# Evaluation of Machine Learning in Empirical Asset Pricing

Ze Yu Zhong

Tuesday 1st October, 2019

## Contents

# 1. Introduction

## *1.1. Topic*

This thesis aims to evaluate the application of machine learning algorithms in empirical asset pricing. While there has been significant recent interest in applying machine learning to the problem of predicting asset returns, there is little literature that focuses on how well these algorithms are at capturing true underlying variables in determining stock returns. 12 different simulated datasets ranging from linear to highly non-linear data generating processes incorporating observed phenomena of cross sectional correlation, persistence, and stochastic volatility, in addition to real world data will be used to assess the performance of linear models, elastic net models, random forests and neural networks. Model performance will be assessed according to their out of sample Mean Absolute Error, Root Mean Square Error, and Predictive $R^2$, in addition to whether or not they were able to identify the correct variables in the data generating process according to a variable importance metric.

## *1.2. Background Literature and Motivations*

This paper is motivated by evaluating the performance of machine learning algorithms in empirical asset pricing, focusing on how well they deal with the many unique problems in financial returns data. Here, we define "performance" to refer to two forms of metrics conventional in the literature: $R^2$ (and more specifically, out of sample $R^2$), and forecast errors (see 2.3.5 for more details).

We first begin by defining "factors" with the more contemporary definition as suggested by Harvey et al. (2016): a collection of regressors to be used in pricing returns that can be used to proxy for unknown underlying risk factors due to their correlation with cross sectional returns. Most notably, their definition rejects the more strict view that risk factors should be variables that have unpredictable variation through time, and that they should be able explain cross sectional return patterns. Harvey et al. (2016) further groups factors into the two broad categories of "common" and individual firm "characteristics." "Common" factors under this definition can be viewed as proxies for sources of risk constant across all firms, such as macroeconomic variables. Examples of this include the Fama-French factors, which are portfolios of assets sorted by certain characteristics to proxy for them. Harvey et al. (2016) notes that individual firm characteristics are unlikely to satisfy the more strict definition because they are often pre-known and display limited time series variation.

Because of this less strict definition, factors introduced and used in the literature often exhibit properties which makes them unsuitable for inclusion in models, such as high persistence, high levels of non-stationarity and cross sectional correlation.

Goetzmann and Jorion (1993) and Ang and Bekaert (2006) note the persistence present in dividend ratio factors. This means that movements in dividend ratios are dominated by movements in price and therefore dividend ratios are correlated with lagged dependent variables on the right hand side of the regression equation. This violates the assumptions of exogeneous regressors (independent from the error term) required for traditional regression models (ordinary least squares) to be unbiased, resulting in t statistics which are biased upwards and increase with time horizon due to autocorrelated errors. Importantly, Goetzmann and Jorion (1993) show that corrections to t statistics using the Generalized Method of Moments and Newey-West standard errors also appear to be biased upwards, making them unreliable.

Goyal and Welch (2003) provide a more comprehensive study on the performance of lagged dividend price ratios, with specific focus on out of sample predictive performance both in terms of $R^2$ and forecast

errors. They conclude that while models incorporating dividend related factors were able to achieve higher in sample performance prior to 1990 than the historical mean, they could not have outperformed the historical mean *out of sample*. Goyal and Welch (2003) attribute this to the increasing persistence and non-stationarity of dividend ratios, noting that they have become like random walks as of 2001. This mirrors the sentiment of (Lettau and Ludvigson (2001), Schwert (2003) and others) who conclude that models incorporating dividend ratios seemed to break down in the 2000s due to a changing economic environment despite having performed well in the 1990s.

Despite the controversy, the prevailing tone within the literature was that various factors such as dividend ratios, earnings price ratio, interest and inflation and other financial indicators were able to predict excess returns, with Lettau and Ludvigson (2001) remarking that this was now "widely accepted." However, Welch and Goyal (2008) extend upon the work of Goyal and Welch (2003) by including a more comprehensive set of variables and time horizons. They conclude that not a single variable had any statistical forecasting power. Crucially, they demonstrate the non-robustness of models incorporating these factors by showing that the significance values of some factors change with the choice of sample periods.

Despite this, the literature has continued to produce more factors: quantitative trading firms were using 81 factor models as the norm by 2014 (Hsu and Kalesnik, 2014), and Harvey and Liu (2019) currently document well over 600 different factors suggested in the literature.

The dramatic increase in the number of factors alone poses challenges to traditional statistical techniques. Harvey et al. (2016) detail the false discovery problem when the number of potential factors is extremely high. The significance of a factor in a traditional regression setting is determined by a single hypothesis test, which by construction carries a level of significance $\alpha$ controlling the type I error rate: the probability of rejecting a "null" hypothesis that a factor is not important, and hence incorrectly concluding that it is significant. When the number of potential factors is large, it is very likely that a factor will be concluded as significant by pure chance. For example, a factor model consisting of 600 factors would find around 30 factors significant by chance at the 5% significance level. Harvey et al. (2016) produce a multiple testing framework to mitigate this, and conclude that many of the historically discovered would have been deemed significant by chance.

Furthermore, Feng et al. (2019) note that the number of potential factors discovered in the literature has increased to the same scale as, if not greater, than the number of stocks considered in a typical portfolio, or the time horizon, producing highly inefficient covariances in a standard cross sectional regression. Moreover, when the number of factors exceeds the sample size, traditional cross sectional regressions become infeasible and do not produce solutions altogether.

It does not help that many factors are cross sectionally correlated, meaning that factors which are discovered to be significant may simply be so because they are correlated with a true, underlying factor and do not provide independent information themselves, a concern which Cochrane (2011) calls the multidimensional challenge. Freyberger et al. (2017) notes that this is especially challenging for traditional regression models, which make strong functional form assumptions and are sensitive to outliers.

More recently, machine learning has emerged within the literature and appear to be well suited to the task of predicting asset returns. The definition of machine learning can be vague and is often context specific; Hastie et al. (2009) in *An Introduction to Statistical Learning* describes statistical (machine) learning as a vast set of tools for understanding data, and *supervised* learning specifically as the process of building a statistical model for the prediction or estimation of an output based on input(s). In the context of asset pricing, we use the term to refer to a diverse collection of:

1. high-dimensional models for statistical prediction,

2. the "regularization" methods for model selection and mitigation of over-fitting input data,

3. and the efficient systematic methods for searching potential model specifications.

The high dimensional and hence flexible nature of machine learning brings more hope to approximating unknown and likely complex data generating processes that underlie excess returns. The flexibility however, comes at a cost of potentially over-fitting in sample data (referred to as training data in the machine learning literature), generalizing poorly and producing poor forecasts. The regularization aspect of machine learning explicitly guards against over-fitting problems and emphasizes out of sample performance. The most explicit example of regularization is the splitting of the dataset into an explicit "training" set used for model fitting, and a "test" set withheld and used solely for evaluating out of sample performance. Finally, machine learning offers tools which are designed to produce an optimal model specification from all possible models with manageable computational cost, all in a systematically consistent way.

The financial literature already has some applications of machine learning. For example, Kozak et al. (2017), Rapach and Zhou (2013) and Freyberger et al. (2017) all apply shrinkage and selection methods from machine learning to assist with the problem of factor selection.

Most importantly, portfolios constructed using machine learning have been demonstrated to outperform traditional models in predicting stock returns (Gu et al. (2018), Hsu and Kalesnik (2014) and Feng et al. (2018)) in terms of out of sample predictive $R^2$ and Sharpe Ratios. Gu et al. (2018) attribute this to machine learning's ability to evaluate and consider non-linear complexities among factors that cannot be feasibly achieved using traditional techniques.

However, there is little work done on how machine learning actually recognises and deals with the challenges of returns prediction documented in the literature. Prior work has been done by Gu et al. (2018), however; only basic simulation designs which were not representative of real financial data were considered. Therefore, the performance of machine learning in empirical financial contexts with specific regards to characteristics such as stochastic and time varying volatility, long term dependence and cross sectionally correlated regressors has not been adequately explored.

Furthermore, Feng et al. (2018) in particular use cross validation as part of their model building procedure, destroying the temporal aspect of returns data, in addition to only using a handful of factors. Gu et al. (2018) produce models using a training sample which ends in the 1970s to ultimately produce forecasts for the most recent 30 years. Given the non-robustness of financial data affecting even traditional regressions which are considered to be more inflexible, more research should be done into the robustness of more flexible machine learning methods with regards to sample selection and periods of returns predictability.

For the aspect of factor selection specifically, Gu et al. (2018) concludes that all of the machine methods agree on the same subset of important factors. However, while their factor importance metrics for regression-tree based methods and neural networks (the most complex methods considered) are mostly consistent, they have differences in terms of the relative importance of each factor, in addition to completely different conclusions for the dividend yield factor.

This paper is the first in focusing on how machine learning algorithms perform in environments with problems exhibited by financial returns data through extending the simulation designs of Gu et al. (2018). In addition, these algorithms will once again be evaluated on real world data, but with only more recent and representative data included in order to test their short term robustness in predicting stock returns.

These two aspects of the study together are able to offer a better glimpse as to how "black box" machine learning algorithms deal with the challenges present in asset pricing, if at all.

# 2. Model Specification and Methodology

## *2.1. Model Specification*

All asset excess monthly returns denoted as $r_{i,t+1}$ are modelled as an additive prediction error model conditional on the true and unobservable information set available to market participants up to and including time $t$, $\mathcal{F}_t$):

$$r_{i,t+1} = E(r_{i,t+1}|\mathcal{F}_t) + \epsilon_{i,t+1} \tag{1}$$

where

$$E(r_{i,t+1}|\mathcal{F}_t) = g^*(z_{i,t}) \tag{2}$$

with stocks indexed as $i = 1, \ldots, N$ and time domain by $t = 1, \ldots, T$. $g^*(z_{i,t})$ represents the model approximation using the $P$ dimensional predictor set $z_{i,t}$. We allow $g^*(z_{i,t})$ to be a flexible function of the predictor set $z_{i,t}$, and most notably, not depend on $i$ or $t$ directly. This means that we do not re-estimate a model for each time period, or independently estimate a model for each stock. Note that $g^*(z_{i,t})$ only contains information in time $t$ for individual stock $i$, meaning that while the model and its parameters will be estimated using $\mathcal{F}_t$ for stock $i$, predictions for $r_{i,t+1}$ will only use information at time $t$ as an input, analogous to using variables lagged by one period.

All machine learning methods are designed to approximate the empirical model $E_t(r_{i,t+1}) = g*(z_{i,t})$ defined in equation (2). We define the baseline set of stock-level covariates $z_{i,t}$ as:

$$z_{i,t} = x_t \otimes c_{i,t} \tag{3}$$

where $c_{i,t}$ is a $P_c \times 1$ matrix of characteristics for each stock $i$, and $x_t$ is a $P_x \times 1$ vector of macroeconomic predictors (and are this common to all stocks, including a constant). Thus $z_{i,t}$ is a $P \times 1$ vector of features for predicting individual stock returns ($P = P_c P_x$) and includes interactions between individual characteristics and macroeconomic characteristics.

## *2.2. Methodology*

Each model will be presented and explained so that a reader without any machine learning background can understand the basic idea behind each model.

### *2.2.1. Sample Splitting*

Imperative to any machine learning technique is the establishment of how the dataset is to be split into training, validation and test sets. The training set is used to initially build the model and provide initial estimates of parameters, whereas the validation set is used to tune model parameters to optimise out of sample performance, thus preventing overfitting. The validation set acts as a simulation of out of sample testing, whereas the test set is used only for evaluation, and is thus truly out of sample.

There are three main approaches to splitting temporal data (such as financial data).

The first is to decide arbitrarily on a single training, validation and test set. This method is straightforward and the least computationally intensive, but is limited and inflexible in evaluating how models perform when more recent data is provided for training.

The second method is a "rolling window" method, where a fixed size or "window" for the training and validation set is first chosen. This window then incrementally move forwards in time to include more recent data, with a set of forecasts for the test sets made for all possible windows.

The third is a "recursive" method, which is the same as the rolling window method, but different in that the training set always contains previous data, with only the validation set staying fixed in size and "rolling" forwards. Hence, it is also referred to as a "growing window."

Both the rolling window and recursive schemes are very computationally intensive. Therefore, a hybrid of the rolling and recursive schemes was considered: the training set is increased by one year with each refit, the validation set remains one year in length but moves forward by one year, and forecasts are made using that model for the subsequent year. The "traditional" cross validation method of randomly sampling to determine a train and validation set was not done to maintain the temporal ordering of the data.

#### *2.2.2. Loss Function*

The choice of the loss function used in models is imperative to machine learning. The loss functions considered are Mean Absolute Error (MAE) and Mean Squared Error [1].

The mean absolute error (MAE) is simply the average magnitude of errors. Because of this, it places equal weighting to all magnitudes of errors and is more robust to outliers.

$$\text{MAE} = \frac{1}{n}\sum_{j=i}^{n}|y_j - \hat{y_j}| \tag{4}$$

It should be noted that minimizing the MAE criterion is equivalent to minimizing 0.5 quantile loss.

The mean squared error (MSE) and root mean squared error (RMSE) are quadratic scoring methods. This means that they place higher weight on large errors. Models that minimize this metric are therefore more sensitive to outliers.

$$\text{MSE} = \frac{1}{n}\sum_{j=i}^{n}(y_j - \hat{y_j})^2 \tag{5}$$

### *2.3. Linear Models*

The least complex model considered is the simple linear regression model. The simple linear model assumes that the underlying conditional expectation $g^*(z_{i,t})$ can be modelled as a linear function of the predictors and the parameter vector $\theta$:

$$g(z_{i,t};\theta) = z'_{i,t}\theta \tag{6}$$

Computing this model with respect to minimizing the mean squared error yields the pooled ordinary least squares estimator (POLS), while minimizing the mean absolute error corresponds to quantile regression predicting the 0.5th quantile, also known as the Least Absolute Deviation (LAD) estimator.

[1] Also referred to as $l_1$ and $l_2$ loss respectively within the machine learning literature

The OLS estimator is known to be consistent when the regressors are exogeneous, and is optimal in the class of linear unbiased etimators when the errors are homoscedastic and serially uncorrelated. This unbiasedness consequently means that linear models have a large amount of variance due to the bias-variance trade-off, which can lead to poorer predictive ability as the estimator's variance will be very high and therefore inefficient. Additionally, OLS struggles when multicollinearity (also known as cross sectional correlation in empirical finance settings) is present, as the coefficients of different variables can behave erratically to small changes in the data or model. This impreciseness in the coefficient estimates further leads to poor predictive performance. The LAD estimator attempts to improve upon the OLS estimator through the use of loss function that is more robust to outliers. The mean absolute error weights all residuals equally, compared to mean squared error which through considering squared residuals gives more emphasis to large residuals.

Linear models can capture non-linearities only if the predictor set $z^*_{i,t}$ contains specified non-linear transformations or interaction terms. Despite being expected to perform poorly, linear models were implemented as a "control."

### *2.3.1. Penalized Linear Models*

Penalized linear models attempt to improve upon linear models through regularization, and are particularly well suited to addressing problems of multicollinearity, which can commonly occur in models with a large number of regressors. They achieve this by lowering the variance of their estimates (thus increasing efficiency) in exchange for introducing a tolerable amount of bias. This can be viewed as the model mechanically decreasing its in sample performance, in hopes that the model will overfit less, particularly to noise in the dataset, thus correctly preserving its fit to true underlying regressors.

Penalized linear models have the same underlying statistical model as simple linear models, but differ in their addition of a new penalty term in the loss function:

$$\mathcal{L}(\theta;.) = \underbrace{\mathcal{L}(\theta)}_{\text{Loss Function}} + \underbrace{\phi(\theta;.)}_{\text{Penalty Term}} \tag{7}$$

Several choices exist for the choice of penalty function $\phi(\theta;.)$. We restrict our scope to the popular "elastic net" penalty (Zou and Hastie, 2005):

$$\phi(\theta;\lambda,\rho) = \lambda(1-\rho)\sum_{j=1}^{P}|\theta_j| + \frac{1}{2}\lambda\rho\sum_{j=1}^{P}\theta_j^2 \tag{8}$$

The elastic net has two hyperparameters: $\lambda$, which controls the overall magnitude of the loss, and $\rho$, which controls the shape of the penalization.

The $\rho = 1$ case corresponds to ridge regression proposed by Hoerl and Kennard (1970), which uses $l_2$ penalty that shrinks all coefficients closer to 0, but not to 0. Ridge regression is therefore a shrinkage method which prevents coefficients from becoming too large and overpowering.

The $\rho = 0$ case corresponds to the popular LASSO and uses absolute ($l_1$) parameter penalization proposed by Tibshirani (1996), which geometrically allows the coefficients to be shrunk to 0. This allows it to impose sparsity, and can be thought of as a variable selection tool.

For $0 < \rho < 1$, the elastic net aims to produce parsimonious models through both shrinkage and selection by combining the properties of LASSO and ridge regression.

The hyperparameters $\lambda$ and $\rho$ are both tuned using the validation sample (see 6.2.2).

