Internet Appendix to:
Machine learning in the Chinese stock market

Markus Leippold[*] Qian Wang[†] Wenyu Zhou[‡]

A Methodology

A.1 Simple linear regression

We take linear regression equipped with the Huber loss function as the reference model, because it is arguably the simplest tool that has been widely used for prediction. Since the linear regression model does not have any hyperparameters, we thus merge the validation sample into the training sample. The model imposes a linear structure on the conditional expectation of stock i's excess return $g(z_{it})$, i.e.,

$$r_{i,t+1} = g(z_{i,t};\theta) + \epsilon_{i,t} = z_{i,t}'\theta + \epsilon_{i,t}, \tag{A.1}$$

where $g(\cdot)$ is a function that describes the relation between excess stock return and the $p \times 1$ vector of predictors, $z_{i,t}$, and θ is the vector of coefficients which includes the intercept term. This model can be estimated handily, with a feasible closed-form solution when l_2 loss is adopted for the objective function. However, linear regression model estimated using the l_2 loss is vulnerable to outliers in the data, which can leads to very bad prediction performance.[1] Given the fact that financial returns and stock-level characteristics are often heavy-tailed, we instead minimize the Huber robust objective

[*] Department of Banking and Finance, University of Zurich, Plattenstrasse 14, 8032 Zurich, Switzerland, email: markus.leippold@bf.uzh.ch.

[†] Department of Banking and Finance, University of Zurich, Plattenstrasse 14, 8032 Zurich, Switzerland, email: qian.wang@bf.uzh.ch.

[‡] International Business School, Zhejiang University, Haining, Zhejiang 314400, China, email: wenyuzhou@intl.zju.edu.cn.

[1] We also consider the linear regression model equipped with the common l_2 loss function in our baseline analysis, which achieves very bad prediction performance as the corresponding R^2_{oos} is negative and its scale is more than an order of magnitude larger than other models. Therefore, we only report the results for the linear regression model equipped with the Huber loss (OLS+H). The results for OLS with l_2 loss are available upon request.

function following Gu et al. (2020), which is given by:

$$L_H(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T} H(r_{i,t+1} - g(z_{i,t};\theta); M), \tag{A.2}$$

where

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

The Huber loss function can be understood as a "trimmed" l_2 loss function, in which the threshold is determined by the tuning parameter M. In addition, the Huber loss function can be embedded into other machine learning methods as well, such as regularized linear models, and tree models. In the main context, we estimate OLS-3, LASSO, Enet, PLS, and GBRT using the Huber robust objective function following the practice in Gu et al. (2020).

A.2 Regularized linear regression

One big problem with simple linear regression is that the estimation results often becomes unreliable when there are a large number of covariates. On the one hand, in the high-dimensional setting, linear regression model estimated without regularization will be inconsistent.[2] On the other hand, some covariates can be highly correlated, or even redundant, resulting in the multicollinearity problem and efficiency loss.

Fortunately, there are many machine learning studies employing regularized linear models that can address these concerns. Popular methods include LASSO, the elastic net, Ridge regression, etc. In our empirical analysis, we include LASSO and the elastic net (Enet) as prediction models. The statistical properties of LASSO have been heavily studied for both *i.i.d.* and time series data.[3] The elastic net is a convex combination of LASSO and Ridge regression, and thus includes both as specially cases. These two methods share the same model specification with a simple linear regression in Eq. A.1, while the main difference is that they shrink the coefficients of irrelevant covariates

[2] Theoretically, the high-dimensional setting refers to the scenario in which the number of covariates P grows with the number of observations NT, or $P > NT$. However, as pointed out in Gu et al. (2020), it is reasonable to compare P with T, instead of with NT, due to the strong cross-sectional correlation between stock returns. Therefore, the prediction problem in this paper can also be understood as a quasi high-dimensional problem.

[3] See Bühlmann and Van De Geer (2011) for LASSO with *i.i.d.* data, and Medeiros and Mendes (2016) for time series data.

towards zero by imposing an extra penalty term to the original loss function. The objective function for LASSO is given by:

$$L_H^{\text{LASSO}}(\theta) = L_H(\theta) + \lambda \sum_{j=1}^{P} |\theta_j|, \tag{A.3}$$

where λ is a hyperparameter that controls the size of the penalty. Furthermore, the objective function for Enet takes the following form:

$$L_H^{\text{Enet}}(\theta) = L_H(\theta) + (1-\rho)\lambda \sum_{j=1}^{P} |\theta_j| + \frac{1}{2}\rho\lambda \sum_{j=1}^{P} \theta_j^2, \tag{A.4}$$

where λ plays the same role as in Eq. A.3, and ρ determines the relative weight between l_1 and l_2 penalties. Following the convention, we do not penalize the intercept term θ_0 in both models. It is clear that the objective function Eq. A.4 degenerates to the one for LASSO when $\rho = 0$, and to the Ridge regression when $\rho = 1$. The tuning of λ and ρ are specified in Section A.7. For more details on the elastic net, see the original paper by Zou and Hastie (2005).

A.3 Partial least squares

Partial least squares (PLS) is a classic dimensional reduction technique that can effectively extract signals among a large number of covariates. This method often outperforms regularized linear models when covariates are highly correlated, which is a common attribute of stock-level characteristics. Unlike LASSO and Enet, which directly penalize all covariates, PLS exploits the covariation between the predicted target and predictors by utilizing a model-averaging approach. We next briefly discuss the main context of the PLS regression. The PLS regression can be represented in its matrix form as follows:

$$R = (ZW_K)\theta_K + \tilde{E}, \tag{A.5}$$

where R is the $NT \times 1$ vector of stock returns $r_{i,t+1}$, Z is the corresponding $NT \times P$ vector of stock-level characteristics, W_K is a $P \times K$ transformation matrix, θ_K is the $K \times 1$ vecor of model parameters, and $\tilde{E}$ is the $NT \times 1$ vector of residuals.[4] For PLS regression, K is the only tuning parameter, which is determined via the validation procedure. The transformation matrix W_K plays

[4] For the sake of notational simplicity, we assume the panel data are balanced when introducing our prediction models.

the crucial role, as it projects the original covariate matrix Z onto a K-dimensional linear space. With a proper transformation matrix, this method can reserve the useful information and rule out the noise as we have $K < P$ in general.

The main idea of PLS regression is to search for the transformation matrix that maximizes the correlation between the forecast target and the transformed covariates. In this sense, the columns of W_K, denoted by $w_1, ..., w_K$, solve a sequence of optimization problems, i.e.,

$$w_j = \arg\max_{w} \ \mathrm{Cov}^2(R, Zw), \quad \text{s.t.} \quad w^{'}w = 1, \quad \mathrm{Cov}(Zw, Zw_l) = 0, \quad l = 1, 2, ..., j-1, \tag{A.6}$$

for all $j = 1, ..., K$. In our analysis, we adopt the build-in algorithm in the *sklearn* package of Python for the calculation of W_K. Lastly, given a solution for W_K, θ_K can be easily estimated by regressing R on ZW_K.

A.4 Tree models

Tree models, including random forests (RF), gradient boosted regression trees (GBRT), and other variants, are very important machine learning techniques. They are fully nonparametric, and flexible enough to handle both classification and regression problems. These two methods, however, both build on a number of basic trees. Therefore, they can also be understood as different ensemble methods depending on the specific procedures taken.

A basic tree is a set of decision rules that cluster observations into one of the multiple subgroups (partitions), usually named as "leaves." More precisely, the structure of a tree is determined by multiple decision nodes and the corresponding splitting variables. At each node, a splitting variable generates two disjoint branches based on a splitpoint. The basic tree "grows" by sequentially developing branches until reaching the "leaves" (terminal nodes). Mathematically, a basic tree with K leaves, and depth L, can be represented as:

$$g(z_{i,t}; \theta, K, L) = \sum_{k=1}^{K} \theta_k \mathbf{1}_{\{z_{i,t} \in C_k(L)\}}, \tag{A.7}$$

where $C_k(L)$ is the k-th partition of the data, and the depth L is the largest number of nodes in a complete branch (from the top node to any terminal node). Suppose stock i with characteristics $z_{i,t}$

is clustered into the k-th leaf, then the basic tree will return θ_k as the predicted stock return. It is noteworthy that θ_k here is defined as the sample average of outcomes within the k-th leaf based on the training data. There are ample studies on selecting splitting variables and splitpoints for the basic tree model, and we refer readers to James et al. (2013) for an excellent description. Even though basic trees are fairly flexible, they are vulnerable to the overfitting problem, which often severely impairs their performance in practice. To address this problem, multiple regularization methods have been proposed, among which the most popular ones are GBRT and RF. We next briefly describe these two methods.

A.4.1 Gradient boosted regression trees

The main idea of the GBRT model is to produce a "strong learner" by recursively combining the prediction results from many "weak learners."[5] The GBRT model proceeds as follows. We start with a GBRT model with only two simple trees. First, we build a simple tree of depth L to fit stock returns based on stock-level characteristics. Next, a second simple tree of the same depth L is built to fit the residuals from the first tree. The forecast from the first tree plus the forecast from the second tree multiplied by the learning rate $v \in (0,1)$ is the ensemble prediction of this basic GBRT model. To build a GBRT model with B trees, we simply build another $B-2$ trees in the similar manner. At each step for the b-th tree, we grow a new tree to fit the residuals from the GBRT model with $b-1$ trees, and add the product of its forecast and the learning rate to the previous final output to form the final output of the GBRT model with b trees. Repeat this step until B trees are grown.

It is noteworthy that there are three hyperparameters for the GBRT model, including the depth of trees L, the number of trees B, and the learning rate v. These parameters are determined based on the validation procedure in Section A.7.

A.4.2 Random forests

Random forest is another important ensemble method that also utilizes forecasts from many underlying simple trees. Unlike GBRT, a random forest model relies on a more general technique

[5] Simple tree models are usually thought as "weak learners," because they are prone to overfit the data and thus perform badly.

known as “bagging” (Breiman, 2001) or bootstrap aggregation. The main idea of the method is to build B separate trees, and average over their forecasts to reduce prediction variation. More precisely, each of those B trees is trained on a bootstrapped sample of the original data, and uses only a randomly drawn subset of covariates for developing branches. Since each tree generated in this way is identically distributed, the expectation of the final output is the same as the expectation of a single tree. However, taking the average can significant benefit from reducing the variance while keeping the bias at a minimal level. Lastly, we note that there are also three tuning parameters for random forests, which include the depth of trees L, the number of trees B, and the number of covariates for building simple trees. See Section A.7 for more details.

