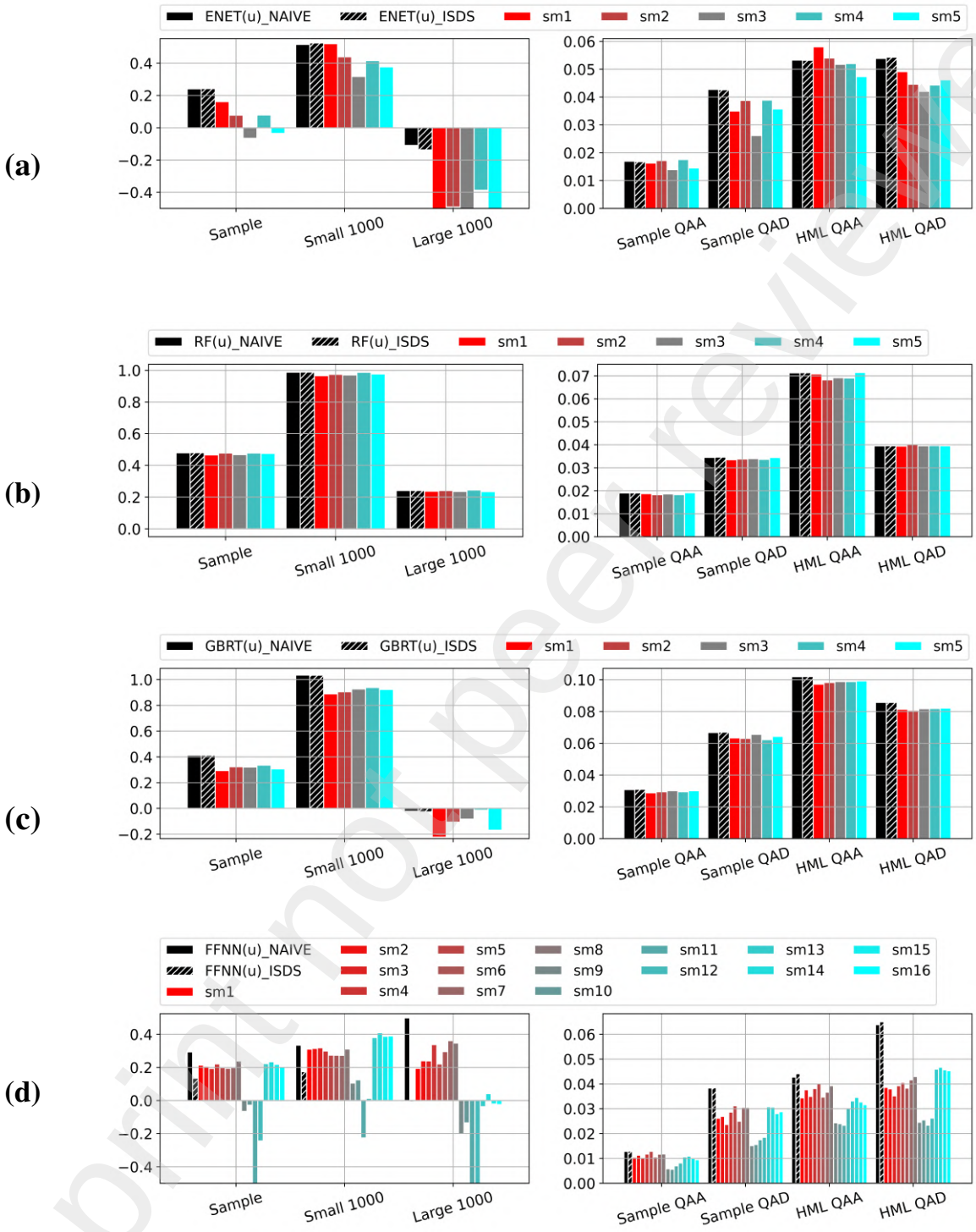


Figure 3: Predictive performance of forecast combinations within model families (first-level forecast combination) estimated on the uninteracted setup. The left side of each panel shows R^2_{oos}, the right side shows QAA and QAD. The figures show the results of **(a)**: elastic net regression, **(b)**: random forest, **(c)**: gradient boosted regression trees and **(d)**: feed forward neural networks. The performance measurement of the forecast combination of models within a given model family is shown as the black line. Submodels, abbreviated as *sm* followed by a number, are shown in shades from red to blue.

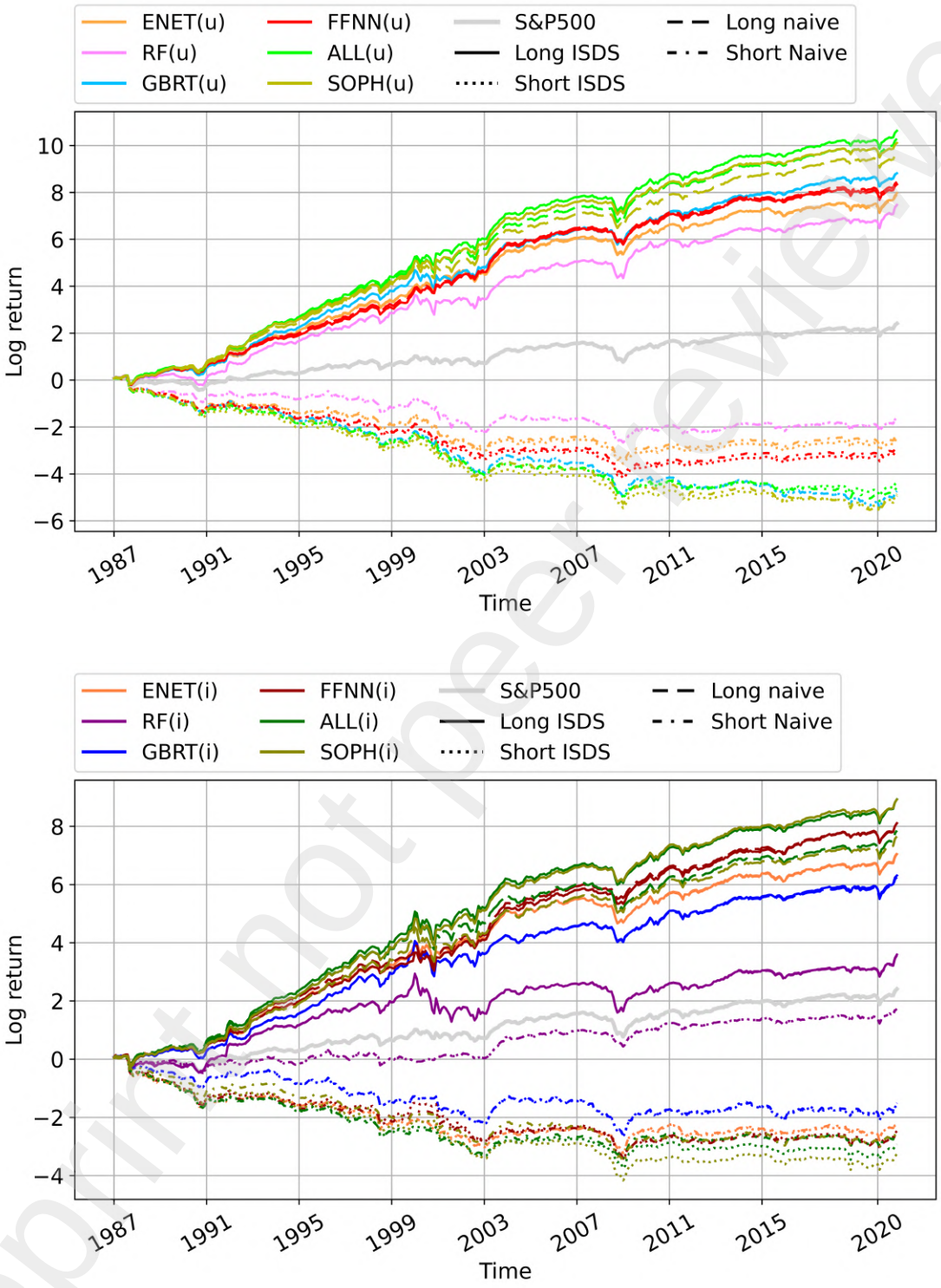


Figure 4: Machine learning decile long- and short portfolios between January 1987 and December 2020. The top figure shows the results on the uninteracted setup, and the lower figure shows the results for the interacted setup. The deciles are based on monthly total excess return predictions. In addition to long and short positions implied by the respective deciles, we show the S&P500 historical total return index.

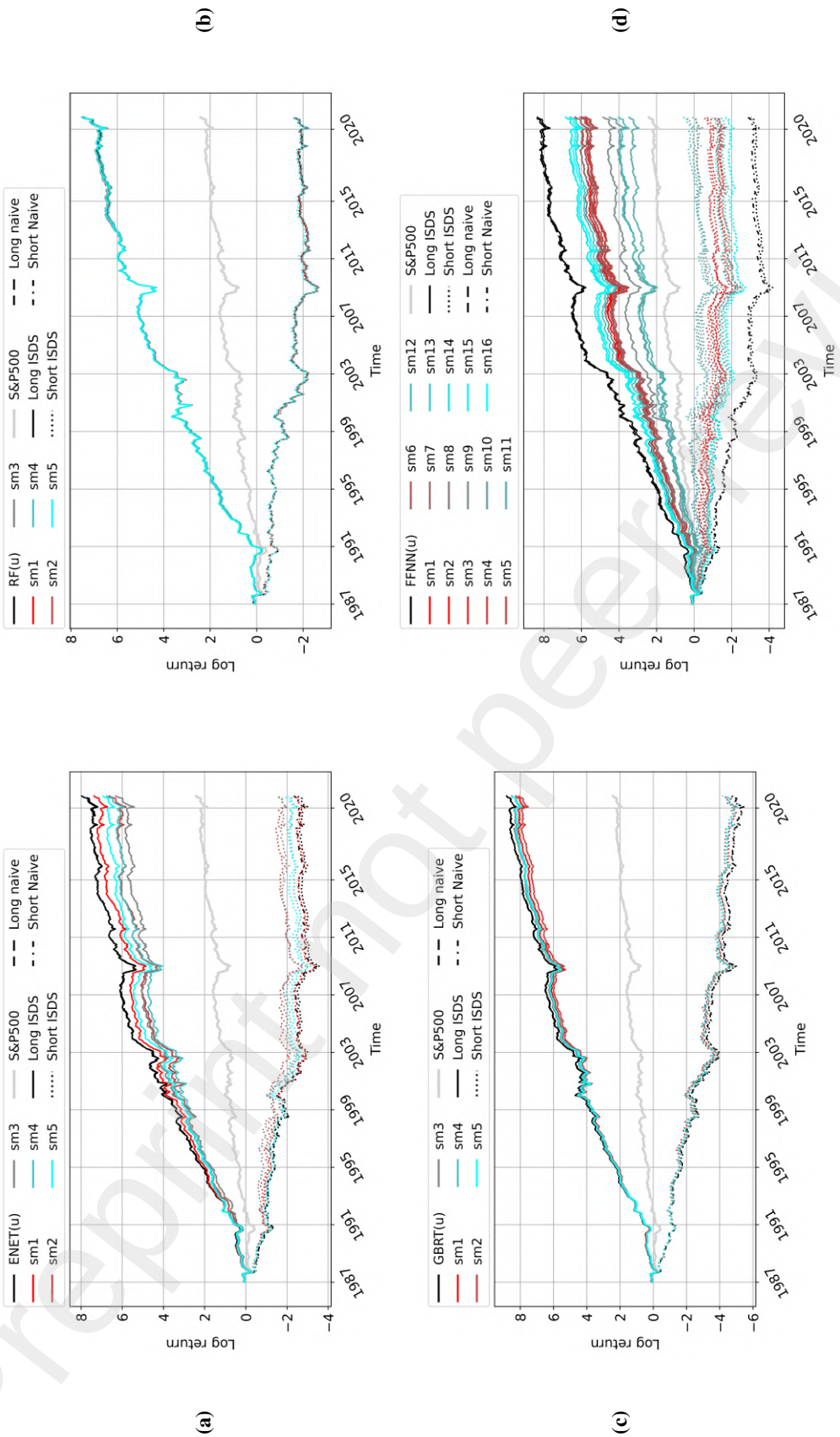


Figure 5: Submodel long- and short portfolio performance between January 1987 and December 2020 estimated on the uninteracted setup. The panels show **(a)**: elastic net regression, **(b)**: random forest, **(c)**: gradient boosted regression trees and **(d)**: feed forward neural networks. The forecast combination for the respective performance measurement inside a model family is shown as the black bar. Submodels, abbreviated as *sm* followed by a number, are shown in shades from red to blue.