### 2.3.2. *Classification and Regression Trees*

Classification and regression trees are fully non-parametric models that can capture complex multi-way interactions. A tree "grows" in a series of iterations. With each iteration, a split ("branch") is made along one predictor such that it is the best split available at that stage with respect to minimizing the loss function. These steps are continued until each observation is its own node, or more commonly until the stopping criterion is met. The eventual model slices the predictor space into rectangular partitions, and predicts the unknown function $g^*(z_{i,t})$ with the average value of the outcome variable in each partition.

The prediction of a tree, $\mathcal{T}$, with $K$ "leaves" (terminal nodes), and depth $L$ is

$$g(z_{i,t};\theta,K,L) = \sum_{k=1}^{K} \theta_k \mathbf{1}_{z_{i,t}\in C_k(L)} \tag{9}$$

where $C_k(L)$ is one of the $K$ partitions in the model.

For this study, only recursive binary trees (the most common and easy to implement) are considered. Though trees were originally proposed and fit with respect to minimizing mean squared error, they can be grown with respect to a variety of loss functions, including mean absolute error, mean squared error, where the loss within each C partition is denoted by $H(\theta, C)$:

$$H(\theta,C) = \frac{1}{|C|} \sum_{z_{i,t}\in C} L(r_{i,t+1} - \theta) \tag{10}$$

where $|C|$ denotes the number of observations in set C (partition). Given $C$, it is clear that the optimal choice for minimising the loss function when it is mean squared error is simply $\theta = \frac{1}{|C|}\sum_{z_{io,t}\in C} r_{i,t+1}$ i.e. the average of the partition, and the median of the partition when the loss function is mean absolute error.

Trees, grown to a deep enough level, are highly unbiased and flexible, as each partition can potentially predict a single, or low number of observations. The trade-off is their high variance and instability. Thus, an ensemble method called "Random Forest" was proposed by Breiman (2001) to regularize trees by combining many different trees into a single prediction.

### 2.3.3. *Random Forests*

Random Forests are an extension of regression trees that attempt to address some of their problems, proposed by Breiman (2001). A random forest algorithm creates $B$ different bootstrap samples from the training dataset, fits an overfit (and hence low bias) regression tree to each dataset using only a random subset $m$ size from all available predictors (also known as dropout), and then averages their forecasts as the final output. The dropout procedure in particular ensures that trees will be unable to use the same predictors when considering splits, lowering the correlation between each tree and thus further reducing the variance across the ensemble model. Thus, the ensemble consisting of only overfit (and hence low bias) which have low uncorrelation is a low bias, yet stable model. Specific details of the random forest algorithm are detailed in the appendix.

### 2.3.4. *Neural Networks*

Neural networks have theoretical underpinnings as "universal approximators" for any function, (Hornik et al. (1989)). They are arguably the most complex type of model available, able to capture several non-linear interactions through their many layers, hence its other name "deep learning." On the flipside,

their high flexibility often means that they are among the most parameterized and least interpretable models, earning them the reputation as a black box model.

Per this analysis, we focus on traditional "feed-forward" networks. The feed forward network consists of an "input layer" of scaled data inputs, one or more "hidden layers" which interact and non-linearly transform the inputs, and finally an output layer that aggregates the hidden layers and transform them a final time for the final output.

More specifically, a neural network consists of layers denoted by $l = 0, 1, \ldots, L$, with $l = 0$ denoting the input layer and $l = L$ denoting the output layer. The input layer is defined by the scaled predictor set, $x^{(0)} = (1, z_1, \ldots, z_N)'$. The model adds complexity through the use of one or more hidden layer, each containing $K^{(l)}$ "neurons". Each neuron linearly aggregates the values of the previous layer, and applies some non-linear "activation function" which we denote as $\alpha$ to its aggregated signal before sending its output to the next layer. The output of neuron $k$ in layer $l$ is then $x_k^{(l)}$. Next, define the vector of outputs for this layer as $x^{(l)} = (1, x_1^{(l)}, \ldots, x_{K^(l)}^{(l)})'$. The recursive output formula for the neural network at each neuron in layer $l > 0$ is then:

$$x_k^{(l)} = \alpha(x^{(l-1)'}\theta_k^{l-1}), \tag{11}$$

where $\alpha()$ represents the activation function for that layer with the final output [2]

$$g(z;\theta) = x^{(L-1)'}\theta^{L-1} \tag{12}$$

The neural network's weight and bias parameters for each layer are estimated by minimizing the loss function with respect to the parameters, i.e. by calculating the partial derivative with respect to a specific weight or bias element.

Due to the complexity and hence non-existent analytical form for this solution, this is typically found via backpropagation, an algorithm which exploits the chain rule of the partial derivative and iteratively finds a local optimum using a first order gradient based algorithm, also known as "gradient descent." The gradient descent algorithm minimizes some function (such as the loss function in the context of machine learning) by iteratively moving in the direction of steepest descent, defined as the negative gradient. Formally, for a loss function $L(x)$ that is defined and has a gradient defined in the neighbourhood of the parameter set $a$, the updating algorithm is:

$$a_{n+1} = a_n - \gamma \Delta F(a_n) \tag{13}$$

where $\gamma$ controls the size of each update. This $\gamma$ parameter is known as the learning rate in neural network training, and controlling this is critical for good performance. As the loss functions of neural networks can be very complex with many local minima, the learning rate should be high enough such that the parameter updates are large enough to skip or jump over them. Too large of a learning rate however, and the neural may fail to converge to a solution at all. Due to computational limitations, we tune the learning rate manually, and employ the ADAM algorithm to apply allow different learning rates for each parameter.

Stochastic gradient descent is a variation of traditional gradient descent which assists with computational feasibility and producing solutions which generalize better, and hence better performance. Instead of optimizing the weight parameters with respect to the entire training sample, a small random subset is instead used, the size of which is called the batch size. This has the effect of potentially increasing

[2] Note that the specification of a constant "1" at the beginning of each layer is the same as specifying a bias term as is popular in other parametrizations.

computational feasibility due to smaller memory requirements. Less understood, but empirically evident is that smaller batch sizes tend to produce better solutions, (Keskar et al. (2016)). It is therefore a critical hyperparameter to tune.

We also employ $l_1$ penalty to impose some sparse restrictions on the weight terms, aiding with the production of better generalization. This simply adds the absolute value of each weight parameter, multiplied by a scaling factor which is a hyperparameter to be tuned to the overall loss function.

"Batch normalization" is a technique for addressing a phenomenon known as internal covariate shift, which occurs when the distributions of each layers' inputs change as the parameters of the previous layer change (Ioffe and Szegedy, 2015). This occurs due to how neural networks repeatedly apply the activation function many times over different hidden layers. For activation functions such as tanh, this results in weight parameters getting pushed and thus "saturated" towards -1 and 1, leading to difficult and slow training. Batch normalization addresses this by normalizing (de-meaning and variance standardizing) the outputs of each layer, hence restoring the representative power of each neuron.

**Model Architecture and Specification** Neural networks with up to 5 hidden layers were considered, each named NNX where X represents the number of hidden layers. The number of neurons is each layer was chosen according to the geometric pyramid rule (Masters, 1993): NN1 has 32 neurons, NN2 has 32 and 16 neurons in the first and second hidden layers respectively, NN3 has 32, 16, and 8 neurons, NN4 has 32, 16, 8, and 4 neurons, and NN5 has 32, 16, 8, 4, 2 neurons respectively. All units are fully connected; that is, each neurons receives input from all neurons the layer before it (see Figure 1). This mimics the methodology in Gu et al. (2018).

Several choices of activation functions exist in the literature. We use the hyperbolic tangent function:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{14}$$

for all hidden layers due to its observed performance, and ability to retain weight terms and hence always receive weight updates during training. This point is particularly important, because the popular ReLU activation function (see see Lecun et al. (2015) and Ramachandran et al. (2017), among others):

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

was considered, but observed to suffer tremendously from the "dying-ReLU problem". This is where ReLU neurons receive weight updates that fail to activate (output a 0), hence making it unable to receive further weight updates and learn. This resulted in networks which outputted the same value for the majority of, if not all inputs. Variants of the ReLU activation function aimed at addressing this issue such as the leaky ReLU, which allows a small, non-zero gradient for negative weights were explored, but still suffered from poor convergence.

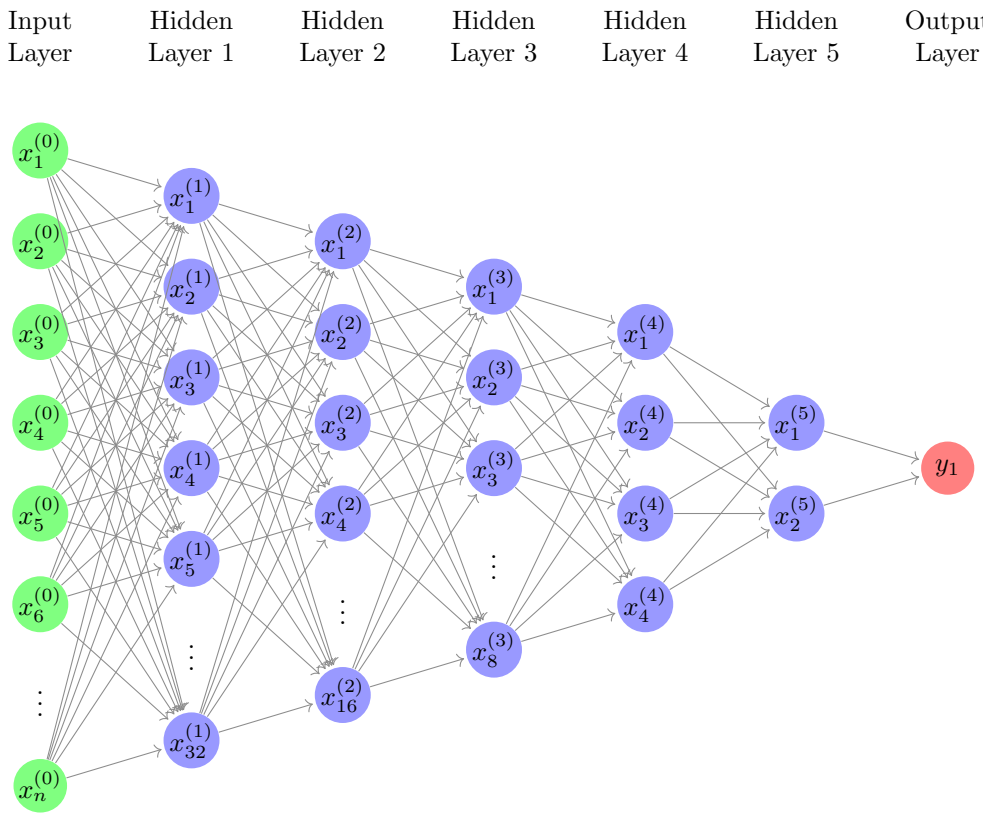



Figure 1. Neural Network 5 (most complex considered), without biases terms drawn

### *2.3.5. Model Evaluation*

**Loss Metrics** Overall predictive performance for individual excess stock returns were assessed using the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE). An out of sample $R^2$ metric was also reported, as is popular in the literature, which we define as:

$$R^2_{OOS} = 1 - \frac{\sum_{(i,t)\in\mathcal{T}_3}(r_{i,t+1} - \widehat{r}_{i,t+1})^2}{\sum_{(i,t)\in\mathcal{T}_3}(r_{i,t+1} - \bar{r}_{i,t+1})^2} \tag{16}$$

where $\mathcal{T}_3$ indicates that the fits are only assessed on the test subsample, which is never used for training or tuning. However, as $R^2$ measures were originally formulated for assessing in sample fit for linear models, the interpretation of this metric is less meaningful in the contexts of forecasting out of sample and non linear models.

**Variable Importance** The importance of each predictor $j$ is denoted as $VI_j$, and is defined as the reduction in predictive R-Squared from setting all values of predictor $j$ to 0, while holding the remaining model estimates fixed. These were then normalized to sum to 1 within each model as a way to assess the relative importance of each predictor for each model. As $VI_j$ can sometimes be negative, or in some cases be 0 across most of the factors, $VI_j$ positively shifted by the magnitude of the smallest $VI_j$ plus a minor offset, then dividing all $VI_j$ by the total:

$$VI_{j,norm} = \frac{VI_j + \min(VI_j) + 1e^{-100}}{\Sigma VI_j + \min(VI_j) + 1e^{-100}} \tag{17}$$

This mechanism was chosen because the other popular normalization mechanism "softmax" was

observed to be unable to preserve the distances between each original $VI_j$ well enough, squishing together all variable importance measures and making discernment between each variable difficult.

# 3. Simulation Study

## *3.1. Simulation Design*

### *3.1.1. Overall Design*

Though Gu et al. (2018) explore the performance of machine learning on simulated returns series, their specification used factors are uncorrelated across $i$, and, in particular, that the factors which enter the return equation are uncorrelated with the factors that do not enter the return equation. As noted by Harvey et al. (2016) and many others, this is not what is observed in practice. Furthermore, their error specification is a white noise process consisting of a gaussian term and a student t error term. Added on to a weak underlying signal representing the predictable portion of the returns process, this amounts to a constant mean white noise process, which fails to capture characteristics of empirical returns, such as time varying volatility and volatility clustering.

Therefore, we simulate a latent factor model with a stochastic volatility process for excess returns $r_{t+1}$, for $t = 1, \ldots, T$:

$$r_{i,t+1} = g\left(z_{i,t}\right) + \beta_{i,t+1} v_{t+1} + e_{i,t+1}; \quad z_{i,t} = (1, x_t)' \otimes c_{i,t}, \quad \beta_{i,t} = (c_{i1,t}, c_{i2,t}, c_{i3,t}) \tag{18}$$

$$e_{i,t+1} = \sigma_{i,t+1} \varepsilon_{i,t+1}; \tag{19}$$

$$\log(\sigma^2_{i,t+1}) = \omega + \gamma \log(\sigma^2_t) + \sigma_u u; \quad u \sim N(0,1) \tag{20}$$

Let $v_{t+1}$ be a $3 \times 1$ vector of errors, and $w_{t+1} \sim N(0,1)$ and $\varepsilon_{i,t+1} \sim N(0,1)$ scalar error terms. The parameters of these are tuned such that the annualized volatility of each return series was approximately 22%, as is often observed empirically.

The matrix $C_t$ is an $N \times P_c$ vector of latent factors, where the first three columns correspond to $\beta_{i,t}$, across the $1 \leq i \leq N$ dimensions, while the remaining $P_c - 3$ factors do not enter the return equation. The $P_x \times 1$ vector $x_t$ is a $3 \times 1$ multivariate time series, and $\varepsilon_{t+1}$ is a $N \times 1$ vector of idiosyncratic errors.

### *3.1.2. Simulating Characteristics*

A simulation mechanism for $C_t$ that gives some correlation across the factors and across time was used. We build in correlation across time among factors by drawing normal random numbers for each $1 \leq i \leq N$ and $1 \leq j \leq P_c$, according to

$$\overline{c}_{ij,t} = \rho_j \overline{c}_{ij,t-1} + \epsilon_{ij,t}; \quad \rho_j \sim \mathcal{U}\left(\frac{1}{2}, 1\right) \tag{21}$$

Then, define the matrix $B$:

$$B := \Lambda\Lambda' + \frac{1}{10}\mathbb{I}_n, \quad \Lambda_i = (\lambda_{i1}, \ldots, \lambda_{i4}), \quad \lambda_{ik} \sim N(0, \lambda_{sd}),\ k = 1, \ldots, 4 \tag{22}$$

Note that $B$ is a positive-semidefinite matrix, and thus can serve as a variance covariance matrix with $\lambda_{sd}$ controlling the density of the matrix, and hence degree of cross sectional correlation. $\lambda_{sd}$ values of 0.01, 0.1 and 1 were used to explore increasing degrees of cross sectional correlation.

To build in cross-sectional correlation, from the $N \times P_c$ matrix $\bar{C}_t$, we simulate characteristics according to

$$\widehat{C}_t = L\overline{C}_t; \quad B = LL' \tag{23}$$

where $L$ represents the lower triangle matrix of $B$ using the Cholesky decomposition.

Finally, the "observed" characteristics for each $1 \leq i \leq N$ and for $j = 1, \ldots, P_c$ are constructed according to:

$$c_{ij,t} = \frac{2}{n+1} \operatorname{rank}\left(\hat{c}_{ij,t}\right) - 1. \tag{24}$$

with the rank transformation normalizing all predictors to be within $[-1, 1]$.

### *3.1.3. Simulating Macroeconomic Series*

For simulation of $x_t$, a $3 \times 1$ multivariate time series, we consider a Vector Autoregression (VAR) model [3], a generalization of the univariate autoregressive model to multiple time series:

$$x_t = Ax_{t-1} + u_t; \quad A = \begin{pmatrix} .95 & 0 & 0 \\ 0 & .95 & 0 \\ 0 & 0 & .95 \end{pmatrix} \quad u_t \sim N\left(\mu = (0,0,0)', \Sigma = I_3\right)$$

### *3.1.4. Simulating Return Series*

We consider three different function for $g(z_{i,t})$:

$$(1)\ g_1\left(z_{i,t}\right) = \left(c_{i1,t}, c_{i2,t}, c_{i3,t} \times x_t'[3,]\right)\theta_0 \tag{25}$$

$$(2)\ g_2\left(z_{i,t}\right) = \left(c_{i1,t}^2, c_{i1,t} \times c_{i2,t}, \operatorname{sgn}\left(c_{i3,t} \times x_t'[3,]\right)\right)\theta_0 \tag{26}$$

$$(3)\ g_3\left(z_{i,t}\right) = \left(1[c_{i3,t} > 0], c_{i2,t}^3, c_{i1,t} \times c_{i2,t} \times 1[c_{i3,t} > 0], \operatorname{logit}\left(c_{i3,t}\right)\right)\theta_0 \tag{27}$$

where $x_t'[3,]$ denotes the third element of the $x_t'$ vector.