A.5 Variable subsample aggregation (VASA)

De Nard et al. (2020) introduce a new subsampling procedure, which they call VASA. VASA reduces the dimension by a subset selection of the predictors, similar as for LASSO or subset selection methods. However, it does not suffer as much from high variability and model-selection bias, as it averages over multiple subsampled (factor) model predictions. The conditional expected return is still assumed to be a linear function and is obtained by averaging over B OLS predictions, each trained on a (pseudo) random subset of the P predictors, i.e.,

$$g_i^{\text{VASA}}(z_{i,t}) \overset{\text{def}}{=} \sum_{b=1}^{B} \omega_b g_{i,b}(\tilde{z}_{i,t,b}) \overset{\text{def}}{=} \sum_{b=1}^{B} \omega_b(\alpha_{i,b} + \tilde{z}'_{i,t,b}\tilde{\beta}_{i,b}) \ , \tag{A.8}$$

where $\omega_b \in [0,1]$ is the weight of the b-th OLS prediction with $\sum_{b=1}^{B} \omega_b = 1$. Additionally, $\tilde{z}_{i,t,b}$ and $\tilde{\beta}_{i,b}$ are κ_b-dimensional vectors where κ_b represents the dimension (subsample size) of submodel b. We assume that the optimal subsampling size only depends on i and t but is constant across the submodels $\kappa \equiv \kappa_b$. The number of submodels B and their dimension κ are tuning parameters. Then, for each submodel $b = 1, \dots, B$, the estimation problem is given by:

$$\underset{a_{i,b},\tilde{b}_{i,b}}{\text{argmin}} \sum_{t=1}^{T} (r_{i,t+1} - a_{i,b} - \tilde{z}'_{i,t,b}\tilde{b}_{i,b})^2. \tag{A.9}$$

To compute the subsampling probabilities, we follow De Nard et al. (2020) and use the in-sample $\mathrm{R}^2_{i,p}$ as a variable importance measure. For more details, we refer to the original paper.

A.6 Neural networks

The final prediction method we introduce is the neural network, which is arguably the most popular machine learning technique in recent years. They have been widely applied for complex machine learning problems, such as computer vision, automated driving, and natural language processing. It is well-known that neural networks can approximate any smooth functions sufficiently well, which is ensured by the universal approximation theorem (Hornik et al., 1989). However, there is still much to learn about these models given the fact that they are among the least transparent and least interpretable machine learning methods. In fact, the name "neural networks" derives from the history that they were first developed as models for the human brain, which also remains highly mysterious given its complex structure.

For our analysis, we focus on the feed-forward, multi-layer neural networks. Similar to a human's decision process, such models consist of an "input layer" for stock-level characteristics, one or more "hidden layers" that process the interactions between those predictors, and an "output layer" that generates a linear output. We consider neural networks with up to five hidden layers. Each layer consists of a certain number of neurons, which are built with the commonly-used rectified linear unit (ReLU), i.e., $\sigma(x) = \max(0, x)$.[6] As an illustration, the predicted stock return of the NN3 model can be written as

$$r_{i,t+1} = \alpha_1 + W_1\sigma(\alpha_2 + W_2\sigma(\alpha_3 + W_3\sigma(\alpha_4 + W_4 z_{i,t}))) + \epsilon_{i,t+1}, \tag{A.10}$$

where the activation function, $\sigma(\cdot)$, is applied elementwise, and $\{\alpha_1, ..., \alpha_4, W_1, ...W_4\}$ is the set of biases and weight matrices to be estimated. The network architectures and the corresponding numbers of parameters of NN1 - NN5 are summarized in Table A.1.

[6] ReLU is often preferred to other alternative activation functions, such as sigmoid, hyperbolic, and softmax, because it overcomes the vanishing gradient problem and its derivative is easy to calculate.

Table A.1

Number of neurons and parameters for all neural network models.

Model	Hidden Layers	Number of Parameters
NN1	32	32(P+1) + 33
NN2	32, 16	32(P+1) + 545
NN3	32, 16, 8	32(P+1) + 673
NN4	32, 16, 8, 4	32(P+1) + 705
NN5	32, 16, 8, 4, 2	32(P+1) + 713

In our analysis, all neural networks are trained using TensorFlow, a powerful machine learning system. As in De Nard et al. (2020), we adopt the Adam optimization algorithm (Kingma and Ba, 2014), early stopping, batch normalization (Ioffe and Szegedy, 2015), ensembles, and dropout (Srivastava et al., 2014) when training our models.

A.7 Hyperparamters

In Table A.2, we summarize the hyperparameters for all prediction models and the corresponding specifications.

Table A.2

Hyperparameters for all models. The table summarizes the ranges of hyperparameters for all machine learning models.

	OLS-3+H	PLS	LASSO+H	Enet+H	GBRT+H
Huber loss $M = 1.35$	√		√	√	√
Specification	bm, mve, mom1m	K	$\lambda \in (10^{-4}, 10^{-1})$	$\rho = 0.5$ $\lambda \in (10^{-4}, 10^{-1})$	#Depth $L = 1 \sim 3$ #Trees $B = 1 \sim 1000$ Learning Rate $LR \in \{0.01, 0.1\}$
	RF	VASA	NN1-NN5		
Specification	#Depth $L = 1 \sim 7$ #Trees $B = 100 \sim 300$ #Features $f = 3 \sim 50$	#Subsamples $B = 1 \sim 300$ #Components $K = 1 \sim 50$	L_1 penalty $\lambda \in (10^{-5}, 10^{-2})$ Learning Rate $LR \in (10^{-4}, 10^{-2})$ Batch Size $B \in \{64, 512, 2048, 10000\}$ Epochs = 100 Patience = 5 Adam Para.= Default Ensemble = 10		

B Details on statistical inference methods

We briefly introduce the statistical inference methods used in the main text for comparing the performance of different prediction models. Here we mainly focus on implementing two statistical tests: the unconditional superior predictability test (Hansen, 2005) and the conditional superior predictability test (Li, Liao and Quaedvlieg, 2020). Reader interested in the technical details should refer to the original papers. To facilitate the discussion, we first introduce some notations following the convention in the forecast evaluation literature. Let $\{F_t^\dagger\}_{t\geq 1}$ denote the variable of interest to be predicted, which in our case is the monthly stock price. We want to compare the performance of J competing models relative to a benchmark model in terms of the forecast $\{F_t^\dagger\}_{t\geq 1}$. Let $\{F_{0,t}\}_{t=1}^n$ and $\{F_{j,t}\}_{t=1}^n$ for $1 \leq j \leq J$ denote the predicted values of the variable of interest in n periods. For a given loss function $L(\cdot,\cdot)$, we let $Y_{j,t}$ denote the performance of the j-th competing model relative to the benchmark model in period t, i.e.,

$$Y_{j,t} = L(F_t^\dagger, F_{j,t}) - L(F_t^\dagger, F_{0,t}). \tag{A.11}$$

It is noteworthy that the benchmark model has a better prediction performance in period t if $Y_{j,t} \geq 0$.

B.1 Unconditional superior predictive ability test

The unconditional superior predictive ability (USPA) test, which is first studied in White (2000) and later refined by Hansen (2005), has been widely used in finance research. We adopt the refined version in Hansen (2005) instead of the original reality check (RC) for data snooping in White (2000), because the former usually has larger statistical power. The USPA test directly compares the unconditional (average) performance of different competing prediction models relative to the benchmark model, with the null hypothesis given by:

$$H_0^{USPA} : \mathbb{E}[Y_{j,t}] \geq 0, \quad \text{for all } 1 \leq j \leq J.$$

It is noteworthy that both the USPA test and CSPA test build on the probabilistic properties of $Y_{j,t}$ directly instead of the underlying loss function and prediction models. Let $\mathbf{Y}_t = (Y_{1,t}, ..., Y_{J,t})'$, $\boldsymbol{\mu} = E[\mathbf{Y}_t]$, $\bar{\mathbf{Y}} = n^{-1}\sum_{t=1}^n \mathbf{Y}_t$, and $\bar{Y}_j = n^{-1}\sum_{t=1}^n Y_{j,t}$ for $j = 1, ..., J$. Under some regularity conditions, one can show that:

$$n^{1/2}(\bar{\mathbf{Y}} - \boldsymbol{\mu}) \xrightarrow{d} \mathcal{N}(0, \boldsymbol{\Omega}), \tag{A.12}$$

where $\boldsymbol{\Omega}$ is the asymptotic covariance matrix of $n^{1/2}(\bar{\mathbf{Y}} - \boldsymbol{\mu})$. Based on this result, Hansen (2005) proposed use of the following test statistic:

$$T_n^{USPA} = \min\Big[\min_{j=1,...,J} \frac{n^{1/2}\bar{Y}_j}{\hat{\omega}_j},\ 0\Big],$$

in which the estimator of the asymptotic variance of $\hat{\omega}_j^2$ is given by:

$$\hat{\omega}_j^2 = \hat{\gamma}_{0,j} + 2\sum_{i=1}^{n-1} K(n,i)\hat{\gamma}_{i,j},$$

where $\hat{\gamma}_{i,j} = n^{-1}\sum_{l=1}^{n-i}(Y_{j,l} - \bar{Y}_j)((Y_{j,l+i} - \bar{Y}_j))$, $i = 0, 1, ..., n-1$, and $K(n,i)$ is the kernel weight for a stationary bootstrap (Hansen (2005)). To invoke a null distribution that is based on Eq. A.12, one also needs to select a consistent estimator of $\boldsymbol{\mu}$, for which Hansen (2005) recommends using $\hat{\mu}_j = \bar{Y}_j 1\{n^{1/2}\bar{Y}_j \geq \sqrt{2\log\log n}\}$. Intuitively, the null hypothesis will be rejected if T_n^{USPA} is smaller than the critical value at the given significance level. However, the critical value can not be calculated analytically in general and the bootstrap method is recommended. For reader's convenience, the implementation steps of the test is presented as follows.

Algorithm for the USPA test (Hansen, 2005)

Step 1. Generate B resembles $\{\mathbf{Y}_t^{(b)} : t = 1, ..., n\}_{b=1}^B$ from $\{\mathbf{Y}_t : t = 1, ..., n\}$ using the stationary bootstrap method in Hansen (2005).[7]

Step 2. For each resample, calculate $T_{(b),n}^{USPA} = \min[\min_{j=1,...,J}(n^{1/2}\bar{Y}_j/\hat{\omega}_j),\ 0]$.

Step 3. Calculate the bootstrap p-value: $\hat{p} = \frac{1}{B}\sum_{b=1}^B 1\{T_{(b),n}^{USPA} < T_n^{USPA}\}$. Reject the USPA null hypothesis at significance level α if $\hat{p} < \alpha$.

