Imperial College Business School

Beyond microcaps - predicting cross-sectional stock returns via machine learning

By

Xuhui Li

May 2021

Thesis submitted in fulfilment of the requirements for the degree of MSc Financial Technology and the Diploma of Imperial College London

Acknowledgements

I would like to thank my supervisor Prof.Andrea Buraschi for his support and constructive advice on best addressing the relevant and cutting-edge questions, and structuring this dissertation. The conversation with him have led to a new direction of this thesis and an exciting journey in discovering achievements of deep learning in finance.

I also would like to express my gratitude to the lecturer of Big Data In Finance II - Dr.Ansgar Walther, whose guidance are an indispensable part in building my understanding of deep learning and some forefront applications of deep learning in finance. His lectures on deep learning with focus on financial applications, and relevant literature he provided have broaden my vision in terms of the orientation of this thesis.

Last but not least, I would like to express my sincerest appreciation to my family, my cohort and colleagues at the Imperial College Business School. Without their throughout support and encouragement, I won't be able to complete this thesis in such an unprecedented and challenging period.

Abstract

The objective of this thesis is to provide a comparative analysis of various machine learning methods in forecasting cross-sectional US stock returns upon excluding small-cap equities. We adopt four machine learning techniques with increasing complexity, ranging across penalised linear methods that include lasso and elastic net, partial least squares (PLS), feedforward neural network (FNN), and feedforward neural network with long short-term memory (LSTM-FNN). R^2_{oos} and long-only machine learning portfolios are used to assess the predictive accuracy and economic gains for each technique. We also identify influential anomalies for all methods. Furthermore, return predictability of machine learning approaches is evaluated under different market states. Last but not least, we investigate return predictability amid two economic recessions (2001 dot-com bubble and 2008 financial crisis), post 2001 decimalisation and post-financial crisis. The research carried out in this thesis could be used for investment management purposes by asset management firms, investment banks and hedge-funds.

Keywords: Machine Learning, Deep Learning, Neural Networks, Long Short-term Memory Network, Cross-sectional Return Predictability, Investment Management, FinTech
JEL Classification: C31, C45, C52, C55, G11, G17

Contents

1 Introduction

In this thesis, we conduct a comparative analysis of various machine learning techniques, with a primary purpose to evaluate machine learning methods in forecasting cross-sectional stock returns exclusively beyond small-cap US equities.

The most fundamental question in empirical asset pricing is to understand why different assets have different expected returns. Advancements in financial technology such as algorithmic trading and artificial intelligence lead to increasing liquidity in the financial market. Market efficiency results in return variation to be largely affected by unexpected events, which makes it proverbially difficult to forecast stock returns (Gu, Kelly, and Xiu 2018). As we are living in a world with surging amount of data, which replaces oil as the most valuable asset in the twenty-first century, it is increasingly important for investors to leverage cutting-edge techniques in machine learning to extract valuable information from the data.

The term "machine learning" could be defined as an abundant set of high-dimensional models used for statistical forecasting, incorporated with regularisation terms for variable selection and mitigation of overfitting (Gu, Kelly, and Xiu 2018). Although the effect of individual anomalies tend to deteriorate over time, machine learning is capable of combining a set of, possibly weak predictors into a significant aggregated signal (Avramov, Cheng, and Metzker 2019).

Machine learning is suitable for measuring equity risk premium for various reasons. First, machine learning is specialised in forecasting, which makes it well-suited to measure risk premium - the conditional expectation of future realised excess return (Gu, Kelly, and Xiu 2018). Second, unlike traditional econometric methods, machine learning techniques could incorporate a far more spacious set of predictors. More specifically, machine learning could effectively tackle multicollinearity that often exists in traditional econometric methods with an emphasis on variable selection and dimension reduction. Third, machine learning has a diverse range of functional forms, ranging from penalised linear methods such as lasso and elastic net, to more complicated nonlinear form such as neuron network, which is perfectly fit for handling high-dimensional predictor variables that often associated with complex nonlinear interactions.

Four machine learning techniques with increasing complexity but decreasing interpretability are employed in this thesis, they are: classic penalised linear methods that include both lasso and elastic net, dimension reduction technique such as partial least squares (PLS), feedforward neural network (FNN) and feedforward neural network with long short-term memory (LSTM-FNN). These methods

have their unique characteristics when compared with others. We hope our analysis could provide an overview of both pros and cons for each method.

To examine the predictability of cross-sectional stock returns, we provide two distinct evaluation metrics: R^2_{oos} and long-only machine learning portfolios. Out-of-sample R^2 is a classic statistical metric used in many literature. But more importantly, to truly assess how well each machine learning method performs economically, we need to evaluate cross-sectional return predictability at a portfolio level. Thus, we construct characteristic long-only machine learning portfolios to detect economic gains for each machine learning method.

There are a large number of candidate predictors that various researchers have argued possess predictive power for stock returns. In our analysis, we are also interested in understanding which set of predictor variables possess forecasting capacity across all machine learning methods.

As Avramov, Cheng, and Metzker (2019) show that machine learning methods often fail to yield robust performance upon standard economic restrictions, especially after excluding microcaps or distressed stocks. We find it valuable to assess cross-sectional return predictability upon excluding small-cap stocks.

In our empirical analysis, we also present return predictability of machine learning methods under different industries. Moreover, we investigate whether machine learning methods possess different predictive accuracy under different market states. It is also meaningful to know how well machine learning techniques perform amid economic recession, and periods after events (e.g., 2001 decimalisation and 2008 financial crisis) that result in structure change in the financial market. We hope by conducting such analysis could aid in tackling some practical investment problems, such as stock picking, market timing and portfolio construction, thereby further cementing the role of machine learning in the fintech industry.

1.1 Literature Review

The empirical literature on measuring predictability of stock return can be broadly categorised into two folds. Proposed by Fama and French (1992, 2015) and Lewellen (2014), the first fold uses linear factor models to explain incremental in cross-section of stock returns associated with firm-level characteristics. Moreover, Fama and French (2008) dissect influential anomalies that explain variation in portfolio-level predictability. The second fold runs time-series regressions of aggregate portfolio returns on a few macroeconomic covariates, exemplified by Welch and Goyal (2003, 2008) and Koijen and Van Nieuwerburgh (2011).

Welch and Goyal (2008) argue that historical mean outperforms linear regression-based models on predictor variables in terms of both in-sample and out-of-sample return predictability. In the past two decades, many literature introduce new techniques for the sake of improving out-of-sample return predictability. Campbell and Thompson (2008) find regression models yield significant out-of-sample predictability once weak restriction imposed on the signs of coefficients and return forecasts. Similar to Campbell and Thompson (2008), Hillebrand, Lee, and Medeiros (2013) improve the forecast performance further by applying bootstrap aggregation (bagging) to smooth parameter restriction. Rapach, Strauss, and Zhou (2010) suggest combining individual predictive regression models, which deliver statistically and economically meaningful performance out-of-sample. The regression-based approaches have a major limitation when a large number of predictors are imposed on the model. Advanced techniques in machine learning are well-suited to handle this drawback by introducing regularisation to the model. Furthermore, models within the field of deep learning can take into account of the interaction between predictors, which otherwise couldn't be found in classic regression-based models.

There is an enormous number of literature regarding using machine learning techniques for return prediction. Heaton, Polson, and Witte (2017) apply deep learning techniques in a variety of financial applications such as risk management, portfolio construction and securities pricing. Gu, Kelly, and Xiu (2018) conduct groundbreaking work by comparing a comprehensive suite of machine learning methods for forecasting the cross-sectional US stock returns. They show that nonlinear methods such as trees and neural networks outperform leading regression-based models in cross-sectional return prediction from a mean-variance investor perspective. Similar to Gu, Kelly, and Xiu (2018), Feng, He, and Polson (2018), Messmer (2017) and Dixon and Polson (2019) use deep neural networks to measure predictability of stock returns. Recently, Chen, Pelger, and Zhu (2019) adopt generative adversarial network to estimate asset pricing models imposed by no-arbitrage constraint.

In our project, we follow their idea on determining the dynamic pattern in macroeconomic time series via Long-Short-Term-Memory network before feeding them into a feedforward neural network. Following their work (Kelly, Pruitt, and Su, 2019) on measuring cross-sectional risk premium via Instrumented Principal Component Analysis (IPCA), Gu, Kelly, and Xiu (2019) propose a latent factor conditional asset pricing model by utilising an unsupervised dimension-reduction technique - autoencoder neural networks. Chinco, Clark-Joseph, and Ye (2019) use Least Absolute Shrinkage and Selection Operator (LASSO) to identify sparse signals in the cross-section of returns; and Freyberger, Neuhierl, and Weber (2020) propose adaptive group LASSO to detect characteristics that provide incremental information for the cross-sectional stock returns. Bryzgalova, Pelger, and Zhu (2019) apply decision trees to forecast cross-sectional stock returns. Although machine learning approaches have rapid advancement in return predictability, they also receive pointed criticisms. Avramov, Cheng, and Metzker (2019) review that profitability extracted by deep learning signals are primarily from difficult-to-arbitrage stocks and limits-to-arbitrage market conditions. Inoue and Kilian (2005) propose that out-of-sample testing is also susceptible to data mining issue. Leung et al. (2020) show that economic gains from machine learning techniques are more limited, and also subject to the ability to take extra risk.

1.2 Structure of the project

The rest of the project would be organised as follow: Section 2 first presents the reduced-form model to measure cross-section of individual stock returns. It then introduces a suite of both linear and nonlinear statistical learning methods. Techniques to improve predictive power and avoid the issue of overfitting specifically for each method would also be presented. This section will finish on the best practice on sample splitting, hyperparameter tuning, evaluation metrics and variable importance dissecting. Section 3 presents the empirical results for U.S. equities. Section 4 concludes.

2 Methodology

In this section we first describe the general reduced-form model to estimate the equity risk premium, we then introduce a suite of machine learning toolkits that we use to predict equity risk premium out of sample. The machine learning methods we use could be classified into two categories: linear and nonlinear. For each machine learning method, we aim to address three key elements. The first one is the explicit form of each statistical model for equity risk premium prediction. Second is the objective function that we use to estimate the model parameters. The last one is the computational algorithm for efficiently identifying the optimal specification among the permutations encompassed by a given method.

To address the problem of return predictability of equity risk premium, we essentially need to derive the estimation of asset pricing model. There are several asset pricing models with machine learning as the main approach to estimate cross-sectional stock returns in recent literature. Both Kelly, Pruitt, and Su (2019) and Kelly, Pruitt, and Su (2017) use beta pricing to esitmate cross-sectional equity risk premium. Gu, Kelly, and Xiu (2018) use a reduced-form model that classify an asset's excess return as a prediction error. Chen, Pelger, and Zhu (2019) and Kozak, Nagel, and Santosh (2017) use a no-arbitrage kernel form to estimate the stochastic discount factors, which explain cross-sectional returns from the conditional moment.

Given the similar purpose of comparing a panel of machine learning methods in measuring cross-sectional equity risk premium, we adopt the reduced-form model from Gu, Kelly, and Xiu (2018) to measure the cross-sectional returns of individual stocks. We derive the excess return of asset i at time t as an addictive prediction model:

$$r_{i,t} = E_{t-1}(r_{i,t}) + \epsilon_{i,t}, \tag{1}$$

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

Machine learning method is best suited to address the estimation of the above asset pricing model. We denote stocks in each month as $i = 1, ..., N_{t-1}$ and months by $t = 1, ..., T$. We aim to deliver the best approximation of $E_{t-1}(r_{i,t})$ as a function of anomalies (M-dimensional vector $z_{i,t-1}$) that maximise the out-of-sample predictability of cross-sectional returns $r_{i,t}$. We make the assumption that $g^*(z_{i,t-1})$ is a flexible and no time-dependent functional form that does not depend either on stock i and time t, which obtains universal estimates of risk premiums for individual

assets across the entire panel. This differs from the traditional approaches in asset pricing, where cross-sectional model are re-estimated at each time period or time-series models are independently estimated for each asset.

2.1 Simple Linear

The linear regression model estimated through ordinary least square (OLS) [1] is regarded as the most concise and interpretable method in statistical learning. In the empirical asset pricing literature, the widely-known Fama and French three-factor model (FF3) (Fama and French 1992) and Fama and French five-factor model (FF5) (Fama and French 2015) are also presented in plain linear regression formats. In our comparative study of machine learning methods, we would also examine the robustness of FF3 and FF5 in explaining the out-of-sample predictability of cross-sectional returns.

The conditional expectations $g^*(.)$ imposed by the simple linear model could be estimated by a linear function of the anomalies and its corresponding parameter vector, θ,

$$g(z_{i,t-1};\theta) = z_{i,t-1}^{'}\theta. \tag{3}$$

We define the ordinary least square, or l_2 objective function as:

$$\zeta(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}(r_{i,t} - g(z_{i,t-1};\theta))^2 \tag{4}$$

We obtain the pooled OLS estimator by minimising $\zeta(\theta)$. It should be noted that the optimal parameters are derived via a fixed-form solution, which do not require gradient-based optimisation.

I. Robust Objective Function

One of the shortcomings of OLS is that in the presence of heavy tail distribution, OLS is no longer a best linear unbiased estimator of the regression coefficients. Statisticians are long aware of this problem, and have devised sophisticated modification on top of the least squares objective function as shown in equation (4). Huber loss objective function (Huber 1996) is one way to tackle the problem of heavy-tailed distribution, which is defined as:

[1] The OLS estimator is also the lowest variance estimator among all other unbiased linear estimators (Puntanen and Styan 1989).

$$\zeta(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T} H(r_{i,t} - g(z_{i,t-1};\theta);\xi) \tag{5}$$

$$where, \qquad H(x;\xi) = \begin{cases} x^2, & \text{if } |x| \leq \xi; \\ 2\xi|x| - \xi^2, & \text{if } |x| > \xi. \end{cases} \tag{6}$$

The Huber loss objective function is not only applied to the simple linear regression, but it could also be imposed upon other machine learning methods such as penalised linear and neural networks that we will introduce in the following subsections.

2.2 Penalised Linear

When the model has a large number of predictor variables, OLS delivers nonzero estimates for all coefficients, which is suboptimal if we would like to implement variable selection. Moreover, OLS estimation is subject to overfitting in the presence of high-dimensional data sets. Thus, it is important to introduce penalisation to the objective function that simultaneously performs variable selection and coefficient estimation.

The general form of the penalised linear could be defined as:

$$\zeta(\theta;.) = \zeta(\theta) + \phi(\theta;.). \tag{7}$$

I. Lasso

Lasso (least absolute shrinkage and selection operator) is first introduced by Tibshirani (1996). Lasso is a penalised method that simultaneously performs variable selection and coefficient estimation via shrinkage. It's a l_1 optimisation problem defined as:

$$\zeta(\theta;\lambda) = \zeta(\theta) + \lambda\sum_{j=1}^{M}|\beta_j|, \tag{8}$$

where λ controls the strength of regularisation. Lasso has the benefit of providing a set of sparse

solutions, thus improving the interpretability of linear regression models. Chinco, Clark-Joseph, and Ye (2019) discover that the predictors identified by Lasso are associated with economically meaningful events in predicting cross-sectional returns.

II. Elastic Net

Another penalised linear technique we consider is the elastic net. Zou and Hastie (2005) show that it often outperforms Lasso in terms of predictive accuracy under simulation. They also pinpoint that the elastic net encourages grouping strongly correlated predictors together. The elastic net could be formulated as:

$$\zeta(\theta; \lambda, \rho) = \zeta(\theta) + \lambda(1-\rho)\sum_{j=1}^{M}|\beta_j| + \frac{1}{2}\lambda\rho\sum_{j=1}^{M}\beta_j^2. \tag{9}$$

The elastic net involves two hyperparameters (λ and ρ) and imposes two regularisers (Lasso and Ridge) simultaneously. When $\rho = 0$, it corresponds to the lasso and uses the l_1 optimisation; when $\rho = 1$, it corresponds to ridge regression and imposes the l_2 optimisation. The elastic net retains benefits from both shrinkage estimation and variable selection when ρ is an intermediate value.

We estimate the parameters of both lasso and elastic net by using the accelerated proximal gradient algorithm devised by Toh and Yun (2010) to optimise the Huber loss objective functions.

2.3 Partial Least Squares (PLS)

Although penalised linear methods could resolve high-dimension problems by incorporating l_1 or l_2 penalties, their predictions are suboptimal in the presence of collinearity amongst predictors [2]. One way to tackle collinearity is to form linear combination of predictors, also known as dimension reduction, which regroups correlated predictors into factors and thus helps decompose noise to generate effective signals. Partial least squares (PLS) is one of the techniques that performs dimension reduction by condensing the predictors according to their covariation with the forecast target [3].

