International Journal of Forecasting 37 (2021) 1338–1354

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Macroeconomic data transformations matter☆

Philippe Goulet Coulombe [a,*], Maxime Leroux [b], Dalibor Stevanovic [b,**], Stéphane Surprenant [b]

[a] University of Pennsylvania, United States of America
[b] Université du Québec à Montréal, Canada

ARTICLE INFO

Keywords:
Machine learning
Big data
Forecasting
Feature engineering
Regularization

ABSTRACT

In a low-dimensional linear regression setup, considering linear transformations/combinations of predictors does not alter predictions. However, when the forecasting technology either uses shrinkage or is nonlinear, it does. This is precisely the fabric of the machine learning (ML) macroeconomic forecasting environment. Pre-processing of the data translates to an alteration of the regularization – explicit or implicit – embedded in ML algorithms. We review old transformations and propose new ones, then empirically evaluate their merits in a substantial pseudo-out-sample exercise. It is found that traditional factors should almost always be included as predictors and moving average rotations of the data can provide important gains for various forecasting targets. Also, we note that while predicting directly the average growth rate is equivalent to averaging separate horizon forecasts when using OLS-based techniques, the latter can substantially improve on the former when regularization and/or nonparametric nonlinearities are involved.

1. Introduction

Following the recent enthusiasm for Machine Learning (ML) methods and widespread availability of big data, macroeconomic forecasting research gradually evolved further and further away from the traditional tightly specified OLS regression. Rather, nonparametric nonlinearity and regularization of many forms are slowly taking the center stage, largely because they can provide sizable forecasting gains when compared with traditional methods (see, among others, Goulet Coulombe (2020a), Goulet Coulombe et al. (2019), Kim and Swanson (2018), Medeiros et al. (2019)), even during the Covid-19 episode (Goulet Coulombe et al., 2021). In such environments, different linear transformations of the informational set X *can* change the prediction and taking first differences may *not* be the optimal transformation for many predictors, even though it guarantees viable frequentist inference. For instance, in penalized regression problems – like Lasso or Ridge –, different rotations of X imply different priors on β in the original regressor space. Moreover, in tree-based model algorithms, the problem of inverting a near singular matrix $X'X$ simply does not happen, making the use of more persistent (and potentially highly cross-correlated regressors) much less harmful. In sum, in the ML macro forecasting environment, traditional data transformations – such as those designed to enforce stationarity (McCracken & Ng, 2016) – may leave some forecasting gains on the table. To guide the growing number of researchers and practitioners in the field, we conduct an extensive pseudo-out-of-sample forecasting exercise to evaluate the virtues of standard and newly proposed data transformations.

☆ We thank the Editor Esther Ruiz, two anonymous referees, and Hugo Couture who provided excellent research assistance. We acknowledge financial support from the Chaire en macroéconomie et prévisions ESG UQAM, Canada.

* Correspondence to: Department of Economics, UPenn, United States of America.
** Correspondence to: Département des sciences économiques, UQAM, Canada.
E-mail addresses: gouletc@sas.upenn.edu (P. Goulet Coulombe), dstevanovic.econ@gmail.com (D. Stevanovic).

https://doi.org/10.1016/j.ijforecast.2021.05.005
0169-2070/

From the ML perspective, it is often suggested that a "feature engineering" step may improve algorithms' performance (Kuhn & Johnson, 2019). This is especially true of Random Forests (RF) and Boosted Trees (BT), two regression tree ensembles widely regarded as the best performing off-the-shelf algorithms within the modern ML canon (Hastie et al., 2009). Among other things, both successfully handle a high-dimensional X by recruiting relevant predictors in a sea of useless ones. This implies the data scientist leveraging some domain knowledge can create plausibly more salient features out of the original data matrix, and let the algorithm decide whether to use them or not. Of course, an extremely flexible model, like a neural network with many layers, could very well create those relevant transformations internally in a data-driven way. Yet, this idyllic scenario is a dead end when data points are few, regressors are numerous, and a noisy y serves as a prediction target. This sort of environment, of which macroeconomic forecasting is a notable example, will often benefit from any prior knowledge one can incorporate in the model. Since transforming the data transforms the prior, doing so properly by including well-motivated rotations of X has the power to increase ML performance on such challenging data sets.

Macroeconomic modelers have been thinking about designing successful priors for a long time. There is a wide literature on Bayesian Vector Autoregressions (VAR) starting with Doan et al. (1984). Even earlier on, the penalized/restricted estimation of lag polynomials was extensively studied (Almon, 1965; Shiller, 1973). The motivation for both strands of work is the large ratio of parameters to observations. Forty years later, many more data points are available, but models have grown in complexity. Consequently, large VARs (Bańbura et al., 2010) and MIDAS regression (Ghysels et al., 2004) still use those tools to regularize over-parametrized models. ML algorithms, usually allowing for sophisticated functional forms, also critically rely on shrinkage. However, when it comes to nonlinear nonparametric methods – especially Boosting and Random Forests – there are no explicit parameters to penalize. Nevertheless, in the case of RF, the ensuing ensemble averaging prediction benefits from ridge-like shrinkage as randomization allows each feature to contribute to the prediction, albeit in a moderate way (Hastie et al., 2009; Mentch & Zhou, 2019). Just like rotating regressor changes the prior in a Ridge regression (see discussion in Goulet Coulombe (2020b)), rotating regressors in such algorithms will alter the implicit shrinkage scheme – i.e., move the prior mean away from the traditional zero. This motivates us to propose two rotations of X that implicitly implement a more time-series-friendly prior in ML models: moving average factors (MAF) and moving average rotation of X (MARX). Other than those motivated above, standard transformations are also being studied. This includes factors extracted by principal components of X and the inclusion of variables in levels to retrieve low-frequency information.

We are interested in predicting stationary targets through a *direct* (in opposition to iterated) forecasting approach. There are at least two ways one can construct direct forecasts of the *average growth* rate of a variable over the next $h > 1$ months – an important quantity for the conduct of monetary policy and fiscal planning. A popular approach is to forecast the final object of interest by projecting it directly on the informational set X (e.g., Stock and Watson 2002a). An alternative is the path average approach where every step until the final horizon is predicted separately. A potential benefit of fitting the whole path first and then constructing the final target is to allow for the selected predictors, the harshness of regularization, and the type of nonlinearities to fully adapt when different relationships arise among the variables during the path.[1] Since those three modeling elements are wildly nonlinear operations in the original input, averaging the path before or after ML is performed can produce very different results.

To evaluate the contribution of data transformations for macroeconomic prediction, we conduct an extensive pseudo-out-of-sample forecasting experiment (38 years, 10 key monthly macroeconomic indicators, 6 horizons) with three linear and two nonlinear ML methods (Elastic Net, Adaptive Lasso, Linear Boosting, Random Forests, and Boosted Trees), and two standard econometric reference models (autoregressive and factor-augmented autoregression).

The main results can be summarized as follows. **First**, combining non-standard data transformations, *MARX*, *MAF* and *Level*, minimizes the RMSE for 8 and 9 variables out of 10 when respectively predicting at short horizons 1 and 3-month ahead. They remain resilient at longer horizons as they are part of the best RMSE specifications around 80% of the time. **Second**, their contribution is magnified when combined with nonlinear ML models – 38 out of 47 cases[2] – with an advantage for Random Forests over Boosted Trees. Both algorithms allow for nonlinearities via tree base learners and make heavy use of shrinkage via ensemble averaging. This is precisely the algorithmic environment we conjectured could benefit most from non-standard transformations of X. **Third**, traditional factors can help tremendously. The overwhelming majority of best information sets for each target included factors. In that regard, this amounts to a clear takeaway message: while ML methods can handle the high-dimensional X (both computationally and statistically), extracting common factors remains straightforward feature engineering that pays off. **Fourth**, the path average approach is preferred to the direct counterpart for almost all real activity variables and at most horizons. Combined with high-dimensional methods that use some form of regularization improves predictability by as much as 30%.

The rest of the paper is organized as follows. In Section 2, we present the ML predictive framework and detail the data transformations and forecasting models. In Section 3, we detail the forecasting experiment and in Section 4 we present main results. Section 5 concludes.

[1] An obvious drawback is that implies estimating and tuning h models rather than one.

[2] There are 47 cases where at least one of these transformations is used.

2. Machine learning forecasting framework

Machine learning algorithms offer ways to approximate unknown and potentially complicated functional forms to minimize the expected loss of a forecast over h periods. The focus of the current paper is to construct a feature matrix susceptible to improve the macroeconomic forecasting performance of off-the-shelf ML algorithms. Let $H_t = [H_{1t}, \dots, H_{Kt}]$ for $t = 1, \dots, T$ be the vector of variables found in a large macroeconomic dataset and let y_{t+h} be our target variable that is supposed stationary. The corresponding prediction problem is given by

$$y_{t+h} = g(f_Z(H_t)) + e_{t+h}. \tag{1}$$

To illustrate the data pre-processing point, define $Z_t \equiv f_Z(H_t)$ as the N_Z-dimensional feature vector, formed by combining several transformations of the variables in H_t.[3] The function f_Z represents the data pre-processing and/or featuring engineering whose effects on forecasting performance we seek to investigate. The training problem for $f_Z = I()$ is

$$\min_{g\in\mathcal{G}} \left\{ \sum_{t=1}^{T} (y_{t+h} - g(H_t))^2 + \text{pen}(g;\tau) \right\}. \tag{2}$$

The function g, chosen as a point in the functional space $\mathcal{G}$, maps transformed inputs into the transformed targets. pen() is the regularization function whose strength depends on some vector/scalar hyperparameter(s) τ. Let $\circ$ denote the function product and $\tilde{g} := g \circ f_Z$. Clearly, introducing a general f_Z leads to

$$\min_{g\in\mathcal{G}} \left\{ \sum_{t=1}^{T} (y_{t+h} - g(f_Z(H_t)))^2 + \text{pen}(g;\tau) \right\}$$
$$\leftrightarrow \min_{\tilde{g}\in\mathcal{G}} \left\{ \sum_{t=1}^{T} \left(y_{t+h} - \tilde{g}(H_t)\right)^2 + \text{pen}(f_Z^{-1} \circ \tilde{g};\tau) \right\}$$

which is, simply, a change of regularization. Now, let $g^*(f_Z^*(H_t))$ be the "oracle" combination of best transformation f_Z and true function g. Let $g(f_Z(H_t))$ be a functional form and data pre-processing selected by the practitioner. In addition, denote $\hat{g}(Z_t)$ and $\hat{y}_{t+h}$ the fitted model and its forecast. The forecast error can be decomposed as

$$y_{t+h} - \hat{y}_{t+h} = \underbrace{g^*(f_Z^*(H_t)) - g(f_Z(H_t))}_{\text{approximation error}} + \underbrace{g(Z_t) - \hat{g}(Z_t)}_{\text{estimation error}} + e_{t+h}. \tag{3}$$

While the intrinsic error e_{t+h} is not shrinkable, the estimation error can be reduced by either adding more relevant data points or restricting the domain $\mathcal{G}$. The benefits of the latter can be offset by a corresponding increase of the approximation error. Thus, an optimal f_Z is one that entails a prior that reduces estimation error at a minimal approximation error cost. Additionally, since most ML algorithms perform variable selection, there is the extra possibility of pooling different f_Z's together and let the algorithm itself choose the relevant restrictions.[4]

The marginal impact of the increased domain $\mathcal{G}$ has been explicitly studied in Goulet Coulombe et al. (2019), with Z_t being factors extracted from the stationarized version of FRED-MD. The primary objective of this paper is to study the relevance of the choice of f_Z, combined with popular ML approximators g.[5] To evaluate the virtues of standard and newly proposed data transformations, we conduct a pseudo-out-of-sample (POOS) forecasting experiment using various combinations of f_Z's and g's.