B.2 Conditional superior predictive ability test

One potential limitation of the USPA test is that two prediction models may have an identical performance on average but can behave very differently under different macroeconomic (market-level)

[7] In this study, we set $B = 10,000$.

conditions. The USPA test cannot detect such differences because it only focuses on the average (unconditional) relative performance of prediction models. Motivated by this observation, Li, Liao and Quaedvlieg (2020) develop an innovative conditional superior predictive ability (CSPA) test based on the developments in the series estimation literature (Chernozhukov et al. (2013); Li, Liao and Quaedvlieg (2020)). The null hypothesis of the CSPA test is given by:

$$H_0^{CSPA} : \mathbb{E}[Y_{j,t}|X_t = x] \geq 0, \quad \text{for all } x \in \mathcal{X}, 1 \leq j \leq J,$$

where X_t is the variable relates to the macroeconomic (market-level) conditions, such as the GDP growth rate, the macroeconomic uncertainty and the volatility index (VIX), and $\mathcal{X}$ is the support of X_t. It is worth noting that $\mathcal{X}$ should be compact, while the compactness condition can be trivially satisfied by implementing any one-to-one transformations in Li, Liao and Gao (2020). The main idea of the CSPA test is to directly estimate the conditional expectation functions $\mathbb{E}[Y_{j,t}|X_t = x]$ using the series estimation method and conduct uniform inference with the help of the strong approximation theory for time series data. Following the notation in Li, Liao and Quaedvlieg (2020), we let $h_{j,n} = \mathbb{E}[Y_{j,t}|X_t = x]$ for all $1 \leq j \leq J$ and $P(x) = (p_1(x), ..., p_{m_n})^\top$ be an $m_n \times 1$ vector of basis functions, such as Legendre polynomial, power series, spline, etc. Here m_n is the number of basis terms used to estimate $h_{j,n}$, which is a divergent number that depends on the sample size n. The conditional expectation functions can then be estimated by a OLS-type regression as follows:

$$\hat{h}_{j,n}(x) = P(x)^\top \hat{b}_{j,n}, \tag{A.13}$$

where:

$$\hat{b}_{j,n} = \Big(n^{-1}\sum_{t=1}^{n} P(X_t)P(X_t)^\top\Big)^{-1}\Big(n^{-1}\sum_{t=1}^{n} P(X_t)Y_{j,t}\Big).$$

To conduct statistical inference on $\hat{h}_{j,n}$, one need to derive its asymptotic distribution, which lies in the asymptotic properties of $\hat{b}_{j,n}$. However, as pointed out in Li, Liao and Quaedvlieg (2020), this is not a standard problem since the dimension of $\hat{b}_{j,n}$ divergences with the sample size n, making the classical central limit theorem invalid. The solution to such inference is to utilize the strong approximation theory for growing-dimensional statistics, which is first developed in Chernozhukov et al. (2013) and later generalized to time series data in Li and Liao (2020). Let $u_t = (u_{1,t}, ..., u_{J,t})^\top$ denote the vector of nonparametric regression error term, where $u_{j,t} = Y_{j,t} - h_j(X_t)$. Li and Liao

(2020) and Li, Liao and Quaedvlieg (2020) show that $n^{-1/2}\sum_{t=1}^{n} u_t \otimes P(X_t)$,[8] which is the random part of $\hat{H}_n(x) = (\hat{h}_{1,n}(x), ..., \hat{h}_{J,n}(x))^\top$ conditional on X_t, and can be approximated sufficiently well by some Jm_n-dimensional Gaussian vector $\tilde{N} \sim \mathcal{N}(0, A_n)$, where the covariance matrix A_n is given by:

$$A_n = \text{Var}\Big(n^{-1/2}\sum_{t=1}^{n} u_t \otimes P(X_t)\Big). \tag{A.14}$$

For the estimation of A_n, Li, Liao and Quaedvlieg (2020) recommend using the pre-whitened HAC estimator and we follow their suggestion when implementing the test. Given the estimated covariance matrix $\hat{A}_n$, the $Jm_n \times Jm_n$ covariance matrix of the estimators $(n^{1/2}(\hat{b}_{1,n} - b^*_{1,n})^\top, ..., n^{1/2}(\hat{b}_{J,n} - b^*_{J,n})^\top)^\top$[9] is given by:

$$\hat{\Omega}_n = \Big(I_J \otimes \hat{Q}_n\Big)^{-1} \hat{A}_n \Big(I_J \otimes \hat{Q}_n\Big)^{-1}, \tag{A.15}$$

where I_J is the $J \times J$ identity matrix and $\hat{Q}_n = (n^{-1}\sum\limits_{t=1}^{n} P(X_t)P(X_t)^\top)^{-1}$. It follows that the standard deviation function of $n^{1/2}(\hat{h}_{j,n}(x) - h_j(x))$ can be estimated by:

$$\hat{\sigma}_{j,n}(x) = \Big(P(x)^\top \hat{\Omega}(j,j) P(x)\Big)^{1/2}, \tag{A.16}$$

where $\hat{\Omega}(j,j)$ is the diagonal submatrix of $\hat{\Omega}_n$, which corresponds to $n^{1/2}(\hat{b}_{j,n} - b^*_{j,n})^\top$. Then we can conduct statistical inference based on $\hat{h}_{j,n}(x)$ for all $1 \le j \le J$ and $x \in \mathcal{X}$. For reader's convenience, we borrow the following implementation algorithm directly from Li, Liao and Quaedvlieg (2020).

Algorithm for the CSPA test (Li, Liao and Quaedvlieg, 2020)

Step 1. Simulate a Jm_n-dimensional Gaussian vectors $\xi = (\xi_1^\top, ..., \xi_J^\top)^\top \sim \mathcal{N}(0, \hat{\Omega}_n)$, where $\xi_j^\top$ is m_n-dimensional and $\hat{\Omega}$ is defined in Eq. A.15. Repeat this step for B times and store $\{\xi^{(1)}, ..., \xi^{(B)}\}$.

Step 2. For each $\xi^{(b)}$, $1 \le b \le B$, calculate $\hat{t}^{(b)}_{j,n}(x) = P(x)^\top \xi^{(b)}_j$ for all $1 \le j \le J$. Set $\tilde{\gamma}_n = 1 - 0.1/\log(n)$. Let $\hat{K}_n$ be the $\tilde{\gamma}_n$-quantile of $\max_{1\le j\le J}\sup_{x\in\mathcal{X}} \hat{t}_{j,n}(x)$ based on the sample $\{\hat{t}^{(b)}_{j,n}(x)\}_{1\le b\le B, 1\le j\le J}$.

[8] $\otimes$ denotes the Kronecker product.

[9] Here $b^*_{j,n}$ is some constant vector, such that $P(x)b^*_{j,n}$ can uniformly approximate $h_j(x)$ sufficiently well on the support of x. For more details, see Li, Liao and Quaedvlieg (2020).

Step 3. Set

$$\hat{\mathcal{V}}_n = \left\{(j,x) : \hat{h}_{j,n}(x) \leq \min_{1\leq j\leq J} \inf_{x\in\mathcal{X}} \left(\hat{h}_{j,n}(x) + n^{-1/2}\hat{K}_n\hat{\sigma}_{j,n}(x)\right) + 2n^{-1/2}\hat{K}_n\hat{\sigma}_{j,n}(x)\right\},$$

where $\hat{\sigma}_{j,n}(x)$ is defined in Eq. A.16.

Step 4. Set $\hat{k}_{n,1-\alpha}$ to be the $(1-\alpha)$-quantile of $\sup_{(j,x)\in\hat{\mathcal{V}}_n} \hat{t}_{j,n}$ and calculate

$$\hat{\eta}_{n,1-\alpha} = \min_{1\leq j\leq J} \inf_{x\in\mathcal{X}} \left(\hat{h}_{j,n}(x) + n^{-1/2}\hat{k}_{n,1-\alpha}\hat{\sigma}_{j,n}(x)\right).$$

Reject the CSPA null hypothesis at significance level α if $\hat{\eta}_{n,1-\alpha} < 0$.

When implementing the CSPA test in the main text, we set $B = 10,000$ and apply the Akaike Information Criterion (AIC) to select the basis functions following the procedure in Li, Liao and Quaedvlieg (2020). One beneficial byproduct of the CSPA test is that it also allows us to learn exactly what are the macroeconomic (market-level) conditions under which the benchmark model outperforms (or does not outperform) the alternatives by plotting the conditional expectation functions $h_{j,n}(x)$ for diagnosis. We also conduct these standard exercises when comparing the performance of different prediction models.

C Stock characteristics

C.1 Variable list

All stock characteristics are summarized in Table C.1.

Table C.1. Details on stock characteristics.

No.	Acronym	Stock Characteristics	Author(s)	Year, Journal	Data Source	Frequency
1	*absacc*	Absolute accruals	Bandyopadhyay, Huang & Wirjant	2010, WP	CSMAR	Semi-annual
2	*acc*	Working capital accruals	Sloan	1996, TAR	CSMAR	Semi-annual
3	*agr*	Asset growth	Cooper, Gulen & Schill	2008, JF	CSMAR	Annual
4	*beta*	Beta	Fama & MacBeth	1973, JPE	WIND	Monthly
5	*betasq*	Beta squared	Fama & MacBeth	1973, JPE	CSMAR	Monthly
6	*bm*	Book-to-market	Rosenberg, Reid & Lanstein	1985, JPM	CSMAR	Quarterly
7	*bm_ia*	Industry-adjusted book to market	Asness, Porter & Stevens	2000, WP	CSMAR	Quarterly
8	*cash*	Cash holdings	Palazzo	2012, JFE	CSMAR	Quarterly
9	*cashdebt*	Cash flow to debt	Ou & Penman	1989, JAE	CSMAR	Quarterly
10	*cashspr*	Cash productivity	Chandrashekar & Rao	2009, WP	CSMAR	Quarterly
11	*cfp*	Cash flow to price ratio	Desai, Rajgopal & Venkatachalam	2004, TAR	CSMAR	Quarterly
12	*cfp_ia*	Industry-adjusted cash flow to price ratio	Asness, Porter & Stevens	2000, WP	CSMAR	Quarterly
13	*chato*	Change in asset turnover	Soliman	2008, TAR	CSMAR	Quarterly
14	*chatoia*	Industry-adjusted change in asset turnover	Soliman	2008, TAR	CSMAR	Quarterly
15	*chcsho*	Change in shares outstanding	Pontiff & Woodgate	2008, JF	CSMAR	Monthly
16	*chempia*	Industry-adjusted change in employees	Asness, Porter & Stevens	1994, WP	CSMAR	Annual
17	*chinv*	Change in inventory	Thomas & Zhang	2002, RAS	CSMAR	Quarterly
18	*chmom*	Change in 6-month momentum	Gettleman & Marks	2006, WP	WIND	Monthly
19	*chpm*	Change in profit margin	Soliman	2008, TAR	CSMAR	Quarterly
20	*chpmia*	Industry-adjusted change in profit margin	Soliman	2008, TAR	CSMAR	Quarterly
21	*chtx*	Change in tax expense	Thomas & Zhang	2011, JAR	CSMAR	Quarterly
22	*cinvest*	Corporate investment	Titman, Wei & Xie	2004, JFQA	CSMAR	Quarterly
23	*currat*	Current ratio	Ou & Penman	1989, JAE	CSMAR	Quarterly
24	*depr*	Depreciation / PP&E	Holthausen & Larcker	1992, JAE	CSMAR	Semi-annual
25	*divi*	Dividend initiation	Michaely, Thaler & Womack	1995, JF	CSMAR	Annual
26	*divo*	Dividend omission	Michaely, Thaler & Womack	1995, JF	CSMAR	Annual
27	*dolvol*	Yuan trading volume	Chordia, Subrahmanyam & Anshuman	2001, JFE	CSMAR	Monthly
28	*dy*	Dividend to price	Litzenberger & Ramaswamy	1982, JF	CSMAR	Annual
29	*ear*	Earnings announcement return	Kishore, Brandt, Santa-Clara & Venkatachalam	2008, WP	CSMAR	Quarterly
30	*egr*	Growth in common shareholder equity	Richardson, Sloan, Soliman & Tuna	2005, JAE	CSMAR	Quarterly
31	*gma*	Gross profitability	Novy-Marx	2013, JFE	CSMAR	Quarterly
32	*grCAPX*	Growth in capital expenditures	Anderson & Garcia-Feijoo	2006, JF	CSMAR	Semi-annual
33	*herf*	Industry sales concentration	Hou & Robinson	2006, JF	CSMAR	Quarterly
34	*hire*	Employee growth rate	Bazdresch, Belo & Lin	2014, JPE	CSMAR	Annual
35	*idiovol*	Idiosyncratic return volatility	Ali, Hwang & Trombley	2003, JFE	WIND	Monthly