To introduce PLS, we need to vectorise the linear model in (1) as:

[2] Most predictor variables in finance are highly correlated low signal to noise ratios.

[3] PLS is different with PCR (Princial Component Regression) in a sense that PCR exploits covariation amongst predictors without incorporating the forecasting returns, which often leads to suboptimal forecast because the principal components that best explain the predictors' variation are not necessary the best factors in terms of optimising the objective function. See Kelly and Pruitt (2013, 2015) for a more detailed explanation.

$$R = Z\theta + E, \tag{10}$$

where R is a vectorised version of $r_{i,t}$ with dimension $NT \times 1$, Z is the $NT \times M$ matrix of stacked predictors $z_{i,t-1}$ and E is a vectoised version of residuals $\epsilon_{i,t}$.

PLS is a dimension reduction approach by transforming the M dimension predictors into a low dimension representation of K linear combinations of predictors, we could then modify equation (10) as:

$$R = (Z\Psi_K)\theta_K + \tilde{E}. \tag{11}$$

Ψ_K is a $M \times K$ matrix with columns $w_1, w_2, ..., w_K$. PLS achieves a dimension-reduced form by projecting each predictor within Z into a set of K components. As a consequence, the predictive coefficient θ_K is now modified into a $K \times 1$ vector instead of $M \times 1$.

PLS seeks to exploit the maximum association between K linear combinations of Z and the forecast target. The estimation of weights in each jth PLS component solves [4]

$$w_j = argmax Cov^2(R, Zw), \quad s.t. \quad w^{'}w = 1, \quad Cov(Z_w, Z_{w_l}) = 0, \quad l = 1, 2, ..., j-1. \tag{12}$$

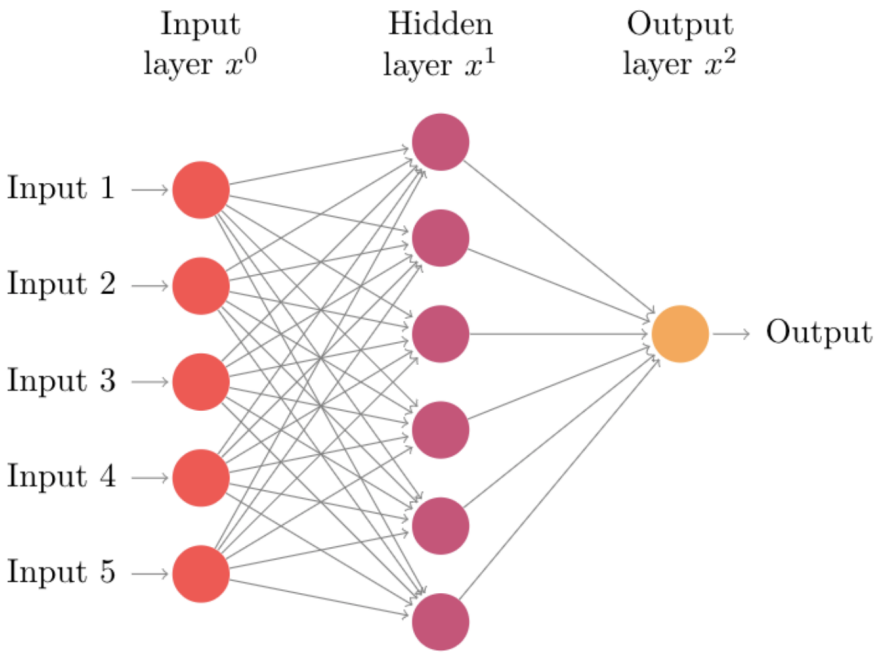
After obtaining a solution for Ψ_K, θ_K is estimated by an OLS regression of R on $Z\Psi_K$. K, the number of components, is the only hyperparameter for PLS.

[4] This problem could be efficiently resolved by using the SIMPLS algorithm from De Jong (1993).

2.4 Feedforward Neural Network (FNN)

A feedforward neural network, also called multilayer perceptrons (MLPs), is a nonlinear method that aims to approximate some function f^*. Given a mapping $y = f(x;\theta)$, a feedforward neural network learns the values of the parameter θ that lead to the best function approximation. It is called feedforward because it's a one-way information flow that starts from the input x, through the intermediate computations used to define f, and finally to the output y (Goodfellow, Bengio, and Courville 2016). It overcomes the limitations of linear models by allowing the interaction between predictor variables.

A feedforward network typically consists of an input layer of raw features, one or multiple hidden layers that concatenate and transform the input units, and an output layer that integrates hidden layers into an approximation for y.



The number of nodes in the input layer is equal to the dimension of the predictor variables, which we set to five in the above example (denoted as z_1, z_2, z_3, z_4, z_5). Each neuron in the hidden layer then combines information linearly from all the input units. Then, each neuron in the hidden layers employs a nonlinear "activation function" f to its integrated signal before transmitting its output to the next layer. For instance, the third neuron in the hidden layer modifies inputs into an

output as $x_3 = f\ (\theta_{3,0}^{(0)} + \sum_{j=1}^{5} z_j \theta_{3,j}^{(0)})$. Finally, the results from each neuron in the hidden layer are linearly integrated into a final output:

$$g(z;\theta) = \theta_0^{(1)} + \sum_{j=1}^{6} x_j^{(1)} \theta_j^{(1)}. \tag{13}$$

The total parameters in the above example is therefore $43 = (5+1) \times 6 + 7$ (six parameters to reach each neuron and seven weights to integrate the neurons into a final output).

There are infinite number of possibilities in terms of the design of a neural network. One has to make many choices when constructing a neural network such as the number of hidden layers, the number of neurons in each layer, and the connection between units. It's computationally expensive and unnecessary to select the optimal network architecture by "brute-force" cross-validation. Instead, a number of network architectures will be estimated according to the rule of thumb ex ante.

The best performing feedforward neural network design, a three-layer feedforward neural network (FNN3) [5] from Gu, Kelly, and Xiu (2018) will be used for comparative study. In their paper, feedforward neural network outperforms both linear and other nonlinear models such as tree models.

An active area of research in feedforward neural networks is the design of hidden units, which does not yet have many canonical theoretical norms (Goodfellow, Bengio, and Courville 2016). Most hidden units could be described as receiving a vector of input x, computing an affine transformation $z = W^T x + b$, then activating an element-wise nonlinear function $g(z)$. The distinguishable nature of most hidden units are the choices of the activation function. There are many well-known choices for the nonlinear activation function such as sigmoid, hyperbolic, and softmax. A standard and popular choice for $g(.)$ is known as the rectified linear units (ReLU), which is defined as [6]

$$ReLU(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{else,} \end{cases} \tag{14}$$

They are very convenient to optimise as they are similar to linear units with a main difference that a rectified linear unit outputs zero across half of its domain, which encourages sparse representation of active neurons and leads to faster and more consistent derivative evaluation. We use ReLU as the

[5] Gu, Kelly, and Xiu (2018) chooses the number of neurons in each layer based on the geometric pyramid rule (also see Masters (1993)). For instance, NN3 has 32, 16, and 8 neurons within each layer.

[6] See, Jarrett et al. (2009), Nair and Hinton (2010), and Glorot, Bordes, and Bengio (2011).

main activation function for all hidden units as what Gu, Kelly, and Xiu (2018) adopt in their paper.

To derive the general formula of the feedforward neural network model, we define $M^{(l)}$ as the number of neurons in each layer $l = 0, ..., L.$ and $x_m^{(l)}$ as the output of neuron m in layer l. Thus, the vector of outputs for this layer could be denoted as $x^l = (x_0^{(l)}, x_1^{(l)}, ..., x_{M^{(l)}}^{(l)})'$. Along with the input layer defined as $x^0 = (z_0, z_1, ..., z_N)'$, the recursive output formula for each hidden units in layer l > 0 and the final output are

$$x_m^{(l)} = ReLU(x^{(l-1)'}\Theta_m^{(l-1)}), \tag{15}$$

$$g(z;\Theta) = x^{(L-1)'}\Theta^{L-1}. \tag{16}$$

The total number of weight parameters that needed to estimate are

$$\sum_{l=1}^{L} M^{(l)}(1 + M^{(l-1)}). \tag{17}$$

Similar to other machine learning methods, to apply gradient-based algorithm we must choose a cost function. The parameters of neural networks are estimated by minimising the l_2 objective function of predictive errors.

As the objective function of neural networks is highly nonconvex, and back propagation easily leads to exploding or vanishing gradients, it's a common practice to train neural networks by using iterative gradient-based optimisation algorithm such as stochastic gradient descent (SGD).

Since neural networks have the characteristic of deep non-linearity and substantial parameters, apart from $l1$ regularisation, several regularisation techniques are also employed, they are: adaptive learning rate, early stopping, batch normalisation, dropout and ensemble learning.

When stochastically optimising the objective function, it is effective and necessary to shrink the learning rate towards zero when the gradient reaches zero to avoid the issue that the directional signal of the gradient is disturbed by the noise amid the calculation of it. The popular optimisation algorithms with adaptive learning rate [7] that are heavily used both in industry and academia are RMSProp, RMSProp with momentum, AdaDelta, and Adam [8]. Adam [9] (Kingma and Ba 2014) is

[7] Schaul et al. (2014) did a comparative study of optimisation algorithms. They found that the family of algorithms with adaptive learning rate yielded robust performance.

[8] For a comprehensive coverage of adaptive learning rate, see Goodfellow, Bengio, and Courville (2016).

[9] The name is derived from the phase"adpative moment." The algorithm is regarded as the hybird of RMSProp and momentum, which incorporate lower-order moments to dynamically adjust the learning rate Goodfellow, Bengio, and Courville (2016).

used as the default optimisation algorithm for feedforward neural network to minimise the l_2 objective function.

Early stopping is considered as one of the most commonly adopted regularisation techniques in deep learning due to its simplicity and effectiveness. By monitoring the training loss and the validation loss at the same time, we terminate the training algorithm until the error of the validation set has not improved for consecutive steps (also known as "patience"). It is highly effective because it releases the computational "burden" of the training procedure by limiting the number of training iterations. Moreover, it provides regularisation without imposing additional penalty terms to the cost function. It could be used either alone or with other regularisation techniques.

Batch normalisation (Ioffe and Szegedy 2015) is a recent methodology used to address the problem of vanishing and exploding gradients, which alters activation gradients in consecutive layers to either reduce or increase in magnitude. It addresses the issue of internal covariate shift [10] when training deep neural networks. The idea of it is to simply add additional "normalised hidden layers" between hidden layers to turn features into similar mean and variance that restore the representation power of the network (Aggarwal et al. 2018).

Another technique to avoid overfitting is to implement dropout when training neural networks, which is generally considered as a computationally inexpensive regulariser when compared with others (Srivastava et al. 2014). It could be thought of as a bagging approach by training ensembles of many neural networks. Intuitively speaking, dropout means dropping out units in a neural network along with their incoming and outgoing connections with a certain probability during training. Mathematically speaking, we could remove units from neural networks by multiplying their output values with zero (Goodfellow, Bengio, and Courville 2016). It is computationally cheap and could be combined with other regularisation methods.

Last but not least, ensemble learning is also a powerful technique for reducing generalisation error by incorporating multiple models (Breiman 1996). Since neural networks randomly initialise weight parameters and select minibatches, which often lead to different errors for members in the ensembles (Goodfellow, Bengio, and Courville 2016). This method often works very well because it reduces the variation of test set errors.

[10] The changing parameters during training result in the change of hidden variable activations, which leads to slower convergence because of the instability of the training data for later layers (Goodfellow, Bengio, and Courville 2016).

2.5 Recurrent Neural Network (RNN)

When a feedforward neural network is extended to include feedback connections, it is called a recurrent neural network. It could be thought of as adding additional loop to the FNN that allows it to exhibit temporal dynamic behaviour. Thus, it is a special type of neural network designed to tackle sequential problems.

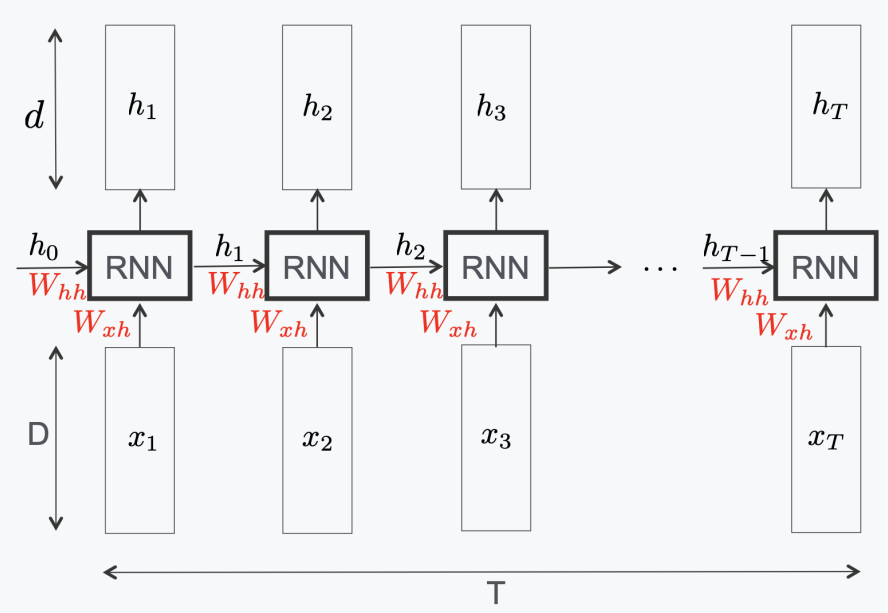


Figure 1: Recurrent Neural Network

As shown in Figure 1, a simple RNN layer is consist of a sequence of inputs $x_1, ..., x_T$ composed by D-dimensional vectors and a sequence of outputs $h_1, ..., h_T$ composed by d-dimensional vectors. At each time step, each recurrent unit will receive two inputs: the hidden state h_{t-1} from the previous unit and the new input x_t at the current timestep. The general process of RNN is formulated as:

$$h_t = tanh(W_{hh}^T h_{t-1} + W_{xh}^T x_t + b_h) \qquad \forall t \in 1, ..., T \tag{18}$$

where,

$$W_{xh} \in \mathbb{R}^{D \times d}, \quad W_{hh} \in \mathbb{R}^{d \times d}, \quad and \quad b_h \in \mathbb{R}^d \tag{19}$$

A conventional RNN can take into account of some time series behaviours but is not capable of

learning long-term dependency. Moreover, a key technical challenge faced with RNN is how to train them effectively. Experiments have shown that the rapid weight update procedure of RNN often results in problems such as vanishing or exploding gradients (Bengio, Simard, and Frasconi 1994; Pascanu, Mikolov, and Bengio 2013). Another improved version of recurrent neural network, called long short-term memory network, is explicitly designed to solve long-term dependency problems.

I. Long Short-Term Memory (LSTM)

Just like the RNN structure, LSTM also has the form of a chain repeating modules of neural networks.

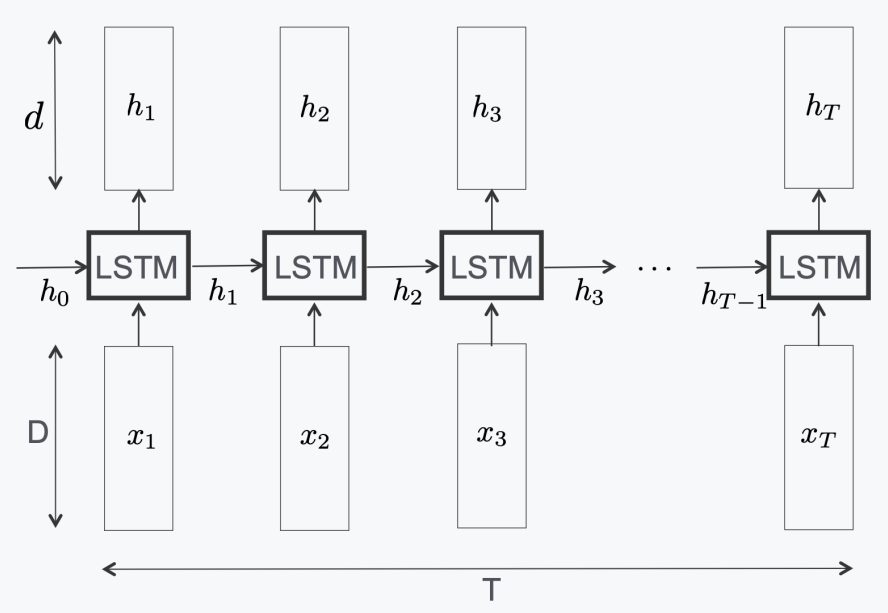


Figure 2: Long Short-Term Memory Network

The key difference between RNN and LSTM is that, unlike the standard RNN, the repeating modules have four single neural network layers, interacting in a unique way. The LSTM consists of a cell state (the memory component of LSTM), and three gates (the information flow controller of LSTM), which are input gate, forget gate, and output gate. The cell state is responsible for keeping track of the ordered sequence of the input variables; the input gate decides which values from the inputs to update the cell state; the forget gate controls what information to discard from the cell and the output gate is in charge of what to output based on the inputs and the memory from the cell.