Finally, a question often overlooked in the forecasting literature is how one should construct the forecast for average growth/difference of the level variable Y_t, which is the popular target in macroeconomic applications. The usual approach – and also the least computationally demanding – is that of fitting the model on $y_{t+h} = \sum_{h'=1}^{h} \Delta Y_{t+h'}/h$ directly and using $\hat{y}_{t+h}^{\text{direct}}$ as prediction, where $\Delta Y_{t+h'} = Y_{t+h'} - Y_{t+h'-1}$ is the simple growth/difference of the variable of interest. Another approach, requiring the estimation of h different functions, is the *path average* approach where each $\Delta Y_{t+h'}$ is fitted separately and the forecast for y_{t+h} is obtained from $\hat{y}_{t+h}^{\text{path-avg}} = \sum_{h'=1}^{h} \widehat{\Delta Y}_{t+h'}/h$.

The common wisdom – from OLS – is that such strategies are interchangeable. But the equivalence does not hold when regularization and nonparametric nonlinearities are involved. For instance, it breaks in the simplest possible departure from OLS, a ridge regression, where

$$\hat{y}_{t+h}^{\text{path-avg}} = \frac{1}{h} \sum_{h'=1}^{h} Z(Z'Z + \lambda_{h'} I)^{-1} Z' \Delta Y_{t+h'}, \tag{4}$$

and only if $\lambda_{h'} = \lambda \ \forall h'$ then

$$\hat{y}_{t+h}^{\text{path-avg}} = Z(Z'Z + \lambda I)^{-1} Z' \frac{\sum_{h'=1}^{h} \Delta Y_{t+h'}}{h} = \hat{y}_{t+h}^{\text{direct}}. \tag{5}$$

This setup naturally includes the known equivalence in the OLS case ($\lambda_{h'} = 0 \ \forall h'$). We get even further from the equivalence with Lasso, Random Forests, and Boosted Trees which all imply the nonlinear hard-thresholding operation of variable selection – and basis expansion creation for the last two. With those, we get even further from the equivalence by having a different $Z_{h'}^* \subset Z$ in each prediction function.

Of course, the path average approach can be rather demanding since it implies h estimation (and likely cross-validation) problems — with the benefit of providing a whole path rather than merely y_{t+h}. The second question address then concerns whether those benefits could additionally include forecasting gains. To investigate this and how this choice interacts with the optimal f_Z, we conduct the whole forecasting exercise using both schemes.

[3] Obviously, in the context of a pseudo-out-of-sample experiment, feature matrices must be built recursively to avoid data snooping.

[4] More concretely, a factor F is a linear combination of X. If an algorithm picks F rather than creating its own combination of different elements of X, it is implicitly imposing a restriction.

[5] There are many recent contributions considering the macroeconomic forecasting problem with econometric and machine learning methods in a big data environment (Kim & Swanson, 2018; Kotchoni et al., 2019). However, they are done using the standard stationary version of the FRED-MD database. Recently, McCracken and Ng (2020) studied the relevance of unit root tests in the choice of stationarity transformation codes for macroeconomic forecasting with factor models.

2.1. Old news

Firstly, we consider more traditional candidates for f_Z.

Including Factors. Common practice in the macroeconomic forecasting literature is to rely on some variant of the transformations proposed by McCracken and Ng (2016) to obtain a stationary X_t out of H_t. Letting $X = [X_t]_{t=1}^T$ and imposing a linear latent factor structure $X = F\Lambda + \epsilon$, we can estimate F by the principal components of X. The feature matrix of the autoregressive diffusion index (FM hereafter) model of Stock and Watson (2002a, 2002b) can be formed as

$$Z_t = \left[y_t, Ly_t, \ldots, L^{p_y} y_t, F_t, LF_t, \ldots, L^{p_f} F_t\right] \tag{6}$$

where L is the lag operator and y_t is the current value of the target. In Goulet Coulombe et al. (2019), factors were deemed the most reliable shrinkage method for macroeconomic forecasting, even when considering ML alternatives. Furthermore, the combination of factors (and nothing else) with nonlinear nonparametric methods is (i) easy, (ii) fast, and (iii) often quite successful. Point (iii) is further re-enforced by this paper's results, especially for forecasting inflation, which contrasts with the results found in Medeiros et al. (2019).

Including Levels. In econometrics, debates on the consequences of unit roots for frequentist inference have a long history,[6] just as does the handling of low-frequency movements for macroeconomic forecasting (Elliott, 2006). Exploiting potential cointegration has been found useful to improve forecasting accuracy under some conditions (e.g., Christoffersen and Diebold (1998), Engle and Yoo (1987), Hall et al. (1992)). From the perspective of engineering a feature matrix, the error correction term could be obtained from a first step regression *à la* Engle and Granger (1987) and is just a specific linear combination of existing variables. When it is unclear which variables should enter the cointegrating vector – or whether there exists any such vector – one can alternatively include both variables in levels and differences into the feature matrix. This sort of approach has been pursued most notably by Cook and Hall (2017) who combine variables in levels, first differences, and even second differences in the feature matrix they provide to various neural network architectures in the forecasting of US unemployment data.[7]

From a purely predictive point of view, using first differences rather than levels is a linear restriction (using the vector $[1, -1]$) on how H_t and H_{t-1} can jointly impact y_t. Depending on the prior/regularization being used with linear regression, this may largely decrease the estimation error or inflate the approximation one.[8] However, it is often admitted that in a time series context (even if Bayesian inference is left largely unaltered by nonstationarity (Sims, 1988)), first differences are useful because they trim out low frequencies which may easily be redundant in large macroeconomic data sets. Using a collection of highly persistent time series in X can easily lead to an unstable $X'X$ inverse (or even a regularized version). Such problems naturally extend to Lasso (Lee et al., 2018). In contrast, tree-based approaches like RF and Boosted Trees do not rely on inverting any matrix. Of course, performing tree-like sample splitting on a trending variable like raw GDP (without any subsequent split on lag GDP), is almost equivalent to split the sample according to a time trend and will often be redundant and/or useless. Nevertheless, there are numerous H_t's where opting for first differencing the data is much less trivial. In such cases, there may be forecasting benefits from augmenting the usual X with levels.

2.2. New avenues

When regressors outnumber observations, regularization, whether explicit or implicit, is necessary. Hence, the ML algorithms we use all entail a prior which may or may not be well suited for a time series problem. There is a wide Bayesian VAR literature, starting with Doan et al. (1984), proposing prior structures that are thought for the multiple blocks of lags characteristic of those models. Additionally, there is a whole strand of older literature that seeks to estimate restricted lag polynomials in Autoregressive Distributed Lags (ARDL) models (Almon, 1965; Shiller, 1973). While the above could be implemented in a parametric ML model with a moderate amount of pain, it is not clear how such priors framed in terms of lag polynomials can be put to use when there is no explicit lag polynomial. A more convenient approach is to **(i)** observe that most nonparametric ML methods implicitly shrink the individual contribution of each feature to zero in a Ridge-ean fashion (Elliott et al., 2013; Hastie et al., 2009) and **(ii)** rotating regressors implies a new prior in the original space. Hence, by simply creating regressors that embody the more sophisticated linear restrictions, we obtain shrinkage better suited for time series.[9] The first step in that direction is Goulet Coulombe (2020a) who proposes Moving Average Factors to specifically enhance RF's prediction and interpretation potential. A second is to find a rotation of the original lag polynomial such that implementing Ridge-ean shrinkage in fact yields (Shiller, 1973) approach to shrinking lag polynomials.

Moving Average Factors. Using factors is a standard approach to summarize parsimoniously a panel of heavily cross-correlated variables. Analogously, one can extract a few principal components from each variable-specific panel of lagged values, i.e.

$$\begin{aligned} \tilde{X}_{t,k} &= \left[X_{t,k}, LX_{t,k}, \ldots, L^{p_{MAF}} X_{t,k}\right] \\ \tilde{X}_{t,k} &= M_t \Gamma_k' + \tilde{\epsilon}_{k,t}, \quad k = 1, \ldots, K \end{aligned} \tag{7}$$

[6] See, for example, Phillips (1991a, 1991b), Sims (1988), Sims et al. (1990), Sims and Uhlig (1991).

[7] Another approach is to consider factor modeling directly with nonstationary data (Bai & Ng, 2004; Banerjee et al., 2014; Peña & Poncela, 2006).

[8] A similar comment would apply to all parametric cointegration restrictions. For recent work on the subject, see for example Chan and Wang (2015).

[9] A cross-section RF-based example is Rodriguez et al. (2006) who propose "Rotation Forest" that build an ensemble of trees based on different rotations of X.

to achieve a similar goal on the time axis. Define a moving average factor as the vector M_k.[10] Mechanically, we obtain weighted moving averages, where the weights are the principal component estimates of the loadings in Γ_k. By construction, those extractions form moving averages of the P_{MAF} lags of $X_{t,k}$ so that it summarizes most efficiently its temporal information.[11] By doing so, the goal to summarize information in $X_{t,k}^{1:P_{MAF}}$ is achieved without modifying any algorithm: we can use the MAFs which compresses information ex-ante. As is the case for standard factors, MAF is designed to maximize the explained variance in $X_{t,k}^{1:P_{MAF}}$, not the fit to the final target. It is the learning algorithm's job to select the relevant linear combinations to maximize the fit.

Moving Average Rotation of X. There are many ways one can penalize a lag polynomial. One, in the Minnesota prior tradition, is to shrink all lag coefficients to *zero* (except for the first self-lag) with increasing harshness in p, the order of the lag. Another is to shrink each β_p to β_{p-1} and β_{p+1} rather than to zero. Intuitively, for higher-frequency series (like monthly data would qualify for here) it is more plausible that a simple linear combination of lags impacts y_t rather than a single one of them with all other coefficients set to zero.[12] For instance, it seems more likely that the average of March, April, and May employment growth could impact, say, inflation, than only May's. Mechanically, this means we expect March, April, and May's coefficients to be close to one another, which motivated the prior $\beta_p \sim N(\beta_{p-1}, \sigma_u^2 I_K)$ and more sophisticated versions of it in other works (Shiller, 1973). Inputting in the ML algorithm a transformed X such that its implicit shrinkage to zero is twisted into this new prior could generate forecasting gains. The only question left is how to make this operational.

The following derivation is a simple translation of Goulet Coulombe (2020b)'s insights for time-varying parameters model to regularized lag polynomials à la Shiller (1973).[13] Consider a generic regularized ARDL model with K variables

$$\min_{\beta_1 \dots \beta_P} \sum_{t=1}^{T} \left(y_t - \sum_{p=1}^{P} X_{t-p}\beta_p \right)^2 + \lambda \sum_{p=1}^{P} \|\beta_p - \beta_{p-1}\|^2. \quad (8)$$

where $\beta_p \in \mathbb{R}^K$, $X_t \in \mathbb{R}^K$, $u_p \in \mathbb{R}^{K \times P}$, and both y_t and ϵ_t are scalars.[14] While we adopt the l_2 norm for this exposition, our main goal is to extend traditional regularized lag polynomial ideas to cases where there is no explicitly specified norm on $\beta_p - \beta_{p-1}$. For instance, Elliott et al. (2013) prove that their Complete Subset Regression procedure implies Ridge shrinkage in a special case. Moving away from linearity makes formal arguments more difficult. Nevertheless, it has been argued several times that model/ensemble averaging performs shrinkage akin to that of a ridge regression (Hastie et al., 2009). For instance, random selection of a subset of eligible features at each split encourage each feature to be included in the predictive function, but in a moderate fashion.[15] The resulting "implicit" coefficient is an average of specifications that included the regressor and some that did not. In the latter case, the coefficient is always zero by construction. Hence, the ensemble shrinks contributions towards zero and the so-called `mtry` hyperparameter guides the level of shrinkage like a bandwidth parameter would (Olson & Wyner, 2018).