Table C.1: Details on stock characteristics (continued)

No.	Acronym	Stock Characteristics	Author(s)	Year, Journal	Data Source	Frequency
36	*ill*	Illiquidity	Amihud	2002, JFM	WIND	Monthly
37	*invest*	Capital expenditures and inventory	Chen & Zhang	2010, JF	CSMAR	Annual
38	*lev*	Leverage	Bhandari	1988, JF	CSMAR	Quarterly
39	*lgr*	Growth in long-term debt	Richardson, Sloan, Soliman & Tuna	2005, JAE	CSMAR	Quarterly
40	*maxret*	Maximum daily return	Bali, Cakici & Whitelaw	2011, JFE	WIND	Monthly
41	*mom12m*	12-month momentum	Jegadeesh	1990, JF	WIND	Monthly
42	*mom1m*	1-month momentum	Jegadeesh & Titman	1993, JF	WIND	Monthly
43	*mom6m*	6-month momentum	Jegadeesh & Titman	1993, JF	WIND	Monthly
44	*mom36m*	36-month momentum	Jegadeesh & Titman	1993, JF	WIND	Monthly
45	*ms*	Financial statement score	Mohanram	2005, RAS	CSMAR	Annual
46	*mve*	Size	Banz	1981, JFE	CSMAR	Monthly
47	*mve_ia*	Industry-adjusted size	Asness, Porter & Stevens	2000, WP	CSMAR	Monthly
48	*nincr*	Number of earnings increases	Barth, Elliott & Finn	1999, JAR	CSMAR	Quarterly
49	*operprof*	Operating profitability	Fama & French	2015, JFE	CSMAR	Quarterly
50	*orgcap*	Organizational capital	Eisfeldt & Papanikolaou	2013, JF	CSMAR	Quarterly
51	*pchcapx_ia*	% Industry adjusted % change in capital expenditures	Ou & Penman	1989, JAE	CSMAR	Annual
52	*pchcurrat*	% change in current ratio	Ou & Penman	1989, JAE	CSMAR	Quarterly
53	*pchdepr*	% change in depreciation	Holthausen & Larcker	1992, JAE	CSMAR	Semi-annual
54	*pchgm_pchsale*	% change in gross margin - % change in sales	Abarbanell & Bushee	1998, TAR	CSMAR	Quarterly
55	*pchquick*	% change in quick ratio	Ou & Penman	1989, JAE	CSMAR	Quarterly
56	*pchsale_pchinvt*	% change in sales - % change in inventory	Abarbanell & Bushee	1998, TAR	CSMAR	Quarterly
57	*pchsale_pchrect*	% change in sales - % change in A/R	Abarbanell & Bushee	1998, TAR	CSMAR	Quarterly
58	*pchsale_pchxsga*	% change in sales - % change in SG&A	Abarbanell & Bushee	1998, TAR	CSMAR	Quarterly
59	*pchsaleinv*	% change sales-to-inventory	Ou & Penman	1989, JAE	CSMAR	Quarterly
60	*pctacc*	Percent accruals	Hafzalla, Lundholm & Van Winkle	2011, TAR	CSMAR	Semi-annual
61	*pricedelay*	Price delay	Hou & Moskowitz	2005, RFS	WIND	Monthly
62	*ps*	Financial statements score	Piotroski	2000, JAR	CSMAR	Quarterly
63	*quick*	Quick ratio	Ou & Penman	1989, JAE	CSMAR	Quarterly
64	*rd*	R&D increase	Eberhart, Maxwell & Siddique	2004, JF	CSMAR	Quarterly
65	*rd_mve*	R&D to market capitalization	Guo, Lev & Shi	2006, JBFA	CSMAR	Quarterly
66	*rd_sale*	R&D to sales	Guo, Lev & Shi	2006, JBFA	CSMAR	Quarterly
67	*realestate*	Real estate holdings	Tuzel	2010, RFS	CSMAR	Quarterly
68	*volatility*	Return volatility	Ang, Hodrick, Xing & Zhang	2006, JF	WIND	Monthly
69	*roaq*	Return on assets	Balakrishnan, Bartov & Faurel	2010, JAE	CSMAR	Quarterly
70	*roavol*	Earnings volatility	Francis, LaFond, Olsson & Schipper	2004, TAR	CSMAR	Quarterly

Table C.1: Details on stock characteristics (continued)

No.	Acronym	Stock Characteristics	Author(s)	Year, Journal	Data Source	Frequency
71	*roeq*	Return on equity	Hou, Xue & Zhang	2015, RFS	CSMAR	Quarterly
72	*roic*	Return on invested capital	Brown & Rowe	2007, WP	CSMAR	Quarterly
73	*rsup*	Revenue surprise	Kama	2009, JBFA	CSMAR	Quarterly
74	*salecash*	Sales to cash	Ou & Penman	1989, JAE	CSMAR	Quarterly
75	*saleinv*	Sales to inventory	Ou & Penman	1989, JAE	CSMAR	Quarterly
76	*salerev*	Sales to receivables	Ou & Penman	1989, JAE	CSMAR	Quarterly
77	*sgr*	Sales growth	Lakonishok, Shleifer & Vishny	1994, JF	CSMAR	Quarterly
78	*sp*	Sales to price	Barbee, Mukherji, & Raines	1996, FAJ	CSMAR	Quarterly
79	*std_dolvol*	Volatility of liquidity (yuan trading volume)	Chordia, Subrahmanyam & Anshuman	2001, JFE	CSMAR	Monthly
80	*std_turn*	Volatility of liquidity (share turnover)	Chordia, Subrahmanyam, & Anshuman	2001, JFE	CSMAR	Monthly
81	*stdacc*	Accrual volatility	Bandyopadhyay, Huang & Wirjanto	2010, WP	CSMAR	Quarterly
82	*stdcf*	Cash flow volatility	Huang	2009, JEF	CSMAR	Quarterly
83	*tang*	Debt capacity/firm tangibility	Almeida & Campello	2007, RFS	CSMAR	Quarterly
84	*tb*	Tax income to book income	Lev & Nissim	2004, TAR	CSMAR	Quarterly
85	*turn*	Share turnover	Datar, Naik & Radcliffe	1998, JFM	CSMAR	Monthly
86	*zerotrade*	Zero trading days	Liu	2006, JFE	CSMAR	Monthly
87	*atr*	Abnormal Turnover Ratio	Pan, Tang & Xu	2015, RF	WIND, CSMAR	Monthly
88	*er_trend*	Trend factor	Liu, Zhou & Zhu	2020, WP	WIND, CSMAR	Monthly
89	*largestholderrate*	Largest shareholder ownership	Gul, Kim & Qiu	2010, JFE	CSMAR	Annual
90	*top10holderrate*	Top 10 shareholders ownership	Gul, Kim & Qiu	2010, JFE	CSMAR	Annual
91	*soe*	State owned enterprise indicator			CSMAR	Annual
92	*private*	Private enterprise indicator			CSMAR	Annual
93	*foreign*	Foreign enterprise indicator			CSMAR	Annual
94	*others*	Other enterprise indicator			CSMAR	Annual

C.2 Variable construction

We closely follow the definitions in Green et al. (2017) and the original papers to construct the stock-level characteristics.

(1) *acc*: We follow the definition of accruals in Sloan (1996) to construct *acc*, i.e.,

$$acc = [(\Delta\text{CA} - \Delta\text{CASH}) - (\Delta\text{CL} - \Delta\text{STD} - \Delta\text{TP}) - \text{Dep}]/\text{Total Assets},$$

where Δ represents the difference between two consecutive periods, CA, CASH, CL, STD, TP, Dep, denote current assets, cash/cash equivalents, current liabilities, debt included in current liabilities, income tax payable, depreciation and amortization expense, respectively. These data are acquired from CSMAR.

(2) *absacc*: Absolute value of *acc*.

(3) *agr*: Annual percentage change in total assets. Data of total assets are acquired from CSMAR.

(4) *beta*: We estimate stock-level beta using weekly returns and value-weighted market returns for three years ending month $t-1$ with at least 52 weeks of returns. Stock returns are acquired from the WIND database.

(5) *betasq*: Stock-level market beta squared.

(6) *bm*: Book-to-market ratio, which equals the book value of equity divided by market capitalization. Data are acquired from CSMAR.

(7) *bm_ia*: This is the industry-adjusted book-to-market ratio introduced in Asness et al. (2000),

$$bm\_ia_{it} = bm_{it} - bm_{Iit},$$

where bm_{Iit} is the equally-weighted average book-to-market ratio of firms in firm i's industry. As firms' industries are reported annually in CSMAR, we let firm i's current industry to be the one reported in the year prior to the current month.

(8) *cash*: Cash and cash equivalents divided by average total assets. Related data are reported in quarterly reports and are acquired from CSMAR.

(9) *cashdebt*: Earnings divided by total liabilities, which is defined similar to that in Ou and Penman (1989). Data are acquired from CSMAR.

(10) *cashspr*: Cash productivity, which is defined as quarter-end market capitalization plus long-term debt minus total assets divided by cash and equivalents. Related data are contained in quarterly reports and acquired from CSMAR.

(11) *cfp*: Operating cash flows divided by quarter-end market capitalization. Related data are contained in quarterly reports and acquired from CSMAR.

(12) *cfp_ia*: This is the industry-adjusted operating cash flows. The way of adjustment is similar to that for *bm_ia*. Data are acquired from CSMAR.

(13) *chato*: Change in sales divided by average total assets. Quarterly data on sales and total assets are acquired from CSMAR.

(14) *chato_ia*: Industry-adjusted change in sales divided by average total assets. Data are acquired from CSMAR.

(15) *chcsho*: Monthly percentage change in shares outstanding. Monthly data on shares outstanding are acquired from CSMAR.

(16) *chempia*: Industry-adjusted change in the number of employees. Related data are available annually on CSMAR and the way of industry adjustment is similar as that for *bm_ia*.

(17) *chinv*: Change in inventory scaled by total assets. Data are available quarterly on CSMAR.

(18) *chmom*: Cumulative returns from months $t-6$ to $t-1$ minus months $t-12$ to $t-7$. Stock returns are acquired from WIND database.

(19) *chpm*: Change in income before extraordinary items scaled by sales. Related data are acquired from CSMAR.

(20) *chpm_ia*: Industry-adjusted change in income before extraordinary items scaled by sales. Data are acquired from CSMAR.

(21) *chtx*: Percentage change in taxes from quarter $t-1$ to t. Data are acquired from CSMAR.