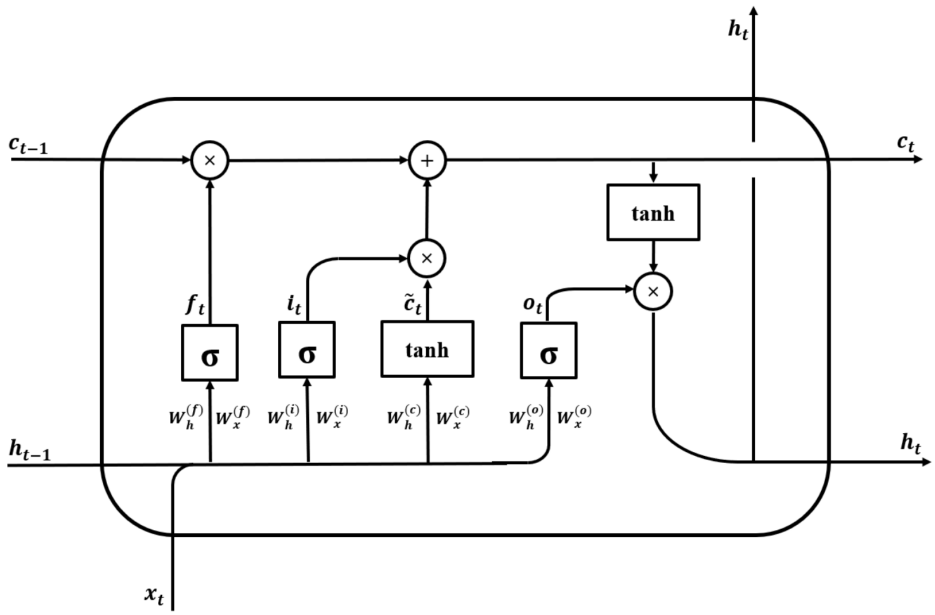


Figure 3: Structure of Long Short-Term Memory Unit

II. Hybrid LSTM-FNN

Following Chen, Pelger, and Zhu (2019), since most macroeconomic time series have time dependency, it is insufficient to only include the lagged value of macroeconomic variables at the last period. Instead, we need to find a model that extracts as much relevant information as possible from a large collection of lagged values to detect the business cycles. The LSTM is well-suited to find hidden states that simultaneously condense a large number of input variables and summarise long-term dependency of input predictors. It is the LSTM that retains the lost information that otherwise would be lost if we only consider the last observation of the macroeconomic predictors.

We take $x_t = m_t$ as the input sequence of macroeconomic times series, and the outputs are the hidden states h_t from the LSTM. The LSTM process is described as followed:

At each time step, a new cell state $\tilde{c}_t$ is updated with current input x_t and previous hidden state h_{t-1}, we have:

$$\tilde{c}_t = tanh(W_h^{(c)} h_{t-1} + W_x^{(c)} x_t + b_0^{(c)}). \tag{20}$$

The input, output and forget gates control the information flow between the input values and the memory cell, they are formulated as:

$$f_t = \sigma(W_h^{(f)} h_{t-1} + W_x^{(f)} x_t + b_0^{(f)}), \tag{21}$$

$$i_t = \sigma(W_h^{(i)} h_{t-1} + W_x^{(i)} x_t + b_0^{(i)}), \tag{22}$$

$$o_t = \sigma(W_h^{(o)} h_{t-1} + W_x^{(o)} x_t + b_0^{(o)}). \tag{23}$$

The sigmoid function σ is the activation function within the memory cell. Defining the element-wise product by $\circ$, the final memory cell and the hidden state are denoted as:

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{24}$$

$$h_t = o_t \circ tanh(c_t). \tag{25}$$

Our hybrid LSTM-FNN approach is to use LSTM condensing the macroeconomic variables m_t into a small number of hidden states h_t. The hidden states h_t along with firm characteristics $I_{i,t}$ are then fed into the feedforward neural network. The hybrid LSTM-FNN takes inputs h_t that condense the large set of macroeconomic time series without losing track of long-term dependency. We then again use the Adam algorithm to solve the l_2 optimisation problem. Figure 4 visually presents our hybrid LSTM-FNN network.

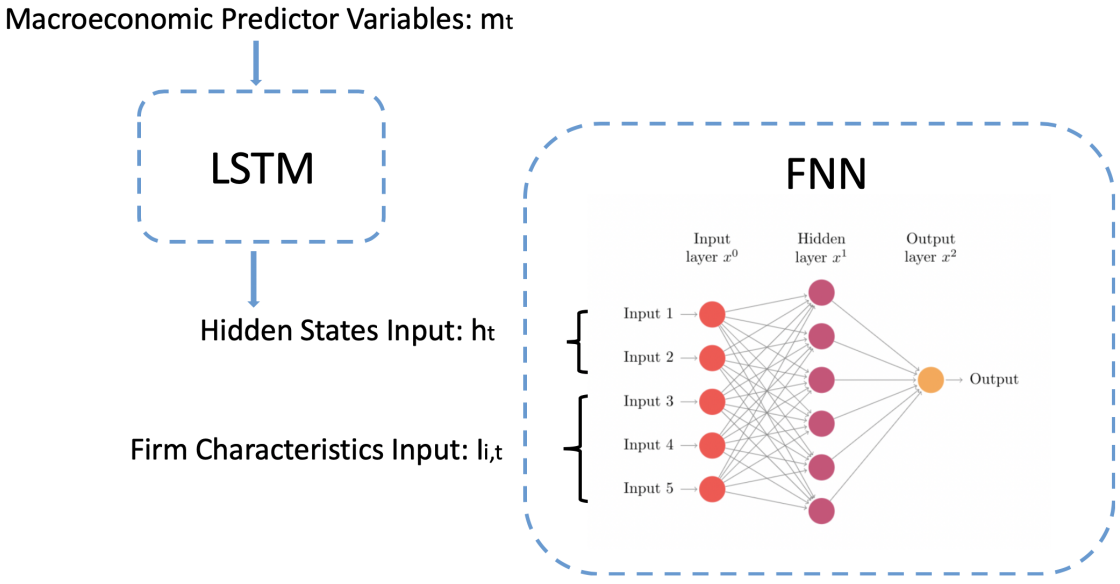


Figure 4: Structure of Hybrid LSTM-FNN

2.6 Data Splitting and Hyperparameter Tuning

As training machine learning models, especially deep neural networks, requires substantial amount of computing power, we adopt a "fixed" spitting scheme that splits the data into training, validation and testing subsamples [11]. The training, validation and testing subsamples are divided into three disjoint time periods without losing the sequential ordering of the data. The scheme first estimates the model from the training set subject to a specific set of hyperparameters. It then use the estimated model from the training set, and iteratively searches for the optimal configuration of hyperparameters from the validation set that results in the optimised objective. It should be noted that the sample fitting in the validation set is not truly out of sample. Instead, finally, the testing set, which is not used for either estimation or hyperparameter tuning, is used to evaluate the predictive performance of the given model. Table 1 summarises the selection of hyperparameters for each model and Table 2 reports the optimal hyperparameters for each model.

Table 1: Selection of Hyperparameters For All Machine Learning Models

	Linear	Lasso	Elastic Net	PLS	FNN3	Hybrid LSTM-FNN3
Huber loss ξ	✓	✓	✓	-	-	-
Learning rate	-	$10^{-2}, 10^{-3}$	$10^{-2}, 10^{-3}$	-	$10^{-2}, 10^{-3}$	$10^{-2}, 10^{-3}$
L_1 penalty λ_1	-	$10^{-3}, 10^{-4}, 10^{-5}$	$10^{-3}, 10^{-4}, 10^{-5}$	-	$10^{-4}, 10^{-5}, 10^{-6}$	$10^{-4}, 10^{-5}, 10^{-6}$
Number of components	-	-	-	2,3,5,7,9,10	-	-
Epochs	-	-	-	-	100	100
Batch Size	-	-	-	-	5000, 8000, 10000	5000, 8000, 10000
Dropout	-	-	-	-	0.3, 0.5, 0.8, 0.95	0.3, 0.5, 0.8, 0.95
Patience	-	-	-	-	5	5
Ensemble	-	-	-	-	10	10
Number of hidden states	-	-	-	-	-	4, 8, 16

Table 2: Optimal Hyperparameters For All Machine Learning Models

	Linear	Lasso	Elastic Net	PLS	FNN3	Hybrid LSTM-FNN3
Huber loss ξ	✓	✓	✓	-	-	-
Learning rate	-	10^{-2}	10^{-2}	-	10^{-3}	10^{-3}
L_1 penalty λ_1	-	10^{-3}	10^{-4}	-	10^{-6}	10^{-6}
Number of components	-	-	-	7	-	-
Epochs	-	-	-	-	100	100
Batch Size	-	-	-	-	10000	10000
Dropout	-	-	-	-	0.95	0.95
Patience	-	-	-	-	5	5
Ensemble	-	-	-	-	10	10
Number of hidden states	-	-	-	-	-	4

2.7 Model Evaluation

Following Gu, Kelly, and Xiu (2018), to evaluate the predictive performance of each method, we denote the out-of-sample R^2 as

[11] Gu, Kelly, and Xiu (2018) use a hybrid scheme in their paper, where the training and validation subsamples gradually include more recent data while the full history of training data are retained and the size of validation and testing subsample remain fixed. For a complete introduction of sample splitting, see, e.g, West (2006).

$$R^2_{oos} = 1 - \frac{\sum_{(i,t-1)\in\tau_3}(r_{i,t} - \hat{r}_{i,t})^2}{\sum_{(i,t-1)\in\tau_3} r^2_{i,t}}, \tag{26}$$

where τ_3 denotes the testing subsample, whose data is not used for either estimation or hyperparameter tuning. Instead of comparing predictions against historical mean returns as shown in many machine learning applications, Gu, Kelly, and Xiu (2018) define the denominator of R^2_{oos} without demeaning. They argue that the noisy nature of historical mean returns makes it a bad forecast comparison object [12]. A better approach would be simply benchmarking $r_{i,t}$ against zero.

In addition, to assess the performance of return predictability from an economic perspective, we also define annualised Sharpe ratio as followed:

$$SR = \frac{\mathbb{E}[R_t]}{\sqrt{Var(R_t)}} \tag{27}$$

2.8 Variable Importance

Apart from evaluating predictive performance of each machine learning method, it is also valuable to detect influential predictors that make a difference on the predictability of cross-sectional stock returns.

As in Sirignano, Sadhwani, and Giesecke (2016), Horel and Giesecke (2019), and Chen, Pelger, and Zhu (2019), we can rank the importance of firm characteristics based on the average absolute gradient. Specifically, the sensitivity of a particular variable zj could be formulated as the average absolute derivative of the weight $g(Z_I;\theta)$ with respect to this variable:

$$Sensitivity(z_j) = \mathbb{E}[|\frac{\partial}{\partial z_j} g(Z_I;\theta)| \mid Z_I = I_{i,t}], \tag{28}$$

$$where \qquad i,t \in \tau_1 \tag{29}$$

We measure the sensitivity of each variable zj within the training set, τ_1.

[12] Gu, Kelly, and Xiu (2018) carry out experiments by benchmarking forecast values against historical mean stock returns, and find that the out-of-sample monthly R^2 of all methods increase by approximately three percentage.

3 Empirical Results for U.S. Equities

3.1 Data

I. Returns and Firm Characteristic Variables

The monthly total individual equity returns are acquired from CRSP for all firms listed in the NYSE, AMEX, and NASDAQ. Our sample period spans from November 1964 to November 2019, totalling 55 years. We divide the full data into 25 years of training sample (November 1964 - November 1989), 5 years of validation sample (November 1989 - November 1994), and the remaining 25 years of out-of-sample testing sample (November 1994 - November 2019). Avramov, Cheng, and Metzker (2019) state that profitability extracted by machine learning-based methods is primarily from difficult-to-arbitrage stocks (or small-cap stocks), and limits-to-arbitrage market conditions [13]. Thus, we decide to evaluate cross-sectional return predictability upon excluding stocks with market equity below 0.01% of the aggregate US market cap. The total number of securities in our sample is around 5800, with the average number of stocks per month around 1,000. Figure 5 shows the number of stocks in each month. The three-month Treasury bill rates obtained from the economic research database at the Federal Reserve Bank at St. Louis (FRED) are used to calculate excess returns.

In addition, we collect 51 characteristic signals listed on Serhiy Kozak's Website [14], where 24 of them are updated annually, 1 of them is updated quarterly and 26 of them are updated monthly. This dataset contains values of anomalies for each stock at any point in time. These anomalies are ranked cross-sectionally, centered, and normalised by the sum of absolute values of all ranks in the cross section. Table 3 presents the full details of these firm characteristics. Moreover, 67 industrial classifications corresponding to the first two digits of Standard Industrial Classification (SIC) code [15] are collected specifically for the analysis of industrial effect.

[13] Small-cap stocks can often achieve higher Sharpe ratios and larger alphas. However, trading in these small-cap stocks is limited due to illiquidity and high spreads.

[14] The dataset could be obtained from https://www.serhiykozak.com/data.

[15] The full details of these code could be found in: https://mckimmoncenter.ncsu.edu/2digitsiccodes/.

II. Macroeconomic Variables

We acquire 135 macroeconmic time series primarily from two sources. We take 128 macroeconomic variables from the FRED-MD database as detailed in McCracken and Ng (2016). Another 7 macroeconomic predictors are constructed following the predictor definition from Welch and Goyal (2008), which are suggested as the macroeconomic covariates for the equity risk premium prediction [16]. These seven variables are: dividend-price ratio (dp), earning-price ratio (ep), book-to-market ratio (bm), net equity expansion (ntis), Treasury-bill rate (tbl), default spread (dfy), and stock variance (svar).

The standard transformations have been applied to these macroeconomic time series data. We apply the transformations suggested in McCracken and Ng (2016), and define transformations for the macroeconomic time series from Welch and Goyal (2008) to obtain stationary time series. The transformations include: (1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $log(x_t)$; (5) $\Delta log(x_t)$; (6) $\Delta^2 log(x_t)$; and (7) $\Delta(x_t/x_{t-1} - 1.0)$. A detailed description of the macroeconomic predictors as well as their corresponding transformations (tCode) could be found in the Appendix.

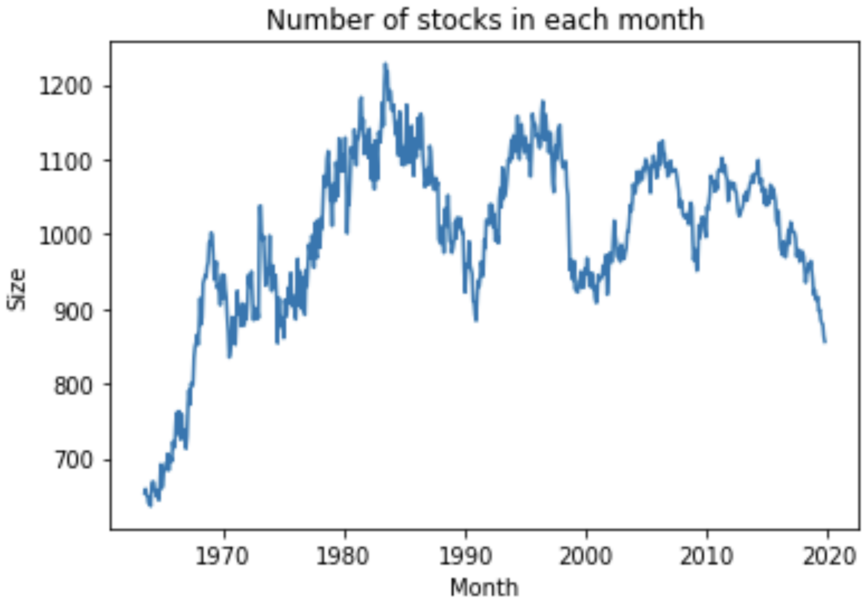


Figure 5: Number of stocks in each month

[16] The monthly data could be collected from Amit Goyal's Website: http://www.hec.unil.ch/agoyal/.