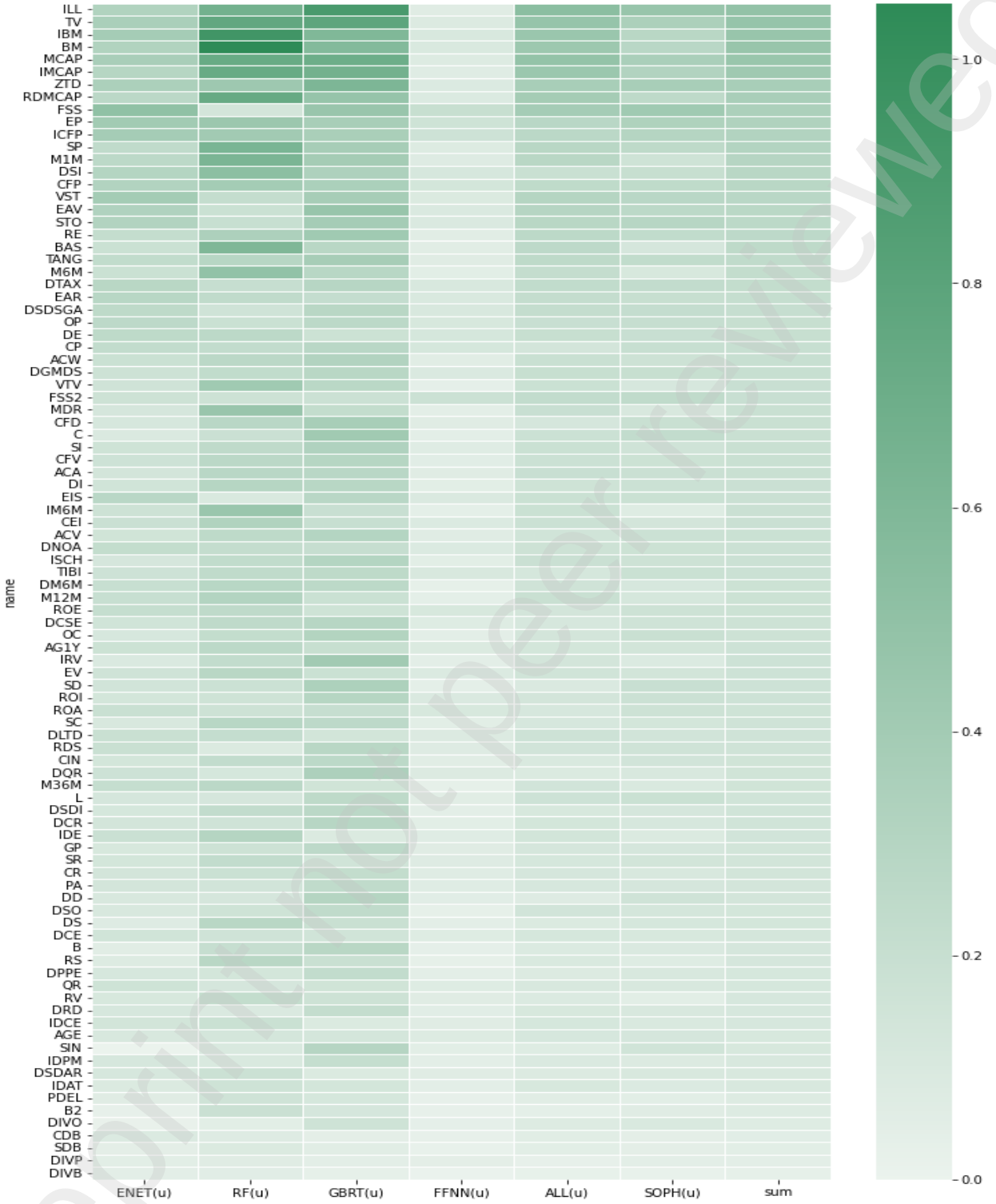


Figure 6: Out-of-sample feature importance for the models using the 94 characteristics for the entire sample of the uninteracted setup. The models are presented along the X-axis. The term "sum" summarises all methods to an average overall influence of the features denoted on the Y-axis. They are ranked from "most influential" in the top to "least influential" in the bottom based on the column "sum". Also see Supplementary Appendix A for explanation of every RPS.

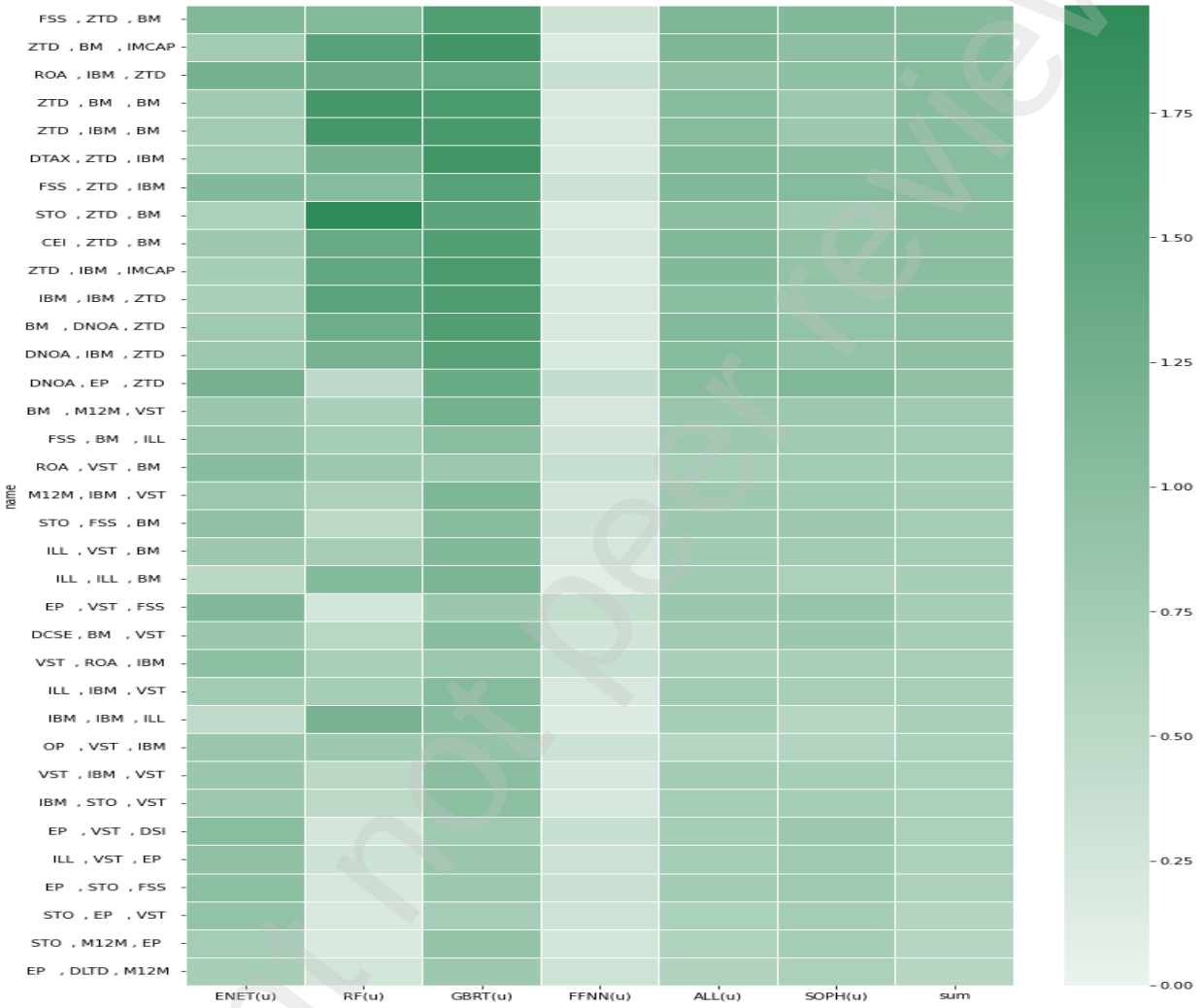
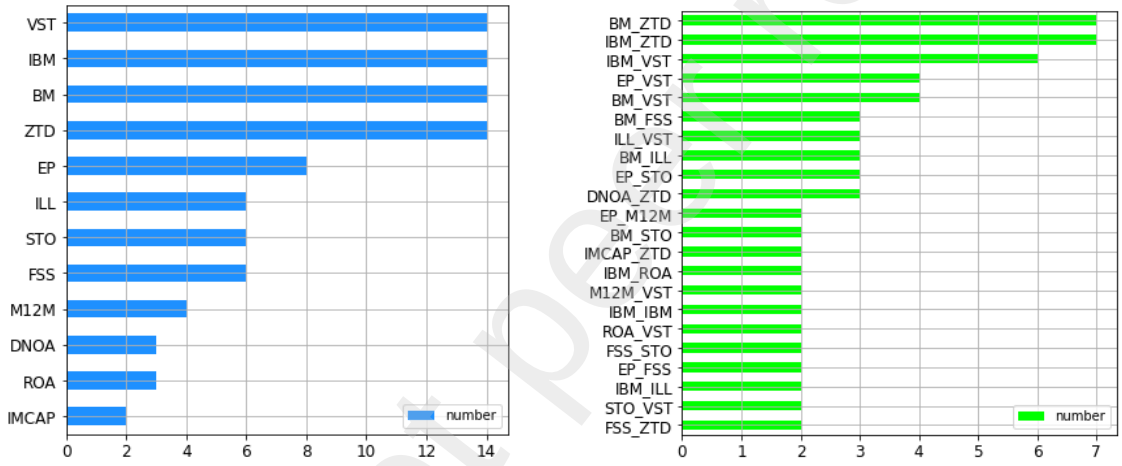


Figure 7: Deep variable influence using three variable combinations. The 35 rows refer to three variable combination decision trees after 25 from our genetic algorithm after generations. The columns refer to models, where "sum" is the average across the models. The intensity of the colour refers to the maximum difference in R^2_{oos} between 2 buckets in the respective tree and model. Also see Supplementary Appendix A for detailed explanation of every RPS.



One variable occurrences **Two variable occurrences**

Figure 8: One- and two-variable occurrences after 25 generations on the left and right side, respectively. Note: One variable occurrence is capped to 14 per survival phase, while two variable occurrences are capped at 7. Variables and combinations occurring once are left out for this analysis.

	ENET(i)	RF(u)	RF(i)	GBRT(u)	GBRT(i)	FFNN(u)
ENET(u)	21.54***	-8.31***	35.52***	-11.95***	35.14***	4.38***
ENET(i)		-16.72***	31.05***	-23.85***	32.75***	-16.65***
RF(u)			41.16***	-0.98	41.81***	9.42***
RF(i)				-42.94***	16.3***	-35.04***
GBRT(u)					40.61***	13.17***
GBRT(i)						-34.65***
		FFNN(i)	ALL(u)	ALL(i)	SOPH(u)	SOPH(i)
ENET(u)		-1.28	-29.21***	14.92***	-25.78***	16.45***
ENET(i)		-21.2***	-40.18***	3.8***	-39.73***	8.2***
RF(u)		7.93***	-2.7**	23.99***	-0.58	26.02***
RF(i)		-36.11***	-43.92***	-36.09***	-42.37***	-25.52***
GBRT(u)		11.38***	-1.99*	28.44***	0.75	27.81***
GBRT(i)		-35.29***	-40.02***	-37.36***	-39.02***	-40.46***
FFNN(u)		-17.9***	-30.21***	13.04***	-31.05***	14.9***
FFNN(i)			-26.91***	15.44***	-26.9***	16.78***
ALL(u)				34.48***	6.86***	30.21***
ALL(i)					-30.06***	10.38***
SOPH(u)						27.28***

Table 1: Diebold-Mariano test statistics between the forecasts of the row model and the column model using ISDS forecast combination. Positive values indicate outperformance of the row model to the column model, and vice versa for negative values. *, **, and *** indicate significance at the 5%, 1% and 0.1% levels, respectively.

