Deep Learning and the Cross-Section of Expected Returns*

Preliminary and Incomplete

Version 0.1.3

Marcial Messmer‡

University of St.Gallen

December 2, 2017

Abstract

Deep learning is an active area of research in machine learning. I train deep feedforward neural networks (DFN) based on a set of 68 firm characteristics (FC) to predict the US cross-section of stock returns. After applying a network optimization strategy, I find that DFN long-short portfolios can generate attractive risk-adjusted returns compared to a linear benchmark. These findings underscore the importance of non-linear relationships among FC and expected returns. The results are robust to size, weighting schemes and portfolio cutoff points. Moreover, I show that price related FC, namely, short-term reversal and the twelve-months momentum, are among the main drivers of the return predictions. The majority of FC play a minor role in the variation of these predictions.

JEL Classification: C52; C53; C58; G12; G17;

Key words: Cross Section of Returns; Deep Learning; Asset Pricing; Factor Models;

*I would like to thank Francesco Audrino, Dmitry Borisenko, Robert Hodrick and participants at the Columbia University PhD Seminar for insightful comments and fruitful discussions. Furthermore, I gratefully acknowledge financial support from the Swiss National Science Foundation and thank Columbia Business School and Robert Hodrick for granting access to the computer cluster — special thanks here for the tech support received by Dan Mechanic.

‡E-mail: marcial.messmer@gmail.com. Address: Bodanstrasse 6, 9000 St.Gallen, Switzerland.

1 Introduction

The rapid development of software and hardware technology in computer science in combination with the access to large amount of data enables the training of large and complex models. Particularly, advances in machine learning and artificial intelligence allow researchers to employ self-learning algorithms in many different areas where faced with prediction problems. In particular, deep learning algorithms show promising results in improving the prediction accuracy of regression and classification problems encountered in other areas of science, for example, image processing, speech recognition and medical drug discovery, see Krizhevsky et al. (2012). This work aims to answer the question if and how recent innovations in deep learning (DL) can improve the prediction of stock returns in a cross-section.

In general DL refers to a rich set of neural network models. However, the deep feedforward network (DFN) specification is the workhorse model of DL applications.[1] The latter marks the focus of this contribution. An appealing, but at the same time intimidating core characteristic steems from the universal approximation theorem, see Hornik et al. (1989) and Cybenko (1989). More explicitly, Hornik et al. (1989) states that a "multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available." This powerful statement implies that we have an extremely flexible framework which allows us to gauge the dependence of some input to some output data without any prior assumption regarding the functional form. Ultimately, this means that the model is purely determined by an algorithm. The key challenge for the researcher is then to train the DFN such that it provides a generalization of the data generating process (DGP).

The finance literature has recently addressed the question which firm characteristics (FC) provide independent information in explaining differences in expected cross-sectional returns - see for example, Harvey et al. (2016), McLean and Pontiff (2016), Green et al. (2017) and Messmer and Audrino (2017). This literature mainly builds on linear or linear-like models, notable exceptions are Freyberger et al. (2017) using non-parametric techniques, and Moritz and Zimmermann (2016) using a CART like approach to predict cross-sectional returns. Linear and other parsimonious models and methods are appealing for various reasons, for example, transparency and computational efficiency, where the former allows for a clear interpretation of the results and the latter for low implementation costs. However, this simplicity potentially misses important non-linear aspects of the true underlying DGP, which often results in a poor performance when evaluated from a prediction perspective. A classical example in the computer science literature is the XOR problem, which the linear model is incapable to capture despite its low

[1] Often DFNs are referred to as feedforward neural networks or multilayer perceptrons.

complexity. Applying DL contributes only marginally in learning more about the underlying forces of the DGP, however, it can help to understand to which degree we still have to learn from a pure data driven approach. Besides the loss of a straightforward model intuition, fitting a DL model is typically computationally much more challenging than a linear model.

Training a deep neural network for stock picking, is at least partly motivated by a recent important contribution to the open source software community. The so called "Tensorflow" (Abadi et al. (2015)) library provides a highly scalable and flexible machine learning framework, allowing an efficient usage of DL networks and is the core implementation of Google's artificial intelligence (AI), unit which is under active development. Moreover, the US cross-section of returns is a relative data rich environment. In this application, roughly 2.1 million observations provide a fertile ground for these parameter rich networks. On the other hand, it is hard to assess ex-ante if these methods are suitable for predicting stock returns, due to the inherently different statistical character they possess — for example, the signal-to-noise ratio of a stock return process is a tiny fraction compared to the processes typically encountered in computer science.

The main contribution of this work lies in the investigation if recent developments in artificial intelligence are of any use predicting cross-sectional stock returns. Applying artificial neural networks (ANN) in finance is not new. Hence, past attempts have to be distinguished. This study can be seen as an extension to earlier attempts of applying ANN to predict stock returns, with the difference of having access to additional regularization techniques, better computational resources and more data. Additionally, it is, to best of my knowledge, the first study which investigates the cross-section and its relation to a rich set of published FC by exploiting a purely data-driven algorithm without any prior assumption on the functional relation between FC and expected return spreads.

Precisely, this paper aims at answering the following main three research questions: First, how can one efficiently employ a DFN framework for the purpose of return predictions? Second, do DFN based predictions add additional economic value compared to a parsimonious linear approach utilizing the same information set? Third, which set of FC drive the prediction results and how far do they differ from recent findings in the literature of FC selection.

The first question is related to the selection problem of the optimal DFN design for this exercise. I address this question by stating the problem as an outer optimization problem. This computational intensive task is tackled by utilizing a random search algorithm as proposed in Bergstra and Bengio (2012) in combination with a one-dimensional grid search for learning rate tuning. The procedure reveals that many network designs fail to deliver reasonable numerical behavior. Despite a relatively high failure rate, I identify architectures which show promising improvements compared to the linear benchmarks based on a validation data set.

The short answer whether economically measurable improvements can be achieved, is yes. I find significant and robust factor α's, which are consistently higher compared to the parsimonious linear benchmark. In many (but not in all) cases I document significant higher Sharpe Ratios (SR). No specification favors the linear model, irrespectively of which performance measure is considered. However, a naive strategy is sensitive to trading cost adjustments for both approaches. Nonetheless, I show that a simple rebalancing frequency adjustment leads to stark improvements. An explicit rebalancing optimization is not carried out and can be seen as a limitation of this work. Over the sample period, I document that DFN based portfolios perform much weaker during high volatility periods compared to times of calmer markets, a phenomena which is characteristical for momentum based strategies. Controlling for momentum exposure during these times, levels the α's significantly into the positive domain.

The answer to the question which FC drive the predictions points unambiguously at price based information, predominantly short-term reversal (providing an explanation for the turnover intensity) and the twelve-months momentum. However, I study the impact purely by looking at prediction changes arising from variation in the input data. As a result, it can not be seen as a perfect measure, but a computational trivial way in gaining model insights at this stage.

This study includes cross-sectional stock data from the CRSP/Compustat database from 1970-2014. In total I use 68 published FC, constructed based on accounting and market data. The focus of the analysis lies solely on large and mid cap stocks, to prevent a potential contamination arising from economically unimportant small and micro-cap stocks, as recently documented in Hou et al. (2017). Notably, the latter study includes more than 400 FC, I instead rely on a much smaller subset, however, I include many prominent FC, such as the market beta, twelve-months momentum, book-to-market ratio, idiosyncratic volatility and profitability.

The paper is structured as follows. Section 2 reviews the relevant literature on finance applications of machine learning and briefly presents important contributions to the field of deep learning. Section 3 gives a detailed description of the deep-learning methodology utilized throughout the study. This part is followed by a brief description of the data. Section 5 includes the network design optimization. The penultimate section covers the portfolio mapping and the analysis corresponding to the long-short returns of these portfolios. The final part concludes.

2 Related Literature

In two seminal contributions Fama and French (1992, 1993) present the three-factor model, relating expected cross-sectional returns to a market, size and value factor. The model would mark the benchmark for the years to come. Irrespective of the presence of these three factors, a huge amount of new FC have been identified explaining additional variation in cross-sectional expected returns. Prominent examples are, momentum (Jegadeesh and Titman (1993)), quality (Novy-Marx (2013)), low-beta (Frazzini and Pedersen (2014)) and low-volatility (Ang et al. (2006)). Harvey et al. (2016) identify 300+ and, more recently, Hou et al. (2017) document and replicate a total of 447 such anomaly FC. This large number of FC is likely according to theses studies a consequence of a publication bias. Consequently, a number of studies address the question of digesting this "factor zoo" (Cochrane (2011)). Notable contributions are Harvey et al. (2016), McLean and Pontiff (2016) and Green et al. (2017). From a methodical point of view, Harvey et al. (2016) introduce the concept of family-wise error rates, McLean and Pontiff (2016) follow a true out-of-sample strategy, and Green et al. (2017) focus on the multivariate linear digestion. Common to all three papers are the findings that many of the published factors are indeed not robust to those alternative testing procedures.

Central to all of these papers is the understanding of the following relationship:

$$R^e_{t+1,n} = f(x_{t,n}, \theta) + \epsilon_{t+1,n}$$

where $f(\cdot)$ defines a generic function with parameters θ, $x_{t,n} = [size_{t,n}, bm_{t,n}, mom_{t,n}, ...]$ the vector of FC and $R^e_{t+1,n}$ the corresponding excess return and $\epsilon_{t+1,n}$ the error at a given point in time $t = 1, ..., T$ for stock $n = 1, ..., N$.

The following studies can be grouped into a rather young strand of the literature on machine learning and asset pricing — mostly shrinkage based approaches in the cross-section of stock returns. Moritz and Zimmermann (2016) use a CART (Breiman et al. (1984)) inspired methodology and find significance performance improvements. DeMiguel et al. (2017) take a portfolio perspective and optimize the problem from an investor's view, with an explicit trading cost term in the objective function. The findings show that only a limited number of FC are important, but trading diversification is an important aspect and leads eventually to a larger set of optimal FC. Kozak et al. (2017) use L1 and L2 norms and estimate the cross-section based on a stochastic discount factor specification, tying FC more to a risk-return relation. Their findings suggest the importance of interactions and characterize a non-sparse set of FC, explaining the cross-section of expected returns. Messmer and Audrino (2017) use the adaptive Lasso (Zou (2006)) to identify a sparser set of FC. The application is motivated based on an extensive cross-sectional simulation study, which indicates that the adaptive Lasso enjoys advantages over Lasso based shrinkage due to less stringent conditions imposed connected to the covariance structure of FC. Freyberger et al. (2017) tackle the problem by using a non-parametric model in combination with the grouped adaptive Lasso. Their approach is motivated by the classical portfolio sorting methodology and robust to extreme values.

The early work of artificial neural networks dates back to McCulloch and Pitts (1943), Hebb (1949) and Widrow et al. (1960). Moreover, two seminal contributions to the field are the studies by Parker (1985) and Williams and Hinton (1986), which facilitate the usage of DFN by advances of the back-propagation algorithm. Recently, the work of Bengio et al. (2007), Hinton (2007), Poultney et al. (2007) mark important achievements of artificial intelligence. For more details on the history of the deep learning and AI literature I refer the reader to Goodfellow et al. (2016).

Not surprisingly, the idea of applying DFNs to financial prediction problems is not new, see, for example, Trippi and DeSieno (1992) or Leung et al. (2000). However, the early work has not been too well received because of various reasons. The main problem connected to DFN is that they are prone to over-fitting. Consequently, the attention which is paid to DFNs is only marginal in the finance literature. Most closely related to this work is a paper by Takeuchi and Lee (2013), who analyze cross-sectional momentum strategies. The authors use a restricted Boltzmann machine approach and show performance improvements compared to the classical counterparts. In contrast to their work, this paper relies on a much wider set of FC and follows a different methodology by applying DFN. Recently, Dixon et al. (2015) show an application of deep learning to predict the returns of currency and commodities future contracts based on high-frequency data. The work is mainly focused on technical aspects, but it reports improvements in predicting market directions.

3 Methods

This section provides a brief introduction into deep learning, however, I focus on concepts mainly relevant for this work, for a detailed overview I refer the reader to Goodfellow et al. (2016), which I use as a main reference as well as a guidance for the notation throughout the paper. Deep learning describes a field of machine learning, which comprises a variety of different neural network model architectures. It aims to provide a generic, trainable framework capable of capturing almost any arbitrary functional dependence the DGP may be subject to. The ultimate goal for the researcher is to find a specific approximation of this specific function f, which expresses some output y as a function of an input vector $\mathbf{x}$. In my case the goal is to capture the functional relationship of expected cross-sectional returns based on a set of FC (which I indicate by k of K total FC). Let $y_{n,t+1}$ denote a prediction target, which is here, an excess return of firm n (of a total of N) at point in time t (of a total of T periods). The expression $f(x_{n,t};\theta)$ defines this function, where $x_{n,t}$ defines the input vector of FC, and θ the set of model parameters. The shape of θ, as well as the model capacity and hence, the richness of the set of functions is a result of the set of hyper-parameters, which has to be determined by the researcher and is denoted by λ. An example of θ is a vector of model coefficients, whereas the number of layers in the DFN is part of the set λ. Hyper-parameter selection is an optimization problem of its own, with the ultimate objective to minimize generalization error, subject to the computational constraints. This important problem is addressed in detail in section 5. First, I present an overview of the neural network anatomy and corresponding optimization problem.

3.1 Deep Learning Anatomy

The first step when specifying a neural network learning task is to select an appropriate network architecture. Possible families and combinations of therof are DFNs, convolutional neural networks (CNNs) or recurrent neural networks (RNNs). I skip a detailed discussion on the technical details and how and which data environment they can be used in. Instead I focus on the fully connected deep-feedforward neural network design, because its application is straightforward and does not require to manipulate, extend or order the input data. The other model families might potentially work as well, but would require non-trivial data extensions and transformations to accommodate the specific structural requirements.[2]

The basic mechanism of the DFN can be described as follows. The network follows an hierarchical structure and consists of three different layers: the **input layer**, which represents the input data, **the hidden layers**, connecting various inputs in a nested and hidden structure, and an **output layer**, reflecting the prediction target — which are ordered in the sequence just described. Figure 1 illustrates the basic structure and flow of a very small network. However, I use this simple example to provide more meaning to the generic function $f(x_{n,t};\theta)$ mentioned before. The Figure illustrates the dependence of x to h, the first hidden layer, which is in this case simply Wx. Typically, it reads as $h = g(b + W'x)$ with bias term b, a vector of length 2, weight matrix W, of dimension 2×2, and activation function $g(\cdot)$. Accordingly, this simple network function reads,

$$\hat{y} = b + w' g(\mathbf{b} + W'x).$$

Hence, the set of θ includes $W, \mathbf{b}, w, b$ and requires 9 model parameters to be estimated. Applying the same logic, wider and deeper models can be expressed in similar fashion, which then only results in a much more nested function $f(x_{n,t};\theta)$ with many more parameters to be estimated.

The activation function $g(\cdot)$ is applied element-wise and can take various forms. Common choices are the more traditional sigmoid activation ($\sigma(x) = \frac{1}{1+\exp(-x)}$) and the popular rectified linear unit. The rectified linear unit, $g(z) = max(0, z)$, has computational advantages over the other candidate functions and is, therefore, often the preferred choice for DFN designs, see Jarrett et al. (2009), Nair and Hinton (2010) and Glorot et al. (2011) for more details.

[2] For example, CNN would require a reasonable topology. However, it is not obvious how to structure the FC input data in comparison to image processing, where the ordering of pixels in 2D or 3D come naturally.

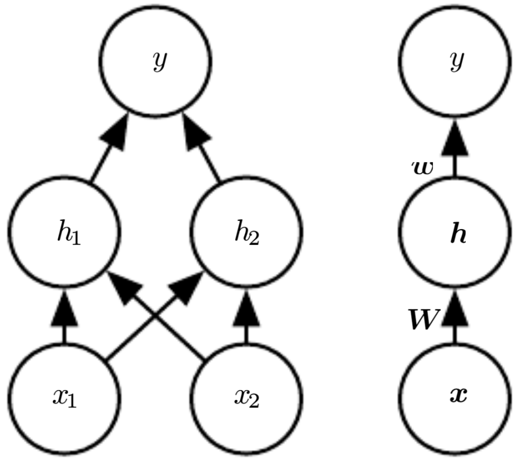


Figure 1: Computational Graph DFN: Base anatomy of a fully connected deep neural network with one hidden layer and two units. The input variables x_1 and x_2 mark the input layer, h_1 and h_2 and two hidden units in the only hidden layer, y marks the output. W and w defines the weight matrix. Illustration from Goodfellow et al. (2016), page 169.

