

# Introduction to “(Re-)Imag(in)ing Price Trends”

By

Jingwen Jiang

University of Chicago

Bryan Kelly

Yale University, AQR Capital Management, and NBER

Dacheng Xiu

University of Chicago Booth School of Business

# Background

---

This paper considers using methods that flexibly learn price patterns that are most predictive of future returns to forecast future returns.

---

The raw predictor data are images from which authors model the predictive association between **images** and **future returns** using a convolutional neural network (CNN).

---

They claim by using CNN they can automatically identify context-independent predictive patterns which can give more accurate return predictions, translate into more profitable investment strategies and are robust to variations.

# Brief Intro

---

In the empirical designs, they first embed 1D time-series data into 2D images depicting price and volumes.

---

Then they feed each training sample into CNN to estimate the probability of a positive subsequent return over short(5-day), medium(20-day), and long (60day) horizons.

---

Afterward, they use CNN-based out-of-sample predictions as signals in several asset pricing analyses.

---

Finally, they attempt to interpret the predictive patterns identified by the CNN.

# Replication task

Mainly focus on:

- Data Preparation
- Model Design
- Workflow Design
- Performance Evaluation
- Interpretation

# Data

- The sample runs from 1993-2019 shows daily opening, high, low prices. The original paper constructs datasets consisting of three scales of horizons(5-day, 20-day, 60-day). Here we just collect the 20-day version. The total size of data is 8.6G.
- We already transferred the OHLC charts into images following the same procedures. Current images have the same resolution (64 \* 60) and added with moving average lines(MA) and volume bars(VB).



# Data

- Images labels take value **1** for positive returns ('up') and **0** for non-positive returns ('down'). In addition, we use **2** to mark the NaN value.

	Date	StockID	EWMA_vol	Retx	Retx_5d	Retx_20d	Retx_60d	Retx_week	Retx_month	Retx_quarter	...	Ret_week	Ret_month	Ret_quarter
<b>0</b>	2017-01-31	10001	0.000450	0.000000	4.370390e-07	-0.000002	-0.011860	NaN	-0.000002	NaN	...	NaN	-0.000002	NaN
<b>1</b>	2017-02-28	10001	0.000180	-0.003937	3.951997e-03	-0.003162	0.003953	NaN	0.003953	NaN	...	NaN	0.009953	NaN
<b>2</b>	2017-03-31	10001	0.000064	0.007936	-7.874612e-03	-0.015749	0.015748	-0.007875	-0.015749	0.017717	...	-0.007875	-0.015749	0.023704
<b>3</b>	2017-04-28	10001	0.000030	0.000000	9.999881e-03	0.016001	0.032002	0.010000	0.016001	NaN	...	0.010000	0.016001	NaN
<b>4</b>	2017-05-31	10001	0.000015	0.000000	4.370390e-07	0.015748	NaN	NaN	0.017717	NaN	...	NaN	0.023703	NaN