Table 3: Details of the Characteristics (Anomalies)

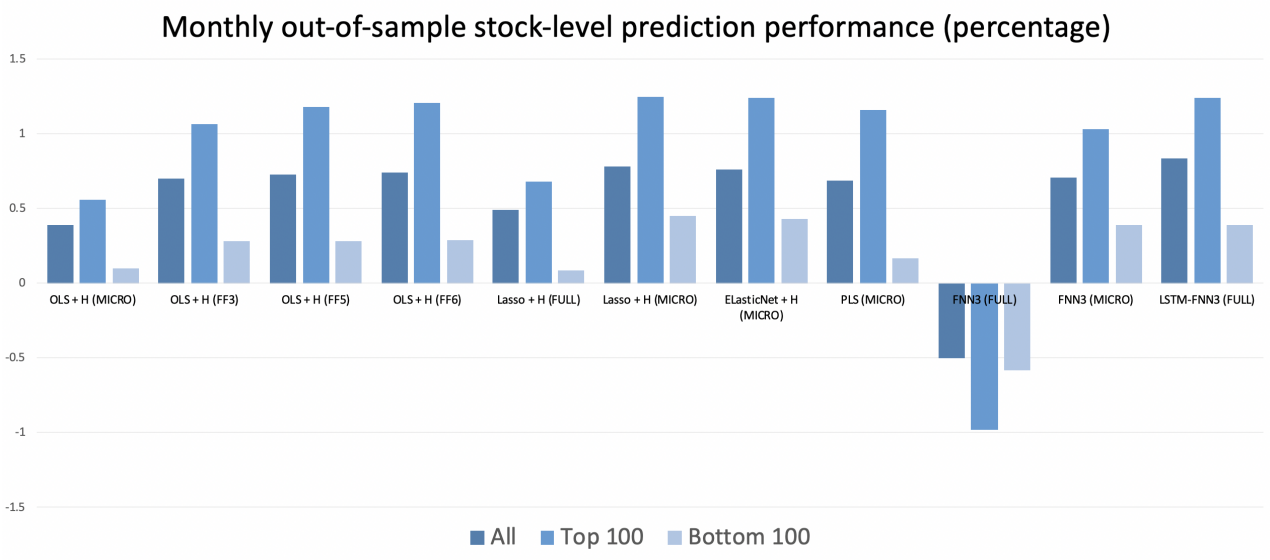
No.	Acronym	Firm characteristics	Paper's author(s)	Rebalanced Frequency
1	size	Size	Fama and French (1993)	Annually
2	value	Value	Fama and French (1993)	Annually
3	prof	Gross Profitability	Novy Marx (2013a)	Annually
4	valprof	Value Profitability	Novy Marx (2013b)	Monthly
5	fscore	Piotroski's F-score	Piotroski (2000)	Annually
6	debtiss	Debt Issurance	Spiess and Affleck-Graves (1999)	Annually
7	repurch	Share Repurchases	Ikenberry et al. (2015)	Annually
8	nissa	Share Issuance	Pontiff and Woodgate (2008)	Annually
9	accruals	Accruals	Sloan (1996)	Annually
10	growth	Asset Growth	Cooper et al. (2008)	Annually
11	aturnover	Asset Turnover	Soliman (2008)	Annually
12	gmargins	Gross Margins	Novy Marx (2013a)	Annually
13	divp	Dividend Yield	Naranjo et al. (1998)	Annually
14	ep	Earnings / Price	Basu (1977)	Annually
15	cfp	Cash Flow / Market Value of Equity	Lakonishok et.al. (1994)	Annually
16	noa	Net Operating Assets	Hirshleifer et al. (2004)	Annually
17	inv	Investment	Lyandres et al. (2007)	Annually
18	invcap	Investment-to-Capital	Xing (2008)	Annually
19	growth	Investment Growth	Xing (2008)	Annually
20	sgrowth	Sales Growth	Lakonishok et al. (1994)	Annually
21	lev	Leverage	Bhandari (1988)	Annually
22	roaa	Returns on Assets	Chen et al. (2011)	Annually
23	roea	Return on Equity	Haugen and Baker (1996)	Annually
24	sp	Sales-to-Price	Barbee Jr et al. (1996)	Annually
25	gltnoa	Growth in LTNOA	Fairfield et al. (2003)	Annually
26	mom	Momentum (6m)	Jagadeesh and Titman (1993)	Monthly
27	indmom	Industry Momentum	Moskowitz and Grinblatt (1999)	Monthly
28	valmon	Value Momentum	Novy Marx (2013b)	Monthly
29	valmonprof	Value Momentum Profitability	Novy Marx (2013b)	Monthly
30	shortint	Short Interest	Dechow et al. (1998)	Monthly
31	mom12	Momentum (1 year)	Jagadeesh and Titman (1993)	Monthly
32	momrev	Momentum Reversal	Jagadeesh and Titman (1993)	Monthly
33	lrrev	Long-term Reversals	DeBondt and Thaler (1985)	Monthly
34	valuem	Value	Asness and Frazzini (2013)	Monthly
35	nissm	Share Issuance	Pontiff and Woodgate (2008)	Monthly
36	sue	PEAD (SUE)	Foster et al. (1984)	Monthly
37	roe	Return on Book Equity	Chen et al. (2011)	Monthly
38	rome	Return on Market Equity	Chen et al. (2011)	Monthly
39	roa	Return on Assets	Chen et al. (2011)	Quarterly
40	strev	Short-term Reversal	Jegadeesh (1990)	Monthly
41	ivol	Idiosyncratic Volatility	Ang et al. (2006)	Monthly
42	beta	Beta Arbitrage	Cooper et al. (2008)	Monthly
43	season	Seasonality	Heston and Sadka (2008)	Monthly
44	indrrev	Industry Relative Reversals	Da et al. (2013)	Monthly
45	indrrevlv	Industry Relative Reversals (Low Vol)	Da et al. (2013)	Monthly
46	indmomrev	Industry Momentum Reversal	Moskowitz and Grinblatt (1999)	Monthly
47	ciss	Composite Issuance	Danial and Titman (2006)	Monthly
48	price	Price	Blume and Husic (1973)	Monthly
49	age	Firm Age	Barry and Brown (1984)	Monthly
50	shvol	Share Volume	Dater et al. (1998)	Monthly
51	dur	Cash flow duration	Dechow et al. (2004)	Monthly

3.2 The Cross-sectional Predictability of Individual Stock Returns

Table 4: Monthly out-of-sample stock-level prediction performance (percentage R^2_{oos})

	All	Top 100	Bottom 100
OLS + H (FULL)	-inf	-inf	-inf
OLS + H (MICRO)	0.39	0.56	0.10
OLS + H (FF3)	0.70	1.07	0.28
OLS + H (FF5)	0.73	1.18	0.28
OLS + H (FF6)	0.74	1.21	0.29
Lasso + H (FULL)	0.49	0.68	0.09
Lasso + H (MICRO)	0.78	1.25	0.45
ElasticNet + H (FULL)	-1.51	-2.91	-1.56
ElasticNet + H (MICRO)	0.76	1.24	0.43
PLS (FULL)	-77.48	-122.54	-57.49
PLS (MICRO)	0.69	1.16	0.17
FNN3 (FULL)	-0.50	-0.98	-0.58
FNN3 (MICRO)	0.71	1.03	0.39
LSTM-FNN3 (FULL)	0.84	1.24	0.39

In this table, we present monthly R^2_{oos} for the entire panel of stocks using OLS estimate on both firm characteristics and macroeconomic variables (named "FULL"), OLS estimated only on firm characteristics (named "MICRO"), OLS with Fama and French three factors (FF3), OLS with Fama and French five factors (FF5), OLS with Fama and French five factors along with momentum, Lasso (FULL), Lasso (MICRO), ElasticNet (FULL), ElasicNet (MICRO), PLS (FULL), PLS (MICRO), Three-layer Feedforward Neural Networks (Full), Three-layer Feedforward Neural Networks (MICRO), Hybrid Long Short-Term Memory - Three-layer Feedforward Neural Network (LSTM-FNN3). "+H" indicates the use of Huber loss instead of the l_2 loss. Moreover, we also report these R^2_{oos} within subsamples that include only the top 100 stocks or bottom 100 stocks by market capitalisation.



The bar charts present a visual comparison of the R^2_{oos} for various machine learning methods (OLS+H (FULL), elastic net (FULL) and PLS (FULL) are excluded due to their excess negative magnitudes).

We present the comparison of machine learning methods in terms of their out-of-sample predictive R^2 in Table 4. Fourteen models are compared in total, which include OLS estimated on both firm characteristics and macroeconomic variables, OLS estimated only on firm characteristics, OLS with Fama and French three factors (FF3, which includes market beta, size, and book-to-market ratio), OLS with Fama and French five factors (FF5, which includes market beta, size, book-to-market, profitability, and investment), OLS with Fama and French five factors along with momentum (FF6), lasso estimated only on firm characteristics, lasso estimated on both firm characteristics and macroeconomic variables, lasso estimated only on firm characteristics, elastic net estimated on both firm characteristics and macroeconomic variables, elastic net estimated only on firm characteristics, PLS estimated on both firm characteristics and macroeconomic variables, PLS estimated only on firm characteristics, three-layer feedforward neural networks estimated on both firm characteristics and macroeconomic variables, three-layer feedforward neural network estimated only on firm characteristics, long short-term memory with three-layer feedforward neural network estimated on both firm characteristics and macroeconomic variables. Furthermore, OLS, lasso and elastic net are reported with their improved version by using Huber loss, which generally perform better than the version without.

The second column of Table 4 presents R^2_{oos} for the entire cross-sectional sample. The OLS model using all 186 predictors (which includes both firm characteristics and macroeconomic variables) produces the worst result overall, indicating that the simple OLS is not suitable to extract valuable information from a large panel of predictors that include both firm characteristics and noisy macroeconomic data. Similarly, the OLS model using only the firm characteristics generates relatively better result - an R^2_{oos} at 0.39%. The poor results from the simple OLSs could be subject to the lack of regularisation that forces OLSs more prone to in-sample overfit. Nevertheless, constraining OLSs to a sparse representation of a few factors (e.g., FF3, FF5, and FF6) generate much robust results - with R^2_{oos} at 0.70%, 0.73% and 0.74% respectively; or by adding the penalisation terms on top of the linear regression such as lasso and elastic net also deduce rather significant outcomes - with R^2_{oos} at 0.78% (lasso MICRO) and 0.76% (elastic net MICRO) respectively, though penalised regression models still can't distill much value from the full panel of features - R^2_{oos} at 0.49% for lasso (FULL) and at -1.51% for elastic net (FULL).

Dimension-reduction technique such as partial least squares estimated only on firm characteristics also yield rather robust result - an R^2_{oos} of 0.69%, though PLS is also not able to bring much predictability from the full panel of features (R^2_{oos} -77.48 %).

The improvement of penalised linear methods (lasso and elastic net) via feature selection and PLS via dimension reduction suggest that characteristic signals are partially redundant and highly correlated. Penalised unnecessary variables or condensing highly correlated anomalies into low-dimension representation can better improve out-of-sample predictive accuracy.

Neural networks overall produce considerably remarkable results. The R^2_{oos} is 0.71% for FNN3 (MICRO). Although neural networks are known as specialising in incorporating complex feature interactions as reported by Gu, Kelly, and Xiu (2018), FNN3 still fails to produce robust results when estimated on both firm characteristics and macroeconomic variables (R^2_{oos} of -0.5%). Thus, we need a technique that both considers the sequential natural of time series data and condenses macroeconomic predictors into a few effective representations. The long short-term memory network comes into play. By transforming a large panel of macroeconomic time series data into a few hidden states before feeding them into FNN3, the hybrid LSTM-FNN3 generates the most outstanding result across all machine learning methods - with an R^2_{oos} at 0.84%.

The third and fourth columns of Table 4 present R^2_{oos} for the top-100 stocks by market capitalisation in each month and the bottom-100 stocks (relatively small stocks) in each month. These results are based on the models estimated on all stocks and forecasting on the two subsamples detailed above. We can see that all machine learning methods have significant improvement among large stocks in terms of R^2_{oos}, with improvement ranging from 0.17% to 0.48%. This finding is consistent with Gu, Kelly, and Xiu (2018), which further ensures that predictability of machine learning methods is not merely driven by picking up small-cap stocks that are illiquid and difficult-to-arbitrage [17]. In fact, relatively small stocks (bottom 100 stocks) have performance less robust than all stocks and big-cap stocks.

3.3 Predictor Importance

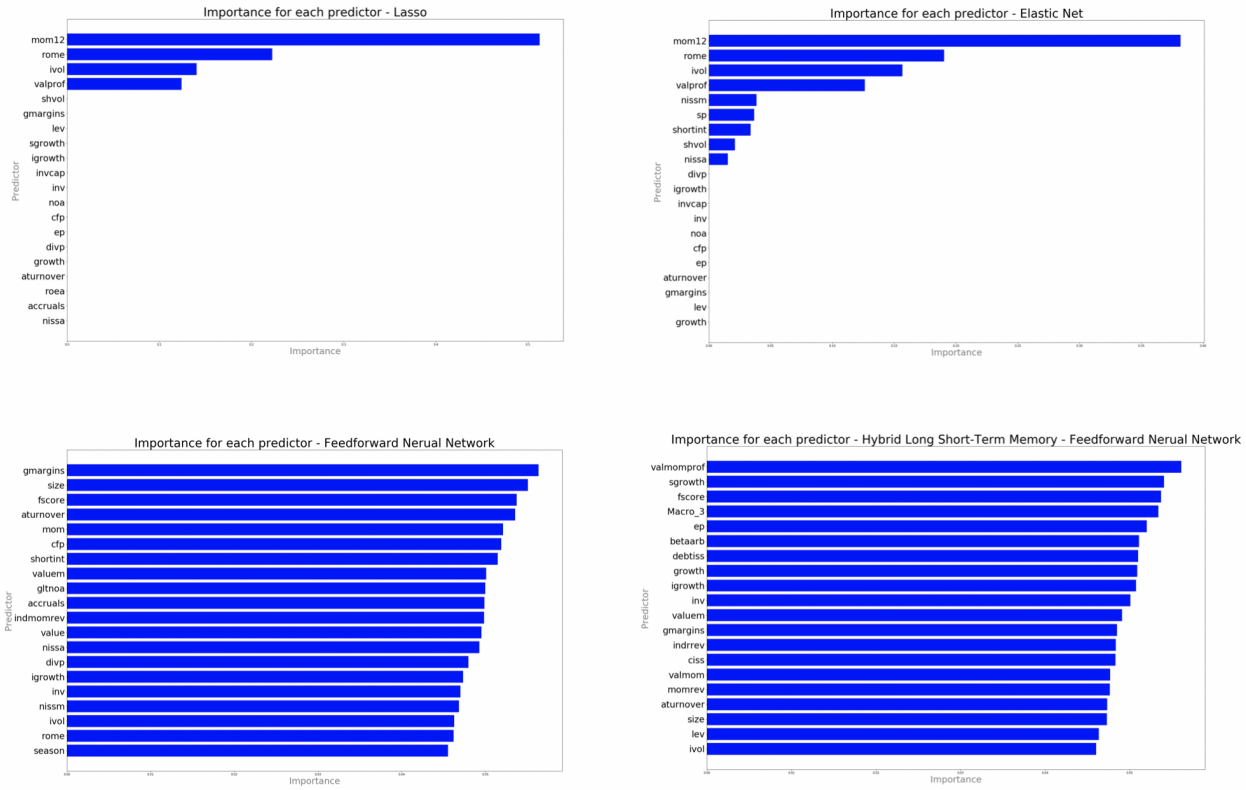


Figure 6: Predictor importance by model

Predictor importance for the top-20 most significant variables in each machine learning method. Covariate importance within each model is normalised to sum to one.

[17] Hou, Xue, and Zhang (2020) empirically show that 65% of the 452 anomalies loss significance upon excluding microcap stocks.