3.2 Optimizing and training the network

3.2.1 Objective Function

A key task of deep learning is model training, which requires an objective function as a starting point. The ultimate objective of the researcher is to minimize a loss function evaluated with respect to the true DGP. Algebraically, it is expressed as follows,

$$J(\theta) = \mathbb{E}_{\mathbf{x},y\sim p_{\text{DGP}}} L(f(\mathbf{x},\theta);y) \tag{1}$$

However, practically the true DGP is mostly unknown, like in the application of this work. Consequently, the empirical counterpart becomes the actual problem at hand. According to equation 1 the empirical objective function can be written as,

$$\hat{J}(\theta) = \mathbb{E}_{\mathbf{x},y\sim p_{\mathbf{x},y}} L(f(\mathbf{x},\theta);y) = \frac{1}{TN}\sum_{t}^{T}\sum_{n}^{N} L(f(\mathbf{x}_{\mathbf{n},\mathbf{t}},\theta);y_{n,t+1}), \tag{2}$$

where I simply replace the underlying DGP by the empirical distribution. Furthermore, I solely deploy a standard mean squared error loss ($L(a,b) = (a-b)^2$) throughout this paper. This is in line with the literature, see for example Green et al. (2017) and Freyberger et al. (2017).

3.2.2 Stochastic-Gradient Descent

In contrast to standard convex optimization problems, DFN optimization is almost exclusively non-convex and, hence, builds heavily on numerical optimization routines. Stochastic-gradient descent (SGD) is the standard optimization principle underlying almost all neural network optimization routines. The idea behind gradient based approaches is to solely iterate towards a low value of the cost function, often the algorithms terminate prior to reaching the (local) minimum of the training cost function. The SGD algorithm requires mainly some initialization strategy for the set of parameters to be estimated. Additionally, exact gradient computations are available for wide and deep DFN choices based on back-propagation, which itself relies heavily on the application of the chain rule, which is required because of the nested functional structure introduced above. The basic algorithm is described in 1. However, many variants of this basic SGD exist, all addressing different problems arising of misbehaved gradient functions. Examples are Momentum based algorithms (Polyak (1964)), which consider not only the most recent gradient for updating the parameters but past gradient values, with a decaying weight function. Other strategies vary the learning rate adaptively with respect to the parameters, examples are, AdaGrad, Duchi et al. (2011), or Adam, Kinga and Adam (2015). Schaul et al. (2013) provide a framework of testing these optimization routines, however, no general statement can be made on which SGD variant is the most robust. More details on SGD and its variants can be found in Saad (1998) and Ruder (2016).

Algorithm 1 Stochastic Gradient Descent

$\hat{\theta} \leftarrow$ random draw ▷ Initialize parameters
$\{(y_{\text{valid}}, X_{\text{valid}}), (y_{\text{train}}, X_{\text{train}})\} \leftarrow \text{split}(X,\, y)$ ▷ Split data into training and validation set
$\kappa \leftarrow$ assign learning rate
for $i = 1$ to K **do** ▷ K max number of iterations
    $\hat{g} \leftarrow \frac{1}{J} \nabla_\theta \sum_j^J L(f(x_j;\theta), y_j)$ with $x_j \in X_{\text{train}}$ and $y_j \in y_{\text{train}}$ ▷ Gradient estimate
    $\hat{\theta} \leftarrow \hat{\theta} - \kappa \hat{g}$
return $\hat{\theta}$ ▷ The parameter set estimate

3.2.3 Mini-batch

Mini-batching is a data splitting strategy for gradient computations. Instead of calculating gradients for the entire training set, gradients are computed and parameters updated based on only a small subsample. This serves two purposes. First, computations are eased due to the much smaller data size and, second, according to Wilson and Martinez (2003), mini-batch training enjoys a regularization benefit. The study by Goyal et al. (2017) investigates the tradeoff between validation error and mini-batch size/computational efficiency, where they document a positive relationship between mini-batch size and validation error. Consequently, I carry out the optimization including mini-batching. [3]

[3] Mini-batch size becomes more critical from a computational perspective once GPUs are in use and should be configured according to the exact hardware specification.

3.3 Regularization

It is easy to imagine that a wide and deep enough DFN design can fit any finite training sample almost always perfectly. Consequently, the machine learning community has developed a range of tools, which aim to find an optimal balance between over- and under-fitting. Goodfellow et al. (2016) present several techniques, of which I utilize early stopping, dropout and norm penalties (L1 and L2), which are described in detail below. However, not all available tools are applicable in this context. For example, commonly used data-augmentation strategies cannot simply be adapted to financial data sets. For more details on regularization I refer the reader to Goodfellow et al. (2016).

3.3.1 Early stopping

Early stopping is an algorithm to reduce over-fitting. It prevents the training process of the network from purely memorizing the training data, by evaluating the validation error. Goodfellow et al. (2016), Bishop (1995b) and Sjöberg and Ljung (1995) show the regularization property of early stopping, which displays a similar character as L^2 regularization. It implicitly constrains the parameter space, due to the limited distance gradient-descent can travel during training. The algorithm terminates training the network after the validation error couldn't be improved for a certain amount of iterations. The number of iterations allowed, without seeing improvements, is called patience. As a side effect it limits the number of training iterations, and hence, increases computational efficiency.

3.3.2 Dropout

Srivastava et al. (2014) introduce Dropout regularization to the machine learning literature. The idea underlying the method is to randomly drop units during training. It can be interpreted as a form of bagging.[4] Figure 2 illustrates graphically the working mechanism applied to the basic network example illustrated in figure 1. Mechanically one can attach an independent Bernoulli random variable to each of the nodes with shared (Dropout) probability p - a hyper-parameter. The Bernoulli random variables are then drawn prior to each mini-batch process, hence, during neural network training a large number of subnetwork is actually optimized. More details can be found in Srivastava et al. (2014).

3.3.3 L1 and L2 Regularization

Norm regularization aim at variance reduction through explicit penalization of the size of the model parameters, which shrink towards zero, the heavier the penalization is pronounced. Specifically L1 and

[4] Bagging is an ensemble strategy which combines several predictions models to compute a weighted prediction output, profiting from variance reduction.

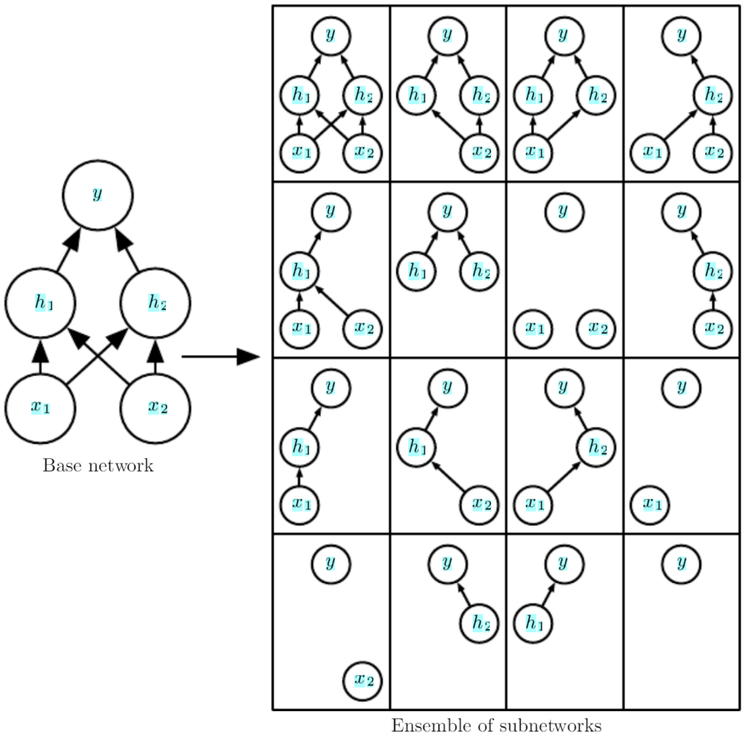


Figure 2: Dropout DFN: Principal working mechanism of dropout regulariation anatomy of a fully connected deep neural network with one hidden layer and two units. The input variables x_1 and x_2 mark the input layer, h_1 and h_2 and two hidden units in the only hidden layer, y marks the output. W and w defines the weight matrix. Illustration from Goodfellow et al. (2016), page 256.

L2 constraints can be imposed, which are known in the linear specification as Lasso, Tibshirani (1996), and Ridge Regression methods. These shrinkage approaches can simply be extended to neural networks in general and specifically to the optimization problem above by adding the corresponding terms to the cost function in equation 2,

$$\tilde{J}(\theta) = \hat{J}(\theta) + \alpha_1 \|\mathbf{W}\|_1 + \alpha_2 \|\mathbf{W}\|_2^2, \tag{3}$$

where $\|W\|_1 = \sum_j^J \|w_j\|$ and $\|W\|_2^2 = \sum_j^J w_j^2$ with J the total number of weight parameters. Note, the bias terms are usually not included into the penalization forms.

4 Data

4.1 Returns and FC construction

The cross sectional FC and returns used in this study are based on the CRSP/Computstat database. The FC construction and notation is identical as in Messmer and Audrino (2017) and, therefore, mainly consistent with the study of Green et al. (2017). Details on the FC construction can be found in the appendix. Like the aforementioned contributions, this work considers only CRSP stocks with share code

10 and 11 which are traded either on the NYSE, AMEX or NASDAQ. Additionally, stocks with missing market capitalization data and/or where book values are unavailable are dropped from the analysis. The alignment of accounting data is performed with a standard lag of six months of the fiscal year end date.[5] Return based FC, such as one-month momentum, beta, maximum return, or six-months momentum are used as of the end of the most recent month.[6] Handling extreme values is a sensitive procedure. Here the study deviates from Messmer and Audrino (2017). Extreme FC observations are controlled for by using relative cross-sectional ranks at each point in time instead of relying on winsorizing the extremes. In the next step, missing data are replaced by the median value of 0.5 at each point in time. Note that the return observations are not adjusted for any extreme values. However, returns are de-meaned for each period to preserve the cross-sectional information. The last step requires the pooling of FC and returns over time, such that the networks can be estimated with the stacked matrix $\mathbf{X}$ and the output return vector $\mathbf{y}$. As in Fama and French (1996), stocks qualify as large if their market capitalization ranks among the top 1000, accordingly mid cap stocks have to fall into the range of rank 1001-2000, and small caps comprise all stocks with rank > 2000.

4.2 Training, Validation and Test data

The trade-off between over-fitting vs. under-fitting is the major concern of training complex machine learning algorithms. The financial econometrics literature typically splits the data into an in-sample and out-of-sample data set to evaluate model performance. However, model complexity and capacity are not a key concern in this field, as the researcher typically has a prior belief of the intrinsic nature of the DGP, which is often derived on economic theory. Hence, model capacity restrictions are implicitly imposed, therefore, parameter tuning is redundant. Pure data learning approaches lack these prior model restrictions and hence, require a more careful approach to tackle model complexity explicitly. This is typically achieved by splitting the data-set into a training, validation and test set. The training set is used to train the model, the validation set is used to evaluate the estimated model on an independent data-set. Cross-validation (CV) is a commonly used specification of validation error evaluation, it provides an estimate of the expected prediction error. Moreover, CV is unbiased if the validation set is independent of the training set, see Bishop (1995a). Commonly used are k-fold CV, which splits the training data into k-equal sized parts. However, cross-validation is computationally costly, as it requires k-optimization routines on $\frac{k-1}{k}$ of the training data. Typically, stock returns are highly cross-sectionally correlated and if at all, only weakly correlated across time. Consequently, cross-validation is performed along the time

[5] For instance, the Compustat data of a firm with fiscal year end date 12/31 are aligned with data 06/30, predicting monthly returns from 6/30 to 7/31.

[6] For example, for the return prediction from from 6/30 to 7/31, the max daily return from the period 5/31-6/30 is used.

axis, such that each training and validation data set does not share any stocks from the same period. The training and validation data set spans the period from 1970-01 until 1981-12. It is exclusively used for model selection. As a result the test data set begins with the first month of 1982 and ends in December of 2014. Note that, I still use a training and validation strategy for estimation purposes.

5 Hyper-parameter optimization

Section 3 introduces important concepts and tools to minimize naive training. Most of these require explicit hyper-parameters which have to be supplied. Hyper-parameters are, for example, the number of hidden layers or the learning rate of the optimization routine. In particular, the hyper-parameter space is typically of high dimension and optimal hyper-parameter tuning an area of research of its own, see, for example, Bergstra and Bengio (2012) or Larochelle et al. (2007).

5.1 Hyper-parameter set

I can define the set of hyper-parameters, λ, which the researcher has to provide to obtain a particular instance of the aforementioned model design. Specifically, the complete set of parameters underlying this work looks as follows: *number of hidden layers (HL), number of units per layer (UN), dropout probability (Drop), L1 regularization strength (L1), L2 regularization strength (L2), the optimization routine (Opt), learning rate (LR), epoch bound (EB), patience (PA), mini-batch size (Mini), the k in K-Fold-CV* and finally, *the type of activation function.*

5.2 Outer optimization problem

A priori it is not known what the optimal combination of these hyper-parameters looks like. Hence, hyper-parameter selection can be seen as an outer optimization problem. Some variables can be fixed due to computational considerations, for others I need to provide bounds to keep the problem tractable. Specifically, choosing between five or ten-Fold CV, I use 5-Fold CV to estimate the validation error, mainly because it more than halves the computational costs. Furthermore, I set the activation function to the popular rectified linear unit (ReLu), motivated by computational advantages over alternative activation functions — due to its trivial gradient calculations. The mini-batch size is set to 128 — a value, which typically provides a good balance of robustness and efficiency. Finally, I set an epoche bound of 200, to limit the training time for each specification.[7] Hence, I am left to determine optimal values for the following set of hyper-parameters: $\lambda := \{\mathrm{HL}, \mathrm{UN}, \mathrm{Drop}, \mathrm{L1}, \mathrm{L2}, \mathrm{Opt}, \mathrm{LR}, \mathrm{PA}\}$. CV provides training ($y_{\text{train}}$,

[7] One epoche defines one complete optimization run through the entire training data set.

X_{train}) and validation (y_{valid}, X_{valid}) data to compute loss function values. Consequently, the optimization problem can be formulated as follows:

$$\lambda_{opt} = \underset{\lambda \in \Lambda}{\arg\min} \sum_{i=1}^{k} \mathcal{L}(y_{\text{valid},i}, X_{\text{valid},i}, \hat{\theta}_{X_{\text{train},i},\lambda}, \lambda). \tag{4}$$

where $\hat{\theta}_{X_{\text{train},\lambda,i}}$ are estimated model parameters from the i-th training data sample given the hyper-parameter λ, k refers to the number of CV folds. Hence, we evaluate a finite number of trials, t, to determine $\hat{\lambda}_{opt}$:

$$\hat{\lambda}_{opt} \equiv \underset{\lambda \in \{\lambda_1,\ldots,\lambda_t\}}{\arg\min} \sum_{i=1}^{k} \mathcal{L}(y_{\text{valid},i}, X_{\text{valid},i}, \hat{\theta}_{X_{\text{train},\lambda,i}}, \lambda).$$

5.3 Curse of Dimensionality

Optimizing equation (4) is a non-convex high-dimensional problem. Grid-search or brute-force is one possible option to solve the problem. However, it suffers from the curse of dimensionality, since the optimization problem of this application is carried out in eight dimensions. Assuming a lower bound of grid-points in each dimension of three (a relatively sparse grid), the routine would need to train $3^8 = 6561$ neural networks. Instead of performing an expensive grid-search, I utilize random search in the fashion of Bergstra and Bengio (2012) to determine an optimal set of hyper-parameters. However, Goodfellow et al. (2016) note that the learning rate is the decisive factor in many applications. Hence, I further tune the best network found in the random search along the learning rate dimension.

5.4 Random Search

Bergstra and Bengio (2012) show that random-search has advantages over grid-search in situations, where the effective-dimensionality of the problem is lower than implied by the length of the set of hyper-parameters. This implies that only a smaller subset of hyper-parameters is crucial of minimizing validation error. The intuition behind this is that grid search wastes resources moving along insensitive dimensions too often instead of varying all dimensions in each trial.

5.4.1 Specification

For each trial I draw a random set of hyper-parameters. I mainly follow Bergstra and Bengio (2012) and Goodfellow et al. (2016) by drawing the respective hyper-parameters independently from the following distribution — however, the following specification is slightly adjusted to reflect computational limitations at the application at hand, the largest model can, hence, be 14 layers deep and approximately 100 units wide:

- HL: random draw from integers ranging from 2 to 14.
- UN: $round(\exp(\eta))$, where $\eta \sim U(log(A), log(B))$ and A and B is set to 0.5 times and 1.5 times, respectively, the number of FC.
- Drop: $\sim U(0, 0.75)$.
- L1: 10^a where $a \sim U(-5, 0)$
- L2: 10^a where $a \sim U(-5, 0)$
- Opt: random draw form the following list of optimizers available in Abadi et al. (2015): 'RMSprop', 'Nadam', 'Adagrad', 'Adadelta', 'Adam', 'Adamax' and 'Nesterov'.
- LR: 10^a where $a \sim U(-5, -1)$
- PA: random draw from integers ranging from 20 to 40.