(22) *cinvest*: Change over one quarter in fixed assets divided by sales - average of this variable for prior three quarters; if sales are zero, then scale by 0.01. Data are acquired from CSMAR.

(23) *currat*: The ratio of current assets to current liabilities. Data are acquired from CSMAR.

(24) *depr*: Depreciation divided by fixed assets. Data are acquired from CSMAR.

(25) *divi*: A dummy variable that equals to 1 if company pays dividends this year but did not in prior year. Data are acquired from CSMAR.

(26) *divo*: A dummy variable that equals to 1 if company does not pay dividends this year but did in prior year. Data are acquired from CSMAR.

(27) *dolvol*: Natural logarithm of trading volume times price per share from month $t-2$. Data are acquired from CSMAR.

(28) *dy*: Total dividends divided by market capitalization at year end. Data are acquired from CSMAR.

(29) *ear*: Sum of daily returns in three days around earnings announcement. Data are acquired from CSMAR.

(30) *egr*: Quarterly percentage change in book value of equity. Data are acquired from CSMAR.

(31) *gma*: Revenue minus cost of goods sold divided by lagged total assets. Quarterly data are acquired from CSMAR.

(32) $grCAPX$: Percentage change in capital expenditures from year $t-2$ to year t. Data are acquired from CSMAR.

(33) *herf*: Sum of squared percentage sales in industry for each company. Sales data and industry code are acquired from CSMAR.

(34) *hire*: Percentage change in number of employees. Related data are acquired from CSMAR.

(35) *idiovol*: Standard deviation of residuals of weekly returns on weekly equally-weighted market returns for three years prior to month end. Data are acquired from the WIND database.

(36) *ill*: Average of daily (absolute return/RMB volume) in month t. Daily data are acquired from WIND database.

(37) *invest*: The sum of annual change in fixed assets and annual change in inventories divided by lagged total assets. Data are acquired from CSMAR.

(38) *lev*: Total liabilities divided by quarter-end market capitalization. Quarterly data are acquired from CSMAR.

(39) *lgr*: Quarterly percentage change in total liabilities. Data are acquired from CSMAR.

(40) *maxret*: Maximum daily return from returns during month $t - 1$. Daily returns are acqired from WIND database.

(41) *mom12m*: 11-month cumulative returns ending one month before month end. Stock returns are acqired from WIND database.

(42) *mom1m*: 1-month cumulative return. Stock returns are acqired from WIND database.

(43) *mom6m*: 5-month cumulative returns ending one month before month end. Stock returns are acqired from WIND database.

(44) *mom36m*: Cumulative returns from months $t - 36$ to $t - 13$. Stock returns are acqired from WIND database.

(45) *ms*: Sum of eight indicator variables for fundamental performance following the corresponding definitions in Mohanram (2005). Data are acquired from CSMAR.

(46) *mve*: Natural log of market capitalization at end of month $t - 1$. Related data are acquired from CSMAR.

(47) *mve_ia*: Industry adjusted natural log of market capitalization at end of month $t - 1$. Related data are acquired from CSMAR.

(48) *nincr*: Number of consecutive quarters (up to eight quarters) with an increase in earnings. Earnings data are acquired from CSMAR.

(49) *operprof*: Quarterly operating profit divided by lagged common shareholders' equity. Related data are acquired from CSMAR.

(50) *orgcap*: Capitalized management expenses. This characteristic uses expense data acquired from CSMAR and is constructed according to the definition in Eisfeldt and Papanikolaou (2013). Data are acquired from CSMAR.

(51) *pchcapx_ia*: Industry adjusted percentage change in capital expenditure. Data are acquired from CSMAR.

(52) *pchcurrat*: Percentage change in current ratio (current liabilities divided by current assets). Data are acquired from CSMAR.

(53) *pchdepr*: Percentage change in depreciation. Data are acquired from CSMAR.

(54) *pchgm_pchsale*: Percentage change in gross margin minus Percentage change in sales. Data are acquired from CSMAR.

(55) *pchquick*: Percentage change in quick ratio. Data are acquired from CSMAR.

(56) *pchsale_pchinvt*: Quarterly percentage change in sales minus quarterly percentage change in inventory. Data are acquired from CSMAR.

(57) *pchsale_pchrect*: Quarterly percentage change in sales minus quarterly percentage change in receivables. Data are acquired from CSMAR.

(58) *pchsale_pchxsga*: Quarterly percentage change in sales minus quarterly percentage change in management expenses. Data are acquired from CSMAR.

(59) *pchsaleinv*: Quarterly percentage change in sales-to-inventory. Data are acquired from CSMAR.

(60) *pctacc*: Same as *acc* except that the numerator is divided by the absolute value of net income; if net income = 0 then net income set to 0.01 for denominator. Data are acquired from CSMAR.

(61) *pricedelay*: The proportion of variation in weekly returns for 36 months ending in month t explained by four lags of weekly market returns incremental to contemporaneous market return. Stock returns are acquired from WIND database.

(62) *ps*: Sum of nine indicator variables that are defined similarly as in Piotroski (2000). Related data are acquired from CSMAR.

(63) *quick*: Quick ratio = (current assets - inventory) / current liabilities. Data are acquired from CSMAR.

(64) *rd*: An indicator variable equal to 1 if R&D expense as a percentage of total assets has an increase greater than 5%. Data are acquired from CSMAR.

(65) *rd_mve*: R&D expense divided by end-of-quarter market capitalization. Data are acquired from CSMAR.

(66) *rd_sale*: R&D expense divided by quarterly sales. Data are acquired from CSMAR.

(67) *realestate*: Investment real estates divided by fixed assets. Data are acquired from CSMAR.

(68) *volatility*: Standard deviation of daily returns from month $t-1$. Stock returns are acquired from WIND.

(69) *roaq*: Income before extraordinary items divided by one quarter lagged total assets. Related data are acquired from CSMAR.

(70) *roavol*: Standard deviation of 16 quarters of income before extraordinary items divided by average total assets. Data are acquired from CSMAR.

(71) *roeq*: Income before extraordinary items divided by lagged common shareholders' equity. Related data are acquired from CSMAR.

(72) *roic*: Quarterly earnings before interest and taxes minus nonoperating income divided by non-cash enterprise value. Related data are acquired from CSMAR.

(73) *rsup*: Sales from quarter t minus sales from quarter $t-1$ divided by quarter-end market capitalization. Related data are acquired from CSMAR.

(74) *salecash*: Quarterly sales divided by cash and cash equivalents. Data are acquired from CSMAR.

(75) *saleinv*: Quarterly sales divided by total inventory. Data are acquired from CSMAR.

(76) *salerev*: Quarterly sales divided by accounts receivable. Data are acquired from CSMAR.

(77) *sgr*: Quarterly percentage change in sales. Data are acquired from CSMAR.

(78) *sp*: Quarterly sales divided by quarter-end market capitalization. Data are acquired from CSMAR.

(79) *std_dolvol*: Monthly standard deviation of daily RMB trading volume. Data are acquired from CSMAR.

(80) *std_turn*: Monthly standard deviation of daily share turnover. Data are acquired from CSMAR.

(81) *stdacc*: Standard deviation of 16 quarters of accruals from month $t-16$ to $t-1$. Data are acquired from CSMAR.

(82) *stdcf*: Standard deviation for 16 quarters of net cash flows divided by sales. Data are acquired from CSMAR.

(83) *tang*: Cash holdings + 0.715 × receivables + 0.547× inventory + 0.535 × fixed assets/total assets. Data are acquired from CSMAR.

(84) *tb*: Tax income, defined as current tax expense divided by enterprise income tax rate in China (25%), divided by total income. Data are acquired from CSMAR.

(85) *turn*: Average monthly trading volume for month $t-3$ to $t-1$ scaled by number of shares outstanding in month t. Related data are acquired from CSMAR.

(86) *zerotrade*: Turnover weighted number of zero trading days in month $t-1$. Related data are acquired from CSMAR.

(87) *atr*: The abnormal turnover ratio (*atr*) is constructed following the definition in Pan et al. (2016). Specifically, for stock i in month t, we run the following regression using daily data

from month $t-7$ to $t-1$,

$$\mathrm{DTR}_{i,t} = \beta_1 + \beta_2 \times \mathrm{DMRT}_t + \sum_{j=1}^{K} c_j \times \mathrm{Dummy\_Event}(j)_{i,t} + \epsilon_{i,t},$$

where $\mathrm{DTR}_{i,t}$ is stock i's daily turnover ratio, DMRT_t is the market turnover ratio, $\mathrm{Dummy\_Event}(j)$, $j = 1, ..., K$ is a sequence of event dummy variables. The aggregated $\hat{\epsilon}_{i,t}$ for the entire month t is defined to be the abnormal turnover ratio (atr). Data are acquired from both WIND database and CSMAR.

(88) *er_trend*: This trend factor is constructed following the definition in Liu et al. (2020). For the sake of simplicity, we refer readers to the original paper for more details. Data are acquired from both WIND database and CSMAR.

(89) *largestholderrate*: Percentage of common shares owned by the largest shareholder. Data are acquired from CSMAR.

(90) *top10holderrate*: Percentage of common shares owned by top 10 shareholders. Data are acquired from CSMAR.

(91) *soe*: A dummy variables that equals 1 if the firm is state-owned. Data are acquired from CSMAR.

(92) *private*: A dummy variable that equals 1 if the firm is privately-owned. Data are acquired from CSMAR.

(93) *foreign*: A dummy variable that equals 1 if the firm is controlled by foreign investors. Data are acquired from CSMAR.

(94) *others*: A dummy variable that equals 1 if the firm is not state-owned, private, or foreign. Data are acquired from CSMAR.

C.3 Discussions on accounting treatment in China

In China, listed companies are required to follow the Chinese Accounting Standards (CAS), a.k.a., the Chinese Generally Accepted Accounting Principles (China GAAP), when running their businesses. The CAS mainly consist of two sets of accounting standards: (1) the Accounting Standards for Business Enterprises (ASBEs) for general companies; and (2) the Accounting Standards for Small-sized Business Enterprises (ASSBEs). We focus on the ASBEs as they are most relevant to publicly traded companies.

The revised ASBEs were introduced by the Ministry of Finance of the PRC in 2006, which consist of one Basic Standard, 38 Specific Standards, and the related application guidance. Interestingly, the ASBEs are more than 90% the same as the International Financial Reporting Standards (IFRS), making most account titles readily comparable across these two systems.[10] In China, all publicly traded companies are required by the government to follow the ASBEs when filing their financial statements. It is worth pointing out that even though public companies in China use a set of more traditional accounting standards before 2006, most account titles needed for signal construction in our study are still available as of today, and more importantly, comparable to those under the current ASBEs.[11]

On the other hand, as the US GAAP are also very similar to the IFRS, most account titles mentioned in Section C.2 can be clearly linked to their counterparts under the ASBEs. Hence, we can follow the definitions in the original papers to construct these stock-level characteristics in most cases. In rare situations, account titles may not directly have their counterparts in the ASBEs, such as the SG&A expenses. When this happens, we conduct some simple calculations to get their equivalents under the ASBEs accordingly.