	ENET(u) ISDS				RF(u) ISDS				GBRT(u) ISDS			
	Pred	Mean	Std	SR	Pred	Mean	Std	SR	Pred	Mean	Std	SR
Low	-1.11	-0.43	5.95	-0.25	-0.22	-0.16	6.54	-0.09	-1.44	-0.87	7.41	-0.41
2nd	-0.4	0.15	5.16	0.1	0.21	0.34	4.97	0.24	-0.19	0.22	5.69	0.13
3rd	-0.04	0.37	4.78	0.27	0.36	0.38	4.75	0.28	0.23	0.42	4.89	0.3
4th	0.23	0.43	4.62	0.33	0.48	0.42	4.61	0.32	0.5	0.54	4.38	0.42
5th	0.47	0.6	4.62	0.45	0.61	0.65	4.54	0.49	0.72	0.68	4.4	0.54
6th	0.71	0.68	4.9	0.48	0.72	0.71	4.68	0.53	0.92	0.71	4.47	0.55
7th	0.96	0.84	5.11	0.57	0.83	0.7	4.79	0.51	1.13	0.87	4.75	0.63
8th	1.24	1.04	5.55	0.65	0.96	0.92	5.1	0.62	1.37	1.02	5.07	0.7
9th	1.6	1.42	6.04	0.82	1.14	1.15	6.09	0.66	1.67	1.26	5.74	0.76
High	2.28	2.22	7.05	1.09	2.1	2.21	8.84	0.87	2.58	2.49	7.95	1.08
H – L	3.39	2.65	4.28	2.14	2.32	2.37	5.54	1.48	4.02	3.36	5.02	2.32
	ENET(u) naive				RF(u) naive				GBRT(u) naive			
	Pred	Mean	Std	SR	Pred	Mean	Std	SR	Pred	Mean	Std	SR
Low	-1.09	-0.39	5.97	-0.23	-0.22	-0.17	6.54	-0.09	-1.44	-0.87	7.41	-0.41
2nd	-0.39	0.12	5.11	0.08	0.21	0.34	4.97	0.24	-0.19	0.22	5.69	0.13
3rd	-0.04	0.35	4.76	0.25	0.36	0.38	4.75	0.28	0.23	0.42	4.91	0.3
4th	0.23	0.43	4.65	0.32	0.48	0.42	4.6	0.31	0.5	0.54	4.38	0.43
5th	0.47	0.59	4.6	0.45	0.61	0.65	4.54	0.5	0.72	0.68	4.4	0.53
6th	0.71	0.68	4.9	0.48	0.72	0.71	4.68	0.52	0.92	0.72	4.47	0.56
7th	0.95	0.85	5.13	0.58	0.83	0.71	4.8	0.51	1.13	0.86	4.75	0.63
8th	1.23	1.06	5.56	0.66	0.96	0.92	5.09	0.62	1.37	1.01	5.07	0.69
9th	1.59	1.43	6.04	0.82	1.14	1.15	6.09	0.65	1.67	1.26	5.73	0.76
High	2.26	2.2	7.07	1.08	2.1	2.21	8.84	0.86	2.58	2.49	7.97	1.08
H – L	3.35	2.59	4.36	2.06	2.32	2.37	5.54	1.48	4.01	3.36	5.05	2.31
	FFNN(u) ISDS				ALL(u) ISDS				SOPH(u) ISDS			
	Pred	Mean	Std	SR	Pred	Mean	Std	SR	Pred	Mean	Std	SR
Low	-0.53	-0.58	5.52	-0.36	-0.7	-0.86	6.28	-0.47	-0.98	-0.96	6.58	-0.51
2nd	-0.32	-0.04	4.96	-0.03	-0.15	0.09	5.06	0.06	-0.34	0.05	5.12	0.03
3rd	-0.21	0.32	4.73	0.24	0.09	0.3	4.53	0.23	-0.07	0.31	4.51	0.24
4th	-0.12	0.39	4.67	0.29	0.26	0.4	4.41	0.32	0.12	0.47	4.41	0.37
5th	-0.05	0.54	4.87	0.38	0.42	0.52	4.58	0.39	0.29	0.55	4.57	0.42
6th	0.03	0.75	5.12	0.51	0.58	0.62	4.83	0.44	0.46	0.68	4.86	0.48
7th	0.12	0.94	5.3	0.62	0.75	0.81	5.09	0.55	0.64	0.86	5.1	0.58
8th	0.21	1.18	5.64	0.73	0.94	1.06	5.41	0.68	0.85	1.1	5.4	0.71
9th	0.33	1.53	5.95	0.89	1.19	1.45	5.92	0.85	1.11	1.49	5.89	0.88
High	0.56	2.29	6.6	1.2	1.77	2.94	7.93	1.28	1.67	2.78	7.42	1.3
H – L	1.09	2.86	3.54	2.8	2.47	3.79	4.82	2.72	2.65	3.74	4.22	3.06
	FFNN(u) naive				ALL(u) naive				SOPH(u) naive			
	Pred	Mean	Std	SR	Pred	Mean	Std	SR	Pred	Mean	Std	SR
Low	0.21	-0.56	5.34	-0.36	-0.38	-0.9	6.5	-0.48	-0.49	-0.94	7.14	-0.46
2nd	0.42	-0.03	4.82	-0.02	0.13	0.03	5.18	0.02	0.17	0.12	5.38	0.08
3rd	0.52	0.27	4.8	0.19	0.34	0.34	4.55	0.26	0.41	0.38	4.59	0.29
4th	0.6	0.42	4.66	0.31	0.49	0.46	4.38	0.36	0.56	0.49	4.42	0.38
5th	0.67	0.53	4.82	0.38	0.62	0.53	4.52	0.41	0.69	0.61	4.44	0.48
6th	0.75	0.74	5.08	0.5	0.75	0.68	4.7	0.5	0.82	0.71	4.59	0.53
7th	0.82	0.93	5.42	0.6	0.89	0.81	4.95	0.57	0.95	0.9	4.92	0.63
8th	0.91	1.16	5.68	0.71	1.05	1.04	5.33	0.67	1.11	1.0	5.23	0.66
9th	1.03	1.55	6.0	0.9	1.25	1.46	5.94	0.85	1.3	1.38	5.86	0.81
High	1.25	2.32	6.69	1.2	1.81	2.88	8.27	1.2	1.83	2.69	7.92	1.18
H – L	1.03	2.87	3.64	2.74	2.18	3.78	5.18	2.52	2.32	3.63	4.83	2.61

Table 2: Decile portfolio statistics. The deciles are based on monthly total excess return predictions. "Mean" is the mean of monthly portfolio returns of stocks in the decile. "Std" is the standard deviation of the returns in the decile. "SR" is the annualised Sharpe ratio of the decile portfolio. "H-L" stands for a portfolio of top decile minus bottom decile.

Panel (a)	Mean	Std	SR(m)	DDmax	Min	Max	Turnover
ENET(u) ISDS	2.65	4.28	0.62	12.24	-7.94	45.39	88.12
ENET(u) naive	2.59	4.36	0.6	13.86	-8.83	45.07	88.52
RF(u) ISDS	2.37	5.54	0.43	27.94	-13.44	62.5	132.41
RF(u) naive	2.37	5.54	0.43	27.95	-13.56	62.53	132.39
GBRT(u) ISDS	3.36	5.02	0.67	39.47	-14.92	52.34	118.67
GBRT(u) naive	3.36	5.05	0.67	39.41	-14.93	52.91	118.52
FFNN(u) ISDS	2.86	3.54	0.81	17.9	-9.27	19.31	116.78
FFNN(u) naive	2.87	3.64	0.79	17.07	-10.1	27.17	113.74
ALL(u) ISDS	3.79	4.82	0.79	13.68	-8.28	56.24	111.24
ALL(u) naive	3.78	5.18	0.73	24.96	-13.82	63.53	111.6
SOPH(u) ISDS	3.74	4.22	0.88	30.12	-9.83	40.98	121.34
SOPH(u) naive	3.63	4.83	0.75	36.04	-13.76	52.46	120.79
market	0.73	5.2	0.14	56.67	-27.19	20.64	0.0
Panel (b)	β	α	t(α)	FF3	t(α)	FF5+mom	t(α)
ENET(u) ISDS	21.99	2.49	12.04	2.53	12.72	2.29	11.6
ENET(u) naive	22.29	2.43	11.55	2.48	12.2	2.23	11.1
RF(u) ISDS	33.89	2.12	8.08	2.22	9.21	2.06	8.48
RF(u) naive	33.86	2.12	8.07	2.22	9.21	2.06	8.48
GBRT(u) ISDS	5.41	3.32	13.22	3.36	13.43	3.26	12.96
GBRT(u) naive	5.61	3.32	13.16	3.36	13.37	3.26	12.9
FFNN(u) ISDS	22.11	2.7	16.1	2.73	16.59	2.46	15.16
FFNN(u) naive	27.0	2.68	15.9	2.71	16.62	2.48	15.31
ALL(u) ISDS	27.72	3.59	15.57	3.67	16.89	3.42	16.0
ALL(u) naive	28.33	3.57	14.33	3.65	15.49	3.48	14.89
SOPH(u) ISDS	13.3	3.64	17.42	3.68	17.96	3.46	17.06
SOPH(u) naive	9.94	3.56	14.79	3.6	15.12	3.47	14.52

Table 3: Monthly HML portfolio performance measures. "Mean", "Std" and "SR(m)" describe the mean, standard deviation and monthly Sharpe ratio of returns. "DDmax" is the maximum drawdown, the maximum loss from the previous high water mark. "Min" and "Max" describe the minimum and maximum monthly return. "Turnover" describes the monthly readjustment factor. "β" is systematic risk to the market return, "α" is Jensen's alpha to the market with its corresponding t-value. FF3 and FF5+mom are the αs of the portfolios to the respective factor models based on Fama and French (1993) and Fama and French (2015) plus momentum.

Predicting Equity Returns with Forecast Combinations of Deep Learning and Ensemble Methods

Eike-Christian Brinkop* **Emese Lazar†** **Marcel Prokopczuk‡**

December 11, 2023

Supplementary Appendix

This Supplementary Appendix is organised as follows. Section A contains the lists for all of our 94 RPS and 8 macro terms, as well as their scientific source. Section B contains additional information about our methodology. Section C contains additional results from our experiments.

*ICMA Centre, Henley Business School, University of Reading
e.brinkop@pgr.reading.ac.uk(corresponding author)
†ICMA Centre, Henley Business School, University of Reading
e.lazar@icmacentre.ac.uk
‡Leibniz University Hanover, Germany
prokopczuk@fcm.uni-hannover.de

A Variable list

Company Specific Variables

Abbreviation	Long Name	Frequency	Literature Source
ACA	Absolute accruals	quarterly	Bandyopadhyay et al. (2010)
ACV	Accrual volatility	quarterly	Bandyopadhyay et al. (2010)
ACW	Working capital accruals	quarterly	Sloan (1996)
AGE	Age of firm	quarterly	Jiang et al. (2005)
AG1Y	Asset growth	quarterly	Cooper et al. (2008)
B	Beta	monthly	Fama and MacBeth (1973)
B2	Beta squared	monthly	Fama and MacBeth (1973)
BAS	Bid–ask spread	monthly	Amihud and Mendelson (1989)
BM	Book-to-market	monthly	Rosenberg et al. (1985)
C	Cash holdings	quarterly	Palazzo (2012)
CDB	Convertible debt indicator	quarterly	Valta (2016)
CEI	Capital expenditures and inventory	quarterly	Thomas and Zhang (2002)
CFD	Cash flow to debt	quarterly	Ou and Penman (1989)
CFP	Cash flow to price ratio	quarterly	Desai et al. (2004)
CFV	Cash flow volatility	quarterly	Huang (2009)
CIN	Corporate investment	quarterly	Titman et al. (2004)
CP	Cash productivity	quarterly	Chandrashekar and Rao (2009)
CR	Current ratio	quarterly	Ou and Penman (1989)
DCE	Growth in capital expenditures	annual	Anderson and Garcia-Feijóo (2006)
DCR	% change in current ratio	quarterly	Ou and Penman (1989)
DCSE	Growth in common shareholder equity	quarterly	Richardson et al. (2005)
DD	% change in depreciation	quarterly	Holthausen and Larcker (1992)
DE	Employee growth rate	annual	Belo et al. (2014)
DGMDS	% change in gross margin – % change in sales	quarterly	Abarbanell and Bushee (1998)
DI	Change in inventory	quarterly	Thomas and Zhang (2002)
DIVB	Dividend initiation	quarterly	Michaely et al. (1995)
DIVO	Dividend omission	quarterly	Michaely et al. (1995)
DIVP	Dividend to price	quarterly	Litzenberger and Ramaswamy (1979)
DLTD	Growth in long-term debt	quarterly	Richardson et al. (2005)
DM6M	Change in 6-month momentum	monthly	Gettleman and Marks (2006)
DNOA	Growth in long-term net operating assets	quarterly	Fairfield et al. (2003)
DPPE	Depreciation/ PP&E	quarterly	Holthausen and Larcker (1992)
DQR	% change in quick ratio	quarterly	Ou and Penman (1989)
DRD	R&D increase	quarterly	Eberhart et al. (2004)
DS	Sales growth	quarterly	Lakonishok et al. (1994)
DSDAR	% change in sales – % change in A/R	annual	Abarbanell and Bushee (1998)