5.4.2 Empirical results

The random search conducted in this study includes three different data specifications: large and mid cap separately and the joint set of large and mid caps. For each data set I run 200 trials for a period of 12 years, using data from 1/1/1970 until 12/31/1981. Applying five fold cross-validation leads to 9.6 years of monthly training data and 2.4 years of validation data. This specification requires the training of 3000 networks. For each trial I calculate the loss function as described in Equation 4. In order to ensure comparability across the trials and networks, the same training and validation data splitting is used in each trial. The motivation behind using five fold cross-validation is the intrinsic random character of the validation score. Five-fold cross-validation provides five (cross-sectionally) independent estimates, and hence, helps to reduce the variance of the estimate. Consequently, time dependencies are neglected.
Table 1 shows the random search results for the joint set of large and mid cap stocks. The search identifies 8 networks, which achieve higher R^2's evaluated with the validation sample. The best network design reaches a value of 7.31% vs. 4.5% with the linear specification. The architecture of the best performing network for large cap stocks has 62 hidden units, 3 layers, and a learning rate of 0.02×10^{-3}; for more details, I refer to Table 2. Hence, this network is not deep. However, the annualized R^2 of 5.06% shows an improvement compared to the linear benchmark of 4.34%. However, only the two best performing designs achieve lower objective function values in comparison to the linear model. The optimal network discovered for mid cap stocks comprises a deeper architecture, with 53 hidden units and 8 layers as Table 3 reveals. Furthermore, and not surprisingly, the overall R^2's are slightly more elevated compared to the

Rank	Rˆ2 DFN	Rˆ2 Lin	LR	L2	L1	dropout	UN	HL	Opt	Patience
1	7.31	4.50	0.01	3.14	0.05	0.02	78	7	Nadam	35
2	6.80	4.50	0.01	0.07	0.04	0.07	51	14	Nadam	33
3	6.30	4.50	0.02	158.76	0.01	0.06	67	2	RMSprop	20
4	6.00	4.50	0.28	44.44	0.04	0.15	46	3	Adam	25
5	5.51	4.50	1.22	0.51	5.22	0.09	91	12	Adamax	30
6	5.32	4.50	1.02	37.81	0.05	0.07	61	5	Adamax	38
7	4.78	4.50	0.02	61.17	1.15	0.01	45	8	Adam	34
8	4.76	4.50	0.24	0.33	0.18	0.29	47	2	Adam	33
9	4.33	4.50	0.01	0.02	0.21	0.26	76	6	Nadam	36
10	4.24	4.50	0.13	3.31	0.88	0.27	47	5	Adam	29

Table 1: The Table displays the random search results for **Large + Mid Cap Stocks:**. It shows the top ten results of the 200 trials performed, sorted by R^2s. The R^2s shown are annualized based on monthly observations. R^2 are given %-points. The columns "LR", "L2", and "L1" are scaled by a factor of 1000. In total 89 out of 200 models show an R^2 below zero, moreover, 61 out of 200 models fail to provide any numerical results.

Rank	Rˆ2 DNN	Rˆ2 Lin	LR	L2	L1	dropout	UN	HL	Opt	Patience
1	5.06	4.34	0.02	0.06	0.26	0.18	62	3	Nadam	37
2	4.39	4.34	4.21	0.05	0.04	0.26	78	2	Adagrad	32
3	3.67	4.34	0.02	41.65	0.18	0.06	33	10	RMSprop	22
4	3.38	4.34	0.03	0.24	2.97	0.07	82	13	Adamax	34
5	3.20	4.34	2.45	138.46	0.02	0.24	57	4	Adamax	38
6	3.19	4.34	0.04	0.57	7.28	0.01	38	5	Nadam	24
7	3.05	4.34	0.67	0.01	0.17	0.37	32	2	Adamax	27
8	3.01	4.34	0.09	258.04	22.41	0.08	37	4	Adam	36
9	2.88	4.34	0.73	0.02	0.01	0.27	57	7	Adamax	28
10	2.57	4.34	40.27	94.13	0.83	0.11	48	9	Adadelta	40

Table 2: The Table displays the random search results for **Large Cap Stocks:**. It shows the top ten results of the 200 trials performed, sorted by R^2s. The R^2s shown are annualized based on monthly observations. R^2 are given %-points. The columns "LR", "L2", and "L1" are scaled by a factor of 1000. In total 75 out of 200 models show an R^2 below zero, moreover, 83 out of 200 models fail to provide any numerical results.

case of large cap stocks. The best DNN model outperforms with an R^2 of 6.18% vs. the linear hurdle of 4.71%. Furthermore, a total of 9 out of 200 models indicate an edge over the parsimonious alternative. Evidently, training DNNs is numerically a tedious challenge as in both cases a majority of the models produce either below zero R^2's or no numerical results at all. It is technically relatively difficult to understand the source of these errors, potential sources are poor starting values or inappropriate learning-rates. A more detailed analysis on this behavior is beyond the scope of this work.

5.5 Learning Rate Tuning

Goodfellow et al. (2016) underscore the sensitivity of DNN optimization with respect to the learning rate. I use the optimal hyper-parameters from the random search conducted in subsection 5.4.2 and define a

Rank	R^2 DNN	R^2 Lin	LR	L2	L1	dropout	UN	HL	Opt	Patience
1	6.18	4.71	0.08	0.22	1.91	0.11	53	8	Adam	22
2	5.42	4.71	1.55	0.06	0.04	0.18	46	5	Adamax	36
3	5.36	4.71	0.03	0.09	0.05	0.18	86	10	Nadam	33
4	5.36	4.71	3.51	1.38	0.02	0.17	70	2	Adam	33
5	5.25	4.71	3.50	0.38	18.29	0.17	66	2	Adamax	25
6	5.01	4.71	3.50	0.38	18.29	0.17	66	2	Adamax	25
7	4.96	4.71	0.06	0.11	0.05	0.19	77	3	RMSprop	30
8	4.92	4.71	0.09	0.24	0.82	0.30	55	2	Adamax	34
9	4.76	4.71	9.21	0.45	3.06	0.15	67	6	Adamax	30
10	4.65	4.71	1.68	0.74	0.43	0.25	80	4	Adagrad	28

Table 3: The Table displays the random search results for **Mid Cap Stocks:**. It shows the top ten results of the 200 trials performed, sorted by R^2s. The R^2s shown are annualized based on monthly observations. R^2 are given %-points. The columns "LR", "L2", and "L1" are scaled by a factor of 1000. In total 92 out of 200 models show an R^2 below zero, moreover, 75 out of 200 models fail to provide any numerical results.

grid along the learning rate dimension to further optimize the given network architectures. Precisely, I use 20 equally spaced points (log-based) and optimize the model accordingly. Figure 3 shows the results, which confirm a strong sensitivity of learning rate variation for the problem at hand. Despite different network structures the optimal learning rate is in a similar region for both large and mid cap stocks. The annualized R^2 improves for the large cap sample around 17% (5.06% vs 5.94%), compared to the optimization result without learning-rate tuning. The gain in case for mid cap stocks is only moderate with an increase of roughly 1% (6.18% vs 6.25%).

5.6 Network Details

Random search has shown that many architectures suffer from numerical failure, hence, I investigate briefly the behavior of the optimal network selected. In this case, a numerical failure is defined when the optimization algorithm returns errors or extreme and error-like numerical predictions. First, Figure 4a presents the development of the loss function evaluated at each iteration step for both the validation and the training data sets. It behave as expected, with an initial decline in both function values, followed by a slow increase of the validation score, which causes the optimization to terminate after 97 iterations. Moreover, figure 4b shows a smooth distribution of the predicted cross-sectional returns after the final step of the optimization, which looks from a numerical point of view unsuspicious. Initially, the output concentrates a large mass around zero. Over time the output distribution becomes wider and more smooth. Towards the end of the optimization procedure the predictions center more around 3, however, with a relative heavy left tail. Ill-behaved networks often show visually odd output distributions, for example, discontinued densities at multiple points or an extreme centering with hardly any variation at

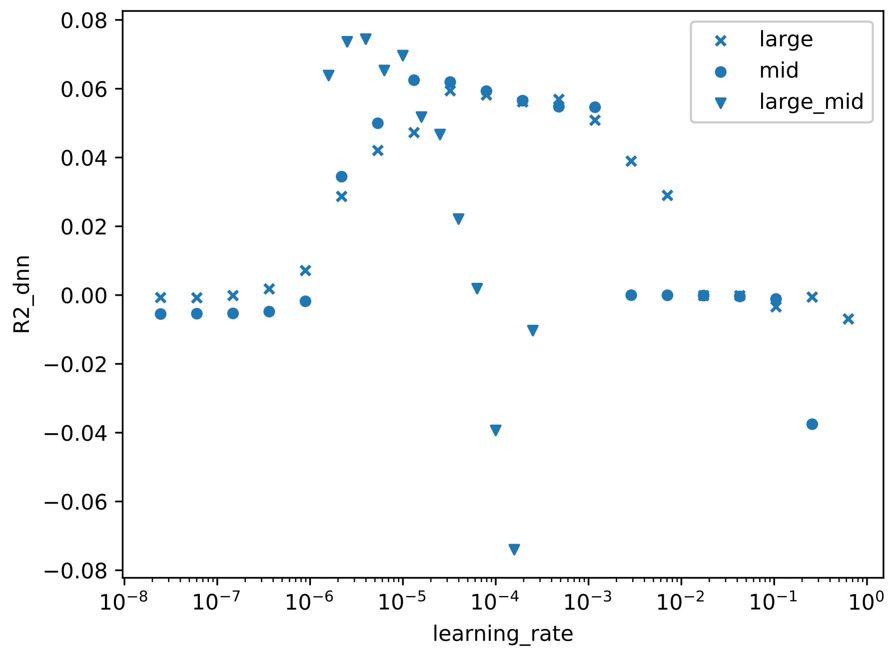
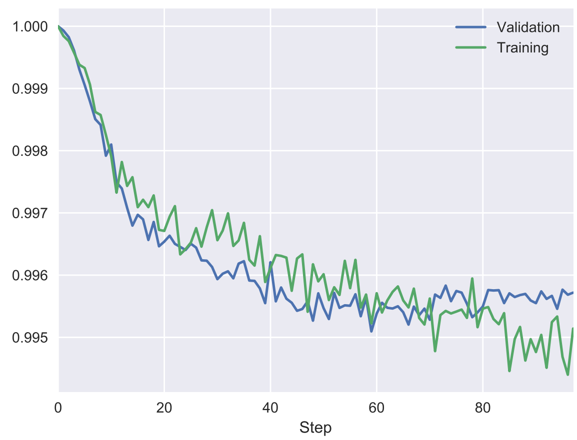
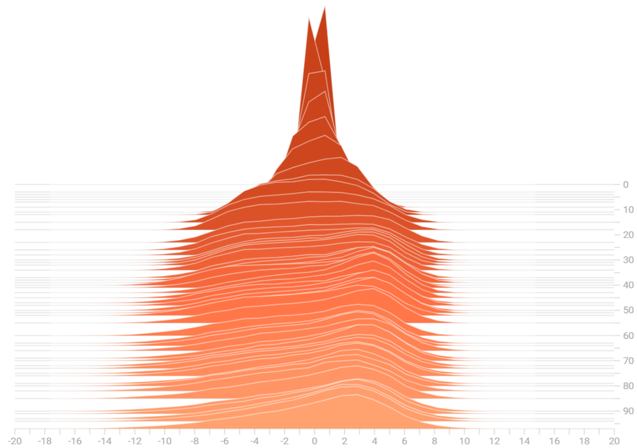


Figure 3: The figure shows the effect of learning-rate variation for DNN optimization. The x-axis captures the variation in the learning rate, the y-axis displays the resulting (annualized) R^2s. Strikingly, the variation is quite strong. Learning rates outside the range of 10^{-6}-10^{-2} produce mainly useless estimates. The optimal learning rate for large caps is roughly $10^{-4.5}$ and for mid caps $10^{-4.9}$. The maximum R^2 values are 5.94 % and 6.25% for the large and mid cap samples.



(a) Validation loss



(b) Output density function

Figure 4: Training Graphs: Figure 4a shows the training and validation loss as a function of the iteration steps, based on a random instance of the optimal DNN determined in 5.5. The loss functions behave as expected, as the training loss is mostly declining from left to right, whereas the validation score plateaus after around 45 steps (with a minimum at step 59) before it rises slightly towards the end. The optimization terminates after 97 steps as it reaches the maximum patience. Figure 4b depicts the histograms of the output values (cross-sectional returns) of the training data over all 97 iteration steps, based on the same instance of the optimal DNN determined in 4a. The distribution in the very front represents the final output prediction, the very first reflects the output values after the first iteration is completed. Note, the Graph is taken from Abadi et al. (2015) visualization tool "Tensorboard".

a specific point.

6 DFN Portfolios

This section analyzes portfolio returns constructed based on deep-learning networks. It contains two different construction approaches, where each of them deals differently with the estimation instability of neural networks. The first subsection describes the estimation and portfolio construction mechanics. The second part contains a standard performance analysis of the portfolio returns, including an analysis conditioning on volatility and trading costs. Finally, I discuss the crucial FC of the prediction outcomes.

6.1 Estimation and Portfolio Construction

Table 2 and 3 reveal that training deep neural networks on cross-sectional returns is numerically not stable in this specific application. Hence, the researcher, or the investor, has to explicitly address the issues arising from it. I follow two different approaches. One way, which I call the **single prediction approach**, simply trains one network and continues with the return prediction and portfolios construction. Alternatively, a **forecast combination approach** can be applied with the objective of variance reduction at the prediction stage. Model estimation uncertainty, despite identical network hyper-parameters, arises because of variation in the training and validation data sets, but also because of the randomness in starting values of the numerical optimization.

The starting point for both portfolio approaches is the optimal network design found in the learning rate tuning in section 5.5. I follow an expanding window approach starting initially with twelve years of data (1/1/1970-12/31/1981). I split the data randomly along time points in a training (4/5) and a validation (1/5) data set and optimize the network by evaluating once more the validation results. I use this parameters for the network for the next twelve months for predicting one-month ahead returns, before re-estimating the same model with additional observations. The estimation frequency of twelve months is the result of the available computational resources. However, return predictions and portfolio weights are calculated on a monthly base by updating the FC. For example, the first prediction is obtained for 1/31/1982 as of 12/31/1981, using the trained network and FC input from 12/31/1981. The prediction for the other 11 months of 1982 all use the same network, with monthly FC input updates. Accordingly, the next network optimization expands the data window by one year, re-trains the network, and provides the first prediction for the first month in the next year. As a result, the predictions are based on past information only. Motivated by the estimation uncertainty, for the reasons mentioned above, this prediction exercise is repeated multiple times with the same network ingredients. It is to expect that each portfolio path realizes a different time-series of long-short returns. Ultimately, this leads to several predictions for each single excess return.[8] These predictions serve two purposes. First, it allows the investigation of the severity of the model uncertainty. Second, the predictions can be recycled for the alternative prediction exercise.

The two approaches differ at the prediction stage. The construction of the **single** prediction based portfolio simply uses the predicted return spreads and goes on with the portfolio formation. In contrast to the latter, the **combination** approach combines several single prediction outcomes. Given the 150 return predictions from the estimation and prediction exercise, one can form a forecast combination prediction straightforwardly. Forecast combination is popular technique, which is often successfully applied in the machine learning literature (Szegedy et al. (2013)), but it has also been employed to some classical financial prediction problems, see Rapach et al. (2010) for more details.[9] Primarily, it aims at variance reduction and is in the basic form an equal weighted average of several prediction models. The idea can be related to portfolio diversification. Most prediction contains some idiosyncratic noise component, which is diversified away through averaging, like the idiosyncratic volatility of stocks in a diversified stock market index. The

[8] I set the number to 150 for the large and mid cap stock sample. These limits are motivated by the computational resources available.

[9] Typically, forecast combinations pools predictions based over different models. The interpretation here is that each different prediction is a result of the same model with different parameters, which can be seen as ensemble of different models.

common systematic prediction component shared among all single predictions remains. Hence, I construct once more 150 portfolios, each based on an average of, for example, 75 randomly drawn predictions from the previous 150 series of naive $\hat{\mathbf{y}}$'s. To be precise the prediction for a particular stock n at time t used to form one combination portfolio is,

$$\hat{y}_{n,t}^{\text{comb}} = \frac{1}{75} \sum_{i=1}^{75} \hat{y}_{i,n,t}^{*\text{single}}$$

where $\hat{y}_{i,n,t}^{*\text{single}} \in \mathcal{Y}$, whose 75 elements are randomly drawn (with replacement) from the set of the 150 single predictions obtained before.

The predictions of the linear benchmark are based on Fama and MacBeth (1973) regressions, where the variable selection is performed as in Green et al. (2017) (without correcting for a family-wise error). The re-estimation window and prediction intervals are identical as for the DFN approach.

