How is Machine Learning Useful for Macroeconomic Forecasting?*

Philippe Goulet Coulombe[1†] Maxime Leroux[2] Dalibor Stevanovic[2‡]
Stéphane Surprenant[2]

[1]University of Pennsylvania
[2]Université du Québec à Montréal

This version: May 27, 2019

Abstract

We move beyond *Is Machine Learning Useful for Macroeconomic Forecasting?* by adding the *how*. The current forecasting literature has focused on matching specific variables and horizons with a particularly successful algorithm. To the contrary, we study a wide range of horizons and variables and learn about the usefulness of the underlying features driving ML gains over standard macroeconometric methods. We distinguish 4 so-called features (nonlinearities, regularization, cross-validation and alternative loss function) and study their behavior in both the data-rich and data-poor environments. To do so, we carefully design a series of experiments that easily allow to identify the "treatment" effects of interest. The fixed-effects regression setup prompt us to use a novel visualization technique for forecasting results that conveys all the relevant information in a digestible format. We conclude that **(i)** more data and non-linearities are true game-changers for macroeconomic prediction, **(ii)** the standard factor model remains the best regularization, **(iii)** cross-validations are not all made equal (but K-fold is as good as BIC) and **(iv)** one should stick with the standard L_2 loss.

Keywords: Machine Learning, Big Data, Forecasting.

*The third author acknowledges financial support from the Fonds de recherche sur la société et la culture (Québec) and the Social Sciences and Humanities Research Council.

†Corresponding Author: gouletc@sas.upenn.edu. Department of Economics, UPenn.

‡Corresponding Author: dstevanovic.econ@gmail.com. Département des sciences économiques, UQAM.

1 Introduction

The intersection of Machine Learning (ML) with econometrics has become an important research landscape in economics. ML has gained prominence due to the availability of large data sets, especially in microeconomic applications, Athey (2018). However, as pointed by Mullainathan and Spiess (2017), applying ML to economics requires finding relevant tasks. Despite the growing interest in ML, little progress has been made in understanding the properties of ML models and procedures when they are applied to predict macroeconomic outcomes.[1] Nevertheless, that very understanding is an interesting econometric research endeavor *per se*. It is more appealing to applied econometricians to upgrade a standard framework with a subset of specific insights rather than to drop everything altogether for an off-the-shelf ML model.

A growing number studies have applied recent machine learning models in macroeconomic forecasting.[2] However, those studies share some shortcomings. Some focus on one particular ML model and on a limited subset of forecasting horizons. Other evaluate the performance for only one or two dependent variables and for a limited time span. The papers on comparison of ML methods are not very extensive and do only a forecasting horse race without providing insights on why some models perform better.[3] As a result, little progress has been made to understand the properties of ML methods when applied to macroeconomic forecasting. That is, so to say, the black box remains closed. The objective of this paper is to bring an understanding of each method properties that goes beyond the coronation of a single winner for a specific forecasting target. We believe this will be much more useful for subsequent model building in macroeconometrics.

Precisely, we aim to answer the following question. What are the key features of ML modeling that improve the macroeconomic prediction? In particular, no clear attempt has been made at understanding why one algorithm might work and another one not. We address this question by designing an *experiment* to identify important characteristics of ma-

[1] Only the unsupervised statistical learning techniques such as principal component and factor analysis have been extensively used and examined since the pioneer work of Stock and Watson (2002a). Kotchoni et al. (2017) do a substantial comparison of more than 30 various forecasting models, including those based on factor analysis, regularized regressions and model averaging. Giannone et al. (2017) study the relevance of sparse modelling (Lasso regression) in various economic prediction problems.

[2] Nakamura (2005) is an early attempt to apply neural networks to improve on prediction of inflation, while Smalter and Cook (2017) use deep learning to forecast the unemployment. Diebold and Shin (2018) propose a Lasso-based forecasts combination technique. Sermpinis et al. (2014) use support vector regressions to forecast inflation and unemployment. Döpke et al. (2015) and Ng (2014) aim to predict recessions with random forests and boosting techniques. Medeiros et al. (2019) improve inflation prediction using random forests. Few papers contribute by comparing some of the ML techniques in forecasting horse races, see Ahmed et al. (2010), Li and Chen (2014), Ulke et al. (2016), Kim and Swanson (2018) and Chen et al. (2019).

[3] Few exceptions are Stock and Watson (2012) and Smeekes and Wijler (2018) who compare performance of generalized shrinkage methods for orthonormal predictors against the dynamic factor model and of sparse and dense models in presence of non-stationary data respectively.

chine learning and big data techniques. The exercise consists of an extensive pseudo-out-of-sample forecasting horse race between many models that differ with respect to the four main features: nonlinearity, regularization, hyperparameter selection and loss function. To control for big data aspect, we consider data-poor and data-rich models, and administer those *patients* one particular ML *treatment* or combinations of them. Monthly forecast errors are constructed for five important macroeconomic variables, five forecasting horizons and for almost 40 years. Then, we provide a straightforward framework to back out which of them are actual game-changers for macroeconomic forecasting.

The main results can be summarized as follows. First, non-linearities either improve substantially the forecasting accuracy. The benefits are significant for industrial production, unemployment rate, term spread, inflation and housing starts and increase with horizons, especially if combined with factor models. Second, in big data framework, alternative regularization methods (Lasso, Ridge, Elastic-net) do not improve over the factor model, suggesting that the factor representation of the macroeconomy is quite accurate as a mean of dimensionality reduction.

Third, the hyperparameter selection by K-fold cross-validation (CV) and the standard BIC (when possible) do better on average that any other criterion. This suggests that ignoring information criteria when opting for more complicated ML models is not harmful. This is also quite convenient: K-fold is the built-in CV option in most standard ML packages. Fourth, replacing the standard in-sample quadratic loss function by the $\bar{\epsilon}$-insensitive loss function in Support Vector Regressions is not useful, except in very rare cases. Fifth, the marginal effects of big data are positive and significant, and improve with horizons.

The state of economy is another important ingredient as it interacts with few features above. Improvements over standard autoregressions are usually magnified if the target falls into an NBER recession period, and the access to data-rich predictor set is particularly helpful. Moreover, the pseudo-out-of-sample cross-validation failure is mainly attributable to its underperformance during recessions.

These results give a clear recommendation for practitioners. For most variables and horizons, start by reducing the dimensionality with principal components and then augment the standard diffusion indices model by a ML non-linear function approximator of choice. Of course, that recommendation is conditional on being able to keep overfitting in check. To that end, if cross-validation must be applied to hyperparameter selection, the best practice is the standard K-fold.

In the remainder of this papers we first present the general prediction problem with machine learning and big data. The Section 3 describes the four important features of machine learning methods. The Section 4 presents the empirical setup, the Section 5 discuss the main results and Section 6 concludes. Appendices A, B, C, D, E and F contain, respectively: tables with overall performance; robustness of treatment analysis; additional figures; results for

absolute loss; description of CV techniques and technical details on forecasting models.

2 Making predictions with machine learning and big data

To fix ideas, consider the following general prediction setup from Hastie et al. (2017)

$$\min_{g \in \mathcal{G}} \{\hat{L}(y_{t+h}, g(Z_t)) + pen(g; \tau)\}, \quad t = 1, \ldots, T \tag{1}$$

where y_{t+h} is the variable to be predicted h periods ahead (target) and Z_t is the N_Z-dimensional vector of predictors made of H_t, the set of all the inputs available at time t. Note that the time subscripts are not necessary so this formulation can represent any prediction problem. This setup has four main features:

1. $\mathcal{G}$ is the space of possible functions g that combine the data to form the prediction. In particular, the interest is how much non-linearities can we allow for? A function g can be parametric or nonparametric.

2. $pen()$ is the penalty on the function g. This is quite general and can accommodate, among others, the Ridge penalty of the standard by-block lag length selection by information criteria.

3. τ is the set of hyperparameters of the penalty above. This could be λ in a LASSO regression or the number of lags to be included in an AR model.

4. $\hat{L}$ the loss function that defines the optimal forecast. Some models, like the SVR, feature an in-sample loss function different from the standard l_2 norm.

Most of (Supervised) machine learning consists of a combination of those ingredients. This formulation may appear too abstract, but the simple predictive regression model can be obtained as a special case. Suppose a quadratic loss function $\hat{L}$, implying that the optimal forecast is the conditional expectation $\mathrm{E}(y_{t+h}|Z_t)$. Let the function g be parametric and linear: $y_{t+h} = Z_t\beta + error$. If the number of coefficients in β is not too big, the penalty is usually ignored and (1) reduces to the textbook predictive regression inducing $\mathrm{E}(y_{t+h}|Z_t) = Z_t\beta$ as the optimal prediction.

2.1 Predictive Modeling

We consider the *direct* predictive modeling in which the target is projected on the information set, and the forecast is made directly using the most recent observables. This is opposed

to *iterative* approach where the model recursion is used to simulate the future path of the variable.[4] Also, the direct approach is the standard practice for in ML applications.

We now define the forecast objective. Let Y_t denote a variable of interest. If $\ln Y_t$ is stationary, we will consider forecasting its level h periods ahead:

$$y_{t+h}^{(h)} = y_{t+h}, \tag{2}$$

where $y_t \equiv \ln Y_t$ if Y_t is strictly positive. Most of the time, we are confronted with I(1) series in macroeconomics. For such series, our goal will be to forecast the average growth rate over the period $[t+1, t+h]$, as in Stock and Watson (2002b) and McCracken and Ng (2016). We shall therefore define $y_{t+h}^{(h)}$ as:

$$y_{t+h}^{(h)} = (1/h)\ln(Y_{t+h}/Y_t). \tag{3}$$

In order to avoid a cumbersome notation, we use y_{t+h} instead of $y_{t+h}^{(h)}$ in what follows.

2.2 *Data-poor* versus *data-rich* environments

Large time series panels are now widely constructed and used for macroeconomic analysis. The most popular is FRED-MD monthly panel of US variables constructed by McCracken and Ng (2016). Fortin-Gagnon et al. (2018) have recently proposed similar data for Canada, while Boh et al. (2017) has constructed a large macro panel for Euro zone. Unfortunately, the performance of standard econometric models tends to deteriorate as the dimensionality of the data increases, which is the well-known curse of dimensionality. Stock and Watson (2002a) first proposed to solve the problem by replacing the large-dimensional information set by its principal components. See Kotchoni et al. (2017) for the review of many dimension-reduction, regularization and model averaging predictive techniques. Another way to approach the dimensionality problem is to use Bayesian methods (Kilian and Lütkepohl (2017)). All the shrinkage schemes presented later in this paper can be seen as a specific prior. Indeed, some of our Ridge regressions will look very much like a direct version of a Bayesian VAR with a Litterman (1979) prior.[5]

Traditionally, as all these series may not be relevant for a given forecasting exercise, one will have to preselect the most important candidate predictors according to economic theories, the relevant empirical literature and own heuristic arguments. Even though the machine learning models do not require big data, they are useful to discard irrelevant predictors

[4] Marcellino et al. (2006) conclude that the direct approach provides slightly better results but does not dominate uniformly across time and series.

[5] Giannone et al. (2015) have shown that a more elaborate hierarchical prior can lead the BVAR to perform as well as a factor model

based on statistical learning, but also to digest a large amount of information to improve the prediction. Therefore, in addition to treatment effects in terms of characteristics of forecasting models, we will also compare the predictive performance of small versus large data sets. The data-poor, defined as H_t^-, will only contain a finite number of lagged values of the dependent variable, while the data-rich panel, defined as H_t^+ will also include a large number of exogenous predictors. Formally, we have

$$H_t^- \equiv \{y_{t-j}\}_{j=0}^{p_y} \quad and \quad H_t^+ \equiv \left[\{y_{t-j}\}_{j=0}^{p_y}, \{X_{t-j}\}_{j=0}^{p_f}\right]. \tag{4}$$

The analysis we propose can thus be summarized in the following way. We will consider two standard models for forecasting.

1. The H_t^- model is the *autoregressive direct* (AR) model, which is specified as:

$$y_{t+h} = c + \rho(L)y_t + e_{t+h}, \quad t = 1, \dots, T, \tag{5}$$

   where $h \geq 1$ is the forecasting horizon. The only hyperparameter in this model is p_y, the order of the lag polynomial $\rho(L)$.

2. The H_t^+ workhorse model is the autoregression augmented with diffusion indices (ARDI) from Stock and Watson (2002b):

$$\begin{aligned} y_{t+h} &= c + \rho(L)y_t + \beta(L)F_t + e_{t+h}, \quad t = 1, \dots, T & (6) \\ X_t &= \Lambda F_t + u_t & (7) \end{aligned}$$

   where F_t are K *consecutive static* factors, and $\rho(L)$ and $\beta(L)$ are lag polynomials of orders p_y and p_f respectively. The feasible procedure requires an estimate of F_t that is usually obtained by principal components analysis (PCA).

Then, we will take these models as two different types of "patients" and will administer them one particular ML treatment or combinations of them. That is, we will upgrade (hopefully) these models with one or many features of ML and evaluate the gains/losses in both environments.

Beyond the fact that the ARDI is a very popular macro forecasting model, there are additional good reasons to consider it as one benchmark for our investigation. While we discuss four features of ML in this paper, it is obvious that the big two are shrinkage (or dimension reduction) and non-linearities. Both goes in completely different directions. The first deals with data sets that have a low observations to regressors ratio while the latter is especially useful when that same ratio is high. Most nonlinearities are created with basis expansions which are just artificially generated additional regressors made of the original data. That

is quite useful in a data-poor environments but is impracticable in data-rich environments where the goal is exactly the opposite, that is, to decrease the effective number of regressors.

Hence, the only way to afford non-linear models with wide macro datasets is to compress the data beforehand and then use the compressed predictors as inputs. Each compression scheme has an intuitive economic justification of its own. Choosing only a handful of series can be justified by some DSGE model that has a reduced-form VAR representation. Compressing the data according to a factor model adheres to the view that are only a few key drivers of the macroeconomy and those are not observed. We choose the latter option as its forecasting record is stellar. Hence, our non-linear models implicitly postulate that a sparse set of latent variables impact the target variable in a flexible way. To take PCs of data to feed them afterward in a NL model is also a standard thing to do from a ML perspective.

2.3 Evaluation

The objective of this paper is to disentangle important characteristics of the ML prediction algorithms when forecasting macroeconomic variables. To do so, we design an *experiment* that consists of a pseudo-out-of-sample forecasting horse race between many models that differ with respect to the four main features above: nonlinearity, regularization, hyperparameter selection and loss function. To create variation around those *treatments*, we will generate forecasts errors from different models associated to each feature.

To test this paper's hypothesis, suppose the following model for forecasting errors

$$e^2_{t,h,v,m} = \alpha_m + \psi_{t,v,h} + v_{t,h,v,m} \tag{8a}$$

$$\alpha_m = \alpha_F + \eta_m \tag{8b}$$

where $e^2_{t,h,v,m}$ are squared prediction errors of model m for variable v and horizon h at time t. $\psi_{t,v,h}$ is a fixed effect term that demean the dependent variable by "forecasting target", that is a combination of t, v and h. α_F is a vector of $\alpha_{\mathcal{G}}$, $\alpha_{pen()}$, α_τ and $\alpha_{\hat{L}}$ terms associated to each feature. We re-arrange equation (8) to obtain

$$e^2_{t,h,v,m} = \alpha_F + \psi_{t,v,h} + u_{t,h,v,m}. \tag{9}$$

H_0 is now $\alpha_f = 0 \quad \forall f \in F = [\mathcal{G},\ pen(),\ \tau,\ \hat{L}]$. In other words, the null is that there is no predictive accuracy gain with respect to a base model that does not have this particular feature.[6] Very interestingly, by interacting α_F with other fixed effects or even variables, we can test many hypothesis about the heterogeneity of the "ML treatment effect". Finally, to get

[6]Note that if we are considering two models that differ in one feature and run this regression for a specific (h, v) pair, the t-test on the sole coefficients amounts to a Diebold and Mariano (1995) test – conditional on having the proper standard errors.

interpretable coefficients, we use a linear combination of $e^2_{t,h,v,m}$ by (h,v) pair that makes the final regressand $(h,v,m)-$specific average a pseudo-out-of-sample R^2.[7] Hence, we define $R^2_{t,h,v,m} \equiv 1 - \frac{e^2_{t,h,v,m}}{\frac{1}{T}\sum_{t=1}^{T}(y_{v,t+h}-\bar{y}_{v,h})^2}$ and run

$$R^2_{t,h,v,m} = \dot{\alpha}_F + \dot{\psi}_{t,v,h} + \dot{u}_{t,h,v,m}. \tag{10}$$

On top of providing coefficients $\dot{\alpha}_F$ interpretable as marginal improvements in OOS-R^2's, the approach has the advantage of standardizing *ex-ante* the regressand and thus removing an obvious source of (v,h)-driven heteroscedasticity. Also, a positive α_F now means (more intuitively) an improvement rather than the other way around.

While the generality of (9) and (10) is appealing, when investigating the heterogeneity of specific partial effects, it will be much more convenient to run specific regressions for the multiple hypothesis we wish to test. That is, to evaluate a feature f, we run

$$\forall m \in \mathcal{M}_f: \quad R^2_{t,h,v,m} = \dot{\alpha}_f + \dot{\phi}_{t,v,h} + \dot{u}_{t,h,v,m} \tag{11}$$

where $\mathcal{M}_f$ is defined as the set of models that differs only by the feature under study f.

3 Four features of ML

In this section we detail the forecasting approaches to create variations for each characteristic of machine learning prediction problem defined in (1).

3.1 Feature 1: selecting the function g

Certainly an important feature of machine learning is the whole available apparatus of non-linear function estimators. We choose to focus on applying the Kernel trick and Random Forests to our two baseline models to see if the non-linearities they generate will lead to significant improvements.

3.1.1 Kernel Ridge Regression

Since all models considered in this paper can easily be written in the dual form, we can use the kernel trick (KT) in both data-rich and data-poor environments. It is worth noting that Kernel Ridge Regression (KRR) has several implementation advantages. First, it has a closed-form solution that rules out convergence problems associated with models trained with gradient descent. Second, it is fast to implement given that it implies inverting a TxT

[7] Precisely: $\frac{1}{T}\sum_{t=1}^{T} 1 - \frac{e^2_{t,h,v,m}}{\frac{1}{T}\sum_{t=1}^{T}(y_{v,t+h}-\bar{y}_{v,h})^2} = R^2_{h,v,m}$

matrix at each step (given tuning parameters) and T is never quite large in macro. Since we are doing an extensive POOS exercise for a long period of time, these qualities are very helpful.

We will first review briefly how the KT is implemented in our two benchmark models. Suppose we have a Ridge regression direct forecast with generic regressors Z_t

$$\min_{\beta} \sum_{t=1}^{T} (y_{t+h} - Z_t\beta)^2 + \lambda \sum_{k=1}^{K} \beta_k^2.$$

The solution to that problem is $\hat{\beta} = (Z'Z + \lambda I_k)^{-1} Z'y$. By the representer theorem of Smola and Schölkopf (2004), β can also be obtained by solving the dual of the convex optimization problem above. The dual solution for β is $\hat{\beta} = Z'(ZZ' + \lambda I_T)^{-1} y$. This equivalence allows to rewrite the conditional expectation in the following way:

$$\hat{E}(y_{t+h}|Z_t) = Z_t\hat{\beta} = \sum_{i=1}^{t} \hat{\alpha}_i \langle Z_i, Z_t \rangle$$

where $\hat{\alpha} = (ZZ' + \lambda I_T)^{-1} y$ is the solution to the dual Ridge Regression problem. For now, this is just another way of getting exactly the same fitted values.

Let's now introduce a general non-linear model. Suppose we approximate it with basis functions $\phi()$

$$y_{t+h} = g(Z_t) + \varepsilon_{t+h} = \phi(Z_t)'\gamma + \varepsilon_{t+h}.$$

The so-called Kernel trick is the fact that there exist a reproducing kernel $K()$ such that

$$\hat{E}(y_{t+h}|Z_t) = \sum_{i=1}^{t} \hat{\alpha}_i \langle \phi(Z_i), \phi(Z_t) \rangle = \sum_{i=1}^{t} \hat{\alpha}_i K(Z_i, Z_t).$$

This means we do not need to specify the numerous basis functions, a well-chosen Kernel implicitly replicates them. For the record, this paper will be using the standard radial basis function kernel

$$K_\sigma(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

where σ is a tuning parameter to be chosen by cross-validation.

Hence, by using the corresponding Z_t, we can easily make our data-rich or data-poor model non-linear. For instance, in the case of the factor model, we can apply it to the regres-

sion equation to implicitly estimate

$$y_{t+h} = c + g(Z_t) + \varepsilon_{t+h}, \tag{12}$$
$$Z_t = \left[\{y_{t-0}\}_{j=0}^{p_y}, \{F_{t-j}\}_{j=0}^{p_f}\right], \tag{13}$$
$$X_t = \Lambda F_t + u_t. \tag{14}$$

In terms of implementation, this means extracting factor via PCA and then get

$$\hat{E}(y_{t+h}|Z_t) = K_\sigma(Z_t, Z)(K_\sigma(Z, Z) + \lambda I_T)^{-1} y. \tag{15}$$

The final set of tuning parameters for such a model is $\tau = \{\lambda, \sigma, p_y, p_f, n_f\}$.

3.1.2 Random forests

Another way to introduce non-linearity in the estimation of the predictive equation is to use regression trees instead of OLS. Recall the ARDI model:

$$y_{t+h} = c + \rho(L)y_t + \beta(L)F_t + \varepsilon_{t+h},$$
$$X_t = \Lambda F_t + u_t,$$

where y_t and F_t, and their lags, constitute the informational set Z_t. This form is clearly linear but one could tweak the model by replacing it by a regression tree. The idea is to split sequentially the space of Z_t into several regions and model the response by the mean of y_{t+h} in each region. The process continues according to some stopping rule. As a result, the tree regression forecast has the following form:

$$\hat{f}(Z) = \sum_{m=1}^{M} c_m \mathrm{I}_{(Z \in R_m)}, \tag{16}$$

where M is the number of terminal nodes, c_m are node means and $R_1, ..., R_M$ represent a partition of feature space. In the diffusion indices setup, the regression tree would estimate a non-linear relationship linking factors and their lags to y_{t+h}. Once the tree structure is known, this procedure can be related to a linear regression with dummy variables and their interactions.

Instead of just using one single tree, which is known to be subject to overfitting, we use Random forests which consist of a certain number of trees using a subsample of observations but also a random subset of regressors for each tree.[8] The hyperparameter to be selected

[8]Only using a subsample of observations would be a procedure called Bagging. Also selecting randomly regressors has the effect of decorrelating the trees and hence improving the out-of-sample forecasting accuracy.

is the number of trees. The forecasts of the estimated regression trees are then averaged together to make one single prediction of the targeted variable.

3.2 Feature 2: selecting the regularization

In this section we will only consider models where dimension reduction is needed, which are the models with H_t^+ – that is, more information than just the past values of y_t. The traditional shrinkage method used in macroeconomic forecasting is the ARDI model that consists of extracting principal components of X_t and to use them as data in an ARDL model. Obviously, this is only one out of many ways to compress the information contained in X_t to run a well-behaved regression of y_{t+h} on it. De Mol et al. (2008) compares Lasso, Ridge and ARDI and finds that forecasts are very much alike. This section can be seen as extending the scope of their study by considering a wider range of models in an updated forecasting experiment that includes the Great Recession (theirs end in 2003).

