Topic 6: Machine Learning

Professor Mihail Velikov

SAFE PhD Course on Anomalies – June, 2024

PennState
Smeal College of Business

Shrinking the cross-section

- Partial list of ever more sophisticated machine-learning techniques:
  - Freyberger, Neuhierl, and Weber (2020)
  - Feng, Polson, and Xu (WP, 2019)
  - Feng, Giglio, and Xiu (JF, 2020)
  - Gu, Kelly, and Xiu (RFS, 2020)
  - Lettau and Pelger (RFS, 2020)
  - Giglio, Liao, and Xiu (RFS, 2021)
  - Giglio and Xiu (JPE, 2021)
  - Chen, Pelger, and Zhu (MS, 2024)
  - Cong, Tang, Wang, and Zhang (WP, 2022)
  - Azevedo, Hoegner, and Velikov (WP, 2023)

- See review paper by Giglio, Kelly, and Xiu (2022)

PennState
Smeal College of Business

Machine learning to measure asset risk premia

- Discussion will follow Gu et al. (RFS, 2020)

- Asset excess returns follow additive prediction model:

$$r_{i,t+1} = E_t\left(r_{i,t+1}\right) + \epsilon_{i,t+1}$$

where

$$E_t\left(r_{i,t+1}\right) = g^{\star}(z_{i,t})$$

and i indicates stock, t indicates time period (e.g., month), z is a vector of expected return predictors , and $g^{\star}(\cdot)$ is the conditional expected return, which is a flexible function of predictors

- Note:
  - $g^{\star}(\cdot)$ is the same (i.e., same functional form) across time and stocks
  - $g^{\star}(\cdot)$ depends on z only through $z_{i,t}$, that is, no past history explicitly incorporated in the functional form

PennState
Smeal College of Business

OLS

- Let's start with a baseline: OLS

- Conditional expectation is linear in predictors:

$$g^{\star}\left(z_{i,t};\theta\right) = z_{i,t}'\theta$$

- Linear model that does not allow for nonlinearities and interactions
  - Also imposes use of all predictors

- The estimation uses standard least squares. That is, we minimize the following "l_2" objective function:

$$\mathcal{L}(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(r_{i,t+1} - g^{\star}\left(z_{i,t};\theta\right)\right)^{2}$$

PennState
Smeal College of Business

OLS: Robust objective function

- Weighted least squares:

$$\mathcal{L}_W(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T} w_{i,t}\left(r_{i,t+1} - g^{\star}(z_{i,t};\theta)\right)^2$$

  - Could improve prediction if you have observations that are more statistically or economically informative (e.g., value-weighting to avoid microstructure bias)

- OLS-Huber

$$\mathcal{L}_H(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T} H\left(r_{i,t+1} - g^{\star}(z_{i,t};\theta), \xi\right)$$

where

$$H(x;\xi) = \begin{cases} x^2, & |x| \le \xi \\ 2\xi|x| - \xi^2, & |x| > \xi \end{cases}$$

  - The Huber loss is a hybrid of square loss for relatively small errors and absolute loss for relatively large errors, where the combination is controlled by a "tuning" parameter ξ

PennState Smeal College of Business

Hyperparameter tuning

- Hyperparameters (aka "tuning parameters") are parameters that help balance a model's overfitting and variance
  - You can think of them as "dials", i.e., way to control the behavior of a machine learning model
  - E.g., penalization parameters in LASSO, number of random trees in a forest, hidden layers in a neural network
  - Little theoretical guidance on how to "tune" these for optimal OOS performance

- Various way to address
  - Sample split into three periods:
    - Training – used to estimate model subject to a specific set of tuning parameters
    - Validation – used for tuning the hyperparameters
      - Check forecast error in validation sample based on variations of tuning parameters & select optimal one
    - Testing – used to calculate OOS predictions and forecast errors for model comparison using estimates from the training sample with hyperparameters chosen from validation sample
  - Cross-validation – sequentially draw from in-sample data to pick training & validation samples

PennState
Smeal College of Business

Penalized linear model

- We append a penalty to the original loss function:

$$\mathcal{L}(\theta;\cdot) = \mathcal{L} + \phi(\theta;\cdot)$$

- "Elastic net" penalty nests Least Absolute Shrinkage and Selection Operator (LASSO, $\rho = 0$) and ridge regression ($\rho = 1$) :

$$\phi(\theta;\lambda,\rho) = \underbrace{\lambda(1-\rho)\sum_{j=1}^{P}|\theta_j|}_{LASSO} + \underbrace{\frac{1}{2}\lambda\rho\sum_{j=1}^{P}\theta_j^2}_{ridge\ regression}$$

  - The LASSO penalty is also referred to as "l_1" parameter penalization
    - Due to its geometry (think of two parameter example - $|\theta_1| + |\theta_2| \le s$) LASSO imposes sparsity (i.e., set some coefficients equal exactly to 0)
    - LASSO is often referred to as a "selection" method
  - The ridge regression penalty is referred to as "l_2" parameter penalization
    - Ridge is a "shrinkage" method that prevents coefficients from getting too big
    - Think of two parameter example again - $(\theta_1^2 + \theta_2^2 \le s)$
  - When $\rho \in (0,1)$, elastic net imposes both shrinkage and selection

PennState
Smeal College of Business

Dimension reduction: PCR and PLS

- Penalized linear models use shrinkage and selection to manage high dimensionality by forcing the coefficients on most regressors near to or exactly to zero
  - Could be problematic if highly correlated predictors
  - Think of an extreme case:
    - Predictors are all equal to target forecast plus some iid noise
    - Better off to take the average (noise would cancel out) than to shrink & select

- Dimension reduction is similar to predictor weighted averaging, as opposed to selection.
  - We can form linear combinations of predictors to reduce noise to better isolate the signal in predictors and to decorrelate otherwise highly dependent predictors
  - Two classic examples:
    - Principal Components Regression (PCR)
    - Partial Least Squares (PLS)

PennState
Smeal College of Business

Dimension reduction: PCR

- Principal Components Regression (PCR) consists of two steps:
  - Use Principal Components Analysis (PCA) to combine predictors into a small set of linear combinations that best preserve the covariance structure
  - Select a few leading components to use in standard predictive regressions

- Drawback is that it is focused on covariance *among predictors*.
  - But the ultimate objective is forecasting

PennState
Smeal College of Business

Dimension reduction: PLS

- Partial Least Squares (PLS) also consists of two steps:
  - For each predictor *j*, estimate its univariate return prediction coefficient φ_j via OLS
  - Average all predictors into a single aggregate component with weights proportional to φ_j
  - Developed by Kelly and Pruitt (2013, 2015)
- Unlike PCR, it exploits covariation of predictors with the forecast target

PennState
Smeal College of Business

Dimension reduction: PCR and PLS

- Vectorized prediction problem:

$$R = Z\theta + E$$

  where:
  - R is $NT \times 1$ vector of returns $r_{i,t+1}$
  - Z is $NT \times P$ vector of stacked $z_{i,t}$ predictors
  - E is $NT \times 1$ vector of residuals $\epsilon_{i,t+1}$

- We need to condense Z from dimension P to a much smaller dimension of K linear combinations of predictors

$$R = (Z\Omega_K)\theta_K + \tilde{E}$$

  where:
  - Ω_K is $P \times K$ matrix with columns $w_1, w_2, \ldots, w_K$
  - w_j is the set of linear combination weights used to create the jth predictive component
  - θ_K is now a $K \times 1$ vector rather than $P \times 1$

PennState
Smeal College of Business

Dimension reduction: PCR and PLS

- PCR chooses the combination weights Ω_K recursively
  - The jth linear combination solves:

$$
\begin{aligned}
w_j &= arg\max_{w} Var(Zw) \\
s.t. \quad & w'w = 1 \\
& Cov(Zw, Zw_l) = 0, l = 1, 2, \ldots, j-1
\end{aligned}
$$

  - We seek the *K* linear combos of Z that most faithfully mimic the full predictor set

- PLS seeks K linear combination of *Z* that have maximal predictive association with the forecast target:

$$
\begin{aligned}
w_j &= arg\max_{w} Cov(R, Zw) \\
s.t. \quad & w'w = 1 \\
& Cov(Zw, Zw_l) = 0, l = 1, 2, \ldots, j-1
\end{aligned}
$$

  - PLS is willing to sacrifice how accurately $Z\Omega_K$ approximates Z in order to find components with more return predictability

PennState
Smeal College of Business

Generalized linear

- When true model is non-linear, linear models can be thought of as a first-order approximation
  - However, there is approximation error due to misspecification
  - Denote:
    - $g^\star(z_{i,t})$ be the true model
    - $g(z_{i,t})$ be the functional form specified by the econometrician
    - $g(z_{i,t};\hat{\theta})$ and $\hat{r}_{i,t+1}$ be the fitted model and its return forecast
  - The model's forecast error can be decomposed as:

$$r_{i,t+1} - \hat{r}_{i,t+1} = \underbrace{g^\star(z_{i,t}) - g(z_{i,t};\theta)}_{approximation\ error} + \underbrace{g(z_{i,t};\theta) - g(z_{i,t};\hat{\theta})}_{estimation\ error} + \underbrace{\epsilon_{i,t+1}}_{intrinsic\ error}$$

    - Intrinsic error is irreducible
    - Estimation error is due to sampling variation and depends on data
    - Approximation error is controlled by the econometrician

- We can reduce approximation error by incorporating more flexible specifications that improve model's ability to approximate the true model
  - Though that comes at a cost of higher chance of overfitting