To get implicit regularized lag polynomial shrinkage, we now rewrite problem (8) as a ridge regression. For all derivations to come, it is less tedious to turn to matrix notations. The Fused Ridge problem is now written as

$$\min_{\boldsymbol{\beta}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})' (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}' \boldsymbol{D}' \boldsymbol{D} \boldsymbol{\beta}$$

where $\boldsymbol{D}$ is the first difference operator. The first step is to reparametrize the problem by using the relationship $\beta_k = C\theta_k$ that we have for all k regressors. C is a lower triangular matrix of ones (for the random walk case) and define $\theta_k = [u_k \quad \beta_{0,k}]$. For the simple case of one parameter and $P = 4$:

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix}.$$

For the general case of K parameters, we have

$$\boldsymbol{\beta} = \boldsymbol{C}\boldsymbol{\theta}, \quad \boldsymbol{C} \equiv I_K \otimes C$$

and $\boldsymbol{\theta}$ is just stacking all the θ_k into one long vector of length KP. Using the reparametrization $\boldsymbol{\beta} = \boldsymbol{C}\boldsymbol{\theta}$, the Fused Ridge problem becomes

$$\min_{\boldsymbol{\theta}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{C}\boldsymbol{\theta})' (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{C}\boldsymbol{\theta}) + \lambda \boldsymbol{\theta}' \boldsymbol{C}' \boldsymbol{D}' \boldsymbol{D} \boldsymbol{C} \boldsymbol{\theta}.$$

Let $\boldsymbol{Z} \equiv \boldsymbol{X}\boldsymbol{C}$ and use the fact that $\boldsymbol{D} = \boldsymbol{C}^{-1}$ to obtain the Ridge regression problem

$$\min_{\boldsymbol{\theta}} (\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\theta})' (\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\theta}) + \lambda \boldsymbol{\theta}' \boldsymbol{\theta}. \quad (9)$$

We arrived at destination. Using $\boldsymbol{Z}$ rather than $\boldsymbol{X}$ in an algorithm that performs shrinkage will implicitly shrink β_p to β_{p-1} rather than to 0. This is obviously much more convenient than modifying the algorithm itself and is directly applicable to *any algorithm* using time series data as input. One question remains: what is $\boldsymbol{Z}$, exactly? For a single polynomial at time t, we have $Z_{t,k} = X_{t,k}C$. C is gradually summing up the columns of $X_{t,k}$ over p. Thus,

[10] While we work directly with the latent factors, a related decomposition called singular spectrum analysis works with the estimate of the summed common components, i.e. with $M_k \Gamma_k'$. Since this decomposition naturally yields a recursive formula, it has been used to forecast macroeconomic and financial variables (Hassani et al., 2009, 2013), usually in a univariate fashion.

[11] P_{MAF} is a tuning parameter analogous to the construction of the panel of variables (usually taken as given) in a standard factor model. We pick $P_{MAF} = 12$. We keep two MAFs for each series and they are obtained by PCA.

[12] This is a dense vs sparse choice. MAFs go all the way with the first view by imposing it via the extraction procedure.

[13] Such reparametrization schemes are also discussed for "fused" Lasso in Tibshirani et al. (2015) and employed for a Bayesian local-level model in Koop (2003).

[14] We use P as a generic maximum number of lags for presentation purposes. In Table 1 we define P_{MARX}.

[15] Recently, Goulet Coulombe (2020) argued that ensemble averaging methods à la RF prunes a latent tree. Following this view, the need for cleverly pre-assembled data combinations is even clearer.

$Z_{t,k,p} = \sum_{p'=1}^{P} X_{t,k,p'}$. Dividing each $Z_{t,k,p}$ by p (just another linear transformation, $\tilde{Z}_{t,k,p}$), it is now clear that $\tilde{\boldsymbol{Z}}$ is a matrix of moving averages. Those are of increasing order (from $p = 1$ to $p = P$) and the last observation in the average is always $X_{t-1,k}$. Hence, we refer to this particular form of feature engineering as Moving Average Rotation of X (MARX).

Recap. We summarize our setup in Table 1. We have five basic sets of transformations to feed the approximation of f_Z^*: (1) single-period differences and growth rates following McCracken and Ng (2016) (X_t and their lags), (2) principal components of X_t (F_t and their lags), (3) variables in levels (H_t and their lags), (4) moving average factors of X_t (MAF_t), and (5) sets of simple moving averages of X_t ($MARX_t$). We consider several forecasting models in order to approximate the true functional form: Autoregressive (AR), Factor Model (FM, à la Stock & Watson, 2002a), Adaptive Lasso (AL), Elastic Net (EN), Linear Boosting (LB), Random Forest (RF), and Boosted Trees (BT). Lastly, we apply those specifications to forecasting both direct and path-average targets. The details on forecasting models are presented in Appendix A.

Furthermore, most ML methodologies that handle well high-dimensional data perform some form or another of variable selection. For instance, RF evaluates a certain fraction of predictors at each split and selects the most potent one. Lasso selects relevant predictors and shrinks others perfectly to zero. By rotating X, we can get these algorithms (and others) to perform restriction/transformation selection. Thus, one should not refrain from studying different combinations of f_Z's.[16] As a result, all the combinations of f_Z thereof are admissible and 16 of them are included in the exercise. Moreover, there is a long-standing worry that well-accepted transformations may lead to some over-differenced X_k's (McCracken & Ng, 2020). Including MARX or MAF (which are both specific partial sums of lags) with X can be seen as bridging the gap between a first difference and keeping H_k in levels. Hence, interacting many f_Z is not only statistically feasible but econometrically desirable given the sizable uncertainty surrounding what is a "proper" transformation of the raw data (Choi, 2015).

3. Forecasting setup

In this section, we present the results of a pseudo-out-of-sample forecasting experiment for a group of target variables at a monthly frequency from the FRED-MD dataset of McCracken and Ng (2016). Our target variables are the industrial production index (INDPRO), total nonfarm employment (EMP), unemployment rate (UNRATE), real personal income excluding current transfers (INCOME), real personal consumption expenditures (CONS), retail and food services sales (RETAIL), housing starts (HOUST), M2 money stock (M2), consumer price index (CPI), and the production price index (PPI). Given that we make predictions at horizons of 1, 3, 6, 9, 12, and 24 months, we are effectively targeting the average growth rate over those periods, except for the unemployment rate for which we target average differences. These series are representative macroeconomic indicators of the US economy, as stated in Kim and Swanson (2018), which is also based on Goulet Coulombe et al. (2019) exercise for many ML models, itself based on Kotchoni et al. (2019) and a whole literature of extensive horse races in the spirit of Stock and Watson (1998). The POOS period starts in January of 1980 and ends in December of 2017. We use an expanding window for estimation starting from 1960M01. Following standard practice in the literature, we evaluate the quality of point forecasts using the root Mean Square Error (RMSE). For the forecasted value at time t of variable v made h steps before, we compute

$$RMSE_{v,h,m} = \sqrt{\frac{1}{\#\text{OOS}} \sum_{t \in \text{OOS}} (y_t^v - \hat{y}_{t-h}^{v,h,m})^2} \tag{10}$$

The standard Diebold and Mariano (2002) (DM) test procedure is used to compare the predictive accuracy of each model against the reference factor model (FM). RMSE is the most natural loss function given that all models are trained to minimize the squared loss in-sample. We also implement the Model Confidence Set (MCS) that selects the subset of best models at a given confidence level (Hansen et al., 2011).

Hyperparameter selection is performed using the BIC for AR and FM and K-fold cross-validation is used for the remaining models. This approach is theoretically justified in time series models under conditions spelled out by Bergmeir et al. (2018). Moreover, Goulet Coulombe et al. (2019) compared it with a scheme that respects the time structure of the data

4. Results

Table 2 shows the best RMSE data transformation combinations as well as the associated functional forms for every target and forecasting horizon. It summarizes the main findings and provides important recommendations for practitioners in the field of macroeconomic forecasting. **First**, including non-standard choices of macroeconomic data transformation, *MARX*, *MAF* and *Level*, minimize the RMSE for 8 and 9 variables out of 10 when respectively predicting 1 and 3-month ahead. Their overall importance is still resilient at longer horizons as they are part of the best specifications of most of the variables. **Second**, their success is often paired with a nonlinear functional form g, 38 out of 47 cases, with an advantage for Random Forests over Boosted Trees. The former is used for 26 of those 38 cases. Both algorithms make heavy use of shrinkage and allow for nonlinearities via tree base learners. This is precisely the algorithmic environment that we precedently conjectured to be where data transformations matter.

Without a doubt, the most visually obvious feature of Table 2 is the abundance of green bullets. As expected, transforming X into factors is probably the most effective form of feature engineering available to the macroeconomic forecaster. Factors are included as part of the

[16] Notwithstanding, some authors have noted that a trade-off emerges between how focused an RF is and its robustness via diversification. Borup et al. (2020) sometimes get improvements over plain RF by adding a Lasso pre-processing step to trim X.

Table 1
Model Specification Summary.

Cases	Feature matrix Z_t
F	$Z_t := \left[\{L^{i-1}F_t\}_1^{p_f}\right]$
F-X	$Z_t := \left[\{L^{i-1}F_t\}_1^{p_f}, \{L^{i-1}X_t\}_1^{p_m}\right]$
F-MARX	$Z_t := \left[\{L^{i-1}F_t\}_1^{p_f}, \{MARX_{yt}^i\}_1^{p_y}, \{MARX_{1t}^i\}_1^{p_m}, \ldots, \{MARX_{Kt}^i\}_1^{p_m}\right]$
F-MAF	$Z_t := \left[\{L^{i-1}F_t\}_1^{p_f}, \{MAF_{yt}^i\}_1^{r_K}, \{MAF_{1t}^i\}_1^{r_K}, \ldots, \{MAF_{Kt}^i\}_1^{r_K}\right]$
F-Level	$Z_t := \left[\{L^{i-1}F_t\}_1^{p_f}, Y_t, H_t\right]$
F-X-MARX	$Z_t := \left[\{L^{i-1}F_t\}_1^{p_f}, \{L^{i-1}X_t\}_1^{p_m}, \{MARX_{yt}^i\}_1^{p_y}, \{MARX_{1t}^i\}_1^{p_m}, \ldots, \{MARX_{Kt}^i\}_1^{p_m}\right]$
F-X-MAF	$Z_t := \left[\{L^{i-1}F_t\}_1^{p_f}, \{L^{i-1}X_t\}_1^{p_m}, \{MAF_{yt}^i\}_1^{r_K}, \{MAF_{1t}^i\}_1^{r_K}, \ldots, \{MAF_{Kt}^i\}_1^{r_K}\right]$
F-X-Level	$Z_t := \left[\{L^{i-1}F_t\}_1^{p_f}, \{L^{i-1}X_t\}_1^{p_m}, Y_t, H_t\right]$
F-X-MARX-Level	$Z_t := \left[\{L^{i-1}F_t\}_1^{p_f}, \{L^{i-1}X_t\}_1^{p_m}, \{MARX_{yt}^i\}_1^{p_y}, \{MARX_{1t}^i\}_1^{p_m}, \ldots, \{MARX_{Kt}^i\}_1^{p_m}, Y_t, H_t\right]$
X	$Z_t := \left[\{L^{i-1}X_t\}_1^{p_m}\right]$
MARX	$Z_t := \left[\{MARX_{yt}^i\}_1^{p_y}, \{MARX_{1t}^i\}_1^{p_m}, \ldots, \{MARX_{Kt}^i\}_1^{p_m}\right]$
MAF	$Z_t := \left[\{MAF_{yt}^i\}_1^{r_K}, \{MAF_{1t}^i\}_1^{r_K}, \ldots, \{MAF_{Kt}^i\}_1^{r_K}\right]$
X-MARX	$Z_t := \left[\{L^{i-1}X_t\}_1^{p_m}, \{MARX_{yt}^i\}_1^{p_y}, \{MARX_{1t}^i\}_1^{p_m}, \ldots, \{MARX_{Kt}^i\}_1^{p_m}\right]$
X-MAF	$Z_t := \left[\{L^{i-1}X_t\}_1^{p_m}, \{MAF_{yt}^i\}_1^{r_K}, \{MAF_{1t}^i\}_1^{r_K}, \ldots, \{MAF_{Kt}^i\}_1^{r_K}\right]$
X-Level	$Z_t := \left[\{L^{i-1}X_t\}_1^{p_m}, Y_t, H_t\right]$
X-MARX-Level	$Z_t := \left[\{L^{i-1}X_t\}_1^{p_m}, \{MARX_{yt}^i\}_1^{p_y}, \{MARX_{1t}^i\}_1^{p_m}, \ldots, \{MARX_{Kt}^i\}_1^{p_m}, Y_t, H_t\right]$

Note: This table show the combinations of data transformation used to assess the individual marginal contribution of each f_Z. Lags of month-to-month (log)-change of the series to forecast are always included.

Table 2
Best model specifications - with target type.