In order to create identifying variations for $pen()$ treatment, we need to generate multiple different shrinkage schemes. Some will also blend in selection, some will not. The alternative shrinkage methods considered in this section will all be specific special cases of a standard Elastic Net (EN) problem:

$$\min_{\beta} \sum_{t=1}^{T} (y_{t+h} - Z_t\beta)^2 + \lambda \sum_{k=1}^{K} \left(\alpha|\beta_k| + (1-\alpha)\beta_k^2 \right) \tag{17}$$

where $Z_t = B(H_t)$ is some transformation of the original predictive set X_t. $\alpha \in [0,1]$ can either be fixed or found via cross-validation (CV) while $\lambda > 0$ always needs to be obtained by CV. By using different B operators, we can generate shrinkage schemes. Also, by setting α to either 1 or 0 we generate LASSO and Ridge Regression respectively. Choosing α by CV also generate an intermediary regularization scheme of its own. All these possibilities are reasonable alternatives to the traditional factor hard-thresholding procedure that is ARDI.

Each type of shrinkage in this section will be defined by the tuple $S = \{\alpha, B()\}$. To begin with the most straightforward dimension, for a given B, we will evaluate the results for $\alpha \in \{0, \hat{\alpha}_{CV}, 1\}$. For instance, if B is the identity mapping, we get in turns the LASSO, Elastic Net and Ridge shrinkage.

Let us now turn to detail different resulting $pen()$ when we vary $B()$ for a fixed α. Three alternatives will be considered.

1. **(Fat Regression)**: First, we will consider the case $B_1() = I()$ as mentioned above. That is, we use the entirety of the untransformed high-dimensional data set. The results of Giannone et al. (2017) point in the direction that specifications with a higher α should do better, that is, sparse models do worse than models where every regressor is kept but shrunk to zero.

2. **(Big ARDI)** Second, we will consider the case where $B_2()$ corresponds to first rotating $X_t \in \mathbb{R}^N$ so that we get N uncorrelated F_t. Note here that contrary to the standard ARDI model, we do not throw out factors according to some information criteria or a scree test: we keep them all. Hence, F_t has exactly the same span as X_t. If we were to run OLS (without any form of shrinkage), using $\phi(L)F_t$ versus $\psi(L)X_t$ would not make any difference in term of fitted values. However, when shrinkage comes in, a similar $pen()$ applied to a rotated regressor space implicitly generates a new penalty. Comparing LASSO and Ridge in this setup will allow to verify whether sparsity emerges in a rotated space. That is, this could be interpreted as looking whether the 'economy' has a sparse DGP, but in a different regressor space that the original one. This correspond to the dense view of the economy, which is that observables are only driven by a few key fundamental economic shocks.

3. **(Principal Component Regression)** A third possibility is to rotate H_t^+ rather than X_t and still keep all the factors. H_t^+ includes all the relevant pre-selected lags. If we were to just drop the F_t using some hard-thresholding rule, this would correspond to Principal Component Regression (PCR). Note that $B_3() = B_2()$ only when no lags are included. Here, the F_t have a different interpretation since they are extracted from multiple t's data whereas the standard factor model used in econometrics typically extract principal components out of X_t in a completely contemporaneous fashion.

To wrap up, this means the tuple S has a total of 9 elements. Since we will be considering both POOS-CV and K-fold CV for each of these models, this leads to a total of 18 models.

Finally, to see clearly through all of this, we can describe where the benchmark ARDI model stands in this setup. Since it uses a hard thresholding rule that is based on the eigenvalues ordering, it cannot be a special case of the Elastic Net problem. While it is clearly using B_2, we would need to set $\lambda = 0$ and select F_t *a priori* with a hard-thresholding rule. The closest approximation in this EN setup would be to set $\alpha = 1$ and fix the value of λ to match the number of consecutive factors selected by an information criteria directly in the predictive regression (19) or using an analytically calculated value based on Bai and Ng (2002). However, this would still not impose the ordering of eigenvalues: the Lasso could happen to select a F_t associated to a small eigenvalue and yet drop one F_t associated with a bigger one.

3.3 Feature 3: Choosing hyperparameters τ

The conventional wisdom in macroeconomic forecasting is to either use AIC or BIC and compare results. It is well known that BIC selects more parsimonious models than AIC. A relatively new kid on the block is cross-validation, which is widely used in the field of

machine learning. The prime reason for the popularity of CV is that it can be applied to any model, which includes those for which the derivation of an information criterion is impossible. Another appeal of the method is its logical simplicity. However, as AIC and BIC, it relies on particular assumptions in order to be well-behaved.

It is not obvious that CV should work better only because it is "out of sample" while AIC and BIC are "in sample". All model selection methods are actually approximations to the OOS prediction error that relies on different assumptions that are sometime motivated by different theoretical goals. Also, it is well known that asymptotically, these methods have similar behavior.[9] Hence, it is impossible *a priori* to think of one model selection technique being the most appropriate for macroeconomic forecasting.

For samples of small to medium size encountered in macro, the question of which one is optimal in the forecasting sense is inevitably an empirical one. For instance, Granger and Jeon (2004) compared AIC and BIC in a generic forecasting exercise. In this paper, we will compare AIC, BIC and two types of CV for our two baseline models. The two types of CV are relatively standard. We will first use POOS CV and then k-fold CV. The first one will always behave correctly in the context of time series data, but may be quite inefficient by only using the end of the training set. The latter is known to be valid only if residuals autocorrelation is absent from the models as shown in Bergmeir et al. (2018). If it were not to be the case, then we should expect k-fold to under-perform. The specific details of the implementation of both CVs is discussed in appendix E.

The contributions of this section are twofold. First, it will shed light on which model selection method is most appropriate for typical macroeconomic data and models. Second, we will explore how much of the gains/losses of using ML can be attributed to widespread use of CV. Since most non-linear ML models cannot be easily tuned by anything else than CV, it is hard for the researcher to disentangle between gains coming from the ML method itself or just the way it is tuned.[10] Hence, it is worth asking the question whether some gains from ML are simply coming from selecting hyperparameters in a different fashion using a method which assumptions are more fit with the data at hand. To investigate that, a natural first step is to look at our benchmark macro models, AR and ARDI, and see if using CV to select hyperparameters gives different selected models and forecasting performances.

[9] Hansen and Timmermann (2015) show equivalence between test statistics for OOS forecasting performance and in-sample Wald statistics. For instance, one can show that Leave-one-out CV (a special case of k-fold) is asymptotically equivalent to Takeuchi Information criterion (TIC), Claeskens and Hjort (2008). AIC is a special case of TIC where we need to assume in addition that all models being considered are at least correctly specified. Thus, under the latter assumption, Leave-one-out CV is asymptotically equivalent to AIC.

[10] Zou et al. (2007) show that the number of remaining parameters in the LASSO is an unbiased estimator of the degrees of freedom and derive LASSO-BIC and LASSO-AIC criteria. Considering these as well would provide additional evidence on the empirical debate of CV vs IC.

3.4 Feature 4: Selecting the loss function

With the exception of the support vector regression (SVR), all of our estimators for the predictive function $g \in \mathcal{G}$ use a quadratic loss function. The objective of this section is to evaluate the importance of a $\bar{\epsilon}$-insensitive loss function for macroeconomic predictions. However, this is not so easily done since the SVR is different from an ARDI model in multiple aspects. Namely, it

- uses a different in-sample loss function;
- (usually) uses a kernel trick in order to obtain non-linearities and
- has different tuning parameters.

Hence, we must provide a strategy to isolate the effect of the first item. That is, if the standard RBF kernel SVR works well, we want to know whether is the effect of the kernel or that of the loss-function. First, while the SVR is almost always used in combination with a kernel trick similar to what described in the previous sections, we will also obtain results for a linear SVR. That isolates the effect of the kernel. Second, we considered the Kernel Ridge Regression earlier. The latter only differs from the Kernel-SVR by the use of different in-sample loss functions. That identifies the effect of the loss function. To sum up, in order to isolate the "treatment effect" of a different in-sample loss function, we will obtain forecasts from

1. the linear SVR with H_t^-;
2. the linear SVR with H_t^+;
3. the RBF Kernel SVR with H_t^- and
4. the RBF Kernel SVR with H_t^+.

What follows is a bird's eye overview of the underlying mechanics of the SVR. As it was the case for the Kernel Ridge regression, the SVR estimator approximates the function $g \in G$ with basis functions. That is, the DGP is still $y_{t+h} = \alpha + \gamma'\phi(Z_t) + \epsilon_{t+h}$. We opted to use the ν-SVR variant which implicitly defines the size $2\bar{\epsilon}$ of the insensitivity tube of the loss function. The hyperparameter ν is selected by cross validation. This estimator is defined by:

$$\min_{\gamma} \frac{1}{2}\gamma'\gamma + C\left[\sum_{j=1}^{T}(\xi_j + \xi_j^*) + T\nu\bar{\epsilon}\right]$$

$$s.t. \begin{cases} y_{t+h} - \gamma'\phi(Z_t) - c \leq \bar{\epsilon} + \xi_t \\ \gamma'\phi(Z_t) + c - y_{t+h} \leq \bar{\epsilon} + \xi_t^* \\ \xi_t, \xi_t^* \geq 0. \end{cases}$$

Where ξ_t, ξ_t^* are slack variables, $\phi()$ is the basis function of the feature space implicitly defined by the kernel used, T is the size of the sample used for estimation and C is an hyperparameter. In case of the RBF Kernel, an additional hyperparameter, σ, has to be cross-validated. Associating Lagrange multipliers λ_j, λ_j^* to the first two types of constraints, we can derive the dual problem (Smola and Schölkopf (2004)) out of which we would find the optimal weights $\gamma = \sum_{j=1}^{T}(\lambda_j - \lambda_j^*)\phi(Z_j)$ and the forecasted values

$$\hat{E}(y_{t+h}|Z_t) = \hat{c} + \sum_{j=1}^{T}(\lambda_j - \lambda_j^*)\phi(Z_j)\phi(Z_j) = \hat{c} + \sum_{j=1}^{T}(\lambda_j - \lambda_j^*)K(Z_j, Z_t). \tag{18}$$

Let us now turn to the resulting loss function of such a problem. Along the in-sample forecasted values, there is an upper bound $\hat{E}(y_{t+h}|Z_t) + \bar{\epsilon}$ and lower bound $\hat{E}(y_{t+h}|Z_t) - \bar{\epsilon}$. Inside of these bounds, the loss function is null. Let $e_{t+h} := \hat{E}(y_{t+h}|Z_t) - y_t$ be the forecasting error and define a loss function using a penalty function $P_{\bar{\epsilon}}$ as $\hat{L}_{\bar{\epsilon}}(\{e_{t+h}\}_{t=1}^{T}) := \frac{1}{T}\sum_{t=1}^{T} P_{\bar{\epsilon}}(e_{t+h})$. For the ν-SVR, the penalty is given by:

$$P_{\bar{\epsilon}}(\epsilon_{t+h|t}) := \begin{cases} 0 & if \quad |e_{t+h}| \leq \bar{\epsilon} \\ |e_{t+h}| - \bar{\epsilon} & otherwise \end{cases}.$$

For other estimators, the penalty function is quadratic $P(e_{t+h}) := e_{t+h}^2$. Hence, the rate of the penalty increases with the size of the forecasting error, whereas it is constant and only applies to excess errors in the case of the ν-SVR. Note that this insensitivity has a nontrivial consequence for the forecasting values. The Karush-Kuhn-Tucker conditions imply that only support vectors, i.e. points lying inside the insensitivity tube, will have nonzero Lagrange multipliers and contribute to the weight vector. In other words, all points whose errors are too big are effectively ignored at the optimum. Smola and Schölkopf (2004) call this the *sparsity* of the SVR. The empirical usefulness of this property for macro data is a question we will be answering in the coming sections.

To sum up, the Table 1 shows a list of all forecasting models and highlights their relationship with each of four features discussed above. The computational details on every model in this list are available in Appendix F.

4 Empirical setup

This section presents the data and the design of the pseudo-of-sample experiment used to generate the treatment effects above.

Table 1: List of all forecasting models

Models	Feature 1: selecting the function g	Feature 2: selecting the regularization	Feature 3: optimizing hyperparameters τ	Feature 4: selecting the loss function
Data-poor models				
AR,BIC	Linear		BIC	Quadratic
AR,AIC	Linear		AIC	Quadratic
AR,POOS-CV	Linear		POOS CV	Quadratic
AR,K-fold	Linear		K-fold CV	Quadratic
RRAR,POOS-CV	Linear	Ridge	POOS CV	Quadratic
RRAR,K-fold	Lineal	Ridge	K-fold CV	Quadratic
RFAR,POOS-CV	Nonlinear		POOS CV	Quadratic
RFAR,K-fold	Nonlinear		K-fold CV	Quadratic
KRRAR,POOS-CV	Nonlinear	Ridge	POOS CV	Quadratic
KRRAR,K-fold	Nonlinear	Ridge	K-fold CV	Quadratic
SVR-AR,Lin,POOS-CV	Linear		POOS CV	$\bar{\epsilon}$-insensitive
SVR-AR,Lin,K-fold	Linear		K-fold CV	$\bar{\epsilon}$-insensitive
SVR-AR,RBF,POOS-CV	Nonlinear		POOS CV	$\bar{\epsilon}$-insensitive
SVR-AR,RBF,K-fold	Nonlinear		K-fold CV	$\bar{\epsilon}$-insensitive
Data-rich models				
ARDI,BIC	Linear	PCA	BIC	Quadratic
ARDI,AIC	Linear	PCA	AIC	Quadratic
ARDI,POOS-CV	Linear	PCA	POOS CV	Quadratic
ARDI,K-fold	Linear	PCA	K-fold CV	Quadratic
RRARDI,POOS-CV	Linear	Ridge-PCA	POOS CV	Quadratic
RRARDI,K-fold	Linear	Ridge-PCA	K-fold CV	Quadratic
RFARDI,POOS-CV	Nonlinear	PCA	POOS CV	Quadratic
RFARDI,K-fold	Nonlinear	PCA	K-fold CV	Quadratic
KRRARDI,POOS-CV	Nonlinear	Ridge-PCR	POOS CV	Quadratic
KRRARDI,K-fold	Nonlinear	Ridge-PCR	K-fold CV	Quadratic
$(B_1, \alpha = \hat{\alpha})$,POOS-CV	Linear	EN	POOS CV	Quadratic
$(B_1, \alpha = \hat{\alpha})$,K-fold	Linear	EN	K-fold CV	Quadratic
$(B_1, \alpha = 1)$,POOS-CV	Linear	Lasso	POOS CV	Quadratic
$(B_1, \alpha = 1)$,K-fold	Linear	Lasso	K-fold CV	Quadratic
$(B_1, \alpha = 0)$,POOS-CV	Linear	Ridge	POOS CV	Quadratic
$(B_1, \alpha = 0)$,K-fold	Linear	Ridge	K-fold CV	Quadratic
$(B_2, \alpha = \hat{\alpha})$,POOS-CV	Linear	EN-PCA	POOS CV	Quadratic
$(B_2, \alpha = \hat{\alpha})$,K-fold	Linear	EN-PCA	K-fold CV	Quadratic
$(B_2, \alpha = 1)$,POOS-CV	Linear	Lasso-PCA	POOS CV	Quadratic
$(B_2, \alpha = 1)$,K-fold	Linear	Lasso-PCA	K-fold CV	Quadratic
$(B_2, \alpha = 0)$,POOS-CV	Linear	Ridge-PCA	POOS CV	Quadratic
$(B_2, \alpha = 0)$,K-fold	Linear	Ridge-PCA	K-fold CV	Quadratic
$(B_3, \alpha = \hat{\alpha})$,POOS-CV	Linear	EN-PCR	POOS CV	Quadratic
$(B_3, \alpha = \hat{\alpha})$,K-fold	Linear	EN-PCR	K-fold CV	Quadratic
$(B_3, \alpha = 1)$,POOS-CV	Linear	Lasso-PCR	POOS CV	Quadratic
$(B_3, \alpha = 1)$,K-fold	Linear	Lasso-PCR	K-fold CV	Quadratic
$(B_3, \alpha = 0)$,POOS-CV	Linear	Ridge-PCR	POOS CV	Quadratic
$(B_3, \alpha = 0)$,K-fold	Linear	Ridge-PCR	K-fold CV	Quadratic
SVR-ARDI,Lin,POOS-CV	Linear	PCA	POOS CV	$\bar{\epsilon}$-insensitive
SVR-ARDI,Lin,K-fold	Linear	PCA	K-fold CV	$\bar{\epsilon}$-insensitive
SVR-ARDI,RBF,POOS-CV	Nonlinear	PCA	POOS CV	$\bar{\epsilon}$-insensitive
SVR-ARDI,RBF,K-fold	Nonlinear	PCA	K-fold CV	$\bar{\epsilon}$-insensitive

Note: PCA stands for Principal Component Analysis, EN for Elastic Net regularizer, PCR for Principal Component Regression.

4.1 Data

We use historical data to evaluate and compare the performance of all the forecasting models described previously. The dataset is FRED-MD, publicly available at the Federal Reserve of St-Louis's web site. It contains 134 monthly US macroeconomic and financial indicators observed from 1960M01 to 2017M12. Many macroeconomic and financial indicators are usually very persistent or not stationary. We follow Stock and Watson (2002b) and McCracken

and Ng (2016) in the choice of transformations in order to achieve stationarity. The details on the dataset and the series transformation are all in McCracken and Ng (2016). Even this is a fairly high-dimensional time series framework, it is not a big data setup as one finds in typical cross-sectional analysis. FRED does contain more than 500,000 time series giving the possibility to considerably augment X_t. However, we stick to FRED-MD for several reasons. First, we want to have the out-of-sample period as long as possible and most of the variables available today do not start early enough.[11] Second, most of the timely available series are (very) disaggregated components of the variables in FRED-MD. Boivin and Ng (2006) show that adding many similar series negatively affects the ability of PC estimator to span the space of common factors. Third, FRED-MD is the standard high-dimensional dataset that has been extensively used in the macroeconomic forecasting literature. Therefore, we prefer to stay with the literature and explore the limits of the above models in that environment.

4.2 Variables of Interest

We focus on predicting five macroeconomic variables: Industrial Production (INDPRO), Unemployment rate (UNRATE), Consumer Price Index (INF), difference between 10-year Treasury Constant Maturity rate and Federal funds rate (SPREAD) and housing starts (HOUST). These are standard candidates in the forecasting literature and are representative macroeconomic indicators of the US economy. In particular, we treat INDPRO, CPI and HOUST as $I(1)$ variables so we forecast the average growth rate over h periods as in equation (3). The unemployment rate is considered $I(1)$ and we target the average first-difference as in (3) but without logs. The spread is modeled as $I(0)$ and the target is constructed as in (2).[12]

4.3 Pseudo-Out-of-Sample Experiment Design

The pseudo-out-of-sample period is 1980M01 - 2017M12. The forecasting horizons considered are 1, 3, 9, 12 and 24 months. Hence, there are 456 evaluation periods for each horizon. All models are estimated recursively with an expanding window.

Hyperparameter fine tuning is done with in-sample criterion (AIC and BIC) and using two types of cross validation (POOS CV and k-fold). The in-sample model selection is standard, we only fix the upper bounds for the set of HPs. In contrast, the CV can be very

[11] This problem can, however, be partially solved if the rolling window strategy for the validation set is used, but then the number of time periods stays low for every forecasting exercise.

[12] The US consumer price index is sometimes modeled as $I(2)$ because of the possible stochastic trend in inflation rate during 70's and 80's ((Stock and Watson, 2002b), McCracken and Ng (2016)). However, since the inflation targeting regime, inflation rate is more stationary and since our pseudo-out-of-sample exercise covers 1980-2017 period, we decided to treat the price index as $I(1)$ as Medeiros et al. (2019). Moreover, we have compared the mean squared predictive errors of best models under $I(1)$ and $I(2)$ alternatives, and found that errors are minimized when predicting the inflation rate directly.

computationally extensive in a long time series evaluation period as in this paper. Ideally, one would re-optimize every model, for every target variable and for each forecasting horizon, for every out-of-sample period. For the POOS CV, where the CV in the validation set mimics the out-of-sample prediction in the test sample, the POOS period consists of last 25% of the validation set. In case of k-fold CV, we set $k = 5$. We re-optimize hyperparameters every two years. This is reasonable since as it is the case with parameters, we do not expect hyperparameters to change drastically with the addition of a few data points.

Appendix E describes both cross-validation techniques in details, while the information on upper / lower bounds and grid search for hyperaparameters for every model is available in Appendix F.

4.4 Forecast Evaluation Metrics

Following a standard practice in the forecasting literature, we evaluate the quality of our point forecasts using the root Mean Square Prediction Error (MSPE). The standard Diebold-Mariano (DM) test procedure is used to compare the predictive accuracy of each model against the reference (ARDI,BIC) model.

We also implement the Model Confidence Set (MCS) introduced in Hansen et al. (2011). The MCS allows us to select the subset of best models at a given confidence level. It is constructed by first finding the best forecasting model, and then selecting the subset of models that are not significantly different from the best model at a desired confidence level. We construct each MCS based on the quadratic loss function and 4000 bootstrap replications. As expected, we find that the $(1 - \alpha)$ MCS contains more models when α is smaller. Following Hansen et al. (2011), we present the empirical results for 75% confidence interval.

These evaluation metrics are standard outputs in a forecasting horse race. They allow to verify the overall predictive performance and to classify models according to DM and MCS tests. Regression analysis from section 2.3 will be used to distinguish the marginal treatment effect of each ML ingredient that we try to evaluate here.

5 Results

We present the results in several ways. First, for each variable, we show standard tables containing the relative root MSPEs (to AR,BIC model) with DM and MCS outputs, for the whole pseudo-out-of-sample and NBER recession periods. Second, we evaluate the marginal effect of important features of ML using regressions described in section 2.3.

5.1 Overall Predictive Performance

Tables 3 - 7, in Appendix A, summarize the overall predictive performance in terms of root MSPE relative to the reference model AR,BIC. The analysis is done for the full out-of-sample as well as for NBER recessions taken separately (i.e., when the target belongs to a recession episode). This address two questions: is ML already useful for macroeconomic forecasting and when?[13]

In case of industrial production, Table 3 shows that principal component regressions B_2 and B_3 with Ridge and Lasso penalty respectively are the best at short-run horizons of 1 and 3 months. The kernel ridge ARDI with POOS CV is best for $h = 9$, while its autoregressive counterpart with K-fold minimizes the MSPE at the one year horizon. Random forest ARDI, the alternative nonlinear approximator, outperforms the reference model by 11% for $h = 24$. During recessions, the ARDI with CV is the best for 1, 3 and 9 months ahead, while the nonlinear SVR-ARDI minimizes the MSPE at the one year horizon. The ridge regression ARDI is the best for $h = 24$. Ameliorations with respect to AR,BIC are much larger during economic downturns, and the MCS selects less models.