# Data: Label Format

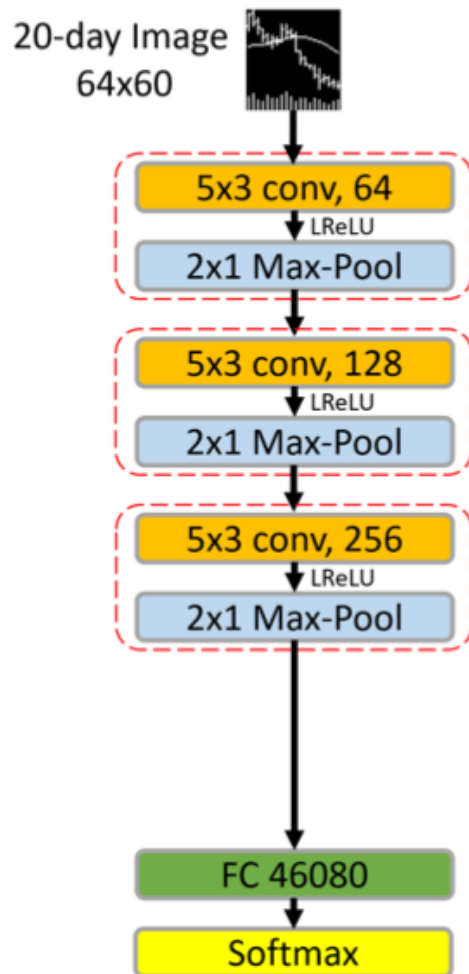
Date	2017-01-31 00:00:00
StockID	10001
EWMA_vol	0.00045
Retx	0.0
Retx_5d	0.0
Retx_20d	-0.000002
Retx_60d	-0.01186
Retx_week	NaN
Retx_month	-0.000002
Retx_quarter	NaN
Retx_tstat	0.0
Retx_5d_tstat	0.00097
Retx_20d_tstat	-0.003558
Retx_60d_tstat	-26.333479
MarketCap	133078.0
Retx_label	0
Retx_5d_label	1
Retx_20d_label	0
Retx_60d_label	0
window_size	20
next_month_ret_0delay	-0.000002
next_month_ret_1delay	-0.000002
Ret	0.0
log_ret	0.0
cum_log_ret	2.075332
Ret_week	NaN
Ret_month	-0.000002
Ret_quarter	NaN
Ret_5d	0.0
Ret_20d	-0.000002
Ret_60d	-0.005954
Ret_65d	0.001998
Ret_180d	NaN
Ret_250d	NaN
Ret_260d	NaN

Retx\_20d: < 0  
Retx\_20d\_label: 0

# Architecture Design

- A core building block consists of three operations:
  - convolution
  - activation
  - pooling
- In the paper, for 20-day images, they build a baseline CNN architecture with 3 conv blocks and connected with a fully connected layer as a classifier head.





You should refer to the design of the conv block in the original paper (including the selection of the size of the convolution kernel, the selection of the convolution method, the design of the pooling layer and the selection of the activation function, etc.)

# Working Flow

- Data Split
  - Consider dividing the entire sample into **training**, **validation** and **testing** samples.
  - In the original paper, they use **the first seven-year sample** (1993-1999) to **train** and **validate** the model, in which **70%** of the sample are randomly selected for training and the remaining **30%** for validation. The **remaining twenty years** of data comprise the out-of-sample test dataset.

- Loss and evaluation

- You can simply treat the prediction analysis as a classification problem. Use Cross Entropy Loss

$$L_{CE}(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$$

- To measure the classification accuracy, a true positive (TP) or true negative (TN) occurs when a predicted “up” probability of greater than 50% coincides with a positive realized return and a probability less than 50% coincides with a negative return. False positives and negatives (FP and FN) are the complementary outcomes.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

- For more evaluation metrics or methods, like Sharpe Ratio, please refer to the original paper.

- Train Process

- The author adopts several ways to combat over-fitting issue and aid efficient computation.
- They applied the Xavier initialization for weights in each layer, which guarantees faster convergence by scaling the initial weights.
- Other techniques like applying dropout, using batch normalization and early stopping also assists better performance.
- ❖ We recommend referring to the training details mentioned in the paper 3.3 when training the baseline model.

# Extensions

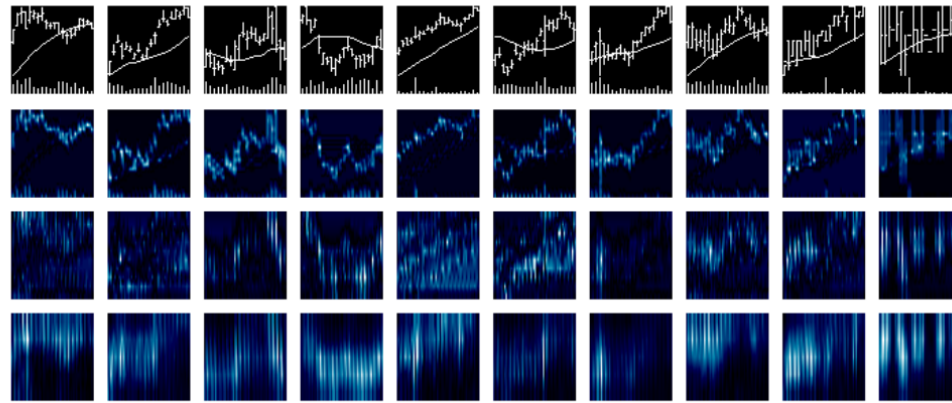
- Ablation studies and test robustness,
  - For example, you can perform the sensitivity analysis of the CNN prediction model to alternate choices in model architecture (e.g., varying the number of filters in each layer or varying the number of layers, like the paper shows in Table 18)

