Predicting the Equity Risk Premium using Machine Learning Techniques

S. Yanki Kalfa
University of California San Diego
skalfa@ucsd.edu

Allan Timmermann
University of California San Diego
atimmermann@ucsd.edu

Terri van der Zwan*
Erasmus University Rotterdam; Tinbergen Institute
t.vanderzwan@ese.eur.nl

Introduction

- Machine learning (ML) offers more flexibility than traditional regression, which primarily focuses on variable selection.
- ML models have potential to fit noisy data; risk of overfitting.
- Little guidance on how to tune ML models.

How well do out-of-sample (OoS) or recursive forecast evaluation methods guard us against the risk of overfitting OoS?

General Framework

Equity Risk Premium

Let $r_{i,t}$ be the excess return of asset i at time t, then

$$r_{i,t} = \underbrace{\mathbb{E}[\, r_{i,t} \mid \mathcal{I}_{t-1}]}_{\text{predictable}} + \underbrace{\varepsilon_{i,t}}_{\text{unpredictable}}. \tag{1}$$

Our **objective** is to model the predictable part with $g(\cdot)$:

$$\mathbb{E}[\, r_{i,t} \mid \mathcal{I}_{t-1}] = g(X_{i,t-1}; \theta), \tag{2}$$

a function of K predictor variables $X_{i,t-1}$ and parameters θ.

Data

- Monthly asset returns (CRSP).
- Firm characteristics $X_{i,t}$ (Gu et al., 2020), filled using B-XS model (Bryzgalova et al., 2022). Cross-sectionally scaled between $[-1, 1]$ + industry dummies.
- Features: $T \times N_t$ = 800,000+ observations, $K = 140$.
  - Training set $\mathcal{T}_1$: Jan 1977 – Dec 1996.
  - Test set $\mathcal{T}_2$: Jan 1997 – Dec 2021.

Estimation Procedure

1. Estimate model parameters θ on $\mathcal{T}_1$ minimizing the l_2 norm:

$$\mathcal{L}(\theta) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(r_{i,t} - g(X_{i,t-1}; \theta) \right)^2. \tag{3}$$

2. Predict using $\hat{\theta}$ on $\mathcal{T}_2$.
3. Update $\mathcal{T}_1$ with 12 months, go to step 1.
4. Evaluate performance using Out-of-Sample R^2 against zero prediction:

$$R^2_{OoS} = 1 - \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_2} \left(r_{i,t} - \hat{r}^{(m)}_{i,t} \right)^2 \Big/ \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_2} r^2_{i,t}. \tag{4}$$

If $R^2_{OoS} > 0$, **model outperforms** zero prediction (%).

Models & Results

Linear Models

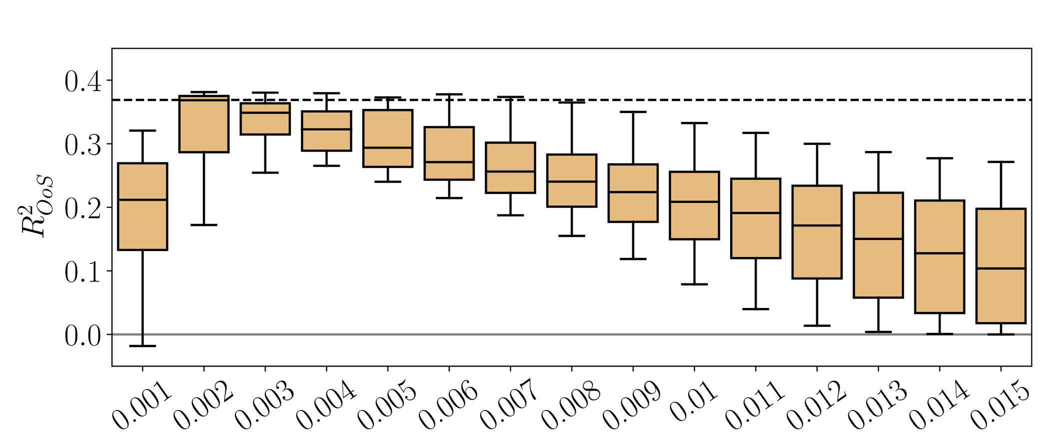
Functional form: $g(X_{i,t-1}; \beta) = \beta_0 + \beta' X_{i,t-1}$,
with **Elastic Net** penalty (**Lasso**: $\lambda = 1$):

$$\mathcal{L}^{EN}(\beta; \alpha, \lambda) = \mathcal{L}(\theta) + \alpha\lambda \sum_{k=0}^{K} |\beta_k| + \frac{\alpha(1-\lambda)}{2} \sum_{k=0}^{K} \beta_k^2. \tag{5}$$