$g_1\left(z_{i,t}\right)$ allows the characteristics to enter the return equation linearly, and $g_2\left(z_{i,t}\right)$ allows the characteristics to enter the return equation interactively and non-linearly. These two specifications correspond to the simulation design used by Gu et al. (2018).

$g_3\left(z_{i,t}\right)$ allows the characteristics to enter in a complex and non-linear fashion. In should be noted however, that because $g_2\left(z_{i,t}\right)$ has a large part of its signal entering through a sgn function, this should make it the most difficult to estimate given the regressors and resulting returns process.

$\theta^0$ was tuned such that the predictive $R^2$ was approximately 5%.

The simulation design results in $3 \times 3 = 12$ different simulated datasets, each with $N = 200$ stocks, $T = 180$ periods and $P_c = 100$ characteristics. Each design was simulated 10 times to assess the robustness of machine learning algorithms. The number of simulations was kept low for computational feasibility.

[3] Other more complex and interactive matrix specifications of $A$ were briefly explored, but these did not appear to have a significant impact on results. More complex designs were observed to only affect the variable importance metrics, but only to a small degree.

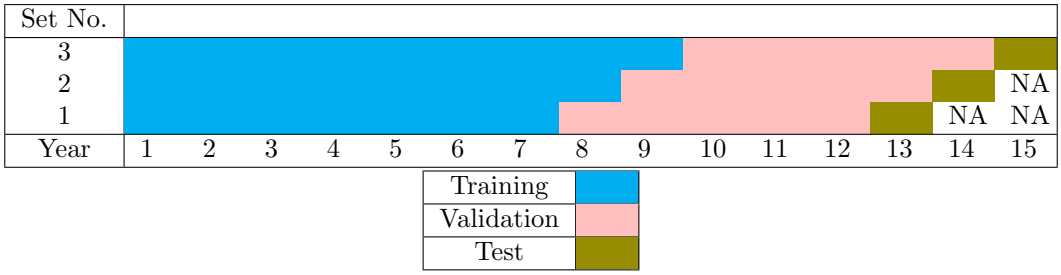



Figure 2. Sample Splitting Procedure

#### *3.1.5. Sample Splitting*

If viewed as monthly periods, $T = 180$ corresponds to 15 years. A data splitting scheme similar to to the scheme to be used in the empirical data study was used: a training:validation length ratio of approximately 1.5 to begin, and a test set that is 1 year in length. We employ the hybrid growing window approach as described earlier in section 2.2.1 (see Figure 2 for a graphical representation).

Other schemes in the forecasting literature such as using an "inner" rolling window validation loop to find the best hyperparameters on average, finally aggregating them in an "outer" loop for a robust error were considered but not implemented for a variety of reasons. Firstly, many of the models are computationally too intensive for this to be feasible for this to be feasible. More importantly, during the model fitting process it was observed that the optimal hyperparameters for the different rolling windows were highly unstable. Thus, this would have made the selection of the best hyperparameters on average across all windows significantly less meaningful.

### *3.2. Simulation Study Results*

In our simulation study, we find that the penalized linear models consistently perform the best, followed extremely closely by random forests.

We also find that models fitted with respect to minimizing mean absolute error (quantile loss) generally perform better, even in terms of the mean squared error metrics. This is not a surprising result, especially considering the stochastic error design which introduces significant shocks to the returns process, leading to large outliers which the mean squared error metric is more sensitive to.

In terms of finding the correct underlying causal regressors, we find that the penalized linear models perform the best at identifying the true data generating regressors, and that this appears to be mostly robust regardless of the amount of cross sectional correlation. The penalized linear models are not perfect, particularly in the $g_2$ specification. On these more difficult specifications, the penalized linear models are generally very conservative, sometimes only identifying a single covariate as important. The penalized linear models also occasionally identified the incorrect covariates completely, though the relative importance assigned to them was small.

Of particular note the instability of the penalized linear models' hyperparameters across different training samples. Though the optimal value for $\alpha$ is generally 1 (corresponding to LASSO and thus a sparse representation), it was not uncommon to see $\alpha$ values swinging between values close to 0 (corresponding to ridge regression, and thus a dense representation) to 1 as the training sample moved forwards in time. As the penalized linear models consistently performed the best and still remained able to correctly identify the true covariates this is not a large issue, but it should be noted that this can lead to interpretation issues.

The random forest models were typically able to correctly identify the true data generating covariates, but struggled with discerning them adequately from other covariates. This is to be expected, likely due to how the random forest algorithms work. The random forest algorithm is an ensemble of tree models, with each tree model only having access to a subset of all available predictors. If this subset does not include the true data generating predictor, then that particular tree will likely select the predictors which have the highest correlation with the true data generating predictor instead. Thus, the resulting ensemble model is likely to believe that cross sectionally correlated predictors are important, relative to the true causal regressor.

The linear models unsurprisingly performed the worst, with both prediction performance and correct variable importance deteriorating noticeably as the degree of cross sectional correlation increased.

**RS:** neural networks still running on simulations at the moment unfortunately

The neural networks were observed to have poor performance when compared to other machine learning models.

It was also noted that across all models, the prediction performance worsened as the training sample size increased on the simulations with stochastic volatility. This is likely due to a larger training sample being more probable to include large outliers due to the stochastic volatility process, which can adversely affect prediction performance.

Overall, it appears that the penalized linear models offered the best performance, both in terms of actual prediction accuracy and identifying true regressors, even when the predictor set contained high degrees of cross sectional correlation and non-linearities. They still, however, do not exhibit perfect performance and can struggle with identifying true regressors, particularly in non-linear settings.

# 4. Empirical Study

## *4.1. Data*

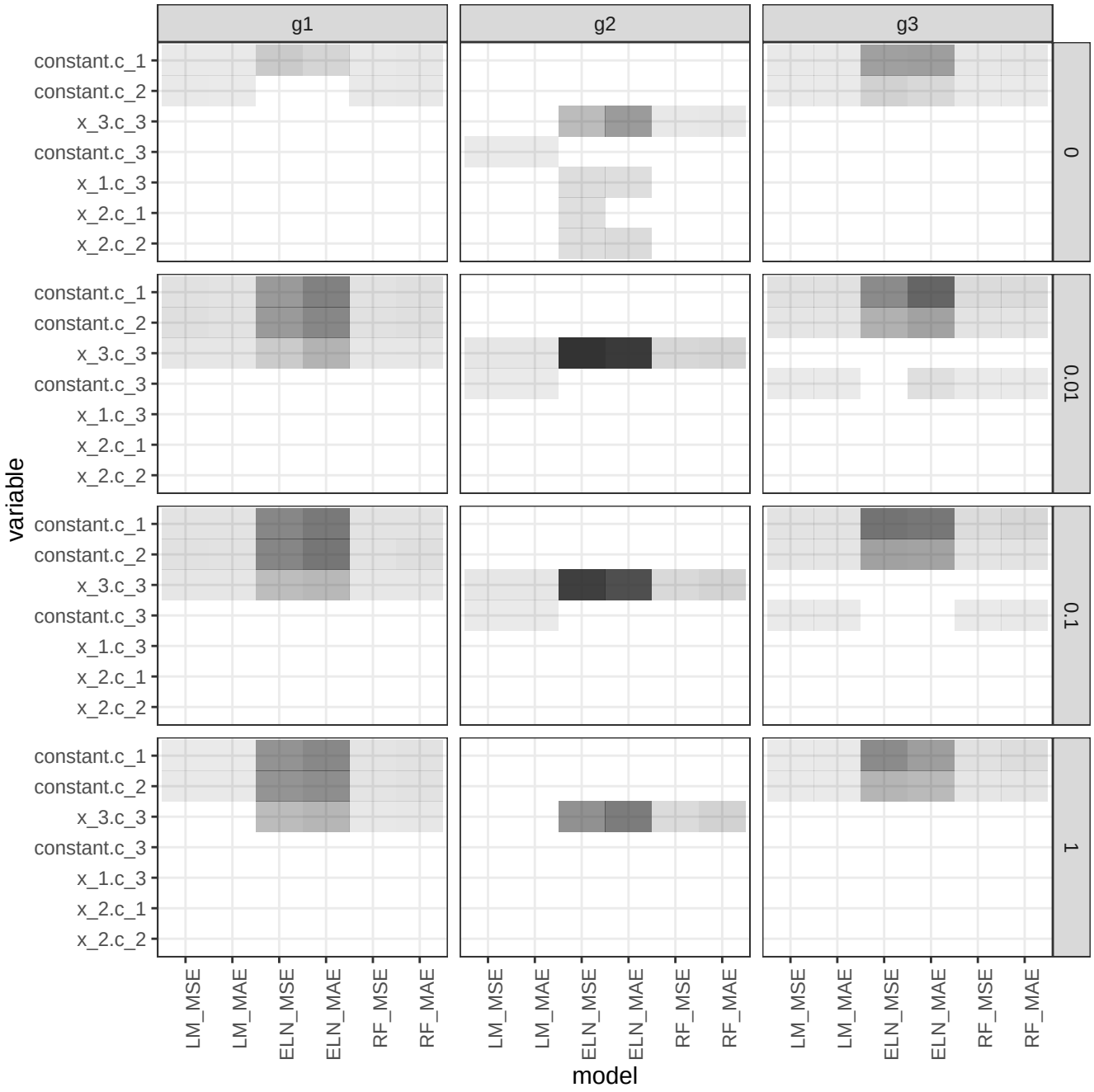
We mimic the data procedure of Gu et al. (2018). This means that we obtain the dataset provided by Gu on his website. This dataset sample begins in March 1957 (the start date of the S&P 500) and ends in December 2016, totalling 60 years. It contains 94 stock level characteristics: 61 updated annually, 13 updated quarterly and 20 updated monthly, in addition to 74 industry dummies corresponding the the first two digits of the Standard Industrial Classification (SIC) codes. It is noted that this dataset contains all securities traded, including those with a CRSP share code other than 10 or 11 and thus includes instruments such as REITs and mutual funds, and those with a share price of less than $5.

We detail our cleaning procedure of this dataset. To reduce the size of the dataset and increase feasibility, the dataset was filtered so that only stocks traded primarily on NASDAQ were included (using the PRIMEXCH variable from WRDS). Then, penny stocks (also referred to as microcaps in the literature) with a stock price of less than $5 were filtered out, as is commonly done in the literature to reduce variability. Stocks without a share code of 10 or 11 (referring to equities) were filtered out, so that securities that are not equities were not included (such as REITs and trust funds). The dataset is provided in a monthly format, which means that many of the factors which are updated only quarterly or annually have very low levels of variability, which can lead to misleading results in the model fitting process. To achieve a balance between having a dataset with enough data points and variability among factors, the dataset was converted to a quarterly format. Quarterly returns were then constructed using the PRC variable according to actual returns (ie not logged differences):

$$RET_t = \frac{PRC_t - PRC_{t-1}}{PRC_{t-1}} \tag{28}$$

Table 1: Simulation Results

Figure 3. Simulation Variable Importance Plots



We allow all stocks which have a quarterly return to enter the dataset, even if they disappear from the dataset for certain periods, as opposed to only keeping stocks which appear continuously throughout the entire period. This was primarily done to reduce survivorship bias in the dataset, which can be very prevalent in financial data, and also allows for stocks which were unlisted and relisted again to feature in the dataset. This has the obvious drawback of introducing some bias in the dataset, as attrition in the dataset is likely to be non-random and correlated with the stocks' returns.

The sic2 variable, corresponding to the stocks' Standard Industrial Classification (SIC) codes was also dropped. The SIC code system suffers from inconsistent logic in classifying companies, and as a system built for pre-1970s traditional industries has been slow in recognizing new and emerging industries. Indeed, WRDS explicitly cautions the use of SIC codes beyond the use of rough grouping of industries, warning that SIC codes are not strictly enforced by government agencies for accuracy, in addition to most large companies belonging to multiple SIC codes over time. Because of this latter point in particular, there can be inconsistencies on the correct SIC code for the same company depending on the data source. Dropping the sic2 variable also reduced the dimensionality of the dataset by 74 columns, significant increasing computational feasibility.

There existed a significant amount of missing data in the dataset. The dataset's columns were first examined, and any characteristics that had over 20% of their data were removed. However, as the amount of missing data increases dramatically going further back in time, a balance between using more periods at the cost of removing more characteristics versus using less periods but keeping more characteristics was needed. 1993 Q3 was determined to be a reasonable time frame to begin the dataset, as was a noticeable increase in data availability after this time. Missing characteristics were then imputed using their cross sectional medians for each year.

We then follow Gu et al. (2018) and construct eight macroeconomic factors following the variable definitions in Welch and Goyal (2008): dividend-price ratio (dp), earnings-price ratio (ep), book-to-market ratio (bm), net equity expansion (ntis), Treasury-bill rate (tbl), term spread (tms), default spread (dfy) and stock variance (svar). These factors were lagged by one period so as to be used to predict one period ahead quarterly returns. The treasury bill rate was also used from this source to proxy for the risk free rate in order to construct excess quarterly returns.

The two sets of factors were then combined to form a baseline set of covariates, which we define throughout all methods and analysis as:

$$z_{i,t} = (1, x_t)' \otimes c_{i,t} \tag{29}$$

where $c_{i,t}$ is a $P_c$ matrix of characteristics for each stock $i$, and $(1, x_t)'$ is a $P_x \times 1$ vector of macroeconomic predictors. $z_{i,t}$ is therefore a $P_x P_c$ vector of features for predicting individual stock returns and includes interactions between stock level characteristics and macroeconomic variables. The total number of covariates in this baseline set is $61 \times (8 + 1) = 549$[4].

The dataset was not normalized for all methods, as only penalized regression and neural networks are sensitive to normalization. For these two methods, the dataset was normalized such that each predictor column had 0 mean and 1 variance.

The final dataset spanned from 1993 Q3 to 2016 Q4 with 202066 individual observations.

We mimic the procedure used in the simulation study. This means that the dataset was split such that the training and validation sets were split such that the training set was approximately 1.5 times the length of the validation set, in order to predict a test set that is one year in length.

[4]As the individual and macroeconomic factors can have similar names, individual and macroeconomic factors were prefixed with ind_ and macro_ respectively.

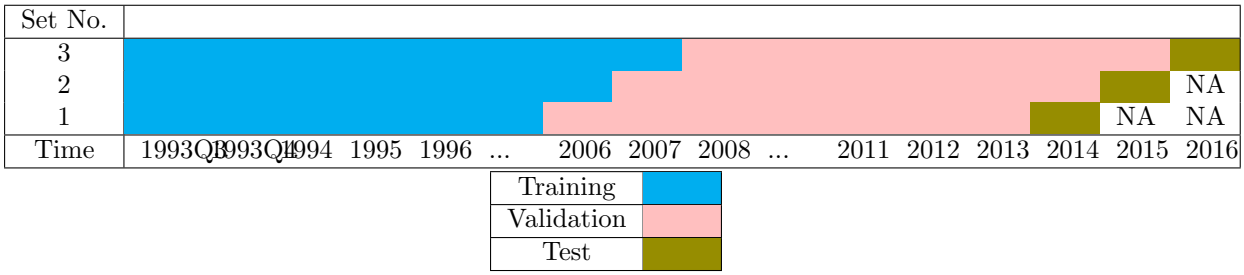



Figure 4. Empirical Data Sample Splitting Procedure

## *4.2. Empirical Data Results*

### *4.2.1. Prediction Accuracy*

In general, results from the simulation study were reproduced in the empirical study. We similarly see that the penalized linear models generally performing the best, with the random forest models offering slightly worse performance, occasionally outperforming penalized linear models. Models fitted with respect to median quantile loss were similarly observed to typically offer improvements across all machine learning models across all loss metrics.

In the empirical data setting, the underlying data generating process of returns is unknown so the ability of the different models at picking out the correct covariates cannot be evaluated. However, we can see that the penalized linear models and random forests (the models with the highest out of sample prediction performance) decide on a similar subset of factors which are determined to be important. In particular, we find that the 1 month momentum and 6 month momentum factors to be the most important factors by far, and this is consistent across all training samples. The conservative sparsity of the penalized linear models seen in the simulated setting is not repeated here; almost all factors are assigned to have small importance. The random forests again seem to struggle with discerning the importance of correlated factors apart: many of the important factors identified are highly correlated with momentum, such as change in momentum, dollar trading volume and return volatility.

The linear models had the worst performance of all the models considered. Interestingly, we find that the linear models assign the controversial dividend price ratio macroeconomic factor as highly important. Their variable importance for individual factors across different training samples is highly non-robust, with the important variables almost completely changing year to year.

The neural networks similarly had non-robust results. There is weak evidence to suggest that more complex neural networks with more layers give better performance. The neural networks were also the only models to assign high variable importance to the individual market value factor, a result that is somewhat consistent across all neural networks considered.