	INDPRO	EMP	UNRATE	INCOME	CONS	RETAIL	HOUST	M2	CPI	PPI
H=1	RF	RF	BT	RF	FM	FM	EN	RF	AL	EN
H=3	RF	RF	RF	RF	RF	BT	EN	AL	RF	EN
H=6	RF	BT	RF	RF	RF	AL	RF	RF	RF	RF
H=9	RF	BT	LB	RF	RF	BT	BT	RF	RF	RF
H=12	RF	BT	LB	RF	RF	BT	RF	BT	RF	RF
H=24	RF	BT	BT	RF	RF	BT	RF	RF	RF	BT

Note: Bullet colors represent data transformations included in the best model specifications: ***F***, ***MARX***, ***X***, ***L*** and ***MAF***. Path average specifications are underlined.

optimal specification for the overwhelming majority of targets. Furthermore, including factors *only* in combination with RF is the best forecasting strategy for both CPI and PPI inflation for the vast majority of horizons. This is in line with findings in Goulet Coulombe et al. (2019) but in contrast with the results found in Medeiros et al. (2019). The major difference with the latter is that they estimate and evaluate models based on a single-month inflation rate, which is only the intermediary step in our path average strategy. In addition, we explore the possibility that F alone could be better than X, rather than always both together. As it turns out, the winning combination is RF using factors as *sole inputs* to directly target the average growth. Finally, the omission of factors from optimal specifications for industrial production growth 3 to 12 months ahead is naturally surprising. This points out that current wisdom based on linear models may not be directly applicable to nonlinear ones. Alternative rotations will sometimes do better.

Plentiful red bullets are populating the top rows of Table 2. Indeed, our most salient new transformation is *MARX*. In combination with nonlinear tree-based models, it contributes to improving forecasting accuracy for real activity series such as industrial production, employment, unemployment rate, and income, while they are best paired with penalized regressions to predict the CPI and PPI inflation rates. The dominance of MARX is particularly striking for real activity series as the transformation is included in *every* best specification for those variables at *all* horizons ranging from one month to a year. We further investigate how those RMSE gains materialize in terms of forecasts around key periods in Section 4.2. While *MAF* performance is often positively correlated with *MARX*, the latter is usually the better of the two, except for longer-run forecasts – like those 2-years where MAF is featured for four variables.

Considering levels is particularly important for the M2 money stock as it is included in the best model for *all* horizons. For other variables, its pertinence is rather sporadic, with at least two horizons featuring it for INDPRO, UNRATE, CONS, and RETAIL.

The preference for $\hat{y}_{t+h}^{\text{direct}}$ vs $\hat{y}_{t+h}^{\text{path-avg}}$ mostly go on a variable by variable basis. However, there is clear consensus $\hat{y}_{t+h}^{\text{path-avg}} \succ \hat{y}_{t+h}^{\text{direct}}$ for all variables which strongly co-move with the business cycle (INDPRO, EMP, UNRATE,

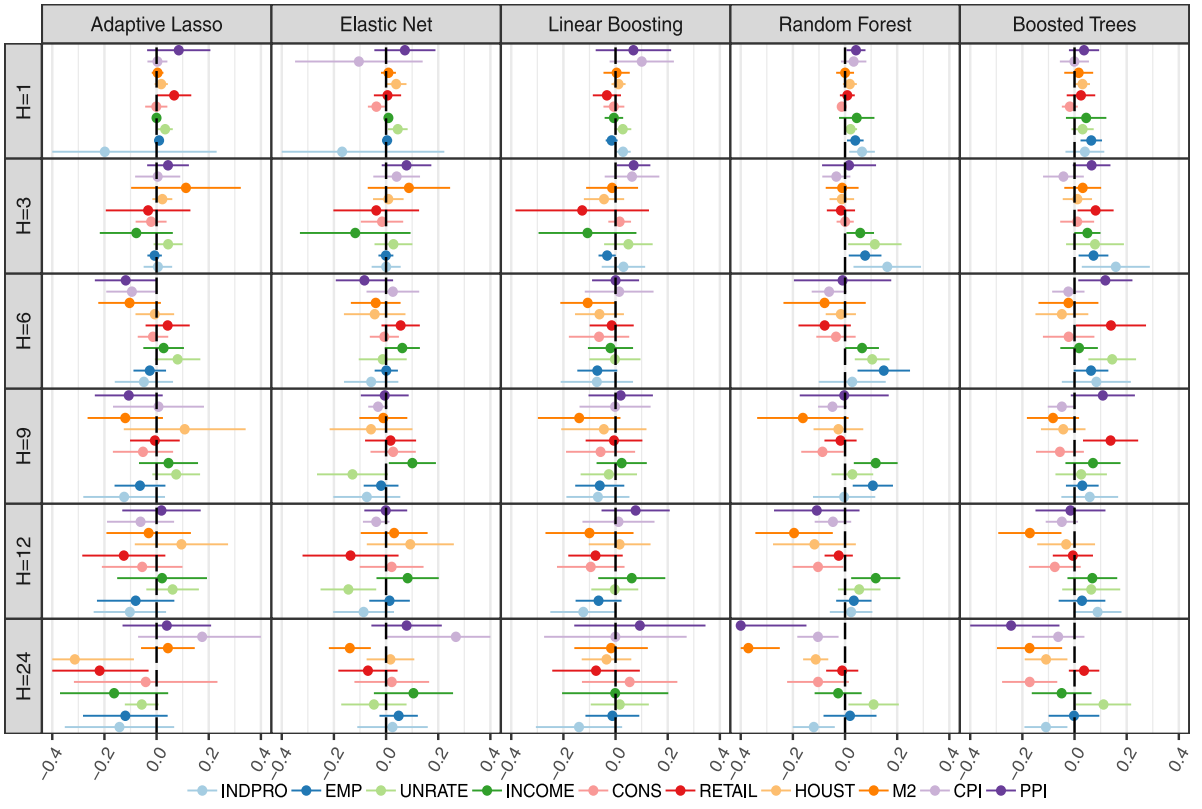


Fig. 1. Distribution of *MARX* Marginal Effects (Average Targets). Note: This figure plots the distribution of $\alpha_f^{(h,v)}$ from Eq. (12) done by (h, v) subsets. That is, it shows the average partial effect on the pseudo-R^2 from augmenting the model with *MARX* featuring, keeping everything else fixed. SEs are HAC. These are the 95% confidence bands.

INCOME, CONS) with the notable exception of retail sales and housing starts. When it comes to nominal targets (M2, CPI, PPI), $\hat{y}_{t+h}^{\text{path-avg}} \prec \hat{y}_{t+h}^{\text{direct}}$ is unanimous for horizons 6 to 12 months, and so are the affiliated data transformations as well as the g choice (all tree ensembles, with 8 out of 9 being RF). The quantitative importance of both types of gains on both sides is studied in Section 4.1, while Section 4.2 looks at implied forecasts to understand when and why $\hat{y}_{t+h}^{\text{path-avg}} \succ \hat{y}_{t+h}^{\text{direct}}$, or the reverse.

These findings are particularly important given the increasing interest in ML macro forecasting. They suggest that traditional data transformations, meant to achieve stationarity, do leave substantial forecasting gains on the practitioners' table. These losses can be successfully recovered by combining ML methods with well-motivated rotations of predictors such as *MARX* and *MAF* (or sometimes by simply including variables in levels) and by constructing the final forecast by the path average approach.

The previous results were desirably expeditive. The detailed results on the underlying performance gains and their statistical significance are presented in Appendix B.

4.1. Marginal contribution of data pre-processing

In order to disentangle *marginal* effects of data transformations on forecast accuracy we run the following regression inspired by Carriero et al. (2019) and Goulet Coulombe et al. (2019):

$$R^2_{t,h,v,m} = \alpha_{\mathcal{F}} + \psi_{t,v,h} + v_{t,h,v,m}, \tag{11}$$

where $R^2_{t,h,v,m} \equiv 1 - \frac{e^2_{t,h,v,m}}{\frac{1}{T}\sum_{t=1}^{T}(y_{v,t+h} - \bar{y}_{v,h})^2}$ is the pseudo-out-of-sample R^2, and $e^2_{t,h,v,m}$ 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 demeans the dependent variable by "forecasting target," that is a combination of t, v, and h. $\alpha_{\mathcal{F}}$ is a vector of α_{MARX}, α_{MAF}, and α_F terms associated to each new data transformation considered in this paper, as well as to the factor model. H_0 is $\alpha_f = 0 \quad \forall f \in \mathcal{F} = [MARX, MAF, F]$. 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 data pre-processing. While the generality of (11) 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

$$\forall m \in \mathcal{M}_f : \quad R^2_{t,h,v,m} = \alpha_f + \psi_{t,v,h} + v_{t,h,v,m} \tag{12}$$

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

MARX. Fig. 1 plots the distribution of $\alpha_{MARX}^{(h,v)}$ from Eq. (11) done by (h, v) subsets. Hence, we allow for heterogeneous

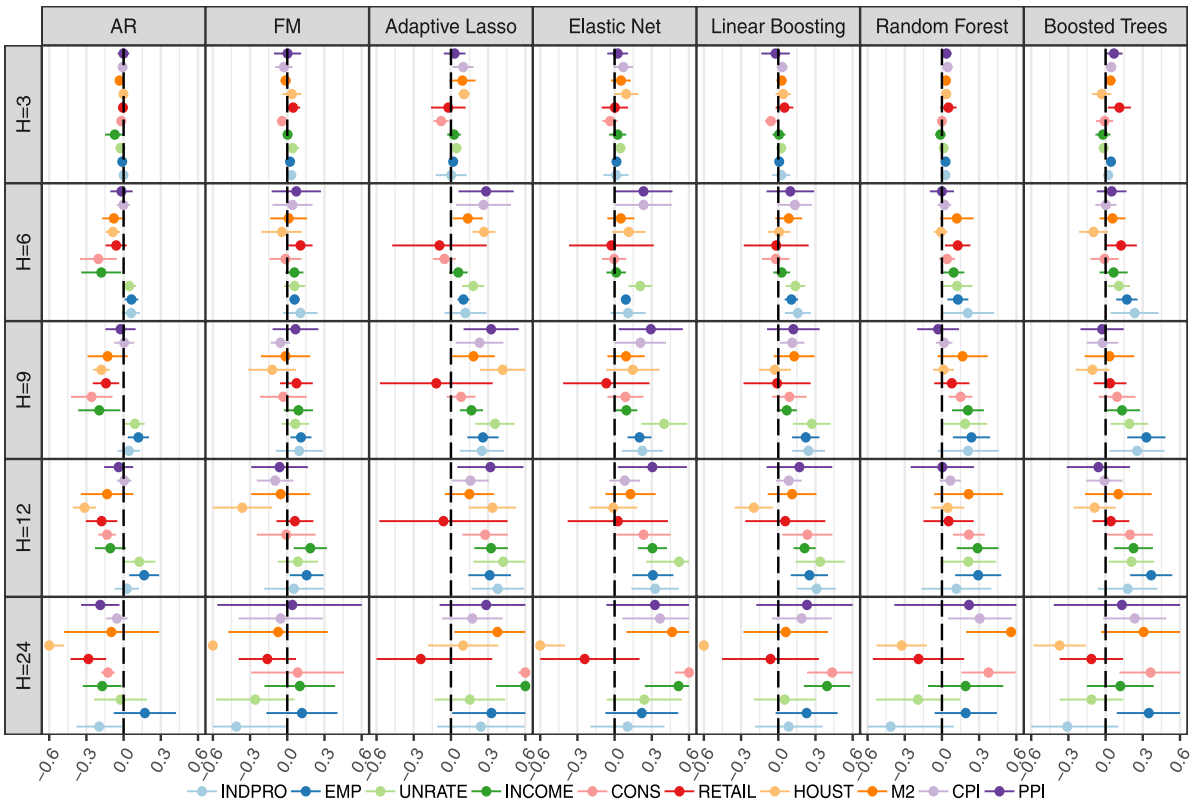


Fig. 2. Distribution of Marginal Effects of Target Transformation. Note: This figure plots the distribution of $\alpha_f^{(h,v)}$ from Eq. (12) done by (h, v) subsets. That is, it shows the average partial effect on the pseudo-R^2 from accumulating single period predictions ($\hat{y}_{t+h}^{\text{path-avg}}$) instead of targeting the average growth rate directly ($\hat{y}_{t+h}^{\text{direct}}$), keeping everything else fixed. SEs are HAC. These are the 95% confidence bands.

effects of the *MARX* transformation according to 60 different targets. The marginal contribution of *MARX* on the pseudo-R^2 depends a lot on models, horizons, and series. However, we remark that at the short-run horizons, when combined with nonlinear methods, it produces positive and significant effects. It particularly improves the forecast accuracy for real activity series like industrial production, labor market series, and income, even at larger horizons. For instance, the gains from using *MARX* with RF achieve 16% when predicting INDPRO at the $h = 3$ horizon, and 14% in the case of employment if $h = 6$. When used with linear methods, the estimates are more often on the negative side, except for inflation rates and M2 at short horizons, and a few special cases at the one and two-year ahead horizons.