Hyper parameters:
- l_1 shrinkage on coefficients: $\alpha \in \{0.001, 0.002, ..., 0.015\}$
- (l_1, l_2) penalty mix: $\lambda \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$

Figure: Sensitivity R^2_{OoS} to α in Elastic Net

- Most gain for varying α
- $\alpha^* \approx 0.003$
- $\alpha > 0.01$: $\hat{r}_{i,t} = 0$
- Lasso: similar outcome
- Validation (dashed) prevents overfitting



Ensemble Models

Functional form: $g(X_{i,t-1}; \theta, L, D) = \sum_{l=1}^{L} \vartheta_l 1_{X_{i,t-1} \in C_l(D)}$, with loss:

$$\mathcal{L}^B(\theta, C) = \frac{1}{V} \sum_{X_{i,t-1} \in C} \left(r_{i,t} - \frac{1}{V} \sum_{X_{i,t-1} \in C} r_{i,t} \right), \tag{6}$$

where $C_l(D)$ is the l-th of the L data partitions, and ϑ_l the corresponding sample average.

Random Forest (RF): bagging procedure

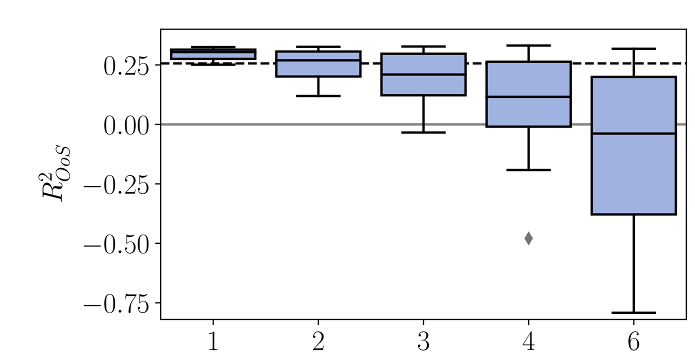
Hyper parameters:
- No. of trees: $B \in \{30, 50, 150, 300, 500\}$
- Max. tree depth: $D \in \{1, 2, 3, 4, 6\}$
- No. of features each split: $V \in \{1, 3, 10, 30, 50\}$

Extreme Gradient Boosting (XGB): boosting procedure

Hyper parameters:
- No. of trees: $B \in \{500, 1000, 1500\}$
- Learning rate: $\eta \in \{0.01, 0.1, 0.2, 0.3\}$
- Max. tree depth: $D \in \{1, 2\}$

Figure: Sensitivity R^2_{OoS} to D in Random Forests

- Ensemble methods: downward risk
- RF: shallow forests best (low D and V)
- XGB: sensitive to hyper parameters
- XGB: $\eta^* = 0.01$ best
- Validation beneficial for both models



Feed-Forward Neural Networks

Functional form: $g(X_{i,t-1}; \theta) = \widetilde{x}^{(H)\prime} \omega_{H+1}$,
with hidden layer $\widetilde{x}^{(\ell)} = f\left(\widetilde{x}^{(\ell-1)\prime} \omega^{(\ell)}\right)$, and weights $\omega^{(\ell)}$.