Abbreviation	Long Name	Frequency	Litreature Source
DSDI	% change in sales – % change in inventory	quarterly	Abarbanell and Bushee (1998)
DSDSGA	% change in sales – % change in SG&A	quarterly	Abarbanell and Bushee (1998)
DSI	% change sales-to-inventory	quarterly	Ou and Penman (1989)
DSO	Change in shares outstanding	monthly	Pontiff and Woodgate (2008)
DTAX	Change in tax expense	quarterly	Thomas and Zhang (2011)
EAR	Earnings announcement return	quarterly	Brandt et al. (2008)
EAV	Abnormal earnings announcement volume	quarterly	Lerman et al. (2007)
EIS	Number of earnings increases	quarterly	Barth et al. (1999)
EP	Earnings-to-price	monthly	Basu (1977)
EV	Earnings volatility	quarterly	Francis et al. (2004)
FSS	Financial statements score	quarterly	Piotroski et al. (2000)
FSS2	Financial statement score	quarterly	Mohanram (2005)
GP	Gross profitability	quarterly	Novy-Marx (2013)
IBM	Industry-adjusted book to market	monthly	Asness et al. (2000)
ICFP	Industry-adjusted cash flow to price ratio	quarterly	Asness et al. (2000)
IDAT	Industry-adjusted change in asset turnover	monthly	Soliman (2008)
IDCE	Industry adjusted % change in capex	quarterly	Abarbanell and Bushee (1998)
IDE	Industry-adjusted change in employees	annual	Asness et al. (2000)
IDPM	Industry-adjusted change in profit margin	monthly	Soliman (2008)
ILL	Illiquidity	monthly	Amihud (2002)
IM6M	Industry momentum	monthly	Moskowitz and Grinblatt (1999)
IMCAP	Industry-adjusted size	monthly	Asness et al. (2000)
IRV	Idiosyncratic return volatility	monthly	Ali et al. (2003)
ISCH	Industry sales concentration	monthly	Hou and Robinson (2006)
L	Leverage	quarterly	Bhandari (1988)
M12M	12-month momentum	monthly	Jegadeesh (1990)
M1M	1-month momentum	monthly	Jegadeesh and Titman (1993)
M36M	36-month momentum	monthly	Jegadeesh and Titman (1993)
M6M	6-month momentum	monthly	Jegadeesh and Titman (1993)
MCAP	Size	monthly	Banz (1981)
MDR	Maximum daily return	monthly	Bali et al. (2011)
OC	Organisational capital	yearly	Eisfeldt and Papanikolaou (2013)
OP	Operating profitability	quarterly	Fama and French (2015)
PA	% accruals	quarterly	Hafzalla et al. (2011)
PDEL	Price delay	monthly	Hou and Moskowitz (2005)
QR	Quick ratio	quarterly	Ou and Penman (1989)
RDMCAP	R&D to market capitalisation	quarterly	Guo et al. (2006)
RDS	R&D to sales	quarterly	Guo et al. (2006)
RE	Real estate holdings	quarterly	Tuzel (2010)
ret	Succeeding return	monthly	–
ROA	Return on assets	quarterly	Balakrishnan et al. (2010)
ROE	Return on equity	quarterly	Hou et al. (2015)
ROI	Return on invested capital	quarterly	Brown and Rowe (2007)
RS	Revenue surprise	quarterly	Kama (2009)
RV	Return volatility	monthly	Ang et al. (2006)
SC	Sales to cash	quarterly	Ou and Penman (1989)
SD	Secured debt	quarterly	Valta (2016)
SDB	Secured debt indicator	quarterly	Valta (2016)

Abbreviation	Long Name	Frequency	Litreature Source
SI	Sales to inventory	quarterly	Ou and Penman (1989)
SINB	Sin stocks	-	Hong and Kacperczyk (2009)
SP	Sales to price	quarterly	Barbee Jr et al. (1996)
SR	Sales to receivables	quarterly	Ou and Penman (1989)
STO	Share turnover	monthly	Datar et al. (1998)
TANG	Debt capacity/firm tangibility	quarterly	Almeida and Campello (2007)
TIBI	Tax income to book income	quarterly	Lev and Nissim (2004)
TV	Dollar trading volume	monthly	Chordia et al. (2001)
VST	Volatility of liquidity (share turnover)	monthly	Chordia et al. (2001)
VTV	Volatility of liquidity (dollar trading volume)	monthly	Chordia et al. (2001)
ZTD	Zero trading days	monthly	Liu (2006)

Table A1: Return Predictive Signals (RPS) used for the return prediction. "Frequency" describes the update frequency of the variable.

Macro Variables

abbreviation	name	source
dp	dividend yield	Welch and Goyal (2007)
ep	earnings-to-price	Welch and Goyal (2007)
bm	book-to-market	Welch and Goyal (2007)
ntis	net equity expansion	Welch and Goyal (2007)
tbl	treasury bill rate	Welch and Goyal (2007)
tms	term spread	Welch and Goyal (2007)
dfy	default spread	Welch and Goyal (2007)
svar	stock variance	Welch and Goyal (2007)

Table A2: Macroeconomic variables used for the return prediction with interaction terms (i).

B Additional Methodology

In this section, we describe details of our machine learning setup.

B.1 Hyperparameters, Sample Splitting, and Validation

The large number of hyperparameters for each model only allows for a handful of them to be optimised during computation, since some of them need reestimation of the model. Most of the setup has to be preset upfront to keep computational demand at a reasonable level. The hyperparameters and predictor coefficients are not estimated simultaneously. To improve on external validity, the models need additional validation on data disjoint to the data used for the initial estimation of the coefficients.

Next, we describe validation in the context of these models. For this, we first provide the details used for methods of sample splitting, which are a driving factor of regularisation and are a hyperparameter preset before estimation.[1] In our estimation setup, we use rolling windows of 25 years for the estimation. The resulting models predict 12 one-month returns until the model is reestimated after one year. This way, the model doesn't rely on too old data and is updated with new data as it becomes available. In line with Welch and Goyal (2007) and Green et al. (2013), we find that the parameters of the RPS are unstable over time. This accounts for changes in corporate data availability, changes in regulation and for advances in computational power and algorithm development as well.[2]

[1] There are several settings in optimisation of the models which are not optimised through validation. For example, the length of estimation sample is optimised by manual trial and error for each algorithm, trying to find an optimal gross setting with respect to the noise ratio in the data. Only a limited amount of these settings are typically tested. To maintain external validity and prevent data mining, these experiments are run on a small subset of data of 5 years.

[2] An example for this is the Security and Exchange commission (SEC)'s change in accounting regulations to also include mandatory cash flow statements in 1993. Many scientific advances in algorithms, such as Breiman (2001), used here are from the past 30 years, while such methods became computationally feasible on a large scale only in the past few years.

Figure C1 shows three different validation schemes. In validation, only the hyperparameters are varied to minimise the target function. The optimal hyperparameters from the validation are then used to reestimate the model in the estimation sample. The process of estimation and validation iteratively searches for the optimal model with respect to weights and hyperparameters. Instead of limiting the validation to the most recent data, groups of data points are randomly selected for validation with the same relative ratio of validation data points as in Gu et al. (2020) without losing the most recent data points for training. In our approach, we group data points by the month of observation for bootstrapping entire months of data instead of completely random observations. We do so, because monthly stock returns follow the market return. A complete random selection would partially violate the idea of a disjoint validation sample and can weaken external validity and test performance.

Additionally, validation simulates an out-of-sample test of the estimation in order to run the model on unknown data to improve its out-of-sample performance.[3] It also allows for the regularisation methods mentioned earlier to further improve the out-of-sample performance and the model's ability to deal with large datasets, outliers, nonlinearities or insignificant predictors. The validation subsample is not used for the final performance evaluation. Using validation samples has two disadvantages. First, a fraction of data can not be used for training or testing, because it is used for validation. Second, the estimation results of the model can depend on the particular choice of estimation and validation subsample and can be affected by the existence of structural breaks.

Cross-validation can solve these problems. [4] It uses multiple versions of the dataset, in which every data point in the estimation/validation sample can be used for estimation or validation. We

[3] Note that this is not truly out-of-sample, because the hyperparameters are optimised using the results of the previous validation process on the same validation data.

[4] Cross-validation is based on the findings of Kohavi (1995).

combine random selection and cross-validation into a bootstrapping cross-validation setting.[5] The rolling window in cross-validation is split into two subsamples, which are the estimation and the test sample. The estimation sample is also used for validation in cross-validation. To guarantee disjoint estimation and validation, the validation and estimation data are bootstrapped without returning from the estimation/validation data at the beginning of each estimation step. This repeated random choice also provides a changing pair of validation and estimation samples, which solves the second problem of a static validation sample. The disadvantage is that bootstrapping the validation data causes slight instability in the optimum and leads to longer optimisation, because the model tries to converge to different validation samples during the cross-validation.

The variety of hyperparameters is large, so a number of parameters, such as the length of the estimation window or the validation fraction, are selected beforehand. Similarly, the optimisation algorithms, the loss function and the values of several hyperparameters are selected during this process. Validating the algorithms, especially when optimising neural networks, is challenging. Every algorithm presented is a result of multiple tries with multiple constellations. In our empirical application, cross-validation is based on rolling windows - this is to account for intertemporal shifts in the models, their performance and to limit the coefficients' statistical impact.

For pre-processing, variables in the dataset are ranked monthly and then normalised between –1 and 1 from low to high ensuring robustness to outliers in the inputs. We have experimented with different settings of pre-processing, such as batch normalisation and normalisation and have found this approach based on rankings to be more robust and efficient than the alternative methods.

[5]We use a five-fold cross-validation, which means that 20% of training data is randomly held back by the algorithms for validation. Also see Rao et al. (2008) and Hastie et al. (2009) for detailed discussions regarding validation methods.

B.2 Genetic Algorithm Details

The algorithm creates a list of three RPS combinations from our pool of variables used for sorting which is considered the population. We use a population size of $p = 100$ sets of variables. Each of these sets contains 3 RPS, which we use as a trade-off between computational cost and depth of analysis.[6] Then, each month, the algorithm builds decision trees that split the dataset into 50–50 buckets for each RPS successively.[7] This results in $2^3 = 8$ total buckets for each 3-RPS set. For each of these buckets, we compute the Feature Importance measure as discussed above.

We use $n_s = 35$ as survival rate for the number of "children" of the population that "survive" and are transferred to the next evolution step or generation. This step takes care of the diversity of the genetic pool in the population. We control for the diversity by limiting the repetition of combinations. A three-RPS combination x, y, z of any order can only be in the survivor group only once. Then, in our 3-RPS sets, we extract all 2-RPS subsets. For 2-RPS subsets we allow for 7 of the same combination x, y inside the 35 survivors. 1-RPS subsets of the 3-RPS sets can appear up to 14 times.[8]

To fill up the population in each generation back to 100, we use mutations and draft new sets. We generate $n_c = 1$ additional items per survivor by mutation and draft $n_p - (n_s * (n_c + 1)) = 30$ additional items as in the initialisation. The latter imitates migration in real world populations and further increases diversity, which can accelerate convergence. For the mutations, every two elements of the survivors combine their sets randomly to form a new set. Additionally, the first $n_s - (n_s \% 2)$ items have a child with a random mutation in a random spot.[9] This process is repeated for $n_g = 35$ number of generations.[10]

[6] The potential number of combinations is $93^3 = 804.357$, although we acknowledge that the sorting is (nearly) equivalent for $[x, y, z]$ and $[y, x, z]$,where x, y and z, are RPS names. This narrows it down to about 140.000 possible sets.