Results for the unemployment rate, table 4, highlight the performance of nonlinear models, Kernel ridge and Random forests, especially for longer horizons. Improvements with respect to the AR,BIC model are bigger for both full OOS and recessions. MCSs are narrower than in case of INDPRO. Similar pattern is observed during NBER recessions. Table 5 summarizes results for the Spread. Nonlinear models are generally the best, combined with data-rich predictors' set. Occasionally, autoregressive models with the kernel ridge or SVR specifications produce minimum MSE.

In the case of inflation, table 6 shows that the kernel ridge autoregressive model with K-fold CV is the best for 3, 9 and 12 months ahead, while the nonlinear SVR-ARDI optimized K-fold cross-validation reduces the MSPE by more than 20% at two year horizon. Random forests models are also very resilient, confirming the findings in Medeiros et al. (2019). However, approximating more general nonlinear behavior by the RBF kernel in KRR models offers a better performance. During recessions, the fat regression models (B_1) are the best at short horizons, while the ridge regression ARDI with K-fold dominates for $h = 9, 12, 24$. Finally, housing starts are best predicted with non-linear data-rich models for almost all horizons, as shown in table 7.

Overall, using data-rich models and nonlinear g functions seems to be a game changer for macroeconomic prediction. SVR specifications are occasionally among the best models as well as the shrinkage methods from section 3.2. In addition, their marginal contribution depends on the state of the economy.

[13] The knowledge of the models that have performed best historically during recessions is of interest for practitioners. If the probability of recession is high enough at a given period, our results can provide an ex-ante guidance on which model is likely to perform best in such circumstances.

5.2 Disentangling ML Treatment Effects

The results in the previous section does not allow easily to disentangle the marginal effects of important features of machine learning as presented in section 3, which is the most important goal of this paper. Before we employ the evaluation strategy depicted in section 2.3, we first use a Random forest as an exploration tool. Since creating the relevant dummies and interaction terms to fully describe the environment is a hard task in presence of many treatment effects, a regression tree well suited to reveal the potential of ML features in explaining the results from our experiment. We report the importance of each features in what is a potentially a very non-linear model.[14] For instance, the tree could automatically create interactions such as $I(NL = 1) * I(h \leq 12)$, that is, some condition on non-linearities and horizon forecast.

Figure 1 plots the relative importance of machine learning features in our macroeconomic forecasting experiment. The space of possible interaction is constructed with dummies for horizon, variable, recession periods, loss function and H_t^+, and categorical variables nonlinearity, shrinkage and hyperameters' tuning that follow the classification as in Table 1. As expected, target variables, forecasting horizons and the state of economy are important elements. Among our features of interest, the nonlinearity turns to be the most relevant, which confirms our overall analysis from the previous section. The data richness is the second important ingredient. The rest of the features are relatively less relevant and appears in the following decreasing order of importance: in-sample loss function, hyperparameters' optimization and alternative shrinkage methods.

Despite its richness in terms of interactions among determinants, the Random forest analysis does not provide the sign of the importance of each feature not it measures their

[14] The importance of each ML ingredient is obtain with feature permutation. The following process describes the estimation of out-of-bag predictor importance values by permutation. Suppose a random forest of B trees and p is the number of features.

1. For tree b, $b = 1, ..., B$:
   (a) Identify out-of-bag observations and indices of features that were split to grow tree b, $s_b \subseteq 1, ..., p$.
   (b) Estimate the out-of-bag error $u^2_{t,h,v,m,\mathbf{b}}$.
   (c) For each feature x_j, $j \in s_b$:
      i. Randomly permute the observations of x_j.
      ii. Estimate the model squared errors, $u^2_{t,h,v,m,\mathbf{b,j}}$, using the out-of-bag observations containing the permuted values of x_j.
      iii. Take the difference $d_{\mathbf{bj}} = u^2_{t,h,v,m,\mathbf{b,j}} - u^2_{t,h,v,m,\mathbf{b}}$.
2. For each predictor variable in the training data, compute the mean, $\bar{d}_j$, and standard deviation, σ_j, of these differences over all trees, $j = 1, ..., p$.
3. The out-of-bag predictor importance by permutation for x_j is $\bar{d}_j / \sigma_j$

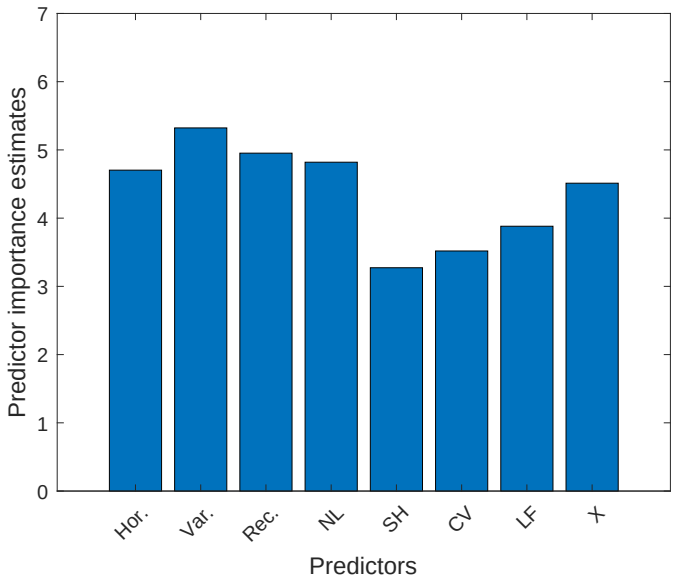

Figure 1: This figure presents predictive importance estimates. Random forest is trained to predict $R^2_{t,h,v,m}$ defined in (10) and use out-of-bags observations to assess the performance of the model and compute features' importance. NL, SH, CV and LF stand for nonlinearity, shrinkage, cross-validation and loss function features respectively. A dummy for H_t^+ models, X, is included as well.

marginal contributions. To do so, and armed with insights from the Random forest analysis, we turn now to regression analysis described in section 2.3.

Figure 2 shows the distribution of $\dot{\alpha}_F^{(h,v)}$ from equation (10) done by (h, v) subsets. Hence, here we allow for heterogeneous treatment effects according to 25 different targets. This figure highlights by itself the main findings of this paper. **First**, non-linearities improve drastically substantially the forecasting accuracy in almost all situations. The effects are positive and significant for all horizons in case of INDPRO and SPREAD, and for most of the cases when predicting UNRATE, INF and HOUST. The improvements of the non-linearity treatment reach up to 0.25% in terms of pseudo-R^2. **Second**, alternative regularization means of dimensionality reduction do not improve on average over the standard factor model, except few cases. Choosing sparse modeling can decrease the forecast accuracy by up to 0.2% of the pseudo-R^2 which is not negligible.

Third, the average effect of CV appears not significant. However, as we will see in section 5.2.3, the averaging in this case hides some interesting and relevant differences between K-fold and POOS CVs, that the Random forest analysis in Figure 1 has picked up. **Fourth**, on average, dropping the standard in-sample squared-loss function for what the SVR proposes is not useful, except in very rare cases. **Fifth** and lastly, the marginal benefits of data-rich models (X) seems roughly to increase with horizons for every variable-horizon pair, except for few cases with spread and housing. Note that this is almost exactly like the picture we described for NL. Indeed, visually, it seems like the results for X are a compressed-range

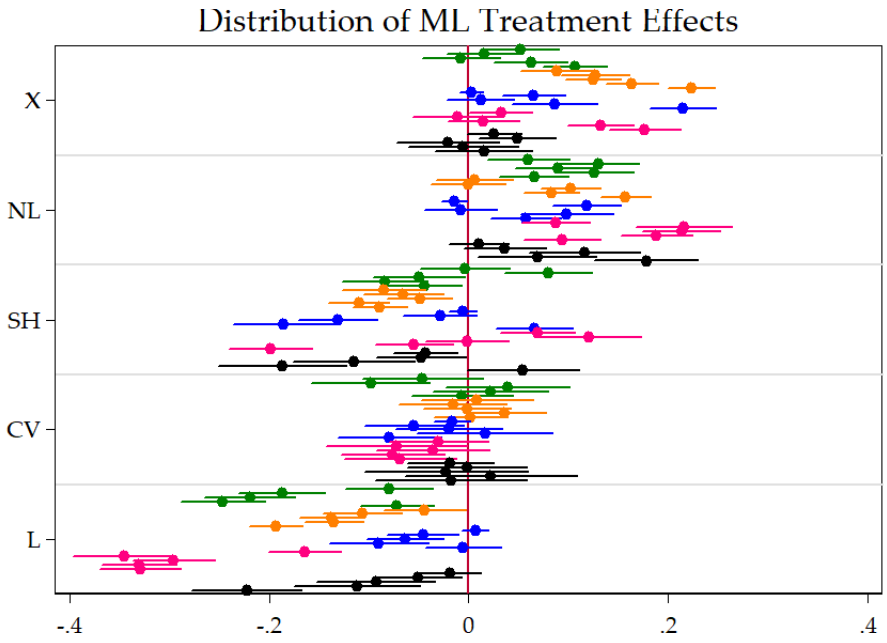

Figure 2: This figure plots the distribution of $\grave{\alpha}_F^{(h,v)}$ from equation (10) done by (h, v) subsets. That is, we are looking at the average partial effect on the pseudo-OOS R^2 from augmenting the model with ML features, keeping everything else fixed. X is making the switch from data-poor to data-rich. Finally, variables are INDPRO, UNRATE, SPREAD, INF and HOUST. Within a specific color block, the horizon increases from $h = 1$ to $h = 24$ as we are going down. As an example, we clearly see that the partial effect of X on the R^2 of INF increases drastically with the forecasted horizon h. SEs are HAC. These are the 95% confidence bands.

version of NL that was translated to the right. Seeing NL models as data augmentation via some basis expansions, we can conclude that for predicting macroeconomic variables, we either need to augment the AR(p) model with more regressors either created from the lags of the dependent variable itself or coming from additional data. The possibility of joining these two forces to create a "data-filthy-rich" model is studied in section 5.2.1.

It turns out these findings are somewhat robust as graphs included in the appendix section B show. ML treatment effects plots of very similar shapes are obtained for data-poor models only (Figure 13), data-rich models only (Figure 14) and recessions periods (Figure 15). The only exception is the data-rich feature that has negative and significant effects for predictions of housing starts when we condition the analysis on the last 20 years of the forecasting exercise (Figure 16).

Figure 3 aggregates by h and v in order to clarify whether variable or horizon heterogeneity matters most. Two facts detailed earlier are now quite easy to see. For both X and NL, the average marginal effects roughly increase in h. In addition, it is now clear that all the variables benefit from both additional information and non-linearities. Alternative shrinkage is least harmful for inflation and housing, and at short horizons. Cross-validation has negative and sometimes significant impacts, while the SVR loss-function is clearly not a good idea.

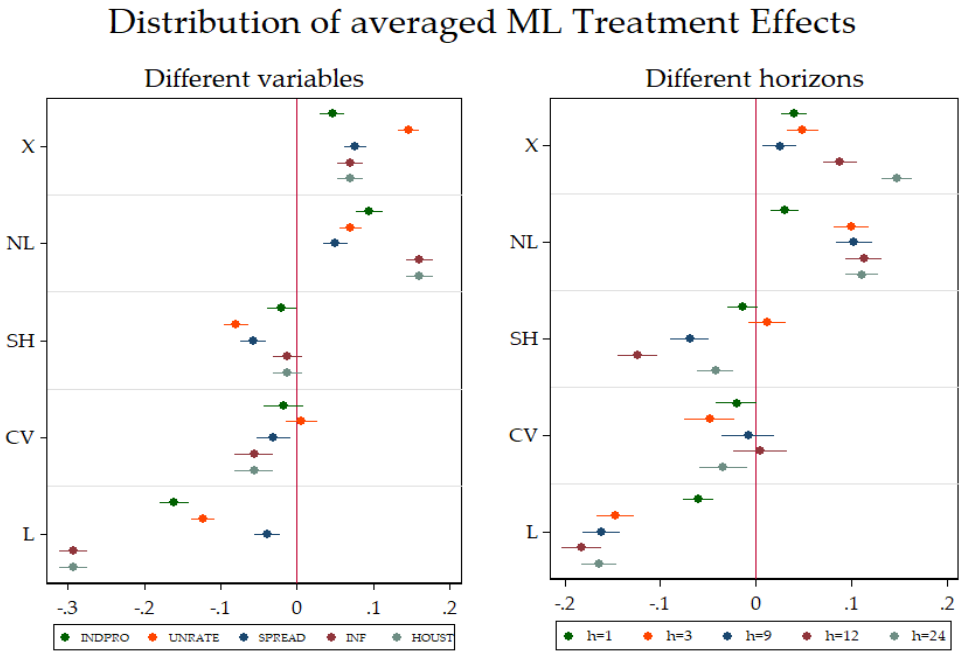

Figure 3: This figure plots the distribution of $\dot{\alpha}_F^{(v)}$ and $\dot{\alpha}_F^{(h)}$ from equation (10) done by h and v subsets. That is, we are looking at the average partial effect on the pseudo-OOS R^2 from augmenting the model with ML features, keeping everything else fixed. X is making the switch from data-poor to data-rich. However, in this graph, $v-$specific heterogeneity and $h-$specific heterogeneity have been integrated out in turns. SEs are HAC. These are the 95% confidence bands.

Finally, appendix D shows that results obtained using the squared loss are very consistent with what one would obtain using the absolute loss. The importance of each feature and the way it behaves according to the variable/horizon pair is the same. Indeed, most of the heterogeneity is variable specific while there are clear horizon patterns emerging when we average out variables.

In what follows we break down averages and run specific regressions as in (11) to study how homogeneous are the $\dot{\alpha}_F$'s reported above.

5.2.1 Non-linearities

Figure 4 suggests that non-linearities can be very helpful at forecasting all the five variables in the data rich-environment. The marginal effects of Random Forests and KRR are almost never statistically different for data-rich models, except for inflation combined with data-rich, suggesting that the common NL feature is the driving force. However, this is not the case for data-poor models where the kernel-type nonlinearity shows significant improvements for all variables, while the random forests have positive impact on predicting INDPRO and inflation, but decrease forecasting accuracy for the rest of the variables.

Figure 5 suggest that non-linearities are in general more useful for longer horizons in data rich environment while the KRR can be harmful in very short horizon. Note again that

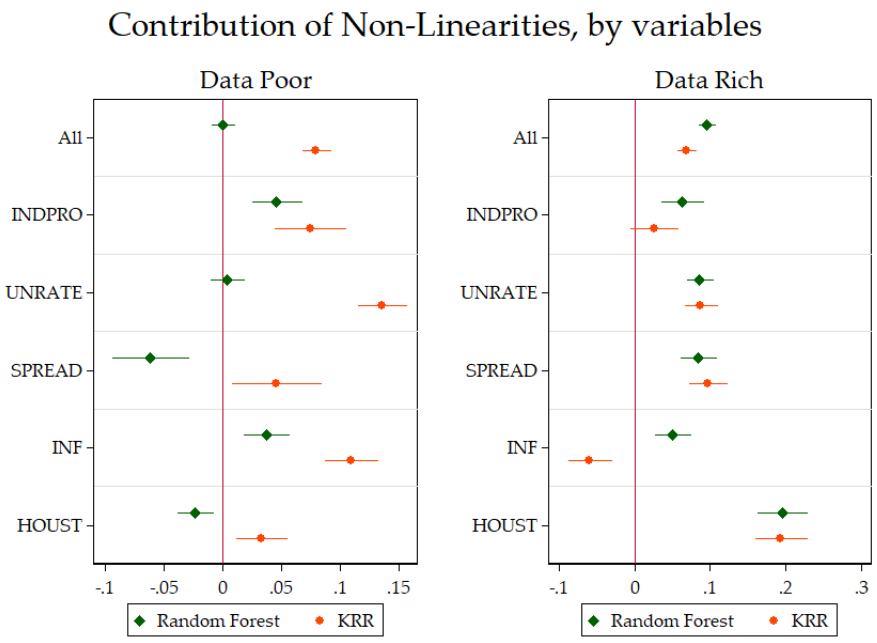

Figure 4: This figure compares the two NL models averaged over all horizons. The unit of the x-axis are improvements in OOS R^2 over the basis model. SEs are HAC. These are the 95% confidence bands.

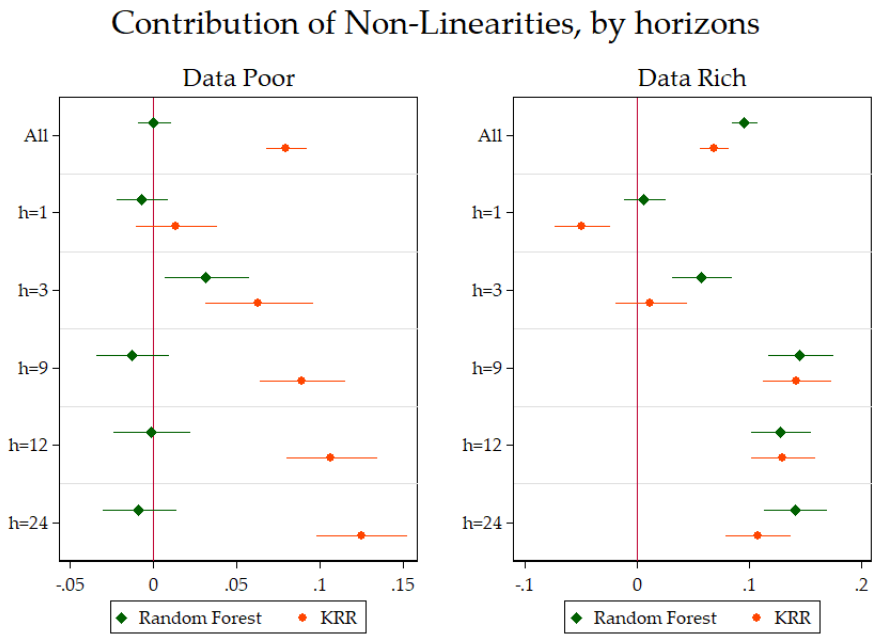

Figure 5: This figure compares the two NL models averaged over all variables. The unit of the x-axis are improvements in OOS R^2 over the basis model. SEs are HAC. These are the 95% confidence bands.

both non-linear models follow the same pattern for data-rich models with Random Forest often being better (but never statistically different from KRR). For data-poor models, it is KRR that has a (statistically significant) growing advantage as h increases.

Seeing NL models as data augmentation via some basis expansions, we can join the two facts together to conclude that the need for a complex and "data-filthy-rich" model arise for predicting macroeconomic variables at longer horizons.

Figure 6 plots the cumulative and 3-year rolling window MSPE for linear and non-linear data-poor and data-rich models, for $h = 12$. The cumulative MSPE clearly shows the positive impact on forecast accuracy of both non-linearities and data-rich environment for all series except INF. The MSPE trajectories have rather parallel trends with non-linear versions of the ARDI model being at the lower enveloppe. In case of INDPRO, their relative improvement tends to shrink over time, while it stays more stable for UNRATE, SPREAD and HOUST. In addition, both types of non-linearities present similar temporal patterns. For CPI inflation the KRR,AR and ARDI dominate until 2010, suggesting that for predicting inflation large data and non-linearities are to some limit interchangeable in terms of MSPE forecast accuracy.

The rolling window (right column of figure 6) depicts the changing level of forecast accuracy. For all series except the SPREAD, there is a common cyclical behavior with two relatively similar peaks: the 1981 and the 2008 recessions. Usually, we remark a lower level of MSPE between 1985 and 2007, which corresponds to the Great Moderation period, and the MSPE are mostly back to that historical average after the Great Recession.

5.2.2 Alternative Dimension Reduction

Figure 7 shows that the ARDI reduces dimensionality in a way that certainly works well with economic data: all competing schemes do at most as good on average. It is overall safe to say that on average, all shrinkage schemes give similar or lower performance, which is in line with conclusions from Stock and Watson (2012) and Kim and Swanson (2018), but contrary to Smeekes and Wijler (2018). No clear superiority for the Bayesian versions of some of these models was also documented in De Mol et al. (2008). This suggests that the factor model view of the macroeconomy is quite accurate in the sense that when we use it as a mean of dimensionality reduction, it extracts the most relevant information to forecast the relevant time series. This is good news. The ARDI is the simplest model to run and results from the preceding section tells us that adding non-linearities to an ARDI can be quite helpful. For instance, B_1 models where we basically keep all regressors do approximately as well as the ARDI when used with CV-POOS. However, it is very hard to consider non-linearities in this high-dimensional setup. Since the ARDI does a similar (or better) job of dimensionality reduction, it is both convenient for subsequent modeling steps and does not loose relevant information.

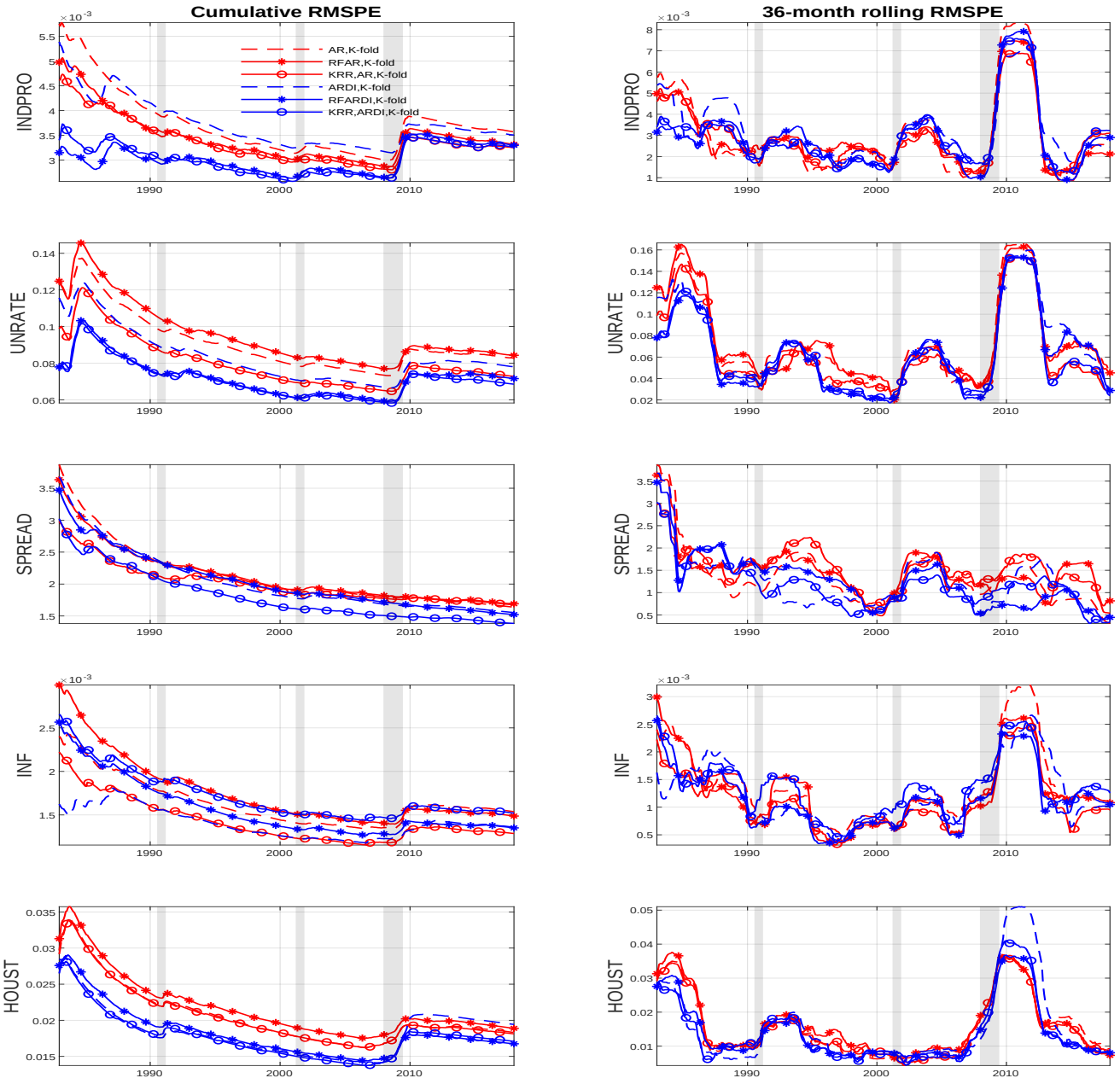

Figure 6: This figure shows the cumulative MSPE (left column) and 3-year rolling window MSPE (right) for linear and non-linear data-poor and data-rich models, at 12-month horizon.