Even though the difference between the ASBEs and the US GAAP looks inessential in our case, it is still helpful to get a more comprehensive picture of these two systems. There are several key differences between the ASBEs and the US GAAP. Firstly, the CAS only allow the historical cost

[10] See Qu and Zhang (2010) for a comprehensive comparison between the ASBEs and the IFRS.

[11] The ASBEs were first established in 1992 when China began transforming towards a market-oriented economy. The initial standards absorbed many important contexts from the IFRS but still kept some conventional accounting practice, especially when dealing with debt. After China entered the WTO, the Ministry of Finance of China issued the Accounting Regulations for Enterprises in 2000, which was already very close to the current ASBEs.

valuation method when valuating fixed assets, while the US GAAP allows companies to choose between the historical cost valuation method and re-evaluating the assets. In this sense, the CAS are more conservative when dealing with fixed assets. Secondly, regrading inventory, the CAS bans the usage of the "Last in, First out' (LIFO) method, which is nevertheless allowed by the US GAAP. Thirdly, the CAS requires fiscal year in accounts must begin on January 1st, while the US GAAP let companies to determine the starting date of their fiscal years. Fourthly, companies submit their financial statements to the government and file their tax returns on a monthly basis in China, while returns can be filed on a quarterly or bi-monthly basis under US GAAP. Last but not the least, the CAS stipulate that expenses are classified according to function, whereas the US GAAP generally classify expenses by nature. Overall, we find that these differences have not changed the construction of stock-level characteristics much in our empirical study.

Finally, it is worth noting that all publicly traded companies in China are required to provide a set of complete financial statements in their half-year reports, including the balance sheet, the income statement, and the cash flow statement, whose formats are the exactly same as that for annual reports (Accounting Standards for Enterprises No. 32). In addition, a large number of public companies in China also provide such information in their quarterly reports. In contrast, the US GAAP only requires public companies to provide a condensed set of financial statements, in which many account titles are not reported. Although interim statements are not necessarily audited both in China and the US, the extra financial information may still help investors to learn about companies better.

C.4 Classification of stock characteristics

Table C.4

Classification of stock characteristics.

Category	**Variable**
C_size	*mve*, *mve_ia*, *herf*, *chinv*, *chcsho*
C_beta	*beta*, *betasq*
C_mom	*mom1m*, *mom6m*, *mom12m*, *mom36m*, *chmom*, *er_trend*, *maxret*
C_liq	*std_dolvol*, *zerotrade*, *atr*, *chaotia*, *std_turn*, *ill*, *turn*, *dolvol*, *pricedelay*
C_vol	*idiovol*, *ear*, *volatility*, *roavol*
C_own	*top10holderrate*, *LargestHolderRate*
C_bpr	*bm*, *bm_ia*, *cfp*, *cfp_ia*, *sp*, *cashspr*, *invest*, *realestate*, *depr*
C_ey	*roeq*, *roaq*, *divo*, *absacc*, *divi*, *salerev*, *chempia*, *nincr*, *chpmia*, *stdacc*, *chtx*, *cash*, *roic*, *chpm*, *stdcf*, *chao*, *dy*, *acc*, *pctacc*, *saleinv*, *operprof*, *pchsale_pchrect*, *salecash*, *tb*, *gma*, *pchdepr*
C_growth	*egr*, *orgcap*, *sgr*, *pchgm_pchsale*, *rsup*, *pchsaleinv*, *rd_sale*, *rd_mve*, *rd*, *cinvest*, *pchsale_pchxsga*, *pchsale_pchinvt*, *agr*, *grCAPX*, *hire*
C_lever	*lev*, *pchquick*, *pchcapx_ia*, *lgr*, *quick*, *ps*, *tang*, *currat*, *ms*, *pchcurrat*, *cashdebt*

Table C.5

Details on Macroeconomic State Variables.

Acronym	Variable	Definition	Frequency	Reference
dp	Dividend Price Ratio	Dividends are 12-month moving sums of dividends paid in the A-share market. The dividend price ratio is the difference between the log of dividends and the log of weighted average stock price in China's A-share market.	Monthly	Welch (2008), Gu et al. (2020)
de	Dividend Payout Ratio	The dividend payout ratio is the difference between the log of dividends and the log of earnings of all stocks listed in China's A-share market.	Annual	Welch (2008), Gu et al. (2020)
bm	Book-to-Market Ratio	The book-to-market ratio is the ratio of book value to market value for all stocks listed in China's A-share market.	Monthly	Welch (2008), Gu et al. (2020)
svar	Stock Variance	The stock variance is computed as sum of squared daily returns on the SSE Composite Index.	Monthly	Welch (2008), Gu et al. (2020)
ep	Earnings Price Ratio	Earnings are 12-month moving sums of earnings of all stocks in the China's A-share market. The earnings price ratio is the difference between the log of weighted average earnings per share and the log of weighted average stock price in China's A-share market.	Monthly	Welch (2008), Gu et al. (2020)

Table C.5.

Details on macroeconomic state variables (continued)

Acronym	Variable	Definition	Frequency	Reference
ntis	Net Equity Expansion	The net equity expansion is the ratio of 12-month moving sums of net issues in China's A-share market divided by the total end-of-year market capitalization of A-share stocks.	Monthly	Welch (2008), Gu et al. (2020)
tms	Term Spread	The term spread is the difference between the yield on 10-year government bond and the 1-year government bond.	Monthly	Welch (2008), Gu et al. (2020)
infl	Inflation	The inflation is the monthly consumer price index from 2000-2020 from the National Bureau of Statistics of China.	Monthly	Welch (2008), Gu et al. (2020)
mtr	Monthly Turnover	The monthly turnover is the ratio of monthly trading volume measured in Chinese yuan to the average daily market value of all stocks in China's A-share market.	Monthly	Baker and Stein (2004)
m2gr	M2 Growth Rate	The monthly M2 growth rate (YoY) from 2000 to 2020 is from the National Bureau of Statistics of China.	Monthly	Chen (2009)
itgr	International Trade Volume Growth Rate	The international trade volume growth rate (YoY) from 2000 to 2020 is from the National Bureau of Statistics of China.	Monthly	Rapach et al. (2013)

D Time variations of R^2_{oos} and variable importance

To obtain a better understanding of model predictability, we also explore the time variations in the out-of-sample R^2_{oos} of our models. In Fig. D.1, we plot the yearly out-of-sample R^2_{oos}, which we define as:

$$R^2_{\text{oos},S}(t) = 1 - \frac{\sum_{i\in\mathcal{I}_t}(r_{i,t} - \hat{r}^{(S)}_{i,t})^2}{\sum_{i\in\mathcal{I}_t} r^2_{i,t}}, \tag{A.17}$$

where $\mathcal{I}_t$ is the set of tradable stocks in year t in the testing sample.

We make two observations. First, all models, except GBRT, experience a significant drop in R^2_{oos} in 2018. In that year, all models excluding GBRT produce negative R^2_{oos}, indicating that naïve predictions of zero returns would have beaten them. However, the dysfunctionality of machine learning models in 2018 is likely due to the Chinese stock market's persistent fall caused by the severe trade conflicts between China and the US. This finding points out a potential weakness for machine learning techniques when predicting stock returns: their performances can be vulnerable to unexpected systematic risk, such as, in this case, the political risk related to a trade war between the US and China. From this perspective, it is even more surprising that GBRT still achieves a positive out-of-sample R^2_{oos} in 2018, which is even larger than those in other periods. We conjecture that its generic model properties cause GBRT's unusual performance. For example, if the model heavily relies on predictors relating to price trends, it may still generate a good performance in 2018. Still, compared to other machine learning methods such as neural networks, GBRT generally produces a lower out-of-sample R^2 in periods other than 2018.

Second, regularized linear models and VASA based on linear submodels not only produce negative R^2_{oos}s in 2018 but also in 2017. In 2017, the Chinese stock market went through a persistent boom for large stocks as the CSI 300 Index[12] increased by almost 30%, and most large stocks had positive returns in almost every month. As noted previously, monthly returns of large stocks in the Chinese market are more challenging to predict, especially for linear models, which is likely why PLS, LASSO, Enet, and VASA attain negative R^2_{oos}s in 2017. On the other hand, neural networks, especially NN2, NN4, and NN5, produce positive R^2_{oos}s in 2017, indicating their prediction performance is quite

[12] The CSI 300 is a capitalization-weighted stock market index designed to replicate the performance of the top 300 stocks traded on the Shanghai Stock Exchange and the Shenzhen Stock Exchange.

robust to specific market conditions associated with certain subgroups of stocks.

In Fig. D.2, we take a closer look at the time variability of the variable importance for NN4.

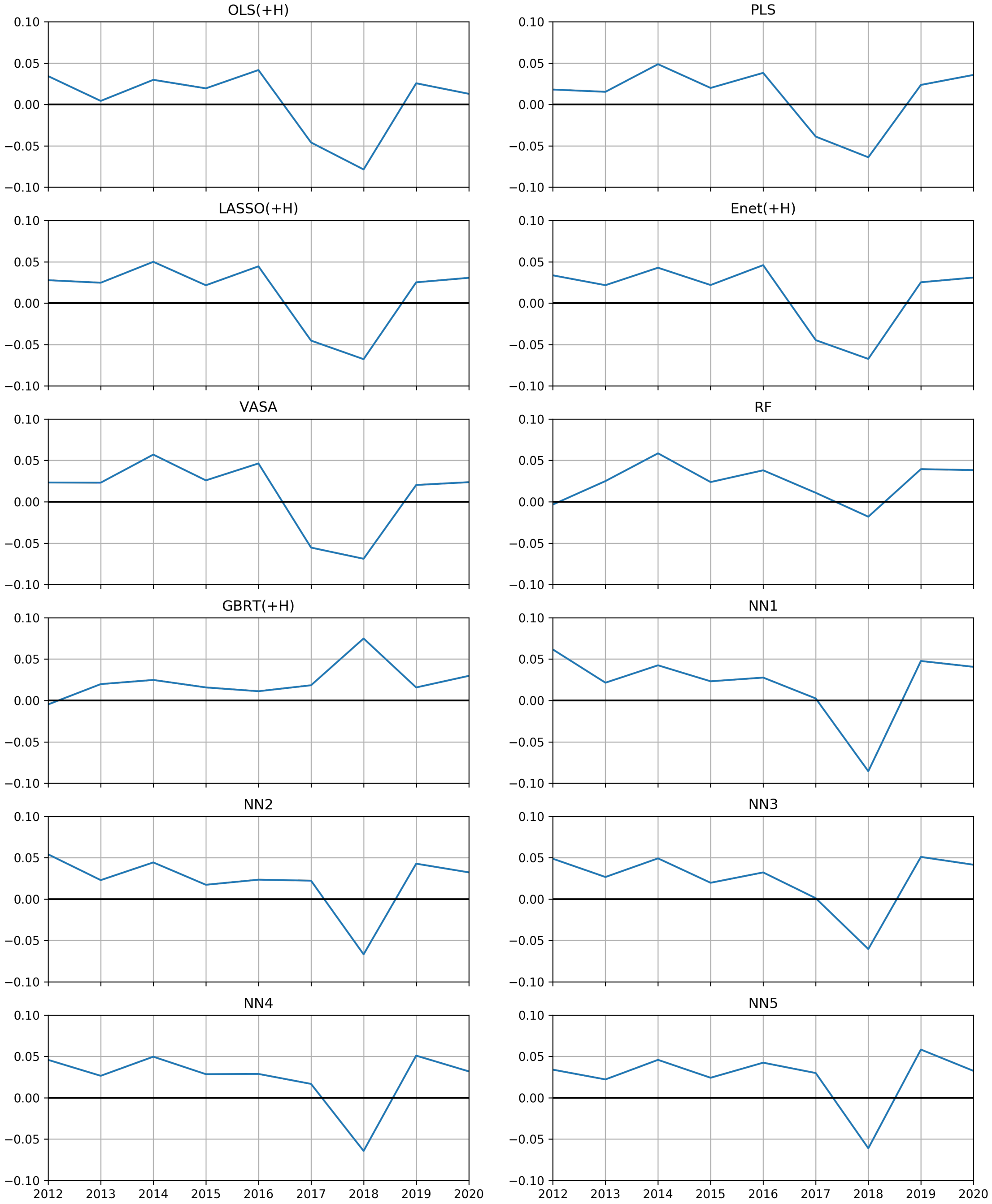


Figure D.1. Annual out-of-sample predictive R^2.