We then move on to investigate the relative importance of individual covariates for the predictive performance of penalised linear models (lasso and elastic net) and neuron networks (FNN3 and LSTM-FNN3) via the methodology described in subsection 2.8. The significance of each predictor is measured by the average magnitude of the gradient with respect to itself. Figure 6 presents the relative importance of the top-20 predictors for each machine learning method. Anomalies are ordered based on their importance so that the most important covariates are on top and the least important covariates are at the bottom. To numerically measure the relative importance of each variable for each specific method, we normalise them to sum to one.

From the upper panel of Figure 6, we can see that penalised linear models (lasso and elastic net) yield sparse representations of a few predictors. The four most important predictors picked by both lasso and elastic net are 1-year momentum (mom12), return on market equity (rome), idiosyncratic volatility (ivol) and value profitability (valprof). Moreover, the elastic net model also includes monthly share issuance (nissm), sales-to-price (sp), short interest (shortint), share volume (shvol), annually share issuance (nissa). However, the covariate importance of neural networks (FNN3 and LSTM-FNN3) are more flatten, where predictability are drawn from a group of weak signals. To summarise, covariates that lie under price trend and liquidity are influential across all methods, which are consistent with results from Gu, Kelly, and Xiu (2018) and Chen, Pelger, and Zhu (2019).

3.4 Economic Grounds of Machine Learning

It is not sufficient to evaluate the economic meaning of machine learning methods if we merely measure predictability of individual stock returns. Instead, we need to evaluate the predictive performance of machine learning methods for aggregated portfolio returns. There are a few advantages to measure predictability of machine learning methods at the portfolio-level.

First, we optimise all of our machine learning models at the stock-level, e.g., minimising cross-sectional individual stock mean square error in each month. Thus, forecasting at the portfolio-level provides additional indirect but valuable evaluation that otherwise would be missed if we only compare R^2_{oos} for each machine learning method. Second, to truly assess the forecasting power of machine learning in return predictability, we need to examine whether aggregated portfolios could produce robust risk-adjusted returns out-of-sample. Third, a good stock-level forecast model is not a guarantee to produce impressive and accurate results at a portfolio-level (Gu, Kelly, and Xiu, 2018). Finally, evaluating at portfolio-level would bring broader economic interest to investors in asset management industry.

I. Machine Learning Portfolios

We devise a unique set of portfolios to exploit machine learning forecasts. At the end of each month, we calculate one-month-ahead out-of-sample stock return forecasts for each machine learning method. Stocks are then sorted into deciles based on each model's forecasts. Both value-weighted and equally-weighted portfolios are then constructed in each month. Unlike Gu, Kelly, and Xiu (2018) and Chen, Pelger, and Zhu (2019), who construct zero-net-investment portfolios that buy the stocks with the highest expected return and sell the ones with the lowest expected return. Here, we only buy the top decile with the highest expected stock returns. There are three reasons for doing that. First, as we exclude stocks with market equity below 0.01% of the aggregate US market cap, the short leg of portfolios do not generate results that are distinct from the long leg. Second, there is ample amount of evidence that anomalies primarily distill their profitability from the short leg of the strategy [18]. Thus, we aim to examine whether our empirical results can challenge these statements. Third, we hope our characteristic portfolios would be of interest to long-only investors such as mutual funds and pension funds.

Table 5 presents results of value-weighted machine learning portfolios. The Sharpe ratio of each machine learning method aligns quite closely with R^2_{oos} of each method reported earlier. The hybrid LSTM-FNN3 outperforms all other machine learning methods, with an annualised out-of-sample Sharpe ratio of 1.091. FNN3 (MICRO) has an annualised Sharpe ratio at 0.964. Penalised linear methods estimated on firm characteristics such as lasso and elastic net also have remarkable results, which earn annualised Sharpe ratio of 1.074 and 0.968 respectively. Consistent with its robust R^2_{oos} reported earlier, PLS estimated on firm characteristics achieves a Sharpe ratio of 0.983. OLSs estimated on FF3, FF5 and FF6 do not yield Sharpe ratio as robust as their R^2_{oos} stated earlier. OLS (FULL) is the worst performing method overall in terms of annualised Sharpe ratio. In general, the results of machine learning models estimated solely on firm characteristics are better than those estimated on the full panel of predictors with exceptions such as LSTM-FNN3, which is in line with R^2_{oos} reported earlier.

As mentioned by Gu, Kelly, and Xiu (2018), given the fact that value-weighted portfolios are less susceptible to trading cost and the fact that our objective function is based on minimising equally predictive error, perhaps studying equal weights is more heuristic in our analysis. Table 6 presents the performance of the equally-weighted machine learning portfolios. The results show that there is no qualitative difference between equal weights and value weights. LSTM-FNN3 and penalised linear

[18] For example, see Hong, Lim, and Stein (2000) and Stambaugh, Yu, and Yuan (2012).

models are still amongst the best performing models. Figure 7 presents the cumulative returns of all machine learning techniques along with a buy-and-hold S&P500 strategy over the out-of-sample period. We can see that all machine learning methods outperform S&P500 over the same period.

Table 5: Performance of the value-weighted machine learning portfolios

	Average Monthly Return %	Annualised Return %	Sharpe Ratio
OLS (FULL)	0.60	6.97	0.456
OLS (MICRO)	1.26	14.75	0.949
OLS (FF3)	1.07	12.33	0.834
OLS (FF5)	1.12	12.67	0.805
OLS (FF6)	1.14	13.17	0.860
Lasso + H (FULL)	1.11	12.69	0.819
Lasso + H (MICRO)	1.23	14.74	1.074
ElasticNet + H (FULL)	1.16	13.64	0.961
ElasticNet + H (MICRO)	1.17	13.77	0.968
PLS (FULL)	1.08	12.54	0.873
PLS (MICRO)	1.29	15.24	0.983
FNN3 (FULL)	1.04	11.45	0.715
FNN3 (MICRO)	1.32	15.56	0.964
LSTM-FNN3 (FULL)	1.26	14.58	1.091

In this table, the performance of prediction-sorted value-weighted portfolios over the 25-year of out-of-sample testing period is presented. We sort stocks into deciles based on their forecasted returns for the next month. We then buy the top decile with the highest expected stock returns. We compare each machine learning method by examining average monthly return, annualised return, and Sharpe ratio.

Table 6: Performance of the equally-weighted machine learning portfolios

	Average Monthly Return %	Annualised Return %	Sharpe Ratio
OLS (FULL)	0.59	6.84	0.447
OLS (MICRO)	1.21	14.03	0.910
OLS (FF3)	1.03	11.78	0.824
OLS (FF5)	1.01	11.43	0.771
OLS (FF6)	1.10	12.58	0.810
Lasso + H (FULL)	1.06	12.02	0.779
Lasso + H (MICRO)	1.18	14.05	1.006
ElasticNet + H (FULL)	1.10	12.84	0.907
ElasticNet + H (MICRO)	1.12	13.13	0.923
PLS (FULL)	1.05	12.12	0.844
PLS (MICRO)	1.21	14.05	0.917
FNN3 (FULL)	1.02	11.31	0.717
FNN3 (MICRO)	1.26	14.74	0.937
LSTM-FNN3 (FULL)	1.26	14.53	1.089

In this table, the performance of prediction-sorted equally-weighted portfolios over the 25-year of out-of-sample testing period is presented. We sort stocks into deciles based on their forecasted returns for the next month. We then buy the top decile with the highest expected stock returns. We compare each machine learning method by examining average monthly return, annualised return, and Sharpe ratio.

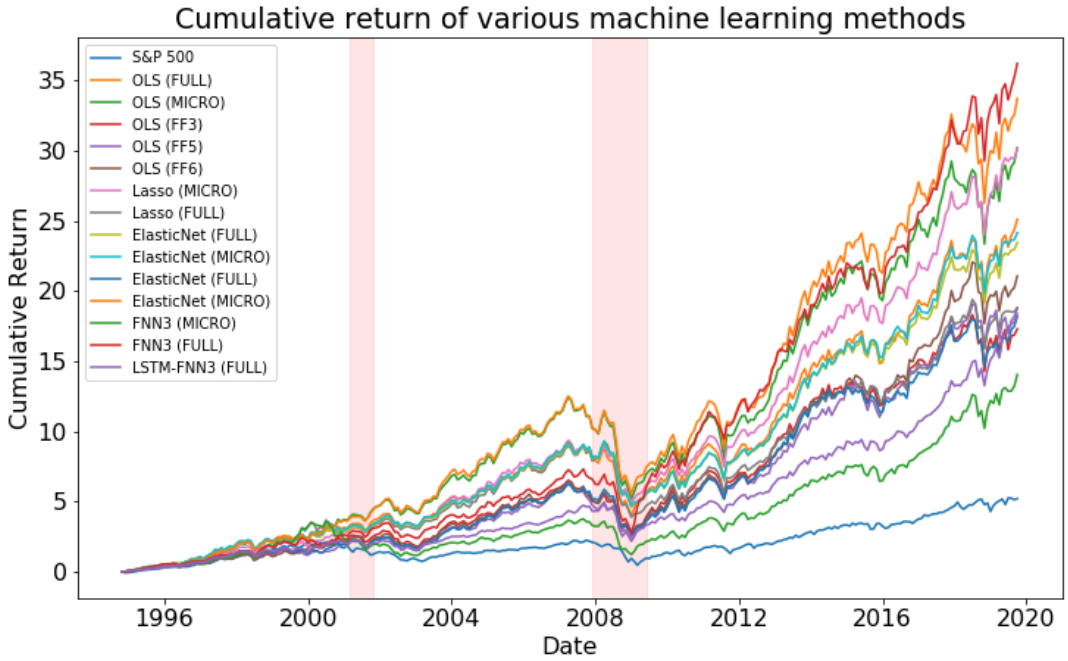


Figure 7: Cumulative returns of value-weighted machine learning portfolios

This figure presents the cumulative returns of value-weighted machine learning portfolios along with S&P500. The shaded periods represent NBER recession dates.

Table 7 and Table 8 present maximum drawdowns, maximum one-month loss, skewness and kurtosis for each machine learning method, for both value-weighted and equally weighted long-only machine learning portfolios defined earlier. The maximum drawdown of a strategy could be formulated as

$$MDD = \max_{0 \le t_1 \le t_2 \le T} (R_{t_1} - R_{t_2}), \tag{30}$$

where R_{t_1} and R_{t_2} are the cumulative returns from date 0 to date t. In our analysis, we define the maximum drawdown of a given method within a year.

Lasso estimated on firm characteristics obtains the smallest maximum drawdown amongst all machine learning models (45.42% for value-weighted strategy and 47.01% for equally-weighted strategy). It then followed by LSTM-FNN3, which has the second smallest maximum drawdown at 49.49% and 51.97% for value and equal weights, respectively. It should be noted that the maximum drawdown (and the maximum one-month loss) experienced for LSTM-FNN3 are less than FNN3 estimated on the full panel of predictors, which are smaller by 10.32% (36.01%) and 12.56% (38.21%)

for value and equal weights, respectively. These results further demonstrate that long short-term memory network has the virtue of detecting business cycle from the economic states, thereby avoiding experiencing extreme crashes. An interesting thing to point out is that the OLS estimated on Fama and French five factors and FF5 along with momentum undergo maximum drawdowns, which are also maximum one-month losses, at 68.37% and 117.29% for equal weights, respectively. On the contrary, penalised linear models such as lasso and elastic net in general have much smaller maximum drawdowns and maximum one-month losses. These finding hint that linear models esimated on pre-specified well-regarded predictors could earn substantial profits while bearing enormous amount of risk. Penalised models estimated on a large number of predictors have the virtue of diversifying the risk under different market conditions. All machine learning methods display moderately negative skewed distribution and excess kurtosis for both value weights and equal weights. Figure 8 presents the maximum drawdowns of value-weighted machine learning portfolios along with a buy-and-hold S&P500 strategy. All machine learning methods experience less maximum drawdowns compared with the market index S&P 500 during the financial crisis, which suggests that machine learning methods are capable of alleviating the downside risk and protect investors from extreme crashes.

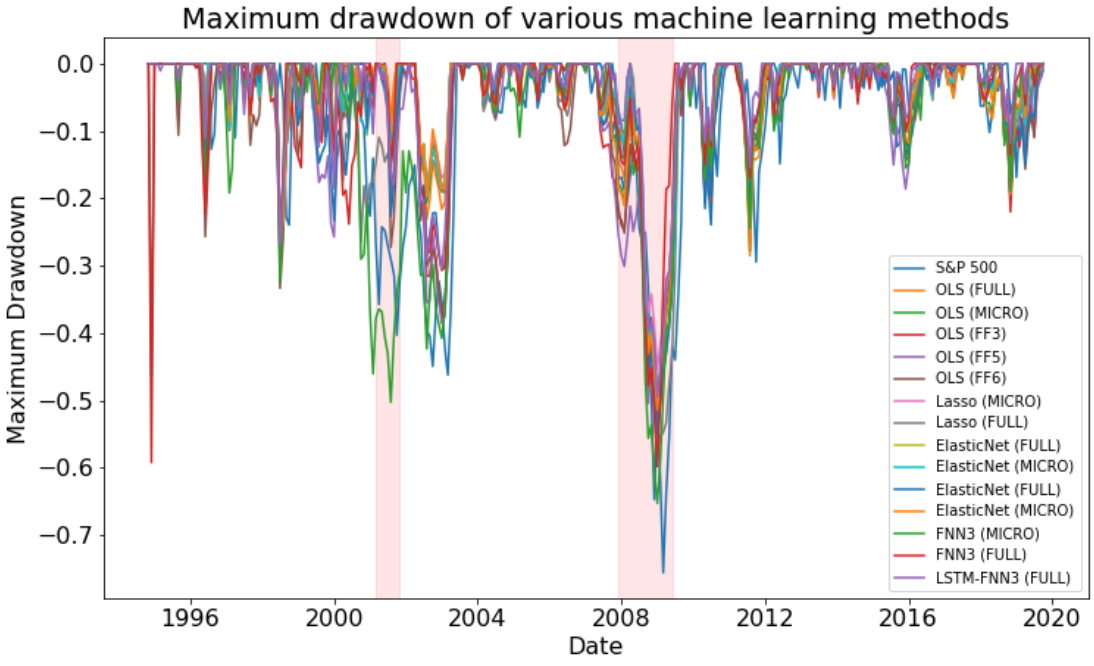


Figure 8: Maximum drawdowns of value-weighted machine learning portfolios

This figure presents the maximum drawdowns of value-weighted machine learning portfolios along with buy-and-hold S&P500. The shaded periods represent NBER recession dates.

Table 7: Maximum drawdowns, maximum one-month loss, skewness and kurtosis of the value-weighted machine learning portfolios

	Maximum Drawdowns %	Maximum One-month Loss %	Skewness	Kurtosis
OLS (FULL)	54.28	21.93	-0.64	2.06
OLS (MICRO)	54.62	22.56	-0.60	2.54
OLS (FF3)	54.08	24.56	-0.65	2.00
OLS (FF5)	55.82	27.59	-0.60	2.36
OLS (FF6)	51.33	46.39	-0.84	1.72
Lasso + H (FULL)	57.87	23.80	-0.74	2.43
Lasso + H (MICRO)	45.42	19.60	-0.55	2.00
ElasticNet + H (FULL)	49.95	21.79	-0.65	1.76
ElasticNet + H (MICRO)	49.62	21.63	-0.62	1.70
PLS (FULL)	55.00	21.96	-0.78	1.42
PLS (MICRO)	51.40	21.42	-0.53	2.11
FNN3 (FULL)	59.81	59.24	-0.06	2.54
FNN3 (MICRO)	65.30	30.04	-0.81	2.19
LSTM-FNN3 (FULL)	49.49	23.23	-0.93	2.73

In this table, we present maximum drawdowns, maximum one-month loss, skewness and kurtosis of value-weighted portfolio.

Table 8: Maximum drawdowns, maximum one-month loss, skewness and kurtosis of the equally-weighted machine learning portfolios

	Maximum Drawdowns %	Maximum One-month Loss %	Skewness	Kurtosis
OLS (FULL)	51.87	21.13	-0.57	1.50
OLS (MICRO)	53.44	22.45	-0.61	2.12
OLS (FF3)	53.01	21.27	-0.60	1.43
OLS (FF5)	68.37	68.37	-0.59	1.86
OLS (FF6)	117.29	117.29	-0.49	1.07
Lasso + H (FULL)	57.90	22.93	-0.62	1.87
Lasso + H (MICRO)	47.01	19.49	-0.57	1.86
ElasticNet + H (FULL)	51.35	21.18	-0.70	1.74
ElasticNet + H (MICRO)	50.87	21.06	-0.68	1.65
PLS (FULL)	53.52	22.97	-0.74	1.55
PLS (MICRO)	51.67	21.37	-0.54	1.79
FNN3 (FULL)	64.53	64.53	-0.25	2.17
FNN3 (MICRO)	62.74	32.66	-0.64	1.64
LSTM-FNN3 (FULL)	51.97	26.32	-0.96	2.09

In this table, we present maximum drawdowns, maximum one-month loss, skewness and kurtosis of equally-weighted portfolio.