Direct vs Path Average. Fig. 2 reports the most unequivocal result of this paper: $\hat{y}_{t+h}^{\text{direct}}$ can prove largely suboptimal to $\hat{y}_{t+h}^{\text{path-avg}}$. For every method using a high-dimensional Z_t shrunk in some way, i.e., *not* the OLS-based AR and FM, $\hat{y}_{t+h}^{\text{path-avg}}$ will do significantly better than the direct approach, with $\alpha_{\text{path-avg}}^{(h,v)}$ sometimes around 30% and highly statistically significant. As mentioned earlier, those gains are most prevalent for the highly cyclical variables and longer horizons. Cases where $\hat{y}_{t+h}^{\text{path-avg}} \prec \hat{y}_{t+h}^{\text{direct}}$ are rare and usually not statistically significant at the 5% level, except for AR and FM which are both fitted by OLS.

How to explain this phenomenon? Aggregating separate horizon forecasts allows to leverage the "bet on sparsity" principle of Hastie et al. (2015). Presume the model for $\widehat{\Delta Y}_{t+h'}$ is sparse for each h', yet different. This implies that the *direct* model for $\hat{y}_{t+h}^{\text{direct}}$ is dense, and a much harder problem to learn. RF, BT, and Lasso will all perform better under sparsity, as every model struggle in a truly dense environment (unless it has a factor structure, upon which it becomes sparse in rotated space). An implication of this is that one should, as much as possible, try to make the problem sparse. Yet, whether sparsity will be more prevalent for $\hat{y}_{t+h}^{\text{path-avg}}$ or $\hat{y}_{t+h}^{\text{direct}}$ depends on true DGP. The evidence from Fig. 2 suggests that DGPs favoring $\hat{y}_{t+h}^{\text{path-avg}}$ are more prevalent in our experiment. What do those look like?

We find it useful to connect this question to recent works on forecasts aggregation, like Bermingham and D'Agostino (2014) who forecast the year on year inflation and compare two strategies: forecasting overall inflation directly vs forecasting individual elements of the consumption basket and using a weighted average of forecasts. They find that using more components and aggregating individual forecasts improves performance.[17]

17 In a similar vein, Marcellino et al. (2003) found that forecasting inflation at the country level and then aggregating the forecasts increases does better than forecasting at the aggregate level (Euro).

They provide a simple example to rationalize their result: forecasting an aggregate variable made of two series with differing levels of persistence using only past values of the aggregate will be misspecified. In ML forecasting context, where Z contains "everything" anyway, this problem translates from misspecification into making once sparse problems into a dense one, which is harder to learn. Consider a toy multi-horizon problem

$$\begin{aligned} \Delta Y_{t+h'} &= \beta_h X_{t,k^*(h')} + \epsilon_{t+h'}, \ h' = 1, 2 \\ y_{t+2} &= \frac{\Delta Y_{t+2} + \Delta Y_{t+1}}{2} \\ \Rightarrow y_{t+2} &= \frac{\beta_1}{2} X_{t,k^*(1)} + \frac{\beta_2}{2} X_{t,k^*(2)} + \frac{\epsilon_{t+1} + \epsilon_{t+2}}{2}. \end{aligned} \tag{13}$$

where one needs to select a single predictor for each horizon. In this simple analogy to a high-dimensional problem, unless $k^*(1) = k^*(2)$, that is, the optimally selected regressor is the same for both horizon, the direct approach implies a "denser" problem – estimating two coefficients rather than one for separate regressions. A scaled-up version of this is that if each horizon along the path implies 25 non-overlapping predictors, then the average growth rate model should have $25 \times h$ predictors, a much harder learning problem.

Of course, the $\hat{y}^{\text{direct}}_{t+h}$ approach might work better, even in a ML environment. For instance, the "aggregated" error term in (13) could have a lower variance if $\text{Corr}(\epsilon_{t+1}, \epsilon_{t+2}) < 0$. Note that this would not imply substantial differences in the OLS paradigm since such errors would rather average out at the aggregation step in $\hat{y}^{\text{path-avg}}_{t+h}$. However, if a regularization level must be picked by cross-validation (like Lasso's λ), an environment where there is a strong common component across h''s for the conditional mean could favor $\hat{y}^{\text{direct}}_{t+h}$. The reason for this is that choosing a regularization level optimized for a single horizon h' could be different than what may be optimal for the final averaged prediction – as exemplified by our ridge regression case of Eqs. (4) and (5). This observation is closely related to that of Granger (1987) who shows that the behavior of the aggregate series can easily be dominated by a common component **even if** it is unimportant for each of the microeconomic unit being aggregated. Translated to our ML-based multi-horizon problem, this means we want to avoid having overly harsh regularization throwing out negligible effects for a given h' whose accumulation overall h''s makes them non-negligible. Thus, if the noise level is much higher for single horizons forecasts, an overly strong $\lambda_{h'}$ for each h' may be chosen whereas λ_h for $\hat{y}^{\text{direct}}_{t+h}$ could be milder and allow for otherwise neglected signals to come through.

These potential explanations are illustrated using variable importance (VI) in Fig. 3. As shown earlier, the path average approach has outperformed the direct one when predicting real activity variables. VI measures in top panels show how models for $\hat{y}^{\text{path-avg}}_{t+h}$ use a much more polarized set of variables whereas those aiming for $\hat{y}^{\text{direct}}_{t+h}$ using a very diverse set of predictors in case of Income and Employment. This shed light on our bet-on-sparsity conjecture, i.e. that $\hat{y}^{\text{path-avg}}_{t+h}$ will have the upper hand if $\Delta\hat{Y}_{t+h'}$ predictive problems are quite heterogenous. In both cases, horizon 1 is quite different from 2-3-4, which also differ from the 5–12 block. It is noted in Figs. 8 and 15 that $\hat{y}^{\text{path-avg}}_{t+h}$ visibly demonstrate a better capacity for autoregressive behavior (even at $h = 12$) which provides it with a clear edge over $\hat{y}^{\text{direct}}_{t+h}$ during recessions. Interestingly, the foundation for this finding is also visible in Fig. 3 for real activity variables: $\hat{y}^{\text{path-avg}}_{t+h}$ reliance on plain AR terms is more than twice that of $\hat{y}^{\text{direct}}_{t+h}$.

The bottom panels show VI measures for CPI inflation and M2 growth. Recall that $\hat{y}^{\text{path-avg}}_{t+h} \prec \hat{y}^{\text{direct}}_{t+h}$ was unambiguous for those variables. Here again, results are in line with the above arguments. The retained predictors' sets are much more *similar* across the two approaches, which results from the presence of a strong common component over horizons (i.e., persistence which constitutes about 75% of normalized VI), which favors $\hat{y}^{\text{direct}}_{t+h}$.

MAF. Fig. 4 plots the distribution of $\alpha^{(h,v)}_{MAF}$, conditional on including X in the model. The motivation for that is that *MAF*, by construction, summarizes the entirety of $[X_{t-p}]_{p=1}^{p=P_{MAF}}$ with no special emphasis on the most recent information.[18] Thus, it is better-advised to always include the raw X with *MAF*, so recent information may interact with the lag polynomial summary if ever needed. *MAF* contributions are overall more muted than that of *MARX*, except when used with the Linear Boosting method. Nevertheless, it is noticed that it shares common gains with the latter as short horizons ($h = 3, 6$) of real activity variables also benefit from it. More convincing improvements are observed for retail sales at the 2-year horizons for nonlinear methods.

Traditional Factors. It has already been documented that factors matter – and a lot (Stock & Watson, 2002a, 2002b). Fig. 5 allows us to evaluate their quantitative effects. Including a handful of factors rather than all of (stationary) X improves substantially and significantly forecast accuracy. The case for this is even stronger when those are used in conjunction with nonlinear methods, especially for prediction at longer horizons. This finding supports the view that a factor model is an accurate depiction of the macroeconomy, as originally suggested in the works of Sargent and Sims (1977) and Geweke (1977) and later expanded in various forecasting and structural analysis applications (Bernanke et al., 2005; Stock & Watson, 2002a). In this line of thought, transforming X into F is not merely a mechanical dimension reduction step. Rather, it is meaningful feature engineering uncovering true latent factors which contain most, if not all, the relevant information about the current state of the economy. Once F's are extracted, the standard diffusion indexes model of Stock and Watson (2002b) can either be upgraded by using linear methods performing variable selection, or nonlinear functional form approximators such as Random Forests and Boosted Trees.

[18] Of course, one could alter the PCA weights in *MAF* to introduce priority on recent lags à la Minnesota-prior, but we leave that possibility for future research.

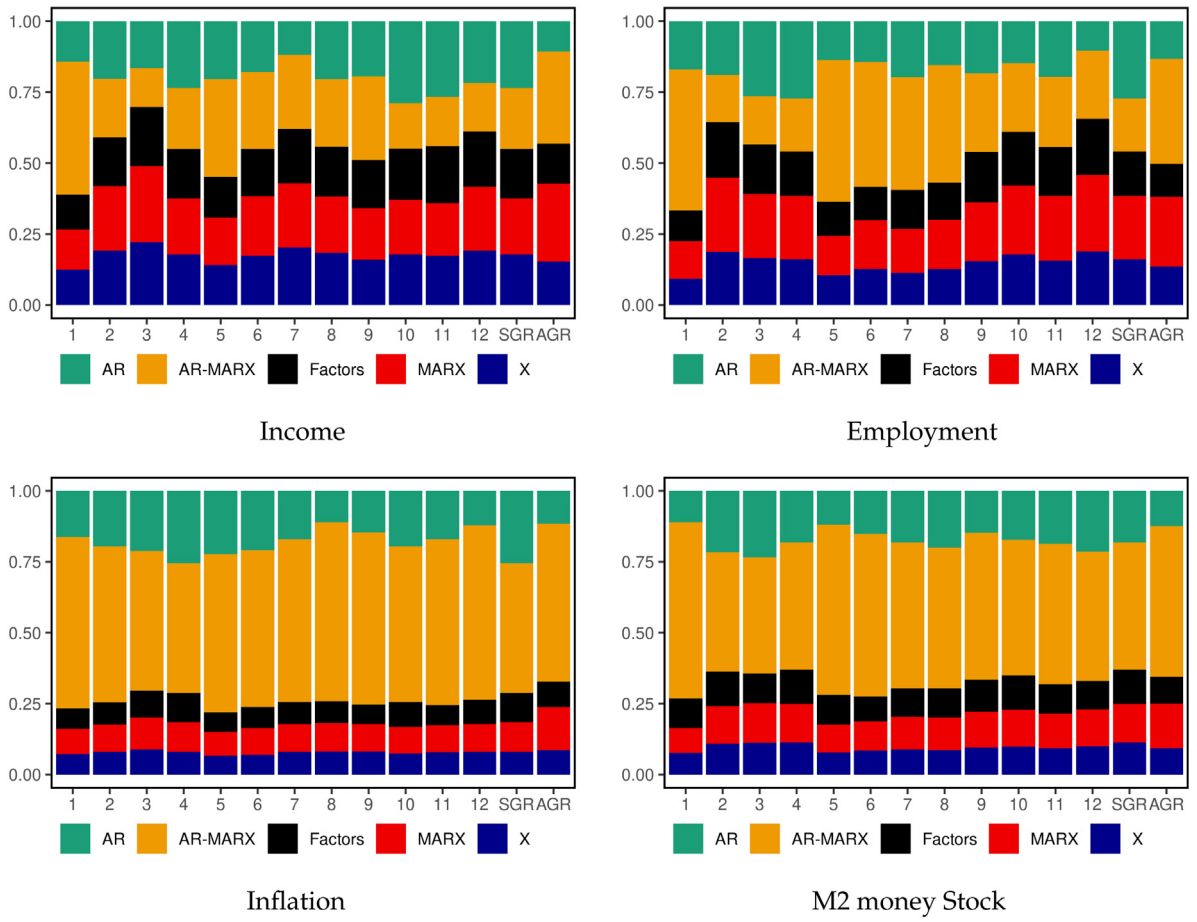


Fig. 3. Variable Importance. Notes: This figure displays the relative variable importance (VI) measures for the Random Forest F-X-MARX model for horizon $H = 12$. Group values are additions of VI for individual series weighted by the share of each groups with the total VI normalized to 1. The first 12 bars reflect horizon-wise differences for the $\hat{y}_{t+h}^{\text{path-avg}}$ models whose forecasts are accumulated and the subsequent bar shows the average importance across those horizons. The last bar displays the equivalent for the $\hat{y}_{t+h}^{\text{direct}}$ model.