This figure shows annual out-of-sample predictive R^2 for each model during the period of 2012-2020.

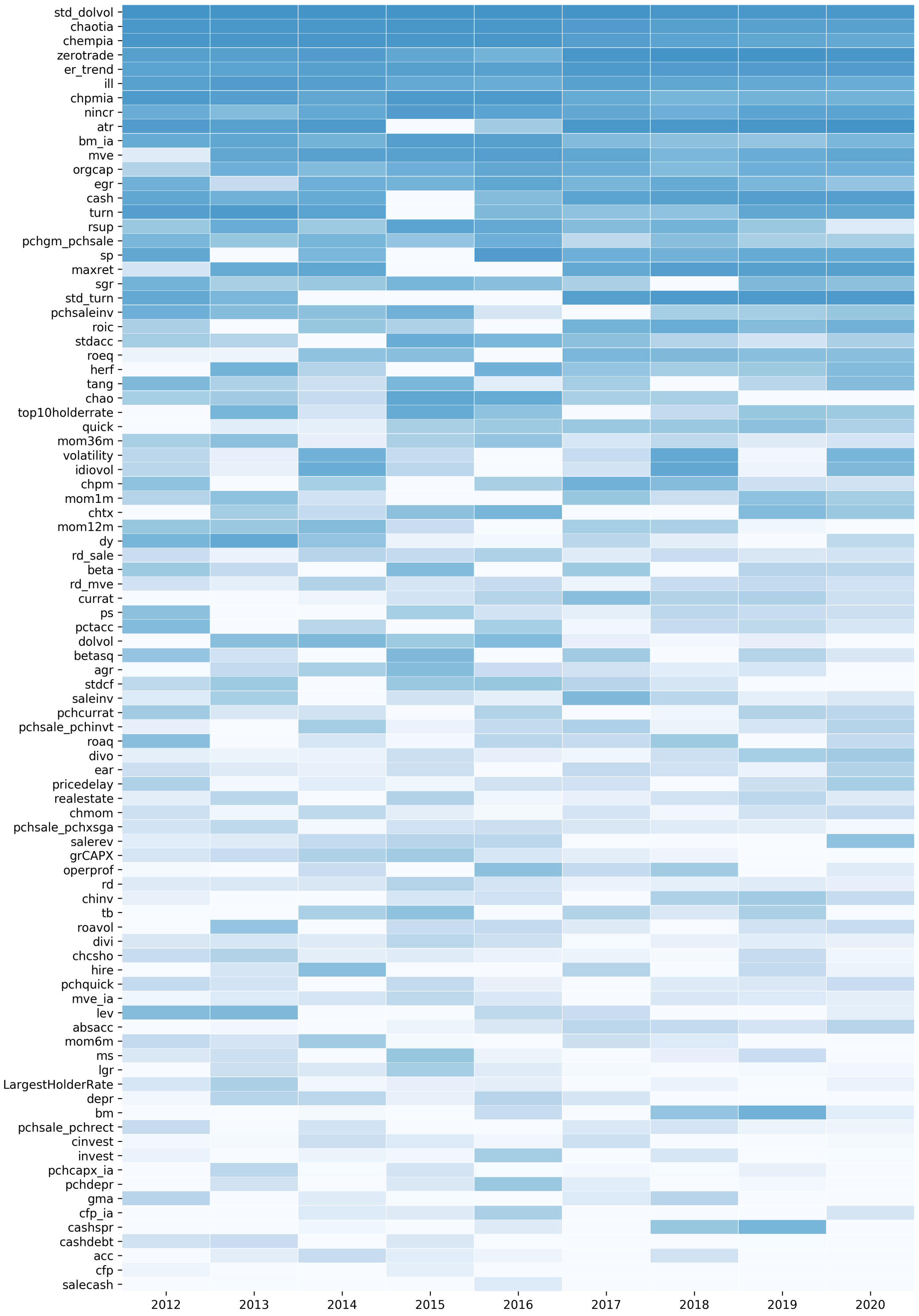


Figure D.2. Characteristic importance for NN4.

This figure shows the ordering of all stock-level characteristics ranked by NN4 across the different years. The vertical axis gives the orderings of the NN4-specific R^2-based variable importance, which is defined similarly as in Fig. 2 in the main text. The horizontal axis indicates the periods in the testing sample, and the color gradient in each column reflects the explanatory power of predictors in a given evaluation period.

E Equal-weighted portfolio analysis

Table E.6

Performance of machine learning portfolios (equal-weighted). This table reports the out-of-sample performance measures for all machine learning models of the equally-weighted long-short and long-only based on the full sample. All measures are based on 103 monthly out-of-sample returns from January 2012 to June 2020. "Avg": average predicted monthly return (%). "Std": the standard deviation of monthly predicted monthly returns (%). "S.R.": Sharpe ratio. "Skew": skewness. "Kurt": kurtosis. "Max DD": the portfolio maximum drawdowns (%). "Max 1M Loss": the most extreme negative monthly return (%).

		Machine Learning Portfolios											
	"$1/N$" Portfolio	OLS-3 +H	PLS	LASSO +H	Enet +H	GBRT +H	RF	VASA	NN1	NN2	NN3	NN4	NN5
Long-Short													
Avg	–	1.50	4.06	4.80	4.88	4.06	3.27	5.14	5.46	5.45	5.80	5.95	5.93
Std	–	4.49	4.69	5.17	5.22	5.02	4.12	5.29	4.72	4.51	5.01	5.02	4.95
S.R.	–	1.15	3.00	3.21	3.24	2.80	2.75	3.36	4.01	4.18	4.00	4.10	4.15
Skew	–	0.16	−0.44	1.29	0.76	1.11	−0.20	1.01	2.14	1.08	1.95	2.66	1.88
Kurt	–	0.37	1.61	6.80	6.27	0.93	0.65	6.22	9.65	3.29	8.81	13.87	7.93
Max DD	–	34.04	14.12	9.07	16.58	13.67	9.86	10.60	4.12	3.86	4.64	4.51	4.49
Max 1M Loss	–	11.40	14.12	9.07	15.26	8.83	9.69	10.60	3.25	3.86	4.64	3.45	4.49
Long-Only													
Avg	1.56	2.24	3.67	4.05	4.20	3.83	3.48	4.38	4.50	4.45	4.74	4.91	4.85
Std	8.44	9.16	7.75	9.08	9.22	8.40	8.50	9.14	9.55	8.73	9.29	9.57	9.71
S.R.	0.64	0.85	1.64	1.54	1.58	1.58	1.42	1.66	1.63	1.77	1.79	1.78	1.73
Skew	0.26	0.57	0.26	1.09	1.05	0.56	0.54	0.99	1.20	0.89	1.20	1.36	1.31
Kurt	1.26	1.38	1.50	4.86	4.50	3.09	1.05	4.38	5.24	3.97	5.41	5.49	5.43
Max DD	54.20	45.35	23.55	23.45	23.03	27.84	27.11	23.12	22.62	21.79	23.23	20.80	21.73
Max 1M Loss	25.56	22.89	20.82	21.85	22.42	23.39	20.25	21.99	22.62	21.26	22.33	19.45	20.11

E.1 Ex ante selection of machine learning methods

A practical problem of real-world investment is to select the best prediction method *ex ante*, which is especially relevant when the number of candidate models is large. We consider two model selection procedures and report their performance for the testing sample. The first one is via simple model averaging. We first predict monthly returns for all stocks using 11 machine learning methods (PLS-NN5) and then construct the long-short and long-only portfolios based on the average predicted stock returns. We report the out-of-sample performance for the model-averaging portfolios in Table E.7, which shows that this procedure works well in the Chinese market. For example, the model-averaging long-only portfolio achieves a Sharpe ratio of 1.76, which is in line with the highest Sharp ratio for a single method shown in Table 6 in the main text.

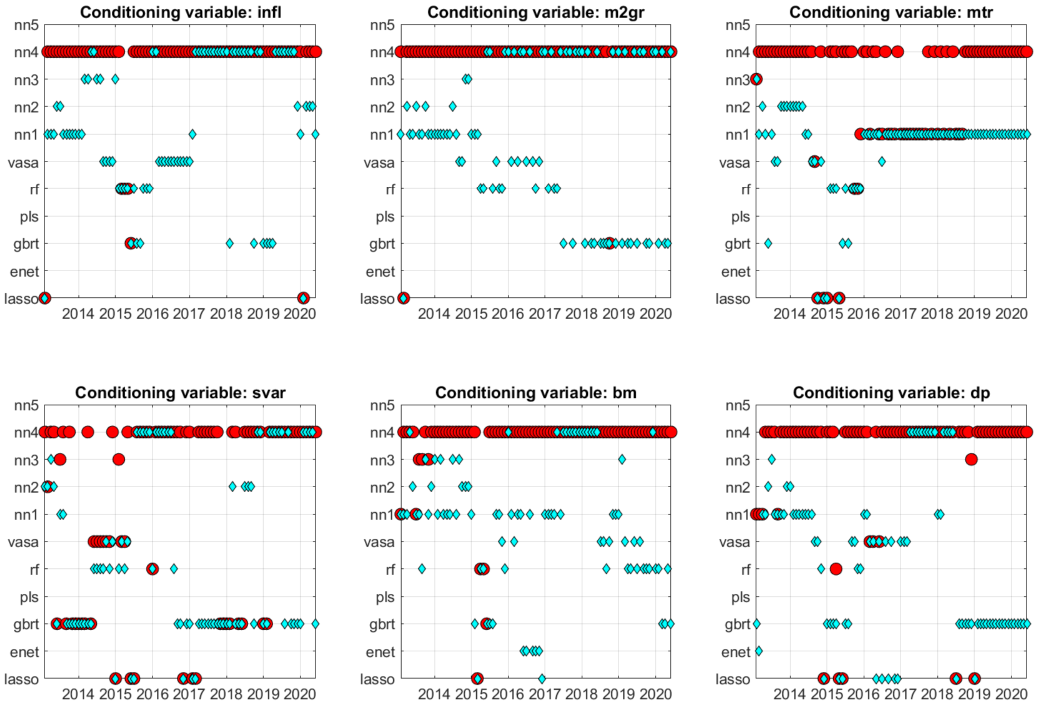
Table E.7

Performance of the model-averaging portfolio (value-weighted). This table reports the out-of-sample performance measures for the model-averaging portfolio based on the full sample. All measures are based on 103 monthly out-of-sample returns from January 2012 to June 2020. "Avg": average predicted monthly return (%). "Std": the standard deviation of monthly predicted monthly returns (%). "S.R.": annualized Sharpe ratio. "Skew": skewness. "Kurt": kurtosis. "Max DD": the portfolio maximum drawdowns (%). "Max 1M Loss": the most extreme negative monthly return (%).