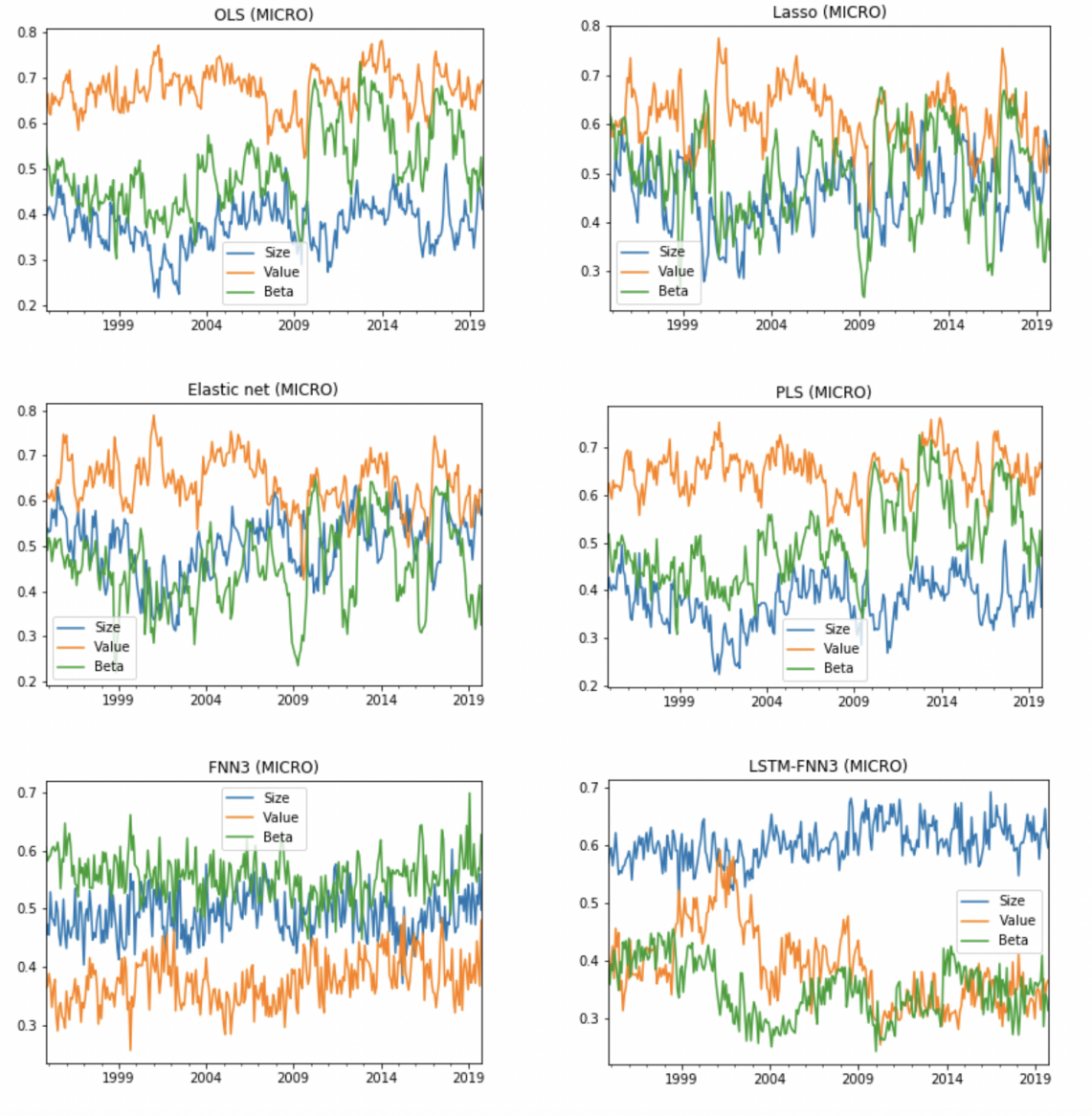


Figure 9: stock anomaly details of value-weighted weighted machine learning portfolios

This figure presents the average magnitude of size, value and beta of the stocks in our value-weighted machine learning portfolios over time. Each anomaly is normalised into range [0,1] in each month.

Figure 9 shows that the average magnitude of size, value and beta of the stocks picked by our value-weighted machine learning portfolios varies over time. Lasso, elastic net and PLS tend to pick stocks with moderate beta, moderate size and relatively high value. FNN3 is prone to pick stocks with moderate size, relatively high beta and relatively low value. LSTM-FNN3 tends to pick stocks with relatively large size and relatively low beta.

3.5 Industrial-level Return Predictability

It is economically meaningful to investigate the predictability, risk-adjusted payoff and potential risk of machine learning methods under different industries. We hope by doing so, investors can have investment targeting towards industries that are more predictable, profitable and less risky. Table 9 presents R^2_{oos}, Sharpe ratio and maximum drawdowns of the value-weighted machine learning portfolios under different industries. We have seven industry categories, they are: A: agriculture, forestry & fishing, manufacturing; B: mining construction; C: transportation & public utilities; D: wholesale trade & retail trade; E: finance, insurance & real estate; F: service; G: tech & internet.

By looking at Panel A of Table 11, fiance, insurance & real estate (E) in general has higher R^2_{oos} across various machine learning methods. This might be due to the fact that fiance, insurance & real estate are the most sensitive industry to our predictors, which are either accounting or economic variables. Tech & internet is the least predictable industry overall.

Panel B of Table 11 shows that transportation & public utilities industry has the highest average Sharpe ratio at 0.888. It then followed by agriculture, forestry & fishing, manufacturing, with an Sharpe ratio at 0.818. Apart from being the least predictable industry, tech & internet also has the lowest Sharpe ratio amongst all industries.

Panel C of Table 8 reports the Maximum Drawdowns of a set of machine learning portfolios for different industries. Transportation & public utilities has the lowest average maximum drawdown at 55.21%. Mining & construction has a relatively large downside risk, with an average maximum drawdown at 285.46%. Unsurprisingly, tech & interest commands the largest downside risk. These results further demonstrate that machine learning techniques are able to detect the downside risk for different industries, which effectively mitigate the opaque nature of machine learning approaches in an interpretable way.

Table 9: Industry name for each code

A	Agriculture, Forestry & Fishing, Manufacturing
B	Mining & Construction
C	Transportation & Public Utilities
D	Wholesale Trade & Retail Trade
E	Finance, Insurance & Real Estate
F	Service
G	Tech & Internet

Table 10: R^2_{oos}, Sharpe ratio and maximum drawdowns of the value-weighted machine learning portfolios under different industries

		A	B	C	D	E	F	G
	Panel A:							
R^2_{oos} (percentage)	OLS (FULL)	-inf	-inf	-inf	-inf	-inf	-inf	-inf
	OLS (MICRO)	0.70	0.15	1.01	0.60	1.11	0.49	0.35
	Lasso (FULL)	0.62	-0.36	0.44	0.84	0.65	0.33	-0.45
	Lasso (MICRO)	0.61	0.30	1.01	0.59	1.13	0.47	0.85
	ElasticNet (FULL)	-1.28	-2.56	-2.58	-1.42	-2.14	-0.66	-3.15
	ElasticNet (MICRO)	0.74	0.18	1.04	0.80	1.27	0.53	0.41
	PLS (FULL)	-76.42	-97.03	-107.31	-77.12	-89.56	-50.99	-80.31
	PLS (MICRO)	0.68	0.17	1.02	0.58	1.14	0.48	0.27
	FNN3 (FULL)	-0.24	-0.47	-1.04	-0.54	-1.01	-0.40	-1.60
	FNN3 (MICRO)	0.39	-0.69	0.12	0.35	0.73	-2.2 $\times 10^7$	-1.16
	LSTM-FNN3 (FULL)	0.51	-0.50	0.34	0.52	0.89	0.12	-0.94
	Panel B:							
Sharpe Ratio	OLS (FULL)	0.478	0.225	0.525	0.433	0.426	0.328	0.136
	OLS (MICRO)	0.822	0.673	0.925	0.938	0.806	0.785	0.262
	Lasso (FULL)	0.794	0.534	0.855	0.552	0.685	0.798	0.143
	Lasso (MICRO)	0.970	0.636	0.847	1.150	0.871	0.896	0.109
	ElasticNet (FULL)	0.883	0.453	0.976	0.754	0.706	0.795	0.108
	ElasticNet (MICRO)	0.901	0.506	0.912	0.762	0.655	0.869	0.179
	PLS (FULL)	0.813	0.446	0.941	0.486	0.976	0.789	0.136
	PLS (MICRO)	0.817	0.670	0.948	0.975	0.847	0.913	0.132
	FNN3 (FULL)	0.794	0.562	0.944	0.670	0.804	0.806	0.085
	FNN3 (MICRO)	0.696	0.246	0.980	0.611	0.543	0.522	0.027
	LSTM-FNN3 (FULL)	1.028	0.677	0.920	0.818	0.843	0.763	0.437
	Average	0.818	0.512	0.888	0.741	0.742	0.751	0.159
	Panel C:							
Maximum Drawdowns (percentage)	OLS (FULL)	58.37	80.39	45.54	39.66	52.45	44.78	89.21
	OLS (MICRO)	61.01	69.93	54.52	35.03	57.69	43.05	132.64
	Lasso (FULL)	56.73	66.09	59.94	48.68	64.18	37.26	9781.56
	Lasso (MICRO)	45.09	497.76	60.21	31.54	57.11	179.72	2071.59
	ElasticNet (FULL)	50.33	69.59	52.90	44.51	57.00	39.60	247.71
	ElasticNet (MICRO)	50.65	312.18	56.48	41.23	59.27	34.54	813.99
	PLS (FULL)	60.04	75.64	52.68	52.94	44.42	97.38	116.39
	PLS (MICRO)	56.58	68.86	57.55	45.72	56.93	39.45	280.93
	FNN3 (FULL)	71.41	69.12	59.50	58.02	70.23	53.00	166.60
	FNN3 (MICRO)	64.44	1741.97	48.29	179.31	71.04	84.59	1049.55
	LSTM-FNN3 (FULL)	39.28	88.55	59.66	47.14	64.23	52.98	108.56
	Average	55.81	285.46	55.21	56.71	59.50	64.21	1350.79

In this table, we present R^2_{oos}, Sharpe ratio and maximum drawdowns for various value-weighted machine learning portfolios under different industries. "A" represents for Agriculture, Forestry & Fishing, Manufacturing; "B" represents for Mining & Construction; "C" represents for Transportation & Public Utilities; "D" represents for Wholesale Trade & Retail Trade; "E" represents for Finance, Insurance, & Real Estate; "F" represents for Service; "G" represents for Tech & Internet.

I. Industry-winner Machine Learning Portfolios

Inspired by Avramov, Cheng, and Metzker (2019), who construct winner and loser portfolios from the intra-industry, we follow a similar fashion by building industry-winner machine learning portfolios. At the end of each month, we calculate one-month-ahead out-of-sample stock return predictions for each machine learning method. Unlike the machine learning portfolios described earlier, where we sort stocks into deciles based on each model's forecasts from a broad market perspective. Here, stocks from different industries are sorted into deciles separately based on each model's predictions. We then buy the top decile in each industry with the highest expected stock returns. Each industry is value-weighted by their overall capitalisation. We devise such strategies with the hope that portfolio risk could be diversified by involvement of stocks from all industries.

Table 11: Performance of the value-weighted industry-winner machine learning portfolios

	Average monthly return (%)	Sharpe Ratio	Maximum Drawdowns (%)
OLS (FULL)	0.61	0.475	57.14
OLS (MICRO)	1.17	0.858	55.65
Lasso + H (FULL)	1.16	0.886	55.41
Lasso + H (MICRO)	1.21	1.050	49.46
ElasticNet + H (FULL)	1.12	0.951	50.83
ElasticNet + H (MICRO)	1.13	0.959	51.22
PLS (FULL)	1.14	0.927	54.20
PLS (MICRO)	1.30	0.989	53.84
FNN3 (FULL)	1.03	0.733	63.45
FNN3 (MICRO)	1.27	0.924	63.29
LSTM-FNN3 (FULL)	1.06	1.006	49.72

In this table, we report the average monthly return, Sharpe ratio, and the maximum drawdown for each machine learning method.

Table 11 reports the performance of the value-weighted industry-winner machine learning portfolios. The results show no qualitative difference as shown in Table 5 and Table 7. Thus, long-only investors could adopt either our machine learning portfolios or industry-winner machine learning portfolios to time the market correctly and make profits from it.

3.6 Time-varying Return Predictability

Apart from investigating whether the cross-sectional return predictability of machine learning methods differ under various industries, it is valuable to understand whether stock return predictability changes over time. Avramov, Cheng, and Metzker (2019) state that the proposed deep learning models from Gu, Kelly, and Xiu (2018) and Chen, Pelger, and Zhu (2019) are more profitable during periods of high investor sentiment, high market vololatility, and low market liquidity. By following their track, we investigate both the R^2_{oos} and the payoff under different market conditions. In addition, we also examine return predictability amid economic recession, post 2001 decimalisation and post 2008 financial crisis.

I. Return Predictability sorted by Market States

In this section, we examine the predictability and payoff of machine learning portfolios in sub-periods sorted by market state variables. The market state variables we consider are: (1) implied market volatility index (VXOCLSx), defined as the CBOE S&P 100 volatility index; (2) consumer sentiment index (UMCSENTx); (3) civilian unemployment rate (UNRATE); and (4) momentum (MOM), defined as the cumulated past performance in the previous six months by skipping the most recent month. Following Avramov, Cheng, and Metzker (2019), the out-of-sample testing period is divided into two sub-periods, i.e., we have high versus low implied market volatility (consumer sentiment index, unemployment rate and momentum) based on the median breakpoint of UMCSENTx (VXOCLSx, UNRATE and MOM) over the full out-of-sample period.

Table 12 reports the R^2_{oos} of value-weighted long-only machine learning portfolios sorted by implied market volatility, consumer sentiment index, unemployment rate and momentum. Several finding are worth pointing out. First, R^2_{oos} is much higher in general when the implied market volatility is high. Second, it's interesting to note that machine learning methods have better predictive results when the consumer index is low. Third, high unemployment rate leads to higher R^2_{oos}. The findings from the second and third points lead us to gauge that machine learning techniques can generate more predictive results amid economic recession. Finally, low momentum results in higher R^2_{oos}, which further reinforces our hypothesis.

Table 13 presents the average monthly returns of value-weighted long-only machine learning portfolios sorted by implied market volatility, consumer sentiment index, unemployment rate and momentum. The average monthly payoff results are consistently with R^2_{oos} as shown in Table 12. In general, machine learning methods have higher average monthly payoff during periods of high

implied volatility [19], low consumer sentiment index, high unemployment rate and low momentum. An economic recession typically represents a market condition with high implied volatility, low consumer sentiment index, high unemployment rate and low momentum. Thus, it is worth investigating whether machine learning techniques can generate better results amid economic recession, which leads us to the next section.

Table 12: R^2_{oos} (percentage) of value-weighted machine learning portfolios sorted by market state variables

	UMCSENTx		VXOCLSx		UNRATE		MOM	
	Low	High	Low	High	Low	High	Low	High
OLS (FULL)	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf
OLS (MICRO)	0.35	1.05	1.38	0.41	0.55	0.97	0.75	0.57
OLS (FF3)	0.33	1.09	1.51	0.37	0.53	1.03	0.79	0.49
OLS (FF5)	0.37	1.10	1.52	0.40	0.55	1.07	0.83	0.50
OLS (FF6)	0.38	1.11	1.53	0.42	0.60	1.01	0.82	0.55
Lasso + H (FULL)	0.06	0.94	1.86	-0.07	0.44	0.59	0.71	-0.002
Lasso + H (MICRO)	0.40	0.95	1.35	0.39	0.59	0.82	0.72	0.56
ElasticNet + H (FULL)	-2.17	-0.82	1.17	-2.60	-0.52	-3.4	-1.42	-1.71
ElasticNet + H (MICRO)	0.44	1.10	1.57	0.43	0.68	0.92	0.85	0.56
PLS (FULL)	-88.45	-66.12	1.62	-109.98	-51.11	-127.59	-0.76	-253.19
PLS (MICRO)	0.34	1.05	1.37	0.41	0.54	0.98	0.76	0.53
FNN3 (FULL)	-0.36	0.87	0.93	-0.04	-0.06	0.82	0.39	-0.08
FNN3 (MICRO)	0.13	-1.15	1.45	-1.3	-1.05	0.54	0.31	-2.35
LSTM-FNN3 (FULL)	-0.17	0.97	1.15	0.08	0.12	0.91	0.53	0.07

In this table, we present R^2_{oos} of value-weighted machine learning portfolios sorted by different market state variables, they are: implied market volatility (UMCSENTx), consumer sentiment index (VXOCLSx), unemployment rate (UNRATE), and momentum (MOM).