All models considered typically preferred sparse parameterizations. That is, most if not all of the individual factors had little to no importance across all models.

When looking at the macroeconomic factors (see Figure 11), there is less consistency on the variable importance between models. Penalized linear models tend to identify the average book to market ratio and the default spread as the most important macroeconomic factors, a result weakly consistent with those of the random forests. The linear models consistently identified the controversial dividend-price ratio as important, a result that was somewhat consistent with the neural networks. Some of the neural networks also occasionally identified the earnings price ratio as important.

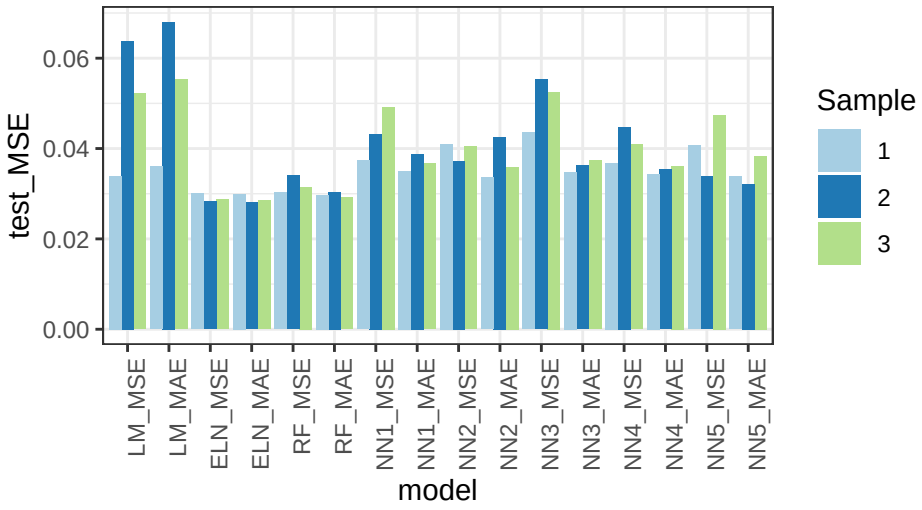



Figure 5. Empirical Test MSE Averaged Across All Samples

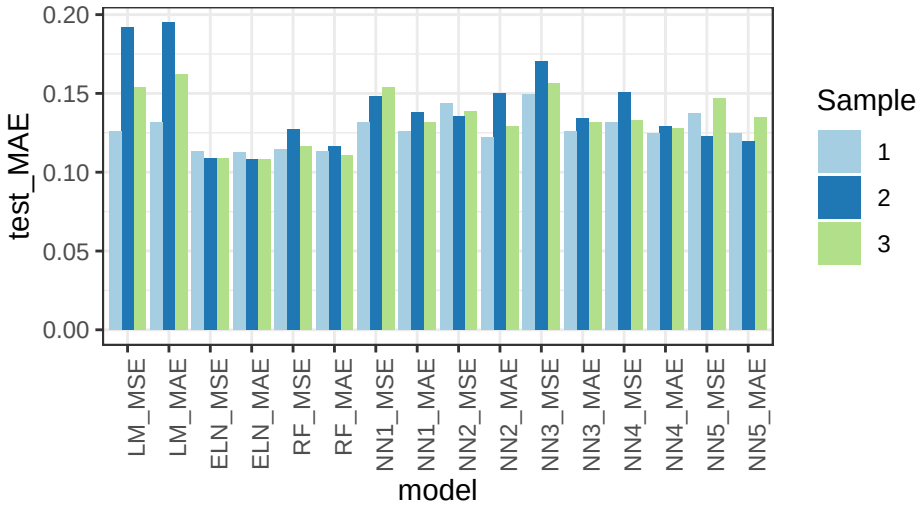



Figure 6. Empirical Test MAE Averaged Across All Samples

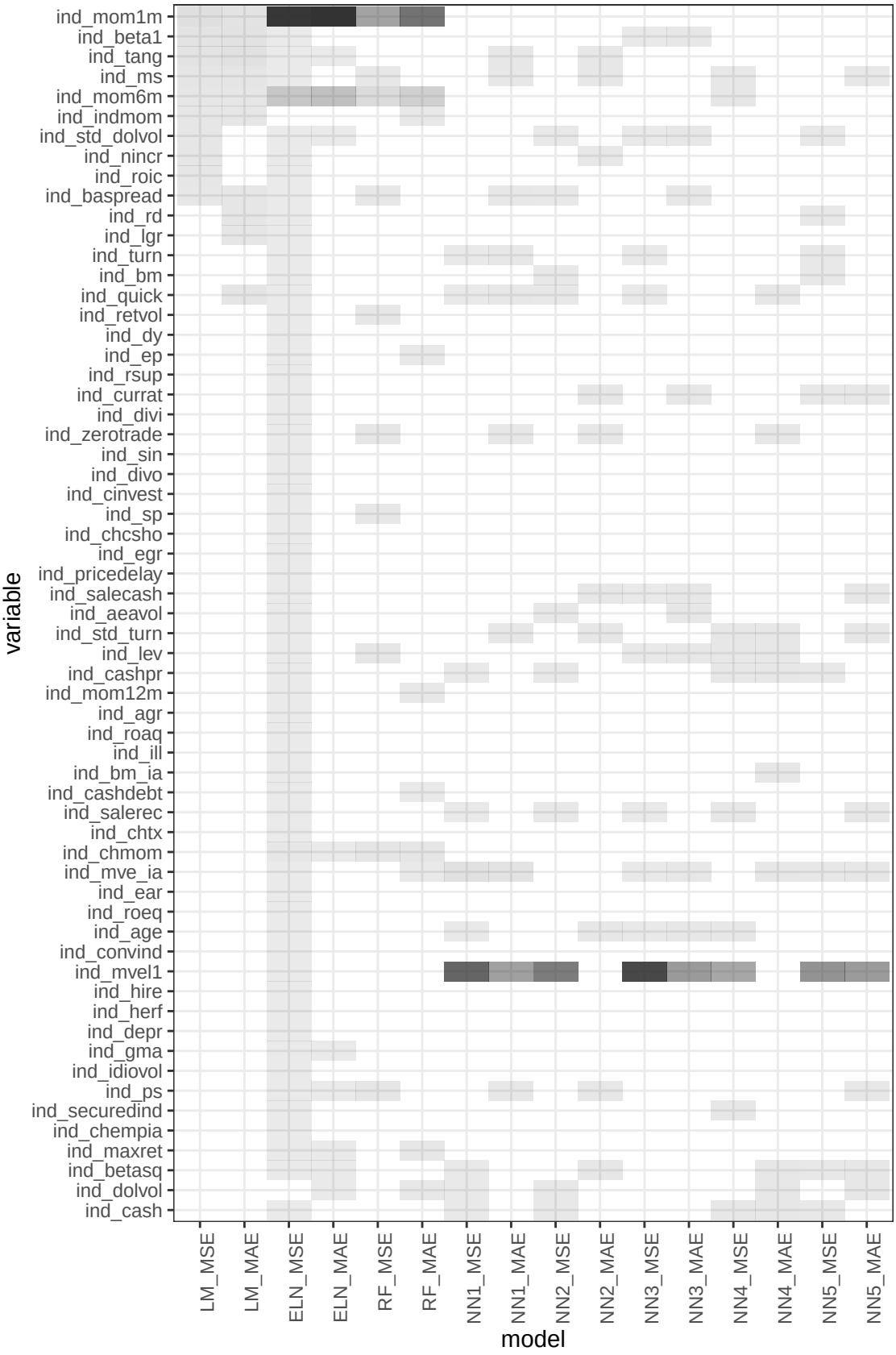



Figure 7. Empirical Data Individual Factor Variable Importance Averaged Across All Samples