4.2. Case study

In this section, we conduct "event studies" to highlight more explicitly the importance of data pre-processing when predicting real activity and inflation indicators. Fig. 6 plots cumulative squared errors for three cases where specific transformations stand out. On the left, we compare the performance of RF when predicting industrial production growth three months ahead, using either *F*, *X* or *F-X-MARX* as feature matrix. The middle panel shows the same exercise for employment growth. On the right, we report one-year ahead CPI inflation forecasts. Industrial production and employment examples document the merits of including *MARX*: its cumulatively summed squared errors (when using RF) are always below the ones produced by using *F* and *X*. The gap widens slowly until the Great Recession, after which it increases substantially. As discussed in Section 4, using common factors with RF constitutes the optimal specification for CPI inflation. Fig. 6 illustrates this finding and shows that the gap between using *F* or *X* widens during the mid-80s, the mid-90s, and just before the Great Recession. To provide a statistical assessment of the stability of forecast accuracy, we consider the fluctuation test of Giacomini and Rossi (2010) in Appendix C.

In Fig. 7, we look more closely at each model's forecasts during the last three recessions and subsequent recoveries. Specifically, we plot the 3-month ahead forecasts for the period covering 3 months before, and 24 months after a recession, for industrial production and employment. The forecasting models are all RF-based and differ by their use of either *F*, *X* or *F-X-MARX*. On the right side, we show the RMSE ratio of each RF specification against the benchmark FM model for the whole POOS and the episode under analysis. In the case of industrial production, the *F-X-MARX* specification outperforms the others during the Great Recession and its aftermath and improves even more upon the benchmark model compared to the full POOS period. We observe on the left panel that forecasts made with *F-X-MARX* are much closer to realized values at the end of the recession and during the recovery. The situation is qualitatively similar during the 2001 recession but the effects are smaller. Including *MARX* also emerges as the best alternative around the

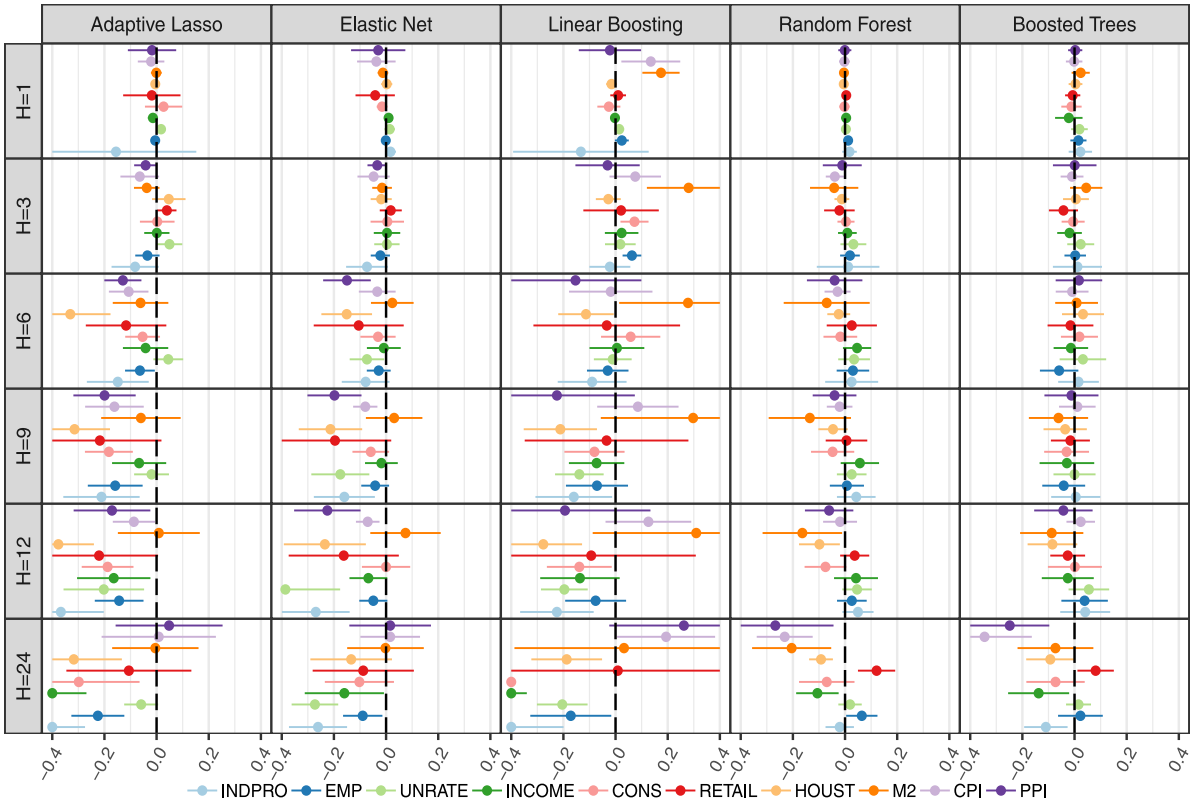


Fig. 4. Distribution of *MAF* Marginal Effects. Notes: This figure plots the distribution of $\alpha_f^{(h,v)}$ from Eq. (12) done by (h, v) subsets. That is, it shows the average partial effect on the pseudo-R^2 from augmenting the model with *MAF* featuring, keeping everything else fixed. SEs are HAC. These are the 95% confidence bands.

1990–1991 recession, but the benchmark model is more competitive for this particular episode.

In the case of employment showcased in Figure 14 in Appendix E, *MARX* again supplants *F* or *X* in all three recessions. For instance, around the Dotcom bubble burst, it displays an outstanding performance, surpassing the benchmark by 40%. However, during the Great Recession, it is outperformed by the traditional factor model. Finally, the *F-X-MARX* combination provides the most accurate forecast during and after the credit crunch recession of the early 1990s.

Fig. 8 illustrate the relative performance of the two target transformations for employment and income 12 months ahead. Again, we focus on the three most recent recession episodes. $\hat{y}_{t+h}^{\text{path-avg}}$ dramatically improves performance over $\hat{y}_{t+h}^{\text{direct}}$ and much of that edge visibly comes from adjusting itself more or less rapidly to new economic conditions. In contrast, $\hat{y}_{t+h}^{\text{direct}}$ is extremely smooth and report something close to the long-run average. Since the last three recessions were characterized by a slow recovery, $\hat{y}_{t+h}^{\text{path-avg}}$ procures much more credible forecasts of employment and income simply by catching up sooner with realized values. This behavior is understandable through the lenses of Fig. 3 where early horizons of $\hat{y}_{t+h}^{\text{path-avg}}$ make a pronounced use of autoregressive terms for both employment (and income, see Figure 15 in Appendix E).

4.3. Extraneous transformations

We evaluate four additional data transformation strategies in combination with direct and path average targets. First, we accommodate for the presence of error correction terms (ECM) by considering the Factor-augmented ECM approach of Banerjee et al. (2014) and include *level* factors estimated from $I(1)$ predictors. Second, we consider volatility factors and data inspired by Gorodnichenko and Ng (2017), where both factors from X^2 and X^2 itself are included as predictors. Third, we evaluate the potential predictive gains from including (Forni et al., 2005)'s dynamic factors in Z.

Figure 10, in Appendix D, reports the distribution of average marginal effects of adding level factors in the predictors' set Z. Their impact is generally small and not significant at short horizons, while it depends on methods and forecasting approaches at longer horizons. In the case of the direct average approach, as depicted in panel 10a, adding level factors generally deteriorates the predictive performance except for M2 with nonlinear methods. The effects are qualitatively similar when the target is achieved by the path average approach, as shown in 10b.

Adding volatility data and factors is generally harmful with linear methods and has almost no significant impact when random forest and boosted trees are used, see

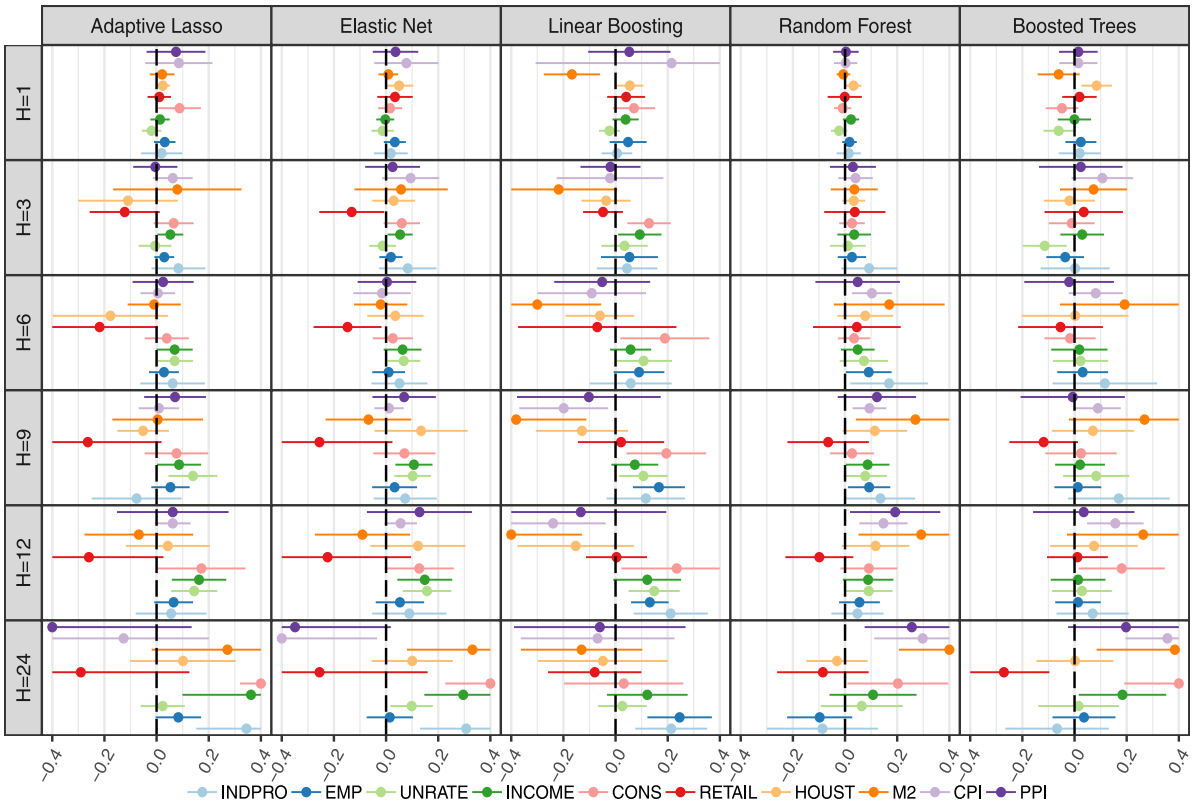


Fig. 5. Distribution of F Marginal Effects. Notes: This figure plots the distribution of $\alpha_f^{(h,v)}$ from Eq. (12) done by (h, v) subsets. That is, it shows the partial effect on the pseudo-R^2 from considering only F featuring versus including only observables X. SEs are HAC. These are the 95% confidence bands.