[7] We leave out the RPS SIN (dummy for sin stocks) for this approach because it is highly skewed across the dataset. Only about 14,000 of its 2,700,000 out-of-sample observations in the test windows are positive.

[8] These numbers are relative to the number of survivors in each generation. If this step is skipped, one or two 2-RPS combinations dominate the survivors. Since we are interested in detecting more than one cluster of influential variables, the genetic diversity is controlled.

[9] Here, % is the notation for modulo or integer division.

[10] We note that in this setting, convergence is achieved.

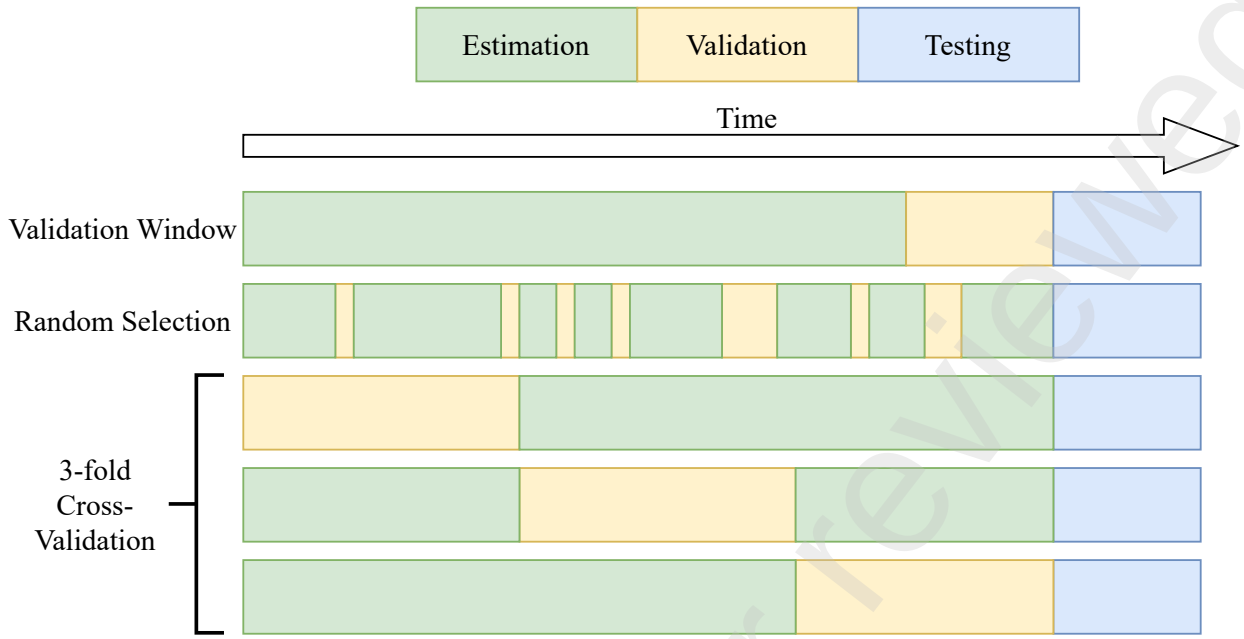


Figure C1: Validation scheme. Gu et al. (2020) use time ordered validation between estimation and testing. Alternatives are a bootstrapping method for selecting samples for validation and X-fold cross-validation with X being the splitting factor. We chose a mix of random selection and 3-fold cross-validation.

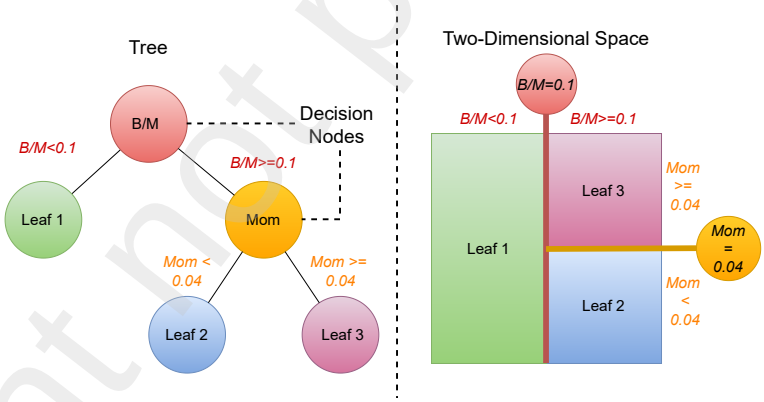


Figure C2: Regression tree. Example of a single tree with 2 nodes and 3 leafs. On the left side is the tree presentation, on the right side is the presentation of that tree in the two dimensions of selection. First, the data gets classified by the book-to-market ratio. The high book-to-market group of data points is then again classified by momentum. This leads to the classification of the data into the leafs shown on the right side, where their associated return is the respective leaf average return.

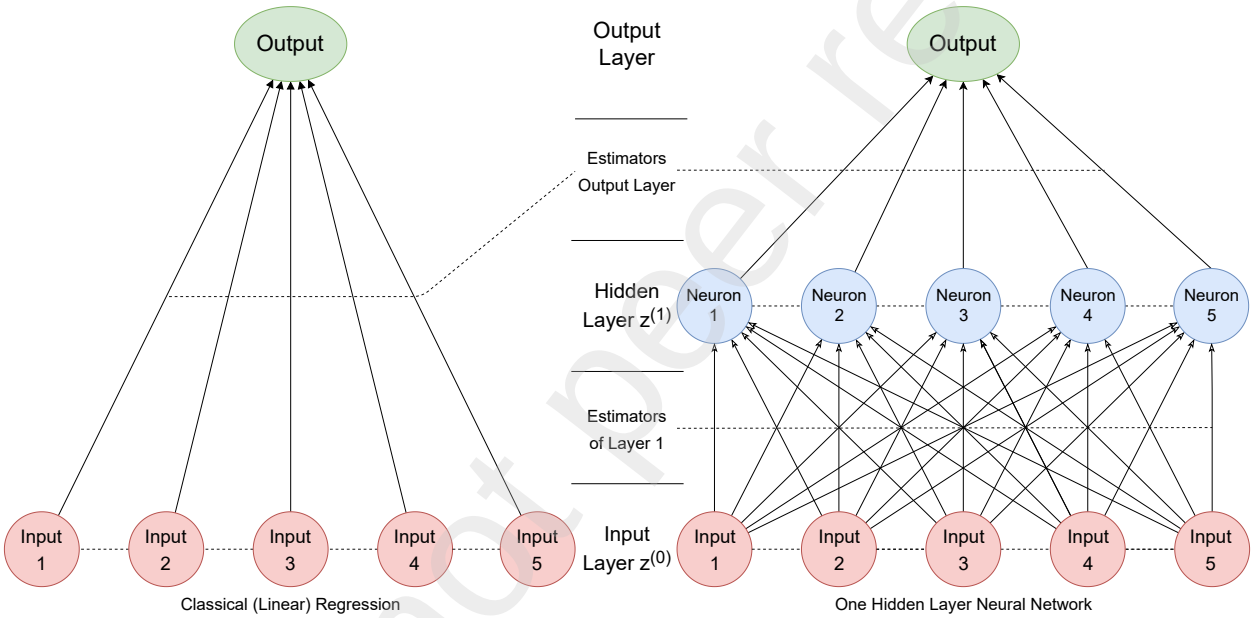


Figure C3: Neural network examples. Red circle indicate input features, blue circles indicate neurons and green circles indicate output knots. Straight continuous lines are synapses. Any neuron and the output knot additionally have an individual constant.

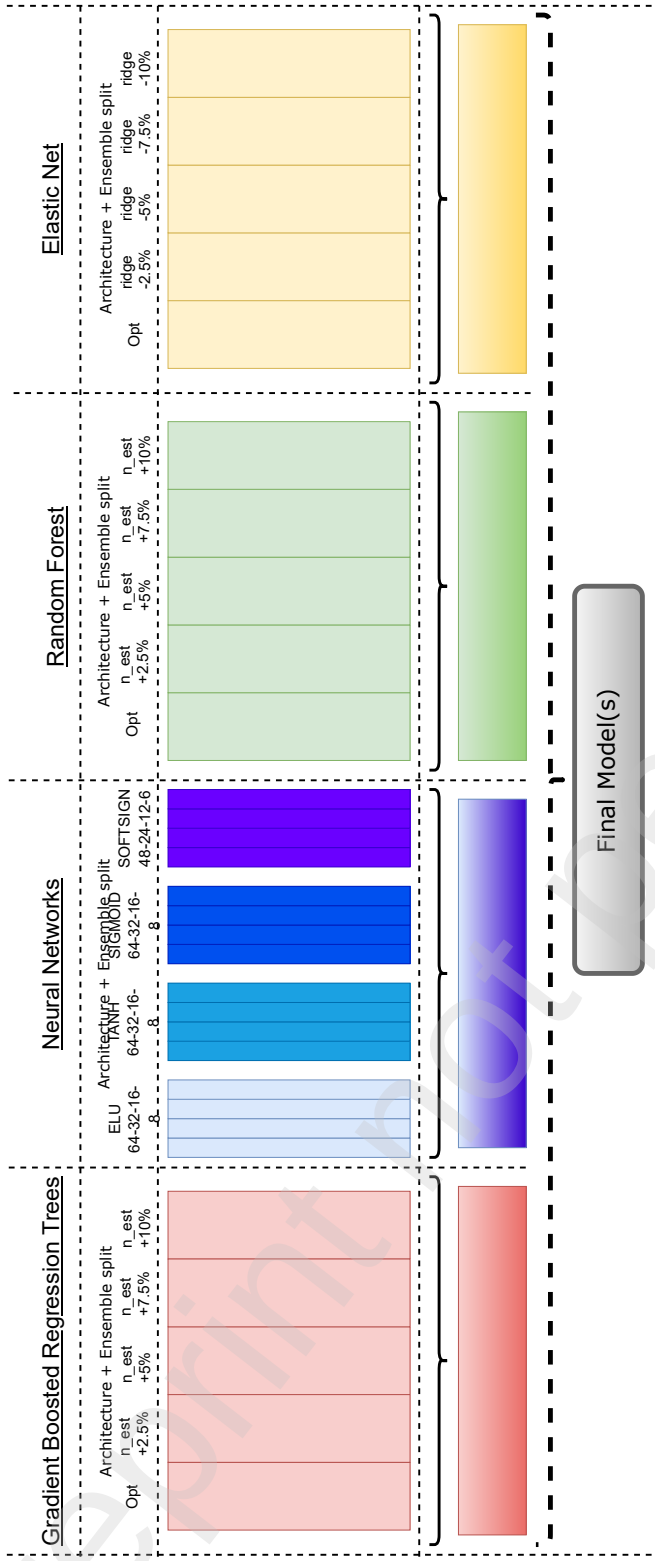


Figure C4: Forecast Combination approach for the used models. Within a model, we distinguish between a pure ensemble split using the same architecture and ensemble splits using different architectures. Every column in the estimation row represents one estimated model. Forecasts are first combined within their model family, after which they may be combined with one or more other algorithms.

C Additional Results

This section contains additional results. First, we present the portfolio results with value weighted decile portfolios. Second, we show additional portfolio stats for submodel performance to highlight the use of multi-level forecast combinations. We also performed an analysis estimated on a value weighted MSE and results of this can be given on request.