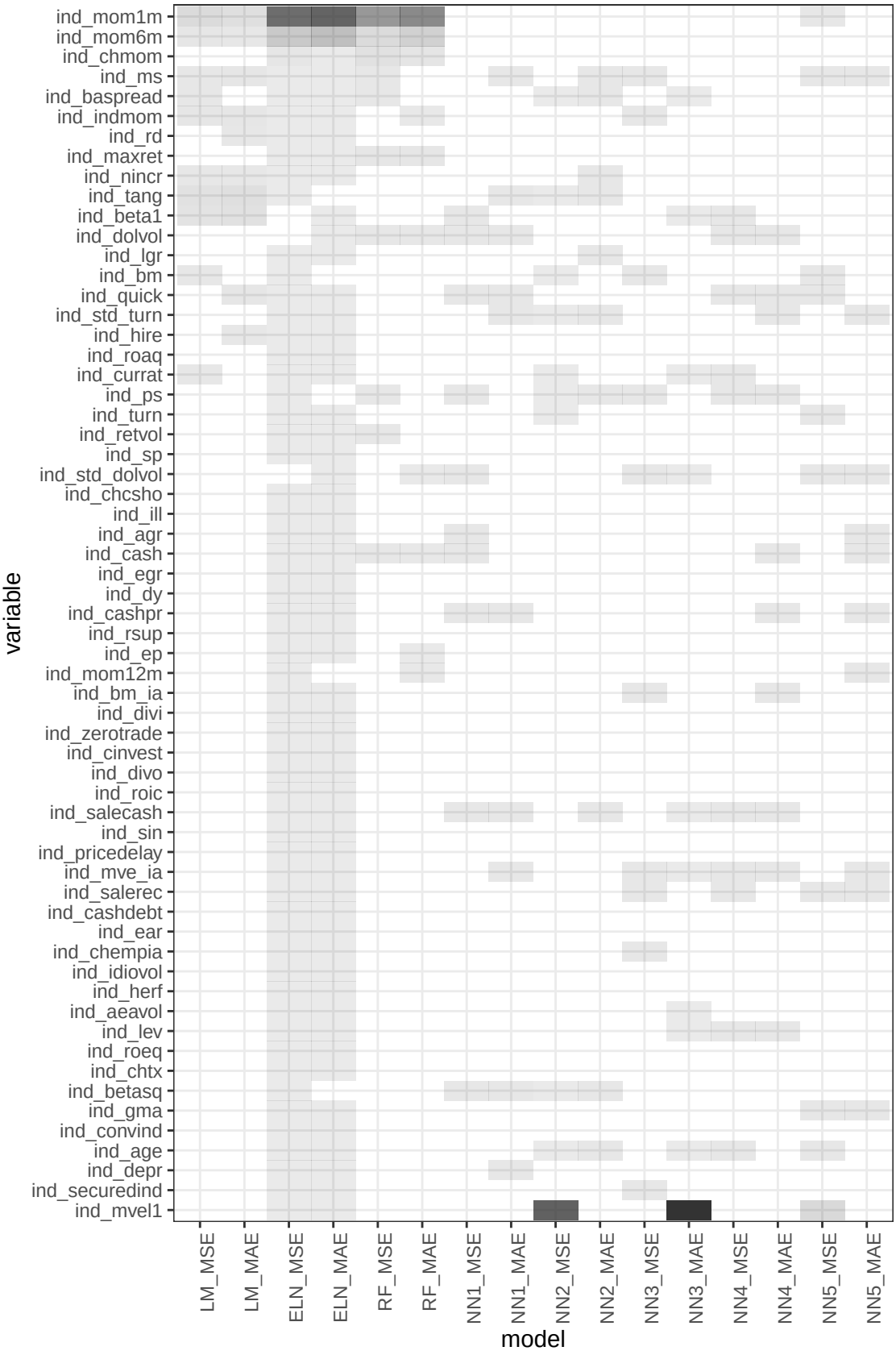



Figure 8. Empirical Data Individual Factor Variable Importance for Sample 1

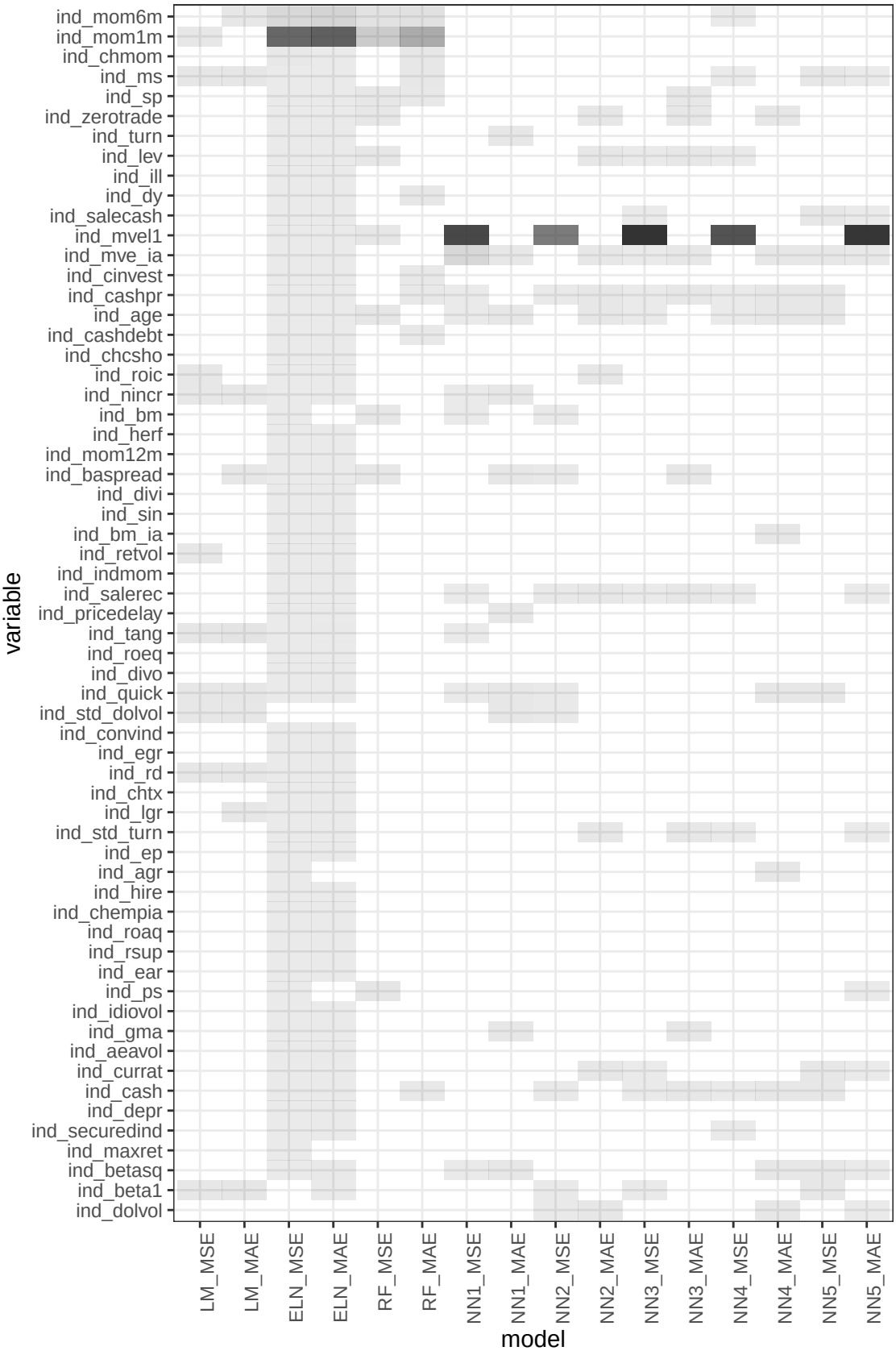



Figure 9. Empirical Data Individual Factor Variable Importance for Sample 2

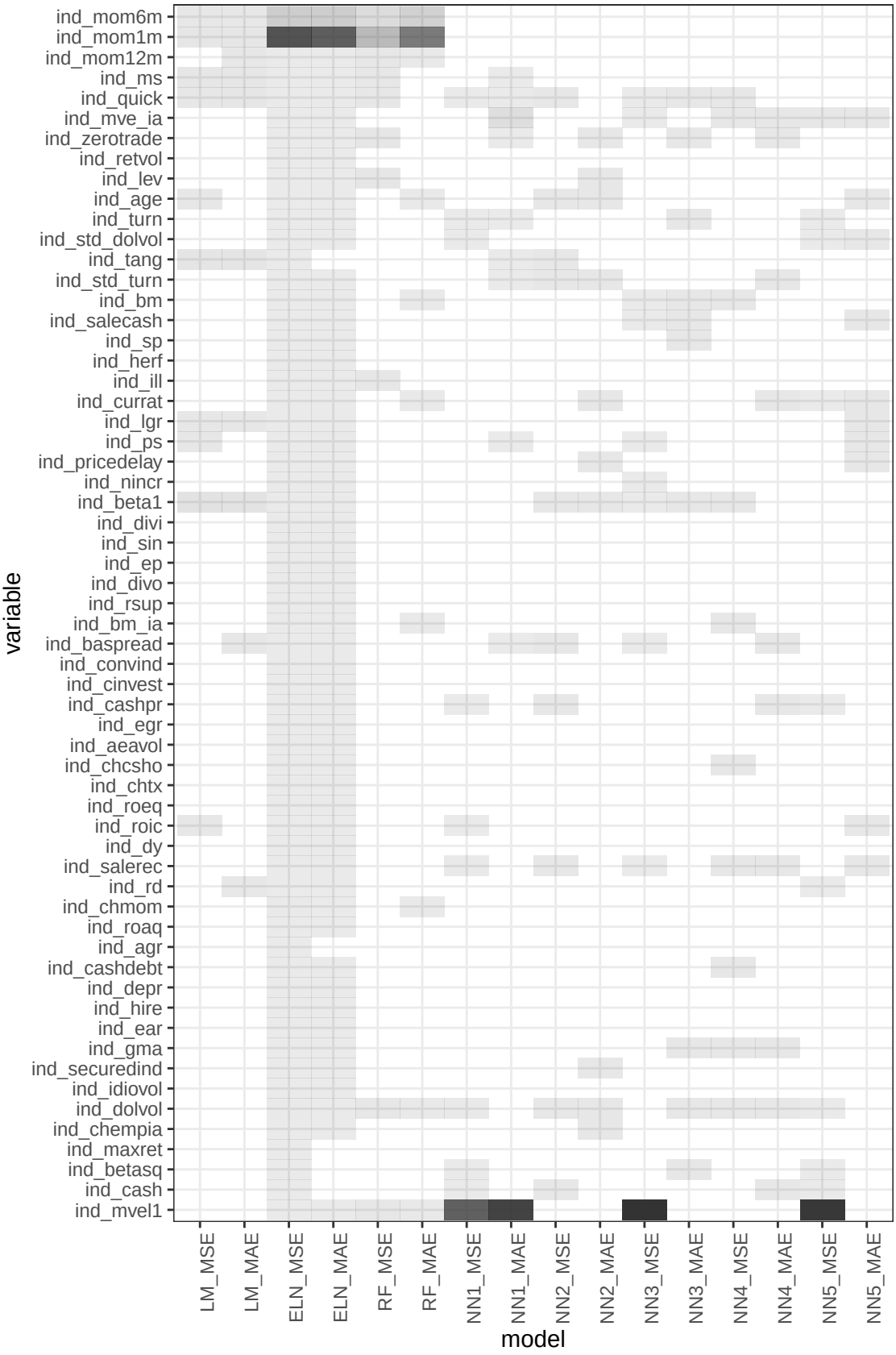



Figure 10. Empirical Data Individual Factor Variable Importance for Sample 3

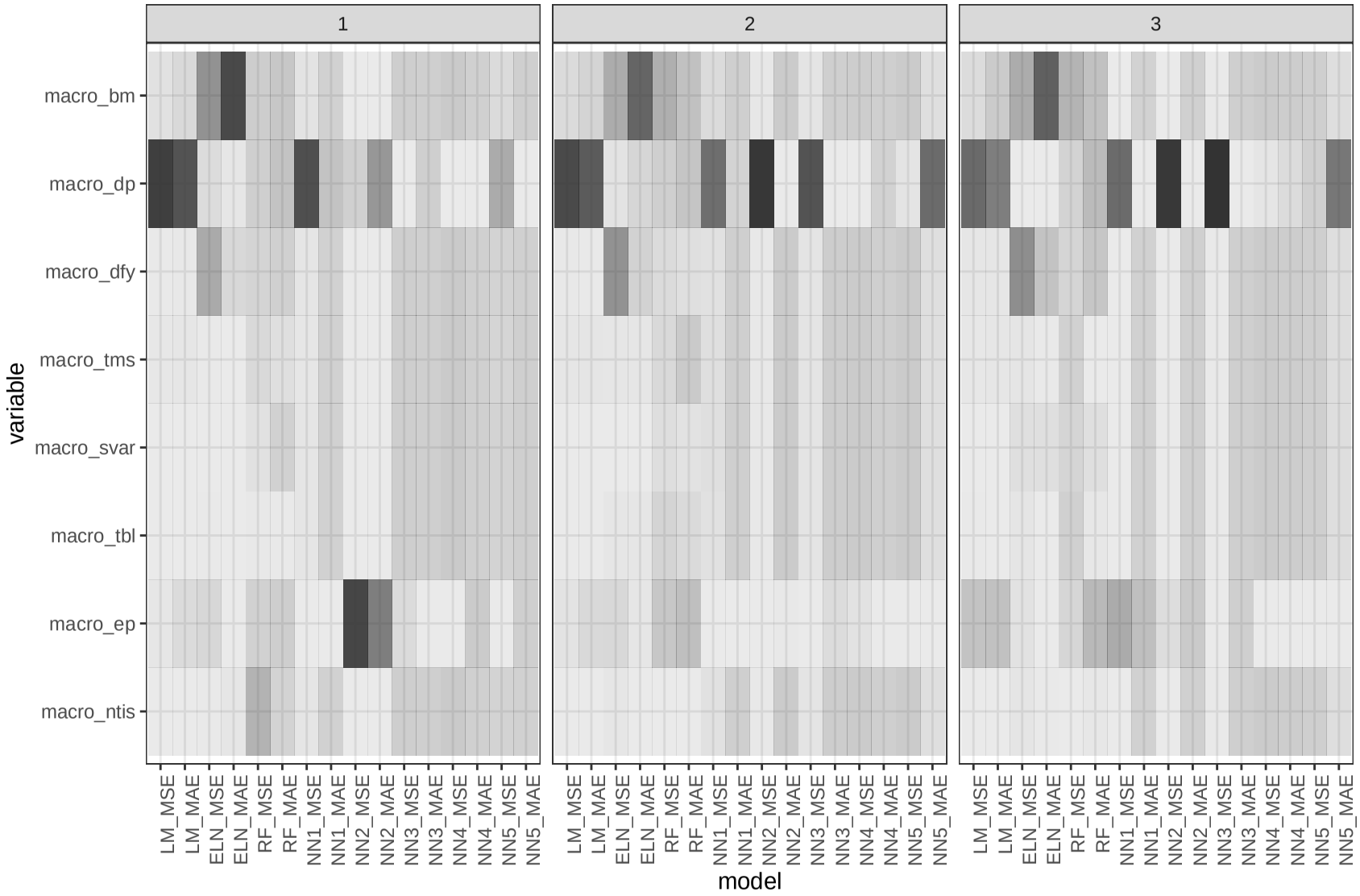



Figure 11. Empirical Data Macroeconomic Factor variable Importance Averaged Across All Samples, faceted by training sample

# 5. Conclusion

Our findings demonstrate that within the contexts considered, the field of machine learning offers tools to improve stock prediction and identification of true underlying regressors. Penalized linear models and to a lesser extent, random forests are the best performing methods.

Importantly, we find that the feed-forward neural network architectures considered fail in the context of stock return prediction, at both prediction performance and variable importance analysis. This is a result consistent across a variety of simulated datasets, as well as empirical data. We also find weak evidence that deeper neural networks with more hidden layers perform better, though this result, like the performance of neural networks in general, is non-robust.

Lastly, we find that the top performing models - the penalized linear models and random forests, tend to agree and correctly identify the correct causal regressors in simulated contexts, as well as agree on the same subset of factors which are important in empirical contexts. We find that the of all the most considered, the penalized linear models are the most consistent at identifying true causal regressors through the simulation study. We find that in the empirical setting, among the individual the individual 1 and 6 month momentum factors are the most powerful predictors of stock returns, according to the penalized linear models and random forests.

Across all models except for linear models, we find that minimizing quantile loss yields better prediction performance.

The overall findings of this paper differ from the sparse literature on similar topics. However, the performance of the penalized linear models with respect to both out of sample prediction performance and variable importance analysis is promising, and our findings show that machine learning provides some tools which may aid in the problems of stock return prediction and risk factor selection in the financial world.

# 6. Appendix

## *6.1. Data*

Table 2: Individual Factors

| No. | Acronym | Firm Characteristic | Author(s) | Data Source | Frequency |
|---|---|---|---|---|---|
| 1 | ind_absacc[4] | Absolute Accruals | Bandyopadhyay et al. (2010) | Compustat | Annual |
| 2 | ind_acc[4] | Working capital accruals | Sloan (1996) | Compustat | Annual |
| 3 | ind_aeavol | Abnormal earnings announcement volume | Lerman et al. (2008) | Compustat | Quarterly |
| 4 | ind_age | # years since first Compustat coverage | Jiang et al. (2005) | Compustat | Annual |
| 5 | ind_agr | Asset growth | Cooper et al. (2008) | Compustat | Annual |
| 6 | ind_baspread | Bid-ask spread | Amihud and Mendelson (1989) | Compustat | Monthly |
| 7 | ind_beta | Beta | Fama and MacBeth (1973) | Compustat | Monthly |
| 8 | ind_betasq | Beta squared | Fama and MacBeth (1973) | Compustat | Monthly |
| 9 | ind_bm | Book-to-market | Rosenberg et al. (1985) | Compustat | Annual |
| 10 | ind_bm_ia | Industry-adjusted book to market | Asness et al. (2000) | Compustat | Quarterly |
| 11 | ind_cash | Cash holdings | Palazzo (2012) | Compustat | Annual |
| 12 | ind_cashdebt | Cashflow to debt | Ou and Penman (1989) | Compustat | Annual |
| 13 | ind_cashpr | Cash productivity | Chandrashekar and Rao (2009) | Compustat | Annual |
| 14 | ind_cfp[4] | Cashflow to price ratio | Desai et al. (2004) | Compustat | Annual |
| 15 | ind_cfp_ia[4] | Industry-adjusted cashfow to price ratio | Asness et al. (2000) | Compustat | Annual |
| 16 | ind_chatoia[4] | Industry-adjusted change in asset turnover | Soliman (2008) | Compustat | Annual |
| 17 | ind_chcsho | Change in shares outstanding | Pontiff and Woodgate (2008) | Compustat | Annual |
| 18 | ind_chempia | Industry-adjusted change in employee | Asness et al. (2000) | Compustat | Annual |
| 19 | ind_chinv[4] | Change in inventory | Thomas and Zhang (2002) | Compustat | Annual |
| 20 | ind_chmom | Change in 6-month momentum | Gettleman and Marks (2006) | Compustat | Monthly |
| 21 | ind_chpmia[4] | Industry-adjusted change in profit margin | Soliman (2008) | Compustat | Annual |
| 22 | ind_chtx | Change in tax expense | Thomas and Zhang (2011) | Compustat | Quarterly |
| 23 | ind_cinvest | Corporate investment | Titman et al. (2004) | Compustat | Quarterly |
| 24 | ind_convind | Convertible debt indicator | Valta (2016) | Compustat | Annual |
| 25 | ind_currat | Current ratio | Ou and Penman (1989) | Compustat | Annual |

[4]The factor was included in the original dataset provided by Gu et al. (2018), but was not used due to missing data issues

| | | | | | |
|---|---|---|---|---|---|
| 26 | ind_depr | Depreciation / PP&E | Holthausen and Larcker (1992) | Compustat | Annual |
| 27 | ind_divi | Dividend initiation | Michaely et al. (1995) | Compustat | Annual |
| 28 | ind_divo | Dividend omission | Michaely et al. (1995) | Compustat | Annual |
| 29 | ind_dolvol | Dollar trading volume | Chordia et al. (2001) | Compustat | Monthly |
| 30 | ind_dy | Dividend to price | Litzenberger and Ramaswamy (1982) | Compustat | Annual |
| 31 | ind_ear | Earnings announcement return | Brandt et al. (2008) | Compustat | Quarterly |
| 32 | ind_egr | Growth in common shareholder eq | Richardson et al. (2005) | Compustat | Annual |
| 33 | ind_ep | Earnings to price | Basu (1977) | Compustat | Annual |
| 34 | ind_gma | Gross profitability | Novy-Marx (2013) | Compustat | Annual |
| 35 | ind_grCAPX[4] | Growth in capital expenditures | Anderson and Garcia-Feijóo (2006) | Compustat | Annual |
| 36 | ind_grltnoa[4] | Growth in long term net operating assets | Fairfield et al. (2003) | Compustat | Annual |
| 37 | ind_herf | Industry sales concentration | Hou and Robinson (2006) | Compustat | Annual |
| 38 | ind_hire | Employee growth rate | Belo et al. (2014) | Compustat | Annual |
| 39 | ind_idiovol | Idiosyncratic return volatility | Ali et al. (2003) | Compustat | Monthly |
| 40 | ind_ill | Illiquidity | Amihud (2002) | Compustat | Monthly |
| 41 | ind_indmom | Industry momentum | Moskowitz and Grinblatt (1999) | Compustat | Monthly |
| 42 | ind_invest[4] | Capital expenditures and inventory | Chen and Zhang (2010) | Compustat | Annual |
| 43 | ind_lev | Leverage | Bhandari (1988) | Compustat | Annual |
| 44 | ind_lgr | Growth in long-term debt | Richardson et al. (2005) | Compustat | Annual |
| 45 | ind_maxret | Maximum daily return | Bali et al. (2011) | Compustat | Monthly |
| 46 | ind_mom12m | 12-month momentum | Jegadeesh (1990) | Compustat | Monthly |
| 47 | ind_mom1 | 1-month momentum | Jegadeesh and Titman (1993) | Compustat | Monthly |
| 48 | ind_mom36m[4] | 36-month momentum | Jegadeesh and Titman (1993) | Compustat | Monthly |
| 49 | ind_mom6m | 6-month momentum | Jegadeesh and Titman (1993) | Compustat | Monthly |
| 50 | ind_ms | Financial statement score | Mohanram (2005) | Compustat | Quarterly |
| 51 | ind_mvel1 | Size | Banz (1981) | Compustat | Monthly |
| 52 | ind_mve_ia | Industry-adjusted size | Asness et al. (2000) | Compustat | Annual |
| 53 | ind_nincr | Number of earnings increases | Barth et al. (1999) | Compustat | Quarterly |
| 54 | ind_operprof[4] | Operating profitability | Fama and French (2015) | Compustat | Annual |

| | | | | | |
|---|---|---|---|---|---|
| 55 | ind_orgcap[4] | Organizational capital | Eisfeldt and Papanikolaou (2013) | Compustat | Annual |
| 56 | ind_pchcapx_ia[4] | Industry adjusted % change in capital expenditures | Abarbanell and Bushee (1998) | Compustat | Annual |
| 57 | ind_pchcurrat[4] | % change in current ratio | Ou and Penman (1989) | Compustat | Annual |
| 58 | ind_pchdepr[4] | % change in depreciation | Holthausen and Larcker (1992) | Compustat | Annual |
| 59 | ind_pchgm_pchsale[4] | % change in gross margin - % change in sales | Abarbanell and Bushee (1998) | Compustat | Annual |
| 60 | ind_pchquick[4] | % change in quick ratio | Ou and Penman (1989) | Compustat | Annual |
| 61 | ind_pchsale_pchinvt[4] | % change in sales - % change in inventory | Abarbanell and Bushee (1998) | Compustat | Annual |
| 62 | ind_pchsale_pchrect[4] | % change in sales - % change in A/R | Abarbanell and Bushee (1998) | Compustat | Annual |
| 63 | ind_pchsale_pchxsga[4] | % change in sales - % change in SG | Abarbanell and Bushee (1998) | Compustat | Annual |
| 64 | ind_pchsaleinv[4] | % change sales-to-inventory | Ou and Penman (1989) | Compustat | Annual |
| 65 | ind_pctacc[4] | Percent accruals | Hafzalla et al. (2011) | Compustat | Annual |
| 66 | ind_pricedelay | Price delay | Hou and Moskowitz (2005) | Compustat | Monthly |
| 67 | ind_ps | Financial statements score | Piotroski (2000) | Compustat | Annual |
| 68 | ind_quick | Quick ratio | Ou and Penman (1989) | Compustat | Annual |
| 69 | ind_rd | R&D increase | Eberhart et al. (2004) | Compustat | Annual |
| 70 | ind_rd_mve[4] | R&D to market capitalization | Guo et al. (2006) | Compustat | Annual |
| 71 | ind_rd_sale[4] | R&D to sales | Guo et al. (2006) | Compustat | Annual |
| 72 | ind_realestate[4] | Real estate holdings | Tuzel (2010) | Compustat | Annual |
| 73 | ind_retvol | Return volatility | Ang et al. (2006) | Compustat | Monthly |
| 74 | ind_roaq | Return on assets | Balakrishnan et al. (2010) | Compustat | Quarterly |
| 75 | ind_roavol[4] | Earnings volatility | Francis et al. (2004) | Compustat | Quarterly |
| 76 | ind_roeq | Return on equity | Hou et al. (2015) | Compustat | Quarterly |
| 77 | ind_roic | Return on invested capital | Brown and Rowe (2007) | Compustat | Annual |
| 78 | ind_rsup | Revenue surprise | Kama (2009) | Compustat | Quarterly |
| 79 | ind_salecash | Sales to cash | Ou and Penman (1989) | Compustat | Annual |
| 80 | ind_saleinv[4] | Sales to inventory | Ou and Penman (1989) | Compustat | Annual |
| 81 | ind_salerec | Sales to receivables | Ou and Penman (1989) | Compustat | Annual |
| 82 | ind_secured[4] | Secured debt | Valta (2016) | Compustat | Annual |
| 83 | ind_securedind | Secured debt indicator | Valta (2016) | Compustat | Annual |

| | | | | | |
|---|---|---|---|---|---|
| 84 | ind_sgr[4] | Sales growth | Barbee Jr et al. (1996) | Compustat | Annual |
| 85 | ind_sin | Sin stocks | Hong and Kacperczyk (2009) | Compustat | Annual |
| 86 | ind_sp | Sales to price | Barbee Jr et al. (1996) | Compustat | Annual |
| 87 | ind_std_dolvol | Volatility of liquidity (dollar trading volume) | Chordia et al. (2001) | Compustat | Annual |
| 88 | ind_std_turn | Volatility of liquidity (share turnover) | Chordia et al. (2001) | Compustat | Monthly |
| 89 | ind_stdacc[4] | Accrual volatility | Bandyopadhyay et al. (2010) | Compustat | Monthly |
| 90 | ind_stdcf[4] | Cashflow volatility | Huang (2009) | Compustat | Quarterly |
| 91 | ind_tang | Debt capacity/rm tangibility | Almeida and Campello (2007) | Compustat | Quarterly |
| 92 | ind_tb[4] | Tax income to book income | Lev and Ohlson (1982) | Compustat | Annual |
| 93 | ind_turn | Share turnover | Datar et al. (1998) | Compustat | Monthly |
| 94 | ind_zerotrade | Zero trading days | Liu (2006) | Compustat | Monthly |