Table 18: Sensitivity to Model Structure and Estimation, I20R20

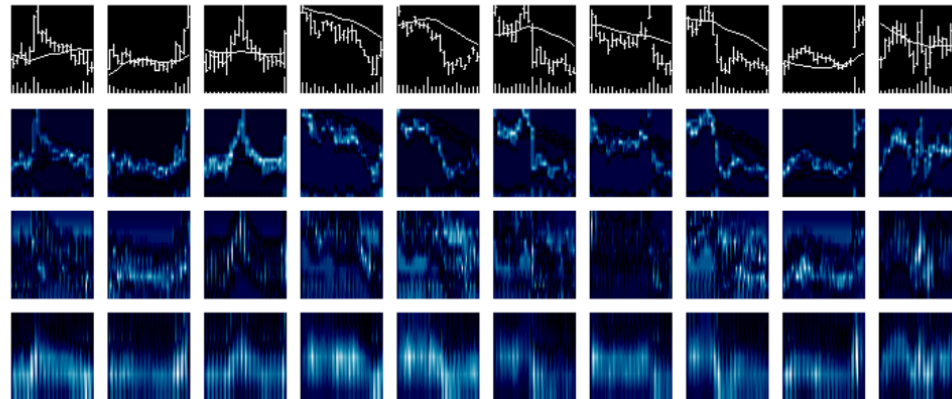
		Loss		Acc.		Correlation		Sharpe Ratio	
		V	T	V	T	Spearman	Pearson	EW	VW
Baseline		<b>0.688</b>	<b>0.692</b>	<b>0.540</b>	<b>0.525</b>	<b>0.052</b>	<b>0.032</b>	<b>2.18</b>	<b>0.56</b>
Filters (64)	32	0.688	0.691	0.541	0.526	0.053	0.032	1.91	0.43
	128	0.692	0.692	0.535	0.527	0.050	0.030	2.01	0.45
Layers (3)	2	0.688	0.692	0.540	0.525	0.047	0.028	1.61	0.19
	4	0.689	0.692	0.538	0.524	0.051	0.031	2.00	0.38
Dropout (0.50)	0.00	0.698	0.695	0.532	0.521	0.044	0.027	2.32	0.54
	0.25	0.691	0.693	0.536	0.525	0.050	0.030	2.11	0.56
	0.75	0.691	0.692	0.530	0.525	0.041	0.024	1.30	0.05
BN (yes)	no	0.685	0.691	0.549	0.528	0.059	0.037	2.44	0.58
Xavier (yes)	no	0.688	0.692	0.540	0.526	0.053	0.033	2.10	0.39
Activation (LReLU)	ReLU	0.689	0.693	0.538	0.520	0.052	0.031	1.71	0.35
Max Pool Size (2×1)	2×2	0.687	0.692	0.545	0.526	0.056	0.034	2.09	0.41
Filter Size (5×3)	3×3	0.689	0.693	0.538	0.522	0.050	0.028	1.39	0.28
	7×3	0.688	0.691	0.541	0.527	0.055	0.033	2.01	0.41
Dilation/Stride (2,1)/(3,1)	(2,1)/(1,1)	0.689	0.692	0.537	0.526	0.052	0.033	2.04	0.53
	(1,1)/(3,1)	0.689	0.692	0.538	0.526	0.049	0.030	1.66	0.17
	(1,1)/(1,1)	0.687	0.692	0.545	0.527	0.055	0.035	2.09	0.52

- Exploring of the interpretability of the CNN model
  - Using a visualization method (Grad-CAM) to understand how different image examples activate the regions of the CNN to trigger 'up' or 'down' return predictions.

Figure 12: I20R20 Grad-CAM for 20 Images from 2019



(a) Images Receiving “Up” Classification



(b) Images Receiving “Down” Classification

Note: Brighter regions of the heatmap correspond to regions with the higher activation. For each panel, the first row is the original images followed by the grad-CAM for each layer in the CNN.

- What's more

- ❖ We encourage you not limited to simple binary classification tasks, since the label files we provided consist of more meaningful attributes, containing both categorical and numerical values.

- For example, you can use the same 20-day horizon images to train your model to predict the return trend of different subsequent  $y$ -days even the detailed return values. ( $y$  can be 5, 20 even larger).

# Introduction to “Empirical Asset Pricing via Machine Learning”

By

Shihao Gu  
University of Chicago

Bryan Kelly  
Yale University, AQR Capital Management, and NBER

Dacheng Xiu  
University of Chicago Booth School of Business



# An Empirical Study of US Equity

Risk premium is difficult to measure: market efficiency forces return variation to be dominated by unforecastable news that obscures risk premiums.

This paper uses machine learning methods to predict asset's excess return.

- Linear models: OLS, elastic net
- Dimension reduction: PLS, PCR
- Generalized linear model
- Tree model: Gradient boosted regression tree, random forest
- Neural network

## Data and feature

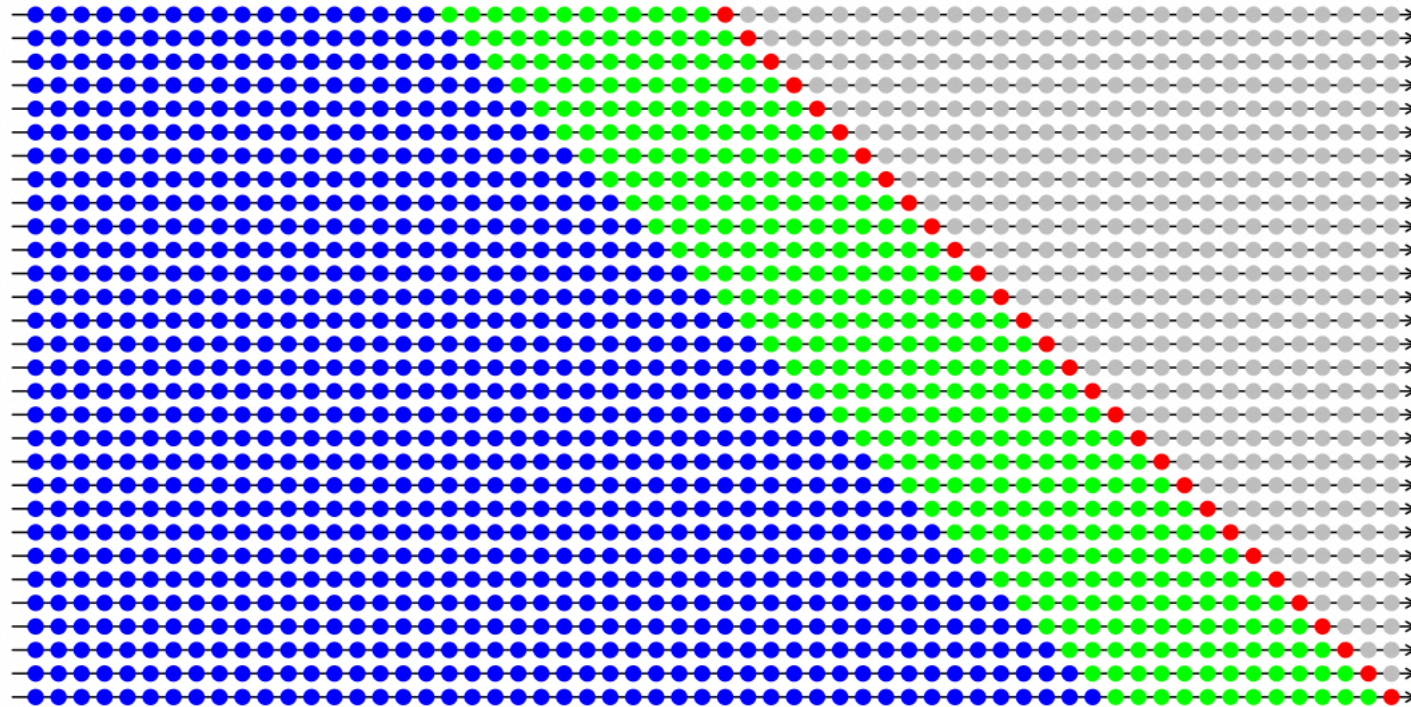
Monthly total individual equity returns for all firms listed in NYSE, AMEX, NASDAQ. Data traverses 60 years from 1957 to 2016.

- 94 firm characteristics
- 8 macroeconomic predictors
- 74 industry dummies

# A Recursive Evaluation Method

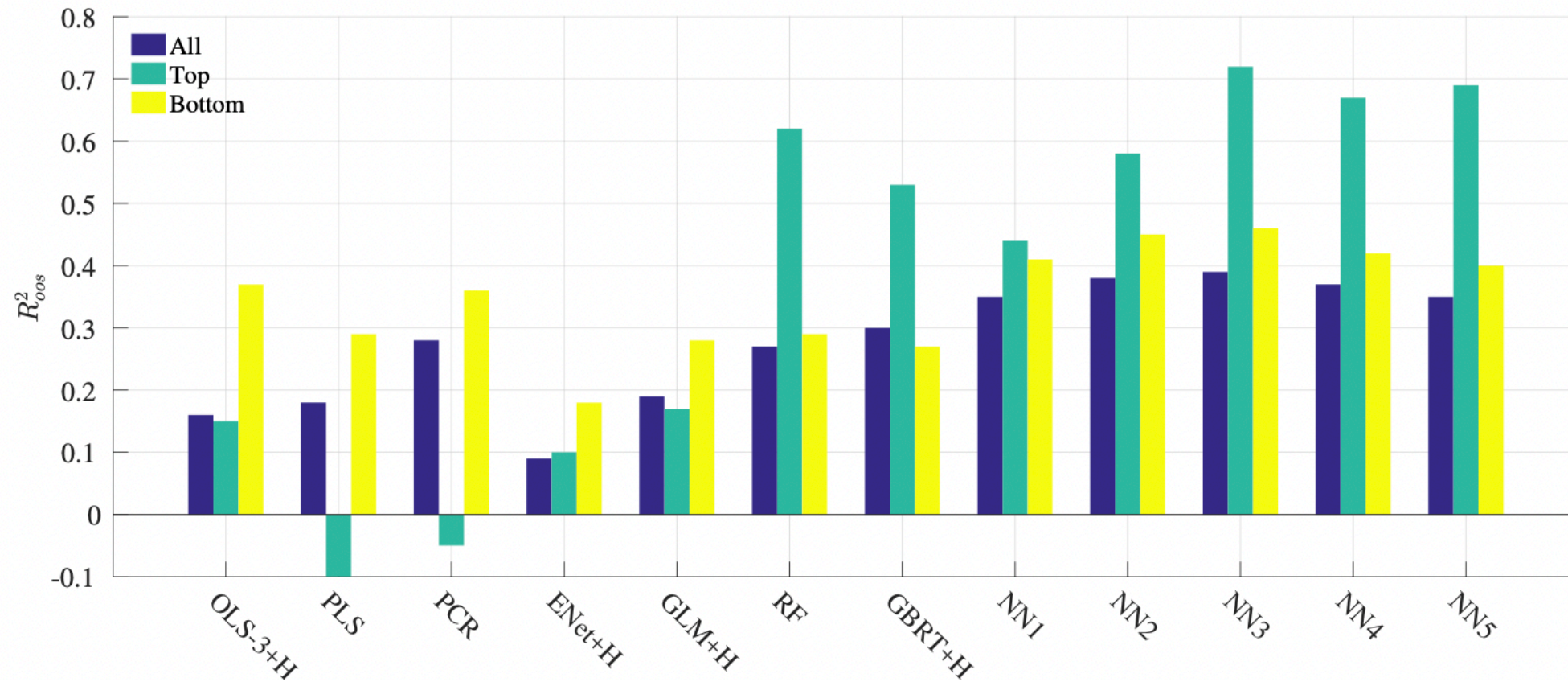
Divide the 60 years of data into 18 years of **training** sample (1957-1974), 12 years of **validation** sample (1975-1986), and the remaining 30 years for out-of-sample **testing** (1987-2016).