C.1 Value weighted portfolios

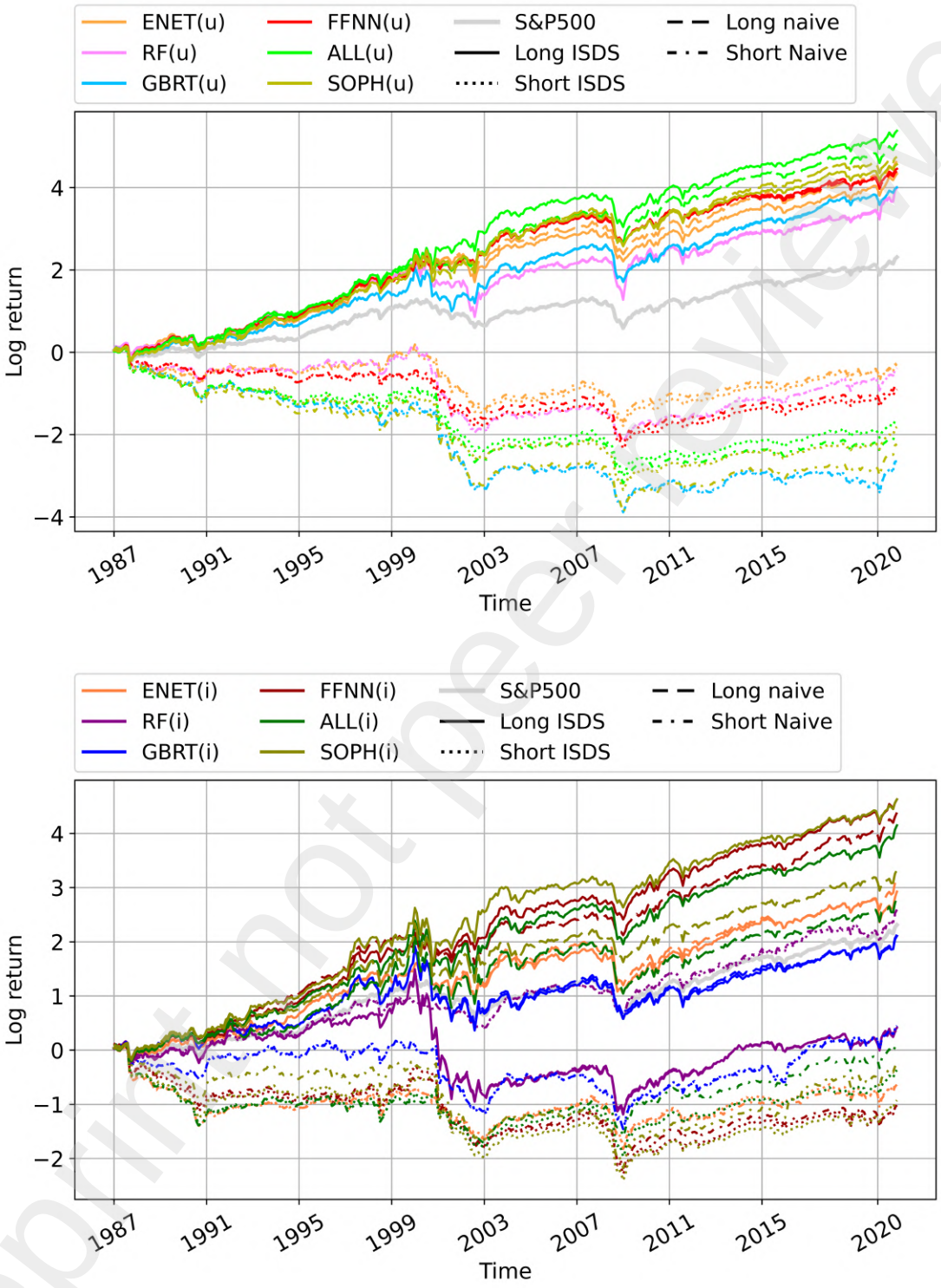


Figure C5: Machine learning (value weighted) decile long- and short portfolios between January 1987 and December 2020. The top figure shows the results on the uninteracted setup, and the lower figure shows the results for the interacted setup. Stocks are equally weighted within each decile. The deciles are based on monthly total excess return predictions. In addition to long and short positions implied by the respective deciles, we show the S&P500 historical total return index.

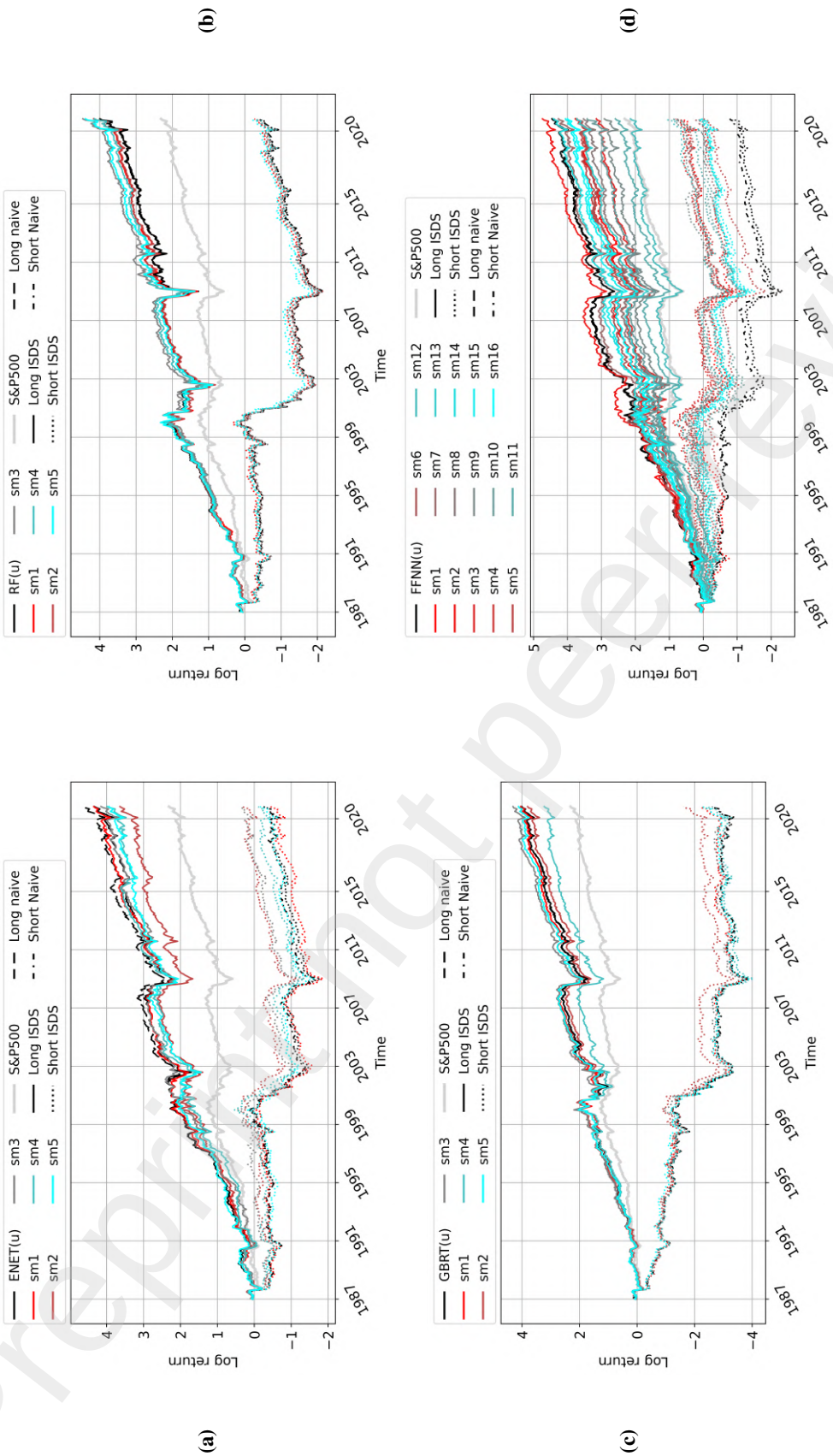


Figure C6: Submodel (value weighted) long- and short portfolio performance between January 1987 and December 2020 estimated on the uninteracted setup. The panels show (**a**): elastic net regression, (**b**): random forest, (**c**): gradient boosted regression trees and (**d**): feed forward neural networks. The forecast combination for the respective performance measurement inside a model family is shown as the black line. Submodels, abbreviated as *sm* followed by a number, are shown in shades from red to blue.

	ENET(u) ISDS				RF(u) ISDS				GBRT(u) ISDS			
	Pred	Mean	Std	SR	Pred	Mean	Std	SR	Pred	Mean	Std	SR
Low	-1.07	0.09	5.51	0.06	-0.14	0.12	6.31	0.07	-1.19	-0.37	7.15	-0.18
2nd	-0.4	0.36	4.81	0.26	0.21	0.36	4.96	0.25	-0.16	0.26	5.71	0.16
3rd	-0.04	0.52	4.67	0.38	0.36	0.47	4.7	0.35	0.24	0.43	4.7	0.32
4th	0.23	0.53	4.42	0.41	0.49	0.52	4.69	0.39	0.51	0.61	4.46	0.47
5th	0.47	0.68	4.46	0.53	0.61	0.61	4.54	0.47	0.72	0.5	4.51	0.39
6th	0.71	0.71	4.42	0.55	0.72	0.72	4.49	0.56	0.93	0.69	4.43	0.54
7th	0.96	0.81	4.52	0.62	0.83	0.74	4.51	0.57	1.14	0.79	4.46	0.61
8th	1.24	0.75	4.68	0.55	0.95	0.82	4.56	0.62	1.37	0.9	4.57	0.68
9th	1.59	1.1	4.9	0.78	1.13	0.93	5.76	0.56	1.66	0.85	5.45	0.54
High	2.17	1.24	5.81	0.74	1.56	1.3	7.99	0.56	2.2	1.21	6.68	0.63
H – L	3.24	1.15	4.44	0.9	1.7	1.18	5.91	0.69	3.39	1.59	5.82	0.94
	ENET(u) naive				RF(u) naive				GBRT(u) naive			
	Pred	Mean	Std	SR	Pred	Mean	Std	SR	Pred	Mean	Std	SR
Low	-1.05	0.1	5.51	0.06	-0.14	0.12	6.31	0.07	-1.19	-0.38	7.13	-0.18
2nd	-0.38	0.35	4.73	0.25	0.21	0.36	4.96	0.25	-0.16	0.26	5.71	0.16
3rd	-0.03	0.57	4.65	0.43	0.36	0.47	4.7	0.35	0.24	0.44	4.7	0.32
4th	0.23	0.49	4.44	0.38	0.49	0.52	4.68	0.39	0.5	0.61	4.49	0.47
5th	0.47	0.64	4.37	0.51	0.61	0.62	4.55	0.47	0.72	0.5	4.5	0.39
6th	0.71	0.79	4.46	0.61	0.72	0.72	4.49	0.56	0.93	0.71	4.41	0.56
7th	0.95	0.83	4.54	0.63	0.83	0.74	4.51	0.57	1.14	0.76	4.47	0.59
8th	1.23	0.77	4.65	0.57	0.95	0.81	4.57	0.62	1.37	0.91	4.56	0.69
9th	1.58	1.04	4.96	0.73	1.13	0.92	5.77	0.55	1.66	0.85	5.44	0.54
High	2.15	1.32	5.8	0.79	1.56	1.3	7.99	0.57	2.19	1.21	6.67	0.63
H – L	3.2	1.22	4.43	0.95	1.7	1.19	5.91	0.7	3.38	1.59	5.79	0.95
	FFNN(u) ISDS				ALL(u) ISDS				SOPH(u) ISDS			
	Pred	Mean	Std	SR	Pred	Mean	Std	SR	Pred	Mean	Std	SR
Low	-0.53	-0.09	5.02	-0.06	-0.62	-0.22	5.7	-0.14	-0.87	-0.27	5.72	-0.16
2nd	-0.32	0.37	4.81	0.26	-0.15	0.27	4.84	0.19	-0.33	0.2	4.82	0.14
3rd	-0.21	0.6	4.81	0.43	0.09	0.35	4.59	0.26	-0.07	0.33	4.79	0.23
4th	-0.12	0.6	4.45	0.47	0.27	0.59	4.65	0.44	0.12	0.63	4.75	0.46
5th	-0.04	0.61	4.63	0.46	0.42	0.61	4.56	0.46	0.29	0.7	4.52	0.53
6th	0.03	0.71	4.59	0.53	0.58	0.64	4.6	0.49	0.46	0.71	4.47	0.55
7th	0.12	0.83	4.53	0.64	0.75	0.86	4.5	0.66	0.64	0.81	4.44	0.63
8th	0.21	1.18	4.87	0.84	0.94	0.95	4.7	0.7	0.85	0.96	4.71	0.71
9th	0.33	1.07	5.1	0.73	1.18	1.09	5.3	0.71	1.1	1.2	5.34	0.78
High	0.52	1.26	5.7	0.77	1.55	1.54	6.52	0.82	1.51	1.33	6.35	0.72
H – L	1.05	1.35	3.64	1.29	2.17	1.76	4.96	1.23	2.37	1.6	4.79	1.15
	FFNN(u) naive				ALL(u) naive				SOPH(u) naive			
	Pred	Mean	Std	SR	Pred	Mean	Std	SR	Pred	Mean	Std	SR
Low	0.21	-0.06	4.91	-0.04	-0.29	-0.27	6.16	-0.15	-0.35	-0.31	6.72	-0.16
2nd	0.42	0.34	4.72	0.25	0.14	0.21	4.98	0.14	0.18	0.16	5.13	0.1
3rd	0.52	0.53	4.73	0.39	0.34	0.43	4.52	0.33	0.41	0.47	4.72	0.34
4th	0.6	0.62	4.44	0.49	0.49	0.53	4.58	0.4	0.56	0.45	4.59	0.34
5th	0.67	0.66	4.58	0.5	0.62	0.64	4.44	0.5	0.69	0.53	4.62	0.39
6th	0.75	0.63	4.85	0.45	0.75	0.68	4.33	0.54	0.82	0.78	4.35	0.62
7th	0.82	0.92	4.77	0.67	0.89	0.8	4.53	0.61	0.96	0.86	4.5	0.66
8th	0.91	1.04	4.66	0.77	1.05	0.89	4.72	0.66	1.11	0.86	4.63	0.64
9th	1.03	1.12	5.11	0.76	1.24	1.08	5.45	0.68	1.3	0.98	5.53	0.61
High	1.21	1.25	5.75	0.76	1.57	1.48	6.8	0.76	1.62	1.4	6.79	0.72
H – L	1.0	1.32	3.62	1.26	1.86	1.76	5.37	1.13	1.97	1.71	5.56	1.06

Table A3: Decile portfolio statistics for (value weighted) decile portfolios. The deciles are based on monthly total excess return predictions. "Mean" is the mean of monthly portfolio returns of stocks in the decile. "Std" is the standard deviation of the returns in the decile. "SR" is the annualised Sharpe ratio of the decile portfolio. "H-L" stands for a portfolio of top decile minus bottom decile.