Table 13: Average monthly returns (percentage) of value-weighted machine learning portfolios sorted by market state variables

	UMCSENTx		VXOCLSx		UNRATE		MOM	
	Low	High	Low	High	Low	High	Low	High
OLS (FULL)	0.50	0.70	0.65	0.55	0.53	0.73	0.63	0.50
OLS (MICRO)	1.10	1.42	1.36	1.16	1.15	1.51	1.29	1.16
OLS (FF3)	1.10	1.05	1.03	1.12	0.99	1.26	1.08	1.06
OLS (FF5)	1.17	1.06	1.11	1.12	0.92	1.54	1.16	1.00
OLS (FF6)	1.14	1.15	1.23	1.05	1.06	1.32	1.17	1.08
Lasso + H (FULL)	0.82	1.41	1.26	0.97	1.17	0.98	1.19	0.89
Lasso + H (MICRO)	1.06	1.40	1.32	1.14	1.18	1.35	1.28	1.09
ElasticNet + H (FULL)	0.98	1.34	1.31	1.01	1.16	1.16	1.25	0.90
ElasticNet + H (MICRO)	0.98	1.36	1.31	1.03	1.17	1.16	1.25	0.93
PLS (FULL)	1.07	1.10	1.25	0.92	1.06	1.14	1.14	0.93
PLS (MICRO)	1.15	1.44	1.35	1.24	1.19	1.52	1.33	1.20
FNN3 (FULL)	0.81	1.26	1.22	0.86	0.96	1.19	1.09	0.88
FNN3 (MICRO)	1.35	1.30	1.25	1.40	1.08	1.86	1.31	1.35
LSTM-FNN3 (FULL)	0.91	1.21	1.19	0.93	0.96	1.27	1.13	0.86

In this table, we present average monthly return of value-weighted machine learning portfolios sorted by different market state variables, they are: implied market volatility (UMCSENTx), consumer sentiment index (VXOCLSx), unemployment rate (UNRATE), and momentum (MOM).

[19] Our finding is consistent with Avramov, Cheng, and Metzker (2019) and Nagel (2012).

II. Return Predictability amid Economic Recession

Our hypothesis in the last section motivates us to investigate return predictability amid economic recession. Table 14 presents R^2_{oos} and average monthly returns of value-weighted long-only machine learning portfolios amid two major economic recessions in the US - the 2001 dot-com bubble and the 2008 financial crisis. We find that both major economic recession do not yield robust predictive results - especially during the 2008 financial crisis, the R^2_{oos} for all machine learning methods are negative. As the R^2_{oos} reflects the statistical perspective of predictive accuracy, it does not truly represent the economic meaning of our machine learning methods. Knowing that the average monthly return of the S&P500 during the 2001 dot-com bubble is -2.40%, we find all machine learning portfolios have positive average monthly payoff during the 2001 dot-com bubble; With an average monthly loss at 2.02% for the S&P500 index during the 2008 financial crisis, all machine learning portfolios deliver smaller loss except for lasso estimated on the full panel of predictors.

Therefore, we conclude that machine learning methods do not yield robust predictive results in terms of R^2_{oos} amid the 2001 dot-com bubble and the financial-crisis, though they are able to experience less average monthly loss compared with the S&P500 index.

Table 14: R^2_{oos} and average monthly returns of value-weighted machine learning portfolios amid economic recession

	2001 dot-com bubble		2008 financial crisis	
	R^2_{oos} (%)	Average Monthly Return (%)	R^2_{oos} (%)	Average Monthly Return (%)
OLS (FULL)	-inf	1.56	-inf	-1.55
OLS (MICRO)	0.14	2.00	-2.02	-1.86
OLS (FF3)	0.09	1.05	-1.85	-1.02
OLS (FF5)	0.23	1.81	-1.78	-1.03
OLS (FF6)	0.14	1.13	-1.89	-1.39
Lasso + H (FULL)	1.40	0.77	-4.86	-2.46
Lasso + H (MICRO)	0.15	1.73	-1.62	-1.44
ElasticNet + H (FULL)	0.34	1.13	-18.75	-1.88
ElasticNet + H (MICRO)	0.25	0.99	-1.61	-1.91
PLS (FULL)	-5.85	0.98	-655.61	-1.64
PLS (MICRO)	0.20	2.08	-2.00	-1.55
FNN3 (FULL)	-0.12	1.85	-11.47	0.12
FNN3 (MICRO)	-0.33	0.32	-3.26	-1.52
LSTM-FNN3 (FULL)	-0.21	0.42	-2.95	-1.16

In this table, we present R^2_{oos} and average monthly returns of value-weighted long-only machine learning portfolio amid two major economic recession (2001 dot-com bubble and 2008 financial crisis) in the U.S. The average monthly return of S&P500 during the 2001 dot-com bubble and the 2008 financial crisis are -2.40% and -2.02%, respectively. The time horizon for the 2001 dot-com bubble is from March 2001 to November 2001, and the time horizon for the 2008 financial crisis is from December 2007 to June 2009.

III. Return Predictability post 2001 Decimalisation

The U.S. stock market experienced plentiful structure change since the millennium, such as the advent of the 2001 decimalisation. Chordia, Subrahmanyam, and Tong (2014) report that the majority of anomalies have attenuated, and the average returns of anomaly-based strategies have approximately halved since the decimalisation in January 2001 due to increased market liquidity and arbitrage activities. Chakravarty, Panchapagesan, and Wood (2005) show that the trading cost declines as a whole, with improvements in most partitions across order size, firm size, and manager style. Thus, it is worth knowing how our machine learning portfolios perform post 2001 decimalisation.

Table 15 presents the R^2_{oos} and average monthly returns of value-weighted long-only machine learning portfolios before and after 2001 decimalisation. Our findings are in line with Chordia, Subrahmanyam, and Tong (2014), the R^2_{oos} is higher for all machine learning portfolios before 2001 decimalisation. The average monthly returns approximately halve after 2001 decimalisation for all machine learning portfolios.

Table 15: R^2_{oos} and average monthly returns of value-weighted machine learning portfolios post 2001 decimalisation

	pre 2001 decimalisation		post 2001 decimalisation	
	R^2_{oos} (%)	Average Monthly Return (%)	R^2_{oos} (%)	Average Monthly Return (%)
OLS (FULL)	-inf	2.03	-inf	0.91
OLS (MICRO)	1.10	2.20	0.42	0.93
OLS (FF3)	0.96	1.73	0.52	0.84
OLS (FF5)	0.96	1.57	0.57	0.96
OLS (FF6)	1.03	1.68	0.54	0.96
Lasso + H (FULL)	1.11	2.02	0.07	0.79
Lasso + H (MICRO)	0.99	1.93	0.45	0.99
ElasticNet + H (FULL)	1.13	1.96	-3.29	0.88
ElasticNet + H (MICRO)	1.01	2.01	0.60	0.87
PLS (FULL)	0.69	1.61	-130.52	0.90
PLS (MICRO)	1.08	2.16	0.43	0.99
FNN3 (FULL)	0.92	1.70	-0.21	0.80
FNN3 (MICRO)	0.99	1.87	-1.51	1.13
LSTM-FNN3 (FULL)	0.96	1.59	-0.003	0.87

In this table, we present R^2_{oos} and average monthly returns of value-weighted long-only machine learning portfolio pre and post 2001 decimalisation.

IV. Return Predictability post 2008 Financial Crisis

A set of regulatory measures such as Dodd-Frank Wall Street Reform and Consumer Protection Act (Congress 2010) are introduced to regulate the activities of financial institutions and better protect consumers. Given these activities result in structure change in the financial market, we find it important to examine return predictability post-financial crisis.

Table 16 presents the R^2_{oos} and average monthly returns of value-weighted long-only machine learning portfolios before and after financial crisis. We find that all machine learning portfolios have significant R^2_{oos} that range from 1.37% to 2.52% (after excluding OLS (FULL)) after the 2008 financial crisis. The average monthly payoff also have slightly increased since the 2008 financial crisis. Although this might seem contradictory to the results from the last section, these empirical findings demonstrate that machine learning strategies remain effective and profitable in recent years.

Table 16: R^2_{oos} and average monthly returns of value-weighted machine learning portfolios post 2008 financial crisis

	pre 2008 financial crisis		post 2008 financial crisis	
	R^2_{oos} (%)	Average Monthly Return (%)	R^2_{oos} (%)	Average Monthly Return (%)
OLS (FULL)	-inf	1.16	-inf	1.25
OLS (MICRO)	0.37	1.27	1.58	1.25
OLS (FF3)	0.28	0.96	1.83	1.24
OLS (FF5)	0.30	0.94	1.88	1.35
OLS (FF6)	0.33	0.97	1.86	1.39
Lasso + H (FULL)	-0.15	1.10	2.24	1.13
Lasso + H (MICRO)	0.41	1.18	1.37	1.30
ElasticNet + H (FULL)	-2.99	1.15	2.52	1.18
ElasticNet + H (MICRO)	0.43	1.15	1.66	1.20
PLS (FULL)	-106.91	0.93	2.39	1.29
PLS (MICRO)	0.35	1.30	1.60	1.29
FNN3 (FULL)	-1.45	1.19	2.07	1.51
FNN3 (MICRO)	-0.27	0.78	1.64	1.39
LSTM-FNN3 (FULL)	-0.12	0.89	1.78	1.30

In this table, we present R^2_{oos} and average monthly returns of value-weighted long-only machine learning portfolio pre and post 2008 financial crisis.

4 Conclusion

In this thesis, we present a comparative analysis for various machine learning techniques in predicting cross-sectional US stock returns upon excluding small-cap stocks. Four machine learning techniques are employed with increasing complexity but decreasing interpretability. These four methods are: penalised linear models that include lasso and elastic net, partial least squares, feedforward neural network and feedforward neural network with long short-term memory.

We use R^2_{oos} to statistically assess the predictive accuracy for each technique. Our empirical results show that all machine learning methods (lasso, elastic net, PLS, FNN3, LSTM-FNN3) estimated only on firm characteristics yield robust R^2_{oos} on all stocks, where LSTM-FNN3 outperform all other methods, with an R^2_{oos} at 0.84%. Apart from LSTM-FNN3, all machine learning methods estimated on both firm characteristics and macroeconomic predictors are not able to obtain great R^2_{oos}. Predictive accuracy from top-100 stocks are better than those from all stocks and bottom-100 stocks. These results demonstrate that cross-sectional return predictability of machine learning methods is not by picking stocks that are small-cap, distressed or limit-to-arbitrage.

We find anomalies that lie under price trend such as 1-year momentum, and liquidity such as share volume and annual share issuance are the influential predictors across all machine learning techniques.

R^2_{oos} is not sufficient if we want to evaluate the economic gains of our machine learning methods. We construct characteristic long-only machine learning portfolios to assess economic grounds of our machine learning methods. Empirical results show that annualised Sharpe ratio are consistent with R^2_{oos} for all machine learning methods. LSTM-FNN3 and lasso (MICRO) are the best performing techniques, with Sharpe ratio at 1.091 and 1.074 for value-weighted machine learning portfolios respectively. There is no qualitative difference in terms of Sharpe ratio for equal weights and value weights. Penalised linear methods such as lasso and elastic net estimated only firm characteristics, and LSTM-FNN3 have the smallest maximum drawdowns, at 45.42%, 49.62%, and 49.49% for value-weighted machine learning portfolios. It should be note that the maximum drawdown of FNN3 is approximately 10% higher than LSTM-FNN3, which demonstrates that long short-term memory network is capable of detecting business cycle, thereby avoiding undergoing extreme crashes.

It is economically meaningful to evaluate predictive accuracy under different industries. Tech & Interest is the least predictive and least profitable industry for all our machine learning portfolios. It also bears extreme downside risk compared with other industries. Industry-winner machine learning

portfolios as detailed in paragraph I. are also as predictable and lucrative as our machine learning portfolios.

We find it valuable to understand whether stock return predictability varies over time. Results show that machine learning methods have high R^2_{oos} and average monthly payoffs during periods of high implied volatility, low consumer sentiment index, high employment rate and low market momentum. We also show that machine learning methods do not reach robust predictive outcomes in terms of R^2_{oos} amid the 2001 dot-com bubble and the 2008 financial crisis, though they are able to undergo less average monthly loss compared with the S&P500 index. Moreover, pre 2001 decimalisation reports better predictive accuracy for all machine learning methods. Last but not least, machine learning-based strategies remain effective and profitable in recent years.

In our future work, it is valuable to appraise the predictive performance of machine learning techniques in the presence of trading costs. Moreover, it is worth conducting statistical experiments to detect predictive performance and stability of machine learning methods.

To conclude, we provide a great deal of evidence of statistical importance and economic meaning of machine learning methods. Lasso and LSTM-FNN3 are the best performing machine learning models overall. Anomalies that lie under price trend and liquidity remain effective for our machine learning methods. Machine learning-based strategies, as demonstrated by our long-only machine learning portfolios, unfold considerable perspective for asset management. We find that our machine learning-based strategies in general generate robust payoffs and display reasonable downside risk. Therefore, our long-only machine learning-based strategies show great potential for stock picking and market timing, which would be of value to long-only financial institutions.

References

[1] Charu C Aggarwal et al. *Neural networks and deep learning*. Springer, 2018.

[2] Doron Avramov, Si Cheng, and Lior Metzker. 'Machine learning versus economic restrictions: Evidence from stock return predictability'. In: *Available at SSRN 3450322* (2019).

[3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 'Learning long-term dependencies with gradient descent is difficult'. In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166.

[4] Leo Breiman. 'Bagging predictors'. In: *Machine learning* 24.2 (1996), pp. 123–140.

[5] Svetlana Bryzgalova, Markus Pelger, and Jason Zhu. 'Forest through the trees: Building cross-sections of stock returns'. In: *Available at SSRN 3493458* (2019).

[6] John Y Campbell and Samuel B Thompson. 'Predicting excess stock returns out of sample: Can anything beat the historical average?' In: *The Review of Financial Studies* 21.4 (2008), pp. 1509–1531.

[7] Sugato Chakravarty, Venkatesh Panchapagesan, and Robert A Wood. 'Did decimalization hurt institutional investors?' In: *Journal of Financial Markets* 8.4 (2005), pp. 400–420.

[8] Luyang Chen, Markus Pelger, and Jason Zhu. 'Deep learning in asset pricing'. In: *Available at SSRN 3350138* (2019).

[9] Alex Chinco, Adam D Clark-Joseph, and Mao Ye. 'Sparse signals in the cross-section of returns'. In: *The Journal of Finance* 74.1 (2019), pp. 449–492.

[10] Tarun Chordia, Avanidhar Subrahmanyam, and Qing Tong. 'Have capital market anomalies attenuated in the recent era of high liquidity and trading activity?' In: *Journal of Accounting and Economics* 58.1 (2014), pp. 41–58.

[11] United States. Congress. *Dodd-Frank Wall Street Reform and Consumer Protection Act: Conference Report (to Accompany HR 4173)*. Vol. 111. 517. US Government Printing Office, 2010.

[12] Sijmen De Jong. 'SIMPLS: an alternative approach to partial least squares regression'. In: *Chemometrics and intelligent laboratory systems* 18.3 (1993), pp. 251–263.

[13] Matthew F Dixon and Nicholas G Polson. 'Deep Fundamental Factor Models'. In: *arXiv preprint arXiv:1903.07677* (2019).

[14] Eugene F Fama and Kenneth R French. 'The cross-section of expected stock returns'. In: *the Journal of Finance* 47.2 (1992), pp. 427–465.

[15] Eugene F Fama and Kenneth R French. 'Dissecting anomalies'. In: *The Journal of Finance* 63.4 (2008), pp. 1653–1678.

[16] Eugene F Fama and Kenneth R French. 'A five-factor asset pricing model'. In: *Journal of financial economics* 116.1 (2015), pp. 1–22.

[17] Guanhao Feng, Jingyu He, and Nicholas G Polson. 'Deep learning for predicting asset returns'. In: *arXiv preprint arXiv:1804.09314* (2018).

[18] Joachim Freyberger, Andreas Neuhierl, and Michael Weber. 'Dissecting characteristics nonparametrically'. In: *The Review of Financial Studies* 33.5 (2020), pp. 2326–2377.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[20] Amit Goyal and Ivo Welch. 'Predicting the equity premium with dividend ratios'. In: *Management Science* 49.5 (2003), pp. 639–654.

[21] Shihao Gu, Bryan T Kelly, and Dacheng Xiu. 'Autoencoder asset pricing models'. In: (2019).

[22] Shihao Gu, Bryan Kelly, and Dacheng Xiu. *Empirical asset pricing via machine learning*. Tech. rep. National Bureau of Economic Research, 2018.

[23] James B Heaton, Nick G Polson, and Jan Hendrik Witte. 'Deep learning for finance: deep portfolios'. In: *Applied Stochastic Models in Business and Industry* 33.1 (2017), pp. 3–12.

[24] Eric Hillebrand, Tae-Hwy Lee, and Marcelo C Medeiros. 'Bagging constrained equity premium predictors'. In: *Essays in Nonlinear Time Series Econometrics (Festschrift for Timo Teräsvirta)* (2013).

[25] Harrison Hong, Terence Lim, and Jeremy C Stein. 'Bad news travels slowly: Size, analyst coverage, and the profitability of momentum strategies'. In: *The Journal of Finance* 55.1 (2000), pp. 265–295.

[26] Enguerrand Horel and Kay Giesecke. 'Towards explainable ai: Significance tests for neural networks'. In: *arXiv preprint arXiv:1902.06021* (2019).

[27] Kewei Hou, Chen Xue, and Lu Zhang. 'Replicating anomalies'. In: *The Review of Financial Studies* 33.5 (2020), pp. 2019–2133.

[28] Peter J Huber. *Robust statistical procedures*. SIAM, 1996.

[29] Atsushi Inoue and Lutz Kilian. 'In-sample or out-of-sample tests of predictability: Which one should we use?' In: *Econometric Reviews* 23.4 (2005), pp. 371–402.

[30] Sergey Ioffe and Christian Szegedy. 'Batch normalization: Accelerating deep network training by reducing internal covariate shift'. In: *arXiv preprint arXiv:1502.03167* (2015).

[31] Bryan T Kelly, Seth Pruitt, and Yinan Su. 'Instrumented principal component analysis'. In: *Available at SSRN 2983919* (2017).

[32] Bryan T Kelly, Seth Pruitt, and Yinan Su. 'Characteristics are covariances: A unified model of risk and return'. In: *Journal of Financial Economics* 134.3 (2019), pp. 501–524.

[33] Bryan Kelly and Seth Pruitt. 'Market expectations in the cross-section of present values'. In: *The Journal of Finance* 68.5 (2013), pp. 1721–1756.

[34] Bryan Kelly and Seth Pruitt. 'The three-pass regression filter: A new approach to forecasting using many predictors'. In: *Journal of Econometrics* 186.2 (2015), pp. 294–316.

[35] Diederik P Kingma and Jimmy Ba. 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980* (2014).

[36] Ralph SJ Koijen and Stijn Van Nieuwerburgh. 'Predictability of returns and cash flows'. In: *Annu. Rev. Financ. Econ.* 3.1 (2011), pp. 467–491.

[37] Serhiy Kozak, Stefan Nagel, and Shrihari Santosh. *Shrinking the cross section*. Tech. rep. National Bureau of Economic Research, 2017.

[38] Edward Leung et al. 'The Promises and Pitfalls of Machine Learning for Predicting Stock Returns'. In: *Available at SSRN* (2020).

[39] Jonathan Lewellen. 'The cross section of expected stock returns'. In: *Forthcoming in Critical Finance Review* (2014).

[40] Timothy Masters. *Practical neural network recipes in C++*. Morgan Kaufmann, 1993.

[41] Michael W McCracken and Serena Ng. 'FRED-MD: A monthly database for macroeconomic research'. In: *Journal of Business & Economic Statistics* 34.4 (2016), pp. 574–589.

[42] Marcial Messmer. 'Deep learning and the cross-section of expected returns'. In: *Available at SSRN 3081555* (2017).

[43] Stefan Nagel. 'Evaporating liquidity'. In: *The Review of Financial Studies* 25.7 (2012), pp. 2005–2039.

[44] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 'On the Difficulties of Training Recurrent Neural Networks'. In: *Icml,(2)* (2013), pp. 1–9.

[45] Simo Puntanen and George PH Styan. 'The equality of the ordinary least squares estimator and the best linear unbiased estimator'. In: *The American Statistician* 43.3 (1989), pp. 153–161.

[46] David E Rapach, Jack K Strauss, and Guofu Zhou. 'Out-of-sample equity premium prediction: Combination forecasts and links to the real economy'. In: *The Review of Financial Studies* 23.2 (2010), pp. 821–862.

[47] Tom Schaul et al. 'Natural evolution strategies'. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 949–980.

[48] Justin Sirignano, Apaar Sadhwani, and Kay Giesecke. 'Deep learning for mortgage risk'. In: *arXiv preprint arXiv:1607.02470* (2016).

[49] Nitish Srivastava et al. 'Dropout: a simple way to prevent neural networks from overfitting'. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[50] Robert F Stambaugh, Jianfeng Yu, and Yu Yuan. 'The short of it: Investor sentiment and anomalies'. In: *Journal of Financial Economics* 104.2 (2012), pp. 288–302.

[51] Robert Tibshirani. 'Regression shrinkage and selection via the lasso'. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288.

[52] Kim-Chuan Toh and Sangwoon Yun. 'An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems'. In: *Pacific Journal of optimization* 6.615-640 (2010), p. 15.

[53] Ivo Welch and Amit Goyal. 'A comprehensive look at the empirical performance of equity premium prediction'. In: *The Review of Financial Studies* 21.4 (2008), pp. 1455–1508.

[54] Kenneth D West. 'Forecast evaluation'. In: *Handbook of economic forecasting* 1 (2006), pp. 99–134.

[55] Hui Zou and Trevor Hastie. 'Regularization and variable selection via the elastic net'. In: *Journal of the royal statistical society: series B (statistical methodology)* 67.2 (2005), pp. 301–320.

5 Appendix

5.1 Appendix A: List of Macroeconomic Variables

Variable Name	Description	Source	tCode
RPI	Real Personal Income	Fred-MD	5
W875RX1	Real personal income ex transfer receipts	Fred-MD	5
DPCERA3M086SBEA	Real personal consumption expenditures	Fred-MD	5
CMRMTSPLx	Real Manu. and Trade Industries Sales	Fred-MD	5
RETAILx	Retail and Food Services Sales	Fred-MD	5
INDPRO	IP Index	Fred-MD	5
IPFPNSS	IP: Final Products and Nonindustrial Supplies	Fred-MD	5
IPFINAL	IP: Final Products (Market Group)	Fred-MD	5
IPCONGD	IP: Consumer Goods	Fred-MD	5
IPDCONGD	IP: Durable Consumer Goods	Fred-MD	5
IPNCONGD	IP: Nondurable Consumer Goods	Fred-MD	5
IPBUSEQ	IP: Business Equipment	Fred-MD	5
IPMAT	IP: Materials	Fred-MD	5
IPDMAT	IP: Durable Materials	Fred-MD	5
IPNMAT	IP: Nondurable Materials	Fred-MD	5
IPMANSICS	IP: Manufacturing (SIC)	Fred-MD	5
IPB51222S	IP: Residential Utilities	Fred-MD	5
IPFUELS	IP: Fuels	Fred-MD	5
CUMFNS	Capacity Utilization: Manufacturing	Fred-MD	2
HWI	Help-Wanted Index for United States	Fred-MD	2
HWIURATIO	Ratio of Help Wanted/No. Unemployed	Fred-MD	2
CLF16OV	Civilian Labor Force	Fred-MD	5
CE16OV	Civilian Employment	Fred-MD	5
UNRATE	Civilian Unemployment Rate	Fred-MD	2
UEMPMEAN	Average Duration of Unemployment (Weeks)	Fred-MD	2
UEMPLT5	Civilians Unemployed - Less Than 5 Weeks	Fred-MD	5
UEMP5TO14	Civilians Unemployed for 5-14 Weeks	Fred-MD	5
UEMP15OV	Civilians Unemployed - 15 Weeks & Over	Fred-MD	5
UEMP15T26	Civilians Unemployed for 15-26 Weeks	Fred-MD	5
UEMP27OV	Civilians Unemployed for 27 Weeks and Over	Fred-MD	5
CLAIMSx	Initial Claims	Fred-MD	5
PAYEMS	All Employees: Total nonfarm	Fred-MD	5
USGOOD	All Employees: Goods-Producing Industries	Fred-MD	5
CES1021000001	All Employees: Mining and Logging: Mining	Fred-MD	5
USCONS	All Employees: Construction	Fred-MD	5
MANEMP	All Employees: Manufacturing	Fred-MD	5
DMANEMP	All Employees: Durable goods	Fred-MD	5
NDMANEMP	All Employees: Nondurable goods	Fred-MD	5
SRVPRD	All Employees: Service-Providing Industries	Fred-MD	5
USTPU	All Employees: Trade, Transportation & Utilities	Fred-MD	5
USWTRADE	All Employees: Wholesale Trade	Fred-MD	5
USTRADE	All Employees: Retail Trade	Fred-MD	5
USFIRE	All Employees: Financial Activities	Fred-MD	5
USGOVT	All Employees: Government	Fred-MD	5
CES0600000007	Avg Weekly Hours : Goods-Producing	Fred-MD	1
AWOTMAN	Avg Weekly Overtime Hours : Manufacturing	Fred-MD	2
AWHMAN	Avg Weekly Hours : Manufacturing	Fred-MD	1
HOUST	Housing Starts: Total New Privately Owned	Fred-MD	4
HOUSTNE	Housing Starts, Northeast	Fred-MD	4
HOUSTMW	Housing Starts, Midwest	Fred-MD	4
HOUSTS	Housing Starts, South	Fred-MD	4
HOUSTW	Housing Starts, West	Fred-MD	4
PERMIT	New Private Housing Permits (SAAR)	Fred-MD	4
PERMITNE	New Private Housing Permits, Northeast (SAAR)	Fred-MD	4
PERMITMW	New Private Housing Permits, Midwest (SAAR)	Fred-MD	4
PERMITS	New Private Housing Permits, South (SAAR)	Fred-MD	4
PERMITW	New Private Housing Permits, West (SAAR)	Fred-MD	4
AMDMNOx	New Orders for Durable Goods	Fred-MD	5
AMDMUOx	Unfilled Orders for Durable Goods	Fred-MD	5
BUSINVx	Total Business Inventories	Fred-MD	5
ISRATIOx	Total Business: Inventories to Sales Ratio	Fred-MD	2
M1SL	M1 Money Stock	Fred-MD	6
M2SL	M2 Money Stock	Fred-MD	6
M2REAL	Real M2 Money Stock	Fred-MD	5
AMBSL	St. Louis Adjusted Monetary Base	Fred-MD	6
TOTRESNS	Total Reserves of Depository Institutions	Fred-MD	6
NONBORRES	Reserves Of Depository Institutions	Fred-MD	7
BUSLOANS	Commercial and Industrial Loans	Fred-MD	6
REALLN	Real Estate Loans at All Commercial Banks	Fred-MD	6
NONREVSL	Total Nonrevolving Credit	Fred-MD	6
CONSPI	Nonrevolving consumer credit to Personal Income	Fred-MD	2

Variable Name	Description	Source	tCode
S&P 500	S&P'Âs Common Stock Price Index: Composite	Fred-MD	5
S&P: indust	S&P's Common Stock Price Index: Industrials	Fred-MD	5
S&P div yield	S&P'Âs Composite Common Stock: Dividend Yield	Fred-MD	2
S&P PE ratio	S&P'Âs Composite Common Stock: Price-Earnings Ratio	Fred-MD	5
FEDFUNDS	Effective Federal Funds Rate	Fred-MD	2
CP3Mx	3-Month AA Financial Commercial Paper Rate	Fred-MD	2
TB3MS	3-Month Treasury Bill	Fred-MD	2
TB6MS	6-Month Treasury Bill	Fred-MD	2
GS1	1-Year Treasury Rate	Fred-MD	2
GS5	5-Year Treasury Rate	Fred-MD	2
GS10	10-Year Treasury Rate	Fred-MD	2
AAA	Moody's Seasoned Aaa Corporate Bond Yield	Fred-MD	2
BAA	Moody'Âs Seasoned Baa Corporate Bond Yield	Fred-MD	2
COMPAPFFx	3-Month Commercial Paper Minus FEDFUNDS	Fred-MD	1
TB3SMFFM	3-Month Treasury C Minus FEDFUNDS	Fred-MD	1
TB6SMFFM	6-Month Treasury C Minus FEDFUNDS	Fred-MD	1
T1YFFM	1-Year Treasury C Minus FEDFUNDS	Fred-MD	1
T5YFFM	5-Year Treasury C Minus FEDFUNDS	Fred-MD	1
T10YFFM	10-Year Treasury C Minus FEDFUNDS	Fred-MD	1
AAAFFM	Moody's Aaa Corporate Bond Minus FEDFUNDS	Fred-MD	1
BAAFFM	Moody'Âs Baa Corporate Bond Minus FEDFUNDS	Fred-MD	1
EXSZUSx	Switzerland / U.S. Foreign Exchange Rate	Fred-MD	5
EXJPUSx	Japan / U.S. Foreign Exchange Rate	Fred-MD	5
EXUSUKx	U.S. / U.K. Foreign Exchange Rate	Fred-MD	5
EXCAUSx	Canada / U.S. Foreign Exchange Rate	Fred-MD	5
WPSFD49207	PPI: Finished Goods	Fred-MD	6
WPSFD49502	PPI: Finished Consumer Goods	Fred-MD	6
WPSID61	PPI: Intermediate Materials	Fred-MD	6
WPSID62	PPI: Crude Materials	Fred-MD	6
OILPRICEx	Crude Oil, spliced WTI and Cushing	Fred-MD	6
PPICMM	PPI: Metals and metal products	Fred-MD	6
CPIAUCSL	CPI : All Items	Fred-MD	6
CPIAPPSL	CPI : Apparel	Fred-MD	6
CPITRNSL	CPI : Transportation	Fred-MD	6
CPIMEDSL	CPI : Medical Care	Fred-MD	6
CUSR0000SAC	CPI : Commodities	Fred-MD	6
CUSR0000SAD	CPI : Durables	Fred-MD	6
CUSR0000SAS	CPI : Services	Fred-MD	6
CPIULFSL	CPI : All Items Less Food	Fred-MD	6
CUSR0000SA0L2	CPI : All items less shelter	Fred-MD	6
CUSR0000SA0L5	CPI : All items less medical care	Fred-MD	6
PCEPI	Personal Cons. Expend.: Chain Index	Fred-MD	6
DDURRG3M086SBEA	Personal Cons. Exp: Durable goods	Fred-MD	6
DNDGRG3M086SBEA	Personal Cons. Exp: Nondurable goods	Fred-MD	6
DSERRG3M086SBEA	Personal Cons. Exp: Services	Fred-MD	6
CES0600000008	Avg Hourly Earnings : Goods-Producing	Fred-MD	6
CES2000000008	Avg Hourly Earnings : Construction	Fred-MD	6
CES3000000008	Avg Hourly Earnings : Manufacturing	Fred-MD	6
MZMSL	MZM Money Stock	Fred-MD	6
DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding	Fred-MD	6
DTCTHFNM	Total Consumer Loans and Leases Outstanding	Fred-MD	6
INVEST	Securities in Bank Credit at All Commercial Banks	Fred-MD	6
VXOCLSx	CBOE S&P 100 Volatility Index: VXO	Fred-MD	1

Variable Name	Description	Source	tCode
dp	Divident-price ratio	Welch and Goyal (2008)	2
ep	Earnings-price ratio	Welch and Goyal (2008)	2
bm	Book-to-market ratio	Welch and Goyal (2008)	5
ntis	Net equity expansion	Welch and Goyal (2008)	2
tbl	Treasury-bill rate	Welch and Goyal (2008)	2
dfy	Default spread	Welch and Goyal (2008)	2
svar	Stock variance	Welch and Goyal (2008)	5