PennState
Smeal College of Business

Generalized linear

- For example, model could add a *K*-term spline series expansion of the predictors:

$$g(z;\theta,p(\cdot)) = \sum_{j=1}^{P} p\left(z_j\right)'\theta_j$$

where:

- $p(\cdot) = (p_1(\cdot), p_2(\cdot), \ldots, p_K(\cdot))'$ is a vector of basis functions
- Parameters are now $K \times N$ matrix $\theta = (\theta_1, \theta_2, \ldots \theta_N)$

- Many choices of spline functions
  - E.g., a spline series of order two (i.e., quadratic splines):

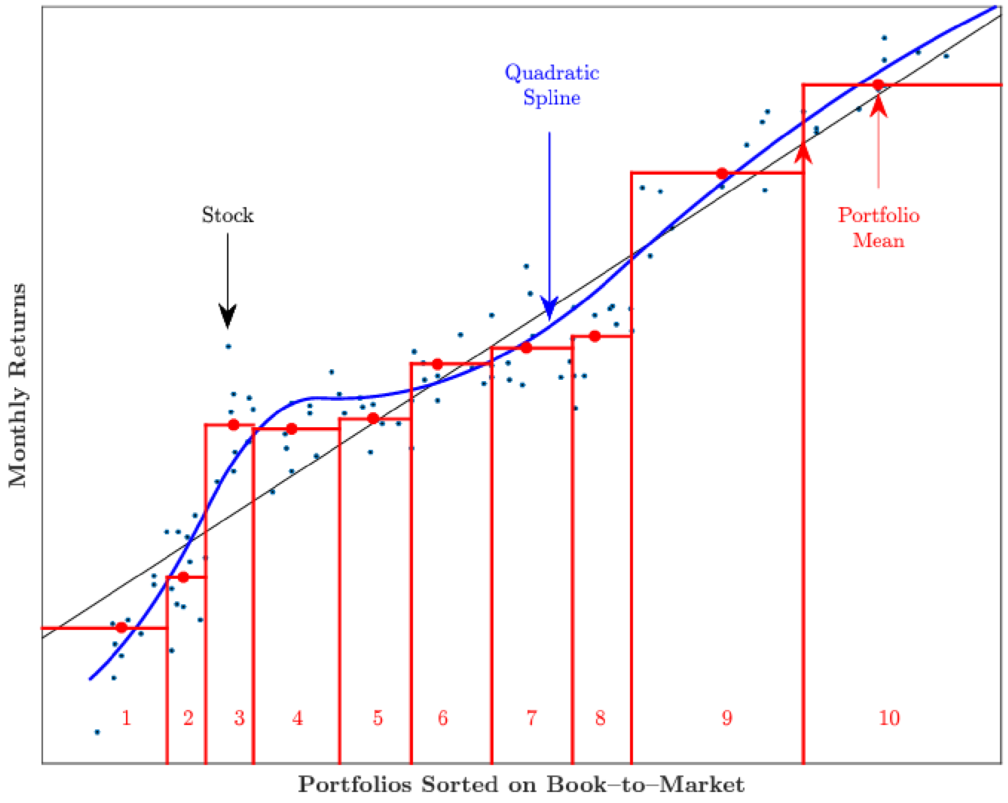
$$(1, z, (z-c_1)^2, (z-c_2)^2, \ldots, (z-c_{K-2})^2)$$

  - where $c_1, c_2, \ldots, c_{K-2}$ are knots

PennState
Smeal College of Business

Quadratic splines example from Freyberger et al. (RFS, 2020)

Figure A.3: **10 Portfolios sorted on Book-to-Market and Nonparametric Estimator**



This figure plots returns on the y-axis against the book-to-market ratio on the x-axis as well as portfolio mean returns and a nonparametric conditional mean function for simulated data.

PennState
Smeal College of Business

Generalized linear

- Estimation
  - Can follow OLS or OLS-Huber
  - The splines multiply the number of parameters, so we need penalization
  - Often used type of penalization for spline expansion setting is the "group LASSO":

$$\phi(\theta;\lambda,K) = \lambda \sum_{j=1}^{P} \left(\sum_{k=1}^{K} \theta_{j,k}^2 \right)^{1/2}$$

- Group LASSO selects all K spline terms associated with given characteristic or none of them
- λ and K are tuning parameters

PennState
Smeal College of Business

Trees

- Nonparametric approach for incorporating interactions among predictors
  - Alternative approach to regressions, we sub-divide, or partition, the predictor space into smaller regions
  - Sequentially sort data into two bins based on one of the predictive variables
  - That is, for a tree $\mathcal{T}$, with K "leaves" (terminal nodes), and depth L:

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

  where $C_k(L)$ is one of the K partitions of the data

  - Each partition is a product of up to L indicator functions of the predictors
  - The constant associated with partition k (denoted θ_k) is defined to be the sample average of outcomes within the partition

- Growing a tree is finding bins that best discriminate among outcomes

PennState
Smeal College of Business

Trees

- For example:

$$g(z_{i,t};\theta,K,L) = \theta_1 \mathbf{1}_{\{size_{i,t}<0.5\}}\mathbf{1}_{\{b/m_{i,t}<0.3\}} + \theta_2 \mathbf{1}_{\{size_{i,t}<0.5\}}\mathbf{1}_{\{b/m_{i,t}\geq 0.3\}} + \theta_3 \mathbf{1}_{\{size_{i,t}\geq 0.5\}}$$

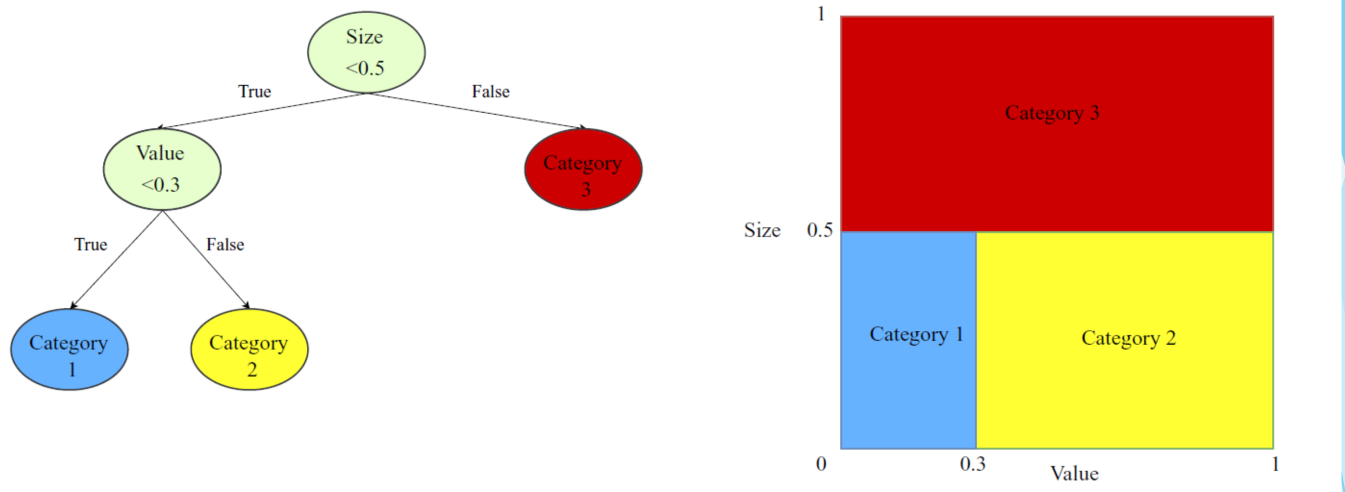


Figure 1
Regression tree example
This figure presents the diagrams of a regression tree (left) and its equivalent representation (right) in the space of two characteristics (size and value). The terminal nodes of the tree are colored in blue, yellow, and red. Based on their values of these two characteristics, the sample of individual stocks is divided into three categories.

PennState
Smeal College of Business

Trees

- Many optimization techniques to find optimal trees

- Breiman et al. (1984) myopically optimize forecast error at the start of each branch
  - Loss associated with the forecast error for a branch C is often called "impurity"
  - Gu et al. (2020) choose l_2 impurity for each branch of the tree:

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

    where $|C|$ denotes the number of observations in set C
  - Given C, optimal choice of θ is $\theta = \frac{1}{|C|} \sum_{z_{i,t} \in C} r_{i,t+1}$
  - We are looking for a branch C that locally minimizes the impurity
  - Branching stops when the number of leaves or the depth of the tree reach a prespecified threshold that can be selected adaptively using validation

PennState
Smeal College of Business

Trees

- Suppose we are at partition C and the two new bins are C_{left} and C_{right}

- We choose the next predictor variable and its threshold to minimize the sum of squared forecast errors:

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

- We compute $\mathcal{L}\left(C, C_{left}, C_{right}\right)$ sequentially for each predictor and choose the one with the minimum loss

- The predicted return is the average of returns of all stocks within the group:

$$\theta_{left} = \frac{1}{|C_{left}|} \sum_{z_{i,t} \in C_{left}} r_{i,t+1} \, ; \theta_{right} = \frac{1}{|C_{right}|} \sum_{z_{i,t} \in C_{right}} r_{i,t+1}$$

PennState
Smeal College of Business

Trees

- Pros
  - Trees are invariant to monotonic transformations of the predictors
  - It can approximate nonlinearities
  - A tree of depth L can capture at most $L - 1$ interactions
- Cons
  - Prone to overfitting
  - Need to be regularized