Panel (a)	Mean	Std	SR(m)	DDmax	Min	Max	Turnover
ENET(u) ISDS	1.15	4.44	0.26	27.9	-15.81	29.99	88.12
ENET(u) naive	1.22	4.43	0.28	26.38	-15.93	29.57	88.52
RF(u) ISDS	1.18	5.91	0.2	32.01	-20.77	36.59	132.41
RF(u) naive	1.19	5.91	0.2	32.02	-20.79	36.58	132.39
GBRT(u) ISDS	1.59	5.82	0.27	40.57	-25.69	42.2	118.67
GBRT(u) naive	1.59	5.79	0.27	40.94	-26.18	41.89	118.52
FFNN(u) ISDS	1.35	3.64	0.37	16.54	-8.3	21.32	116.78
FFNN(u) naive	1.32	3.62	0.36	19.22	-9.77	20.0	113.74
ALL(u) ISDS	1.76	4.96	0.35	23.71	-16.0	35.64	111.24
ALL(u) naive	1.76	5.37	0.33	34.7	-24.13	36.61	111.6
SOPH(u) ISDS	1.6	4.79	0.33	33.29	-18.37	36.56	121.34
SOPH(u) naive	1.71	5.56	0.31	41.43	-23.92	41.37	120.79
market	0.67	4.42	0.15	51.91	-22.72	12.78	0.0
Panel (b)	β	α	t(α)	FF3	t(α)	FF5+mom	t(α)
ENET(u) ISDS	5.03	1.12	5.02	1.12	5.04	0.84	3.87
ENET(u) naive	5.48	1.18	5.33	1.19	5.38	0.91	4.22
RF(u) ISDS	19.48	1.05	3.57	1.05	3.6	0.97	3.22
RF(u) naive	19.46	1.06	3.6	1.06	3.63	0.98	3.26
GBRT(u) ISDS	-8.72	1.64	5.64	1.65	5.77	1.42	5.14
GBRT(u) naive	-8.32	1.65	5.68	1.66	5.81	1.43	5.18
FFNN(u) ISDS	11.59	1.28	7.05	1.27	7.03	1.07	6.07
FFNN(u) naive	14.12	1.22	6.83	1.22	6.81	1.04	5.98
ALL(u) ISDS	10.28	1.69	6.82	1.69	6.87	1.43	6.05
ALL(u) naive	9.82	1.69	6.3	1.69	6.29	1.52	5.77
SOPH(u) ISDS	9.29	1.54	6.41	1.53	6.38	1.32	5.77
SOPH(u) naive	-2.07	1.72	6.17	1.72	6.16	1.49	5.55

Table A4: Monthly (value weighted) HML portfolio performance measures. "Mean", "Std" and "SR(m)" describe mean, standard deviation and monthly Sharpe ratio of returns. "DDmax" is the maximum drawdown, the maximum loss from the previous high water mark. "Min" and "Max" describe the minimum and maximum monthly return. "Turnover" describes the monthly readjustment factor. "β" is systematic risk to the market return, "α" is Jensen's alpha to the market with its corresponding t value. FF3 and FF5+mom are the αs of the portfolios to the respective factor models based on Fama and French (1993) and Fama and French (2015) plus momentum.

C.2 Submodels (equally weighted portfolios)

In this section, we present results for the submodels portfolio performance in the first level forecast combination within the model family.

	mean	std	sr(m)	DDmax	min	max	turnover
ENET(u) ISDS	2.65	4.28	0.62	12.24	-7.94	45.39	88.12
ENET(u) naive	2.59	4.36	0.6	13.86	-8.83	45.07	88.52
sm1	2.44	4.3	0.57	17.35	-8.66	45.15	94.39
sm2	2.1	5.0	0.42	31.64	-18.6	53.64	90.27
sm3	1.97	4.24	0.46	27.35	-16.68	28.77	98.23
sm4	2.24	4.76	0.47	31.16	-14.23	43.75	95.02
sm5	2.21	4.18	0.53	32.04	-19.99	25.28	97.3

Table C1: Monthly submodel performance of ENET(u) against sub models. "sm" and a number is a submodel of the first layer forecast combination within each architecture. See Table 3 panel(a) for description.

	mean	std	sr(m)	DDmax	min	max	turnover
RF(u) ISDS	2.37	5.54	0.43	27.94	-13.44	62.5	132.41
RF(u) naive	2.37	5.54	0.43	27.95	-13.56	62.53	132.39
sm1	2.35	5.44	0.43	24.46	-11.98	59.49	131.14
sm2	2.38	5.46	0.44	28.11	-13.28	60.29	131.66
sm3	2.38	5.5	0.43	28.27	-15.08	60.26	131.36
sm4	2.37	5.5	0.43	24.63	-12.31	61.74	131.19
sm5	2.36	5.41	0.44	26.89	-13.53	59.76	131.29

Table C2: Monthly submodel performance of RF(u) against sub models. "sm" and a number is a submodel of the first layer forecast combination within each architecture. See Table 3 panel(a) for description.

	mean	std	sr(m)	DDmax	min	max	turnover
GBRT(u) ISDS	3.36	5.02	0.67	39.47	-14.92	52.34	118.67
GBRT(u) naive	3.36	5.05	0.67	39.41	-14.93	52.91	118.52
sm1	3.14	5.01	0.63	40.86	-14.18	53.86	113.31
sm2	3.04	4.82	0.63	39.75	-14.63	48.11	113.55
sm3	3.19	4.71	0.68	37.81	-13.57	45.26	113.53
sm4	3.18	4.8	0.66	38.24	-15.23	49.54	113.38
sm5	3.18	4.71	0.67	36.82	-15.69	46.95	113.38

Table C3: Monthly submodel performance of GBRT(u) against sub models. "sm" and a number is a submodel of the first layer forecast combination within each architecture. See Table 3 panel(a) for description.

	mean	std	sr(m)	DDmax	min	max	turnover
FFNN(u) ISDS	2.86	3.54	0.81	17.9	-9.27	19.31	116.78
FFNN(u) naive	2.87	3.64	0.79	17.07	-10.1	27.17	113.74
sm1	1.9	3.08	0.62	16.23	-11.68	21.87	119.57
sm2	1.68	3.47	0.48	17.57	-12.42	22.72	114.74
sm3	1.62	3.95	0.41	31.03	-16.22	37.72	117.4
sm4	1.79	3.49	0.51	18.95	-9.66	24.53	116.93
sm5	1.71	3.3	0.52	36.78	-13.42	14.81	119.17
sm6	1.62	3.66	0.44	26.19	-11.71	39.39	118.13
sm7	1.79	3.2	0.56	24.57	-12.55	14.97	121.44
sm8	1.88	3.15	0.6	16.93	-6.11	16.32	116.47
sm9	1.21	3.64	0.33	21.02	-11.28	21.72	124.16
sm10	1.14	4.2	0.27	27.09	-27.09	22.08	121.52
sm11	0.67	4.12	0.16	49.78	-18.06	22.52	124.24
sm12	0.82	4.04	0.2	48.51	-38.11	16.65	124.49
sm13	2.01	2.87	0.7	12.93	-7.53	17.12	103.16
sm14	2.16	3.02	0.72	14.13	-7.05	17.99	103.01
sm15	1.98	3.07	0.64	12.81	-7.4	21.68	102.95
sm16	2.1	2.86	0.73	12.27	-9.02	18.64	101.62

Table C4: Monthly submodel performance of FFNN(u) against sub models. "sm" and a number is a submodel of the first layer forecast combination within each architecture. See Table 3 panel(a) for description.

C.3 Submodels (value weighted portfolios)

In this section, we present results for the submodels portfolio performance of value weighted HML portfolios in the first level forecast combination within the model family. This includes returns, Sharpe ratios, and common risk measures.

	mean	std	sr(m)	DDmax	min	max	turnover
ENET(u) ISDS	1.15	4.44	0.26	27.9	-15.81	29.99	88.12
ENET(u) naive	1.22	4.43	0.28	26.38	-15.93	29.57	88.52
sm1	1.16	4.57	0.25	21.44	-15.27	30.41	94.39
sm2	0.86	4.86	0.18	39.81	-22.37	24.5	90.27
sm3	0.91	4.43	0.21	37.81	-23.25	26.95	98.23
sm4	1.02	4.48	0.23	39.43	-14.82	28.19	95.02
sm5	1.02	4.13	0.25	31.73	-18.28	30.82	97.3

Table C5: Monthly submodel performance of ENET(u) against sub models on (value weighted) HML portfolios. "sm" and a number is a submodel of the first layer forecast combination within each architecture. See Table 3 panel(a) for description.

	mean	std	sr(m)	DDmax	min	max	turnover
RF(u) ISDS	1.18	5.91	0.2	32.01	-20.77	36.59	132.41
RF(u) naive	1.19	5.91	0.2	32.02	-20.79	36.58	132.39
sm1	1.19	5.91	0.2	32.69	-19.97	36.7	131.14
sm2	1.19	5.85	0.2	28.51	-19.42	36.11	131.66
sm3	1.27	6.02	0.21	23.54	-18.48	37.63	131.36
sm4	1.18	5.69	0.21	22.76	-17.39	36.04	131.19
sm5	1.23	5.88	0.21	33.54	-20.04	35.98	131.29

Table C6: Monthly submodel performance of RF(u) against sub models on (value weighted) HML portfolios. "sm" and a number is a submodel of the first layer forecast combination within each architecture. See Table 3 panel(a) for description.

	mean	std	sr(m)	DDmax	min	max	turnover
GBRT(u) ISDS	1.59	5.82	0.27	40.57	-25.69	42.2	118.67
GBRT(u) naive	1.59	5.79	0.27	40.94	-26.18	41.89	118.52
sm1	1.48	5.72	0.26	46.62	-21.69	37.8	113.31
sm2	1.29	5.53	0.23	45.46	-27.99	33.69	113.55
sm3	1.59	5.55	0.29	40.95	-25.35	42.29	113.53
sm4	1.36	5.51	0.25	46.11	-26.11	37.22	113.38
sm5	1.54	5.41	0.28	44.83	-26.16	39.5	113.38

Table C7: Monthly submodel performance of GBRT(u) against sub models on (value weighted) HML portfolios. "sm" and a number is a submodel of the first layer forecast combination within each architecture. See Table 3 panel(a) for description.

	mean	std	sr(m)	DDmax	min	max	turnover
FFNN(u) ISDS	1.35	3.64	0.37	16.54	-8.3	21.32	116.78
FFNN(u) naive	1.32	3.62	0.36	19.22	-9.77	20.0	113.74
sm1	0.8	3.46	0.23	22.66	-19.03	17.84	119.57
sm2	1.11	3.69	0.3	20.69	-7.36	20.19	114.74
sm3	0.62	3.78	0.16	41.29	-13.52	21.57	117.4
sm4	0.77	3.69	0.21	31.59	-13.01	22.55	116.93
sm5	0.95	3.55	0.27	30.69	-11.97	14.33	119.17
sm6	0.78	3.82	0.2	40.36	-19.78	25.84	118.13
sm7	1.01	3.46	0.29	27.72	-11.79	19.01	121.44
sm8	0.69	3.31	0.21	24.98	-11.96	18.74	116.47
sm9	0.54	3.63	0.15	28.83	-14.24	15.61	124.16
sm10	0.82	3.54	0.23	21.71	-10.86	19.48	121.52
sm5	0.95	3.55	0.27	30.69	-11.97	14.33	119.17
sm11	0.35	4.37	0.08	53.82	-22.92	29.62	124.24
sm12	0.2	3.71	0.05	51.69	-23.98	15.21	124.49
sm13	0.91	3.06	0.3	13.99	-10.74	14.59	103.16
sm14	1.09	2.84	0.38	16.9	-7.52	10.79	103.01
sm15	0.99	3.48	0.28	17.52	-10.83	21.61	102.95
sm16	0.97	2.91	0.33	15.64	-9.24	16.38	101.62

Table C8: Monthly submodel performance of FFN(u) against sub models on (value weighted) HML portfolios. "sm" and a number is a submodel of the first layer forecast combination within each architecture. See Table 3 panel(a) for description.