Once the return predictions are collected, I form value-weighted (VW) and equal-weighted (EW) portfolios and calculate standard performance measures without considering any trading costs at first. I use a decile style cutoff procedure, by going long the upper 10% and short the lower 10% of the predicted return series.[10]

6.2 Performance Analysis

Table 4 shows the long-short portfolio results for **large cap stocks**. Strikingly, the 150 single prediction approach portfolios realize a wide range of portfolio outcomes, as a result of the expected estimation uncertainty. Sharpe ratio (SR) differences between the "Min" and "Max" portfolio are around 0.5 ($\approx$ 100%) for both the value-weighted and equally-weighted portfolio. The median portfolio realizes a higher SR compared to the linear model in both cases, however, the differences are statistically not significant. Finally, the Table includes two α estimates, one estimated with respect to the Fama and French (2014) 5-factor model and the other with a version extended by the classical momentum factor. Except of one case, all DFN portfolio paths realize higher and significant α's compared to the linear benchmark. Despite the large performance variation, the worst performing portfolio over the sample period still competes reasonably well, compared to the linear benchmark based on the α's measures. Furthermore, you can see in Table 4 that the VW portfolios perform slightly better compared to the EW version, which holds for both the DFN and the linear model.

Furthermore, Figure 14 illustrates the corresponding return indexes graphically. It emphasizes the economic relevance of the estimation uncertainty, as you can observe by the spread of the "Min" and the "Max" return index values. The linear as well as the mean DFN portfolio realizes gains mostly prior to the year 2004. Furthermore, the Figure visualizes that the outperformance relative to the linear model

[10] The appendix includes the results using a 30-70 or a Fama and French (1993) style (FF-style) cutoff procedure.

does not stem from a distinct period or event but rather accumulates gradually over time over the first two thirds of the sample period. In the last years from the beginning of 2004 to the end of 2014 both strategies flatten out and realize only marginal gains, partly a result of the losses suffered during the financial crisis.

One obvious problem associated with the DFN portfolios is the high uncertainty with respect to the potential return realizations. Most investors would prefer a less volatile return distribution, and would rather not be exposed to economically important risk arising from model optimization instability. The combination approach mitigates much of this risk, as you can see in the lower half of Table 4. As desired, one can observe a less volatile performance behavior for both VW and EW, simply measured by the difference of the "Min" and the "Max" realizations. Clearly, the objective of a variance reduction is achieved. As a result, the median portfolio shows a higher mean return, which results in a higher SR. The factor α's are on a similar level compared to the single predictions used before. Irrespective of the improvement, the Ledoit and Wolf (2008) test does not reject the null hypothesis of no difference in SR for the VW portfolio vs. the linear model. On the other hand, all Factor model α's associated with the DFN approach dominate the linear benchmark.

For both approaches, you can observe that a large fraction of the five-factor α's are absorbed by momentum exposure, as the difference between the regression α's is relatively high. Despite this difference, the momentum adjusted DFN α's are still mostly significant at the 1% level.

The same exercise can be repeated for **mid cap stocks**. The findings are documented in Table 5. Overall the performance statistics of both the linear and the DFN portfolio look impressive. Differences between VW and EW are small, with slight advantages for the EW schema. The single based predictions consistently realize higher SR's compared to the benchmark. These differences are, except in case of the min DFN portfolio realizations, significant. The higher SR's are a consequence of higher means, as the volatility measures are comparable. Most DFN portfolios generate higher factor α's than the benchmark, once more the min portfolios fall short of the linear model. Forecast combination portfolios master the linear hurdle in all cases, as we can observe in Table 5. All SR differences are significantly favoring the DFN approach at the 1% level in case of the max and median portfolios and at the 5% level worst performing DFN portfolio.

6.3 Volatility Regimes

The DFN portfolios seem to be prone to losses during volatile periods as Figures 14 and 6 indicate graphically. A more informative measure than visual inspection is the analysis of returns conditioning on the volatility state. For this purpose, I split the sample into two parts — a low- and high-volatility regime.

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	12.47	18.20	0.69	15.0***(4.81)	11.2***(4.03)	0.0788
Min			8.45	18.12	0.47	10.5***(3.26)	7.2** (2.26)	-0.64
Max			17.28	17.99	0.96	19.0***(5.69)	15.0***(5.35)	0.987
Linear			9.74	14.72	0.66	4.9* (1.93)	3.3 (1.28)	
Median	EW	single	12.12	16.39	0.74	13.8***(4.84)	9.8***(3.95)	0.48
Min			8.27	18.12	0.46	7.2** (2.03)	4.1 (1.19)	-0.441
Max			16.42	16.77	0.98	18.0***(6.10)	13.8***(5.88)	1.12
Linear			9.07	15.60	0.58	3.9 (1.54)	2.3 (0.89)	
Median	VW	Comb 75	13.59	18.82	0.72	14.6***(4.72)	10.7***(3.54)	0.212
Min			12.38	18.74	0.66	12.9***(4.11)	8.8***(3.10)	-0.00352
Max			14.57	19.34	0.75	15.3***(4.66)	11.0***(3.75)	0.321
Linear			9.74	14.72	0.66	4.9* (1.93)	3.3 (1.28)	
Median	EW	Comb 75	13.03	17.34	0.75	12.2***(4.05)	8.0***(2.96)	0.538
Min			12.50	17.51	0.71	12.3***(4.10)	8.0***(2.97)	0.415
Max			13.57	17.37	0.78	13.0***(4.42)	8.8***(3.39)	0.626
Linear			9.07	15.60	0.58	3.9 (1.54)	2.3 (0.89)	

Table 4: Naive Portfolio, Large Cap Stocks, decile Cutoffs: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). Mean, standard deviations(std) and factor α's are displayed annualized. The Sharpe ratio (SR) is expressed in monthly figures. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

The volatility classification into high and low is based on a standard GARCH(1,1) volatility estimation. Hansen and Lunde (2005) show that the parsimonious GARCH(1,1) is a reasonable choice to proxy the latent volatility process when compared to more sophisticated approaches. The low-volatility regime are all months which the volatility estimate belongs to the lowest 85% of the estimates, the high-volatility regime is then defined by the remaining months. We can see the conditional performance measures in Table 6. It documents large variations in average returns for the DFN strategy for both large and mid cap stocks, which perform better during less volatile times. For example, the difference in mean of the median portfolio for large cap stocks is 29.8 in annualized terms! The opposite holds for the linear approach, as the mean is higher during more uncertain periods. As a result, both strategies have a much different character during episodes of high volatility. This is reflected in the SR's, which are higher (lower) in the low (high) volatility state for the DFN strategy compared to the linear model.

Moreover, despite the negative mean, a large α is measured in case of the momentum augmented five-factor model in the high volatility regime. As pure momentum-strategies are known to be prone to crashes, see for example Daniel and Moskowitz (2016), multi-signal based long-short portfolios load

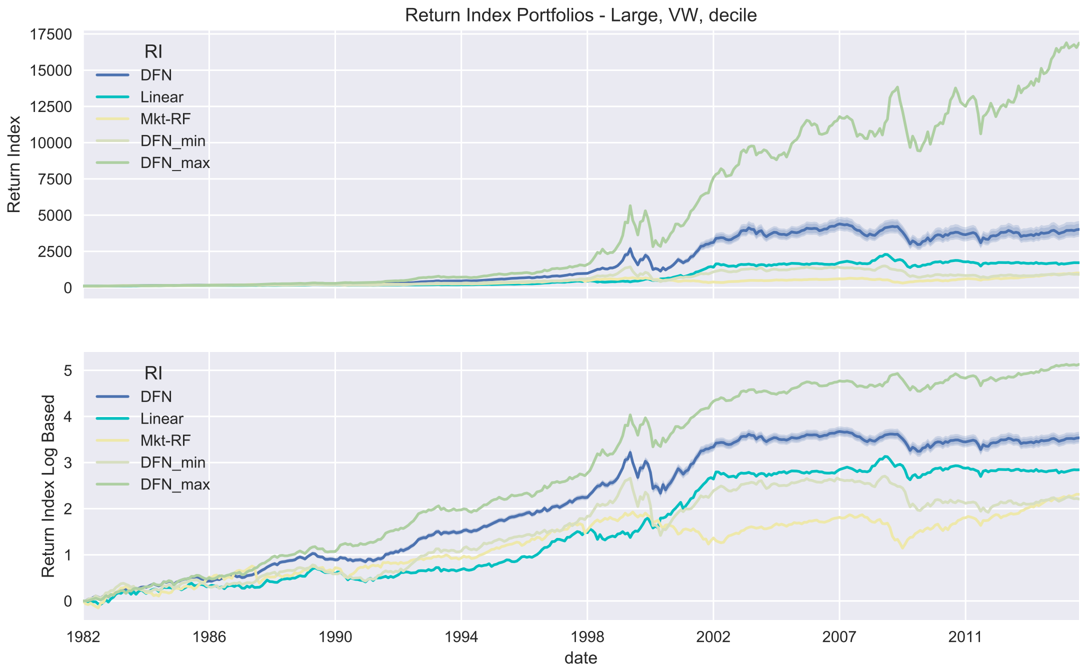


Figure 5: The figure exhibits selected paths of the 150 portfolios using large cap stocks with value weighted decile long-short portfolios for the years 1982-2014. "DFN" reflects the average of the 150 portfolio paths, which is embedded into the corresponding 1% and 10%-confidence bounds. The difference between the best and worst performing DFN portfolio reflects the uncertainty arising from model estimation.

momentum exposure but indicate more resilience to momentum drawdowns.

6.4 Trading Costs

The previous section shows attractive return characteristics of both strategies considered. Naturally, the question arises how much do the performance measures deteriorate once trading costs are accounted for. Hence, I use the same portfolios obtained in the previous section, but explicitly correct them for trading costs. The definition of the trading cost function is as in Brandt et al. (2009), Hand and Green (2011) and DeMiguel et al. (2017). The approach explicitly factors in the stock size (decreasing in size) and through time (decreasing from year to year). The function is defined as follows:

$$c_{n,t} = \underbrace{\left(1+3\frac{\max\left(\Delta_{\text{days}}(date(t), 1/1/2002), 0\right)}{\Delta_{\text{days}}(1/1/1974, 1/1/2002)}\right)}_{\text{time effect}} \underbrace{(0.006 - 0.0025 \times (1 - z(\text{ME}_{n,t})))}_{\text{size effect}}, \tag{5}$$

where $\Delta_{\text{days}}(a,b)$ counts the difference in days (date b minus date a), and $z(\text{ME}_{n,t}))$ measures the normalized market capitalization of stock n at time t.[11] Hence, the highest trading cost in the analysis is

[11] Large cap stock and mid cap stock market capitalization are always calculated w.r.t to the entire cross-section.

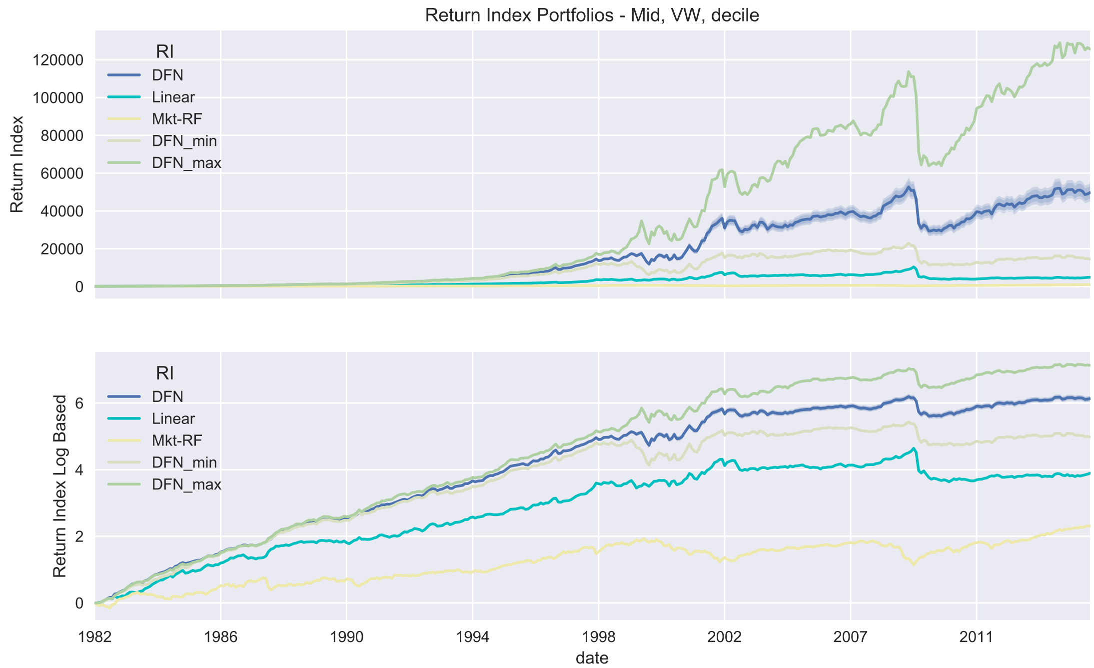


Figure 6: The figure exhibits the selected paths of 150 estimated portfolios using mid cap stocks for the years 1982-2014, using value weighted portfolios. "DFN" reflects the average of the 150 portfolio paths, which is embedded into the corresponding 1% and 10%-confidence bounds.

observed for the smallest stock at the beginning of our portfolio formation period at 1982/1/1 of around 190 basis points (BP) and 35 BP for the largest stock post the time trend end date on 1/1/2002.

Table 7 shows the adjusted performance characteristics. Not surprisingly, given a monthly average turnover (TO) of 218% and 324%, respectively. Consequently, non of the two strategies realizes positive mean returns when portfolio weights are updated monthly and trading costs are accounted for (the TO impact for the single strategy is very close to the combination approach, hence, they are not reported). All realize a TO of significant magnitude, notable, the linear model realizes the highest TO for all frequencies considered. A trivial TO reduction measure is to reduce the trading frequency. Hence, I consider several rebalancing policies, varying the trading activity monthly ranging from one to six months. This trade-off is reported in table 7. For example, the TO decreases around 45-50% for the two portfolios when switching from a monthly rebalancing to only one every 2nd month. Despite an almost monotonic decline for both strategies the mean returns do not fall dramatically, hence, reducing the rebalancing frequencies offers a more attractive cost-return balance. In this example, the maximum cost adjusted return for the DFN long-short portfolio is achieved with a frequency of five months, and six in case of the linear model. However, the linear model never achieves a positive mean return net of costs.

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	20.01	16.67	1.20	16.9***(6.28)	12.8***(4.57)	1.89*
Min			16.72	17.16	0.97	12.9***(4.41)	8.7***(3.10)	0.752
Max			23.30	16.81	1.39	23.1***(7.58)	18.4***(6.78)	2.48**
Linear			13.43	16.84	0.80	15.6***(6.64)	12.0***(5.18)	
Median	EW	single	20.26	16.93	1.20	18.9***(6.77)	14.3***(5.74)	2.14**
Min			17.23	16.49	1.04	16.2***(5.70)	12.3***(4.53)	1.19
Max			23.37	16.33	1.43	22.0***(7.54)	17.4***(7.42)	2.82***
Linear			13.47	17.69	0.76	16.2***(6.72)	12.5***(5.54)	
Median	VW	Comb 75	22.65	16.65	1.36	19.7***(7.08)	15.2***(5.87)	2.62***
Min			21.93	16.65	1.32	18.6***(6.75)	14.2***(5.53)	2.34**
Max			23.25	16.42	1.42	20.1***(7.35)	15.7***(6.21)	2.89***
Linear			13.43	16.84	0.80	15.6***(6.64)	12.0***(5.18)	
Median	EW	Comb 75	22.57	16.62	1.36	19.9***(7.11)	15.7***(6.20)	2.71***
Min			22.09	16.84	1.31	19.4***(6.82)	15.0***(5.91)	2.54**
Max			23.03	16.51	1.39	20.8***(7.46)	16.6***(6.54)	2.8***
Linear			13.47	17.69	0.76	16.2***(6.72)	12.5***(5.54)	

Table 5: Naive Portfolio, Mid Cap Stocks, decile Cutoffs: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). All performance measures are displayed annualized. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

Size	Volatility	Portfolio	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
large	high	DFN	-11.64	31.25	-0.37	1.2 (0.11)	28.9***(3.18)	-1.5
		Linear	15.15	22.57	0.67	12.6 (1.07)	29.7***(3.30)	
	low	DFN	18.22	15.56	1.17	16.2***(4.65)	6.7** (2.38)	1.6
		Linear	8.77	12.84	0.68	2.8 (1.01)	-0.7 (-0.26)	
mid	high	DFN	3.33	26.23	0.13	-8.8 (-0.64)	23.2***(3.31)	-0.784
		Linear	16.67	29.57	0.56	-5.5 (-0.38)	22.2***(3.02)	
	low	DFN	26.06	14.13	1.84	23.2***(7.59)	14.2***(5.92)	4.05***
		Linear	12.85	13.42	0.96	16.8***(6.57)	9.8***(4.89)	

Table 6: Volatility Regimes, Decile Cutoffs, VW, Sample Period 1982-01 until 2014-12: The table depicts performance measures conditional on the market volatility regime. The regime "high" are all months in which the GARCH(1,1) volatility estimate belongs to the 15% highest estimates, "low" accordingly to the lowest 85%. The DFN portfolio refers to the median forecast combination portfolio ("Comb 75"). All performance measures are displayed in annual terms. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark.