PennState
Smeal College of Business

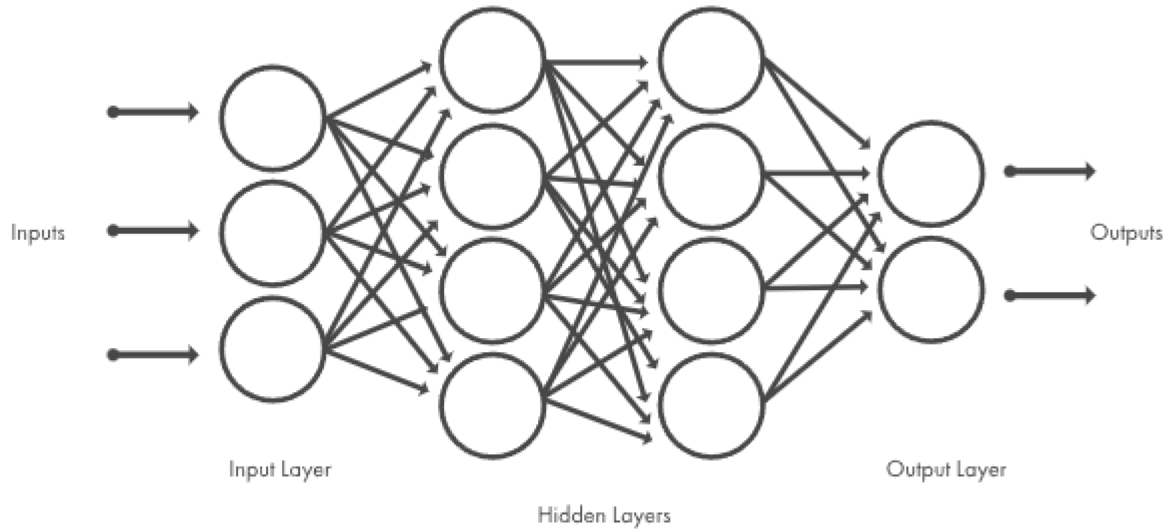
Random forest

- Ensemble method that combines forecasts from many shallow trees
- For each tree a subset of predictors is drawn at each potential branch split
- This lowers the correlation among different trees and improves the prediction
- The depth, L, of the trees is a tuning (hyper) parameter and is optimized in the validation stage

PennState
Smeal College of Business

Neural networks

- Type of machine learning approach inspired by how neurons signal to each other in the human brain
  - Combines several processing layers using simple elements operating in parallel
  - The network consists of:
    - Input layer
    - One or more hidden layers
    - Output layer



Neural networks

- Types of neural networks
  - Feedforward neural network
    - Shallow neural network (i.e., includes only one or two hidden layers)
  - Convolutional neural network
    - Deep neural network (i.e., many layers) architecture widely used in image processing
    - Characterized by convolutional layers that shift windows across the input with nodes that share weights
  - Recurrent neural network
    - Deep neural network with feedback loops that model sequential dependencies in the input
    - Most popular example is long short-term memory network (LSTM)

PennState
Smeal College of Business

Neural networks

- A nonlinear feed forward network consists of:
  - Input layer
  - One or more hidden layer
  - Output layer

- Analogous to axons in a biological brain, layers of the network represent groups of "neurons" with each layer connected by "synapses" that transmit signals among neurons of different layer

- Deep learning reflects the notion that the number of hidden layers is large

- For finance, a few layers is large enough

PennState
Smeal College of Business

Neural networks

- Lots of choices to make
  - Number of hidden layers
  - Number of neurons in each layer
  - How do we connect the units?
  - What activation function do you use?

- Not a lot of research in finance applications
  - Still a lot to be done

PennState
Smeal College of Business

Neural networks

- Gu et al. (2020) consider NN with up to 5 hidden layers
  - NN1: 1 hidden layer with 32 neurons
  - NN2: 2 hidden layers with 32 and 16 neurons, respectively
  - NN3: 3 hidden layers with 32, 16, and 8 neurons, respectively
  - NN4: 4 hidden layers with 32, 16, 8, and 4 neurons, respectively
  - NN5: 5 hidden layers with 32, 16, 8, 4, and 2 neurons, respectively
  - Choice of neurons is known as geometric pyramid rule (Masters, 1993)

- Nonlinear activation function choices include sigmoid $\left(f(x) = \frac{1}{(1+e^{-x})}\right)$, hyperbolic $\left(\mathrm{f}(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}\right)$, and softmax $\left(f\left(x_j^i\right) = \frac{\exp\left(x_j^i\right)}{\sum_k \exp\left(x_k^i\right)}\right)$
  - Gu et al. (2020) choose the rectified linear unit function, defined as:

$$ReLU(x) = \max(x, 0) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Neural networks

- So their NN architecture can be mathematically expressed as follows
  - Let $K^{(l)}$ denote the number of neurons in each layer $l = 1, \dots, L$
  - Let $x_k^{(l)}$ denote the output of neuron k in layer l
  - Define the vector of outputs for layer l (augmented to include constant, $x_0^{(l)}$ as:
$$x^{(l)} = \left(1, x_1^{(l)}, \dots, x_{K^{(l)}}^{(l)}\right)'$$
  - Define the input layer using the raw predictors:
$$x^{(0)} = (1, z_1, \dots, z_N)'$$
  - For each layer, $l > 0$:
$$x^{(l)} = ReLU\left(x^{(l-1)\prime}\theta_k^{l-1}\right)$$
  - And the final output is:
$$g(z;\theta) = x^{(L-1)'}\theta^{(L-1)}$$

PennState
Smeal College of Business

Neural networks

- With the ReLU activation function and one hidden layer with K neurons:

$$
\begin{aligned}
g(z;\theta) &= x^{(1)\prime}\theta^{(1)} \\
&= \left(1, x_1^{(1)}, \dots, x_K^{(1)}\right)' \left(\theta_0^{(1)}, \theta_1^{(1)}, \dots \theta_K^{(1)}\right) \\
&= \left(1, ReLU(x^{(0)\prime}\theta_1^{(0)}), \dots, ReLU(x^{(0)\prime}\theta_K^{(0)})\right)' \left(\theta_0^{(1)}, \theta_1^{(1)}, \dots \theta_K^{(1)}\right) \\
&= \left(1, max(x^{(0)\prime}\theta_1^{(0)}, 0), \dots, max(x^{(0)\prime}\theta_K^{(0)}, 0)\right)' \left(\theta_0^{(1)}, \theta_1^{(1)}, \dots \theta_K^{(1)}\right) \\
&= \theta_0^{(1)} + \sum_{i=1}^{K} \max(x^{(0)\prime}\theta_i^{(0)}, 0) \times \theta_i^{(1)} \\
&= \theta_0^{(1)} + \sum_{i=1}^{K} \max\left(\theta_{i,0}^{(0)} + \sum_{j=1}^{N} z_j\, \theta_{i,j}^{(0)}, 0\right) \times \theta_i^{(1)}
\end{aligned}
$$

- The number of parameters in the hidden layer, is $K \times (N+1)$, plus another $K+1$ parameters for the output layer
- The optimization minimizes the sum of squared errors, like OLS
- Also include a LASSO regularization l_1 penalty
- With linear activation function, just ignore MAX operator

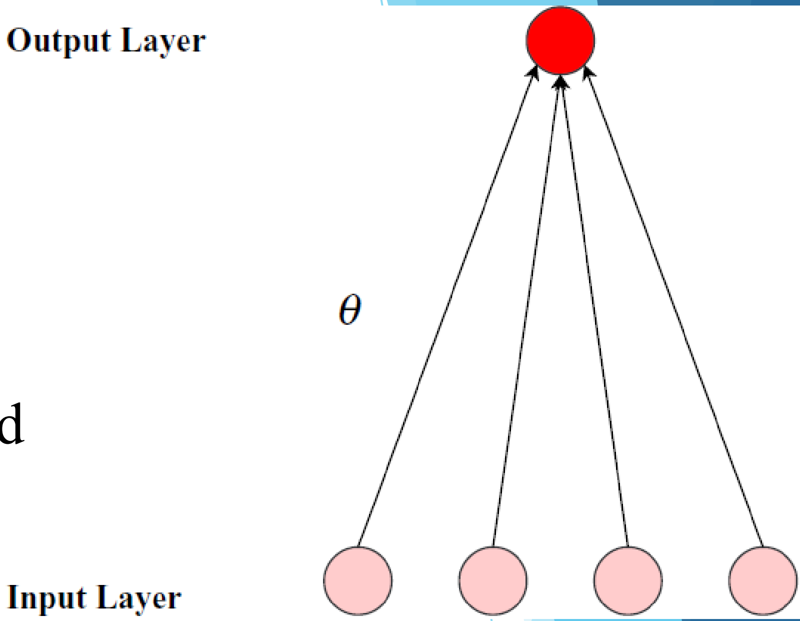
PennState
Smeal College of Business

Neural networks: simple examples

- A simple example: no hidden layers

- Assume 4 predictors $(z_1, \ldots, z_4)$

- Each predictor is amplified or attenuated according to a 5-dimensional parameter vector, θ (4 predictors + intercept)



- Output layer aggregates the weighted signals into the forecast:

$$\theta_0 + \sum_{k=1}^{4} z_k \theta_k$$

- Which is a linear regression

PennState Smeal College of Business

Neural networks: simple examples

- Assume 1 hidden layer with 5 neurons
  - Each neuron draws information linearly from the inputs
  - It then applies a nonlinear "activation function" f to its aggregated signal before sending the output to the next layer:

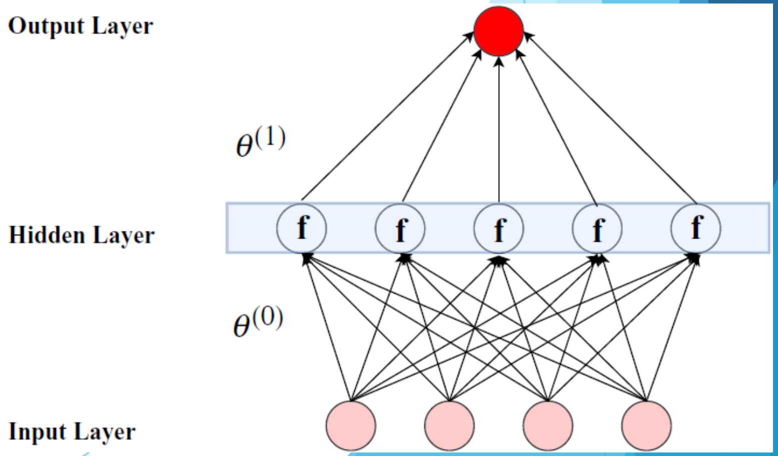
$$x_2^{(1)} = f\left(\theta_{2,0}^{(0)} + \sum_{j=1}^{4} z_j \theta_{2,j}^{(0)}\right)$$

where the subscripts indicate the neuron and superscripts indicate the layer

  - Finally, we linearly aggregate results from neurons into ultimate output forecast:

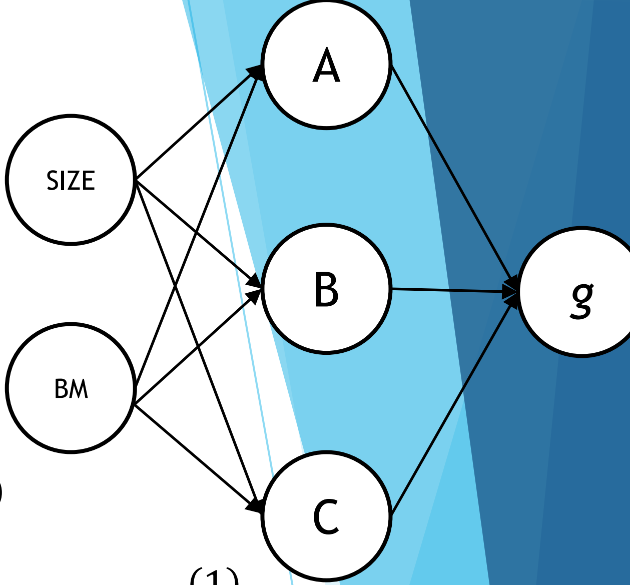
$$g(z;\theta) = \theta_0^{(1)} + \sum_{j=1}^{4} x_j^{(1)} \theta_j^{(1)}$$

  - A total of $31 = (4+1) \times 5 + 6$ parameters



PennState Smeal College of Business

Neural networks: simple examples



- Two inputs: size and book-to-market
- One hidden layer with three neurons: A, B, and C
- θ's are the weights

$$g(z;\theta) = x^{(1)'}\theta^{(1)}$$
$$= \theta_0^{(1)} + \sum_{i=1}^{K} \max\left(\theta_{i,0}^{(0)} + \sum_{j=1}^{N} z_j\, \theta_{i,j}^{(0)}\,, 0\right) \times \theta_i^{(1)}$$
$$= \theta_0^{(1)} + (\max(\theta_A^{(0)} + size \times \theta_{A,size}^{(0)} + bm \times \theta_{A,bm}^{(0)}, 0) \times \theta_A^{(1)}$$
$$+ (\max(\theta_B^{(0)} + size \times \theta_{B,size}^{(0)} + bm \times \theta_{B,bm}^{(0)}, 0) \times \theta_B^{(1)}$$
$$+(\max(\theta_C^{(0)} + size \times \theta_{C,size}^{(0)} + bm \times \theta_{C,bm}^{(0)}, 0) \times \theta_C^{(1)}$$

- We have $K \times (N+1) + K + 1 = 3 \times (2+1) + 3 + 1 = 13$ parameters
- Output is the predicted return
- Implement it by finding the vector of θ's that minimizes the sum of squared errors (data vs output), by aggregating across all stocks and all out-of-sample months

PennState
Smeal College of Business

Neural networks: regularization

- Neural networks are estimated by minimizing the penalized l_2 objective function (least squares) of prediction errors
  - High degree of nonlinearity and nonconvexity in neural networks make standard optimization techniques prohibitively computationally intensive
  - Common solution is to use stochastic gradient descent (SGD), which uses a small random subset of data

- Similarly, we impose regularization. Gu et al. (2020) implement:
  - LASSO l_1 penalty
  - Learning rate shrinkage
    - Helps the optimization when the gradient approaches zero
  - Early stopping
    - Start with parsimonious (near zero) parameter set, compare training vs validation sample errors
  - Batch normalization
    - Controls the variability of predictors across different regions (layers and nodes)
  - Ensembles – random seeds to initialize NN estimation & average forecasts across networks

PennState
Smeal College of Business

Gu et al. (2020) data

- Data
  - 1957-2016 divided into training (1957-1974), validation (1975-1986), and out-of-sample testing (1987-2016)
  - 94 anomalies ($c_{i,t}$)
  - 74 industry dummies (2-digit SIC)
  - 8 macroeconomic predictors: dividend-price ratio (dp), earnings-price ratio (ep), book-to-market ratio (bm), net equity expansion (ntis), Treasury-bill rate (tbl), term spread (tms), default spread (dfy), and stock variance (svar)
- Set of stock-level predictors:

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

where:

  - x_t are the macroeconomic predictors
  - $c_{i,t}$ are the stock-level characteristics
- Total number of covariates is $94 \times (8 + 1) + 74 = 920$

PennState
Smeal College of Business

Gu et al. (2020) results

- We can observe the “virtue of complexity”
  - More complicated models perform better
  - See Kelly, Malamud, and Zhou (JF, Forthcoming) for a theory of why that should be the case

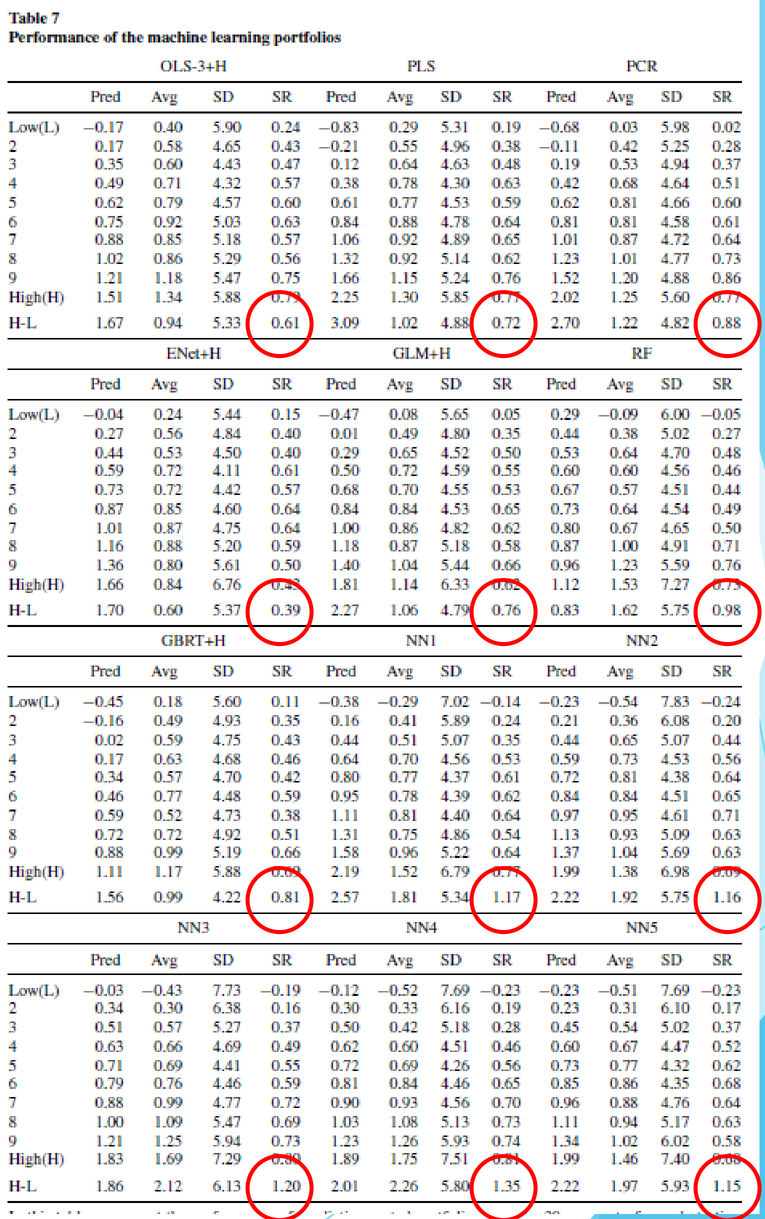


Table 7
Performance of the machine learning portfolios

	OLS-3+H				PLS				PCR			
	Pred	Avg	SD	SR	Pred	Avg	SD	SR	Pred	Avg	SD	SR
Low(L)	−0.17	0.40	5.90	0.24	−0.83	0.29	5.31	0.19	−0.68	0.03	5.98	0.02
2	0.17	0.58	4.65	0.43	−0.21	0.55	4.96	0.38	−0.11	0.42	5.25	0.28
3	0.35	0.60	4.43	0.47	0.12	0.64	4.63	0.48	0.19	0.53	4.94	0.37
4	0.49	0.71	4.32	0.57	0.38	0.78	4.30	0.63	0.42	0.68	4.64	0.51
5	0.62	0.79	4.57	0.60	0.61	0.77	4.53	0.59	0.62	0.81	4.66	0.60
6	0.75	0.92	5.03	0.63	0.84	0.88	4.78	0.64	0.81	0.81	4.58	0.61
7	0.88	0.85	5.18	0.57	1.06	0.92	4.89	0.65	1.01	0.87	4.72	0.64
8	1.02	0.86	5.29	0.56	1.32	0.92	5.14	0.62	1.23	1.01	4.77	0.73
9	1.21	1.18	5.47	0.75	1.66	1.15	5.24	0.76	1.52	1.20	4.88	0.86
High(H)	1.51	1.34	5.88	0.79	2.25	1.30	5.85	0.77	2.02	1.25	5.60	0.77
H-L	1.67	0.94	5.33	0.61	3.09	1.02	4.88	0.72	2.70	1.22	4.82	0.88