D References

Abarbanell, J. S., & Bushee, B. J. (1998). Abnormal returns to a fundamental analysis strategy. *The Accounting Review*, *73*, 19–45.

Ali, A., Hwang, L.-S., & Trombley, M. A. (2003). Arbitrage risk and the book-to-market anomaly. *Journal of Financial Economics*, *69*(2), 355–373.

Almeida, H., & Campello, M. (2007). Financial constraints, asset tangibility, and corporate investment. *The Review of Financial Studies*, *20*(5), 1429–1460.

Amihud, Y. (2002). Illiquidity and stock returns: Cross-section and time-series effects. *Journal of Financial Markets*, *5*(1), 31–56.

Amihud, Y., & Mendelson, H. (1989). The effects of beta, bid–ask spread, residual risk, and size on stock returns. *The Journal of Finance*, *44*(2), 479–486.

Anderson, C. W., & Garcia-Feijóo, L. (2006). Empirical evidence on capital investment, growth options, and security returns. *The Journal of Finance*, *61*(1), 171–194.

Ang, A., Hodrick, R. J., Xing, Y., & Zhang, X. (2006). The cross-section of volatility and expected returns. *The Journal of Finance*, *61*(1), 259–299.

Asness, C. S., Porter, R. B., & Stevens, R. L. (2000). Predicting stock returns using industry-relative firm characteristics. *Working Paper*.

Balakrishnan, K., Bartov, E., & Faurel, L. (2010). Post loss/profit announcement drift. *Journal of Accounting and Economics*, *50*(1), 20–41.

Bali, T. G., Cakici, N., & Whitelaw, R. F. (2011). Maxing out: Stocks as lotteries and the cross-section of expected returns. *Journal of Financial Economics*, *99*(2), 427–446.

Bandyopadhyay, S. P., Huang, A. G., & Wirjanto, T. S. (2010). The accrual volatility anomaly. *Working Paper*.

Banz, R. W. (1981). The relationship between return and market value of common stocks. *Journal of Financial Economics*, *9*(1), 3–18.

Barbee Jr, W. C., Mukherji, S., & Raines, G. A. (1996). Do sales–price and debt–equity explain stock returns better than book–market and firm size? *Financial Analysts Journal*, *52*(2), 56–60.

Barth, M. E., Elliott, J. A., & Finn, M. W. (1999). Market rewards associated with patterns of increasing earnings. *Journal of Accounting Research*, *37*(2), 387–413.

Basu, S. (1977). Investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis. *The Journal of Finance*, *32*(3), 663–682.

Belo, F., Lin, X., & Bazdresch, S. (2014). Labor hiring, investment, and stock return predictability in the cross section. *Journal of Political Economy*, *122*(1), 129–177.

Bhandari, L. C. (1988). Debt/equity ratio and expected common stock returns: Empirical evidence. *The Journal of Finance*, *43*(2), 507–528.

Brandt, M. W., Kishore, R., Santa-Clara, P., & Venkatachalam, M. (2008). Earnings announcements are full of surprises. *Working Paper*.

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

Brown, D. P., & Rowe, B. (2007). The productivity premium in equity returns. *Working Paper*.

Chandrashekar, S., & Rao, R. K. (2009). *The productivity of cash and the cross-section of expected stock returns* (Working Paper). University of Texas.

Chordia, T., Subrahmanyam, A., & Anshuman, V. R. (2001). Trading activity and expected stock returns. *Journal of Financial Economics*, *59*(1), 3–32.

Cooper, M. J., Gulen, H., & Schill, M. J. (2008). Asset growth and the cross-section of stock returns. *The Journal of Finance*, *63*(4), 1609–1651.

Datar, V. T., Naik, N. Y., & Radcliffe, R. (1998). Liquidity and stock returns: An alternative test. *Journal of Financial Markets*, *1*(2), 203–219.

Desai, H., Rajgopal, S., & Venkatachalam, M. (2004). Value-glamour and accruals mispricing: One anomaly or two? *The Accounting Review*, *79*(2), 355–385.

Eberhart, A. C., Maxwell, W. F., & Siddique, A. R. (2004). An examination of long-term abnormal stock returns and operating performance following R&D increases. *The Journal of Finance*, *59*(2), 623–650.

Eisfeldt, A. L., & Papanikolaou, D. (2013). Organization capital and the cross-section of expected returns. *The Journal of Finance*, *68*(4), 1365–1406.

Fairfield, P. M., Whisenant, J. S., & Yohn, T. L. (2003). Accrued earnings and growth: Implications for future profitability and market mispricing. *The Accounting Review*, *78*(1), 353–371.

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, *33*(1), 3–56.

Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, *116*(1), 1–22.

Fama, E. F., & MacBeth, J. D. (1973). Risk, return, and equilibrium: Empirical tests. *Journal of Political Economy*, *81*(3), 607–636.

Francis, J., LaFond, R., Olsson, P. M., & Schipper, K. (2004). Costs of equity and earnings attributes. *The Accounting Review*, *79*(4), 967–1010.

Gettleman, E., & Marks, J. M. (2006). Acceleration strategies. *Working Paper*.

Green, J., Hand, J. R., & Zhang, X. F. (2013). The supraview of return predictive signals. *Review of Accounting Studies*, *18*(3), 692–730.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*(5), 2223–2273.

Guo, R.-J., Lev, B., & Shi, C. (2006). Explaining the short-and long-term IPO anomalies in the US by R&D. *Journal of Business Finance & Accounting*, *33*(3-4), 550–579.

Hafzalla, N., Lundholm, R., & Matthew Van Winkle, E. (2011). Percent accruals. *The Accounting Review*, *86*(1), 209–236.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised learning. In *Elements of statistical learning* (pp. 485–585). Springer, Luxembourg.

Holthausen, R. W., & Larcker, D. F. (1992). The prediction of stock returns using financial statement information. *Journal of Accounting and Economics*, *15*(2-3), 373–411.

Hong, H., & Kacperczyk, M. (2009). The price of sin: The effects of social norms on markets. *Journal of Financial Economics*, *93*(1), 15–36.

Hou, K., & Moskowitz, T. J. (2005). Market frictions, price delay, and the cross-section of expected returns. *The Review of Financial Studies*, *18*(3), 981–1020.

Hou, K., & Robinson, D. T. (2006). Industry concentration and average stock returns. *The Journal of Finance*, *61*(4), 1927–1956.

Hou, K., Xue, C., & Zhang, L. (2015). Digesting anomalies: An investment approach. *The Review of Financial Studies*, *28*(3), 650–705.

Huang, A. G. (2009). The cross section of cashflow volatility and expected stock returns. *Journal of Empirical Finance*, *16*(3), 409–429.

Jegadeesh, N. (1990). Evidence of predictable behavior of security returns. *The Journal of Finance*, *45*(3), 881–898.

Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, *48*(1), 65–91.

Jiang, G., Lee, C. M., & Zhang, Y. (2005). Information uncertainty and expected returns. *Review of Accounting Studies*, *10*(2-3), 185–221.

Kama, I. (2009). On the market reaction to revenue and earnings surprises. *Journal of Business Finance & Accounting*, *36*(1-2), 31–50.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, *14*(2), 1137–1145.

Lakonishok, J., Shleifer, A., & Vishny, R. W. (1994). Contrarian investment, extrapolation, and risk. *The Journal of Finance*, *49*(5), 1541–1578.

Lerman, A., Livnat, J., & Mendenhall, R. R. (2007). The high-volume return premium and post-earnings announcement drift. *Working Paper*.

Lev, B., & Nissim, D. (2004). Taxable income, future earnings, and equity values. *The Accounting Review*, *79*(4), 1039–1074.

Litzenberger, R. H., & Ramaswamy, K. (1979). The effect of personal taxes and dividends on capital asset prices: Theory and empirical evidence. *Journal of Financial Economics*, *7*(2), 163–195.

Liu, W. (2006). A liquidity-augmented capital asset pricing model. *Journal of Financial Economics*, *82*(3), 631–671.

Michaely, R., Thaler, R. H., & Womack, K. L. (1995). Price reactions to dividend initiations and omissions: Overreaction or drift? *The Journal of Finance*, *50*(2), 573–608.

Mohanram, P. S. (2005). Separating winners from losers among low book-to-market stocks using financial statement analysis. *Review of Accounting Studies*, *10*(2), 133–170.

Moskowitz, T. J., & Grinblatt, M. (1999). Do industries explain momentum? *The Journal of Finance*, *54*(4), 1249–1290.

Novy-Marx, R. (2013). The other side of value: The gross profitability premium. *Journal of Financial Economics*, *108*(1), 1–28.

Ou, J. A., & Penman, S. H. (1989). Financial statement analysis and the prediction of stock returns. *Journal of Accounting and Economics*, *11*(4), 295–329.

Palazzo, B. (2012). Cash holdings, risk, and expected returns. *Journal of Financial Economics*, *104*(1), 162–185.

Piotroski, J. D., et al. (2000). Value investing: The use of historical financial statement information to separate winners from losers. *Journal of Accounting Research*, *38*, 1–52.

Pontiff, J., & Woodgate, A. (2008). Share issuance and cross-sectional returns. *The Journal of Finance*, *63*(2), 921–945.

Rao, R. B., Fung, G., & Rosales, R. (2008). On the dangers of cross-validation. an experimental evaluation. *Proceedings of the 2008 SIAM International Conference on Data Mining*.

Richardson, S. A., Sloan, R. G., Soliman, M. T., & Tuna, I. (2005). Accrual reliability, earnings persistence and stock prices. *Journal of Accounting and Economics*, *39*(3), 437–485.

Rosenberg, B., Reid, K., & Lanstein, R. (1985). Persuasive evidence of market inefficiency. *The Journal of Portfolio Management*, *11*(4), 9–16.

Sloan, R. G. (1996). Do stock prices fully reflect information in accruals and cash flows about future earnings? *The Accounting Review*, *71*(3), 289–315.

Soliman, M. T. (2008). The use of dupont analysis by market participants. *The Accounting Review*, *83*(3), 823–853.

Thomas, J., & Zhang, F. X. (2011). Tax expense momentum. *Journal of Accounting Research*, *49*(3), 791–821.

Thomas, J., & Zhang, H. (2002). Inventory changes and future returns. *Review of Accounting Studies*, *7*(2-3), 163–187.

Titman, S., Wei, K. J., & Xie, F. (2004). Capital investments and stock returns. *Journal of Financial and Quantitative Analysis*, *39*(4), 677–700.

Tuzel, S. (2010). Corporate real estate holdings and the cross-section of stock returns. *The Review of Financial Studies*, *23*(6), 2268–2302.

Valta, P. (2016). Strategic default, debt structure, and stock returns. *Journal of Financial and Quantitative Analysis*, *51*(1), 197–229.

Welch, I., & Goyal, A. (2007). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, *21*(4), 1455–1508.