Table 3: Macroeconomic Factors

| No. | Acronym | Macroeconomic Factor |
|---|---|---|
| 1 | macro_dp | Dividend Price Ratio |
| 2 | macro_ep | Earnings Price Ratio |
| 3 | macro_bm | Book to Market Ratio |
| 4 | macro_ntis | Net Equity Expansion |
| 5 | macro_tbl | Treasury Bill Rate |
| 6 | macro_tms | Term Spread |
| 7 | macro_dfy | Default Spread |
| 8 | macro_svar | Stock Variance |

## *6.2. Additional Results*

### *6.2.1. Simulation Study*

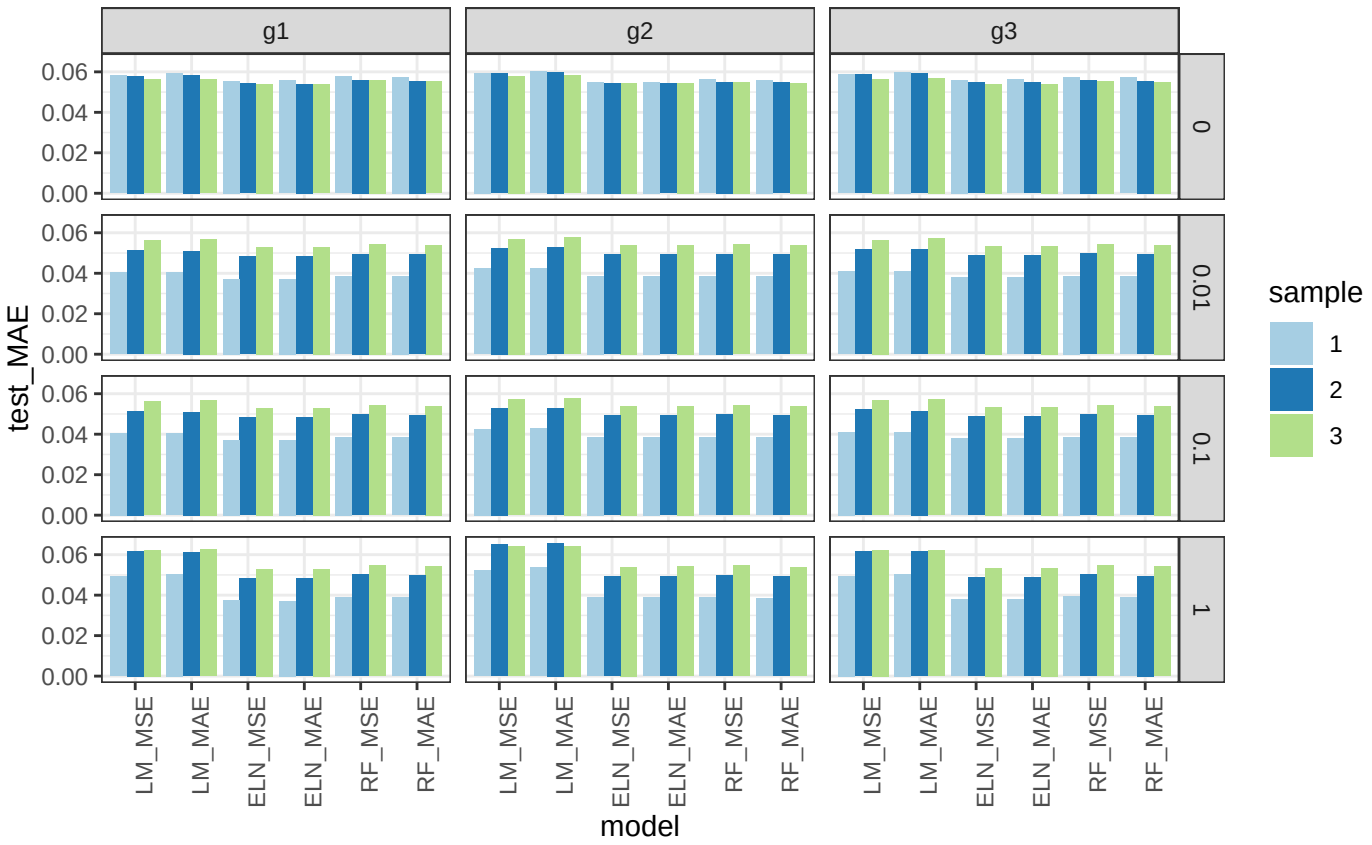



Figure 12. Simulation Test MAE

*6.2.2. Empirical Study*

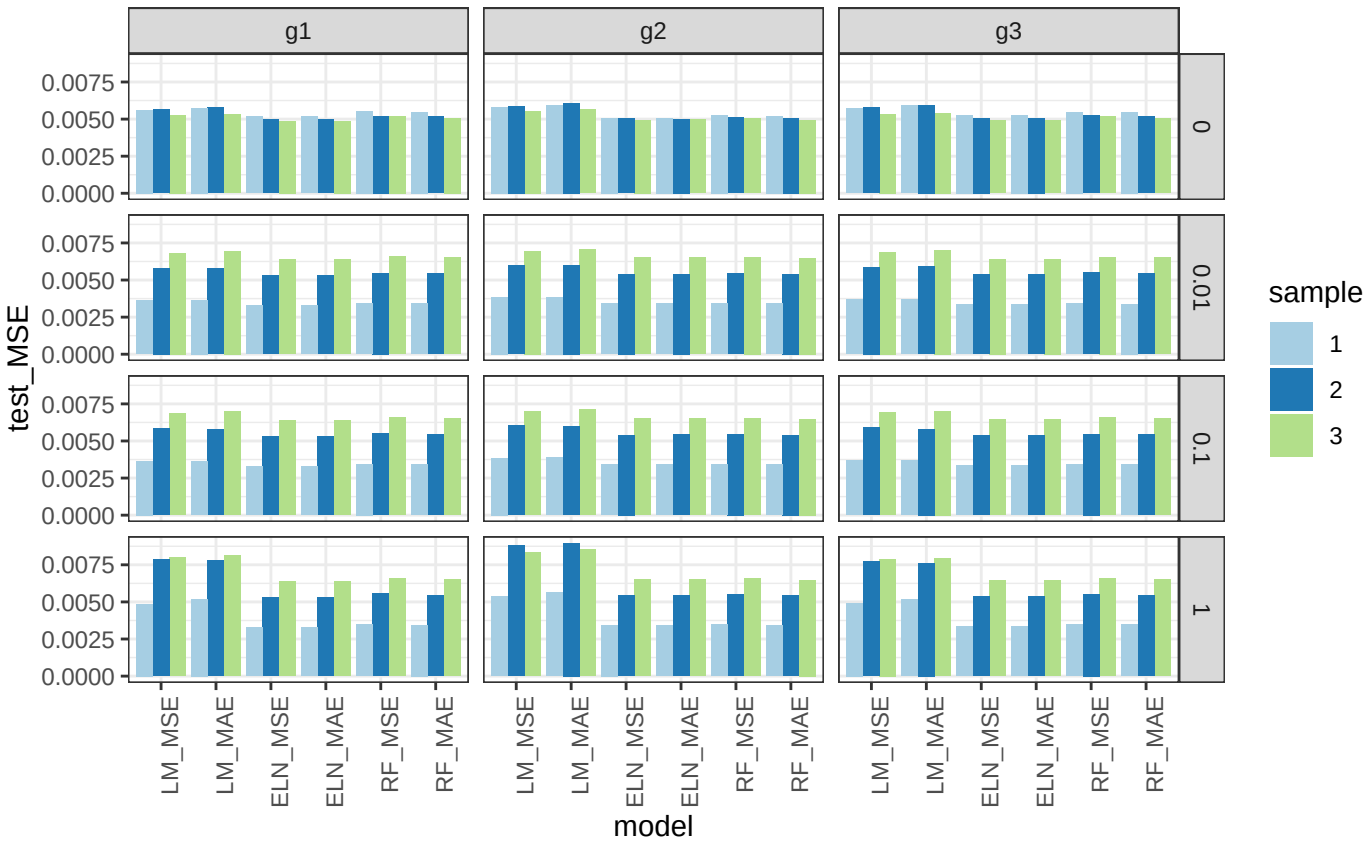



Figure 13. Simulation Test MSE

## *6.3. Computational Details*

### *6.3.1. Linear Models*

OLS used for fitting wrt MSE.

quantreg package used for fitting wrt MAE. (minimizing 0.5 quantile loss is equivalent to minimizing mae).

### *6.3.2. Penalized Linear*

The package **hqreg** was used to fit penalized regression models with respect to MSE and MAE.

This package efficiently calculates a regularization path of penalization values given a value for $\alpha$. This means that it is much more efficient to instead only supply a grid for $\alpha$, let the algorithm decide its own path of penalization values. The combination of these two parameters which produces the best results on the validation set were then chosen. Note that this is the approach originally suggested by Zou and Hastie (2005).

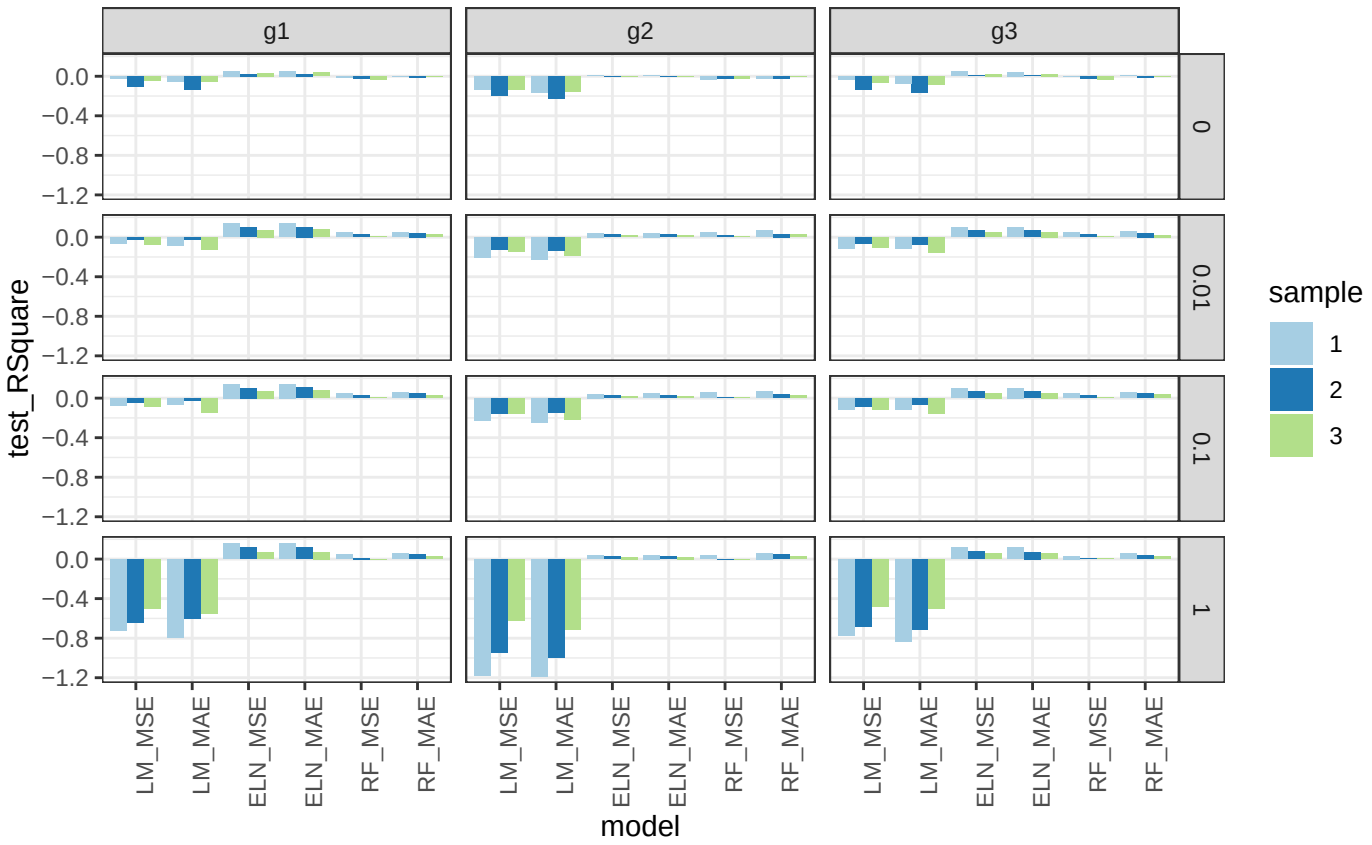



Figure 14. Simulation Test R Squared

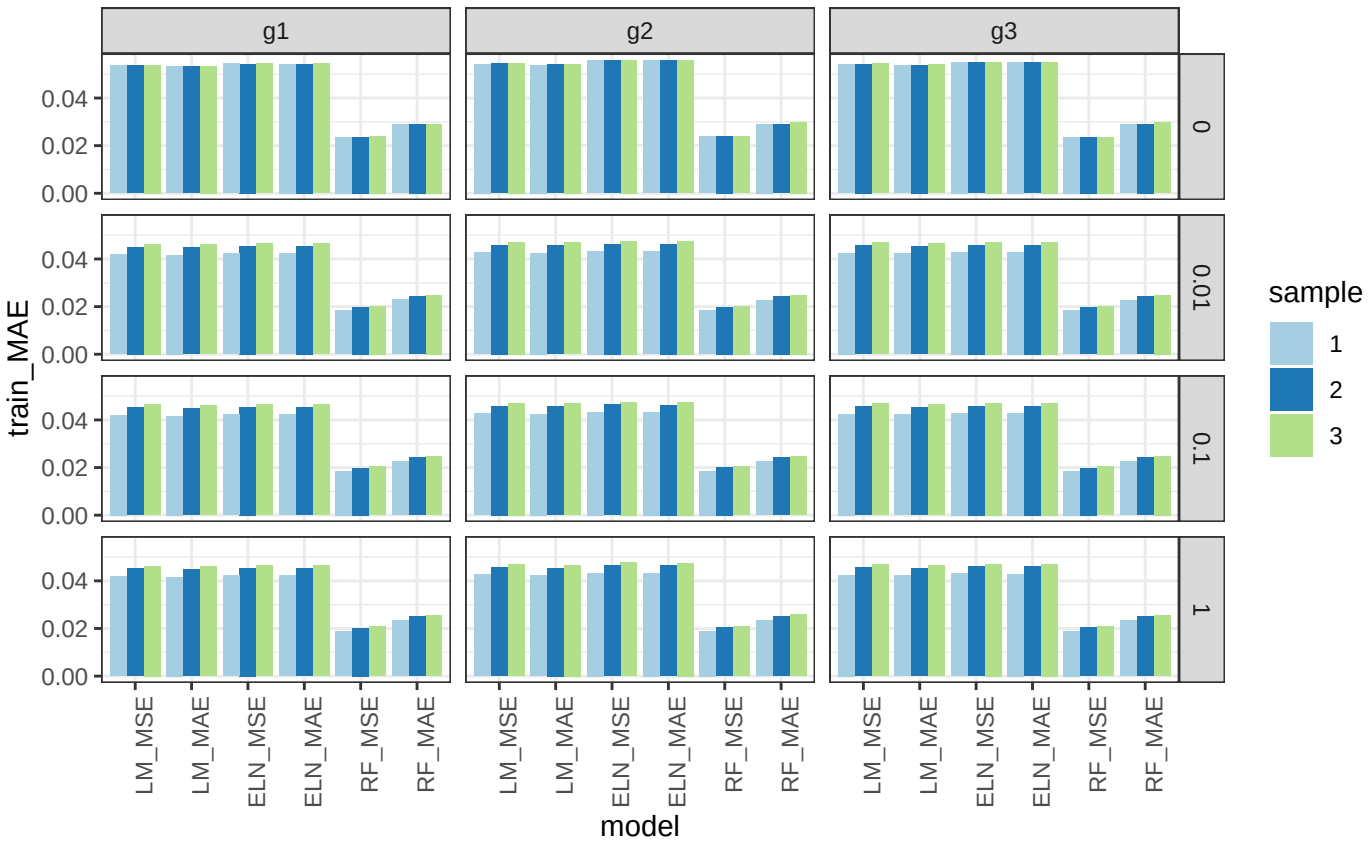



Figure 15. Simulation Train MAE

### *6.3.3. Classification and Regression Trees*

For full details of the Classification and Regression Tree algorithm see Breiman (1984).