	ENet+H				GLM+H				RF			
	Pred	Avg	SD	SR	Pred	Avg	SD	SR	Pred	Avg	SD	SR
Low(L)	−0.04	0.24	5.44	0.15	−0.47	0.08	5.65	0.05	0.29	−0.09	6.00	−0.05
2	0.27	0.56	4.84	0.40	0.01	0.49	4.80	0.35	0.44	0.38	5.02	0.27
3	0.44	0.53	4.50	0.40	0.29	0.65	4.52	0.50	0.53	0.64	4.70	0.48
4	0.59	0.72	4.11	0.61	0.50	0.72	4.59	0.55	0.60	0.60	4.56	0.46
5	0.73	0.72	4.42	0.57	0.68	0.70	4.55	0.53	0.67	0.57	4.51	0.44
6	0.87	0.85	4.60	0.64	0.84	0.84	4.53	0.65	0.73	0.64	4.54	0.49
7	1.01	0.87	4.75	0.64	1.00	0.86	4.82	0.62	0.80	0.67	4.65	0.50
8	1.16	0.88	5.20	0.59	1.18	0.87	5.18	0.58	0.87	1.00	4.91	0.71
9	1.36	0.80	5.61	0.50	1.40	1.04	5.44	0.66	0.96	1.23	5.59	0.76
High(H)	1.66	0.84	6.76	0.43	1.81	1.14	6.33	0.62	1.12	1.53	7.27	0.73
H-L	1.70	0.60	5.37	0.39	2.27	1.06	4.79	0.76	0.83	1.62	5.75	0.98

	GBRT+H				NN1				NN2			
	Pred	Avg	SD	SR	Pred	Avg	SD	SR	Pred	Avg	SD	SR
Low(L)	−0.45	0.18	5.60	0.11	−0.38	−0.29	7.02	−0.14	−0.23	−0.54	7.83	−0.24
2	−0.16	0.49	4.93	0.35	0.16	0.41	5.89	0.24	0.21	0.36	6.08	0.20
3	0.02	0.59	4.75	0.43	0.44	0.51	5.07	0.35	0.44	0.65	5.07	0.44
4	0.17	0.63	4.68	0.46	0.64	0.70	4.56	0.53	0.59	0.73	4.53	0.56
5	0.34	0.57	4.70	0.42	0.80	0.77	4.37	0.61	0.72	0.81	4.38	0.64
6	0.46	0.77	4.48	0.59	0.95	0.78	4.39	0.62	0.84	0.84	4.51	0.65
7	0.59	0.52	4.73	0.38	1.11	0.81	4.40	0.64	0.97	0.95	4.61	0.71
8	0.72	0.72	4.92	0.51	1.31	0.75	4.86	0.54	1.13	0.93	5.09	0.63
9	0.88	0.99	5.19	0.66	1.58	0.96	5.22	0.64	1.37	1.04	5.69	0.63
High(H)	1.11	1.17	5.88	0.69	2.19	1.52	6.79	0.77	1.99	1.38	6.98	0.69
H-L	1.56	0.99	4.22	0.81	2.57	1.81	5.34	1.17	2.22	1.92	5.75	1.16

	NN3				NN4				NN5			
	Pred	Avg	SD	SR	Pred	Avg	SD	SR	Pred	Avg	SD	SR
Low(L)	−0.03	−0.43	7.73	−0.19	−0.12	−0.52	7.69	−0.23	−0.23	−0.51	7.69	−0.23
2	0.34	0.30	6.38	0.16	0.30	0.33	6.16	0.19	0.23	0.31	6.10	0.17
3	0.51	0.57	5.27	0.37	0.50	0.42	5.18	0.28	0.45	0.54	5.02	0.37
4	0.63	0.66	4.69	0.49	0.62	0.60	4.51	0.46	0.60	0.67	4.47	0.52
5	0.71	0.69	4.41	0.55	0.72	0.69	4.26	0.56	0.73	0.77	4.32	0.62
6	0.79	0.76	4.46	0.59	0.81	0.84	4.46	0.65	0.85	0.86	4.35	0.68
7	0.88	0.99	4.77	0.72	0.90	0.93	4.56	0.70	0.96	0.88	4.76	0.64
8	1.00	1.09	5.47	0.69	1.03	1.08	5.13	0.73	1.11	0.94	5.17	0.63
9	1.21	1.25	5.94	0.73	1.23	1.26	5.93	0.74	1.34	1.02	6.02	0.58
High(H)	1.83	1.69	7.29	0.80	1.89	1.75	7.51	0.81	1.99	1.46	7.40	0.68
H-L	1.86	2.12	6.13	1.20	2.01	2.26	5.80	1.35	2.22	1.97	5.93	1.15

PennState
Smeal College of Business

Gu et al. (2020) results

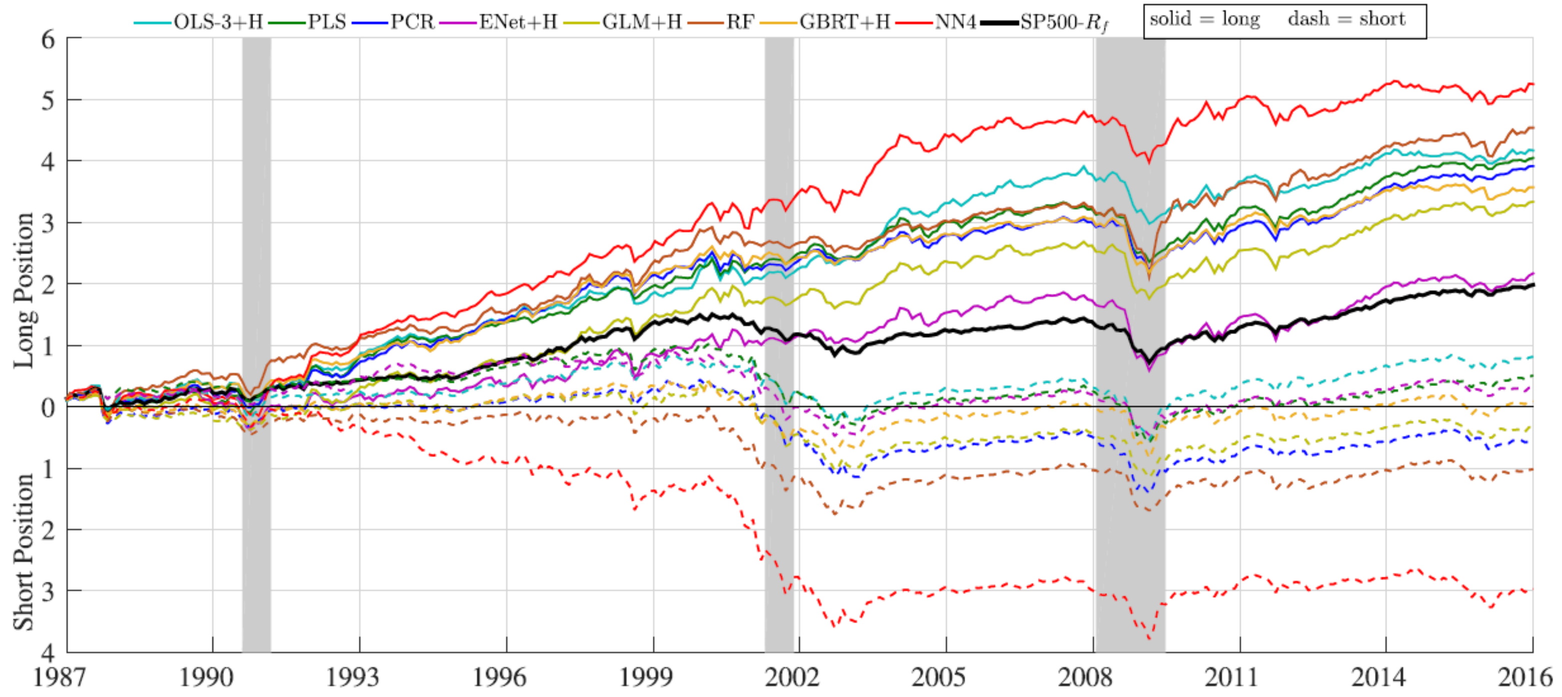


Figure 9
Cumulative return of machine learning portfolios
The figure shows the cumulative log returns of portfolios sorted on out-of-sample machine learning return forecasts. The solid and dashed lines represent long (top decile) and short (bottom decile) positions, respectively. The shaded periods show NBER recession dates. All portfolios are value weighted.

PennState
Smeal College of Business

Freyberger, Neuhierl, and Weber (2020)

- Summary
  - Adaptive group LASSO method with quadratic splines on 62 characteristics to screen & combine them
  - Conclude only 13 have incremental explanatory power
  - Report some crazy Sharpe ratios

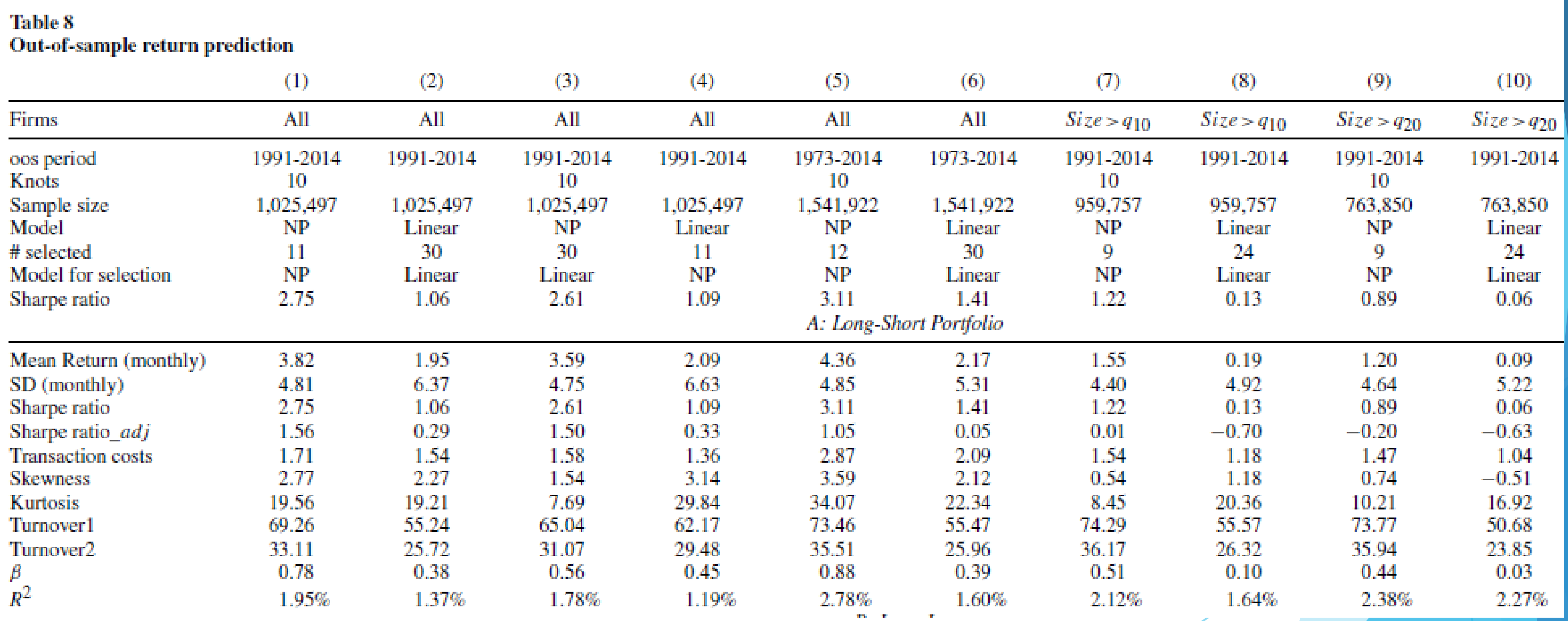


Table 8
Out-of-sample return prediction

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Firms	All	All	All	All	All	All	$Size > q_{10}$	$Size > q_{10}$	$Size > q_{20}$	$Size > q_{20}$
oos period	1991-2014	1991-2014	1991-2014	1991-2014	1973-2014	1973-2014	1991-2014	1991-2014	1991-2014	1991-2014
Knots	10		10		10		10		10	
Sample size	1,025,497	1,025,497	1,025,497	1,025,497	1,541,922	1,541,922	959,757	959,757	763,850	763,850
Model	NP	Linear	NP	Linear	NP	Linear	NP	Linear	NP	Linear
# selected	11	30	30	11	12	30	9	24	9	24
Model for selection	NP	Linear	Linear	NP	NP	Linear	NP	Linear	NP	Linear
Sharpe ratio	2.75	1.06	2.61	1.09	3.11	1.41	1.22	0.13	0.89	0.06
					A: Long-Short Portfolio					
Mean Return (monthly)	3.82	1.95	3.59	2.09	4.36	2.17	1.55	0.19	1.20	0.09
SD (monthly)	4.81	6.37	4.75	6.63	4.85	5.31	4.40	4.92	4.64	5.22
Sharpe ratio	2.75	1.06	2.61	1.09	3.11	1.41	1.22	0.13	0.89	0.06
Sharpe ratio_*adj*	1.56	0.29	1.50	0.33	1.05	0.05	0.01	−0.70	−0.20	−0.63
Transaction costs	1.71	1.54	1.58	1.36	2.87	2.09	1.54	1.18	1.47	1.04
Skewness	2.77	2.27	1.54	3.14	3.59	2.12	0.54	1.18	0.74	−0.51
Kurtosis	19.56	19.21	7.69	29.84	34.07	22.34	8.45	20.36	10.21	16.92
Turnover1	69.26	55.24	65.04	62.17	73.46	55.47	74.29	55.57	73.77	50.68
Turnover2	33.11	25.72	31.07	29.48	35.51	25.96	36.17	26.32	35.94	23.85
β	0.78	0.38	0.56	0.45	0.88	0.39	0.51	0.10	0.44	0.03
R^2	1.95%	1.37%	1.78%	1.19%	2.78%	1.60%	2.12%	1.64%	2.38%	2.27%

PennState
Smeal College of Business

Issues with combining characteristics

- Novy-Marx (2016)
  - Backtests on signals that combine characteristics are severely biased
  - His paper was motivated by some of the earlier papers that combine characteristics
    - Piotroski's F-score (2000)
      - 9 signals
    - Asness et. al. Quality Score (2014)
      - 21 signals
    - Stambaugh-Yuan "mispricing factor" (2016)
      - 11 signals
  - But is even more relevant now with the literature we just discussed

PennState
Smeal College of Business

Novy-Marx (2016)

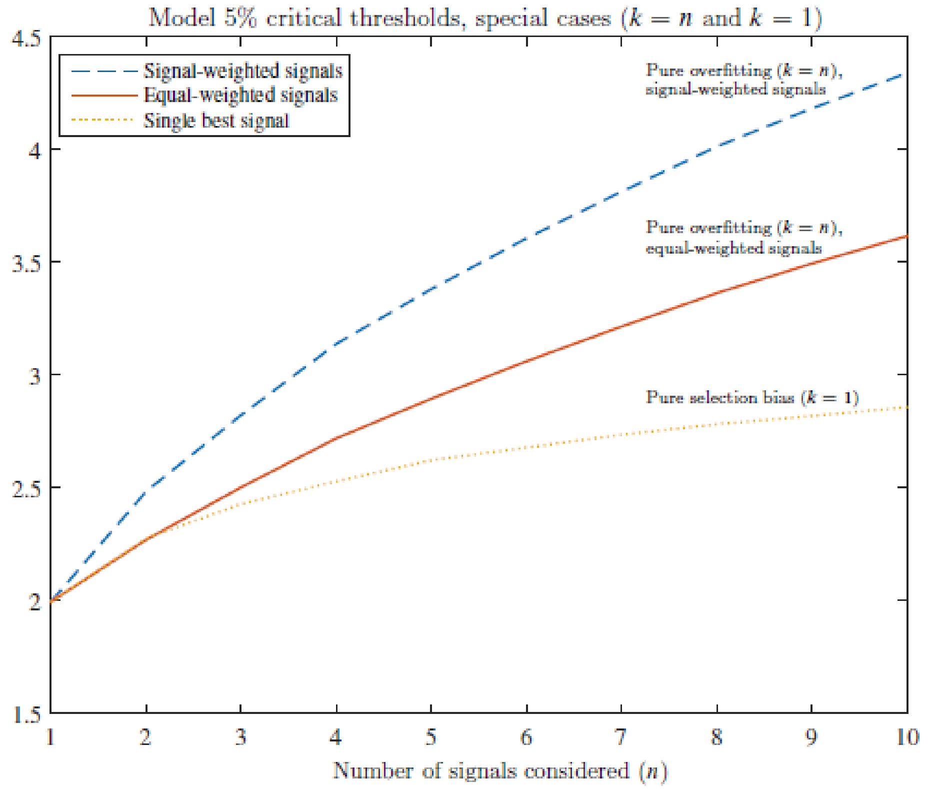


Fig. 1. Five percent critical t-statistics for the active return on best 1-of-n and best n-of-n strategies. The bottom, dotted line shows 5% critical thresholds for the pure selection bias case, when the investigator presents the strongest result from a set of n random strategies, where $n \in \{1, 2, ..., 10\}$. The middle, solid line and the top, dashed line, shows 5% critical thresholds when there is pure overfitting bias. In these cases stocks are selected by combining all n random signals, but the underlying signals are individually signed so that each predicts positive in-sample active returns. In the top, dashed line signals are signal-weighted, while in the middle, solid line signals are equal-weighted. Critical values come from generating 100,000 sets of n randomly generated signals. Returns are signal-weighted, with stocks held in proportion to the signal used for strategy construction, and rebalanced annually at the start of each year. Return data come from CRSP, and the sample covers January 1995 through December 2014.

PennState
Smeal College of Business

Novy-Marx (2016)

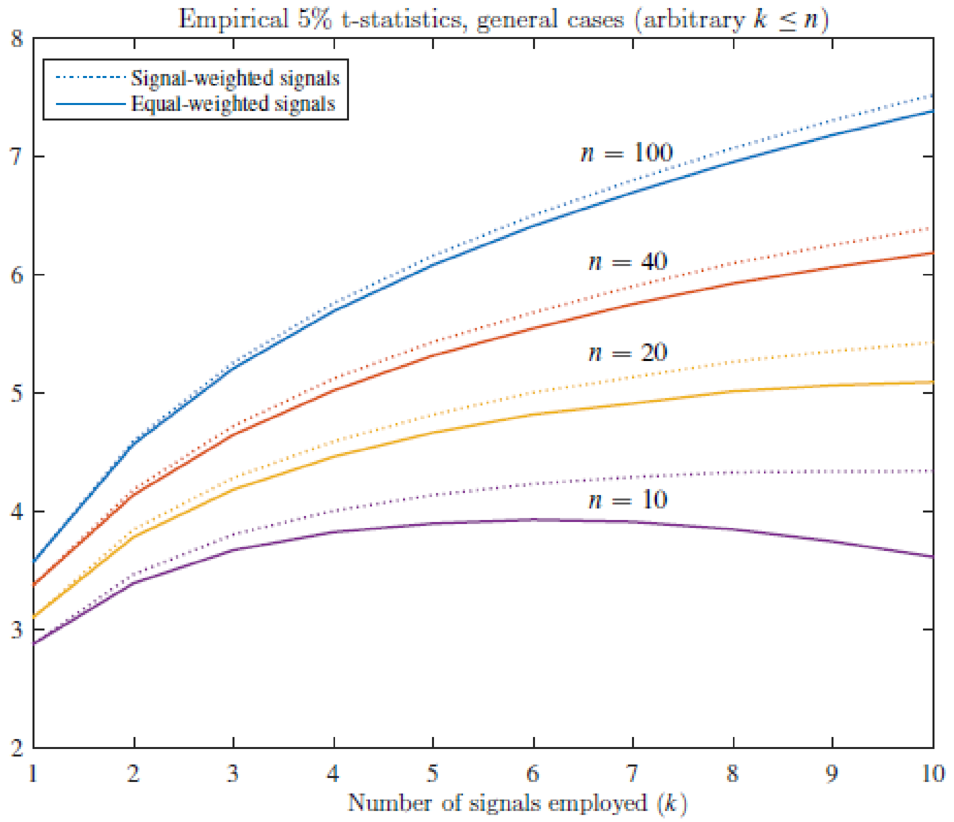


Fig. 2. Five percent critical t-statistics for best k-of-n strategies. The figure shows 5% critical thresholds for strategies selected using a signal constructed by combining the best $k = 1, 2, \ldots, 10$ performing signals, when the investigator considered $n \in \{10, 20, 40, 100\}$ candidate signals. Solid lines show the cases when the composite signal is constructed by equal-weighting the k best performing candidate signals, and dotted lines the cases when the composite signal is constructed by signal-weighting the signals. Critical values come from generating 100,000 sets of n randomly generated signals. Returns are signal-weighted, with stocks held in proportion to the signal used for strategy construction, and rebalanced annually at the start of each year. Return data come from CRSP, and the sample covers January 1995 through December 2014.

PennState
Smeal College of Business

Issues with combining characteristics

- Kozak, Nagel, and Santosh (JFE, 2020)
  - Reverse engineer a stochastic discount factor using Bayesian methods
    - Beware!

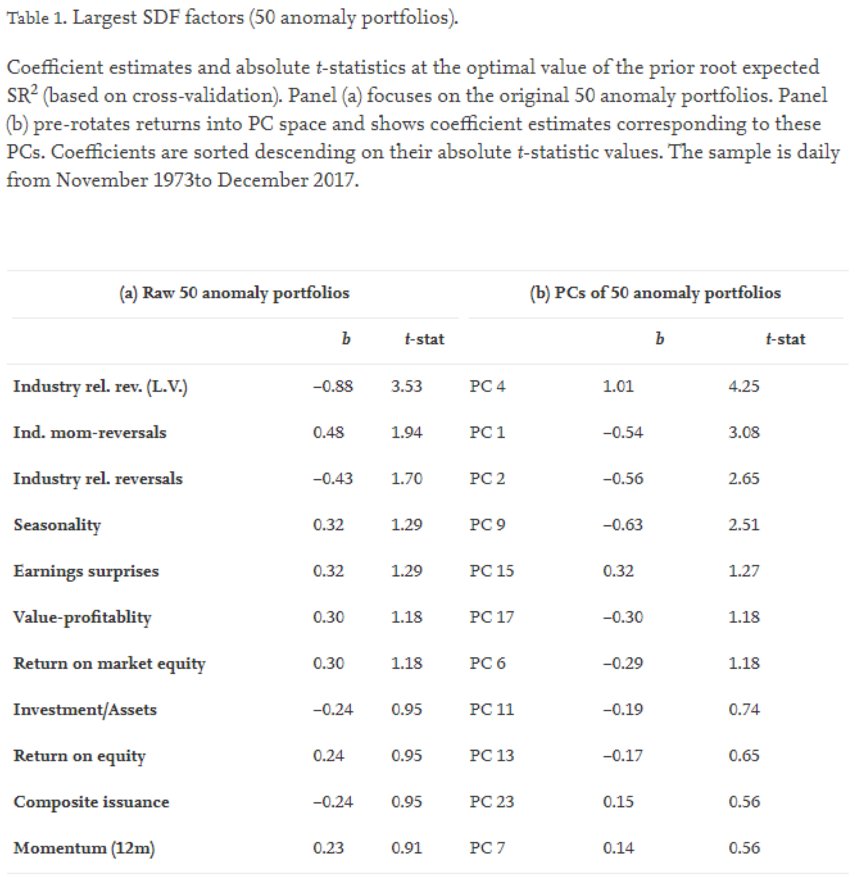


Table 1. Largest SDF factors (50 anomaly portfolios).

Coefficient estimates and absolute *t*-statistics at the optimal value of the prior root expected SR^2 (based on cross-validation). Panel (a) focuses on the original 50 anomaly portfolios. Panel (b) pre-rotates returns into PC space and shows coefficient estimates corresponding to these PCs. Coefficients are sorted descending on their absolute *t*-statistic values. The sample is daily from November 1973to December 2017.

(a) Raw 50 anomaly portfolios			(b) PCs of 50 anomaly portfolios		
	b	*t*-stat		*b*	*t*-stat
Industry rel. rev. (L.V.)	−0.88	3.53	PC 4	1.01	4.25
Ind. mom-reversals	0.48	1.94	PC 1	−0.54	3.08
Industry rel. reversals	−0.43	1.70	PC 2	−0.56	2.65
Seasonality	0.32	1.29	PC 9	−0.63	2.51
Earnings surprises	0.32	1.29	PC 15	0.32	1.27
Value-profitablity	0.30	1.18	PC 17	−0.30	1.18
Return on market equity	0.30	1.18	PC 6	−0.29	1.18
Investment/Assets	−0.24	0.95	PC 11	−0.19	0.74
Return on equity	0.24	0.95	PC 13	−0.17	0.65
Composite issuance	−0.24	0.95	PC 23	0.15	0.56
Momentum (12m)	0.23	0.91	PC 7	0.14	0.56

PennState
Smeal College of Business

Issues with combining characteristics

- Avramov, Cheng, and Metzker (MS, 2021)
  - Shows that strategies based on deep learning signals derive their profitability from difficult-to-arbitrage stocks and during high limits-to-arbitrage market states.
  - Excluding microcaps, distressed stocks, or episodes of high market volatility significantly reduces profitability.
  - Machine learning-based performance further deteriorates in the presence of reasonable trading costs.

PennState
Smeal College of Business

Avramov, Cheng, and Metzker (MS, 2021)

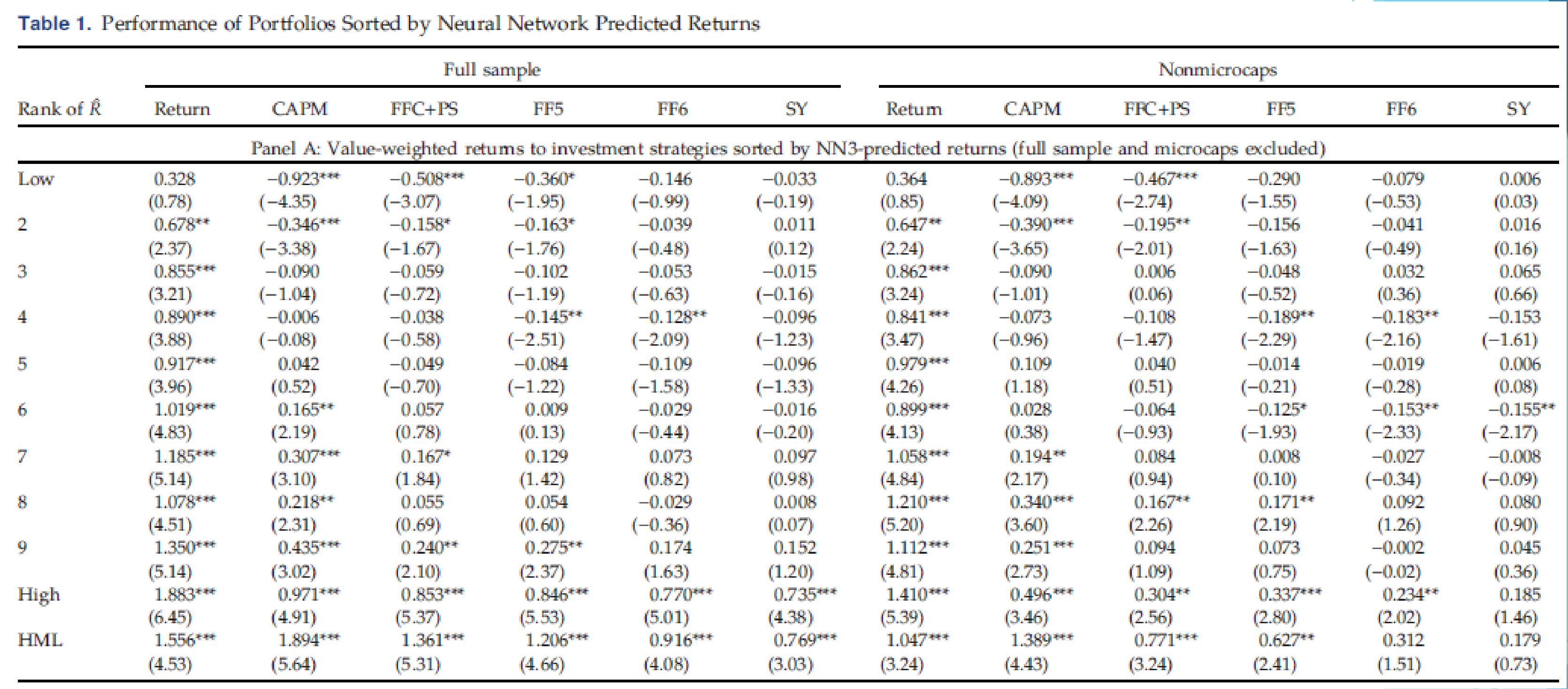
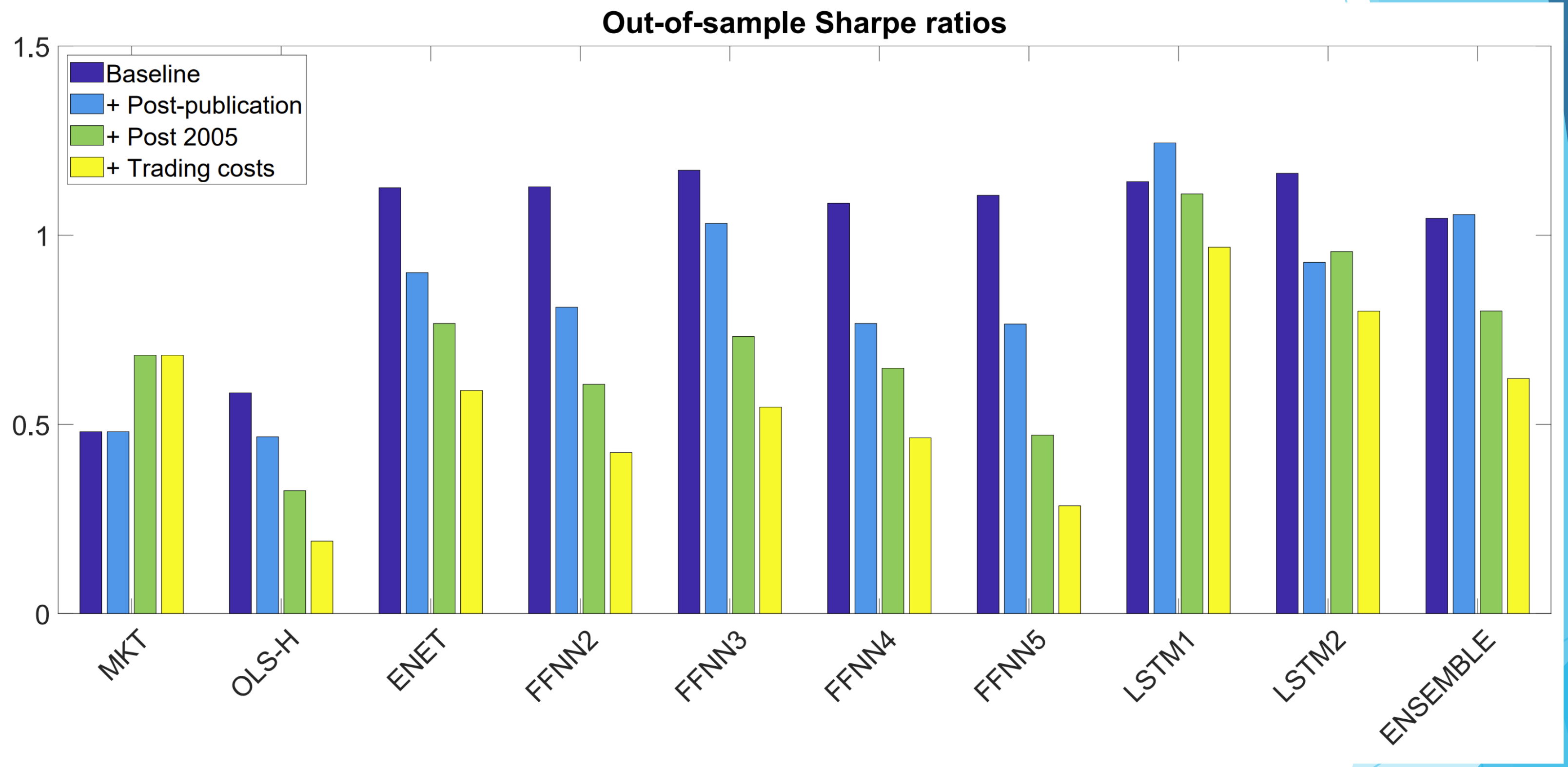


Table 1. Performance of Portfolios Sorted by Neural Network Predicted Returns

	Full sample						Nonmicrocaps					
Rank of $\hat{R}$	Return	CAPM	FFC+PS	FF5	FF6	SY	Return	CAPM	FFC+PS	FF5	FF6	SY
Panel A: Value-weighted returns to investment strategies sorted by NN3-predicted returns (full sample and microcaps excluded)												
Low	0.328	−0.923***	−0.508***	−0.360*	−0.146	−0.033	0.364	−0.893***	−0.467***	−0.290	−0.079	0.006
	(0.78)	(−4.35)	(−3.07)	(−1.95)	(−0.99)	(−0.19)	(0.85)	(−4.09)	(−2.74)	(−1.55)	(−0.53)	(0.03)
2	0.678**	−0.346***	−0.158*	−0.163*	−0.039	0.011	0.647**	−0.390***	−0.195**	−0.156	−0.041	0.016
	(2.37)	(−3.38)	(−1.67)	(−1.76)	(−0.48)	(0.12)	(2.24)	(−3.65)	(−2.01)	(−1.63)	(−0.49)	(0.16)
3	0.855***	−0.090	−0.059	−0.102	−0.053	−0.015	0.862***	−0.090	0.006	−0.048	0.032	0.065
	(3.21)	(−1.04)	(−0.72)	(−1.19)	(−0.63)	(−0.16)	(3.24)	(−1.01)	(0.06)	(−0.52)	(0.36)	(0.66)
4	0.890***	−0.006	−0.038	−0.145**	−0.128**	−0.096	0.841***	−0.073	−0.108	−0.189**	−0.183**	−0.153
	(3.88)	(−0.08)	(−0.58)	(−2.51)	(−2.09)	(−1.23)	(3.47)	(−0.96)	(−1.47)	(−2.29)	(−2.16)	(−1.61)
5	0.917***	0.042	−0.049	−0.084	−0.109	−0.096	0.979***	0.109	0.040	−0.014	−0.019	0.006
	(3.96)	(0.52)	(−0.70)	(−1.22)	(−1.58)	(−1.33)	(4.26)	(1.18)	(0.51)	(−0.21)	(−0.28)	(0.08)
6	1.019***	0.165**	0.057	0.009	−0.029	−0.016	0.899***	0.028	−0.064	−0.125*	−0.153**	−0.155**
	(4.83)	(2.19)	(0.78)	(0.13)	(−0.44)	(−0.20)	(4.13)	(0.38)	(−0.93)	(−1.93)	(−2.33)	(−2.17)
7	1.185***	0.307***	0.167*	0.129	0.073	0.097	1.058***	0.194**	0.084	0.008	−0.027	−0.008
	(5.14)	(3.10)	(1.84)	(1.42)	(0.82)	(0.98)	(4.84)	(2.17)	(0.94)	(0.10)	(−0.34)	(−0.09)
8	1.078***	0.218**	0.055	0.054	−0.029	0.008	1.210***	0.340***	0.167**	0.171**	0.092	0.080
	(4.51)	(2.31)	(0.69)	(0.60)	(−0.36)	(0.07)	(5.20)	(3.60)	(2.26)	(2.19)	(1.26)	(0.90)
9	1.350***	0.435***	0.240**	0.275**	0.174	0.152	1.112***	0.251***	0.094	0.073	−0.002	0.045
	(5.14)	(3.02)	(2.10)	(2.37)	(1.63)	(1.20)	(4.81)	(2.73)	(1.09)	(0.75)	(−0.02)	(0.36)
High	1.883***	0.971***	0.853***	0.846***	0.770***	0.735***	1.410***	0.496***	0.304**	0.337***	0.234**	0.185
	(6.45)	(4.91)	(5.37)	(5.53)	(5.01)	(4.38)	(5.39)	(3.46)	(2.56)	(2.80)	(2.02)	(1.46)
HML	1.556***	1.894***	1.361***	1.206***	0.916***	0.769***	1.047***	1.389***	0.771***	0.627**	0.312	0.179
	(4.53)	(5.64)	(5.31)	(4.66)	(4.08)	(3.03)	(3.24)	(4.43)	(3.24)	(2.41)	(1.51)	(0.73)

PennState
Smeal College of Business

Preliminary results from Azevedo, Hoegner, and Velikov (WP, 2023)



PennState
Smeal College of Business