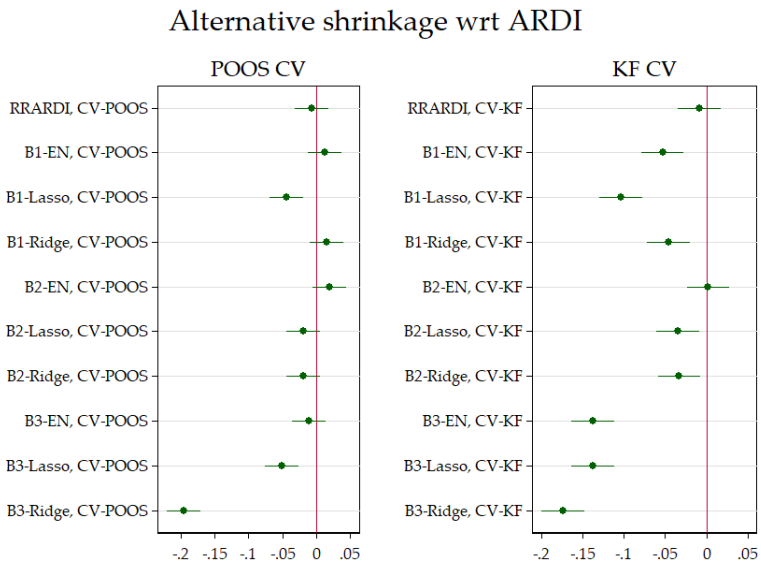

Figure 7: This figure compares models of section 3.2 averaged over all variables and horizons. The unit of the x-axis are improvements in OOS R^2 over the basis model. The base models are ARDIs specified with POOS-CV and KF-CV respectively. SEs are HAC. These are the 95% confidence bands.

Obviously, the deceiving average behavior of alternative (standard) shrinkage methods does not mean there cannot be interesting (h, v) cases where using a different dimensionality reduction has significant benefits as discussed in section 5.1 and Smeekes and Wijler (2018). Furthermore, LASSO and Ridge can still be useful to tackle specific time series econometrics problems (other than dimensionality reduction), as shown with time-varying parameters in Goulet Coulombe (2019).

5.2.3 Hyperparameter Optimization

Figure 8 shows how many total regressors are kept by different model selection methods. As expected, BIC is almost always the lower envelope of each of these graphs and is the only true guardian of parsimony in our setup. AIC also selects relatively sparse models. It is also quite visually clear that both cross-validations favors larger models, especially when combined with Ridge regression. Most likely as a results of expanding window setup, we remark a common upward trends for all model selection methods in case of INDPRO, UNRATE and SPREAD (at least after 1985). This is not the case for inflation where large models has been selected during the Great Inflation period and most recently since 2005. In case of housing starts, there is downward trend since 2000's which is consistent with the finding in Figure 16 that data-poor models do better in last 20 years for that variable. Finally, CV-POOS has quite a distinctive behavior. It is more volatile and seems to select bigger models for unemployment rate, spread and housing. While K-fold also selects models of considerable size, it does so in a more slowly growing fashion. This is not surprising given

the fact that K-fold samples from all available data to build the CV criterion: adding new data points only gradually change the average. CV-POOS is a shorter window approach that offers flexibility against structural hyperparameters change at the cost of greater variance and vulnerability of rapid change of regimes in the data.

We know that different model selection methods lead to quite different models, but what about their predictions? First, let us note that changes in OOS-R^2 are much smaller in magnitude for CV (as can seen easily in figures 2 and 3) than for other studied ML treatment effects. Nevertheless, Table D tells many interesting tales. The models included in the regressions are the standard linear ARs and ARDIs (that is, excluding the Ridge versions) that have all been tuned using BIC, AIC, CV-POOS and CV-KF. First, we see that overall, only CV-POOS is distinctively worse, especially in data-rich environment, and that AIC and CV-KF are not significantly different from BIC on average. For data-poor models and during recessions, AIC and CV-KF are being significantly better than BIC in downturns, while CV-KF seems harmless. The state-dependent effects are not significant in data-rich environment. A conclusion is that, for that class of models, we can safely opt for either BIC or CV-KF. Assuming some degree of external validity beyond that model class, we can be re-assured that the quasi-necessity of leaving ICs behind when opting for more complicated ML models is not harmful.

Table 2: CV comparaison

	(1) All	(2) Data-rich	(3) Data-poor	(4) Data-rich	(5) Data-poor
CV-KF	-0.0380	-0.314	0.237	-0.494	-0.181
	(0.800)	(0.711)	(0.411)	(0.759)	(0.438)
CV-POOS	-1.351	-1.440*	-1.262**	-1.069	-1.454***
	(0.800)	(0.711)	(0.411)	(0.759)	(0.438)
AIC	-0.509	-0.648	-0.370	-0.580	-0.812
	(0.800)	(0.711)	(0.411)	(0.759)	(0.438)
CV-KF * Recessions				1.473	3.405**
				(2.166)	(1.251)
CV-POOS * Recessions				-3.020	1.562
				(2.166)	(1.251)
AIC * Recessions				-0.550	3.606**
				(2.166)	(1.251)
Observations	91200	45600	45600	45600	45600

Standard errors in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

We will now consider models that are usually always tuned by CV and compare the performance of the two CVs by horizon and variables.

Since we are now pooling multiple models, including all the alternative shrinkage mod-

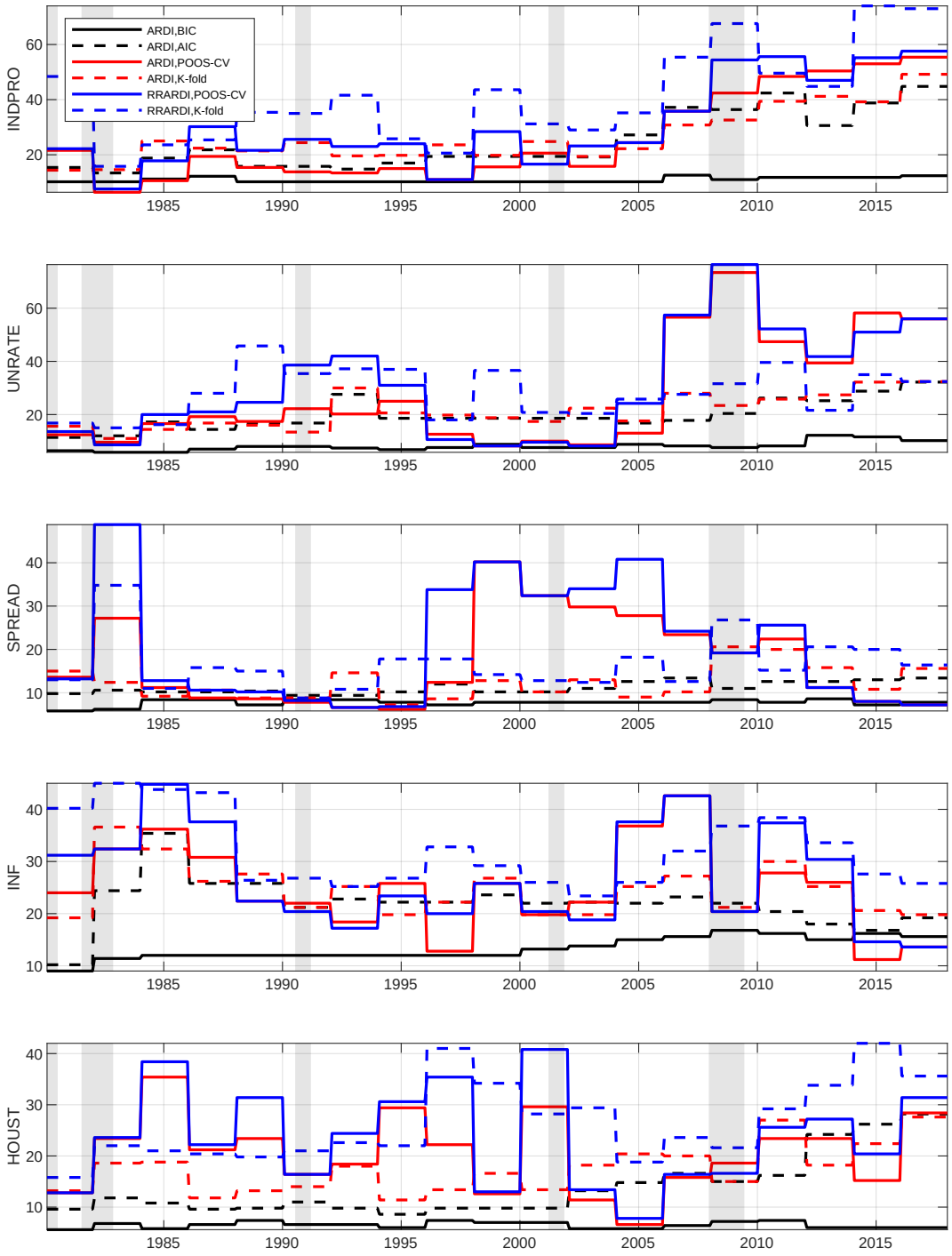

Figure 8: This figure shows the total number of regressors for the linear ARDI models. Results averaged across horizons.

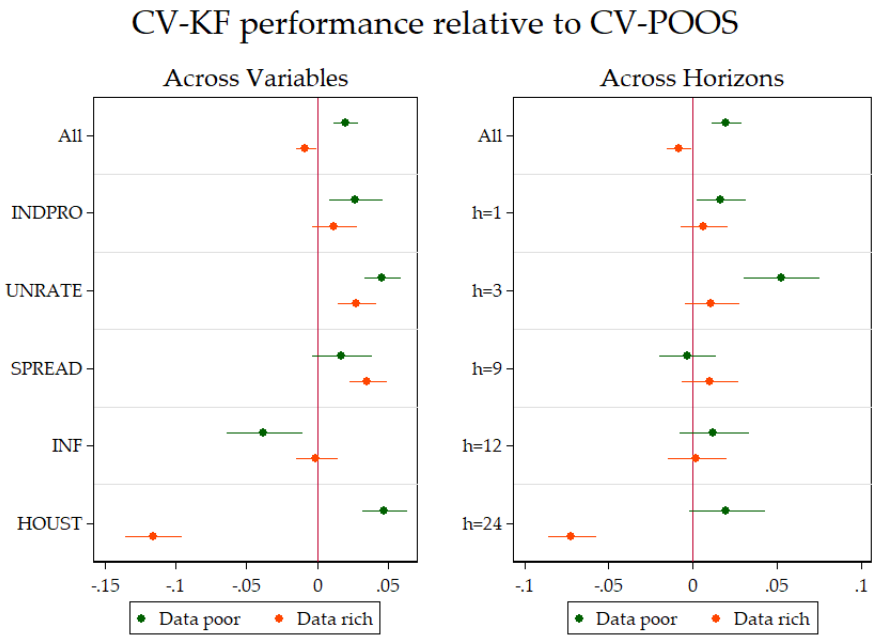

Figure 9: This figure compares the two CVs procedure averaged over all the models that use them. The unit of the x-axis are improvements in OOS R^2 over the basis model. SEs are HAC. These are the 95% confidence bands.

els, if a clear pattern only attributable to a certain CV existed, it would most likely appear in Figure 9. What we see are two things. First, CV-KF is at least as good as CV-POOS on average for almost all variables and horizons, irrespective of the informational content of the regression. The exceptions are HOUST in data-rich and INF in data-poor frameworks, and the two-year horizon with large data. Figure 10's message has the virtue of clarity. CV-POOS's failure is mostly attributable to its poor record in recessions periods for the first three variables at any horizon. Note that this is the same subset of variables that benefits from adding in more data (X) and non-linearities as discussed in 5.2.1.

Intuitively, by using only recent data, CV-POOS will be more robust to gradual structural change but will perhaps have an Achilles heel in regime switching behavior. If the optimal hyperparameters are state-dependent, then a switch from expansion to recession at time t can be quite harmful. K-fold, by taking the average over the whole sample, is less immune to such problems. Since results in 5.1 point in the direction that smaller models are better in expansions and bigger models in recessions, the behavior of CV and how it picks the effective complexity of the model can have an important effect on overall predictive ability. This is exactly what we see in Figure 10: CV-POOS is having a hard time in recessions with respect to K-fold.

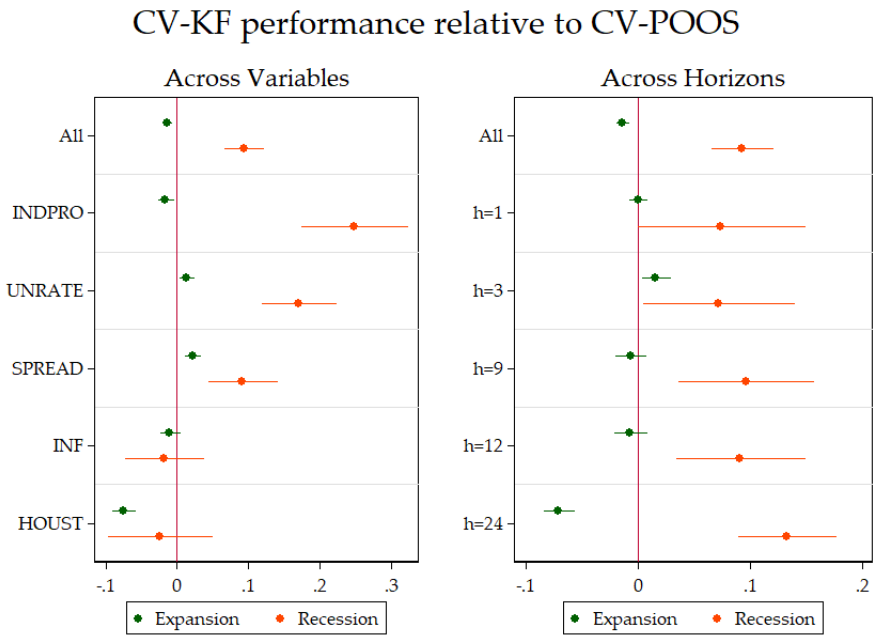

Figure 10: This figure compares the two CVs procedure averaged over all the models that use them. The unit of the x-axis are improvements in OOS R^2 over the basis model. SEs are HAC. These are the 95% confidence bands.

5.2.4 Loss Function

In this section, we investigate whether replacing the l_2 norm as an in-sample loss function for the SVR machinery helps in forecasting. We again use as baseline models ARs and ARDIs trained by the same corresponding CVs. The very nature of this ML feature is that the model is less sensible to extreme residuals, thanks to the ϵ-insensitivity tube. We first compare linear models in Figure 11. Clearly, changing the loss function is mostly very harmful and that is mostly due to recessions period. However, in expansions, the linear SVR is better on average than a standard ARDI for UNRATE and SPREAD, but these small gains are clearly offset (on average) by the huge recession losses.

The SVR (or the better-known SVM) is usually used in its non-linear form. We hereby compare KRR and SVR-NL to study whether the loss function effect could reverse when a non-linear model is considered. Comparing these models makes sense since they both use the same kernel trick (with a RBF kernel). Hence, like linear models of Figure 11, models in Figure 12 only differ by the use of a different loss function $\hat{L}$. It turns out conclusions are exactly the same as for linear models with the negative effects being slightly smaller in non-linear world. There are few exceptions: inflation rate and on-month ahead horizon during recessions. Furthermore, Figures 17 and 18 in Appendix C confirm that these findings are valid for both the data-rich and the data-poor environments. Hence, these results confirms that $\hat{L}$ is not the most salient feature of ML, at least for macroeconomic forecasting. If researchers are interested in using its kernel trick to bring in non-linearities, they should

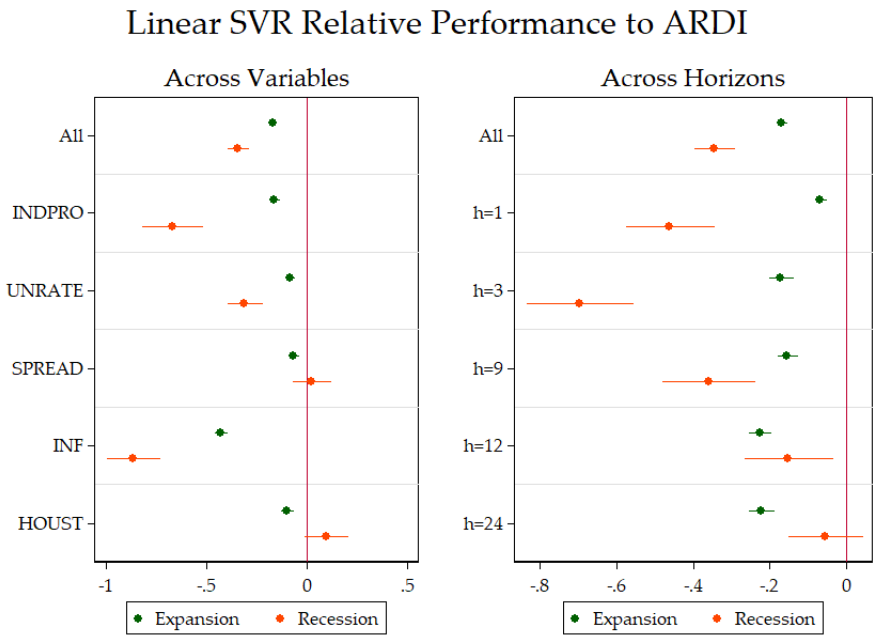

Figure 11: This graph displays the marginal (un)improvments by variables and horizons to opt for the SVR in-sample loss function in both recession and expansion periods. The unit of the x-axis are improvements in OOS R^2 over the basis model. SEs are HAC. These are the 95% confidence bands.

rather use the lesser-known KRR.

6 Conclusion

In this papers we have studied important underlying features driving machine learning techniques in the context of macroeconomic forecasting. We have considered many machine learning methods in a substantive POOS setup over almost 40 years for 5 key variables and 5 different horizons. We have classified these models by "features" of machine learning: non-linearities, regularization, cross-validation and alternative loss function. The four aspects of ML are nonlinearities, regularization, cross-validation and alternative loss function. The data-rich and data-poor environments were considered. In order to recover their marginal effects on forecasting performance, we designed a series of experiments that easily allow to identify the treatment effects of interest.

First, non-linearities either improve substantially the forecasting accuracy. The benefits are significant for industrial production, unemployment rate, term spread, inflation and housing starts and increase with horizons, especially if combined with factor models.

The first result point in the direction that non-linearities are the true game-changer for the data rich environment, as they improve substantially the forecasting accuracy for all macroeconomic variables in our exercise and especially when predicting at long horizons.

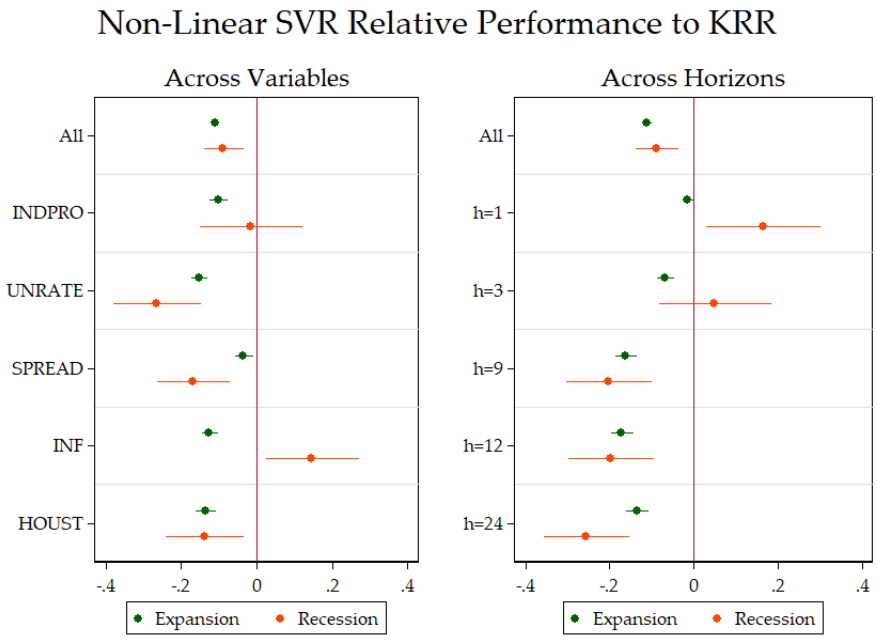

Figure 12: This graph displays the marginal (un)improvments by variables and horizons to opt for the SVR in-sample loss function in both recession and expansion periods. The unit of the x-axis are improvements in OOS R^2 over the basis model. SEs are HAC. These are the 95% confidence bands.

This gives a stark recommendation for practitioners. It recommends for most variables and horizons what is in the end a partially non-linear factor model – that is, factors are still obtained by PCA. The best of ML (at least of what considered here) can be obtained by simply generating the data for a standard ARDI model and then feed it into a ML non-linear function of choice. The second result is that the standard factor model remains the best regularization. Third, if cross-validation has to be applied to select models' features, the best practice is the standard K-fold. Finally, one should stick with the standard L_2 loss function.

References

Ahmed, N. K., Atiya, A. F., Gayar, N. E., and El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5):594–621.

Athey, S. (2018). The impact of machine learning on economics. *The Economics of Artificial Intelligence, NBER volume*, Forthcoming.

Bai, J. and Ng, S. (2002). Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221.

Bergmeir, C. and Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213.

Bergmeir, C., Hyndman, R. J., and Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics and Data Analysis*, 120:70–83.

Boh, S., Borgioli, S., Coman, A. B., Chiriacescu, B., Koban, A., Veiga, J., Kusmierczyk, P., Pirovano, M., and Schepens, T. (2017). European macroprudential database. Technical report, IFC Bulletins chapters, 46.

Boivin, J. and Ng, S. (2006). Are more data always better for factor analysis. *Journal of Econometrics*, 132:169–194.

Chen, J., Dunn, A., Hood, K., Driessen, A., and Batch, A. (2019). Off to the races: A comparison of machine learning and alternative data for predicting economic indicators. Technical report, Bureau of Economic Analysis.

Claeskens, G. and Hjort, N. L. (2008). *Model selection and model averaging*. Cambridge University Press, Cambridge, U.K.