Portfolio	Avg	Std	S.R.	Skew	Kurt	Max DD	Max 1M
Long-short	5.10	5.76	3.06	1.02	4.24	8.51	8.51
Long-only	4.24	8.36	1.76	1.31	6.08	20.63	19.81

The second procedure utilizes the CSPA test to select the method that performs the best conditional on the current macroeconomic condition. Given a pre-specified conditioning variable and the benchmark model, we estimate the conditional expected loss differential functions and the corresponding confidence bounds using the out-of-sample predictions in the testing sample on an annual basis. These plots then form a decision rule for selecting the forecast method month by month. More precisely, given the current macroeconomic condition X_t, the benchmark model is selected if the estimated loss differential to all other models is positive. If a loss differential is significantly negative for a set of alternative models, we chose the model with the most negative loss differential.

In Fig. E.3, we plot the selected models for different conditioning variables across time when


Conditioning variable: infl

Conditioning variable: m2gr

Conditioning variable: mtr

Conditioning variable: svar

Conditioning variable: bm

Conditioning variable: dp

Figure E.3. Dynamic model selection based on CSPA test
This figure shows the monthly model selection based on the CSPA test and the prevailing value of the conditioning variable. The diamond markers represent the model when using directly the change in the loss function, and the circles represent the models when requiring a 10% confidence level. Portfolio formation starts in January 2013 and ends in June 2020.

we choose NN4 as our benchmark model for the full sample. We either use the direct change in the loss function or its 90% confidence bound as a decision rule for model selection. At the end of each month, we check whether the expected loss differential or its 90% confidence bound is below zero at the conditioning variable's actual value. If this is the case, we switch to the model with the largest negative loss differential and use that model for investing in the following month. The resulting model choices are plotted in Fig. E.3 as blue diamonds (when using the expected loss differential) and red circles (when using the 90% confidence bound). Clearly, using the confidence bound for the decision rule provides some additional robustness. Under this rule, there are only a few deviations from the benchmark model. Interestingly, these deviations happen for most conditioning variables during 2015, when the Chinese market is in a crisis. During that time, the CSPA-based model selection particularly favors Lasso and, to some extent, GBRT. When using the expected loss differential, the model selection varies much more, but with some preferred candidates like NN1, GBRT, and VASA.

To explore whether the aforementioned ex-ante model selection rule also provides satisfying performance, we analyze the long-short and the long-only strategy for the full sample. Table E.8 reports the results. For the long-short strategy, we find that using the CSPA-based selection criterion is highly beneficial compared to the model-averaging approach in Table E.7. The long-short portfolio performs best when we take the confidence bound as a decision criterion across the six conditioning variables examined; the average Sharpe ratio is 4.34, compared to a Sharpe ratio of 3.06 in Table 10 in the main text. Even if we take the expected loss differential, we still get an average Sharpe ratio of 3.9. The portfolios perform better than average when we condition on $infl$, bm, and dp.

Table E.8

Performance of CSPA-based model selection portfolios (value-weighted). This table reports the out-of-sample performance measures for the CSPA-based portfolio based on the full sample. All measures are based on 91 monthly out-of-sample returns from January 2013 to June 2020. "Avg": average predicted monthly return (%). "Std": the standard deviation of monthly predicted monthly returns (%). "S.R.": annualized Sharpe ratio. "Skew": skewness. "Kurt": kurtosis. "Max DD": the portfolio maximum drawdowns (%). "Max 1M Loss": the most extreme negative monthly return (%).

	infl	m2gr	mtr	svar	bm	dp	infl	m2gr	mtr	svar	bm	dp
	Long-short						Long-short, 10%					
Avg	4.54	4.88	4.80	5.17	5.08	4.98	5.67	6.10	5.96	5.39	5.64	5.74
Std	4.57	4.34	4.11	5.06	3.83	4.43	3.83	5.36	5.21	5.56	3.81	4.38
S.R.	3.44	3.89	4.05	3.54	4.59	3.89	5.13	3.94	3.96	3.35	5.12	4.53
Skew	-0.20	0.42	0.11	0.02	0.56	-0.05	0.49	2.36	2.66	0.88	0.53	0.12
Kurt	3.57	3.19	4.48	3.29	3.08	3.93	3.37	14.07	16.00	7.86	3.45	4.20
Max DD	18.62	4.46	8.62	7.94	3.79	8.67	4.43	4.43	4.43	8.62	4.43	8.67
Max 1M	9.01	3.47	9.01	7.98	3.86	9.07	3.47	3.47	3.47	9.01	3.47	9.07
	Long-only						Long-only, 10%					
Avg	4.13	4.26	4.28	4.46	4.27	4.40	4.67	5.07	4.87	4.64	4.63	4.74
Std	8.59	8.62	8.67	8.46	8.99	8.94	8.98	9.81	9.77	9.30	8.94	8.73
S.R.	1.67	1.71	1.71	1.82	1.65	1.70	1.80	1.79	1.73	1.73	1.79	1.88
Skew	0.35	0.61	0.41	0.52	0.42	0.68	0.43	1.40	1.36	1.18	0.41	0.44
Kurt	4.19	4.25	4.30	4.39	3.97	5.11	4.00	8.29	8.72	7.27	3.96	4.12
Max DD	25.49	19.55	24.05	20.05	25.74	20.30	23.91	18.81	20.92	20.05	23.91	18.73
Max 1M	20.25	19.46	20.25	19.46	20.25	20.25	19.46	19.46	22.61	19.46	19.46	19.46

A similar conclusion holds when we investigate the performance of the long-only strategies. However, it turns out that the superiority of the CSPA-based selection versus the model-averaging approach vanishes. When we select models based on the 90% confidence bound, the average Sharpe ratio is 1.79, only slightly above the Sharpe ratio of 1.76 for the model-averaging approach. Using the expected loss differential even leads to a slightly underperforming average Sharpe ratio of 1.71. Hence, for the long-only portfolio, the two ex-ante selection strategies lead to a similar performance. Nevertheless, the long-only portfolios, $infl$, bm, and dp, perform well if we use the confidence bound as the decision criterion. Therefore, it would be interesting to explore the role of the conditioning variable further. We leave this issue as a potential avenue for future research.

References

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

Baker, M. and Stein, J. C.: 2004, Market liquidity as a sentiment indicator, *Journal of Financial Markets* **7**(3), 271–299.

Breiman, L.: 2001, Random forests, *Machine learning* **45**(1), 5–32.

Bühlmann, P. and Van De Geer, S.: 2011, *Statistics for high-dimensional data: methods, theory and applications*, Springer Science & Business Media.

Chen, S.-S.: 2009, Predicting the bear stock market: Macroeconomic variables as leading indicators, *Journal of Banking & Finance* **33**(2), 211–223.

Chernozhukov, V., Lee, S. and Rosen, A. M.: 2013, Intersection bounds: estimation and inference, *Econometrica* **81**(2), 667–737.

De Nard, G., Hediger, S. and Leippold, M.: 2020, Subsampled factor models for asset pricing: The rise of vasa, *Available at SSRN 3557957* .

Eisfeldt, A. L. and Papanikolaou, D.: 2013, Organization capital and the cross-section of expected returns, *The Journal of Finance* **68**(4), 1365–1406.

Green, J., Hand, J. and Zhang, F.: 2017, The characteristics that provide independent information about average US monthly stock returns, *The Review of Financial Studies* **30**, 4389–4436.

Gu, S., Kelly, B. and Xiu, D.: 2020, Empirical asset pricing via machine learning, *The Review of Financial Studies* **33**(5), 2223–2273.

Hansen, P. R.: 2005, A test for superior predictive ability, *Journal of Business & Economics Statistics* **23**, 365–380.

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

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

James, G., Witten, D., Hastie, T. and Tibshirani, R.: 2013, *An introduction to statistical learning*, Vol. 112, Springer.

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

Li, J. and Liao, Z.: 2020, Uniform nonparametric inference for time series, *Journal of Econometrics* **219**(1), 38–51.

Li, J., Liao, Z. and Gao, M.: 2020, Uniform nonparametric inference for time series using stata, *Working paper* .

Li, J., Liao, Z. and Quaedvlieg, R.: 2020, Conditional superior predictive ability, *Review of Economic Studies* . Forthcoming.

Liu, Y., Zhou, G. and Zhu, Y.: 2020, Trend factor in China, *Available at SSRN 3402038* .

Medeiros, M. C. and Mendes, E. F.: 2016, $\ell 1$-regularization of high-dimensional time-series models with non-gaussian and heteroskedastic errors, *Journal of Econometrics* **191**(1), 255–271.

Mohanram, P. S.: 2005, Separating winners from losers among lowbook-to-market stocks using financial statement analysis, *Review of accounting studies* **10**(2-3), 133–170.

Ou, J. A. and Penman, S. H.: 1989, Financial statement analysis and the prediction of stock returns, *Journal of accounting and economics* **11**(4), 295–329.

Pan, L., Tang, Y. and Xu, J.: 2016, Speculative trading and stock returns, *Review of Finance* **20**(5), 1835–1865.

Piotroski, J. D.: 2000, Value investing: The use of historical financial statement information to separate winners from losers, *Journal of Accounting Research* pp. 1–41.

Qu, X. and Zhang, G.: 2010, Measuring the convergence of national accounting standards with international financial reporting standards: The application of fuzzy clustering analysis, *The International Journal of Accounting* **45**(3), 334–355.

Rapach, D. E., Strauss, J. K. and Zhou, G.: 2013, International stock return predictability: what is the role of the united states?, *The Journal of Finance* **68**(4), 1633–1662.

Sloan, R.: 1996, Do stock prices fully reflect information in accruals and cash flows about future earnings? (Digest summary), *Accounting Review* **71**(3), 289–315.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: 2014, Dropout: a simple way to prevent neural networks from overfitting, *The journal of machine learning research* **15**(1), 1929–1958.

Welch, I.: 2008, The link between Fama-French time-series tests and Fama-Macbeth cross-sectional tests, *Technical report*, UCLA Anderson School of Management. Available at `https://ssrn.com/abstract=1271935`.

White, H. L.: 2000, A reality check for data snooping, *Econometrica* **68**(5), 1097–1126.

Zou, H. and Hastie, T.: 2005, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**(2), 301–320.