Freq	Model	Costs	Mean	Std	SR	TO-%	FF-5 α	FF-5 + Mom α	SR Test
1	Comb	No	12.48	18.20	0.69		15.0***(4.81)	11.2***(4.04)	0.0794
		Yes	-12.73	17.93	-0.71	218	-10.2***(-3.31)	-13.8***(-5.12)	3.13***
	Linear	No	9.74	14.72	0.66		4.9* (1.93)	3.3 (1.28)	
		Yes	-26.71	14.81	-1.80	324	-31.0***(-11.54)	-32.6***(-12.31)	
2	Comb	No	12.44	17.92	0.69		14.0***(4.52)	10.3***(3.80)	0.167
		Yes	-1.69	18.08	-0.09	124	0.1 (0.04)	-3.4 (-1.29)	1.52
	Linear	No	9.73	15.31	0.64		3.4 (1.17)	2.9 (1.06)	
		Yes	-8.87	16.00	-0.55	165	-14.6***(-4.95)	-15.0***(-5.37)	
3	Comb	No	10.77	18.79	0.57		14.4***(4.48)	10.8***(3.59)	0.285
		Yes	0.70	19.53	0.04	90	4.4 (1.34)	0.6 (0.21)	1.12
	Linear	No	7.15	15.04	0.48		2.0 (0.72)	2.1 (0.71)	
		Yes	-5.21	15.96	-0.33	111	-10.1***(-3.46)	-10.2***(-3.31)	
4	Comb	No	9.42	18.69	0.50		13.0***(4.34)	9.4***(3.45)	0.506
		Yes	1.10	18.97	0.06	74	4.6 (1.52)	1.2 (0.44)	1.1
	Linear	No	5.03	14.92	0.34		1.5 (0.53)	1.5 (0.53)	
		Yes	-4.36	15.55	-0.28	84	-8.0***(-2.87)	-7.8***(-2.78)	
5	Comb	No	9.33	17.61	0.53		13.8***(4.42)	11.2***(3.69)	0.491
		Yes	2.60	18.27	0.14	61	6.8** (2.11)	4.1 (1.30)	0.825
	Linear	No	5.72	15.43	0.37		4.4 (1.40)	4.5 (1.30)	
		Yes	-1.73	16.13	-0.11	68	-3.3 (-1.01)	-3.4 (-0.95)	
6	Comb	No	7.92	19.80	0.40		11.9***(3.64)	9.0***(2.62)	0.0143
		Yes	2.13	20.16	0.11	52	6.2* (1.85)	3.2 (0.91)	0.428
	Linear	No	5.66	14.34	0.39		1.6 (0.64)	1.9 (0.72)	
		Yes	-0.70	15.02	-0.05	57	-4.6* (-1.76)	-4.4* (-1.67)	

Table 7: Large Cap Stocks, Decile Cutoffs, VW Weighted: The table shows the performance impact once trading costs are explicitly accounted for. The column "Costs" compares the results with (Yes) and without (No) trading costs. Trading costs are calculated as in Brandt et al. (2009). TO-% indicates the average of the monthly portfolio turnover (the upper bound of a long-short portfolio is 400%). The column "Freq" refers to the rebalancing frequency in months.

6.5 Which FC drive the prediction?

Another important aspect is to understand which FC are the fundamental drivers behind the predictions. It is intrinsically hard to assess the relationship between a FC and and the predicted returns as the DFN has no clear interpretation available, comparable to a regression coefficient in the standard linear model. The output, a result of a nested structure, can theoretically be traced back to the input level, however, it is a rather challenging endeavor. Alternatively, I can manipulate the input matrix. Consequently, I separately replace each FC input vector with zeros and measure the changes compared to the predicted output when all FC are included. The measure I construct is a mean-squared deviation (MSD). Precisely, the MSD is calculated as follows:

$$\mathrm{MSD}_{t,k,p} = \frac{1}{N}\sum_{n=1}^{N}(\hat{f}_p(\mathbf{x}_{t,n}) - \hat{f}_p(\mathbf{x}_{t,n})|_{x_{t,n,k}=0})^2 \tag{6}$$

where $\hat{f}_p$ is the function estimate of run p. Alternatives, like a sub-sample analysis is computationally prohibitive but would in general be preferable. Hence, the MSD approach allows model insights at almost zero computational costs.[12]

Table 8 displays the average MSD for the top 10 FC (from highest to lowest) for the three samples of interest.[13] The two predominant FC for both large and mid cap stocks are the short-term reversal and the twelve months momentum. Moreover, price based FC have by far the largest impact on the prediction outcomes, as for large and mid cap stocks the top five and the top eight FC, respectively, are purely reflecting return information. Figures 12 and 13 exhibit this behavior over time. Strikingly, the pictures are dominated by only very few FC over time, indicating that predictions are insensitive to the majority of FC included in this work. Notable, short-term reversal becomes slightly less relevant in case of large cap stocks towards the end of the sample period, the dominating role is taken over by the twelve months momentum. Another interesting observation is the impact of the stock market correction around the beginning of the 2000s, which leads to a short distortion of the FC sensitivities. Finally, towards the end of the sample period a slight increase of other FC can be documented, as the colors partially darken in the upper half of the FC.

	Size	1	2	3	4	5	6	7	8	9	10
FC	l+m	mom1m	mom12m	chg_mom6m	retvol	maxret	rd_mve	mom6m	ep	idiovol	dolvol
MSD		67.70	44.21	24.07	19.81	14.20	14.08	13.94	13.01	12.18	10.19
FC	large	mom1m	mom12m	chg_mom6m	mom6m	retvol	ep	chpm_ia	mom36m	maxret	cashdebt
MSD		8.74	8.73	5.21	2.81	1.79	1.72	1.58	1.26	1.17	1.14
FC	mid	mom1m	mom12m	maxret	retvol	mom6m	idiovol	chg_mom6m	age	rd_mve	depr
MSD		20.25	16.13	7.62	6.06	4.50	2.96	2.90	2.36	2.18	2.10

Table 8: The table presents the MSD pooled over time and predictions conditioned on the respective size sample. The MSD is calculated asin equation 6. Row "l+m" represents the joint sample of large and mid caps.

[12] A thorough sub-sample approach takes sequentially one FC out and repeats the procedures presented in this work. Consequently, this would consume approximately 68 times the resources required to conduct the sensitivity check.

[13] The average is calculated as follows: $M\bar{S}D_k = \frac{1}{TP}\sum_{t=1}^{T}\sum_{p=1}^{P}\mathrm{MSD}_{t,k,p}$.

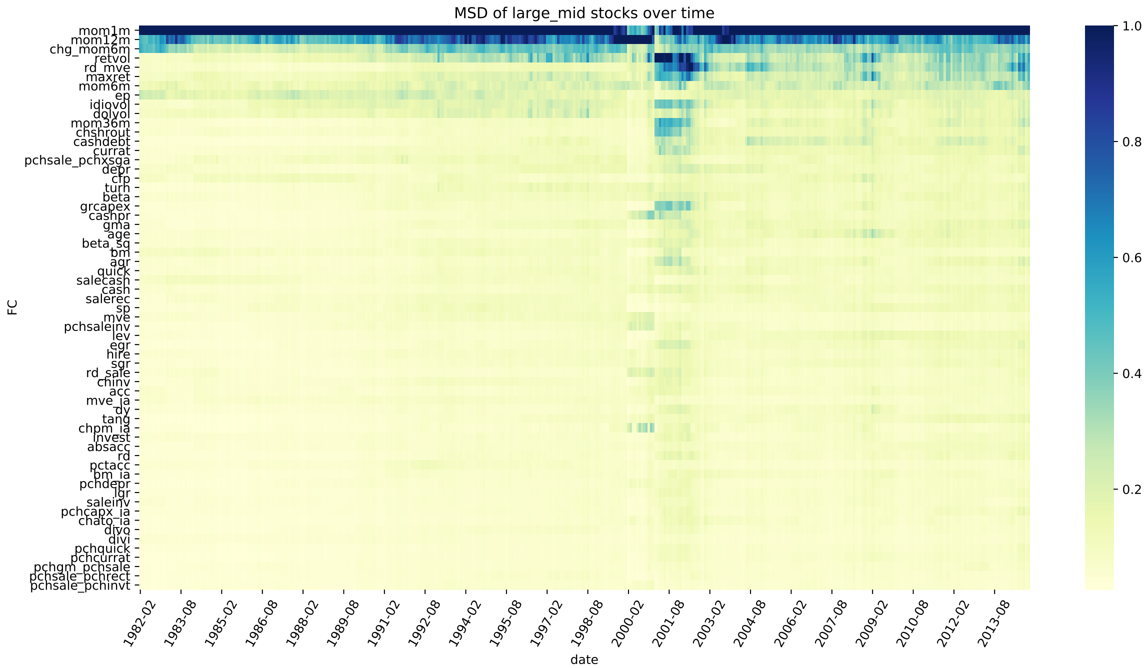


Figure 7: This sample includes the joint set of **large and mid cap stocks** and data from 1982-2014. The figure shows the expanding window of the MSD as calculated in equation 6 for all FC (sorted from top to bottom by the pooled average, from highest to lowest). The values are cross-sectionally normalized to range between 0 and 1. The highest value is obtained for *mom1m*, followed by *mom12m*. Moreover, the top five FC are all solely based on past price information.

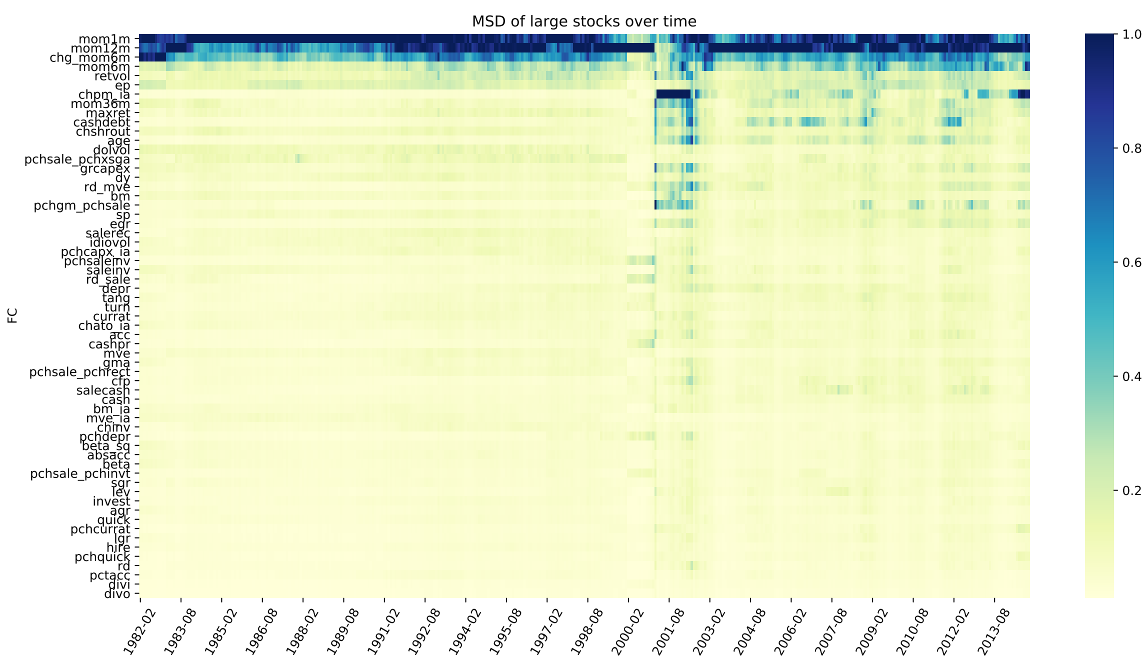


Figure 8: This sample includes **large cap stocks** and data from 1982-2014. The figure shows the expanding window of the MSD as calculated in equation 6 for all FC (sorted from top to bottom by the pooled average, from highest to lowest). The values are cross-sectionally normalized to range between 0 and 1. The highest value is obtained for *mom1m*, followed by *mom12m*. Moreover, the top five FC are all solely based on past price information.

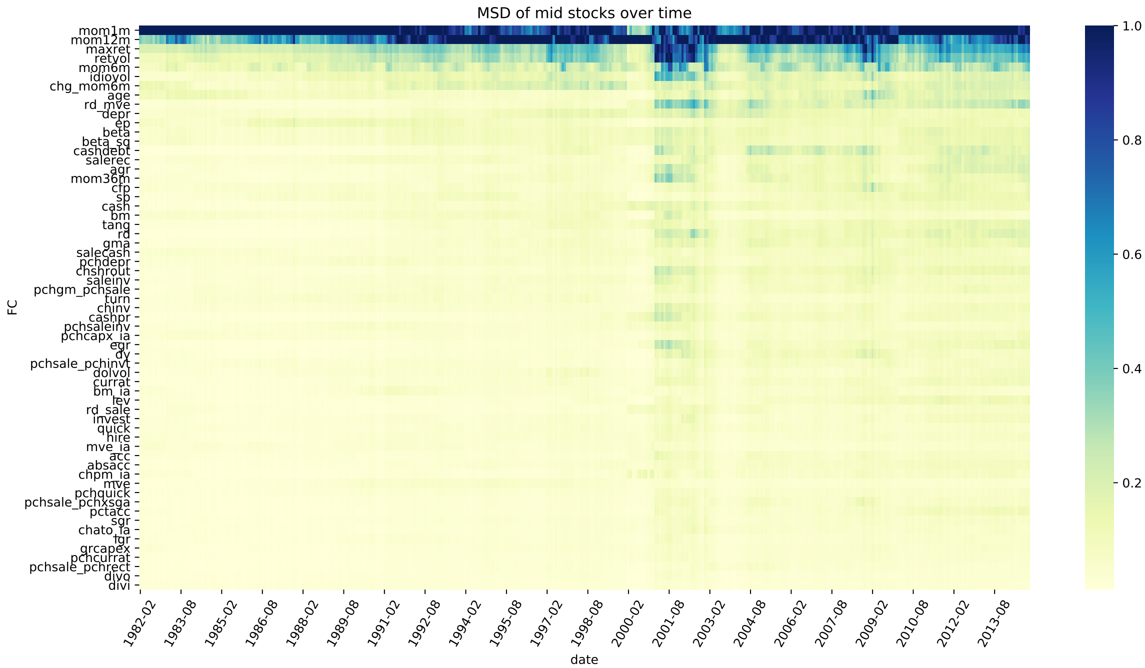


Figure 9: This sample includes **mid cap stocks** and data from 1982-2014. The figure shows the expanding window of the MSD as calculated in equation 6 for all FC (sorted from top to bottom by the pooled average, from highest to lowest). The values are cross-sectionally normalized to range between 0 and 1. The highest value is obtained for *mom1m*, followed by *mom12m*. Moreover, the top five FC are all solely based on past price information.

7 Conclusion

The study shows that deep feedforward neural networks provide a framework which prediction improvements of cross-sectional returns can be achieved when compared to a linear benchmark. One of the key tasks here is the identification of an optimal network design, which is robust to numerical deficiencies and competitive in terms of its prediction accuracy. I document that the majority of randomly determined architectures fail numerically, others perform poorly relative to the parsimonious benchmark or suffer from a stark estimation instability. Once these shortcomings are carefully bridged, long-short portfolios with attractive risk-adjusted returns can be constructed. I document that most predictions are driven by price-based FC. Moreover, a large fraction of FC have a negligible impact on the predictions. However, one cannot detangle whether these market inefficiencies stem from mispricing or are a result of computational and technological constraints faced by investors at the time.

This work is basic by design, and there are many ways to improve the deep-learning application. Moreover, slight deviations in the objective functions might cater other purposes more appropriately. Other obvious limitations are the specific selection of the underlying FC, which can be altered arbitrarily.

Furthermore, the implications from an investors perspective are manifold. Generally, deep-learning appears to be an attractive approach to stock selection. However, many problems need to be addressed first. For example, the rebalancing strategy needs to be optimized as a result of the prohibitive trading costs associated with a plain portfolio sorting strategy. Second-moments need to be accounted for and the risk capacity during crises periods has to be managed. These are all open questions, which are left for future research.

Finally, this study displays that non-linearities are important determinants of expected cross-sectional return spreads. There is no claim that deep-learning is the best way of exploiting these non-linearties. However, a distinct feature of this work is that it does not impose any prior on the functional relation between FC and the expected return process. The machine learns this non-linear dependence structure purely by itself.

8 Bibliography

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abarbanell, J. S. and B. J. Bushee (1997). Fundamental analysis, future earnings, and stock prices. *Journal of Accounting Research*, 1–24.

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

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

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

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

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

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

Bandyopadhyay, S. P., A. G. Huang, and T. S. Wirjanto (2010). The accrual volatility anomaly. Technical report, working paper, University of Waterloo.

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

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

Bengio, Y., Y. LeCun, et al. (2007). Scaling learning algorithms towards ai. *Large-scale kernel machines 34*(5), 1–41.

Bergstra, J. and Y. Bengio (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research 13*(Feb), 281–305.

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

Bishop, C. M. (1995a). *Neural networks for pattern recognition.* Oxford university press.

Bishop, C. M. (1995b). Regularization and complexity control in feed-forward networks. In *Proceedings International Conference on Artificial Neural Networks ICANNâ€™95*, pp. 807–814.