Adopt a **recursive performance evaluation scheme**.

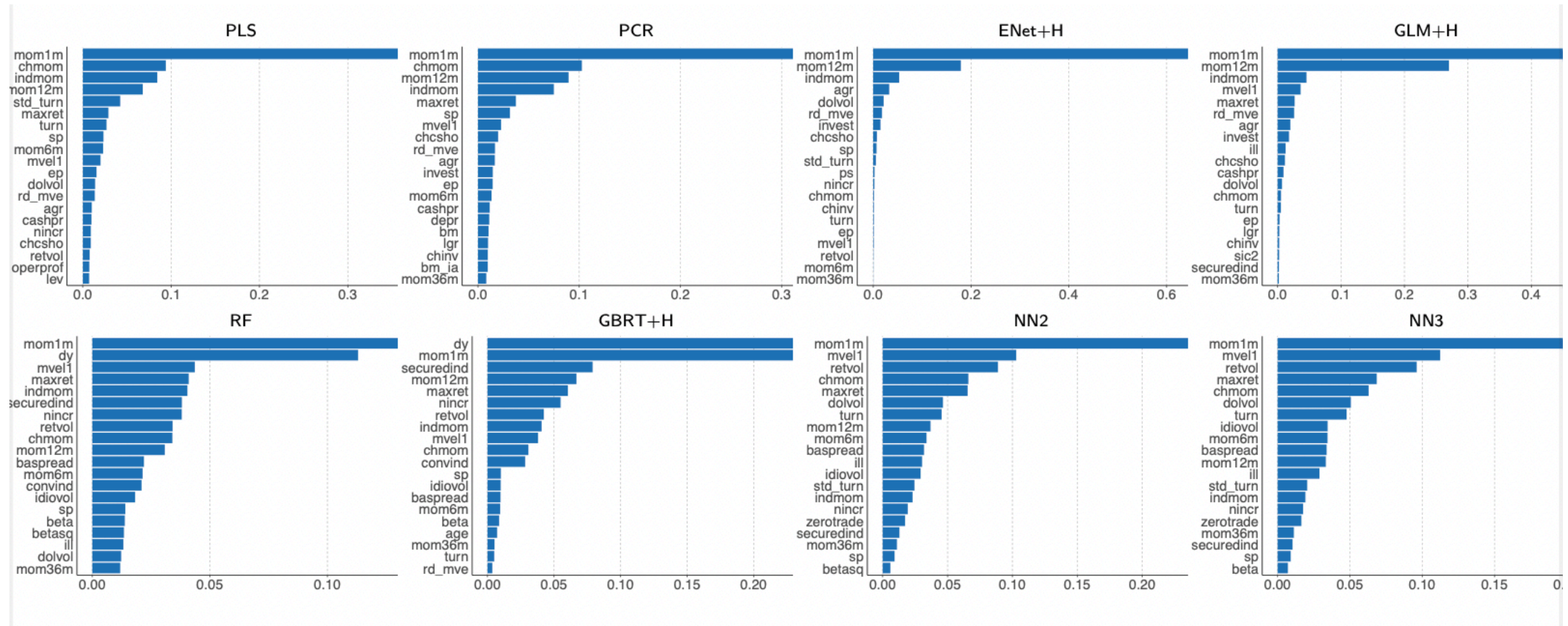


# Individual Stock Returns Prediction

Empirical results are not optimized over hyperparameters, and therefore are more conservative.



# Characteristic Importance



# Requirements for replication

- Replicate **at least 6 methods** (e.g., OLS, elastic net, PLS, PCR, random forest, Gradient Boosting, neural network, etc.), and analyze your results carefully. Hints of parameter chosen are presented in the paper.
- Include the **variable importance** in your analysis.
- Adopt the '**recursive evaluation method**' described above.
- You do not need to replicate the results of section 2.4: Portfolio forecast.
- Supplementary material can be helpful to you.