**Algorithm 1:** Classification and Regression Tree

Initialize ;

**for** *d from 1 to L* **do**

**for** *i in* $C_l(d-1), l = 1, \ldots, 2^{d-1}$ **do**

For each feature $j = 1, 2, \ldots, P$, and each threshold level $\alpha$, define a split as $s = (j, \alpha)$ which divides $C_l(d-1)$ into $C_{left}$ and $C_{right}$:

$$C_{left}s = \{z_j \leq \alpha\} \cap C_l(d-1); C_{right}s = \{z_j > \alpha\} \cap C_l(d-1)$$

Define the impurity function:

$$\mathcal{L}(C, C_{left}, C_{right}) = \frac{|C_{left}|}{|C|} H(C_{left}) + \frac{|C_{right}|}{|C|} H(C_{right})$$

where

$$H(C) = \frac{1}{|C|} \sum^{z_{i,t} \in C} (r_{i,t+1} - \theta)^2, \theta = \frac{1}{|C|} \sum_{z_{i,t} \in C} r_{i,t+1}$$

and $|C|$ denotes the number of observations in set C

Find the optimal split

$$s^* \leftarrow \underset{s}{argmin} \mathcal{L}(C(s), C_{left}(s), C_{right}(s))$$

Update nodes (partition the data):

$$C_{2l-1}(d) \leftarrow C_{left}(s^*), C_{2l}(d) \leftarrow C_{right}(s^*)$$

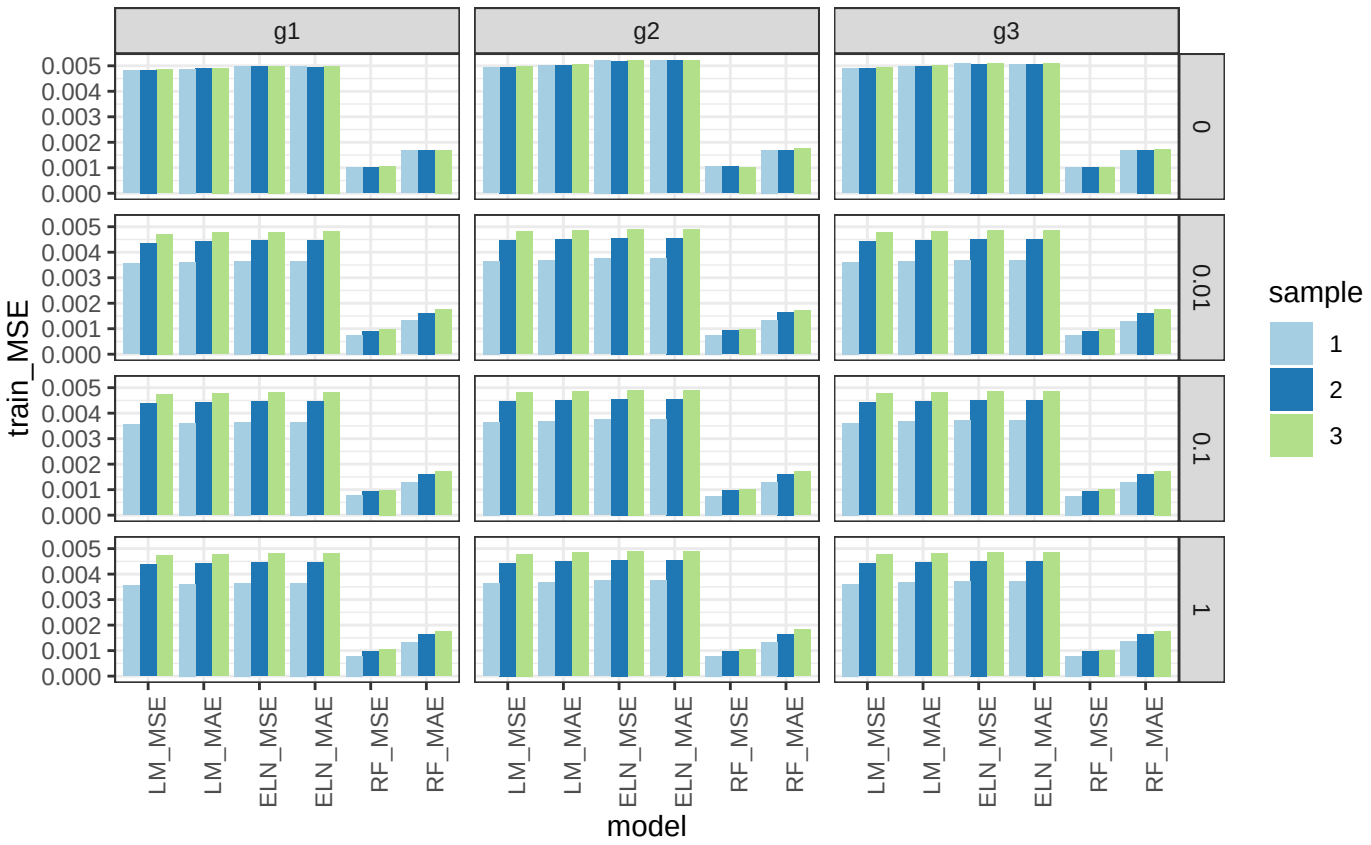



Figure 16. Simulation Train MSE

### *6.3.4. Random Forest*

The **randomforestSRC** package was used to fit both random forests for mean regression and quantile regression.

To maintain computational feasibility, all random forest models were grown using 50 trees, and their *mtry nodesize* hyperparameters tuned.

Rather than use the randomly determined out of bag (OOB) observations for hyperparameter tuning, we explicitly use a separate validation set that is closer to the test set in time, in order to maintain temporal ordering of the data.

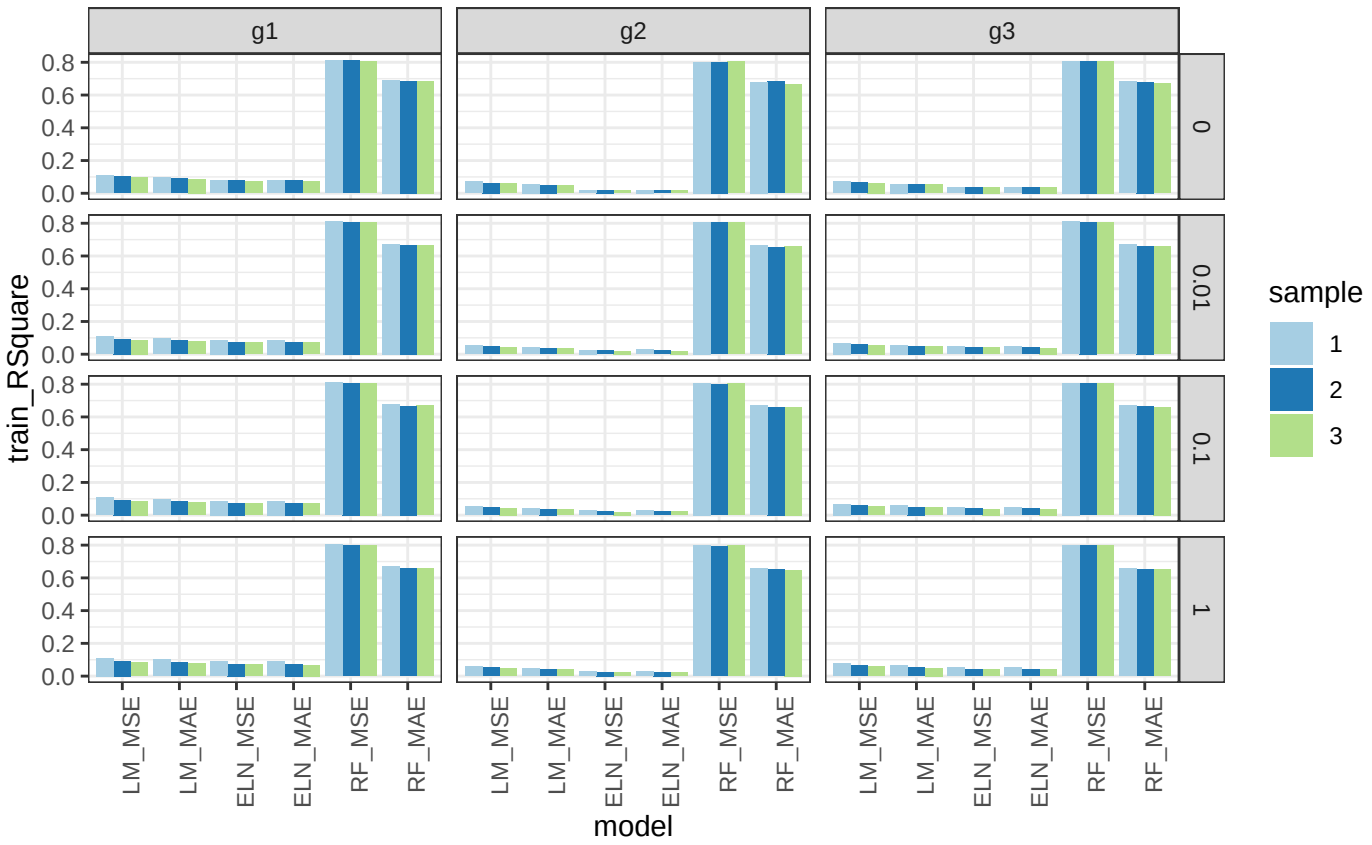



Figure 17. Simulation Train R Squared

For full details of the Random Forest algorithm see Breiman (2001).

**Algorithm 2:** Random Forest

**for** $b$ *from* 1 *to* $B$ **do**

Draw bootstrap samples $(z_{i,t}, r_{i,t+1}), (i,t) \in Bootstrap(b)$ from the dataset Grow a tree $T_b$ using Algorithm, using only a random subsample, say $\sqrt{P}$ of all features Denote the resulting $bth$ tree as

$$\hat{g}_b(z_{i,t}, \hat{\theta}_b, L) = \sum_{k=1}^{2^L} \theta_b^k \mathbf{1}_{z_{i,t} \in C_k(L)}$$

**end**

**Result:** The final random forest prediction is given by the output of all trees:

$$\hat{g}_b(z_{i,t}; L, B) = \frac{1}{B} \sum_{b=1}^{B} \hat{g}_b(z_{i,t}, \hat{\theta}_b, L)$$

### *6.3.5. Neural Networks*

Nueral Networks were fit using the **keras** package, using the **tensorflow** backend.

Due to computational limitations, a systematic grid search or random search approach for hyperparameters was not feasible. Instead, common hyperparameters that are usually grid searched were tuned manually.

ADAM algorithm for stochastic gradient descent and learning rate shrinkage as detailed by Kingma