Bondt, W. F. and R. Thaler (1985). Does the stock market overreact? *The Journal of Finance 40*(3), 793–805.

Brandt, M. W., P. Santa-Clara, and R. Valkanov (2009). Parametric portfolio policies: Exploiting characteristics in the cross-section of equity returns. *The Review of Financial Studies 22*(9), 3411–3447.

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1984). *Classification and regression trees.* CRC press.

Brown, D. P. and B. Rowe (2007). The productivity premium in equity returns. Available at SSRN: https://ssrn.com/abstract=993467.

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

Chen, L. and L. Zhang (2010). A better three-factor model that explains more anomalies. *Journal of Finance 65*(2), 563–595.

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

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

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

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS) 2*(4), 303–314.

Daniel, K. and T. J. Moskowitz (2016). Momentum crashes. *Journal of Financial Economics 122*(2), 221–247.

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

DeMiguel, V., A. Martin-Utrera, F. J. Nogales, and R. Uppal (2017). A portfolio perspective on the multitude of firm characteristics. Available at SSRN: https://ssrn.com/abstract=2912819.

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

Dixon, M., D. Klabjan, and J. H. Bang (2015). Implementing deep neural networks for financial market prediction on the intel xeon phi. pp. 6.

Duchi, J., E. Hazan, and Y. Singer (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research 12*(Jul), 2121–2159.

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

Fama, E. F. and K. R. French (1992). The cross-section of expected stock returns. *The Journal of Finance 47*(2), 427–465.

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

Fama, E. F. and K. R. French (1996). Multifactor explanations of asset pricing anomalies. *The Journal of Finance 51*(1), 55–84.

Fama, E. F. and K. R. French (2014). A five-factor asset pricing model. *Journal of Financial Economics*.

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

Frazzini, A. and L. H. Pedersen (2014). Betting against beta. *Journal of Financial Economics 111*(1), 1–25.

Freyberger, J., A. Neuhierl, and M. Weber (2017). Dissecting characteristics non-parametrically. Available at SSRN: https://ssrn.com/abstract=2820700.

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

Glorot, X., A. Bordes, and Y. Bengio (2011). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315–323.

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

Goyal, P., P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Green, J., J. R. Hand, and F. Zhang (2017). The characteristics that provide independent information about average us monthly stock returns. *Review of Financial Studies*. Forthcoming.

Guo, R.-J., B. Lev, and C. Shi (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., R. Lundholm, and E. Matthew Van Winkle (2011). Percent accruals. *The Accounting Review 86*(1), 209–236.

Hahn, J. and H. Lee (2009). Financial constraints, debt capacity, and the cross-section of stock returns. *The Journal of Finance 64*(2), 891–921.

Hand, J. R. and J. Green (2011). The importance of accounting information in portfolio optimization. *Journal of Accounting, Auditing & Finance 26*(1), 1–34.

Hansen, P. R. and A. Lunde (2005). A forecast comparison of volatility models: Does anything beat a garch(1,1)? *Journal of Applied Econometrics 20*(7), 873–889.

Harvey, C. R., Y. Liu, and H. Zhu (2016). â€¦ and the cross-section of expected returns. *The Review of Financial Studies 29*(1), 5.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological approach*. John Wiley & Sons.

Hinton, G. E. (2007). To recognize shapes, first learn to generate images. *Progress in brain research 165*, 535–547.

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

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

Hou, K., C. Xue, and L. Zhang (2017). Replicating anomalies. Available at SSRN: https://ssrn.com/abstract=2961979.

Jarrett, K., K. Kavukcuoglu, Y. LeCun, et al. (2009). What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2146–2153. IEEE.

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

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

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

Kinga, D. and J. B. Adam (2015). A method for stochastic optimization. In *International Conference on Learning Representations (ICLR).*

Kozak, S., S. Nagel, and S. Santosh (2017). Shrinking the cross section. Available at SSRN: https://ssrn.com/abstract=2945663.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105.

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

Larochelle, H., D. Erhan, A. Courville, J. Bergstra, and Y. Bengio (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pp. 473–480. ACM.

Ledoit, O. and M. Wolf (2008). Robust performance hypothesis testing with the sharpe ratio. *Journal of Empirical Finance 15*(5), 850–859.

Leung, M. T., H. Daouk, and A.-S. Chen (2000). Forecasting stock indices: a comparison of classification and level estimation models. *International Journal of Forecasting 16*(2), 173–190.

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

Loughran, T. and J. R. Ritter (1995). The new issues puzzle. *The Journal of Finance 50*(1), 23–51.

McCulloch, W. S. and W. Pitts (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics 5*(4), 115–133.

McLean, R. D. and J. Pontiff (2016). Does academic research destroy stock return predictability? *The Journal of Finance 71*(1), 5–32.

Messmer, M. and F. Audrino (2017). The (adaptive) lasso in the zoo - firm characteristic selection in the cross-section of expected returns. Available at SSRN: https://ssrn.com/abstract=2930436.

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

Moritz, B. and T. Zimmermann (2016). Tree-based conditional portfolio sorts: The relation between past and future stock returns. *Available at SSRN*.

Nair, V. and G. E. Hinton (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

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

Ou, J. A. and S. H. Penman (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.

Parker, D. B. (1985). Learning logic.

Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics 4*(5), 1–17.

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

Poultney, C., S. Chopra, Y. L. Cun, et al. (2007). Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pp. 1137–1144.

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

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

Rosenberg, B., K. Reid, and R. Lanstein (1985). Persuasive evidence of market inefficiency. *The Journal of Portfolio Management 11*(3), 9–16.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Saad, D. (1998). Online algorithms and stochastic approximations. *Online Learning 5*.

Schaul, T., I. Antonoglou, and D. Silver (2013). Unit tests for stochastic optimization. *arXiv preprint arXiv:1312.6055*.

Sjöberg, J. and L. Ljung (1995). Overtraining, regularization and searching for a minimum, with application to neural networks. *International Journal of Control 62*(6), 1391–1407.

Sloan, R. (1996). Do stock prices fully reflect information in accruals and cash flows about future earnings?(digest summary). *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.

Srivastava, N., G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*(1), 1929–1958.

Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Takeuchi, L. and Y.-Y. A. Lee (2013). Applying deep learning to enhance momentum trading strategies in stocks. *Working Paper*.

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

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

Trippi, R. R. and D. DeSieno (1992). Trading equity index futures with a neural network. *The Journal of Portfolio Management 19*(1), 27–33.

Widrow, B., M. E. Hoff, et al. (1960). Adaptive switching circuits. *IRE WESCON convention record 4*(1), 96–104.

Williams, D. and G. Hinton (1986). Learning representations by back-propagating errors. *Nature 323*(6088), 533–538.

Wilson, D. R. and T. R. Martinez (2003). The general inefficiency of batch training for gradient descent learning. *Neural Networks 16*(10), 1429–1451.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association 101*(476), 1418–1429.

A Appendix

A.1 Computational Remarks

Despite computational benefits of random search, optimizing hyper-parameters can still be a computationally exhaustive task - ultimately depending on the number of trials performed and the allowed model complexity. Access to large CPU/GPU resources are crucial. This work is feasible only through access to a high performance computing grid and cloud computing resources. RAM is typically not the critical factor. I note that Abadi et al. (2015) provides two implementations, one for standard CPU designs and the other for modern GPU architecture. GPUs offer often significant performance benefits over classical CPUs design, and hence, can be a crucial factor in training and model capacity. This work is exclusively trained on CPU machines.

A.2 Additional Figures

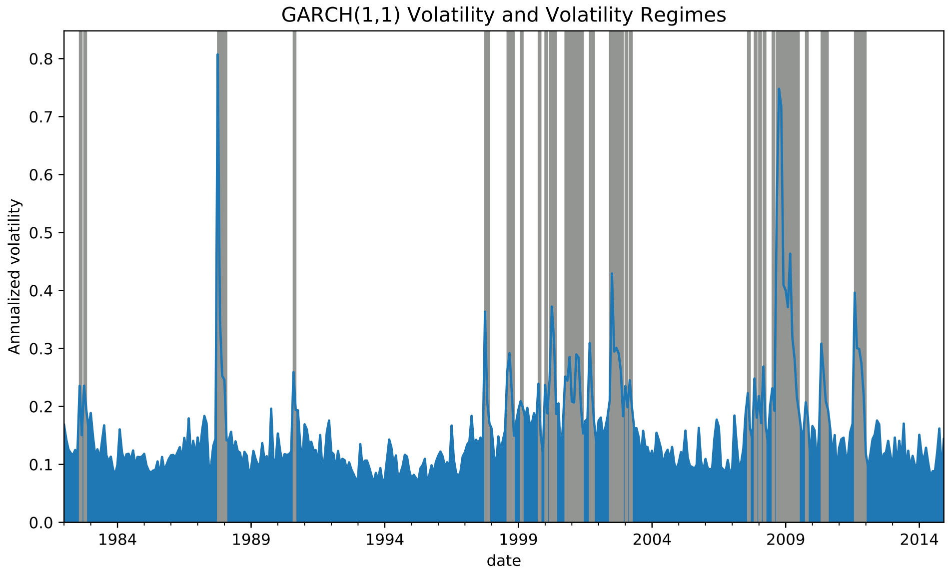


Figure 10: The graph depicts the estimated GARCH(1,1) process. Precisely, $\sigma_t^2 = \alpha + \beta_0 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$, The parameters estimated with normal distributed errors are: $\hat{\alpha} = 0.0079$, $\hat{\beta}_0 = 0.093$ and $\hat{\beta}_1 = 0.9019$. The model is estimated based on daily market factor returns with data from 1962-2014. The figure shows the corresponding month end values from 1982-2014. The area shaded in grey marks the highest 15% of the volatility estimates for the monthly subsample.

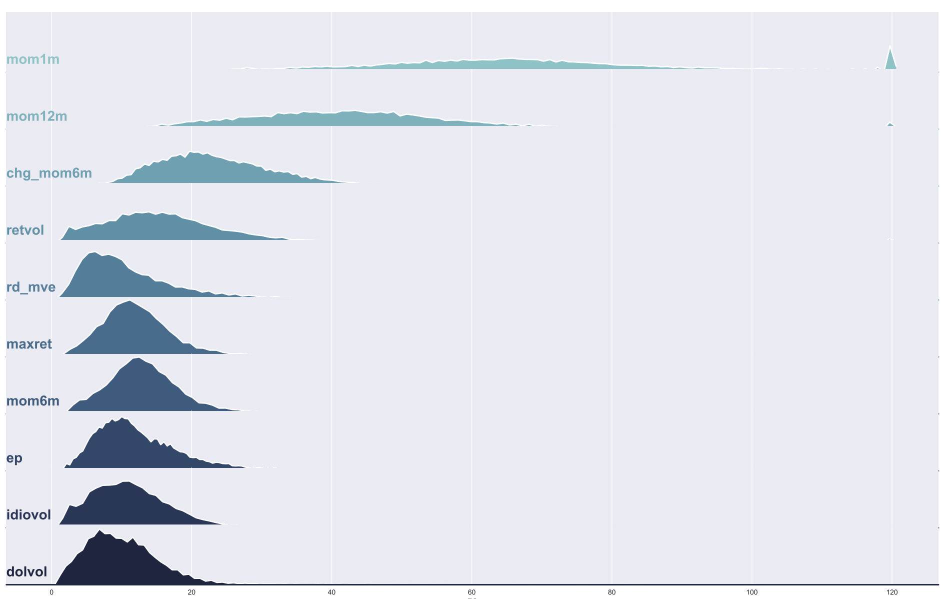


Figure 11: This sample includes **large and mid cap stocks** and data from 1982-2014. The figure shows the density of the MSD as calculated in equation 6 for the top 10 FC (measured by the average MSD and sorted from top to bottom, from highest to lowest). Short-term reversal or *mom1m* together with the twelve months momentum are causing the biggest deviations in the prediction of returns when they are set to zero. Values greater than 120 are truncated for better visual scaling.

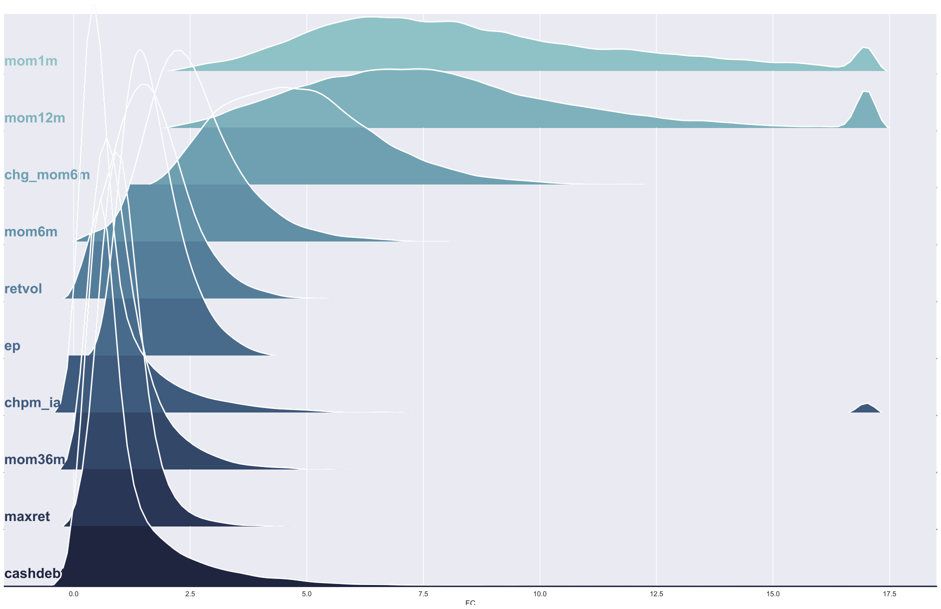


Figure 12: This sample includes **large cap stocks** and data from 1982-2014. The figure shows the density of the MSD as calculated in equation 6 for the top 10 FC (measured by the average MSD and sorted from top to bottom, from highest to lowest). *mom1m* or short-term reversal together with the change in six-months momentum are causing the biggest deviations in the prediction of returns when they are set to zero. Values greater than 17 are truncated for better visual scaling.

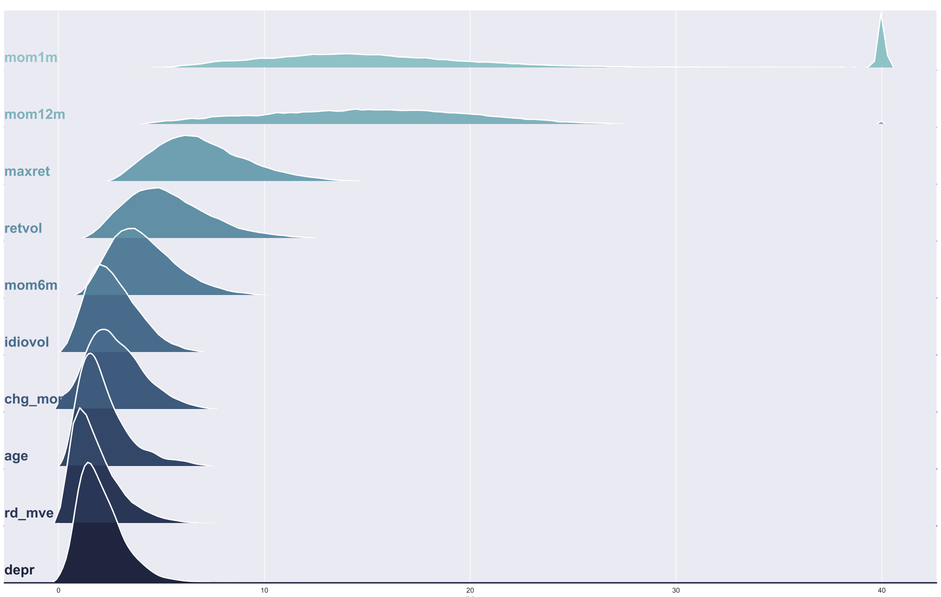


Figure 13: This sample includes **mid cap stocks** and data from 1982-2014. The figure shows the density of the MSD as calculated in equation 6 for the top 10 FC (measured by the average MSD and sorted from top to bottom, from highest to lowest). Short-term reversal or *mom1m* together with the twelve months momentum are causing the biggest deviations in the prediction of returns when they are set to zero. Values greater than 40 are truncated for better visual scaling.