De Mol, C., Giannone, D., and Reichlin, L. (2008). Forecasting using a large number of predictors: Is bayesian shrinkage a valid alternative to principal components? *Journal of Econometrics*, 146:318–328.

Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business and Economic Statistics*, 13:253–263.

Diebold, F. X. and Shin, M. (2018). Machine learning for regularized survey forecast combination: Partially-egalitarian lasso and its derivatives. *International Journal of Forecasting*, forthcoming.

Döpke, J., Fritsche, U., and Pierdzioch, C. (2015). Predicting recessions with boosted regression trees. Technical report, George Washington University, Working Papers No 2015-004, Germany.

Estrella, A. and Mishkin, F. (1998). Predicting us recessions: Financial variables as leading indicators. *Review of Economics and Statistics*, 80:45–61.

Fortin-Gagnon, O., Leroux, M., Stevanovic, D., and Surprenant, S. (2018). A large canadian database for macroeconomic analysis. Technical report, Department of Economics, UQAM.

Giannone, D., Lenza, M., and Primiceri, G. (2015). Prior selection for vector autoregressions. *Review of Economics and Statistics*, 97(2):436–451.

Giannone, D., Lenza, M., and Primiceri, G. (2017). Macroeconomic prediction with big data: the illusion of sparsity. Technical report, Federal Reserve Bank of New York.

Goulet Coulombe, P. (2019). Sparse and dense time-varying parameters using machine

learning. Technical report.

Granger, C. W. J. and Jeon, Y. (2004). Thick modeling. *Economic Modelling*, 21:323–343.

Hansen, P., Lunde, A., and Nason, J. (2011). The model confidence set. *Econometrica*, 79(2):453–497.

Hansen, P. R. and Timmermann, A. (2015). Equivalence between out-of-sample forecast comparisons and wald statistics. *Econometrica*, 83(6):2485–2505.

Hastie, T., Tibshirani, R., and Friedman, J. (2017). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer-Verlag, New York.

Kilian, L. and Lütkepohl, H. (2017). *Structural vector autoregressive analysis*. Cambridge University Press.

Kim, H. H. and Swanson, N. R. (2018). Mining big data using parsimonious factor, machine learning, variable selection and shrinkage methods. *International Journal of Forecasting*, 34(2):339–354.

Koenker, R. and Machado, J. A. F. (1999). Goodness of fit and related inference processes for quantile regression. *Journal of the American Statistical Association*, 94(448):1296–1310.

Kotchoni, R., Leroux, M., and Stevanovic, D. (2017). Macroeconomic forecast accuracy in a data-rich environment. Technical report, CIRANO, 2017s-05.

Li, J. and Chen, W. (2014). Forecasting macroeconomic time series: LASSO-based approaches and their forecast combinations with dynamic factor models. *International Journal of Forecasting*, 30:996–1015.

Litterman, R. B. (1979). Techniques of forecasting using vector autoregressions. Technical report.

Marcellino, M., Stock, J. H., and Watson, M. W. (2006). A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series. *Journal of Econometrics*, 135:499–526.

McCracken, M. W. and Ng, S. (2016). Fred-md: A monthly database for macroeconomic research. *Journal of Business and Economic Statistics*, 34(4):574–589.

Medeiros, M. C., Vasconcelos, G. F. R., Veiga, A., and Zilberman, E. (2019). Forecasting inflation in a data-rich environment: Benefits of machine learning methods. Technical report, Pontifical Catholic University of Rio de Janeiro.

Mullainathan, S. and Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, 31(2):574–589.

Nakamura, E. (2005). Inflation forecasting using a neural network. *Economics Letters*, 86(3):373–378.

Ng, S. (2014). Boosting recessions. *Canadian Journal of Economics*, 47(1):1–34.

Sermpinis, G., Stasinakis, C., Theolatos, K., and Karathanasopoulos, A. (2014). Inflation and unemployment forecasting with genetic support vector regression. *Journal of Forecasting*, 33(6):471–487.

Smalter, H. A. and Cook, T. R. A. (2017). Macroeconomic indicator forecasting with deep neural networks. Technical report, Federal Reserve Bank of Kansas City.

Smeekes, S. and Wijler, E. (2018). Macroeconomic forecasting using penalized regression methods. *International Journal of Forecasting*, 34(3):408–430.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3):199–211.

Stock, J. H. and Watson, M. W. (2002a). Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 97:1167–1179.

Stock, J. H. and Watson, M. W. (2002b). Macroeconomic forecasting using diffusion indexes. *Journal of Business and Economic Statistics*, 20(2):147–162.

Stock, J. H. and Watson, M. W. (2012). Generalized shrinkage methods for forecasting using many predictors. *Journal of Business and Economic Statistics*, 4(30):481–493.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting*, 16(4):437–450.

Ulke, V., Sahin, A., and Subasi, A. (2016). A comparison of time series and machine learning models for inflation forecasting: empirical evidence from the USA. *Neural Computing and Applications*, 1.

Zou, H., Hastie, T., and Tibshirani, R. (2007). On the “degrees of freedom" of the Lasso. *The Annals of Statistics*, 35(5):2173–2192.

A Detailed overall predictive performance

Table 3: Industrial Production: Relative Root MSPE

	Full Out-of-Sample					NBER Recessions Periods				
Models	h=1	h=3	h=9	h=12	h=24	h=1	h=3	h=9	h=12	h=24
Data-poor (H_t^-) models										
AR,BIC (RMSPE)	0,0765	**0.0515**	**0.0451**	**0.0428**	**0.0344**	0,127	0,1014	0,0973	0,0898	0,0571
AR,AIC	0.991*	**1.000**	**0.999**	**1.000**	**1.000**	0.987*	1	1	1	1
AR,POOS-CV	0,999	1.021***	**0.985***	**1.001**	1.032*	1,01	1.023***	0.988*	1	1.076**
AR,K-fold	0.991*	**1.000**	**0.987***	**1.000**	1.033*	0.987*	1	0.992*	1	1.078**
RRAR,POOS-CV	1,003	1.041**	**0.989**	**0.993***	**1.002**	1.039**	1.083**	0,991	0,993	1.016**
RRAR,K-fold	0.988**	**1.000**	**0.991**	**1.001**	**1.027**	0,992	1.007**	0,995	1.001**	1.074**
RFAR,POOS-CV	0,995	1,045	**0.985**	**0.955**	**0.991**	1,009	1,073	0.902***	0.890**	0,983
RFAR,K-fold	0,995	**1.020**	**0.960**	**0.930****	**0.983**	0,999	1,013	0.894***	0.887***	0.970*
KRR-AR,POOS-CV	1,023	1,09	**0.980**	**0.944**	**0.982**	1,117	1.166*	0.896**	0.853***	0.903***
KRR,AR,K-fold	**0.947*****	**0.937****	**0.936**	**0.910***	**0.959**	0.922**	0.902**	0.835***	**0.799*****	0.864***
SVR-AR,Lin,POOS-CV	1.134***	1.226***	1.114***	1.132***	**0.952***	1.186**	1.285***	1.079**	1.034***	0.893***
SVR-AR,Lin,K-fold	1.069*	1.159**	1.055**	1.042***	1.016***	1.268***	1.319***	1.067***	1.035***	1.013***
SVR-AR,RBF,POOS-CV	0,999	1.061***	**1.020**	**1.048**	**0.980**	1.062*	1.082***	0.876***	0.941***	0.930***
SVR-AR,RBF,K-fold	**0.978***	**1.004**	1.080*	1.193**	1.017***	0,992	1,009	0,989	1.016***	1.012***
Data-rich (H_t^+) models										
ARDI,BIC	**0.946***	**0.991**	1,037	**1.004**	**0.968**	**0.801*****	**0.807*****	0.887**	0.833***	0.784***
ARDI,AIC	**0.959***	**0.968**	**1.017**	**0.998**	**0.943**	0.840***	**0.803*****	0.844**	**0.798****	**0.768*****
ARDI,POOS-CV	0,994	**1.015**	**0.984**	**0.968**	**0.966**	0.896***	**0.698*****	**0.773*****	**0.777*****	0.812***
ARDI,K-fold	**0.940***	**0.977**	**1.013**	**0.982**	**0.912***	**0.787*****	**0.812*****	0.841**	**0.808****	**0.762*****
RRARDI,POOS-CV	**0.994**	**1.032**	**0.987**	**0.973**	**0.948**	0.908**	**0.725*****	0.793***	**0.778*****	0.861**
RRARDI,K-fold	**0.943****	**0.977**	**0.986**	**0.990**	**0.921**	0.847**	**0.718*****	**0.794*****	**0.796*****	**0.702*****
RFARDI,POOS-CV	**0.948****	**0.991**	**0.951**	**0.919***	**0.899****	0.865**	**0.802*****	0.837***	**0.782*****	0.819***
RFARDI,K-fold	**0.953****	**1.016**	**0.957**	**0.924***	**0.890****	0.889***	**0.864***	0.846***	**0.803*****	**0.767*****
KRR-ARDI,POOS-CV	1,038	**1.016**	**0.921***	**0.934**	**0.959**	1.152*	1,021	0.847***	0.814***	0.886**
KRR,ARDI,K-fold	**0.971**	**0.983**	**0.923***	**0.914***	**0.959**	1,006	0,983	0.827***	**0.793*****	0.848***
$(B_1, \alpha = \hat{\alpha})$,POOS-CV	1,014	**1.001**	**1.023**	**0.996**	**0.946**	1,067	0,956	0,979	0.916**	0.855***
$(B_1, \alpha = \hat{\alpha})$,K-fold	**0.957****	**0.952**	**1.029**	1,046	1,051	0.908**	0.856***	0.874**	**0.816*****	0.890*
$(B_1, \alpha = 1)$,POOS-CV	**0.971***	**1.013**	1.067*	**1.020**	**0.955**	0,991	0,889	1,01	0.935*	0.880**
$(B_1, \alpha = 1)$,K-fold	**0.957****	**0.952**	**1.029**	**1.046**	1,051	0.908**	0.856***	0.874**	**0.816*****	0.890*
$(B_1, \alpha = 0)$,POOS-CV	1,047	1.112**	**1.021**	1,051	**0.969**	1.134*	1.182**	0,997	1,005	0.821***
$(B_1, \alpha = 0)$,K-fold	1,025	1.056*	1,065	1,082	1,052	1,032	0,974	0,923	0,929	0.847***
$(B_2, \alpha = \hat{\alpha})$,POOS-CV	1,061	**0.968**	**0.975**	**0.999**	**0.923****	1,237	0.810***	0.889***	0.904**	0.869**
$(B_2, \alpha = \hat{\alpha})$,K-fold	1,098	**0.949**	**0.993**	**0.974**	**0.970**	1,332	**0.801*****	0.896**	0.851***	**0.756*****
$(B_2, \alpha = 1)$,POOS-CV	**0.973**	1,045	**1.012**	**1.023**	**0.920****	1,034	1,033	0,997	0,957	0.839***
$(B_2, \alpha = 1)$,K-fold	**0.956****	1,022	1,032	**1.025**	**0.990**	0,961	0,935	0,959	0.913**	0.809***
$(B_2, \alpha = 0)$,POOS-CV	**0.933*****	**0.955**	**0.972**	**0.937**	**0.913****	0.902**	**0.781*****	0.904**	0.840***	0.807***
$(B_2, \alpha = 0)$,K-fold	**0.937****	**0.927****	**0.961**	**0.927**	**0.959**	0.871***	**0.787*****	0.858***	**0.775*****	0.776***
$(B_3, \alpha = \hat{\alpha})$,POOS-CV	**0.980**	**0.994**	**1.016**	1,05	**0.952**	1,032	0,95	0,957	0,97	0.861***
$(B_3, \alpha = \hat{\alpha})$,K-fold	**0.973****	**0.946****	1,042	**0.948**	**0.997**	1,016	0.916**	0,938	0.825***	0.827***
$(B_3, \alpha = 1)$,POOS-CV	**0.969***	1,053	1,053	1.080*	**0.956**	0,972	0,946	1,002	1,014	0.906**
$(B_3, \alpha = 1)$,K-fold	**0.946*****	**0.913****	**0.994**	**0.976**	1,01	0.924**	0.829***	0.888*	**0.803*****	0.822***
$(B_3, \alpha = 0)$,POOS-CV	**0.976**	1,049	1,04	1,063	**0.973**	1,034	1,061	0,997	0.932*	0.846***
$(B_3, \alpha = 0)$,K-fold	0,981	1,01	1,03	**1.011**	**0.985**	1,002	0,997	0,95	**0.826*****	0.787***
SVR-ARDI,Lin,POOS-CV	**0.989**	1.165**	1.216**	1.193**	**1.034**	0.915*	0.900**	1,006	0.862**	**0.778*****
SVR-ARDI,Lin,K-fold	1.109**	1.367***	**1.024**	**1.038**	**1.028**	1,129	1,133	**0.776*****	**0.808*****	**0.726*****
SVR-ARDI,RBF,POOS-CV	**0.968***	**0.986**	1.100*	**0.960**	**0.936***	0,958	0.900*	0.873**	**0.760*****	0.820***
SVR-ARDI,RBF,K-fold	**0.951***	**0.946**	**0.993**	**0.952**	**1.001**	0.860**	**0.793*****	**0.806*****	**0.777*****	0.791***

*Note: The numbers represent the relative, with respect to AR,BIC model, root MSPE. Models retained in model confidence set are in bold, the minimum values are underlined, while ***, **, * stand for 1%, 5% and 10% significance of Diebold-Mariano test.*

Table 4: Unemployment rate: Relative Root MSPE

	Full Out-of-Sample					NBER Recessions Periods				
Models	h=1	h=3	h=9	h=12	h=24	h=1	h=3	h=9	h=12	h=24
Data-poor (H_t^-) models										
AR,BIC (RMSPE)	1,9578	1,1905	1,0169	1,0058	0,869	2,5318	2,0826	1,8823	1,7276	1,0562
AR,AIC	0,991	0,984	0,988	0.993***	1	0,958	0.960**	0.984*	1	1
AR,POOS-CV	0,988	0,999	1,002	0,995	0,987	0,978	0.980**	0,996	0,998	1,04
AR,K-fold	0,994	0,984	0,989	0.986***	0,991	0.956*	0.960**	0,998	1	1,038
RRAR,POOS-CV	0,989	1	1,002	0.990*	0.972**	0,984	0.988*	0,997	0.991*	1,001
RRAR,K-fold	0,988	0.982*	0.983*	0.989**	0,999	0,963	0.971*	0,992	0,995	1,033
RFAR,POOS-CV	0,983	0,995	0,968	1	1,002	0,989	1,003	0.929**	0.951**	0,994
RFAR,K-fold	0,98	0,985	0,979	1,006	0,99	0,985	0,972	0.896***	0.943*	0,983
KRR-AR,POOS-CV	0,99	1,04	**0.882*****	**0.889*****	0.876***	1,04	1,116	0.843***	0.883***	0.904**
KRR,AR,K-fold	**0.940*****	**0.910*****	**0.878*****	**0.869*****	0.852***	0.847***	0.838***	**0.788*****	**0.798*****	0.908**
SVR-AR,Lin,POOS-CV	1,028	1.133**	1.130***	1.108***	1.174***	1.065*	1.274***	1.137***	1.094***	1.185***
SVR-AR,Lin,K-fold	0,993	1.061**	1.068***	1.045***	1.013***	1.062**	1.108***	1.032**	1,011	1.018***
SVR-AR,RBF,POOS-CV	1,019	1.094*	1,029	1.076**	1,01	1.097**	1.247**	1.047*	1.034***	1.112*
SVR-AR,RBF,K-fold	0,997	1,011	1.078**	1.053*	0,993	1,026	1,009	1,058	1,023	0,985
Data-rich (H_t^+) models										
ARDI,BIC	**0.937****	**0.893****	0,938	0,939	0.875***	**0.690*****	0.715***	**0.798*****	0.782***	**0.783*****
ARDI,AIC	**0.933****	**0.878*****	0,928	0,953	0.893**	**0.720*****	0.719***	**0.798*****	0.799***	**0.787*****
ARDI,POOS-CV	**0.924*****	**0.913***	0,957	0.925*	**0.856*****	**0.686*****	**0.676*****	0.840**	**0.737*****	**0.777*****
ARDI,K-fold	**0.935****	**0.895****	0,929	0,93	0.915**	**0.696*****	0.697***	**0.801*****	0.807***	**0.787*****
RRARDI,POOS-CV	**0.924*****	**0.896***	0,968	0,946	0.870***	**0.711*****	**0.635*****	0.849**	**0.768*****	**0.767*****
RRARDI,K-fold	**0.940****	**0.899****	0,946	0.931*	0.908**	**0.755****	**0.681*****	**0.803*****	0.790***	**0.753*****
RFARDI,POOS-CV	**0.934*****	0,945	**0.857*****	**0.842*****	**0.763*****	**0.724*****	0.769***	**0.718*****	**0.734*****	**0.722*****
RFARDI,K-fold	**0.932*****	**0.897*****	**0.873****	**0.854*****	**0.785*****	**0.749*****	0.742***	**0.731*****	**0.720*****	**0.710*****
KRR-ARDI,POOS-CV	**0.959***	**0.961**	**0.839*****	**0.813*****	**0.804*****	1,01	1,017	**0.748*****	**0.732*****	0.828***
KRR,ARDI,K-fold	**0.938*****	**0.907****	**0.827*****	**0.817*****	**0.795*****	0,925	0,933	**0.785*****	**0.729*****	**0.814*****
$(B_1, \alpha = \hat{\alpha})$,POOS-CV	0,979	**0.945**	0,976	0,953	0.913***	1,049	0.899*	0,933	0.910*	0.871***
$(B_1, \alpha = \hat{\alpha})$,K-fold	0,971	**0.925****	**0.867*****	0.919*	0.925*	0.787***	0.848***	0.840***	0.839***	0.829**
$(B_1, \alpha = 1)$,POOS-CV	0.947***	**0.937***	0,962	0.922**	0.889***	0.857**	0.789***	0.888**	0.860***	0.915*
$(B_1, \alpha = 1)$,K-fold	0,971	**0.925****	**0.867*****	0.919*	0.925*	0.787***	0.848***	0.840***	0.839***	0.829**
$(B_1, \alpha = 0)$,POOS-CV	1.238**	1.319**	1,021	1,07	1,01	1.393*	1.476*	0,979	0,972	**0.764*****
$(B_1, \alpha = 0)$,K-fold	1.246**	0,994	1.062*	1.077*	1,018	1,322	0,963	0,991	0,933	0.802***
$(B_2, \alpha = \hat{\alpha})$,POOS-CV	**0.907*****	**0.918****	0.926*	0.936*	0.911**	**0.756*****	0.767***	0.869**	0.832***	0.808***
$(B_2, \alpha = \hat{\alpha})$,K-fold	**0.917*****	**0.900*****	0.915*	0,931	0,974	**0.728*****	0.777***	0.829***	**0.738*****	**0.713*****
$(B_2, \alpha = 1)$,POOS-CV	**0.914*****	0,955	1,057	1,011	0.883***	0.810***	0.830***	1,029	0,952	0.795***
$(B_2, \alpha = 1)$,K-fold	0,97	**0.901****	0,991	0,983	0.918**	0.837**	0.754***	0,903	0.833***	**0.753*****
$(B_2, \alpha = 0)$,POOS-CV	**0.908*****	**0.893*****	0,991	0.922*	0.889***	0.781**	0.769***	0,915	0.786***	**0.788*****
$(B_2, \alpha = 0)$,K-fold	**0.949****	**0.898*****	0.908**	0.906**	0,967	0,875	0.777***	0.817***	**0.756*****	**0.741*****
$(B_3, \alpha = \hat{\alpha})$,POOS-CV	**0.949****	**0.888*****	0,952	0,943	0.874***	0,933	0.843***	0.886**	0.829***	0.827***
$(B_3, \alpha = \hat{\alpha})$,K-fold	**0.937****	**0.910*****	**0.882****	0.923*	0.921**	0.836*	0.831***	0.868***	0.839***	**0.795*****
$(B_3, \alpha = 1)$,POOS-CV	**0.929*****	**0.921****	0,958	0,983	0.884***	0.812**	0.771***	0.864**	0.851**	0.845***
$(B_3, \alpha = 1)$,K-fold	0,968	0.941*	**0.861*****	0.907*	0,943	0.808**	0.806***	0.832***	0.873**	**0.736*****
$(B_3, \alpha = 0)$,POOS-CV	**0.948****	0,974	0,994	1,066	0.946*	0,979	1,03	0,956	0.877**	**0.799*****
$(B_3, \alpha = 0)$,K-fold	0,969	**0.918*****	0,983	0,998	0.945*	0,963	0.901*	0,957	0.912*	**0.730*****
SVR-ARDI,Lin,POOS-CV	**0.960***	1,041	1,072	0,929	1,028	0,872	0.858*	0,941	0.809***	**0.779*****
SVR-ARDI,Lin,K-fold	0.959*	**0.873*****	**0.838*****	0,926	0,946	0.801**	0.791***	**0.756*****	**0.800****	0.872*
SVR-ARDI,RBF,POOS-CV	**0.966**	**0.995**	1,016	0,957	0.872***	0,938	0.859*	0,937	0.786***	**0.777****
SVR-ARDI,RBF,K-fold	**0.943****	0,958	**0.871****	0.911*	0.930*	**0.769*****	0.796***	**0.770*****	**0.763*****	**0.787*****

*Note: The numbers represent the relative, with respect to AR,BIC model, root MSPE. Models retained in model confidence set are in bold, the minimum values are underlined, while ***, **, * stand for 1%, 5% and 10% significance of Diebold-Mariano test.*