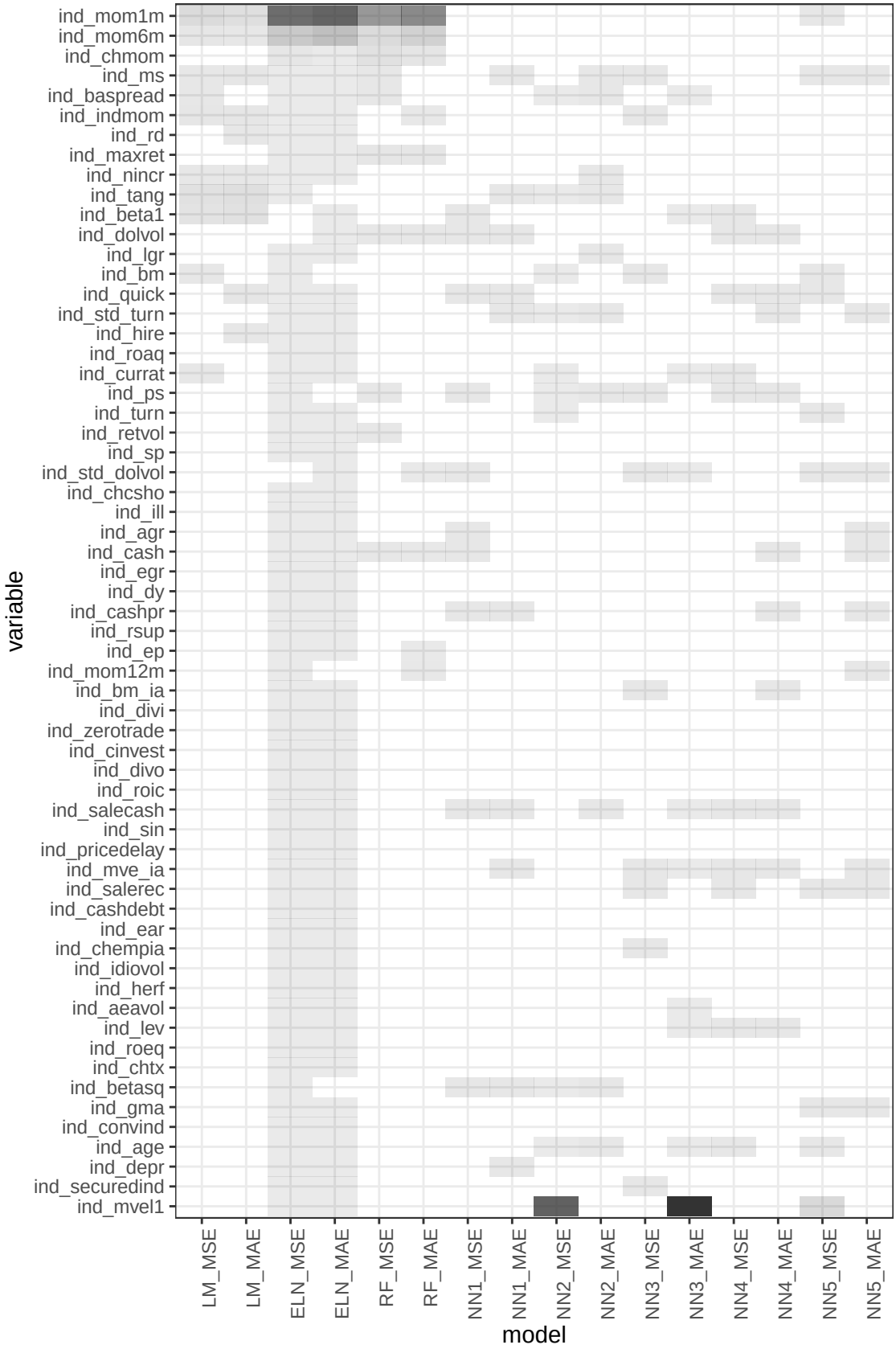



Figure 18. Empirical Data Individual Factor Variable Importance for Sample 1

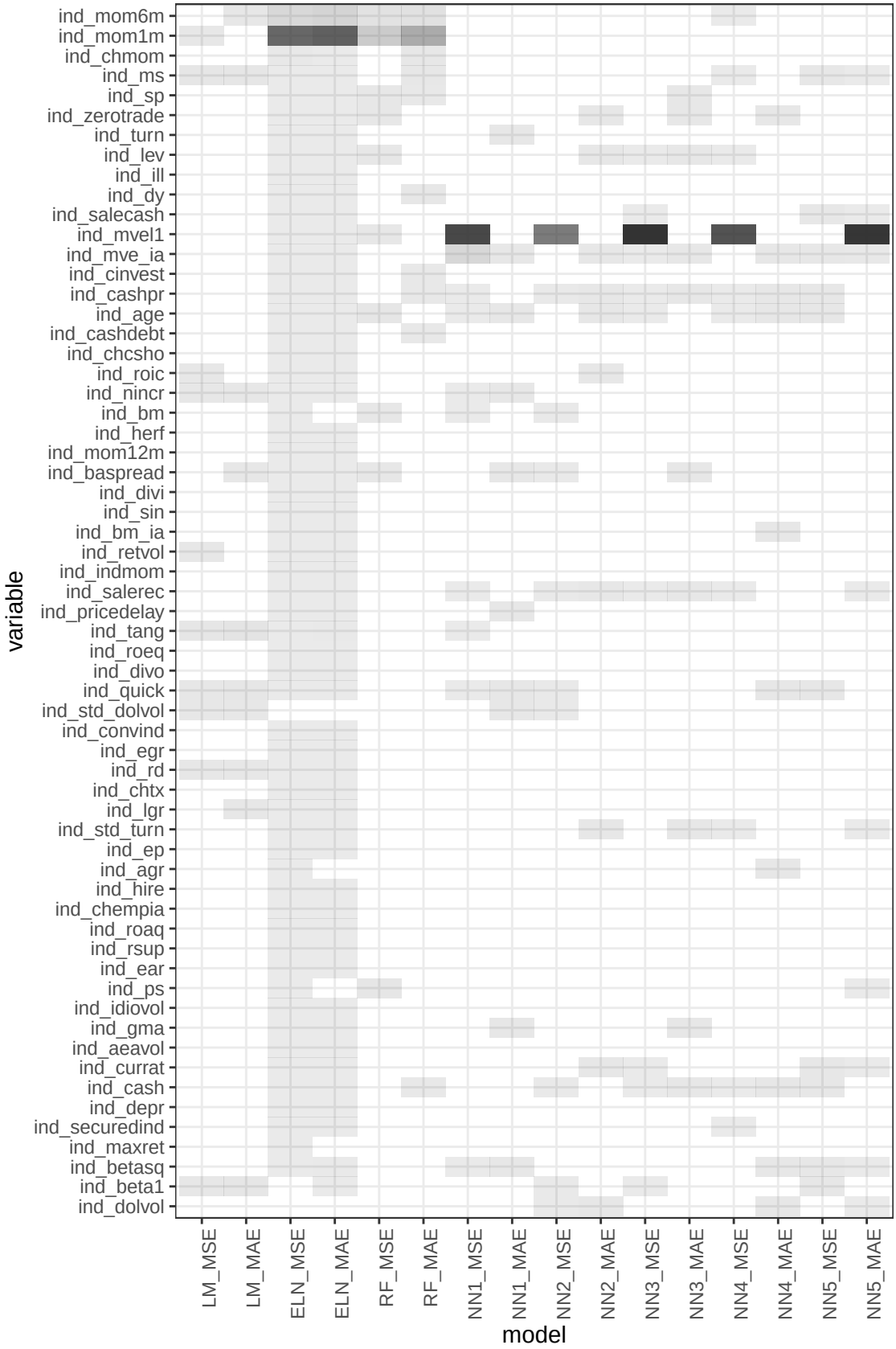



Figure 19. Empirical Data Individual Factor Variable Importance for Sample 2

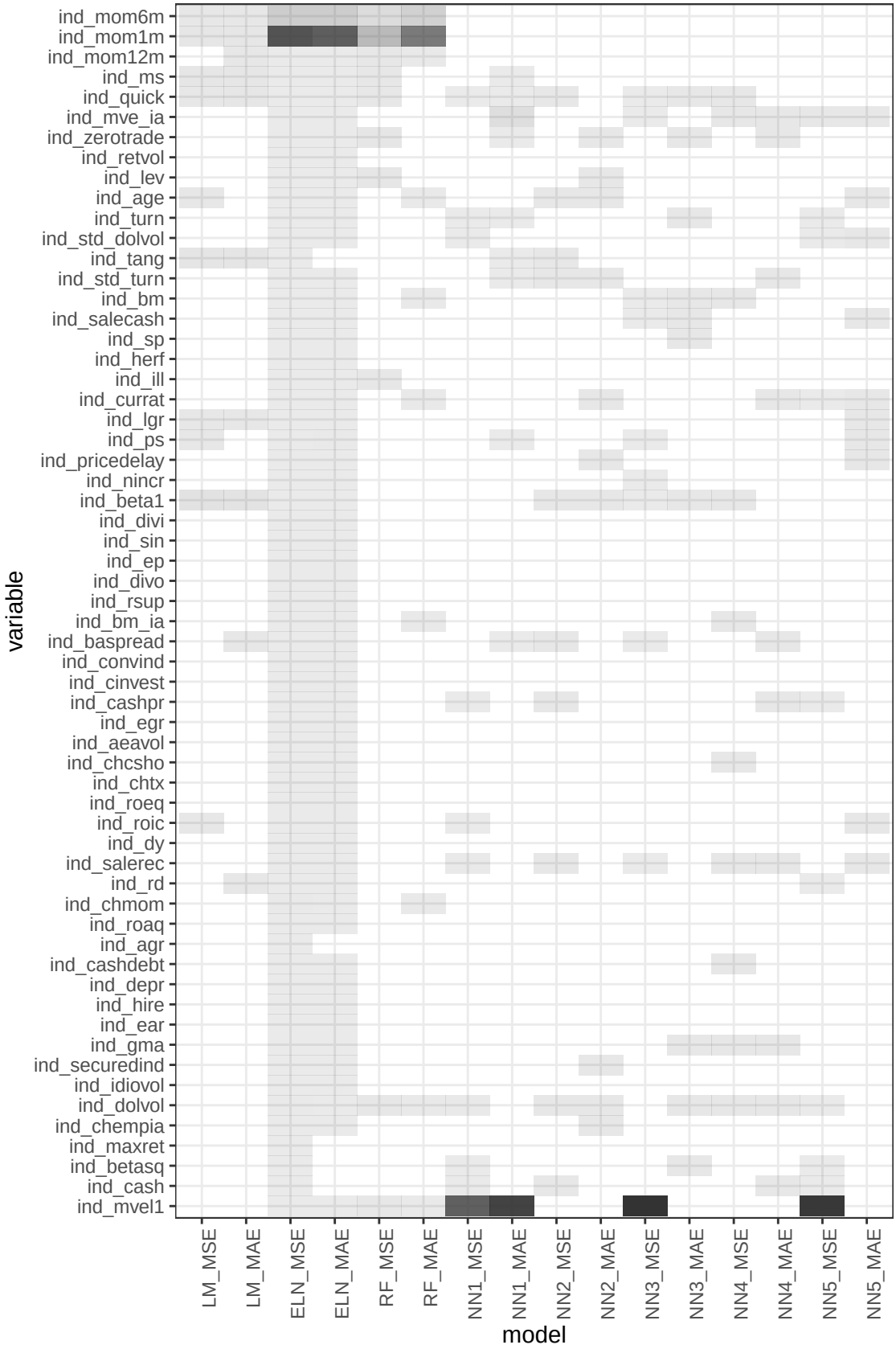



Figure 20. Empirical Data Individual Factor Variable Importance for Sample 3

| Hyperparameter | Options Considered | Optimal Choice |
|---|---|---|
| Activation Function | ReLU, Leaky ReLU with $\alpha = (0.01, 0.1)$, tanh | tanh |
| Optimizer | ADAM, NADAM | ADAM |
| Learning Rate | (0.1, 0.01, 0.001, 0.0001) | 0.01 |
| L1 Penalty | (1, 0.1, 0.01) | 0.01 |
| Batch Size | (32, 64, 128, 256, 512, 1024, 2048, 4096) | 256 (Empirical), 128 (Simulated) |
| Early Stopping Patience | (10, 20, 30, 40, 50) | 40 |

Table 4: Hyperparameters Considered for Neural Networks

and Ba (2014).

**Algorithm 3:** Early stopping via validation

Initialize $j = 0$, $\epsilon = \infty$ and select the patience parameter $p$ (max iterations)
**while** $j$ ¡ $p$ **do**
  Update $\theta$ using the training algorithm Calculate the prediction error from the validation sample, denoted as $\epsilon'$
  **if** $\epsilon' < \epsilon$ **then**
    $j \leftarrow 0$
    $\epsilon \leftarrow \epsilon'$
    $\theta' \leftarrow \theta$
  **else**
    $j \leftarrow j + 1$
  **end**
**end**
**Result:** $\theta'$ is the final parameter estimate

Batch Normalization Algorithm as detailed by Ioffe and Szegedy (2015).

**Algorithm 4:** Batch Normalization for one activation over one batch

Input: Values of $x$ for each activation over a batch $\mathcal{B} = x_1, x_2, \ldots, x_N$
$\mu_{\mathcal{B}} \leftarrow \frac{1}{N}\sum_{i=1}^{N} x_i$
$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{\mathcal{B}})^2$
$\hat{x_i} \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}}$
$y_i \leftarrow \gamma\hat{x_i} + \beta := BN_{\gamma,\beta}(x_i)$
**Result:** $y_i = BN_{\gamma,\beta}(x_i) : i = 1, 2, ..., N$

#### *6.3.6. Tuning of Simulated Datasets*

The simulated datasets were tuned according to the following statistics: average individual time series R squared, average annualized volatility, and cross sectional R squared, and true/predictive R squared.

The methodology for evaluating average time series R squared and cross sectional R squared is consistent with that detailed by Cochrane (2005). The steps are reproduced here for reference; for complete details refer to Cochrane (2005).

First evaluate the following OLS model:

$$R_{it} = a_i + \beta_i' f_{it} + \epsilon_{it} \tag{30}$$

where $f_{it}$ represents the *true* factors in the returns process. The corresponding R-squared value for this time series regression is calculated across all stocks and averaged to yield the average time series R-Squared.

A cross sectional regression for the risk premia is then run across assets of average returns on the factor coefficients:

$$\bar{R_{it}} = \alpha_i = \beta_i' \lambda \tag{31}$$

where the $\beta_i'$ are the estimated coefficients from each time series regressions run previously. The corresponding R squared for this regression is the cross sectional R squared.

The true R squared is a measure of signal to noise ratio i.e. how much of the returns data is due to This is simply calculated by running a pooled ordinary least squares regression on the entire panel, using the underlying g() as the "predictions.' The resulting R squared value is therefore a measure of how much of the panel can be explained by the g() term exclusive of any noise or error terms.

# References


Abarbanell, J. S., Bushee, B. J., 1998. Abnormal returns to a fundamental analysis strategy. Accounting Review pp. 19–45.

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

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

Amihud, Y., 2002. Illiquidity and stock returns: cross-section and time-series effects. Journal of financial markets 5, 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, 479–486.

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

Ang, A., Bekaert, G., 2006. Stock return predictability: Is it there? The Review of Financial Studies 20, 651–707.

Ang, A., Hodrick, R. J., Xing, Y., Zhang, X., 2006. The Cross-Section of Volatility and Expected Returns. The Journal of Finance 61, 259–299.

Asness, C. S., Porter, R. B., Stevens, R. L., 2000. Predicting stock returns using industry-relative firm characteristics. Available at SSRN 213872 .

Balakrishnan, K., Bartov, E., Faurel, L., 2010. Post loss/profit announcement drift. Journal of Accounting and Economics 50, 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, 427–446.

Bandyopadhyay, S. P., Huang, A. G., Wirjanto, T. S., 2010. The accrual volatility anomaly. Unpublished Manuscript, University of Waterloo .

Banz, R. W., 1981. The relationship between return and market value of common stocks. Journal of financial economics 9, 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, 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, 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, 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, 129–177.

Bhandari, L. C., 1988. Debt/equity ratio and expected common stock returns: Empirical evidence. The journal of finance 43, 507–528.

Brandt, M. W., Kishore, R., Santa-Clara, P., Venkatachalam, M., 2008. Earnings announcements are full of surprises. SSRN eLibrary .

Breiman, L., 1984. Classification and Regression Trees. Routledge.

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

Brown, D. P., Rowe, B., 2007. The productivity premium in equity returns. Available at SSRN 993467 .

Chandrashekar, S., Rao, R. K., 2009. The productivity of corporate cash holdings and the cross-section of expected stock returns. McCombs Research Paper Series No. FIN-03-09 .

Chen, L., Zhang, L., 2010. A better three-factor model that explains more anomalies. Journal of Finance 65, 563–595.

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

Cochrane, J. H., 2005. Asset pricing. Princeton Univ. Press, Princeton, NJ, rev. ed ed., oCLC: 265653065.

Cochrane, J. H., 2011. Presidential Address: Discount Rates. The Journal of Finance 66, 1047–1108.

Cooper, M. J., Gulen, H., Schill, M. J., 2008. Asset Growth and the Cross-Section of Stock Returns. The Journal of Finance 63, 1609–1651.

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

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

Diebold, F. X., Mariano, R. S., 2002. Comparing predictive accuracy. Journal of Business & economic statistics 20, 134–144.

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, 623–650.

Eisfeldt, A. L., Papanikolaou, D., 2013. Organization capital and the cross-section of expected returns. The Journal of Finance 68, 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, 353–371.

Fama, E. F., French, K. R., 2015. A five-factor asset pricing model. Journal of financial economics 116, 1–22.

Fama, E. F., MacBeth, J. D., 1973. Risk, return, and equilibrium: Empirical tests. Journal of political economy 81, 607–636.

Feng, G., Giglio, S., Xiu, D., 2019. Taming the Factor Zoo: A Test of New Factors. Tech. Rep. w25481, National Bureau of Economic Research, Cambridge, MA.

Feng, G., He, J., Polson, N. G., 2018. Deep Learning for Predicting Asset Returns. arXiv:1804.09314 [cs, econ, stat] ArXiv: 1804.09314.

Francis, J., LaFond, R., Olsson, P. M., Schipper, K., 2004. Costs of equity and earnings attributes. The accounting review 79, 967–1010.

Freyberger, J., Neuhierl, A., Weber, M., 2017. Dissecting characteristics nonparametrically. Tech. rep., National Bureau of Economic Research.

Gettleman, E., Marks, J. M., 2006. Acceleration strategies. Tech. rep., Working Paper, Seton Hall Univeristy.

Goetzmann, W. N., Jorion, P., 1993. Testing the predictive power of dividend yields. The Journal of Finance 48, 663–679.

Goyal, A., Welch, I., 2003. Predicting the equity premium with dividend ratios. Management Science 49, 639–654.

Gu, S., Kelly, B., Xiu, D., 2018. Empirical asset pricing via machine learning. Tech. rep., National Bureau of Economic Research.

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, 550–579.

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

Harvey, C. R., Liu, Y., 2019. A Census of the Factor Zoo. Social Science Research Network p. 7.

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

Harvey, D., Leybourne, S., Newbold, P., 1997. Testing the equality of prediction mean squared errors. International Journal of Forecasting 13, 281–291.

Hastie, T., Tibshirani, R., Friedman, J. H., 2009. The elements of statistical learning: data mining, inference, and prediction. Springer series in statistics, Springer, New York, NY, second ed.

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

Holthausen, R. W., Larcker, D. F., 1992. The prediction of stock returns using financial statement information. Journal of accounting and economics 15, 373–411.

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

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural networks 2, 359–366.

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

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

Hou, K., Xue, C., Zhang, L., 2015. Digesting Anomalies: An Investment Approach. Review of Financial Studies 28, 650–705.

Hsu, J., Kalesnik, V., 2014. Finding smart beta in the factor zoo. Research Affiliates (July) .

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

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

Jegadeesh, N., 1990. Evidence of predictable behavior of security returns. The Journal of finance 45, 881–898.

Jegadeesh, N., Titman, S., 1993. Returns to buying winners and selling losers: Implications for stock market efficiency. The Journal of finance 48, 65–91.

Jiang, G., Lee, C. M., Zhang, Y., 2005. Information uncertainty and expected returns. Review of Accounting Studies 10, 185–221.

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

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P. T. P., 2016. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. arXiv:1609.04836 [cs, math] ArXiv: 1609.04836.

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

Kozak, S., Nagel, S., Santosh, S., 2017. Shrinking the cross section. Tech. rep., National Bureau of Economic Research.

Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

Lerman, A., Livnat, J., Mendenhall, R. R., 2008. The High-Volume Return Premium and Post-Earnings Announcement Drift. SSRN Electronic Journal .

Lettau, M., Ludvigson, S., 2001. Consumption, Aggregate Wealth, and Expected Stock Returns. The Journal of Finance 56, 815–849.

Lev, B., Ohlson, J. A., 1982. Market-based empirical research in accounting: A review, interpretation, and extension. Journal of Accounting research pp. 249–322.

Litzenberger, R. H., Ramaswamy, K., 1982. The effects of dividends on common stock prices tax effects or information effects? The Journal of Finance 37, 429–443.

Liu, W., 2006. A liquidity-augmented capital asset pricing model. Journal of financial Economics 82, 631–671.

Masters, T., 1993. Practical Neural Network Recipes in C++. Morgan Kaufmann, google-Books-ID: 7Ez_Pq0sp2EC.

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

Mohanram, P. S., 2005. Separating winners from losers among lowbook-to-market stocks using financial statement analysis. Review of accounting studies 10, 133–170.

Moskowitz, T. J., Grinblatt, M., 1999. Do industries explain momentum? The Journal of finance 54, 1249–1290.

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

Ou, J. A., Penman, S. H., 1989. Financial statement analysis and the prediction of stock returns. Journal of accounting and economics 11, 295–329.

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

Piotroski, J. D., 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, 921–945.

Ramachandran, P., Zoph, B., Le, Q. V., 2017. Searching for Activation Functions. arXiv:1710.05941 [cs] ArXiv: 1710.05941.

Rapach, D., Zhou, G., 2013. Forecasting Stock Returns. In: *Handbook of Economic Forecasting*, Elsevier, vol. 2, pp. 328–383.

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, 437–485.

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

Schwert, G. W., 2003. Anomalies and market efficiency. Handbook of the Economics of Finance 1, 939–974.

Sloan, R. G., 1996. Do Stock Prices Fully Reflect Information in Accruals and Cash Flows about Future Earnings? The Accounting Review 71, 289–315.

Soliman, M. T., 2008. The use of DuPont analysis by market participants. The Accounting Review 83, 823–853.

Thomas, J., Zhang, F. X., 2011. Tax expense momentum. Journal of Accounting Research 49, 791–821.

Thomas, J. K., Zhang, H., 2002. Inventory changes and future returns. Review of Accounting Studies 7, 163–187.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58, 267–288.

Titman, S., Wei, K. J., Xie, F., 2004. Capital investments and stock returns. Journal of financial and Quantitative Analysis 39, 677–700.

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

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

Welch, I., Goyal, A., 2008. A Comprehensive Look at The Empirical Performance of Equity Premium Prediction. Review of Financial Studies 21, 1455–1508.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the Elastic Net. Journal of the Royal Statistical Society, Series B 67, 301–320.