Architecture:
- Hidden layers, $H \in \{1, 2, 3, 4, 5\}$, with 32, 16, 8, 4, and 2 neurons
- Activation function, $f(\cdot) \in \{\text{linear, ReLu}\}$

Hyper parameters:
- Adam learning rate: $\eta \in \{10^{-4}, 10^{-3}, 10^{-2}\}$
- l_1 shrinkage penalty: $\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}\}$

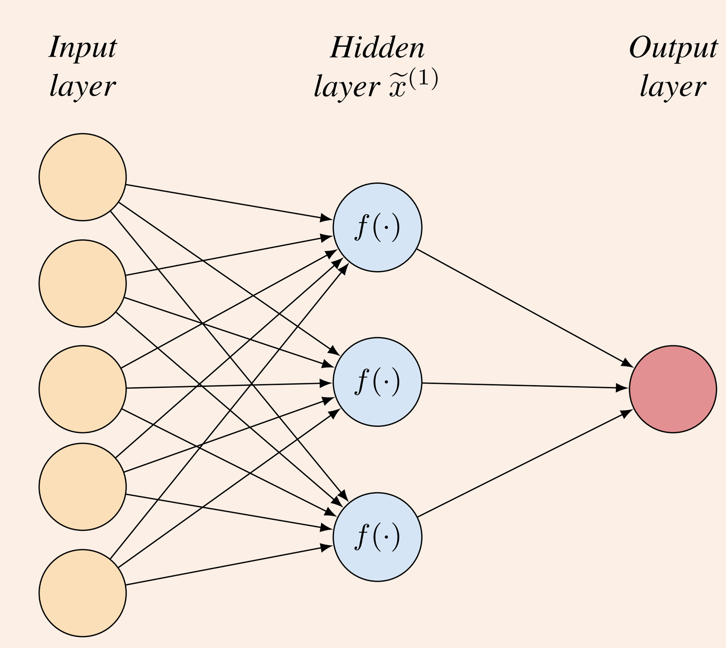
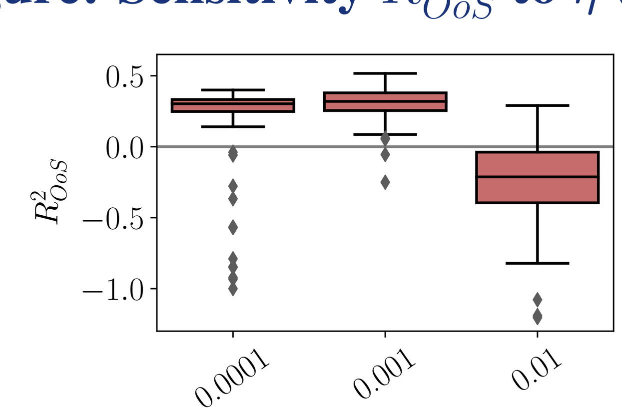
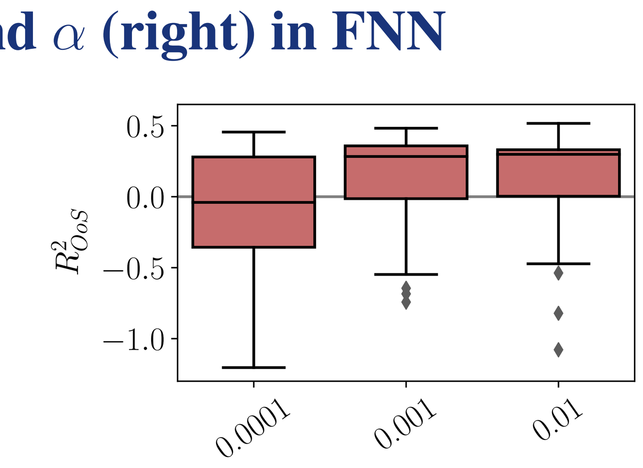


Figure: Sensitivity R^2_{OoS} to η (left) and α (right) in FNN





Architecture: not too much effect
* $H^* = 3, 4$, but minimal impact
* ReLu activation preferred

Hyper parameters: most gain
* Adam learning rate $\eta^* = 0.001$
* α^* around 0.001, 0.01

Summary & Further Research

- **Hyper parameter grid** crucial impact on OoS performance.
- Ensembles and neural nets provide flexibility but risk poor OoS performance.
- **Safest choice**: linear model with l_1 penalty; $\alpha < 0.01$.
- **Validation** seems to help guard against risk of overfitting.
- Further research:
  * Explore: LSTM, other models.
  * Improve: validation methods, grid selection.
  * Assess: (economic) significance.

References

Bryzgalova, S., S. Lerner, M. Lettau, and M. Pelger (2022). Missing financial data. *Working paper.*

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