Table 5: Term spread: Relative Root MSPE

	Full Out-of-Sample					NBER Recessions Periods				
Models	h=1	h=3	h=9	h=12	h=24	h=1	h=3	h=9	h=12	h=24
Data-poor (H_t^-) models										
AR,BIC (RMSPE)	**6.4792**	12,8246	**16.3575**	20,0828	22,2091	**13.3702**	23,16	23,5697	31,597	23,0842
AR,AIC	**1.002***	0,998	**1.053***	1.034**	1.041**	**1.002**	1,001	1,034	0,993	0,972
AR,POOS-CV	**1.055***	1.139*	**1.000**	0,969	1.040**	**1.041**	1,017	0.895*	0.857*	0,972
AR,K-fold	**1.001**	1	**1.003**	0,979	1.038*	**1.002**	0,998	0,911	0.890*	0,983
RRAR,POOS-CV	**1.055****	1.142*	**1.004**	0,998	1,016	**1.036**	1,014	0,899	0,966	0.945**
RRAR,K-fold	**1.044***	0,992	**1.027**	0,96	1,015	**1.024**	0,982	0,959	0.795**	0.957*
RFAR,POOS-CV	**0.997**	0,886	1.125***	1,019	1.107**	**0.906**	**0.816**	1,039	0.747**	1.077**
RFAR,K-fold	**0.991**	0,941	1.136***	1,011	1.084**	**0.909**	0,823	1,023	0.764*	1,038
KRR-AR,POOS-CV	1.223**	0,881	**0.949**	**0.888****	0.945*	**1.083**	**0.702**	**0.788*****	0.758***	0,948
KRR,AR,K-fold	**1.141**	0,983	**1.098****	0,999	1,048	**0.999**	**0.737**	0.833*	**0.663****	**0.924**
SVR-AR,Lin,POOS-CV	1.158**	1.326***	**1.071***	1,045	1,045	1.111*	1,072	0.894*	0.828*	0,967
SVR-AR,Lin,K-fold	1.191**	1,056	**1.018**	0,963	0,993	1,061	1,009	0.886**	0.845**	0.916***
SVR-AR,RBF,POOS-CV	**1.006**	1,039	**1.050***	0,951	0,969	**0.964**	0,902	0.876*	0.761**	**0.864*****
SVR-AR,RBF,K-fold	**0.985**	0,911	**1.038**	0,946	0.933**	**0.990**	**0.737**	0.851**	0.747*	0,968
Data-rich (H_t^+) models										
ARDI,BIC	**0.953**	0,971	**0.979**	0,93	**0.892*****	**0.921**	0,9	**0.790*****	**0.633*****	1,049
ARDI,AIC	**0.970**	0,956	**1.019**	0,944	0.917**	**0.929**	0,867	**0.814*****	**0.647*****	1,076
ARDI,POOS-CV	**0.954**	1,015	**1.067**	0,991	**0.915****	**0.912**	0,92	0,958	0.769**	1,087
ARDI,K-fold	**0.991**	1,026	**1.001**	0,928	0,939	**0.958**	0,967	0.812***	**0.662*****	1,041
RRARDI,POOS-CV	**0.936**	0,994	**1.078**	0,991	0,964	**0.896**	**0.850**	0,952	0.784**	1,092
RRARDI,K-fold	**1.015**	0,992	**1.018**	0,934	0,981	**0.978**	0,899	0.881*	**0.635*****	1.163*
RFARDI,POOS-CV	**0.988**	**0.830***	**0.957**	**0.873****	**0.921****	**0.804**	**0.691**	**0.785*****	**0.606*****	0,985
RFARDI,K-fold	**1.010**	0,883	**0.997**	0,909	0.935**	**0.808**	**0.778**	0.827**	**0.626*****	0,97
KRR-ARDI,POOS-CV	1.355**	0,898	**0.993**	**0.856****	**0.884*****	**0.861**	**0.682***	**0.772*****	**0.621****	**0.905***
KRR,ARDI,K-fold	1.382***	0,96	**0.974**	**0.827****	**0.862*****	**0.858**	**0.684***	**0.754*****	**0.569*****	0.912*
$(B_1, \alpha = \hat{\alpha})$,POOS-CV	1,114	1,06	1.126***	1,021	**0.866*****	**1.009**	0,981	1,02	**0.701****	1,012
$(B_1, \alpha = \hat{\alpha})$,K-fold	**1.089**	1.149**	1.199**	1.106*	0,969	**1.001**	1,041	0,885	0.767**	0,941
$(B_1, \alpha = 1)$,POOS-CV	1.125*	1,115	1.172***	1,072	**0.844*****	1,071	1,006	1,033	0,833	0,96
$(B_1, \alpha = 1)$,K-fold	**1.089**	1.149**	1.199**	1.106*	0,969	**1.001**	1,041	0,885	0.767**	0,941
$(B_1, \alpha = 0)$,POOS-CV	1.173**	1.312**	1.176***	1,088	0,978	1,089	1,065	0,981	0,799	0,966
$(B_1, \alpha = 0)$,K-fold	1.163*	1,059	**1.069**	0,929	**0.921****	**1.041**	0,869	**0.810****	0.729**	**0.880***
$(B_2, \alpha = \hat{\alpha})$,POOS-CV	**1.025**	0,993	**1.101****	1,028	**0.897*****	**0.918**	0,908	1,02	**0.651*****	0,989
$(B_2, \alpha = \hat{\alpha})$,K-fold	**0.976**	0,954	**1.098***	1,059	0.935*	**0.931**	0,875	0,938	0.779*	0,952
$(B_2, \alpha = 1)$,POOS-CV	1,062	0,968	**1.125****	1,049	0.926***	**0.897**	0,855	1,058	0,79	1,001
$(B_2, \alpha = 1)$,K-fold	**0.980**	0,938	**1.130****	1,01	0.950*	**0.948**	0,858	0,976	0.679**	1,001
$(B_2, \alpha = 0)$,POOS-CV	1.118*	1,082	**1.097****	1,008	**0.901*****	**1.004**	0,919	1,008	**0.669*****	1,016
$(B_2, \alpha = 0)$,K-fold	1,102	0,988	**1.047**	1,041	**0.919****	**0.985**	0,909	0.870*	0.757*	0,986
$(B_3, \alpha = \hat{\alpha})$,POOS-CV	**0.971**	0,964	**1.089****	1,076	0.933*	**0.887**	**0.837**	0,908	0.783*	0.904**
$(B_3, \alpha = \hat{\alpha})$,K-fold	**0.968**	0,944	**1.009**	0,999	**0.898*****	**0.895**	0,872	0.883**	0.744**	**0.907*****
$(B_3, \alpha = 1)$,POOS-CV	**1.006**	1,066	**1.059***	1,039	**0.896*****	**0.894**	1,131	0,974	0.764*	0,987
$(B_3, \alpha = 1)$,K-fold	**0.994**	0,924	**1.037**	0,96	0,975	**0.934**	0,852	0.834**	0.712**	1,01
$(B_3, \alpha = 0)$,POOS-CV	1.181*	0,961	**1.104****	1,056	0.937**	1,215	0,901	1,013	0,825	0.919*
$(B_3, \alpha = 0)$,K-fold	**0.999**	0,953	**1.036**	0,94	0,97	**0.897**	0,845	0,923	0.735**	0.925**
SVR-ARDI,Lin,POOS-CV	**1.062**	0,967	**1.164****	1.113*	1,065	1,016	**0.762***	1,117	0.714**	1,097
SVR-ARDI,Lin,K-fold	**0.990**	0,98	**1.011**	**0.922**	**0.909****	**0.935**	0,885	0.825**	**0.667****	0,994
SVR-ARDI,RBF,POOS-CV	**0.972**	0,937	**1.069**	1,039	1,068	**0.875**	**0.741**	**0.796*****	0.707***	1.204*
SVR-ARDI,RBF,K-fold	**1.018**	0,938	**1.123**	**0.914***	**0.882*****	**0.931**	**0.781**	0.858**	0.778**	**0.858****

*Note: The numbers represent the relative, with respect to AR,BIC model, root MSPE. Models retained in model confidence set are in bold, the minimum values are underlined, while ***, **, * stand for 1%, 5% and 10% significance of Diebold-Mariano test.*

Table 6: CPI Inflation: Relative Root MSPE

	Full Out-of-Sample					NBER Recessions Periods				
Models	h=1	h=3	h=9	h=12	h=24	h=1	h=3	h=9	h=12	h=24
Data-poor (H_t^-) models										
AR,BIC (RMSPE)	0,0312	0,0257	0,0194	0,0187	0,0188	0,0556	0,0484	0,032	0,0277	0,0221
AR,AIC	0.969***	0,984	0.976*	0,988	0,995	1	0.970**	0,999	0,992	1,005
AR,POOS-CV	0.966**	0,988	0,997	0,992	1,009	0.961**	0,981	0,995	0,978	1,003
AR,K-fold	0.972**	0.976**	0.975*	0,988	0,987	1,002	0.965***	0,998	0,992	1,005
RRAR,POOS-CV	0.969**	0,984	0,99	0,993	1,006	0.961**	0,982	0,995	0.963*	0,998
RRAR,K-fold	0.964***	0.979**	0.970*	0.980*	0,989	0,989	0.973**	0,996	0,992	0,997
RFAR,POOS-CV	0,983	**0.944***	**0.909***	**0.930**	1,022	1,018	0,998	1,063	1,047	0,998
RFAR,K-fold	**0.975**	**0.927****	**0.909***	**0.956**	0,998	1,032	0,972	1,065	1,103	1,019
KRR-AR,POOS-CV	**0.972**	**0.905****	**0.872****	**0.872****	0.907**	1,023	**0.930****	0,927	0,91	0.852*
KRR,AR,K-fold	**0.931*****	**0.888*****	**0.836****	**0.827*****	0,942	0,965	**0.920****	0,92	0,915	0,975
SVR-AR,Lin,POOS-CV	1.119**	1.291**	1.210***	1.438***	1.417***	1,116	1.196**	1.204**	1,055	1.613***
SVR-AR,Lin,K-fold	1.239***	1.369**	1.518***	1.606***	1.411***	1.159*	1.326*	1.459**	1.501*	1,016
SVR-AR,RBF,POOS-CV	0,988	1,004	1.086*	1.068**	1.127**	0,999	1,004	0,969	1.091**	1.501***
SVR-AR,RBF,K-fold	0,99	1,025	1,025	1,003	1.370***	0,965	0,979	0,996	0.896**	1.553**
Data-rich (H_t^+) models										
ARDI,BIC	0,96	**0.973**	1,024	**0.895***	0.880*	0.919*	**0.906***	0.779*	0.755**	0.713**
ARDI,AIC	**0.954**	**0.990**	1,034	**0.895**	0,884	0,925	**0.898**	0.778*	**0.736****	0.676**
ARDI,POOS-CV	**0.950**	**0.984**	1,017	**0.910**	0,916	0.916*	**0.913***	0.832**	0.781***	**0.669****
ARDI,K-fold	**0.941***	**0.990**	1,028	**0.873***	**0.858***	0.891**	**0.900**	0.784*	**0.709*****	**0.635****
RRARDI,POOS-CV	**0.943***	**0.975**	1,001	**0.917**	0,914	0.905*	**0.912***	0.828**	**0.780*****	**0.666****
RRARDI,K-fold	**0.943****	**0.983**	1,022	**0.875***	**0.882**	0.927*	**0.901**	**0.744****	**0.664*****	**0.613****
RFARDI,POOS-CV	**0.947****	**0.908*****	**0.853****	**0.914***	0,979	0,976	**0.939****	0,988	1,051	0,964
RFARDI,K-fold	**0.936*****	**0.907*****	**0.854****	**0.868****	0.909*	0,962	**0.933****	0,979	0,93	1,003
KRR-ARDI,POOS-CV	1,006	1,043	0,959	0,972	1,067	1,046	1,093	0,952	0,948	0,946
KRR,ARDI,K-fold	**0.985**	0,999	0,983	0,977	0,938	0,998	0,99	1,023	1,022	0,986
$(B_1, \alpha = \hat{\alpha})$,POOS-CV	**0.918****	**0.916***	0,976	0,96	1,026	**0.803*****	**0.900***	0,8	0,848	0,974
$(B_1, \alpha = \hat{\alpha})$,K-fold	**0.908****	**0.921***	1,012	1,056	1.092*	**0.823****	**0.873***	**0.774**	0,836	1,069
$(B_1, \alpha = 1)$,POOS-CV	**0.960**	**0.908****	1,11	1,03	1,076	**0.813****	**0.889***	0,794	0,825	0,989
$(B_1, \alpha = 1)$,K-fold	**0.908****	**0.921***	1,012	1,056	1.092*	**0.823****	**0.873***	**0.774**	0,836	1,069
$(B_1, \alpha = 0)$,POOS-CV	0,971	1,035	1.114*	1,048	1.263**	0.848**	**0.906**	0,935	0,881	0,99
$(B_1, \alpha = 0)$,K-fold	**0.945***	1,057	1.246**	1.289**	1.260***	**0.850*****	**0.939**	0,954	0,944	1,095
$(B_2, \alpha = \hat{\alpha})$,POOS-CV	**0.923****	**0.956****	**0.940**	0,934	0,945	0.871*	0,959	0.803*	0.802*	0.822*
$(B_2, \alpha = \hat{\alpha})$,K-fold	**0.921****	**0.963***	0,995	0,956	1,037	0.868*	0.957*	0.817*	0.778**	0,861
$(B_2, \alpha = 1)$,POOS-CV	**0.942**	**0.959**	1.158*	1.174**	1.151**	0,877	0,927	0,799	0,907	1,087
$(B_2, \alpha = 1)$,K-fold	**0.922****	**0.970**	1,066	0,995	1.168*	0,879	0,929	0,853	0.816*	1,009
$(B_2, \alpha = 0)$,POOS-CV	**0.921****	**0.940**	1,079	0,959	1,071	0.857*	**0.881**	1,129	0,883	0,851
$(B_2, \alpha = 0)$,K-fold	**0.919****	**0.929***	0,997	1,011	1.212**	0.865*	**0.883**	0,825	0,961	0,853
$(B_3, \alpha = \hat{\alpha})$,POOS-CV	**0.935***	**0.941*****	**0.961**	**0.849****	0.901*	0.889*	**0.947****	0.791**	0.785**	0.808**
$(B_3, \alpha = \hat{\alpha})$,K-fold	**0.938***	**0.952****	**0.937**	**0.915**	0,952	0.891*	0.958*	0.801*	0.784**	0,91
$(B_3, \alpha = 1)$,POOS-CV	**0.933***	**0.960**	1,076	1	1,017	**0.856***	**0.917***	**0.755***	**0.769****	0,86
$(B_3, \alpha = 1)$,K-fold	**0.943**	0,978	1,006	**0.894**	1,002	0,889	0,946	0,805	0.806*	0,879
$(B_3, \alpha = 0)$,POOS-CV	**0.946***	**0.939****	**0.896***	**0.871****	1,022	0.894*	**0.931****	0,865	0,875	0,896
$(B_3, \alpha = 0)$,K-fold	**0.921*****	**0.975**	**0.926**	**0.920**	1,106	0.877***	**0.936**	0,839	0,892	1,147
SVR-ARDI,Lin,POOS-CV	1.148***	1.202*	1.251***	1.209***	1.219**	1,068	1,053	0,969	0,969	0,943
SVR-ARDI,Lin,K-fold	1.115***	1.390**	1.197**	1,114	1.177*	1,058	1.295*	0,944	0,954	1,036
SVR-ARDI,RBF,POOS-CV	0,963	1,031	1,002	0,962	0,951	0,922	**0.915**	0,848	0,861	0,996
SVR-ARDI,RBF,K-fold	**0.951****	1,002	0,997	0,945	**0.797*****	0.927*	0,964	0.816**	0.826**	**0.659****

*Note: The numbers represent the relative, with respect to AR,BIC model, root MSPE. Models retained in model confidence set are in bold, the minimum values are underlined, while ***, **, * stand for 1%, 5% and 10% significance of Diebold-Mariano test.*

Table 7: Housing starts: Relative Root MSPE

	Full Out-of-Sample					NBER Recessions Periods				
Models	h=1	h=3	h=9	h=12	h=24	h=1	h=3	h=9	h=12	h=24
Data-poor (H_t^-) models										
AR,BIC (RMSPE)	**0.9040**	**0.4142**	**0.2499**	0,2198	0,1671	**1.2526**	0,6658	0,4897	0,4158	0,2954
AR,AIC	**0.998**	**1.019**	**1.000**	**1.000**	1	1,01	0.965*	1	1	1
AR,POOS-CV	**1.001**	**1.012**	**1.019***	1,01	1.036**	1,015	**0.936****	1.011*	1,013	1.057**
AR,K-fold	**0.993**	**1.017**	**1.001**	1	1,02	1,01	**0.951****	1	1	1,036
RRAR,POOS-CV	**1.007**	**1.007**	**1.008**	1,009	1.031**	1.027*	**0.939****	1,001	1,013	1.050**
RRAR,K-fold	**0.999**	**1.014**	**0.998**	**0.998**	1.024*	1,013	**0.941****	1.000**	0,999	1.042**
RFAR,POOS-CV	1.030***	**1.026***	**1.028***	1.045**	1,018	1,023	**0.941***	**0.992**	1.048*	1,013
RFAR,K-fold	1.017*	**1.022**	**1.007**	1.031**	1,008	1,02	**0.942***	**0.990**	1,026	1,01
KRR-AR,POOS-CV	**0.995**	**0.999**	**0.969***	1.044*	1.037*	**0.990**	**0.972**	**0.971**	1.050**	0,993
KRR,AR,K-fold	**0.977***	**0.975**	**0.957****	**0.989**	1,001	**0.985**	**0.976**	1,01	1,006	1,004
SVR-AR,Lin,POOS-CV	1.032***	**0.997**	**1.044*****	1.064***	1.223**	1.024*	**0.962***	**0.986***	0,984	**0.957*****
SVR-AR,Lin,K-fold	1.036***	1,031	**1.002**	1,006	1,002	1,013	0,976	1,002	1,009	1,004
SVR-AR,RBF,POOS-CV	**1.008**	1.047**	**1.023**	1.035***	1.060***	**1.014**	0,981	**0.947*****	1,015	1,017
SVR-AR,RBF,K-fold	1,009	**1.011**	**1.012****	1.020***	1.034**	1.021*	0.969*	1.010***	1.017**	1,001
Data-rich (H_t^+) models										
ARDI,BIC	**0.973***	**0.989**	**1.031**	1,051	1,05	**0.946**	1,139	1,048	0,988	0,944
ARDI,AIC	**0.992**	**0.995**	**1.018**	1,06	1,078	**1.000**	1,113	1,025	1,025	0,96
ARDI,POOS-CV	1,01	**1.007**	**1.080**	1,027	0,998	1,023	1,128	1,054	1,015	1,021
ARDI,K-fold	**0.992**	**0.984**	**1.026**	1,061	1,094	**1.011**	1,093	1,027	1,027	0,958
RRARDI,POOS-CV	**0.998**	**1.007**	**1.043**	**0.996**	1,082	1,008	1,119	1,041	0,991	1,022
RRARDI,K-fold	**0.998**	**0.988**	**1.051**	1,064	1,089	1,017	1,118	1,033	0,998	0,941
RFARDI,POOS-CV	**0.997**	**0.944****	**0.930****	**0.920***	0.899**	**0.982**	**0.971**	**0.965**	0,957	0,972
RFARDI,K-fold	**0.994**	**0.962**	**0.939***	**0.914***	**0.838*****	**0.993**	0,985	**0.986**	**0.943**	0.902*
KRR-ARDI,POOS-CV	**0.980**	**0.943*****	**0.915****	**0.942****	**0.884*****	**0.941***	**0.952***	**0.949**	0.964**	0,986
KRR,ARDI,K-fold	**0.982****	**0.949****	**0.928**	**0.933**	**0.889****	**0.973**	**0.973**	**1.003**	1,022	0,994
$(B_1, \alpha = \hat{\alpha})$,POOS-CV	**1.006**	**1.000**	**1.063**	1,016	**0.895****	1,023	1,099	**0.985**	1,026	1,022
$(B_1, \alpha = \hat{\alpha})$,K-fold	1.040*	1.095**	1.250**	1.335**	1.151*	1.096*	1.152**	1,021	1,127	**0.890**
$(B_1, \alpha = 1)$,POOS-CV	1.032**	1,039	1,155	1,045	0,949	1,013	1,063	**0.961**	1,025	1,062
$(B_1, \alpha = 1)$,K-fold	1.040*	1.095**	1.250**	1.335**	1.151*	1.096*	1.152**	1,021	1,127	**0.890**
$(B_1, \alpha = 0)$,POOS-CV	**0.982**	**0.977**	1,084	1.337**	0,959	**0.999**	1,017	1,014	1.152**	0,964
$(B_1, \alpha = 0)$,K-fold	**0.982**	**1.006**	1.137*	1.158**	1,007	**0.994**	1,03	**1.017**	1,067	**0.809****
$(B_2, \alpha = \hat{\alpha})$,POOS-CV	1,044	**0.992**	**0.975**	0,988	0,969	1,177	1.126*	1,034	0,989	0,972
$(B_2, \alpha = \hat{\alpha})$,K-fold	**0.988**	**1.003**	**1.069**	1.193**	1,069	1,11	1.188*	1,085	1.133*	0,917
$(B_2, \alpha = 1)$,POOS-CV	**1.001**	**1.000**	**0.967**	1,02	0.940*	**0.961**	1,047	**0.943**	0,985	1,006
$(B_2, \alpha = 1)$,K-fold	**0.989**	1,095	1.245**	1.203*	1,093	1,007	1.322***	1,1	**0.919**	**0.848****
$(B_2, \alpha = 0)$,POOS-CV	1.091*	**0.949**	**0.987**	**0.971**	0,939	1,255	1,027	**0.992**	**0.956**	0,994
$(B_2, \alpha = 0)$,K-fold	1,066	**1.068**	1,19	1,044	1,064	1,248	1.332**	1,057	**0.896*****	0,917
$(B_3, \alpha = \hat{\alpha})$,POOS-CV	1,009	**0.951***	**0.935**	0,99	**0.891****	1,028	1,019	**0.958**	**0.963**	0,987
$(B_3, \alpha = \hat{\alpha})$,K-fold	**0.998**	**0.977**	**1.007**	1,055	1,044	1,019	1,115	1,017	**0.979**	**0.882***
$(B_3, \alpha = 1)$,POOS-CV	**0.997**	**0.975**	**1.024**	0,996	0.928*	**0.976**	1,001	1,021	**0.940**	1,001
$(B_3, \alpha = 1)$,K-fold	1,013	**1.040**	1,071	1,106	1,145	1,042	1.219*	1,036	0,992	1,009
$(B_3, \alpha = 0)$,POOS-CV	1.022*	**0.951***	**0.962**	**0.944**	0.932*	1,022	0,981	**0.930**	**0.915****	1,001
$(B_3, \alpha = 0)$,K-fold	1.030**	**1.003**	**1.005**	1,011	1,029	**0.986**	1,114	**0.998**	**0.955**	0,934
SVR-ARDI,Lin,POOS-CV	**0.998**	1.078*	1.154*	1.137*	1,142	1,047	1,111	**0.989**	1,009	1,111
SVR-ARDI,Lin,K-fold	**0.992**	**0.971**	**1.017**	1,038	1,11	1,007	1,021	**0.988**	**0.937**	0,959
SVR-ARDI,RBF,POOS-CV	**0.991**	**1.004**	**1.010**	1,044	1,034	**0.987**	1,095	**0.981**	0,969	1,096
SVR-ARDI,RBF,K-fold	**1.003**	**0.998**	**1.045**	1,078	1.162*	1,022	1,081	1,03	0,984	1,026

*Note: The numbers represent the relative, with respect to AR,BIC model, root MSPE. Models retained in model confidence set are in bold, the minimum values are underlined, while ***, **, * stand for 1%, 5% and 10% significance of Diebold-Mariano test.*

B Robustness of Treatment Effects Graphs

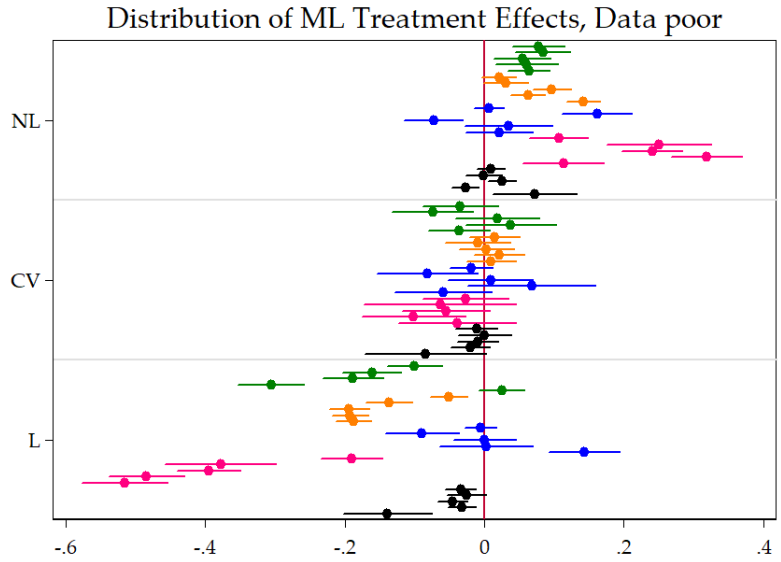

Figure 13: This figure plots the distribution of $\dot{\alpha}_F^{(h,v)}$ from equation 10 done by (h, v) subsets. The subsample under consideration here is **data-poor models**. The unit of the x-axis are improvements in OOS R^2 over the basis model. Variables are INDPRO, UNRATE, SPREAD, INF and HOUST. Within a specific color block, the horizon increases from $h = 1$ to $h = 24$ as we are going down. SEs are HAC. These are the 95% confidence bands.