A.3 Additional Tables

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	7.36	10.02	0.73	7.7***(4.09)	6.8***(3.61)	0.621
Min			4.69	11.13	0.42	4.3** (2.21)	2.8 (1.41)	-0.417
Max			10.22	9.42	1.08	10.2***(6.32)	8.7***(5.80)	1.93*
Linear			5.93	10.96	0.54	2.1 (1.14)	1.1 (0.58)	
Median	EW	single	7.85	9.54	0.82	7.3***(4.33)	5.7***(3.25)	0.821
Min			5.47	10.41	0.53	5.4***(2.81)	3.7** (1.98)	-0.0673
Max			10.28	8.18	1.26	9.0***(6.35)	7.3***(5.59)	2.33**
Linear			5.80	10.59	0.55	2.0 (1.12)	1.1 (0.59)	
Median	VW	Comb 75	8.29	10.60	0.78	8.5***(4.91)	6.8***(3.97)	0.846
Min			7.82	10.61	0.74	7.5***(4.32)	5.9***(3.33)	0.704
Max			9.03	10.69	0.84	8.8***(5.13)	7.1***(4.21)	1.06
Linear			5.93	10.96	0.54	2.1 (1.14)	1.1 (0.58)	
Median	EW	Comb 75	8.70	10.12	0.86	7.9***(4.48)	5.9***(3.34)	0.924
Min			8.47	10.37	0.82	7.9***(4.38)	6.0***(3.23)	0.783
Max			9.03	10.07	0.90	8.3***(4.80)	6.3***(3.67)	1.01
Linear			5.80	10.59	0.55	2.0 (1.12)	1.1 (0.59)	

Table 9: Large Cap Stocks, FF-style Cutoffs, Sample Period 1982-01 until 2014-12: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). All performance measures are displayed annualized. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	11.68	10.97	1.06	10.5***(5.72)	7.6***(4.37)	1.06
Min			9.71	11.37	0.85	9.6***(5.13)	6.8***(3.62)	0.204
Max			14.21	10.47	1.36	12.3***(6.92)	9.4***(6.14)	2.2**
Linear			9.10	11.34	0.80	9.5***(5.97)	7.0***(4.25)	
Median	EW	single	11.73	10.80	1.09	11.3***(6.42)	8.4***(5.24)	1.49
Min			9.69	11.22	0.86	9.8***(5.23)	7.0***(3.90)	0.354
Max			14.39	10.34	1.39	12.6***(7.12)	9.8***(6.84)	2.57**
Linear			9.11	11.73	0.78	10.0***(6.19)	7.5***(4.80)	
Median	VW	Comb 75	12.14	10.60	1.15	10.6***(6.10)	7.8***(4.53)	1.74*
Min			11.87	10.70	1.11	10.4***(5.97)	7.6***(4.35)	1.54
Max			12.51	10.67	1.17	10.9***(6.21)	8.0***(4.68)	1.82*
Linear			9.10	11.34	0.80	9.5***(5.97)	7.0***(4.25)	
Median	EW	Comb 75	12.26	10.66	1.15	11.2***(6.33)	8.3***(5.13)	1.95*
Min			11.93	10.68	1.12	10.8***(6.16)	8.0***(4.94)	1.83*
Max			12.67	10.53	1.20	11.3***(6.56)	8.5***(5.43)	2.25**
Linear			9.11	11.73	0.78	10.0***(6.19)	7.5***(4.80)	

Table 10: Mid Cap Stocks, FF-style Cutoffs, Sample Period 1982-01 until 2014-12: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). All performance measures are displayed annualized. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	18.52	20.12	0.92	18.6***(4.68)	14.0***(3.04)	0.0509
Min			13.07	20.63	0.63	16.3***(4.16)	10.8** (2.29)	-0.709
Max			23.30	18.47	1.26	23.3***(6.31)	18.3***(4.02)	0.944
Linear			13.87	15.39	0.90	8.7***(2.64)	7.2** (2.29)	
Median	EW	single	18.53	18.72	0.99	20.5***(5.42)	13.7***(4.49)	0.293
Min			12.72	19.26	0.66	14.7***(3.69)	10.0** (2.25)	-0.502
Max			22.65	17.68	1.28	24.6***(6.56)	17.6***(5.33)	0.992
Linear			14.40	16.81	0.86	8.3***(2.78)	7.1** (2.50)	
Median	VW	Comb 75	20.06	21.06	0.95	23.6***(6.00)	16.6***(4.20)	0.139
Min			18.75	20.09	0.93	20.9***(5.35)	14.3***(3.84)	0.0853
Max			21.14	20.40	1.04	23.7***(6.24)	17.1***(4.36)	0.358
Linear			13.87	15.39	0.90	8.7***(2.64)	7.2** (2.29)	
Median	EW	Comb 75	19.80	19.05	1.04	21.2***(5.69)	14.2***(3.90)	0.406
Min			19.17	19.08	1.01	20.7***(5.53)	13.6***(3.77)	0.333
Max			20.61	19.26	1.07	21.7***(5.67)	14.5***(3.90)	0.477
Linear			14.40	16.81	0.86	8.3***(2.78)	7.1** (2.50)	

Table 11: Large Cap Stocks, Decile Cutoffs, Sample Period 1982-01 until 2003-12: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). All performance measures are displayed annualized. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	0.71	13.92	0.05	-1.6 (-0.36)	-2.2 (-0.57)	-0.16
Min			-5.64	14.50	-0.39	-7.4* (-1.77)	-7.7** (-1.97)	-1.36
Max			6.30	12.17	0.52	3.6 (0.93)	3.3 (0.91)	0.863
Linear			1.51	13.00	0.12	-0.1 (-0.02)	-0.5 (-0.15)	
Median	EW	single	-0.22	11.17	-0.02	-4.5 (-1.57)	-4.8* (-1.82)	0.253
Min			-4.58	11.73	-0.39	-6.1* (-1.93)	-6.5** (-2.32)	-0.822
Max			4.51	10.06	0.45	2.5 (0.84)	2.1 (0.85)	1.33
Linear			-1.51	12.38	-0.12	-0.4 (-0.12)	-0.9 (-0.26)	
Median	VW	Comb 75	0.99	14.43	0.07	-1.3 (-0.28)	-1.7 (-0.43)	-0.121
Min			-0.49	15.05	-0.03	-3.6 (-0.78)	-4.1 (-1.01)	-0.402
Max			3.17	14.69	0.22	0.9 (0.21)	0.5 (0.12)	0.248
Linear			1.51	13.00	0.12	-0.1 (-0.02)	-0.5 (-0.15)	
Median	EW	Comb 75	-0.48	12.09	-0.04	-2.1 (-0.62)	-2.6 (-0.87)	0.243
Min			-1.20	12.44	-0.10	-2.9 (-0.81)	-3.4 (-1.12)	0.0765
Max			0.19	11.94	0.02	-1.1 (-0.32)	-1.6 (-0.52)	0.409
Linear			-1.51	12.38	-0.12	-0.4 (-0.12)	-0.9 (-0.26)	

Table 12: Large Cap Stocks, Decile Cutoffs, Sample Period 2004-01 until 2014-12: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). All performance measures are displayed annualized. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	27.65	16.07	1.72	24.9***(7.34)	19.4***(6.00)	1.69*
Min			23.80	16.98	1.40	20.8***(5.77)	14.9***(4.12)	0.606
Max			32.13	16.42	1.96	29.6***(8.56)	23.2***(7.44)	2.28**
Linear			19.66	16.24	1.21	21.4***(7.31)	17.0***(6.53)	
Median	EW	single	27.87	16.48	1.69	25.9***(7.22)	19.9***(5.92)	2.0**
Min			23.57	16.97	1.39	21.1***(5.89)	15.2***(4.30)	0.835
Max			32.28	16.47	1.96	30.2***(8.70)	24.0***(7.71)	2.72***
Linear			19.66	17.30	1.14	22.2***(7.40)	17.5***(6.77)	
Median	VW	Comb 75	29.97	17.42	1.72	27.6***(7.80)	21.1***(6.18)	1.72*
Min			29.10	17.26	1.69	25.7***(7.26)	19.5***(5.69)	1.59
Max			30.90	16.94	1.82	28.0***(8.11)	21.8***(6.52)	2.16**
Linear			19.66	16.24	1.21	21.4***(7.31)	17.0***(6.53)	
Median	EW	Comb 75	30.07	17.54	1.71	27.9***(7.94)	21.6***(6.50)	2.12**
Min			29.35	17.52	1.68	27.2***(7.64)	20.9***(6.22)	1.93*
Max			30.70	17.22	1.78	28.2***(8.24)	22.0***(6.62)	2.32**
Linear			19.66	17.30	1.14	22.2***(7.40)	17.5***(6.77)	

Table 13: Mid Cap Stocks, Decile Cutoffs, Sample Period 1982-01 until 2003-12: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). All performance measures are displayed annualized. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	5.18	14.13	0.37	2.3 (0.62)	1.6 (0.52)	0.92
Min			0.34	14.99	0.02	-1.2 (-0.28)	-1.9 (-0.58)	-0.11
Max			9.56	13.62	0.70	7.8** (2.10)	7.2** (2.11)	1.65*
Linear			1.02	17.50	0.06	6.2* (1.83)	5.5 (1.55)	
Median	EW	single	5.57	14.48	0.38	4.2 (0.98)	3.5 (1.01)	1.02
Min			1.90	15.20	0.12	2.2 (0.58)	1.5 (0.42)	0.207
Max			10.23	12.61	0.81	8.3** (2.31)	7.8** (2.37)	1.99**
Linear			1.18	17.99	0.07	6.5* (1.90)	5.7* (1.70)	
Median	VW	Comb 75	7.75	14.19	0.55	6.3 (1.61)	5.6* (1.73)	1.5
Min			6.89	14.59	0.47	5.6 (1.45)	4.9 (1.53)	1.3
Max			8.70	14.31	0.61	7.4* (1.91)	6.7** (2.03)	1.66*
Linear			1.02	17.50	0.06	6.2* (1.83)	5.5 (1.55)	
Median	EW	Comb 75	7.60	13.58	0.56	6.4 (1.62)	5.7* (1.74)	1.57
Min			6.40	13.82	0.46	5.3 (1.32)	4.6 (1.35)	1.27
Max			8.52	13.61	0.63	7.4* (1.86)	6.7** (1.99)	1.75*
Linear			1.18	17.99	0.07	6.5* (1.90)	5.7* (1.70)	

Table 14: Mid Cap Stocks, Decile Cutoffs, Sample Period 2004-01 until 2014-12: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). All performance measures are displayed annualized. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

Freq	Model	Costs	Mean	Std	SR	TO-%	FF-5 α	FF-5 + Mom α	SR Test
1	Comb	No	20.02	16.67	1.20		16.9***(6.28)	12.9***(4.57)	1.89*
		Yes	-8.73	16.06	-0.54	232	-11.5***(-4.51)	-15.3***(-5.87)	3.3***
	Linear	No	13.43	16.84	0.80		15.6***(6.64)	12.0***(5.18)	
		Yes	-22.53	16.20	-1.39	299	-20.2***(-8.95)	-23.6***(-10.88)	
2	Comb	No	16.35	16.33	1.00		13.9***(5.27)	9.9***(3.73)	1.06
		Yes	1.03	17.01	0.06	127	-1.1 (-0.43)	-5.0** (-1.97)	2.16**
	Linear	No	12.45	15.97	0.78		13.6***(5.60)	10.4***(4.90)	
		Yes	-5.79	16.87	-0.34	153	-4.3* (-1.81)	-7.4***(-3.52)	
3	Comb	No	14.31	16.19	0.88		12.3***(4.49)	8.6***(3.46)	1.65*
		Yes	3.13	17.05	0.18	94	1.3 (0.46)	-2.6 (-1.05)	2.27**
	Linear	No	8.60	16.06	0.54		9.5***(4.35)	7.3***(3.22)	
		Yes	-3.88	17.08	-0.23	107	-2.8 (-1.25)	-5.2** (-2.23)	
4	Comb	No	11.96	16.11	0.74		9.9***(3.94)	6.4***(2.70)	1.08
		Yes	2.82	16.54	0.17	77	0.7 (0.28)	-2.6 (-1.11)	1.55
	Linear	No	7.99	15.41	0.52		9.3***(4.75)	7.3***(3.47)	
		Yes	-2.02	15.97	-0.13	85	-0.8 (-0.38)	-2.6 (-1.21)	
5	Comb	No	12.38	15.55	0.80		9.5***(3.86)	6.4***(2.87)	2.25**
		Yes	4.56	16.21	0.28	67	1.5 (0.60)	-1.8 (-0.79)	2.24**
	Linear	No	5.95	16.45	0.36		8.2***(3.74)	5.7** (2.30)	
		Yes	-2.22	17.20	-0.13	70	-0.2 (-0.09)	-2.9 (-1.11)	
6	Comb	No	8.90	16.06	0.55		7.6***(2.92)	4.3* (1.69)	0.814
		Yes	1.93	16.68	0.12	60	0.7 (0.28)	-2.5 (-0.95)	1.01
	Linear	No	5.70	15.10	0.38		7.1***(3.50)	5.2** (2.45)	
		Yes	-1.42	15.70	-0.09	61	0.1 (0.05)	-1.8 (-0.85)	

Table 15: Mid Cap Stocks, Decile Cutoffs, VW Weighted: The table shows the performance impact once trading costs are explicitly accounted for. The column "Costs" compares the results with (Yes) and without (No) trading costs. Trading costs are calculated as in Brandt et al. (2009). TO-% indicates the average of the monthly portfolio turnover (the upper bound of a long-short portfolio is 400%). The column "Freq" refers to the rebalancing frequency in months.

ID	Acronym	Name	Description	Reference
1	beta	Beta	Measured based on 3 years (min 52 weeks) weekly excess returns with standard ols ($y = c + \beta x$)	Fama and MacBeth (1973)
2	beta_sq	Beta squared	Simply obtained by squaring the β based on the beta from # 1	Fama and MacBeth (1973)
3	retvol	Volatility	Volatility is measured by the standard deviation of daily returns of the previous months	Ang et al. (2006)
4	maxret	Maximum return	Maximum return is defined over the max of the daily returns in month $t-1$	Bali et al. (2011)
5	idiovol	Idiosyncratic volatility	Calculated based on the residuals of regression in # 1	Ali et al. (2003)
6	mom1m	1-month momentum	Return in month $t-1$	Jegadeesh (1990)
7	mom6m	6-month momentum	Cumulative return over 5 months ending in $t-2$	Jegadeesh and Titman (1993)
8	mom12m	12-month momentum	Cumulative return over 11 months ending in $t-2$	Jegadeesh (1990)
9	mom36m	36-month momentum	Cumulative return over 24 months ending in $t-13$	Bondt and Thaler (1985)
10	mve	Market capitalization (size)	log of (SHROUT$\times$PRC)	Banz (1981)
11	ep	Earnings-to-price	Earnings per share	Basu (1977)
12	dy	Dividends-to-price	Yearly dividends (dvt) divided by market cap at fiscal year	Litzenberger and Ramaswamy (1979)
13	bm	Book-to-market	Book value of equity (ceq) divided by market cap	Rosenberg et al. (1985)
14	lev	Leverage	Total liabilities (lt) divided by market cap	Bhandari (1988)
15	currat	Current ratio	Current assets (act) divided by current liabilities (lct)	Ou and Penman (1989)
16	pchcurrat	Pct change in current ratio	Percentage change in currat from year $t-1$ to t	Ou and Penman (1989)
17	quick	Quick ratio	Current assets (act) minus inventory (invt), divided by current liabilities (lct)	Ou and Penman (1989)
18	pchquick	Pct change in quick ratio	Percentage change in quick from year $t-1$ to t	Ou and Penman (1989)
19	salecash	Sales-to-cash	Annual sales (sale) divided by cash and cash equivalents (che)	Ou and Penman (1989)
20	salerec	Sales-to-receivables	Annual sales (sale) divided by accounts receivable (rect)	Ou and Penman (1989)
21	saleinv	Sales-to-inventory	Annual sales (sale) divided by total invetory (invt)	Ou and Penman (1989)
22	pchsaleinv	Pct change in sales-to-inventory	Percentage change in saleinv from year $t-1$ to t	Ou and Penman (1989)
23	cashdebt	Cashflow-to-debt	Earnings before depreciation and extraordinary items (ib + dp) divded by avg total liabilities (lt)	Ou and Penman (1989)
24	baspread	Illiquidity (bid-ask-spread)	Monthly avg of daily bid-ask spread divided by avg of daily bid-ask spread	Amihud and Mendelson (1989)
25	depr	Depreciation-to-gross PP&E	Depreciation expense (dp) divided by gross PPE (ppegt)	Holthausen and Larcker (1992)
26	pchdepr	Pct change in Depreciation-to-gross PP&E	Percentage change in depr from year $t-1$ to t	Holthausen and Larcker (1992)
27	mve_ia	Industry-adjusted firm size	Log market caps are adjusted by log of the mean of the industry	Asness et al. (2000)
28	cfp_ia	Industry-adjusted cashflow-to-price	Industry adjusted cash flow-to-price ratio equal weighted average	Asness et al. (2000)
29	bm_ia	Industry-adjusted book-to-market	Industry adjusted book-to-market equal weighted average	Asness et al. (2000)
30	sgr	Annual sales growth	Percentage change in sales from year $t-1$ to t	Lakonishok et al. (1994)
31	ipo	IPO	Indicated by 1 if first 12 months available on CRSP monthly file	Loughran and Ritter (1995)
32	divi	Dividend initiation	Indicated by 1 if company pays dividends but did not in prior year.	Michaely et al. (1995)
33	divo	Dividend omission	Indicated by 1 if company does not pay dividends but did in prior year.	Michaely et al. (1995)
34	sp	Sales-to-price	Annual sales (sale) divided by market cap	Barbee Jr et al. (1996)
35	acc	WC accruals	(ib) - (oancf)/(at), if (oancf) is missing then (ib)-(delta_act)-(delta_che) -(delta_lct) + (delta_dlc) + (txp-dp) where each item 0 if missing	Sloan (1996)
36	turn	Share turnover	Average monthly trading volume for the three months $t-3$ to $t-1$ divided by SHROUT at $t-1$	Datar et al. (1998)
37	pchsale_pchinvt	Delta pct change sales vs. inventory	Difference of percentage changes in sales (sale) and inventory (invt)	Abarbanell and Bushee (1997)
38	pchsale_pchrect	Delta pct change sales vs. receivables	Difference of percentage changes in sales (sale) and receivables (rect)	Abarbanell and Bushee (1997)
39	pchcapx_ia	CAPEX	Industry adjusted (two digit SIC) fiscal year mean adjusted percentage change in capital expenditures (capx)	Abarbanell and Bushee (1997)
40	pchgm_pchsale	Delta pct gross margin vs. sales	Annual percentage change in gross margin (sale minus cogs) minus percentage change in sales (sale)	Abarbanell and Bushee (1997)

Table 16: The table displays the firm characteristics used. Most definitions are taken from Green et al. (2017). If not otherwise stated, accounting ratios always refer to fiscal year end values. The table is taken from Messmer and Audrino (2017).