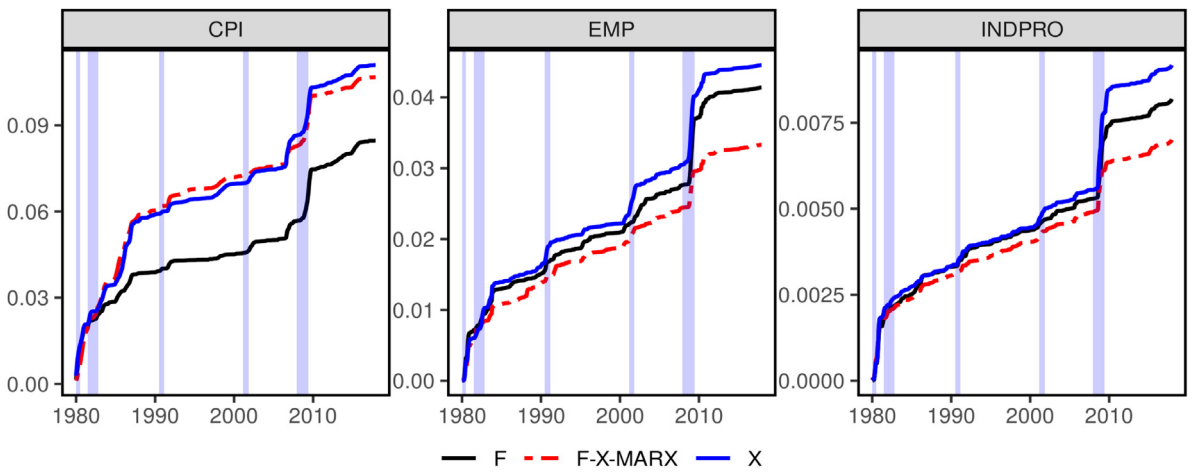
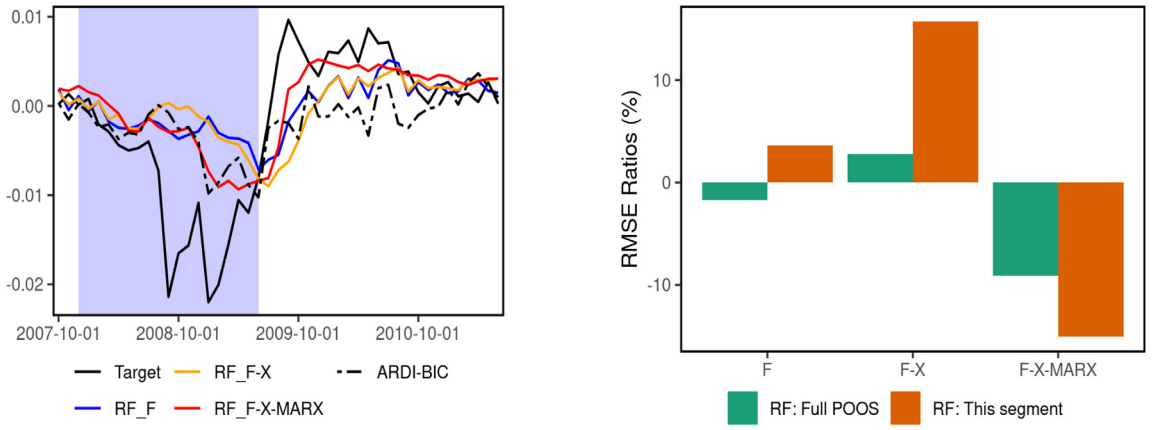
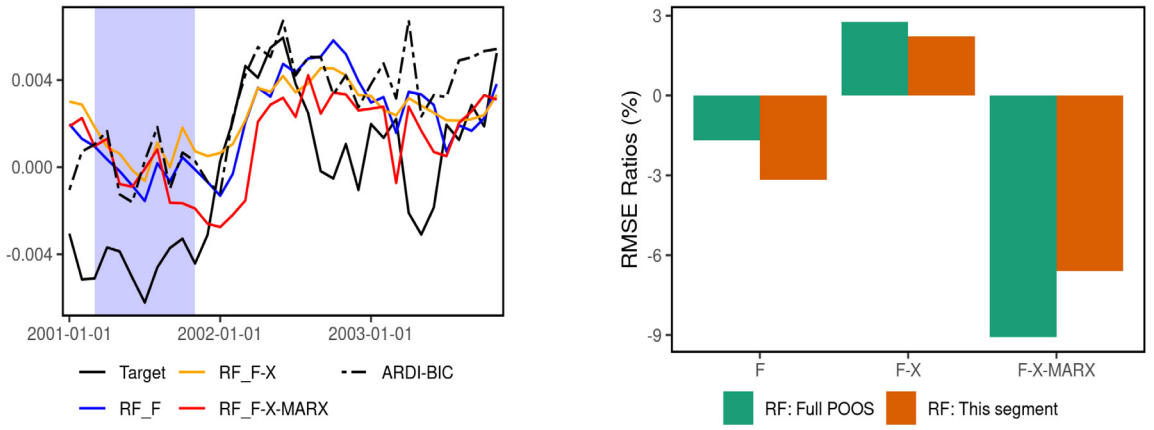


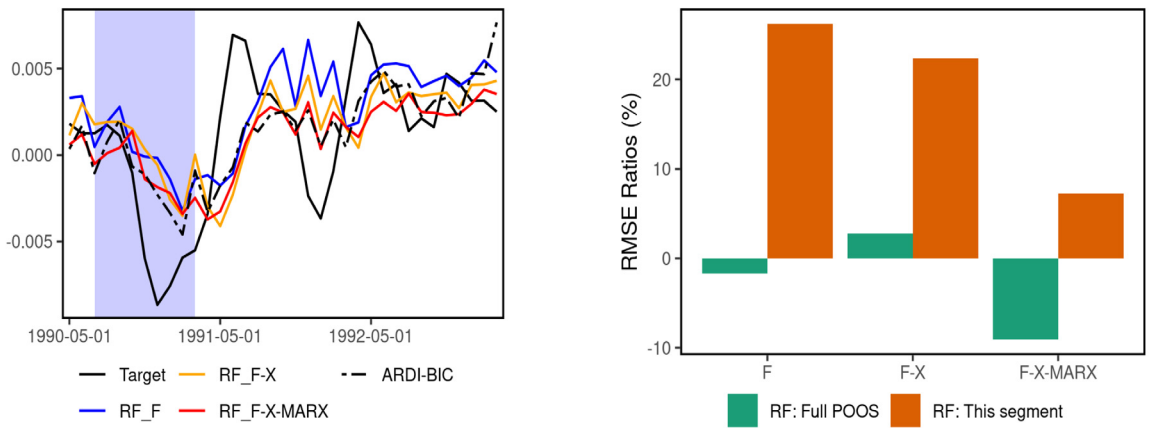
Fig. 6. Cumulative Squared Error (Direct). Notes: Cumulative squared forecast errors for INDPRO and EMP (3 months) and CPI (12 months). All use the Random Forest model and the direct approach. CPI and EMP have been scaled by 100.



(a) Recession Episode of 2007-12-01

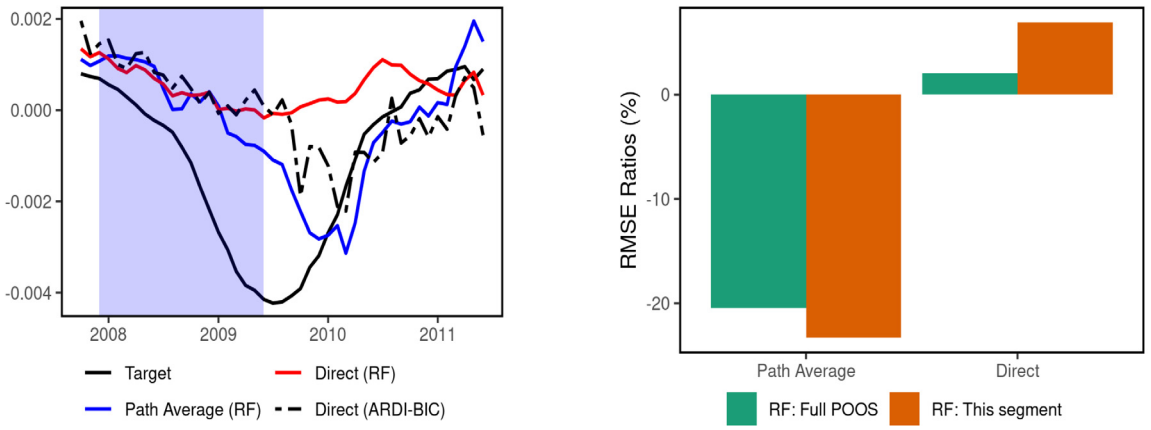


(b) Recession Episode of 2001-03-01

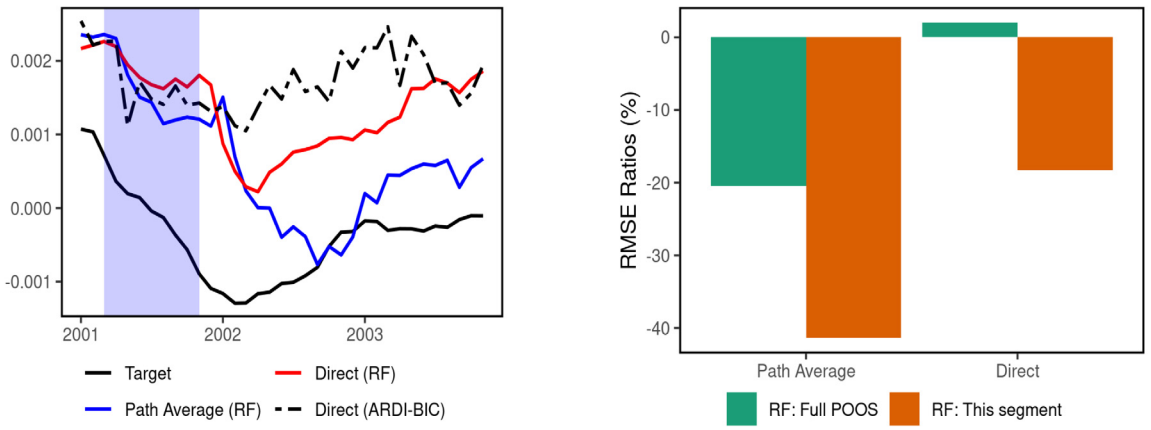


(c) Recession Episode of 1990-07-01

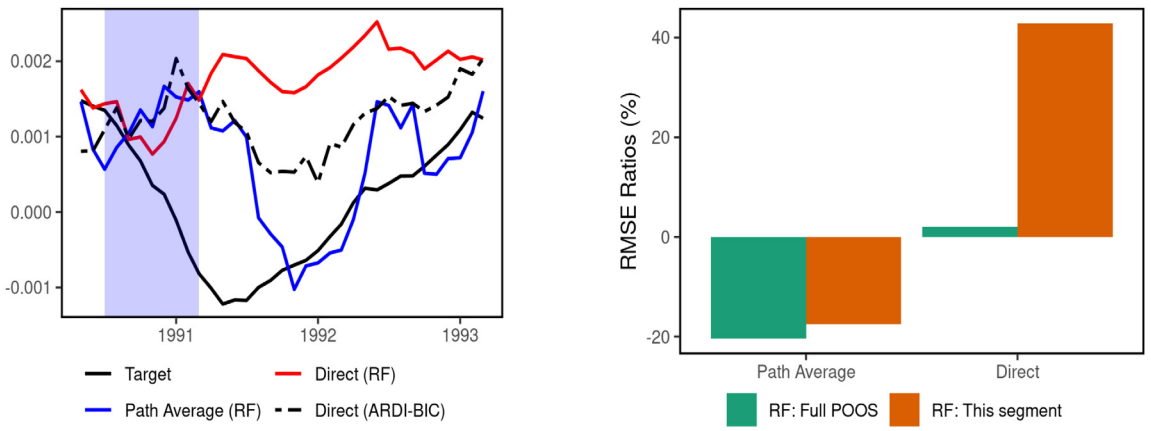
Fig. 7. Case of Industrial Production (Direct). Notes: The figure covers 3 months before and 24 months after the recession. RMSE ratios are relative to FM model and the episode RMSE refers to the visible time period.