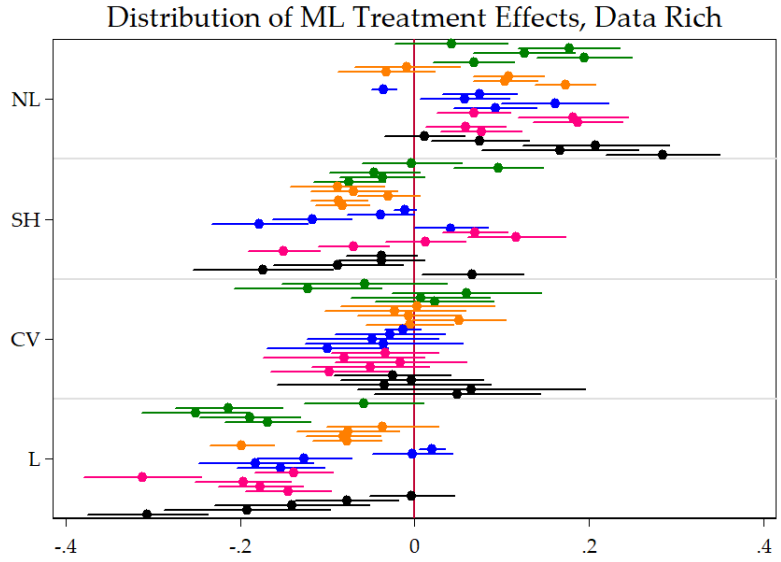

Figure 14: This figure plots the distribution of $\dot{\alpha}_F^{(h,v)}$ from equation 10 done by (h, v) subsets. The subsample under consideration here is **data-rich models**. The unit of the x-axis are improvements in OOS R^2 over the basis model. Variables are INDPRO, UNRATE, SPREAD, INF and HOUST. Within a specific color block, the horizon increases from $h = 1$ to $h = 24$ as we are going down. SEs are HAC. These are the 95% confidence bands.

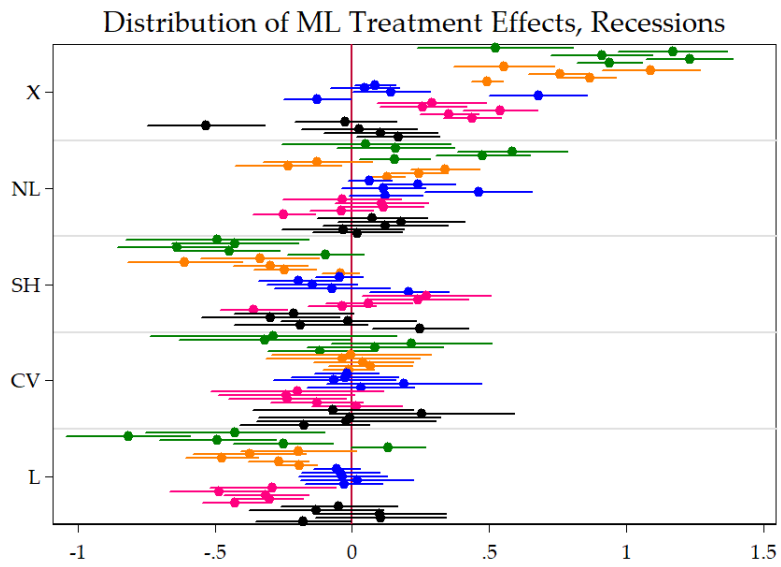

Figure 15: This figure plots the distribution of $\grave{\alpha}_F^{(h,v)}$ from equation 10 done by (h, v) subsets. The subsample under consideration here are **recessions**. The unit of the x-axis are improvements in OOS R^2 over the basis model. Variables are INDPRO, UNRATE, SPREAD, INF and HOUST. Within a specific color block, the horizon increases from $h = 1$ to $h = 24$ as we are going down. SEs are HAC. These are the 95% confidence bands.

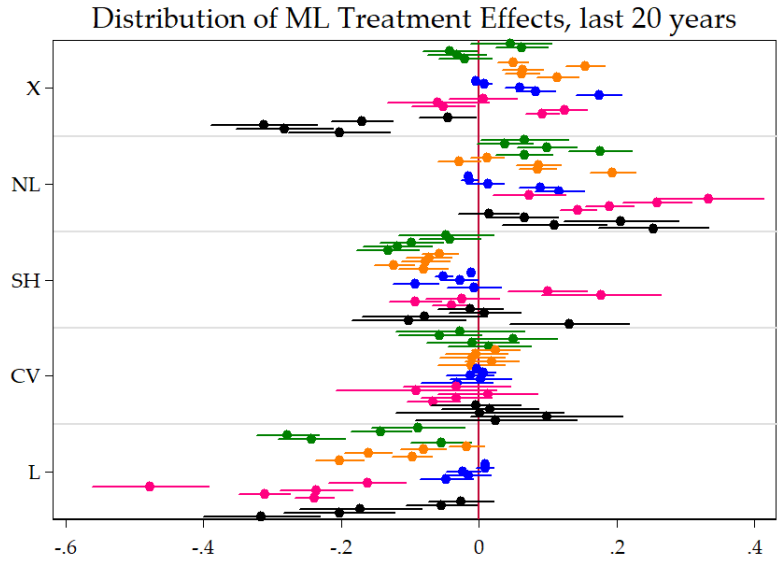

Figure 16: This figure plots the distribution of $\grave{\alpha}_F^{(h,v)}$ from equation 10 done by (h, v) subsets. The subsample under consideration here are **the last 20 years**. The unit of the x-axis are improvements in OOS R^2 over the basis model. Variables are INDPRO, UNRATE, SPREAD, INF and HOUST. Within a specific color block, the horizon increases from $h = 1$ to $h = 24$ as we are going down. SEs are HAC. These are the 95% confidence bands.

C Additional Graphs

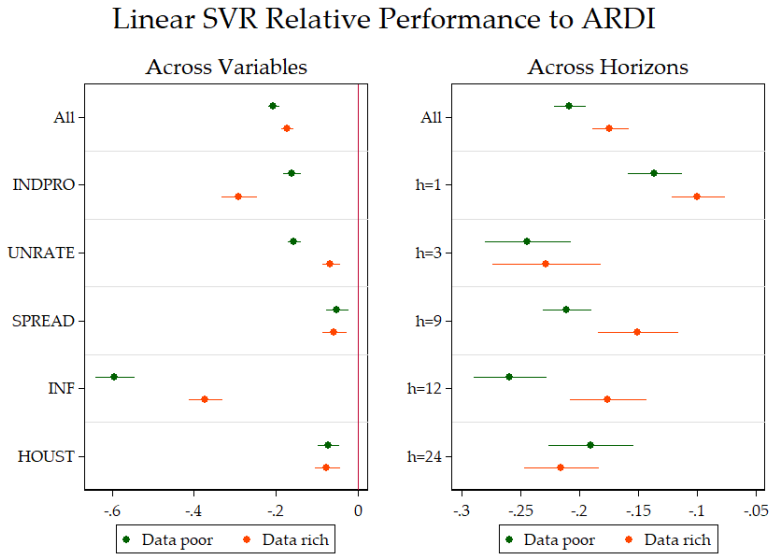

Figure 17: This graph display the marginal (un)improvments by variables and horizons to opt for the SVR in-sample loss function in comparing the data-poor and data-rich environments for linear models. The unit of the x-axis are improvements in OOS R^2 over the basis model. SEs are HAC. These are the 95% confidence bands.

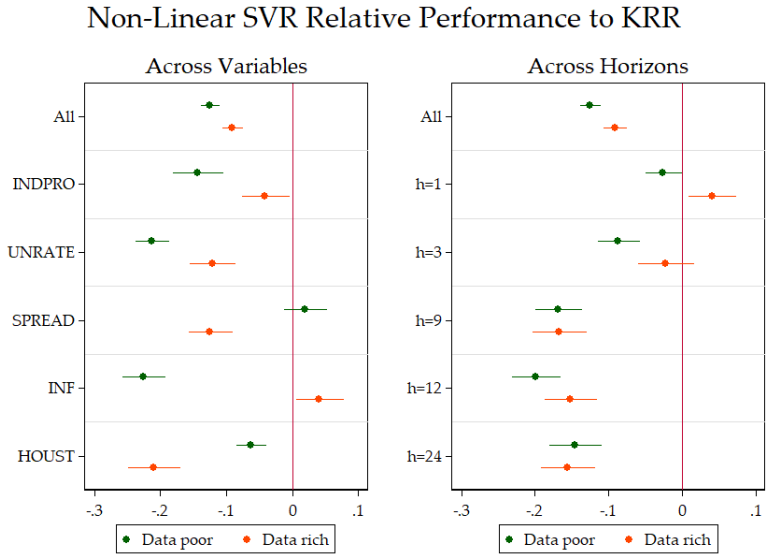

Figure 18: This graph display the marginal (un)improvments by variables and horizons to opt for the SVR in-sample loss function in comparing the data-poor and data-rich environments for non-linear models. The unit of the x-axis are improvements in OOS R^2 over the basis model. SEs are HAC. These are the 95% confidence bands.

D Results with absolute loss

In this section we present results for a different out-of-sample loss function that is often used in the literature: the absolute loss. Following Koenker and Machado (1999), we generate the pseudo-R^1 in order to perform regressions (10) and (11): $R^1_{t,h,v,m} \equiv 1 - \frac{|e_{t,h,v,m}|}{\frac{1}{T}\sum_{t=1}^{T}|y_{v,t+h} - \bar{y}_{v,h}|}$. Hence, the figure included in this section are exact replication of those included in the main text except that the target variable of all the regressions has been changed.

The main message here is that results obtained using the squared loss are very consistent with what one would obtain using the absolute loss. The importance of each feature, figure 19, and the way it behaves according to the variable/horizon pair is the same. Indeed, most of the heterogeneity is variable specific while there are clear horizon patterns emerging when we average out variables. For instance, we clearly see by comparing figures 21 and 3 that more data and non-linearities usefulness increase linearly in h. CV is flat around the 0 line. Alternative shrinkage and loss function both are negative and follow a boomerang shape (they are not as bad for short and very long horizons, but quite bad in between).

The pertinence of non-linearities and the impertinence of alternative shrinkage follow very similar behavior to what is obtained in the main body of this paper. However, for non-linearities, the data-poor advantages are not robust to the choice of MSPE vs MAPE. Fortunately, besides that, the figures are all very much alike.

Results for the alternative in-sample loss function also seem to be independent of the proposed choices of out-of-sample loss function. Only for hyperparameters selection we do get slightly different results: CV-KF is now sometimes worse than BIC in a statistically significant way. However, the negative effect is again much stronger for CV-POOS. CV-KF still outperforms any other model selection criteria on recessions.

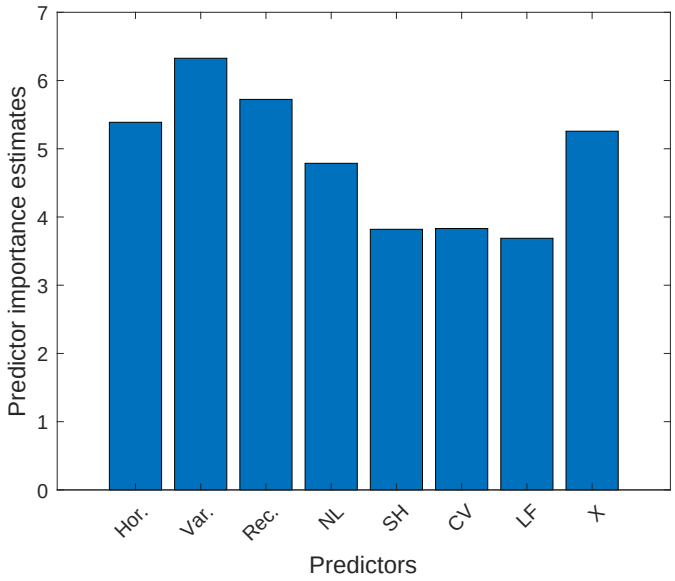

Figure 19: This figure presents predictive importance estimates. Random forest is trained to predict $R^1_{t,h,v,m}$ defined in (10) and use out-of-bags observations to assess the performance of the model and compute features' importance. NL, SH, CV and LF stand for nonlinearity, shrinkage, cross-validation and loss function features respectively. A dummy for H_t^+ models, X, is included as well.

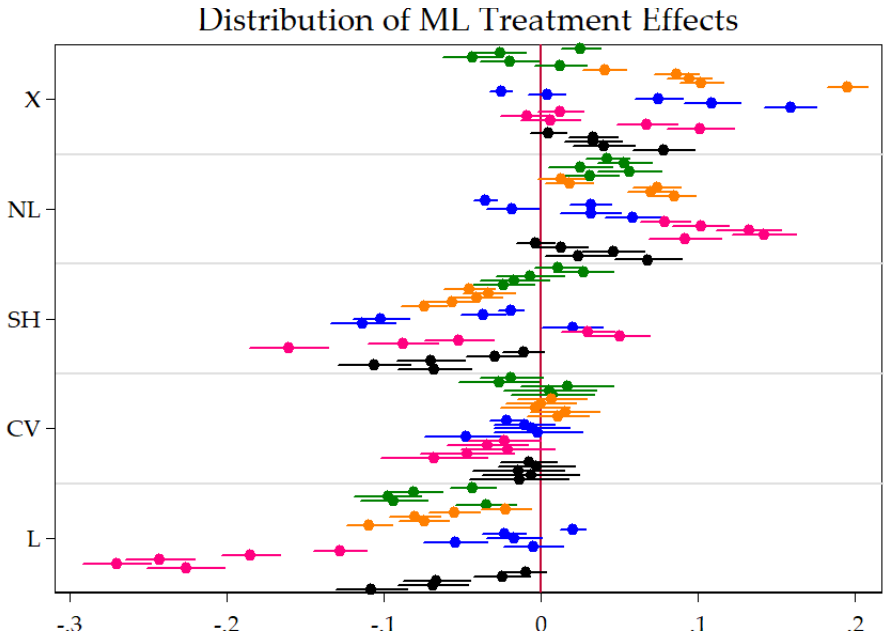

Figure 20: This figure plots the distribution of $\dot{\alpha}_F^{(h,v)}$ from equation (10) done by (h, v) subsets. That is, we are looking at the average partial effect on the pseudo-OOS R^1 from augmenting the model with ML features, keeping everything else fixed. X is making the switch from data-poor to data-rich. Finally, variables are INDPRO, UNRATE, SPREAD, INF and HOUST. Within a specific color block, the horizon increases from $h = 1$ to $h = 24$ as we are going down. As an example, we clearly see that the partial effect of X on the R^1 of INF increases drastically with the forecasted horizon h. SEs are HAC. These are the 95% confidence bands.

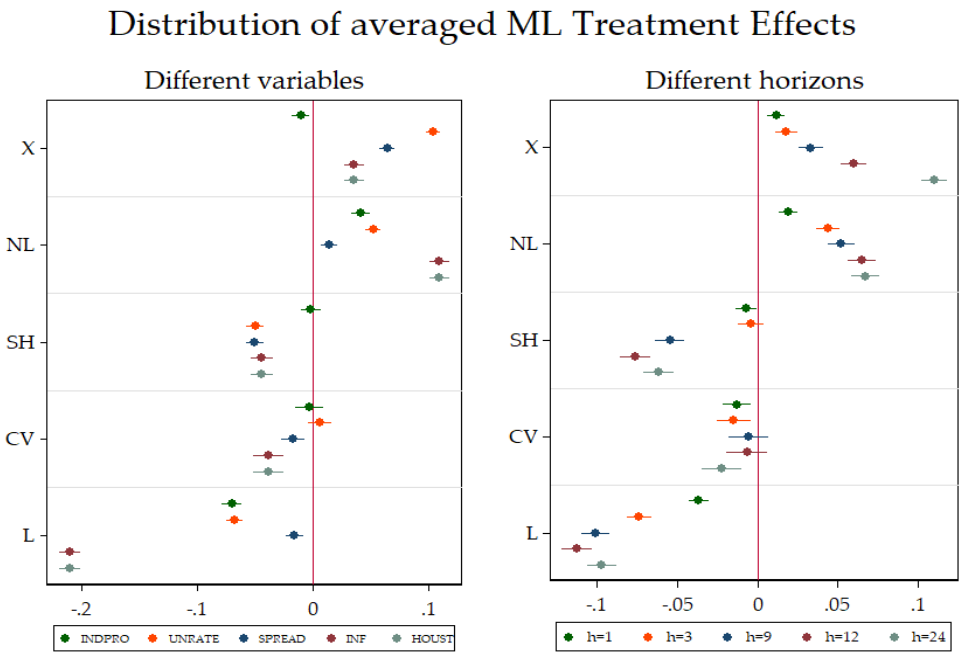

Figure 21: This figure plots the distribution of $\dot{\alpha}_F^{(v)}$ and $\dot{\alpha}_F^{(h)}$ from equation (10) done by h and v subsets. That is, we are looking at the average partial effect on the pseudo-OOS R^1 from augmenting the model with ML features, keeping everything else fixed. X is making the switch from data-poor to data-rich. However, in this graph, $v-$specific heterogeneity and $h-$specific heterogeneity have been integrated out in turns. SEs are HAC. These are the 95% confidence bands.

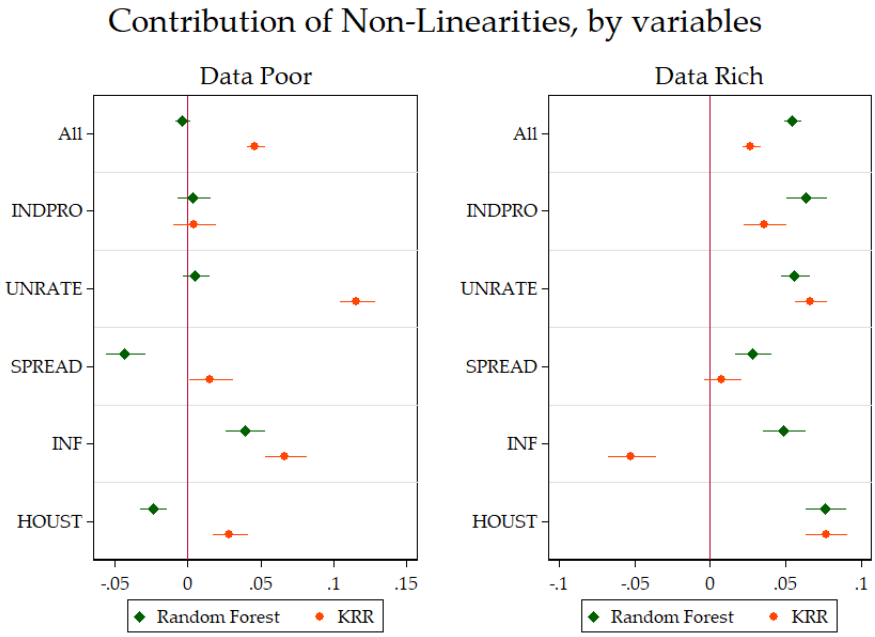

Figure 22: This compares the two NL models averaged over all horizons. The unit of the x-axis are improvements in OOS R^1 over the basis model. SEs are HAC. These are the 95% confidence bands.

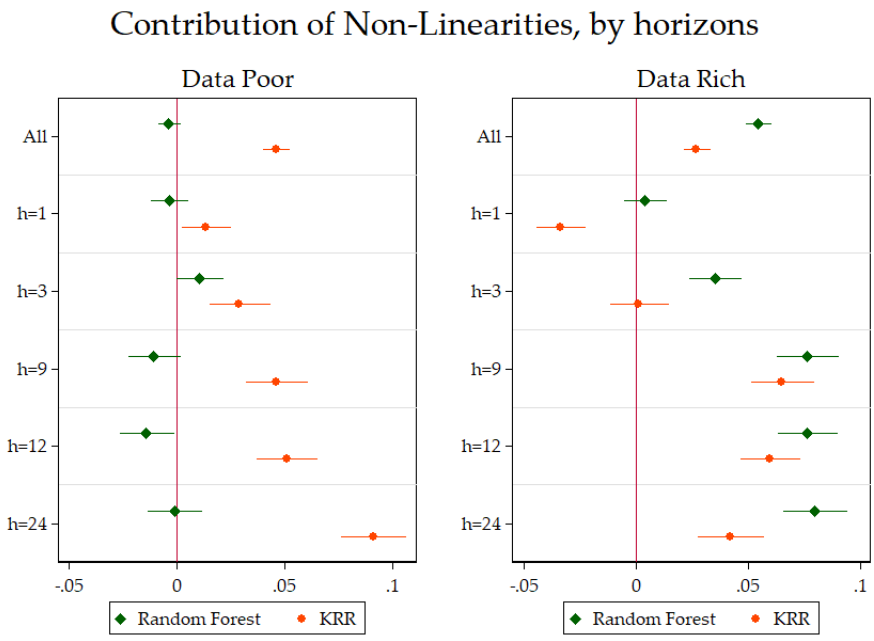

Figure 23: This compares the two NL models averaged over all variables. The unit of the x-axis are improvements in OOS R^1 over the basis model. SEs are HAC. These are the 95% confidence bands.

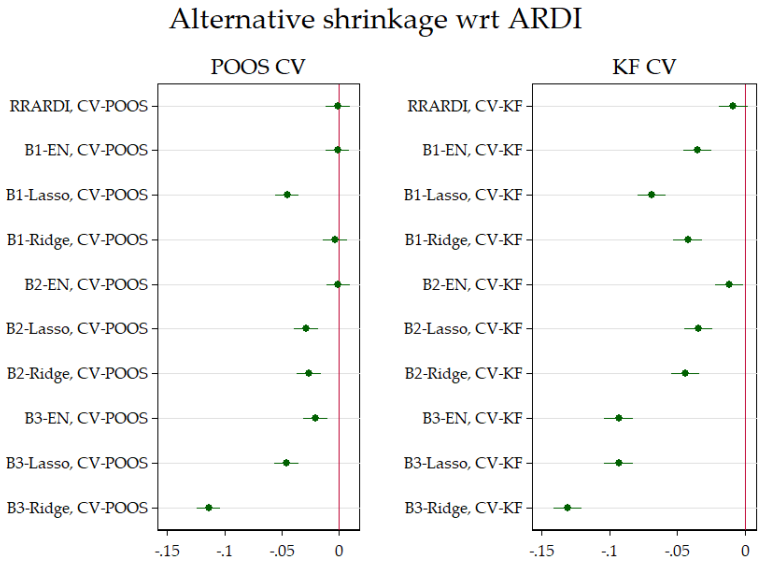

Figure 24: This compares models of section 3.2 averaged over all variables and horizons. The unit of the x-axis are improvements in OOS R^1 over the basis model. The base models are ARDIs specified with POOS-CV and KF-CV respectively. SEs are HAC. These are the 95% confidence bands.

Table 8: CV comparaison

	(1) All	(2) Data-rich	(3) Data-poor	(4) Data-rich	(5) Data-poor
CV-KF	0.0114	-0.0233	0.0461	-0.221	-0.109
	(0.375)	(0.340)	(0.181)	(0.364)	(0.193)
CV-POOS	-0.765*	-0.762*	-0.768***	-0.700	-0.859***
	(0.375)	(0.340)	(0.181)	(0.364)	(0.193)
AIC	-0.396	-0.516	-0.275	-0.507	-0.522**
	(0.375)	(0.340)	(0.181)	(0.364)	(0.193)
CV-KF * Recessions				1.609	1.264*
				(1.037)	(0.552)
CV-POOS * Recessions				-0.506	0.747
				(1.037)	(0.552)
AIC * Recessions				-0.0760	2.007***
				(1.037)	(0.552)
Observations	91200	45600	45600	45600	45600

Standard errors in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

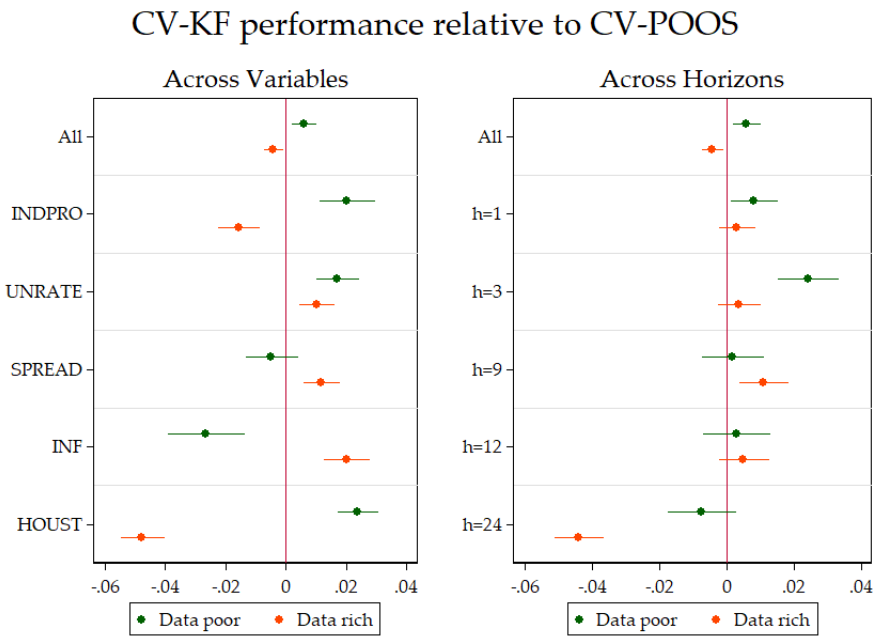

Figure 25: This compares the two CVs procedure averaged over all the models that use them. The unit of the x-axis are improvements in OOS R^1 over the basis model. SEs are HAC. These are the 95% confidence bands.

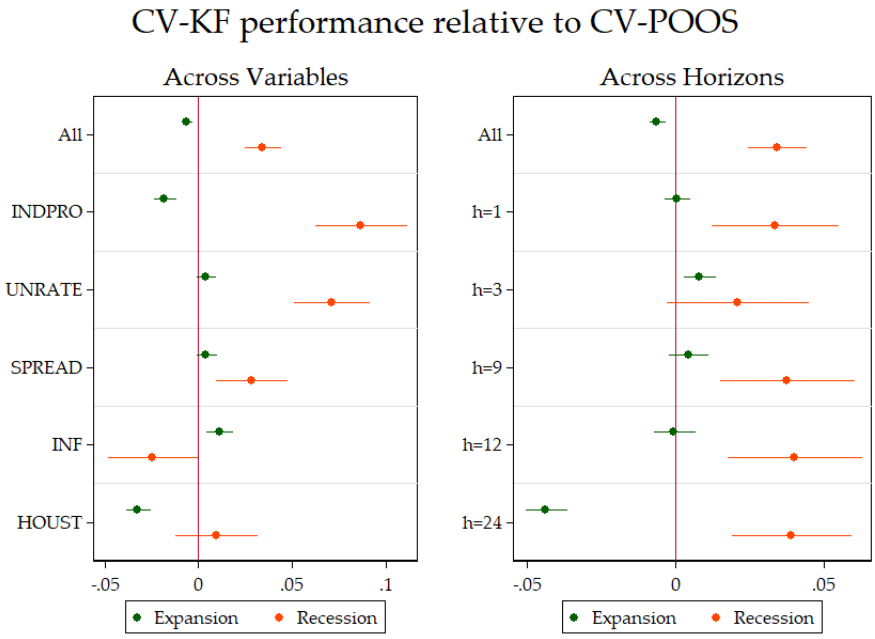

Figure 26: This compares the two CVs procedure averaged over all the models that use them. The unit of the x-axis are improvements in OOS R^1 over the basis model. SEs are HAC. These are the 95% confidence bands.

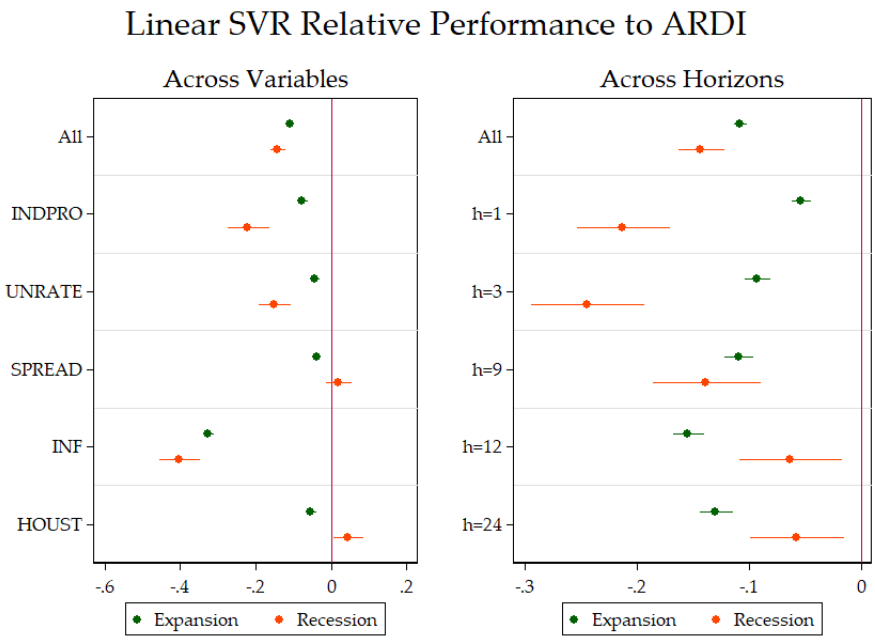

Figure 27: This graph display the marginal (un)improvments by variables and horizons to opt for the SVR in-sample loss function in **both the data-poor and data-rich environments**. The unit of the x-axis are improvements in OOS R^1 over the basis model. SEs are HAC. These are the 95% confidence bands.

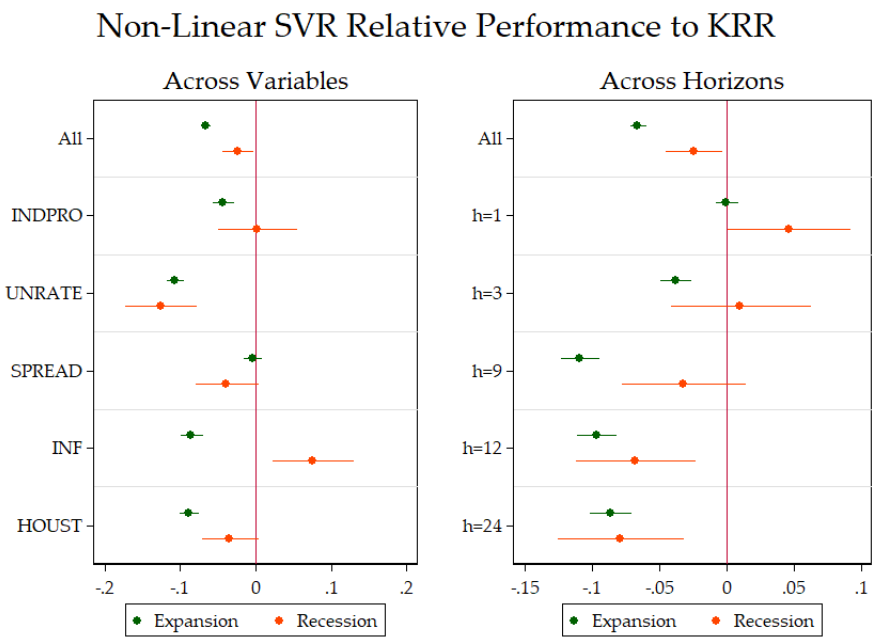

Figure 28: This graph display the marginal (un)improvments by variables and horizons to opt for the SVR in-sample loss function in **both recession and expansion periods**. The unit of the x-axis are improvements in OOS R^1 over the basis model. SEs are HAC. These are the 95% confidence bands.

E Detailed Implementation of Cross-validations

All of our models involve some kind of hyperparameter selection prior to estimation. To curb the overfitting problem, we use two distinct methods that we refer to loosely as cross-validation methods. To make it feasible, we optimize hyperparameters every 24 months as the expanding window grows our in-sample set. The resulting optimization points are the same across all models, variables and horizons considered. In all other periods, hyperparameter values are frozen to the previous values and models are estimated using the expanded in-sample set to generate forecasts.

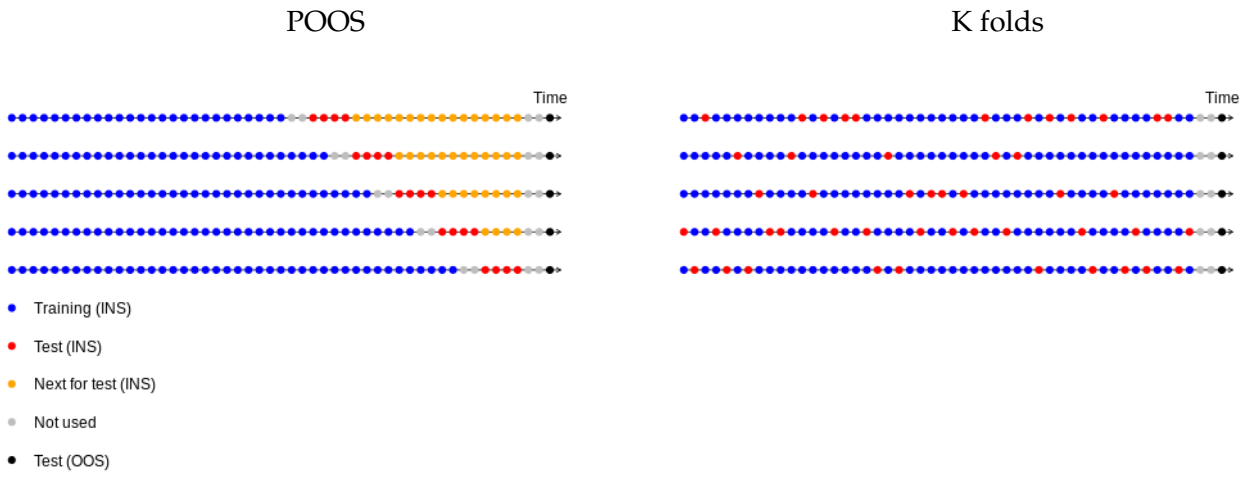

Figure 29: Illustration of cross-validation methods

Notes: Figures are drawn for 3 months forecasting horizon and depict the splits performed in the in-sample set. The pseudo-out-of-sample observation to be forecasted here is shown in black.

The first cross-validation method we consider mimics in-sample the pseudo-out-of-sample comparison we perform across model. For each set of hyperparameters considered, we keep the last 25% of the test set as a comparison window. Models are estimated every 12 months, but the training set is gradually expanded to keep the forecasting horizon intact. This exercice is thus repeated 5 times. Figure 29 shows a toy example with smaller jumps, a smaller comparison window and a forecasting horizon of 3 months, hence the gaps. Once hyperparameters have been selected, the model is estimated using the whole in-sample set and used to make a forecast in the pseudo-out-of-sample window we use to compare all models (the black dot in the figure). This approach is a compromise between two methods used to evaluate time series models detailed in Tashman (2000), rolling-origin recalibration and rolling-origin updating.[15] For a simulation study of various cross-validation methods in a

[15] In both cases, the last observation (the origin of the forecast) of the training set is rolled forward. However, in the first case, hyperparameters are recalibrated and, in the second, only the information set is updated.

time series context, including the rolling-origin recalibration method, the reader is referred to Bergmeir and Benítez (2012). We stress again that the compromise is made to bring down computation time.

The second cross-validation method, K-fold cross-validation, is based on a re-sampling scheme (Bergmeir et al. (2018)). We chose to use 5 folds, meaning the in-sample set is randomly split into five disjoint subsets, each accounting on average for 20 % of the in-sample observations. For each one of the 5 subsets and each set of hyperparameters considered, 4 subsets are used for estimation and the remaining corresponding observations of the in-sample set used as a test subset to generate forecasting errors. This is illustrated in figure 29 where each subsets is illustrated by red dots on different arrows.

Note that the average mean squared error in the test subset is used as the performance metric for both cross-validation methods to perform hyperparameter selection.

F Forecasting models in detail

F.1 Data-poor (H_t^-) models

In this section we describe forecasting models that contain only lagged values of the dependent variable, and hence use a small amount of predictors, H_t^-.

Autoregressive Direct (AR) The first univariate model is the so-called *autoregressive direct* (AR) model, which is specified as:

$$y_{t+h}^{(h)} = c + \rho(L)y_t + e_{t+h}, \quad t = 1, \dots, T,$$

where $h \geq 1$ is the forecasting horizon. The only hyperparameter in this model is p_y, the order of the lag polynomial $\rho(L)$. The optimal p is selected in four ways: (i) Bayesian Information Criterion (AR,BIC); (ii) Akaike Information Criterion (AR,AIC); (iii) Pseudo-out-of-sample cross validation (AR,POOS-CV); and (iv) K-fold cross validation (AR,K-fold). The lag order is selected from the following subset $p_y \in \{1, 3, 6, 12\}$. Hence, this model enters the following categories: linear g function, no regularization, in-sample and cross-validation selection of hyperparameters and quadratic loss function.

Ridge Regression AR (RRAR) The second specification is a penalized version of the previous AR model that allows potentially more lagged predictors by using Ridge regression. The model is written as in (F.1), and the parameters are estimated using Ridge penalty. The Ridge hyperparameter is selected with two cross validation strategy, which gives two models: RRAR,POOS-CV and RRAR,K-fold. The lag order is selected from the following subset

$p_y \in \{1, 3, 6, 12\}$ and for each of these value we choose the Ridge hyperparameter. This model creates variation on following axes: linear g, Ridge regularization, cross-validation for tuning parameters and quadratic loss function.

Random Forests AR (RFAR) A popular way to introduce nonlinearities in the predictive function g is to use a tree method that splits the predictors space in a collection of dummy variables and their interactions. Since a standard tree regression is prompt to the overfitt, we use instead the random forest approach described in Section 3.1.2. As in the literature we set the number of predictors in each tree to one third of all the predictors and the observations in each set are sampled with replacement to get as many observations in the trees as in the full sample. The number of lags of y_t, is chosen from the subset $p_y \in \{1, 3, 6, 12\}$ with cross-validation while the number of trees is selected internally with out-of-bag observations. This model generates nonlinear approximation of the optimal forecast, without regularization, using both CV techniques with the quadratic loss function: RFAR,K-fold and RFAR,POOS-CV.

Kernel Ridge Regression AR (KRRAR) This specification adds a nonlinear approximation of the function g by using the Kernel trick as in Section 3.1.1. The model is written as in (12) and (13) but with the autoregressive part only

$$y_{t+h} = c + g(Z_t) + \varepsilon_{t+h},$$
$$Z_t = \left[\{y_{t-0}\}_{j=0}^{p_y}\right],$$

and the forecast is obtained using the equation (15). The hyperparameters of Ridge and radial basis function kernel are selected by two cross-validation procedure, which gives two forecasting specifications: KRRAR,POOS-CV and KRRAR,K-fold. Z_t consist of y_t and its p_y lags, $p_y \in \{1, 3, 6, 12\}$. This model is representative of a nonlinear g function, Ridge regularization, cross-validation to select τ and quadratic $\hat{L}$.

Support Vector Regression AR (SVR-AR) We use the SVR model to create variation among the loss function dimension. In the data-poor version the predictors set Z_t contains y_t and a number of lags chosen from $p_y \in \{1, 3, 6, 12\}$. The hyperparameters are selected with both cross-validation techniques, and we consider two kernels to approximate basis functions, linear and RBF. Hence, there are four versions: SVR-AR,Lin,POOS-CV, SVR-AR,Lin,K-fold, SVR-AR,RBF,POOS-CV and SVR-AR,RBF,K-fold. The forecasts are generated using (18).

F.2 Data-rich (H_t^+) models

We now describe forecasting models that use a large dataset of predictors, including the autoregressive components, H_t^+.

Diffusion Indices (ARDI) The reference model in the case of large predictor set is the autoregression augmented with diffusion indices from Stock and Watson (2002b):

$$y_{t+h}^{(h)} = c + \rho(L)y_t + \beta(L)F_t + e_{t+h}, \quad t = 1, \dots, T \tag{19}$$
$$X_t = \Lambda F_t + u_t \tag{20}$$

where F_t are K *consecutive static* factors, and $\rho(L)$ and $\beta(L)$ are lag polynomials of orders p_y and p_f respectively. The feasible procedure requires an estimate of F_t that is usually done by PCA.The optimal values of hyperparamters p, K and m are selected in four ways: (i) Bayesian Information Criterion (ARDI,BIC); (ii) Akaike Information Criterion (ARDI,AIC); (iii) Pseudo-out-of-sample cross validation (ARDI,POOS-CV); and (iv) K-fold cross validation (ARDI,K-fold). These are selected from following subsets: $p_y \in \{1, 3, 6, 12\}$, $K \in \{3, 6, 10\}$, $p_f \in \{1, 3, 6, 12\}$. Hence, this model following features: linear g function, PCA regularization, in-sample and cross-validation selection of hyperparameters and L^2.

Ridge Regression Diffusion Indices (RRARDI) As for the small data case, we explore how a regularization affects the predictive performance of the reference model ARDI above. The predictive regression is written as in (19) and p_y, p_f and K are selected from the same subsets of values as for the ARDI case above. The parameters are estimated using Ridge penalty. All the hyperparameters are selected with two cross validation strategies, giving two models: RRARDI,POOS-CV and RRARDI,K-fold. This model creates variation on following axes: linear g, Ridge regularization, CV for tuning parameters and L^2.

Random Forest Diffusion Indices (RFARDI) We also explore how nonlinearities affect the predictive performance of the ARDI model. The model is as in (19) but a Random Forest of regression trees is used. The ARDI hyperparameters are chosen from the grid as in the linear case, while the number of trees is selected with out-of-bag observations. Both POOS and K-fold CV are used to generate two forecasting models: RFARDI,POOS-CV and RFARDI,K-fold. This model generates nonlinear treatment, with PCA regularization, using both CV techniques with the quadratic loss function.

Kernel Ridge Regression Diffusion Indices (KRRARDI) As for the autoregressive case, we can use the Kernel trick to generate nonlinear predictive functions g. The model is represented by equations (12) - (14) and the forecast is obtained using the equation (15).

The hyperparameters of Ridge and radial basis function kernel, as well as p_y, K and p_f are selected by two cross-validation procedures, which gives two forecasting specifications: KRRARDI,POOS-CV and KRRARDI,K-fold. We use the same grid as in ARDI case for discrete hyperparameters. This model is representative of a nonlinear g function, Ridge regularization with PCA, cross-validation to select τ and quadratic $\hat{L}$.

Support Vector Regression ARDI (SVR-ARDI) We use four versions of the SVR model: (i) SVR-ARDI,Lin,POOS-CV; (ii) SVR-ARDI,Lin,K-fold; (iii) SVR-ARDI,RBF,POOS-CV; and (iv) SVR-ARDI,RBF,K-fold. The SVR hyperparameters are selected with cross validation while the ARDI hyperparameters are chosen using a grid that search in the same subsets as the ARDI model. The forecasts are generated from equation (18). This model creates variations in all categories: nonlinear g, PCA regularization, CV and $\bar{\epsilon}$-insensitive loss function.

F.2.1 Generating shrinkage schemes

The rest of the forecasting models relies on using different B operators to generate variations across shrinkage schemes, as depicted in section 3.2.

B_1: taking all observables H_t^+ When B is identity mapping, we consider $Z_t = H_t^+$ in the Elastic Net problem (17), where H_t^+ is defined by (4). The following lag structures for y_t and X_t are considered, $p_y \in \{1, 3, 6, 12\}$ $p_f \in \{1, 3, 6, 12\}$, and the exact number is cross-validated. The hyperparameter λ is always selected by two cross validation procedures, while we consider three cases for α: $\hat{\alpha}$, $\alpha = 1$ and $\alpha = 0$, which correspond to EN, Ridge and Lasso specifications respectively. In case of EN, α is also cross-validated. This gives six combinations: $(B_1, \alpha = \hat{\alpha})$,POOS-CV; $(B_1, \alpha = \hat{\alpha})$,K-fold; $(B_1, \alpha = 1)$,POOS-CV; $(B_1, \alpha = 1)$,K-fold; $(B_1, \alpha = 0)$,POOS-CV and $(B_1, \alpha = 0)$,K-fold. They create variations within regularization and hyperparameters' optimization.

B_2: taking all principal components of X_t Here $B_2()$ rotates X_t into N factors, F_t, estimated by principal components, which then constitute Z_t to be used in (17). Same lag structures and hyperparameters' optimization from the B_1 case are used to generate the following six specifications: $(B_2, \alpha = \hat{\alpha})$,POOS-CV; $(B_2, \alpha = \hat{\alpha})$,K-fold; $(B_2, \alpha = 1)$,POOS-CV; $(B_2, \alpha = 1)$,K-fold; $(B_2, \alpha = 0)$,POOS-CV and $(B_2, \alpha = 0)$,K-fold.

B_3: taking all principal components of H_t^+ Finally, $B_3()$ rotates H_t^+ by taking all principal components, where H_t^+ lag structure is to be selected as in the B_1 case. Same variations and hyperparameters' selection are used to generate the following six specifications: $(B_3, \alpha = \hat{\alpha})$,POOS-CV; $(B_3, \alpha = \hat{\alpha})$,K-fold; $(B_3, \alpha = 1)$,POOS-CV; $(B_3, \alpha = 1)$,K-fold; $(B_3, \alpha = 0)$,POOS-CV and $(B_3, \alpha = 0)$,K-fold.