ID	Acronym	Name	Description	Reference
41	pchsale_pchxsga	Delta pct sales vs. SGaA	Annual percentage change in sales (sale) minus percentage change in SGaA (xsga)	Abarbanell and Bushee (1997)
42	dolvol	Dollar trading volume	Log of trading volume times price per share from month t-2	Chordia et al. (2001)
43	std_dolvol	Volatility trading volume	Monthly standard deviation of daily trading volume	Chordia et al. (2001)
44	std_turn	Volatility turnover	Monthly standard deviation of daily share turnover	Chordia et al. (2001)
45	chinv	Change in inventory	First difference of inventory (invt) devided by total assets	Thomas and Zhang (2002)
46	pchemp_ia	Industry-adjusted pch in employees	Industry adjusted percentage change in employees	Asness et al. (2000)
47	cfp	Cashflow-to-price	Operating cash flows (oancf) scaled by market capitalization (fiscal year end)	Desai et al. (2004)
48	rd	R&D Increase	If annual increase in R&D expenses (xrd) scaled by total assets (at) >0.05, 1, else 0	Eberhart et al. (2004)
49	lgr	Pct change in long-term debt	Annual percentage change in long term debt (lt)	Richardson et al. (2005)
50	egr	Pct change in book equity	Annual percentage change in book equity (ceq)	Richardson et al. (2005)
51	rd_sale	R&D-to-sales	R&D expenses(xrd) scaled by sales (sale)	Guo et al. (2006)
52	rd_mve	R&D-to-market	R&D expenses(xrd) scaled by market cap	Guo et al. (2006)
53	chg_mom6m	change in mom6m	difference of mom6m measured at t and $t-6$	Gettleman and Marks (2006)
54	hire	Pct change in employee	Annual percentage change in employee (emp)	Belo et al. (2014)
55	agr	Asset growth	Annual percentage change in assets (at)	Cooper et al. (2008)
56	cashpr	Cash productivity	Market cap plus long term debt (dltt) minus assets (at) divided by cash (che)	Chandrashekar and Rao (2009)
57	gma	Gross-profitability	Sales (sale) minus costs of goods sold (cogs) devided by one-year lagged assets(at)	Novy-Marx (2013)
58	cash	Cash-to-assets	Cash (che) divided by assets(at)	Palazzo (2012)
59	pctacc	Accruals-to-income	(ib) minus (oancf) divided by abs ((ib)), when (ib) equals 0, it is set to 0.01, if (oancf) is missing then (ib)-(delta_act)-(delta_che) -(delta_lct) + (delta_dlc) + (txp-dp) where each item 0 if missing	Hafzalla et al. (2011)
60	absacc	Absolut accruals	Absolute value of acc	Bandyopadhyay et al. (2010)
61	roic	Return on invested capital	Earnings before interest and taxes (ebit) - non-operating income (nopi), devided by non-cash enterprise value (ceq+lt-che)	Brown and Rowe (2007)
62	grcapex	Pct change in two year CAPX	Percentage change in two year capital expenditure (capx)	Anderson and Garcia (2006)
63	tang	Debt capacity-to-firm-tangability	(Cash (che) + 0.715 receivables (rect) + 0.547 inventory(invt) + 0.535 (ppegt))/ total assets (at)	Hahn and Lee (2009)
64	chshrout	Change in shares-outstanding	Yearly percentage change in outstanding shares (SHROUT)	Pontiff and Woodgate (2008)
65	invest	CAPEX and inventory	Yearly difference in gross property, plant and equipment (ppegt) + diff in (invt) / (t-1) total assets (at)	Chen and Zhang (2010)
66	age	Years since CS coverage	Years since first compustat coverage years(datadate - min(datadate))	Jiang et al. (2005)
67	chpm_ia	Industry-adjusted change in profit margin	Industry adjusted (two-digit SIC) change in profit margin (ib/sale)	Soliman (2008)
68	chato_ia	Industriy-adjusted change in asset turnover	Industry adjusted (two-digit SIC) change in asset turnover (sale/at)	Soliman (2008)

Table 17: The table displays the firm characteristics used. Most definitions are taken from Green et al. (2017). If not otherwise stated, accounting ratios always refer to fiscal year end values.The table is taken from Messmer and Audrino (2017).

A.4 Preliminary Figures and Tables

Due to the computational intensity not all results are yet fully ready to be included in the main sections of the manuscript. The following part includes only figures and tables of the joint set of large and mid cap stocks. It contains 55 out of 75 portfolio runs.

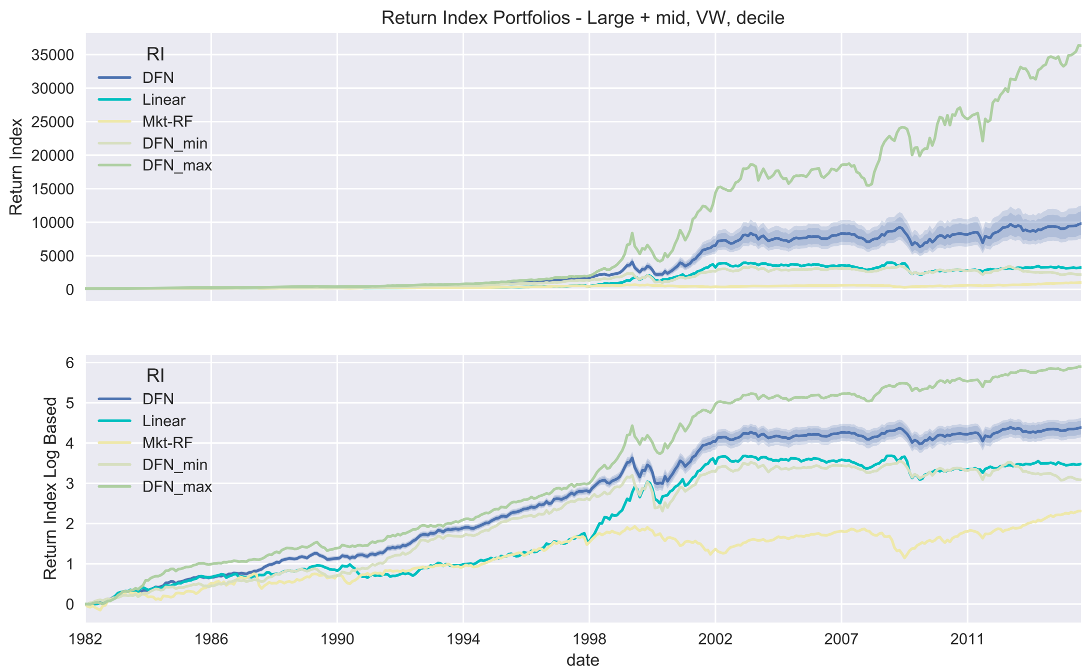


Figure 14: The figure exhibits selected paths of the 55 portfolios using large cap stocks with value weighted decile long-short portfolios for the years 1982-2014. "DFN" reflects the average of the 55 portfolio paths, which is embedded into the corresponding 1% and 10%-confidence bounds. The difference between the best and worst performing DFN portfolio reflects the uncertainty arising from model estimation.

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	15.15	18.17	0.83	15.0***(4.57)	11.6***(3.46)	0.654
Min			11.04	17.85	0.62	10.3***(3.50)	7.0** (2.26)	-0.224
Max			19.53	17.49	1.12	17.6***(5.59)	14.4***(5.02)	1.73*
Linear			12.28	18.22	0.67	13.4***(3.82)	9.0***(2.95)	
Median	EW	single	17.42	15.05	1.16	16.9***(6.48)	13.6***(5.20)	0.986
Min			13.38	15.83	0.85	11.2***(3.74)	7.5***(2.72)	0.036
Max			20.48	15.23	1.34	18.5***(6.89)	14.7***(5.90)	1.9*
Linear			14.49	17.36	0.83	17.1***(4.83)	12.6***(4.35)	
Median	VW	Comb 30	17.54	19.16	0.92	17.0***(5.00)	13.2***(3.82)	1.02
Min			16.27	18.81	0.86	14.7***(4.33)	10.8***(3.23)	0.79
Max			19.23	18.34	1.05	18.0***(5.54)	14.2***(4.37)	1.55
Linear			12.28	18.22	0.67	13.4***(3.82)	9.0***(2.95)	
Median	EW	Comb 30	19.32	16.23	1.19	17.6***(6.16)	13.4***(5.01)	1.27
Min			18.73	16.37	1.14	17.0***(5.98)	12.7***(4.85)	1.15
Max			20.12	16.03	1.26	18.5***(6.58)	14.3***(5.57)	1.54
Linear			14.49	17.36	0.83	17.1***(4.83)	12.6***(4.35)	

Table 18: Large + mid Cap Stocks, Decile Cutoffs, Sample Period 1982-01 until 2014-12: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). All performance measures are displayed annualized. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	8.77	10.54	0.83	8.1***(4.57)	6.8***(3.65)	0.857
Min			5.97	11.19	0.53	6.5***(3.46)	4.8** (2.42)	-0.307
Max			10.85	10.36	1.05	10.8***(6.25)	9.5***(5.49)	1.56
Linear			7.66	12.53	0.61	8.8***(3.98)	6.1***(2.94)	
Median	EW	single	10.61	8.04	1.32	9.9***(7.22)	8.2***(6.29)	1.48
Min			8.80	10.44	0.84	8.2***(4.38)	5.9***(3.23)	-0.0186
Max			13.01	8.48	1.53	10.8***(6.74)	8.9***(5.60)	2.6***
Linear			9.84	11.59	0.85	11.3***(4.86)	8.6***(4.28)	
Median	VW	Comb 30	10.69	11.20	0.95	10.8***(5.95)	9.2***(4.66)	1.3
Min			10.25	11.05	0.93	10.4***(5.70)	8.9***(4.44)	1.18
Max			11.44	11.14	1.03	11.7***(6.21)	10.2***(5.13)	1.55
Linear			7.66	12.53	0.61	8.8***(3.98)	6.1***(2.94)	
Median	EW	Comb 30	12.46	9.73	1.28	11.4***(6.79)	9.1***(5.33)	1.54
Min			12.22	10.09	1.21	11.5***(6.72)	9.0***(5.26)	1.25
Max			12.97	9.53	1.36	11.9***(7.20)	9.6***(5.85)	1.92*
Linear			9.84	11.59	0.85	11.3***(4.86)	8.6***(4.28)	

Table 19: Large + mid Cap Stocks, FF-style Cutoffs, Sample Period 1982-01 until 2014-12: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). All performance measures are displayed annualized. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	21.07	19.44	1.08	22.9***(6.10)	16.8***(4.00)	0.405
Min			16.24	18.28	0.89	14.5***(3.54)	9.4** (2.31)	-0.21
Max			26.72	18.77	1.42	28.6***(8.24)	23.5***(5.79)	1.37
Linear			18.67	19.46	0.96	21.5***(4.50)	14.5***(3.55)	
Median	EW	single	24.15	18.06	1.34	23.6***(6.45)	17.6***(4.80)	0.398
Min			19.38	17.47	1.11	18.5***(4.76)	12.9***(3.59)	-0.199
Max			29.20	15.22	1.92	28.1***(8.43)	22.8***(6.34)	1.95*
Linear			21.61	18.27	1.18	26.1***(5.44)	19.4***(4.94)	
Median	VW	Comb 30	24.77	19.96	1.24	26.6***(6.54)	20.7***(4.47)	0.895
Min			22.95	19.94	1.15	23.5***(5.63)	17.6***(3.81)	0.599
Max			26.52	19.50	1.36	27.4***(6.78)	21.2***(4.76)	1.26
Linear			18.67	19.46	0.96	21.5***(4.50)	14.5***(3.55)	
Median	EW	Comb 30	26.84	17.60	1.52	26.4***(7.21)	19.7***(5.66)	0.899
Min			26.08	17.81	1.46	25.7***(7.07)	19.2***(5.44)	0.724
Max			27.69	17.32	1.60	27.3***(7.75)	20.7***(6.03)	1.07
Linear			21.61	18.27	1.18	26.1***(5.44)	19.4***(4.94)	

Table 20: Large + mid Cap Stocks, Decile Cutoffs, Sample Period 1982-01 until 2003-12: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). All performance measures are displayed annualized. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

Portfolio	Weights	Type	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
Median	VW	single	2.26	15.18	0.15	-1.8 (-0.40)	-2.1 (-0.52)	0.44
Min			-2.20	15.39	-0.14	-6.1 (-1.36)	-6.6 (-1.61)	-0.281
Max			7.28	13.73	0.53	2.7 (0.75)	2.3 (0.68)	1.39
Linear			-0.43	14.84	-0.03	-0.8 (-0.21)	-1.3 (-0.35)	
Median	EW	single	3.68	11.99	0.31	2.1 (0.62)	1.4 (0.55)	1.04
Min			-0.30	12.85	-0.02	-1.2 (-0.33)	-1.8 (-0.70)	-0.164
Max			6.58	11.12	0.59	7.1** (2.34)	6.5***(2.86)	1.68*
Linear			0.35	14.62	0.02	2.0 (0.55)	1.3 (0.39)	
Median	VW	Comb 30	3.51	15.99	0.22	-0.4 (-0.10)	-0.8 (-0.20)	0.649
Min			0.98	16.01	0.06	-3.0 (-0.64)	-3.5 (-0.83)	0.241
Max			4.72	14.97	0.32	1.1 (0.28)	0.7 (0.19)	0.945
Linear			-0.43	14.84	-0.03	-0.8 (-0.21)	-1.3 (-0.35)	
Median	EW	Comb 30	4.36	11.91	0.37	3.0 (0.90)	2.4 (0.89)	1.26
Min			3.48	12.64	0.28	2.1 (0.59)	1.5 (0.50)	0.994
Max			5.71	12.25	0.47	4.1 (1.17)	3.4 (1.25)	1.59
Linear			0.35	14.62	0.02	2.0 (0.55)	1.3 (0.39)	

Table 21: Large + mid Cap Stocks, Decile Cutoffs, Sample Period 2004-01 until 2014-12: The table exhibits performance measures for several portfolio schemas. It contains value-weighted (VW) and equally-weighted (EW) portfolios, both for single and forecast combination based predictions ("Comb 75", combines 75 single model predictions). All performance measures are displayed annualized. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark. The portfolio classification into "Median", "Min" and "Max" portfolios are determined based on the the final return index values. 150 DFN portfolios are considered.

Size	Volatility	Portfolio	mean	std	SR	FF-5 α	FF-5 + Mom α	SR Test
large_mid	high	DFN	-1.90	31.97	-0.06	4.9 (0.36)	30.2***(2.83)	-0.137
		Linear	0.55	30.00	0.02	2.4 (0.14)	31.1***(2.99)	
	low	DFN	20.83	15.06	1.38	17.2***(5.27)	8.3***(3.12)	1.83*
		Linear	14.38	15.17	0.95	16.0***(3.93)	5.4* (1.73)	

Table 22: Volatility Regimes, Decile Cutoffs, VW, Sample Period 1982-01 until 2014-12: The table depicts performance measures conditional on the market volatility regime. The regime "high" are all months in which the GARCH(1,1) volatility estimate belongs to the 15% highest estimates, "low" accordingly to the lowest 85%. The DFN portfolio refers to the median forecast combination portfolio ("Comb 75"). All performance measures are displayed in annual terms. The values in brackets of the α-columns indicate t-values, which are based on standard deviation estimates corrected for HAC, using a lag of one period. "FF 5" abbreviates the Fama and French (2014) five-factor model, "FF 5 + mom" the momentum augmented version of the latter. The Sharpe ratio test column reflects t-values of the Ledoit and Wolf (2008) test, which are measured against the linear benchmark.