(a) Recession Episode of 2007-12-01



(b) Recession Episode of 2001-03-01



(c) Recession Episode of 1990-07-01

Fig. 8. Case of Employment (Path Average). Notes: The figure plots 12-month ahead forecasts for the period covering 3 months before and 24 months after the recession. RMSE ratios are relative to FM model for average growth rates and the episode RMSE refers to the visible time period and Random Forest models use F-X-MARX.

Figure 11.[19] Hence, letting ML methods generate nonlinearities proves to be more resilient than including simple power terms. This also suggests that volatility or other uncertainty proxies may not be the major sources of nonlinearities for macroeconomic dynamics since they would otherwise be an indispensable form of feature engineering from which variable selection algorithms build their predictions.

Finally, Figures 12 and 13 evaluate the marginal predictive content of dynamic factors as opposed to MAF and static factors (PCs) respectively. Considering dynamic factors as opposed to MAF improves the predictability at longer horizons when used to construct $\hat{y}_{t+h}^{\text{direct}}$, while their effects are rather small with $\hat{y}_{t+h}^{\text{path-avg}}$. When it comes to the choice between dynamic and static factors, the results are in general quantitatively small but suggest that standard principal components are preferred, especially in combination with nonlinear methods, which is analogous to the findings of Boivin and Ng (2005) in linear environments.

5. Conclusion

This paper studies the virtues of standard and newly proposed data transformations for macroeconomic forecasting with machine learning. The classic transformations comprise the dimension reduction of stationarized data through principal components and the inclusion of level variables to take into account low-frequency movements. Newly proposed avenues include moving average factors (MAF) and moving average rotation of X (MARX). The last two were motivated by the need to compress the information within a lag polynomial, especially if one desires to keep X close to its original – interpretable – space. In addition to the aforementioned transformations focusing on X, we considered two pre-processing alternatives for the target variable, namely the direct and path average approaches.

To evaluate the contribution of data transformations for macroeconomic prediction, we have considered three linear and two nonlinear ML methods (Elastic Net, Adaptive Lasso, Linear Boosting, Random Forests, and Boosted Trees) in a substantive pseudo-out-of-sample forecasting exercise was done over 38 years for 10 key macroeconomic indicators and 6 horizons. With the different permutations of f_Z's available from the above, we have analyzed a total of 15 different information sets. The combination of standard and non-standard data transformations (*MARX*, *MAF*, *Level*) is shown to minimize the RMSE, particularly at shorter horizons. Those consistent gains are usually obtained when a nonlinear nonparametric ML algorithm is being used. This is precisely the algorithmic environment we conjectured could benefit most from our proposed f_Z's. Additionally, traditional factors are featured in the overwhelming majority of best information sets for each target. Therefore, while ML methods can handle the high-dimensional X (both computationally and statistically), extracting common factors remains straightforward feature engineering that works.

The way the prediction is constructed can make a great difference. The path average approach is more accurate than the direct one for almost all real activity variables (and at various horizons). The gains can be as large as 30% and are mostly observed when the path average approach is used in conjunction with regularization and/or nonparametric nonlinearity.

As the number of researchers and practitioners in the field is ever-growing, we believe those insights constitute a strong foundation on which stronger ML-based systems can be developed to further improve macroeconomic forecasting.

[19] The very weak contribution of volatility terms to BT or RF is expected given that those transformations are locally monotone (i.e, for all points where $X_{k,t} > 0$ or $X_{k,t} < 0$) and trees are invariant to monotone transformations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ijforecast.2021.05.005.

References

Almon, S. (1965). The distributed lag between capital appropriations and expenditures. *Econometrica*, 178–196.

Bai, J., & Ng, S. (2004). A PANIC attack on unit roots and cointegration. *Econometrica*, *72*(4), 1127–1177.

Bańbura, M., Giannone, D., & Reichlin, L. (2010). Large Bayesian vector auto regressions. *Journal of Applied Econometrics*, *25*(1), 71–92.

Banerjee, A., Marcellino, M., & Masten, I. (2014). Forecasting with factor-augmented error correction models. *International Journal of Forecasting*, *30*(3), 589–612.

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

Bermingham, C., & D'Agostino, A. (2014). Understanding and forecasting aggregate and disaggregate price dynamics. *Empirical Economics*, *46*(2), 765–788.

Bernanke, B., Boivin, J., & Eliasz, P. (2005). Measuring the effects of monetary policy: a factor-augmented vector autoregressive (FAVAR) approach. *Quarterly Journal of Economics*, *120*, 387–422.

Boivin, J., & Ng, S. (2005). Understanding and comparing factor-based forecasts. *International Journal of Central Banking*, *1*, 117–151.

Borup, D., Christensen, B. J., Mühlbach, N. N., Nielsen, M. S., et al. (2020). Targeting predictors in random forest regression. *Technical report*, Department of Economics and Business Economics, Aarhus University.

Carriero, A., Galvão, A. B., & Kapetanios, G. (2019). A comprehensive evaluation of macroeconomic forecasting methods. *International Journal of Forecasting*, *35*(4), 1226–1239.

Chan, N., & Wang, Q. (2015). Nonlinear regressions with nonstationary time series. *Journal of Econometrics*, *185*(1), 182–195.

Choi, I. (2015). *Almost all about unit roots: foundations, developments, and applications*. Cambridge University Press.

Christoffersen, P. F., & Diebold, F. X. (1998). Cointegration and long-horizon forecasting. *Journal of Business & Economic Statistics*, *16*(4), 450–456.

Cook, T., & Hall, A. S. (2017). Macroeconomic indicator forecasting with deep neural networks. *Technical report*, Federal Reserve Bank of Kansas City, Research Working Paper.

Diebold, F. X., & Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *20*(1), 134–144.
Doan, T., Litterman, R., & Sims, C. (1984). Forecasting and conditional projection using realistic prior distributions. *Econometric Reviews*, *3*(1), 1–100.
Elliott, G. (2006). Forecasting with trending data. In G. Elliott, C. Granger, & A. Timmermann (Eds.), *Handbook of economic forecasting*: *Vol. 1*, *Handbook of economic forecasting* (pp. 555–604). Elsevier, https://ideas.repec.org/h/eee/ecofch/1-11.html.
Elliott, G., Gargano, A., & Timmermann, A. (2013). Complete subset regressions. *Journal of Econometrics*, *177*(2), 357–373.
Engle, R. F., & Granger, C. W. (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica*, 251–276.
Engle, R. F., & Yoo, B. S. (1987). Forecasting and testing in co-integrated systems. *Journal of Econometrics*, *35*(1), 143–159.
Forni, M., Hallin, M., Lippi, M., & Reichlin, L. (2005). The generalized dynamic factor model: One-sided estimation and forecasting. *Journal of the American Statistical Association*, *100*(471), 830–840.
Geweke, J. (1977). Latent variables in socio-economic models. In D. J. Aigner, & A. S. Goldberger (Eds.), *The dynamic factor analysis of economic time series* (pp. 365–383). North-Holland Publishing Company.
Ghysels, E., Santa-Clara, P., & Valkanov, R. (2004). The MIDAS touch: Mixed data sampling regression models.
Giacomini, R., & Rossi, B. (2010). Forecast comparisons in unstable environments. *Journal of Applied Econometrics*, *25*(4), 595–620.
Gorodnichenko, Y., & Ng, S. (2017). Level and volatility factors in macroeconomic data. *Journal of Monetary Economics*, *91*, 52–68.
Goulet Coulombe, P. (2020). To bag is to prune.
Goulet Coulombe, P. (2020a). The macroeconomy as a random forest. ArXiv Preprint arXiv:2006.12724.
Goulet Coulombe, P. (2020b). Time-varying parameters as ridge regressions.
Goulet Coulombe, P., Leroux, M., Stevanovic, D., & Surprenant, S. (2019). How is machine learning useful for macroeconomic forecasting? *Technical report*, CIRANO Working Papers, 2019s-22.
Goulet Coulombe, P., Marcellino, M., & Stevanovic, D. (2021). Can machine learning catch the COVID-19 recession? *Technical report*, CIRANO Working Papers, 2021s-09.
Granger, C. W. (1987). Implications of aggregation with common factors. *Econometric Theory*, *3*(2), 208–222.
Hall, A. D., Anderson, H. M., & Granger, C. W. (1992). A cointegration analysis of treasury bill yields. *The Review of Economics and Statistics*, 116–126.
Hansen, P. R., Lunde, A., & Nason, J. M. (2011). The model confidence set. *Econometrica*, *79*(2), 453–497.
Hassani, H., Heravi, S., & Zhigljavsky, A. (2009). Forecasting European industrial production with singular spectrum analysis. *International Journal of Forecasting*, *25*(1), 103–118.
Hassani, H., Soofi, A. S., & Zhigljavsky, A. (2013). Predicting inflation dynamics with singular spectrum analysis. *Journal of the Royal Statistical Society: Series a (Statistics in Society)*, *176*(3), 743–760.
Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.
Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.
Kim, H. H., & 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.
Koop, G. M. (2003). *Bayesian econometrics*. John Wiley & Sons Inc..
Kotchoni, R., Leroux, M., & Stevanovic, D. (2019). Macroeconomic forecast accuracy in a data-rich environment. *Journal of Applied Econometrics*, *34*(7), 1050–1072.
Kuhn, M., & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. CRC Press.
Lee, J. H., Shi, Z., & Gao, Z. (2018). On LASSO for predictive regression. ArXiv Preprint arXiv:1810.03140.
Marcellino, M., Stock, J. H., & Watson, M. W. (2003). Macroeconomic forecasting in the euro area: Country specific versus area-wide information. *European Economic Review*, *47*(1), 1–18.
McCracken, M. W., & Ng, S. (2016). FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, *34*(4), 574–589.
McCracken, M., & Ng, S. (2020). FRED-QD: A quarterly database for macroeconomic research. *Technical report*, National Bureau of Economic Research.
Medeiros, M. C., Vasconcelos, G. F., Veiga, A., & Zilberman, E. (2019). Forecasting inflation in a data-rich environment: the benefits of machine learning methods. *Journal of Business & Economic Statistics*, 1–22.
Mentch, L., & Zhou, S. (2019). Randomization as regularization: A degrees of freedom explanation for random forest success. ArXiv Preprint arXiv:1911.00190.
Olson, M. A., & Wyner, A. J. (2018). Making sense of random forest probabilities: a kernel perspective. ArXiv Preprint arXiv:1812.05792.
Peña, D., & Poncela, P. (2006). Nonstationary dynamic factor analysis. *Journal of Statistical Planning and Inference*, *136*(4), 1237–1257.
Phillips, P. C. (1991). Optimal inference in cointegrated systems. *Econometrica*, 283–306.
Phillips, P. C. (1991). To criticize the critics: An objective Bayesian analysis of stochastic trends. *Journal of Applied Econometrics*, *6*(4), 333–364.
Rodriguez, J. J., Kuncheva, L. I., & Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(10), 1619–1630.
Sargent, T., & Sims, C. (1977). Business cycle modeling without pretending to have too much a priori economic theory. In C. Sims (Ed.), *New methods in business cycle research*. Minneapolis: Federal Reserve Bank of Minneapolis.
Shiller, R. J. (1973). A distributed lag estimator derived from smoothness priors. *Econometrica*, 775–788.
Sims, C. A. (1988). Bayesian skepticism on unit root econometrics. *Journal of Economic Dynamics and Control*, *12*(2–3), 463–474.
Sims, C. A., Stock, J. H., & Watson, M. W. (1990). Inference in linear time series models with some unit roots. *Econometrica*, 113–144.
Sims, C. A., & Uhlig, H. (1991). Understanding unit rooters: A helicopter tour. *Econometrica*, 1591–1599.
Stock, J. H., & Watson, M. W. (1998). A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series. *Technical report*, National Bureau of Economic Research.
Stock, J. H., & Watson, M. W. (2002a). Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, *97*(460), 1167–1179.
Stock, J. H., & Watson, M. W. (2002b). Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, *20*(2), 147–162.
Tibshirani, R., Wainwright, M., & Hastie, T. (2015). *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC.