UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
FACULDADE DE CIÊNCIAS ECONÔMICAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ECONOMIA

KLAUS BÖESCH

STATISTICAL LEARNING METHODS FOR TIME SERIES FORECASTING

Porto Alegre
2020

KLAUS BÖESCH

STATISTICAL LEARNING METHODS FOR TIME SERIES FORECASTING

Dissertação submetida ao Programa de Pós-Graduação em Economia da Faculdade de Ciências Econômicas da UFRGS, como requisito parcial para obtenção do título de Mestre em Economia, área de concentração: Economia Aplicada.

Orientador: Prof. Dr. Flávio A. Ziegelmann

Porto Alegre
2020

CIP - Catalogação na Publicação

Boesch, Klaus
Statistical learning methods for time series forecasting / Klaus Boesch. -- 2020.
64 f.
Orientador: Flavio Augusto Ziegelmann.

Dissertação (Mestrado) -- Universidade Federal do Rio Grande do Sul, Faculdade de Ciências Econômicas, Programa de Pós-Graduação em Economia, Porto Alegre, BR-RS, 2020.

1. Time series. 2. Statistical learning. 3. Machine learning. 4. High-dimensional methods. 5. Inflation forecasting. I. Ziegelmann, Flavio Augusto, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

KLAUS BÖESCH

STATISTICAL LEARNING METHODS FOR TIME SERIES FORECASTING

Dissertação submetida ao Programa de Pós-Graduação em Economia da Faculdade de Ciências Econômicas da UFRGS, como requisito parcial para obtenção do título de Mestre em Economia, área de concentração: Economia Aplicada.

Aprovada em: Porto Alegre, 25 de março de 2020.

BANCA EXAMINADORA:

Prof. Dr. Flávio Augusto Ziegelmann - Orientador
PPGE/UFRGS

Prof. Dr. Cristiano Augusto Coelho Fernandes
DEE/PUC-Rio

Prof. Dr. Rodrigo dos Santos Targino
EMAp/FGV

Prof. Dr. Eduardo de Oliveira Horta
PPGE/UFRGS

RESUMO

Métodos para alta dimensão tem ganhado cada vez mais importância na literatura econométrica, onde a inclusão de um grande número de variáveis econômicas e financeiras pode melhorar significativamente o desempenho de previsões de séries temporais. Porém, em um contexto de alta dimensão, métodos tradicionais enfrentam alguns problemas relacionados à interpretabilidade do modelo, variância elevada, correlação espúria, eficiência computacional, entre outros. Métodos de aprendizado estatístico são desenvolvidos para um melhor desempenho de previsão, combinando técnicas de regularização e ajuste empírico dos parâmetros e são capazes de lidar com alguns destes problemas na presença de conjuntos de dados de alta dimensão. No artigo que integra esta dissertação, utilizamos uma variedade de métodos de aprendizado estatístico para previsão de séries temporais, realizando diversos exercícios numéricos, incluindo simulação Monte Carlo e análise empírica de dados. Propusemos um método chamado WLadaENet (weighted lag adaptive Elastic Net), que combina regularização quadrática com penalização ponderada semelhante ao método adaENet, porém, penaliza mais severamente coeficientes de variáveis com defasagens mais elevadas, como o método WLadaLASSO. Em nossas simulações, o método WLadaENet apresenta um bom desempenho em termos de seleção de variáveis quando o modelo é esparso e em termos de previsão fora da amostra, mesmo quando o modelo não é esparso e apresenta não linearidades. Em nossa primeira aplicação empírica realizamos previsões entre 1 e 12 meses da inflação do Brasil e do núcleo de inflação, incluindo previsões da inflação acumulada. Na segunda aplicação realizamos previsões para a inflação dos Estados Unidos. O método Ridge apresenta um bom desempenho para as previsões da inflação brasileira e um desempenho moderado para as previsões da inflação norte americana utilizando-se amostras de tamanho semelhante. Porém, quando utilizada uma amostra muito maior para a previsão da inflação norte americana, o desempenho método Ridge reduz drasticamente, enquanto o desempenho do método L_2Boost melhora significativamente, principalmente pra a inflação acumulada. O método proposto no artigo, WLadaENet, também apresenta um bom desempenho para previsão da inflação norte americana neste caso.

Palavras-chave: Aprendizado estatístico. Aprendizado de máquina. Métodos para alta dimensão. Previsão de inflação. LASSO. Random forests. Boosting.

ABSTRACT

High-dimensional methods are getting more and more present in the literature and the inclusion of a large set of economic and financial predictors can improve the time series forecasting performance significantly. However, in high-dimensional context traditional methods face some challenges related to the model‘s interpretability, high variance, spurious correlation, computational efficiency, among others. Statistical learning methods, which are designed to improve out-of-sample prediction by combining regularization and empirical tuning, are able to handle some of these issues in high-dimensional context. In the paper contained into this dissertation we employ a variety of high-dimensional statistical learning methods in order to perform time series forecasting, carrying out a simulation study and presenting two empirical applications. A method we call WLadaENet (weighted lag adaptive Elastic Net) is proposed, which combines quadratic regularization and the adaptive weighted LASSO shrinkage similarly to adaENet, but further penalizes coefficients of higher-lagged variables like WLadaLASSO. In our Monte Carlo implementation, the WLadaENet presents a good performance in terms of variable selection when the model is sparse and in terms of forecasting even when the model is not sparse and nonlinearities are included. In the first application we perform forecasts of Brazilian inflation and the core inflation from 1 to 12 months ahead, including the accumulated inflation. In the second application we forecast U.S. inflation. We find that Ridge Regression has a good performance to forecast Brazilian inflation, and a moderate performance to forecast U.S. inflation when the data have (almost) the same size, in turn, when we use a larger sample the performance of Ridge Regression decreases while L_2Boost improves its performance, especially when accumulated inflation is considered. The method WLadaENet proposed in the paper also presents a good performance to forecast U.S. inflation in this case.

Keywords: Time series. Statistical learning. Machine learning. High-dimensional methods. Inflation forecasting. LASSO. Random forests. Boosting.

SUMÁRIO

1 INTRODUÇÃO

O artigo que integra esta dissertação consiste em um estudo metodológico e empírico de métodos de aprendizado estatístico para alta dimensão, para a realização de previsão de séries temporais. Aprendizado estatístico refere-se à uma variedade de ferramentas para a compreensão dos dados, podendo ser classificadas como supervisionadas ou não supervisionadas (JAMES *et al.*, 2013). Os métodos estudados aqui são classificados, em geral, como métodos de aprendizado estatístico supervisionado. Métodos para alta dimensão estão se tornando cada vez mais importantes na literatura e a inclusão de um grande número de variáveis financeiras e econômicas potencialmente contribui com ganhos consideráveis de desempenho em previsões de séries temporais.

Porém, em um contexto de alta dimensionalidade, os métodos tradicionais apresentam três principais problemas. O primeiro problema é que as estimativas frequentemente apresentam variância elevada, apesar de um baixo viés, reduzindo a exatidão das previsões. O segundo problema é relacionado à interpretabilidade do modelo. Considerando-se um grande conjunto de variáveis preditoras, é preferível determinar um subconjunto menor de variáveis que exibam os efeitos que melhor explicam a variabilidade da variável de resposta (TIBSHIRANI, 1996; KONZEN; ZIEGELMANN, 2016). O terceiro problema ocorre quando o número de variáveis preditoras excede o tamanho da amostra, neste caso o método de Mínimos Quadrados Ordinários (MQO) não pode ser implementado. Outro impacto da alta dimensionalidade é a correlação espúria (FAN, 2014): isto refere-se a variáveis que na teoria não são correlacionadas, mas cuja correlação amostral é elevada. Este problema pode resultar em inferências estatísticas inválidas.

Métodos de aprendizado estatístico são capazes de evitar ou contornar alguns dos problemas descritos acima, em um contexto de alta dimensionalidade. Estes métodos são desenvolvidos para um melhor desempenho de previsão fora da amostra. Gu *et al.* (2019) e Mullainathan e Spiess (2017) apontam as três principais características dos métodos de aprendizado estatístico para alta dimensão. Aprendizado estatístico (ou aprendizado de máquina) contém uma variada coleção de métodos de previsão estatística para alta dimensão que combinam dois elementos: regularização e ajuste empírico dos parâmetros. A natureza de alta dimensionalidade desses métodos realçam sua flexibilidade relativa às técnicas econométricas de predição mais tradicionais (GU *et al.*, 2019). Os métodos de aprendizado estatístico tipicamente têm um regularizador associado a eles e o ajuste empírico dos parâmetros permite a escolha do nível apropriado de regularização (MULLAINATHAN; SPIESS, 2017).

No artigo que compõe o Capítulo 2 empregamos uma variedade de métodos de aprendizado estatístico (incluindo métodos baseados em encolhimento dos coeficientes, em árvores de regressão e em *boosting*) para realização de previsão de séries temporais. Em particular, propusemos um método chamado WLadaENet (weighted lag adaptive Elastic Net), que combina regularização quadrática e a penalização ponderada adaptativa similar ao método adaENet introduzido por Zou e Zhang (2009), porém, penalizando mais severamente coeficientes das variáveis de defasagens mais elevadas, como o método WLadaLASSO proposto por Konzen e Ziegelmann

(2016). Para comparar o desempenho dos métodos de aprendizado estatístico, realizamos um estudo baseado em simulações onde são apresentadas três especificações de processos geradores de dados e variamos o número de defasagens utilizadas assim como o tamanho das amostras geradas. Também apresentamos duas aplicações empíricas, onde na primeira aplicação empírica utilizamos o conjunto de métodos de aprendizado estatístico para realizar previsões da inflação brasileira e do núcleo de inflação, enquanto na segunda aplicação, utilizamos estes métodos para realizar previsões da inflação norte americana.

2 STATISTICAL LEARNING METHODS FOR TIME SERIES FORECASTING

Klaus Böesch[1] and Flavio A. Ziegelmann [2]

March, 2020

Abstract. We perform time series forecasting employing a wide collection of high-dimensional statistical learning methods, carrying out a simulation study and presenting two empirical applications. A method we call WLadaENet (weighted lag adaptive Elastic Net) is proposed, which combines quadratic regularization and the adaptive weighted LASSO shrinkage similarly to adaENet, but further penalizes coefficients of higher-lagged variables like WLadaLASSO. In our Monte Carlo implementation, the WLadaENet presents a good performance in terms of variable selection when the model is sparse and in terms of forecasting even when the model is not sparse and nonlinearities are included. In the first application we perform forecasts of Brazilian inflation and the core inflation from 1 to 12 months ahead, including the accumulated inflation. In the second application we forecast U.S. inflation. We find that Ridge Regression has a good performance to forecast Brazilian inflation, and a moderate performance to forecast U.S. inflation when the data have (almost) the same size. In turn, when the sample size increases the performance of Ridge Regression decreases while L_2Boost improves its performance, especially when accumulated inflation is considered. WLadaENet also presents a good performance to forecast U.S. inflation.

Keywords. Time series. Statistical learning. Machine learning. High-dimensional methods. Inflation forecasting. LASSO. Random forests. Boosting.

JEL Classifications. C53, E37

[1] Graduate Program in Economics, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brazil, e-mail: klausboesch@gmail.com

[2] Department of Statistics and Graduate Program in Economics, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brazil, e-mail: flavioaz@mat.ufrgs.br

2.1 INTRODUCTION

This work is a methodological and empirical study of high-dimensional statistical learning methods in order to perform time series forecasting. Statistical learning refers to a variety of tools used for analyzing data, which can be classified as supervised or unsupervised (JAMES *et al.*, 2013). Most of the methods we study here belong to the former type.

High-dimensional methods are becoming more and more important in the econometrics literature and the inclusion of a large number of financial and economic variables potentially contributes to considerable gains for time series forecasting. However, in the high-dimensional context the traditional methods present three main problems. The first is that estimates often have low bias but large variance, reducing the accuracy of the forecasts. The second is related to model interpretabillity. Considering a large set of predictors, it is preferable to determine a smaller subset of predictors that exhibits the effects that most explain the variability of the response variable (TIBSHIRANI, 1996; KONZEN; ZIEGELMANN, 2016). The third problem occurs when the number of predictors exceeds the sample size in which case by construction, the ordinary least squares (OLS) method cannot be implemented. Another impact of high-dimensionality is spurious correlation (FAN, 2014): it refers to variables that are not correlated in theory, but whose sample correlation is high. This problem may lead to erroneous statistical inference.

Statistical learning methods are able to handle some of the issues. These methods are designed to improve out-of-sample prediction. Gu *et al.* (2019) and Mullainathan and Spiess (2017) point out the three main characteristics of (high-dimensional) statistical learning[3] methods. Statistical learning contains a diverse collection of high-dimensional methods for statistical prediction that combine two elements, namely, regularization and empirical tuning. The high-dimensional nature of these methods enhances their flexibility relative to more traditional econometric prediction techniques (GU *et al.*, 2019). Statistical learning methods typically have a regularizer associated with them and the empirical tuning allows one to chose the level of regularization appropriately (MULLAINATHAN; SPIESS, 2017).

In this work we employ a variety of statistical learning methods (including methods based on shrinkage, regression trees and boosting) to perform time series forecasting. In particular, we propose a method we call WLadaENet (weighted lag adaptive Elastic Net), which combines quadratic regularization and the adaptive weighted LASSO shrinkage similarly to adaENet introduced by Zou and Zhang (2009), but further penalizes coefficients of higher-lagged variables like the WLadaLASSO proposed by Konzen and Ziegelmann (2016).

In order to compare the performance of the statistical learning methods considered, a simulation study is carried out, with three different data-generating process and vary the number of lags used as well as the sample size of simulated data. We also present two empirical applications, where in the first one we employ a set of statistical learning methods to forecast

[3] In fact, Gu *et al.* (2019) and Mullainathan and Spiess (2017) refer to these methods as *machine learning*, however in this work we are using *statistical learning* in a more general context, according to Hastie *et al.* (2009).

Brazilian inflation and the core inflation, while in the second application we employ these methods to perform U.S. inflation forecasting.

2.2 LITERATURE REVIEW

High-dimensional methods are getting more and more present in the econometrics literature and the inclusion of a large collection of economic and financial predictors can improve the time series forecasting performance significantly. For instance, Medeiros and Vasconcelos (2016) point out that some advances in computer science, statistics and econometrics allow us to handle large and complex datasets and show that (on average) high-dimensional methods can produce smaller forecasting errors for macroeconomic variables forecasting when a large set of predictors is considered.

James *et al.* (2013) describe statistical learning as a vast collection of tools for understanding data which can be classified as supervised or unsupervised, where supervised statistical learning consists in building a statistical model for predicting, or estimating, an output based on just a single or a large set of inputs, while in unsupervised statistical learning problems, there are inputs but no supervised output. Gu *et al.* (2019) in turn define machine learning as "(i) a diverse collection of high-dimensional methods for statistical prediction, combined with (ii) so-called 'regularization' methods for model selection and mitigation of overfit, and (iii) efficient algorithms for searching among a vast number of potential model specifications". In the same direction Mullainathan and Spiess (2017) highlight the ability of machine learning to deal with high-dimensional data and present the two key elements which allow machine learning to handle overfitting: regularization and empirical tuning.

Considering the multiple linear regression model, when the number of predictors q approaches the sample size T, the least squares method becomes inefficient or even inconsistent and begins to overfit noise instead of extracting signal (GU *et al.*, 2019) . The task of regularization is to reduce the model's complexity, improving its out-of-sample stability and, consequently, reducing its in-sample performance, whereas empirical tuning is used to determine the optimal level of complexity, reducing the model's fit of noise and preserving its fit of the signal (MULLAINATHAN; SPIESS, 2017; GU *et al.*, 2019) . As the methods employed in this work are referred to in the literature as (supervised) machine learning as well as (supervised) statistical learning methods, we are using high-dimensional statistical learning methods in a broad way.

Medeiros and Mendes (2016) study the asymptotic properties of adaLASSO in sparse, high-dimensional, linear time-series models and find that its properties allow the adaLASSO to be applied to a variety of applications in empirical finance and macroeconomics. The authors also present an application to forecast U.S. inflation using many predictors, where adaLASSO delivers superior forecasts compared to traditional benchmarks such as autoregressive and factor models. Medeiros and Vasconcelos (2016) employ high-dimensional methods (LASSO-family and *bagging*) to forecast macroeconomic variables, and show that high-dimensional methods provide smaller forecast errors than autoregressive and factor models. Medeiros *et al.* (2016) use

LASSO and adaLASSO to forecast Brazilian inflation and observe that LASSO-based methods have the smallest errors for short-horizon forecasts. Konzen and Ziegelmann (2016) test LASSO-type penalty methods for covariate selection and forecasting, and propose the WLadaLASSO (which is based on further penalizing lagged predictors) method that presents good results for U.S. risk premium and U.S. inflation forecasting. Garcia *et al.* (2017) employ a variety of high-dimensional methods to forecast Brazilian inflation, including LASSO-based methods, Complete Subset Regression (CSR) and Random Forest (RF), with LASSO and adaLASSO displaying the best performance for shorter horizons, while CSR is the best method for longer horizons. Medeiros *et al.* (2019) added a variety of high-dimensional methods to forecast U.S. inflation, such as bagging, Ridge Regression, Elastic Net (and its adaptive version) and Jackknife Model Averaging (JMA), finding that RF robustly outperforms the other methods. Gu *et al.* (2019) provide a comparative analysis of methods in the machine learning repertoire, using these methods to forecast stock returns, and identify the best performing methods as trees-based models and neural networks.

The following sections provide some theoretical background of the methods employed in this work.

2.2.1 Benchmark Methods

Two benchmark methods based on univariate models are considered in this work: the Random Walk (RW) and the Autoregressive (AR).

2.2.1.1 Random Walk (RW)

Suppose that $\{y_t\}$, $t = 0,1,2...$, follows a Random Walk process, which is a cumulative sum of iid random variables. Defining $y_0 = 0$, a RW with zero mean is given by $y_t = \varepsilon_1 + ... + \varepsilon_t$, $t = 1,2...$, where $\{\varepsilon_t\}$ is iid noise, such that $y_{t+1} - y_t = \varepsilon_{t+1}$. As we are interested in forecast y_{t+h}, we have that $y_{t+h} = y_t + \underbrace{\sum_{j=1}^{h} \varepsilon_{t+j}}_{u_{t+h}}$ and the best h-step ahead forecast for y_{t+h}, for all t, is given by $\hat{y}_{t+h} = y_t$, $h = 1,..,H$.

2.2.1.2 Autoregressive (AR)

The Autoregressive model of order p is given by

$$y_{t+1} = \phi_0 + \phi_1 y_t + ... + \phi_p y_{t-p+1} + u_{t+1}, \tag{2.1}$$

where y_t is stationary under certain conditions and the sequence $\{u_t\}$ is in general supposed as a white noise (with mean 0 and variance σ^2). We are interested in forecast y_{t+h}, $h = 1,...,H$ and we only have information up to time t. An usual approach to estimate $\hat{y}_{t+h}$ is to recursively estimate one-step-ahead $\hat{y}_{t+1}$, $\hat{y}_{t+2}$, ..., $\hat{y}_{t+h-1}$, $\hat{y}_{t+h}$, based on Equation (2.1), pluggin in each step the previous forecasts. Alternatively, in this work we employ a direct forecast approach, similarly to Medeiros *et al.* (2019), where the forecast of y_{t+h} is given by

$$\hat{y}_{t+h} = \hat{\phi}_{0,h} + \hat{\phi}_{1,h} y_t + ... + \hat{\phi}_{p,h} y_{t-p+1}, \quad t = 1, ..., T. \tag{2.2}$$

2.2.2 Shrinkage Methods

Considering the linear model

$$y_{t+h} = \beta_0 + \beta_1 x_{t,1} + ... + \beta_q x_{t,q} + u_{t+h}, \quad t = 1, ..., T. \tag{2.3}$$

The forecasting of y_{t+h} is denoted as $\hat{y}_{t+h}$ and given by

$$\hat{y}_{t+h} = \hat{\beta}_0 + \hat{\beta}_1 x_{t,1} + ... + \hat{\beta}_q x_{t,q}, \quad t = 1, ..., T. \tag{2.4}$$

This section presents a class of methods that shrink the regression coefficients by imposing a penalty on their size, i.e, the estimated coefficients in a linear model minimize a penalized residual sum of squares

$$\hat{\beta}(\Theta) = \underset{b_0,...,b_q}{\arg\min} \left[\underbrace{\sum_{t=1}^{T-h} \left(y_{t+h} - b_0 - \sum_{j=1}^{q} b_j x_{t,j} \right)^2}_{RSS(b_0,...,b_q)} + p(b_1, ..., b_q; \boldsymbol{\theta}) \right], \tag{2.5}$$

where $\hat{\beta}(\boldsymbol{\theta}) = (\hat{\beta}_0, \hat{\beta}_1(\boldsymbol{\theta}), ..., \hat{\beta}_q(\boldsymbol{\theta}))$, $p(.)$ is a penalty function and $\boldsymbol{\theta}$ is a vector of tuning parameters. The following shrinkage methods consider different choices for the penalty function which are summarized in Table 1.

Table 1 – Penalty functions

Method	Penalty function
Ridge	$p(.) = \lambda \sum_{j=1}^{q} b_j^2$
LASSO	$p(.) = \lambda \sum_{j=1}^{q} \|b_j\|$
ENet	$p(.) = \lambda [\rho \sum_{j=1}^{q} \|b_j\| + (1-\rho) \sum_{j=1}^{q} b_j^2]$
adaLASSO	$p(.) = \lambda \sum_{j=1}^{q} \|\hat{\beta}_j^*\|^{-\tau} \|b_j\|$
adaENet	$p(.) = \lambda [\rho \sum_{j=1}^{q} \|\hat{\beta}_j^*\|^{-\tau} \|b_j\| + (1-\rho) \sum_{j=1}^{q} b_j^2]$
WLadaLASSO	$p(.) = \lambda \sum_{j=1}^{q} (\|\hat{\beta}_j^*\| e^{-\alpha l_j})^{-\tau} \|b_j\|$
WLadaENet	$p(.) = \lambda [\rho \sum_{j=1}^{q} (\|\hat{\beta}_j^*\| e^{-\alpha l_j})^{-\tau} \|b_j\| + (1-\rho) \sum_{j=1}^{q} b_j^2]$

Source: Own elaboration (2020).

2.2.2.1 Ridge Regression (RR)

Hoerl and Kennard (1970) introduce the Ridge Regression and demonstrate that this method obtains estimations with a little bias increase and substantial variance reduction in return, and therefore increasing the prediction accuracy. The ridge coefficients minimize a penalized residual sum of squares, where the penalization term takes the form $p(.) = \lambda \sum_{j=1}^{q} b_j^2$ and the minimization problem becomes

$$\hat{\beta}^{RR} = \underset{b_0,\ldots,b_q}{\arg\min} \left[RSS(b_0,...,b_q) + \lambda \sum_{j=1}^{q} b_j^2 \right], \quad (2.6)$$

where the complexity (or tuning) parameter $\lambda \geq 0$ controls the amount of shrinkage (the larger the value of λ is, the greater the amount of shrinkage and the more the coefficients are shrunk toward zero (HASTIE *et al.*, 2009)).

Another way to write the Ridge problem is

$$\hat{\beta}^{RR} = \underset{b_0,\ldots,b_q}{\arg\min} \left[RSS(b_0,...,b_q) \right] \text{ subject to } \sum_{j=1}^{q} b_j^2 \leq s, \quad (2.7)$$

where the constraint imposed on the parameters size is explicit. There is a one-to-one correspondence between the parameters λ in (2.6) and s in (2.7). Let $s_0 = \sum_{j=1}^{q} (\hat{\beta}_j^{OLS})^2$. If $s \geq s_0$ (or $\lambda = 0$), Ridge estimates will be the same as OLS estimates and values of $s < s_0$ (or $\lambda > 0$) cause a shrinkage of the coefficients toward zero. The Ridge problem has an analytic solution given by

$$\hat{\beta}^{RR} = (X^T X + \lambda I)^{-1} X^T \boldsymbol{y}_h, \quad (2.8)$$

in matrix notation, where X is the design matrix, whose entry t, j corresponds to the t_{th} observation of the j_{th} predictor, $\boldsymbol{y}_h$ is a vector containing the observations of the response variable and I is the qxq identity matrix. The solution (2.8) adds a positive constant to the diagonal of $X^T X$ before inversion, making the problem non-singular, even if $X^T X$ is not full rank (HASTIE *et al.*, 2009). Thus Ridge Regression can be implemented even when the number of predictors is greater than the sample size.

2.2.2.2 Least Absolute Shrinkage and Selection Operator (LASSO)

As Ridge Regression obtains non-zero estimates for all coefficients it is not a variable selection method (KONZEN; ZIEGELMANN, 2016) and hence does not provide an easily interpretable model. Subset selection methods (such as Best-Subset Selection, Forward- and Backward-Stepwise Selection, and Forward-Stagewise Regression), in turn, provide interpretable models but often exhibit high variance and, consequently, do not reduce the prediction error of the full model (HASTIE *et al.*, 2009). Tibshirani (1996) proposes a technique called LASSO, for Least Absolute Shrinkage and Selection Operator. This method shrinks some coefficients while setting others to zero, retaining the good features of both subset selection and Ridge Regression techniques. As Ridge Regression, the LASSO coefficients minimize a sum of squared residuals subject to a penalty of the coefficients size, but adopts a L_1 norm instead of a L_2 penalty norm used in Ridge Regression. The LASSO penalty takes the form $p(.) = \lambda \sum_{j=1}^{q} |b_j|$ and the LASSO estimator is given by

$$\hat{\beta}^{LASSO} = \underset{b_0,\ldots,b_q}{\arg\min} \left[RSS(b_0,...,b_q) + \lambda \sum_{j=1}^{q} |b_j| \right]. \quad (2.9)$$

As in Ridge Regression λ controls the amount of penalization that is applied to the coefficients. The tuning parameter λ is usually chosen by data-driven techniques such as cross-validation, or through the use of information criteria (GARCIA *et al.*, 2017). According to Konzen and Ziegelmann (2016), in a cross-section framework the value of λ is traditionally chosen via cross-validation, whereas in a time series set-up choosing the parameter λ using the Bayesian Information Criterion (BIC) is most suitable.

The L_1 LASSO penalty constraint makes the solution nonlinear in y_{t+h} , and unlike Ridge Regression, the LASSO minimization problem in Equation (2.9) does not have a closed form expression (HASTIE *et al.*, 2009). Computing the LASSO solution is a quadratic programming problem.

2.2.2.3 Adaptive LASSO (adaLASSO)

As LASSO does not have the oracle properties and in some situations the variable selection can be inconsistent. Zou (2006) proposes a new methodology called Adaptive LASSO (adaLASSO), in order to avoid LASSO deficiencies, resulting in a method that enjoys the oracle properties. The oracle properties refer to: 1. Consistency in variable selection; 2. Asymptotic normality. The adaLASSO is a weighed version of LASSO that assigns different weights to different coefficients on the LASSO penalty term. The penalty is given by $p(.) = \lambda \sum_{j=1}^{q} \omega_j |b_j|$ and the adaLASSO estimator is given by

$$\hat{\beta}^{adaLASSO} = \underset{b_0,\ldots,b_q}{\arg\min} \left[RSS(b_0, \ldots, b_q) + \lambda \sum_{j=1}^{q} \omega_j |b_j| \right], \tag{2.10}$$

where $\omega_j = |\hat{\beta}_j^*|^{-\tau}$ and $\hat{\beta}_j^*$ is the coefficient from a first-step estimation and where τ is another tuning parameter.

Zou (2006) shows that if the weights are data-driven and cleverly chosen, then the weighted LASSO can have the oracle properties. So he suggested using $\hat{\beta}^{OLS}$ as $\hat{\beta}^*$ unless collinearity is a concern, in which case he suggests $\hat{\beta}^{RR}$ for the best Ridge Regression fit, because it is more stable than $\hat{\beta}^{OLS}$. The value of τ must be positive and the most common value used is $\tau = 1$. If $\tau = 0$ we have the traditional LASSO.

2.2.2.4 Weighted Lag adaptive LASSO (WLadaLASSO)

In time series context when adaLASSO is performed, each lagged variable can enter as a predictor candidate and its individual penalty depends only on $\hat{\beta}^*$. Assuming that the coefficients of variables with higher lags approach zero Konzen and Ziegelmann (2016) propose a modified version of adaLASSO with weighted lags, called Weighted Lag Adaptative LASSO (WLadaLASSO). The WLadaLASSO penalty is given by $p(.) = \lambda \sum_{j=1}^{q} \omega_j^* |b_j|$ and the estimator is given by

$$\hat{\beta}^{WLadaLASSO} = \underset{b_0,\ldots,b_q}{\arg\min} \left[RSS(b_0, \ldots, b_q) + \lambda \sum_{j=1}^{q} \omega_j^* |b_j| \right], \tag{2.11}$$

where $\omega_j^* = \left(|\hat{\beta}_j^*|e^{-\alpha l_j}\right)^{-\tau}$, $\tau > 0$, $\alpha \geq 0$ and l_j represents the lag order of variable x_j. They suggest pluggin in $\hat{\beta}^{RR}$ as $\hat{\beta}^*$ and chose the value of the tuning parameter α using the BIC. Rearranging ω_j^* results in $\omega_j^* = |\hat{\beta}_j^*|^{-\tau}e^{\tau\alpha l_j}$ that is equal to the adaLASSO weight ω_j multiplied by a factor $e^{\tau\alpha l_j}$. Note that if $\alpha = 0$, $\omega_j^* = \omega_j$, then we have the traditional adaLASSO. If either $\alpha \to \infty$ or $l_j \to \infty$, then $\omega_j^* \to \infty$.

2.2.2.5 Elastic Net (ENet)

According to Zou and Hastie (2005), LASSO has some limitations: (i) If the number of predictors q is bigger than the sample size T, then LASSO selects at most T variables before it saturates; (ii) If there is a group of variables among which the pairwise correlations are very high, LASSO tends to select only one variable from that group; (iii) For $T > q$, if there are high correlations between predictors, the prediction performance of LASSO is dominated by Ridge Regression. Therefore LASSO is an inappropriate variable selection method in some situations. In order to avoid some of the deficiencies of LASSO, Zou and Hastie (2005) propose the Elastic Net as a new regularization technique. Elastic Net penalty is a convex combination between L_1 norm LASSO penalty and L_2 norm Ridge Regression penalty and is given by $p(.) = \lambda[\rho\sum_{j=1}^{q}|b_j| + (1-\rho)\sum_{j=1}^{q}b_j^2]$, $0 < \rho < 1$. The Elastic Net estimator takes the form[4]

$$\hat{\beta}^{ENet} = \underset{b_0,\ldots,b_q}{\arg\min}\left\{RSS(b_0,\ldots,b_q) + \lambda\left[\rho\sum_{j=1}^{q}|b_j| + (1-\rho)\sum_{j=1}^{q}b_j^2\right]\right\}. \quad (2.12)$$

LASSO, Ridge Regression and OLS can be seen as special cases of Elastic Net. If $\rho = 1$ the problem reduces to LASSO while $\rho = 0$ corresponds to Ridge Regression, and finally, if $\lambda = 0$ there is no penalty and the problem reduces to OLS. According to Zou and Hastie (2005), Elastic Net simultaneously performs automatic variable selection and continuous shrinkage similar to LASSO, and additionally can select groups of correlated variables. Moreover, Elastic Net often outperforms LASSO in terms of prediction accuracy. The authors propose an efficient algorithm called LARS-EN for computing the entire Elastic Net regularization paths with computational effort comparable to a single OLS fit.

2.2.2.6 Adaptive Elastic Net (adaENet)

Zou and Zhang (2009) propose the adaptive Elastic Net and establish its oracle properties under weak regularity conditions. The adaptive penalization of Elastic Net combines quadratic regularization and the adaptive weighted lasso shrinkage, and is given by $p(.) = \lambda[\rho\sum_{j=1}^{q}\hat{\omega}_j|b_j| + (1-\rho)\sum_{j=1}^{q}b_j^2]$. The adaptive Elastic Net estimates are given by

$$\hat{\beta}^{adaENet} = \underset{b_0,\ldots,b_q}{\arg\min}\left\{RSS(b_0,\ldots,b_q) + \lambda\left[\rho\sum_{j=1}^{q}\hat{\omega}_j|b_j| + (1-\rho)\sum_{j=1}^{q}b_j^2\right]\right\}, \quad (2.13)$$

[4] Zou and Hastie (2005) call Equation (2.12) the Naive Elastic Net, and prefer a rescaled version which they call Elastic Net, we follow Friedman *et al.* (2010) and do not use this distinction.

where $\hat{\omega}_j = (|\hat{\beta}_j^*| + T^{-1})^{-\tau}$, $\hat{\beta}_j^* = \hat{\beta}_j^{ENet}$ and τ is a positive constant. According to Zou and Zhang (2009) the L_2 penalty further regularizes the adaptive LASSO fit whenever the collinearity may cause serious trouble and the L_1 regularization parameters is responsible for the sparsity of the estimates. They show that this method deals with the collinearity problem better than the other oracle-like methods, enjoying improved finite sample performance.

Remark: As our implementation of the adaptive Elastic Net is based on the R package *glmnet* where the parameter *penalty.factor* (which represents the vector of weights $\omega = (\omega_1, ..., \omega_q)^T$) multiplies λ, then we have, in fact, the penalty $p(.) = \lambda[\rho \sum_{j=1}^q \hat{\omega}_j |b_j| + (1-\rho) \sum_{j=1}^q \hat{\omega}_j b_j^2]$.

2.2.2.7 Weighted Lag adaptive Elastic Net (WLadaENet)

We propose a method that we call Weighted Lag Adaptive Elastic Net (WLadaENet), a combination between adaENet and WLadaLASSO, where the idea is similar to the adaptive Elastic Net, but penalizing further the coefficients of higher-lagged covariates. The penalization of WLadaENet is given by $p(.) = \lambda[\rho \sum_{j=1}^q \hat{\omega}_j^* |b_j| + (1-\rho) \sum_{j=1}^q b_j^2]$ and the minimization problem is similar to adaENet and the estimator is given by

$$\hat{\beta}^{WLadaENet} = \underset{b_0,...,b_q}{\arg\min} \left\{ RSS(b_0, ..., b_q) + \lambda \left[\rho \sum_{j=1}^{q} \hat{\omega}_j^* |b_j| + (1-\rho) \sum_{j=1}^{q} b_j^2 \right] \right\}, \qquad (2.14)$$

where $\omega_j^* = \left[\left(|\hat{\beta}_j^*| \right) e^{-\alpha l_j} \right]^{-\tau}$, if either OLS or RR are employed in the first stage. In turn, if LASSO or ENet are employed, $\hat{\omega}_j^* = \left[\left(|\hat{\beta}_j^*| + T^{-1} \right) e^{-\alpha l_j} \right]^{-\tau}$. Similarly to WLadaLASSO, $\tau > 0$, $\alpha \geq 0$, l_j represents the lag order and $\hat{\beta}_j^*$ are the coefficient estimates of the first stage.

Remark: Here the situation is the same as the described to adaENet implementation. Thus we have in fact the penalty $p(.) = \lambda[\rho \sum_{j=1}^q \hat{\omega}_j^* |b_j| + (1-\rho) \sum_{j=1}^q \hat{\omega}_j^* b_j^2]$.

2.2.3 Ensemble Methods

According to Hastie *et al.* (2009), ensemble learning consists in constructing a prediction method $\hat{F}_h$ by combining the strengths of simple base estimators $\hat{f}_h$, such that the prediction rule is given by

$$\hat{y}_{t+h} = \hat{F}_h(\boldsymbol{x}_t) = \lambda \sum_{b=1}^{B} \hat{f}_{h,b}(\boldsymbol{x}_t), \qquad (2.15)$$

where $\lambda \in (0,1)$ can be a learning rate or a weight.

2.2.3.1 Complete Subset Regression (CSR)

Subset Selection methods retain only a subset of the variables, and eliminate others from the model, where least squares regression is employed to estimate the coefficients of the retained covariates. Elliott *et al.* (2013) introduce the Complete Subset Regression which for a given set of potential predictor variables combines forecasts from all possible linear regressions with a fixed number of predictors. For a set of K predictor candidates there are $n_{k,K} = K!/[(K-k)!k!]$ combinations of $k < K$ variables. Elliott *et al.* (2015) consider large-dimensional sets of potential predictors where the CSR is unfeasible and indicate a pre-testing procedure as a possible solution.

Consider the liner model

$$y_{t+h} = \gamma^T \boldsymbol{z}_t + \delta^T \boldsymbol{w}_t + u_{t+h}, \quad t = 1, ..., T$$

where $\boldsymbol{z}_t$ is a $Px1$ vector of predictors which is always included in the forecasting and $\boldsymbol{w}_t$ is a $Kx1$ vector of variables that we are not sure whether are useful predictors. We follow Garcia *et al.* (2017) and Medeiros *et al.* (2019) and use a pre-testing procedure where for each variable in $\boldsymbol{w}_t$ we fit a linear regression of y_{t+h} by OLS and use the absolute values of the t-statistic to select the $\tilde{K} < K$ most relevant variables. The CSR forecast is given by

$$\hat{y}_{t+h} = B^{-1} \sum_{b=1}^{B} \hat{\beta}_b^T \boldsymbol{x}_t = \underbrace{\left[B^{-1} \left(\sum_{b=1}^{B} \hat{\beta}_b^T \right) \right]}_{(\hat{\beta}^{CSR})^T} \boldsymbol{x}_t, \tag{2.16}$$

where $B = n_{k,\tilde{K}} = \tilde{K}!/[(\tilde{K}-k)!k!]$, $\boldsymbol{x}_t = (\boldsymbol{z}_t, \boldsymbol{w}_t)$ and $\hat{\beta}_b = (\hat{\gamma}_b^T, \hat{\delta}_b^T)^T$, $b = 1, ..., B$.

2.2.3.2 Componentwise L_2 Boosting (L_2Boost)

Originally designed for classification, boosting was afterwards extended to encompass regression problems. Boosting is a procedure that combines the outputs of many *base* or *weak learners* iteratively in order to achieve high accuracy (BÜHLMANN; YU, 2003; HASTIE *et al.*, 2009). According to Friedman (2002) gradient boosting constructs additive regression models by sequentially fitting a simple parameterized function (base learner) to current "pseudo"-residuals by least-squares at each iteration, where the "pseudo"-residuals are the gradient of the loss functional being minimized, with respect to the model values at each training data point, evaluated at current step. The boosting technology builds an ensemble model by conducting a regularized and supervised search in a high-dimensional space of weak learners (HASTIE *et al.*, 2009).

Bühlmann and Yu (2003) present a computationally simple variant of boosting algorithms, L_2Boost, which is constructed from a functional gradient descent algorithm employing the L_2-loss function. For L_2Boost method the loss function is given by $L(y_{t+h}, \hat{F}(\boldsymbol{x}_t)) = [y_{t+h} - \hat{F}(\boldsymbol{x}_t)]^2/2$ such that the gradient of $L(.)$ is $\hat{u}_{t+h} = y_{t+h} - \hat{F}(\boldsymbol{x}_t), \quad t = 1, ..., T-h$.

Based on Bühlmann and Yu (2003), and Bai and Ng (2009) the Componentwise L_2Boosting algorithm follows the steps bellow:

1. Let $\hat{F}_0(\boldsymbol{x}_t) = \hat{\beta}_0 = (T-h)^{-1} \sum_{t=1}^{T-h} y_{t+h}$ and $\hat{\beta}_{1,0} = \hat{\beta}_{2,0} = ... = \hat{\beta}_{q,0} = 0$.

2. For $b = 1, ..., B$:

(a) Compute the pseudo-residuals $\hat{u}_{t+h,b} = y_{t+h} - \hat{F}_{b-1}(\boldsymbol{x}_t)$, $t = 1, ..., T-h$.

(b) For $j = 1, ..., q$, regress $\hat{u}_{t+h,b}$ on $x_{t,j}$ to obtain $\tilde{\beta}_{j_b}$.

(c) Compute $\hat{e}_{t+h,j,b} = \hat{u}_{t+h,b} - x_{t,j}\tilde{\beta}_{j_b}$, $t = 1, ..., T-h$, $j = 1, .., q$.

(d) Select $j_b^* = \arg\min_j [\sum_{t=1}^{T-h} (\hat{e}_{t+h,j,b})^2]$.
(e) Define $\hat{f}_b = \boldsymbol{x}_{j_b^*}\tilde{\beta}_{j_b^*}$, where $\boldsymbol{x}_{j_b^*} = (x_{1,j_b^*}, ..., x_{T-h,j_b^*})^T$.
(f) Update $\hat{F}_b(\boldsymbol{x}_t) = \hat{F}_{b-1}(\boldsymbol{x}_t) + \lambda \hat{f}_b$ and $\hat{\beta}_{j,b} = \hat{\beta}_{j,b-1} + \lambda \tilde{\beta}_{j_b^*}$, where $0 < \lambda \leq 1$.
(g) Update $\hat{\beta}_{i,b} = \hat{\beta}_{i,b-1}$, $i \neq j$.
(h) Compute the Information Criterion (IC).
3. Output $\hat{\beta}^{\mathrm{L_2Boost}} = (\hat{\beta}_0, \hat{\beta}_{1,B^*}, ..., \hat{\beta}_{q,B^*})$, where $B^* = \arg\min_{1 \leq b \leq B} \mathrm{IC}(b)$.
The $\mathrm{L_2}$Boost forecast is given by

$$\hat{y}_{t+h} = \hat{\beta}_0 + \hat{\beta}_{1,B^*} x_{t,1} + ... + \hat{\beta}_{q,B^*} x_{t,q}. \tag{2.17}$$

In order to avoid overfit Bühlmann (2006) proposes a stopping rule using the corrected AIC_c criterion:

$$\mathrm{AIC}_c(b) = \log(\hat{\sigma}_b^2) + \frac{1 + \mathrm{df}_b/(T-h)}{1 - (\mathrm{df}_b + 2)/(T-h)}, \tag{2.18}$$

where $\mathrm{df}_b = \mathrm{trace}(\mathcal{B}_b)$, $\hat{\sigma}_b^2 = (T-h)^{-1} \sum_{t=1}^{T-h} \{\underbrace{[(I_{T-h} - \mathcal{B}_b)\boldsymbol{y}_h]_t}_{y_{t+h} - \hat{F}_b(\boldsymbol{x}_t)}\}^2$, $\mathcal{B}_b = \mathcal{B}_{b-1} + \lambda P_{j_b^*}(I_{T-h} - \mathcal{B}_{b-1})$, $P_{j_n^*} = x_{j_b^*}(x_{j_b^*}^T x_{j_b^*})^{-1} x_{j_b^*}^T$ and I_{T-h} is the $(T-h)$x$(T-h)$ identity matrix.

For time series, Bai and Ng (2009) use the BIC as IC of the stopping rule.

$$\mathrm{BIC}(b) = \log(\hat{\sigma}_b^2) + \frac{\mathrm{df}_b \log(T-h)}{(T-h)}. \tag{2.19}$$

2.2.3.3 Random Forest (RF)

Breiman (2001) proposes the Random Forest to improve the variance reduction of *bagging* predictors (BREIMAN, 1996) by reducing the correlation between the trees. Regression trees partition the space of predictors into disjoint regions $\{R_k\}_{k=1}^K$ and these regions are represented by the terminal nodes (or leaves) of the corresponding tree. According to Hastie *et al.* (2009) a regression tree with K regions (terminal nodes or leaves) can be formally defined by the equation

$$T(\boldsymbol{x}_t; \boldsymbol{\theta}) = \sum_{k=1}^{K} \beta_k \mathbb{I}_{R_k}(\boldsymbol{x}_t), \tag{2.20}$$

with parameters $\boldsymbol{\theta} = \{R_k, \beta_k\}_{k=1}^K$ and where $\mathbb{I}_{R_k}(.)$ is a product of indicator functions such that

$$\mathbb{I}_{R_k}(\boldsymbol{x}_t) = \begin{cases} 1 & \text{if} \quad \boldsymbol{x}_t \in R_k, \\ 0 & \text{otherwise.} \end{cases} \tag{2.21}$$

Adopting as minimization criterion the sum of squares, the best $\hat{\beta}_k$ is given by the average of the y_{t+h} that lay in region $\hat{R}_k$

$$\hat{\beta}_k = \frac{\sum_{t=1}^{T-h} \mathbb{I}_{\hat{R}_k}(\boldsymbol{x}_t) y_{t+h}}{\sum_{t=1}^{T-h} \mathbb{I}_{\hat{R}_k}(\boldsymbol{x}_t)}. \tag{2.22}$$

The partition process starts at the root node with the whole sample, where the pair split-point/split-variable, that most reduces the Residual Sum of Squares (RSS) is selected

$$(s^*, j^*) = \underset{s,j=1,\ldots,q}{\arg\max} \left[RSS_0(\boldsymbol{y}_h, \hat{\beta}_0) - RSS_1(\boldsymbol{y}_h, \hat{\beta}_1(s,j)) - RSS_2(\boldsymbol{y}_h, \hat{\beta}_2(s,j)) \right], \tag{2.23}$$

where $s \in (\min(\{x_{t,j}\}_{t=1}^{T-h}), \max(\{x_{t,j}\}_{t=1}^{T-h}))$.

For the Random Forest algorithm consider $Z = \{(y_{t+h}, \quad \boldsymbol{x}_t)\}_{t=1}^{T-h}$ as a learning set. Take B bootstrap sub-samples from Z, with replacement and size $T-h$. A regression tree is estimated for each bootstrapped sub-sample Z_b^*, $b = 1,..,B$. For each tree the process starts in the root node with the whole sample, and then the following steps are recursively repeated, for each terminal node of the tree until either the minimum node size n_{min} is reached or the reduction of RSS is bellow a certain value:

1. Select d variables from the set of q variables at random, where $d < q$.

2. Choose the pair split-point/variable that most reduces the RSS among the d variables.

3. Split the current node into two new nodes.

The RF process results in an ensemble of trees $\{T_b\}_{b=1}^{B}$. The final prediction of RF is given by

$$\hat{y}_{t+h} = \hat{F}_B^{RF}(\boldsymbol{x}_t) = B^{-1} \sum_{b=1}^{B} \underbrace{\left[\sum_{k=1}^{K_b} \hat{\beta}_{k,b} \mathbb{I}_{\hat{R}_{k,b}}(\boldsymbol{x}_t) \right]}_{T_b(\boldsymbol{x}_t, \hat{\boldsymbol{\theta}}_b)} \tag{2.24}$$

2.2.3.4 Boosting Regression Trees (B.Trees)

Boosting is a general approach that can be applied to many statistical learning methods as well as to improve the prediction accuracy of regression trees where each tree is iteratively grown (or constructed) using information from previously grown trees (JAMES *et al.*, 2013).

For boosting regression trees, Friedman (2001) considers the case where each base learner is an K-terminal node regression tree. $\{R_k\}_{k=1}^{K}$ are disjoint regions that collectively cover the space of all joint values of the predictor variables.

In boosting algorithm, for regression trees the update at each iteration has the form

$$\hat{F}_b(\boldsymbol{x}_t) = \hat{F}_{b-1}(\boldsymbol{x}_t) + \lambda \sum_{k=1}^{K} \hat{\beta}_{k,b} \mathbb{I}_{\hat{R}_{k,b}}(\boldsymbol{x}_t), \tag{2.25}$$

where $\{\hat{R}_{k,b}\}_{k=1}^{K}$ are the regions defined by the terminal nodes of the tree at the b_{th} iteration. The trees are constructed to predict the pseudo-responses $\{\hat{u}_{t+h,b}\}_{t=1}^{T-h}$ by least squares. The $\{\hat{\beta}_{k,b}\}_{k=1}^{K}$ are the corresponding least squares coefficients

$$\hat{\beta}_{k,b} = \frac{\sum_{t=1}^{T-h} \mathbb{I}_{\hat{R}_{k,b}}(\boldsymbol{x}_t) \hat{u}_{t+h,b}}{\sum_{t=1}^{T-h} \mathbb{I}_{\hat{R}_{k,b}}(\boldsymbol{x}_t)}. \tag{2.26}$$

Then the algorithm of gradient boosting for regression trees, based on Friedman (2001), Hastie *et al.* (2009) and James *et al.* (2013), is given by the following steps:

1. Initialize $\hat{F}_0(\boldsymbol{x}_t) = \hat{\beta}_0 = \arg\min_c \sum_{t=1}^{T-h} L(\boldsymbol{y}_{t+h}, c)$.

2. For $b = 1, ..., B$:

(a) For $t = 1, ..., T-h$, compute

$$\hat{u}_{t+h,b} = -\left[\frac{\partial L(y_{t+h}, F(\boldsymbol{x}_t))}{\partial F(\boldsymbol{x}_t)}\right]_{F=\hat{F}_{b-1}}.$$

(b) Fit a regression tree to the targets $\hat{u}_{t+h,b}$ with K_b terminal nodes.

(c) Update $\hat{F}_b(\boldsymbol{x}_t) = \hat{F}_{b-1}(\boldsymbol{x}_t) + \lambda \sum_{k=1}^{K_b} \hat{\beta}_{k,b} \mathbb{I}_{R_{k,b}}(\boldsymbol{x}_t)$.

3. Output the boosted model $\hat{F}_B^{B.Trees}(\boldsymbol{x}_t) = \hat{\beta}_0 + \lambda \sum_{b=1}^{B} \left[\sum_{k=1}^{K_b} \hat{\beta}_{k,b} \mathbb{I}_{R_{k,b}}(\boldsymbol{x}_t)\right]$.

Boosting has three tuning parameters (JAMES *et al.*, 2013): (i) The number of trees B, (ii) the shrinkage parameter λ and (iii) the number of splits d_b in each tree. If B is too large, boosting tends to overfit. The parameter λ controls the boosting learning rate, where smaller values of λ usually require larger values of B, in order to achieve good performance. The number of splits $d_b = K_b - 1$ is the interaction depth and controls the complexity of the boosted ensemble. d_b splits involve at most d_b variables and often $d_b = 1$ ($K_b = 2$) works well (JAMES *et al.*, 2013). In this case, the boosted stump (a tree consisting of a single split) ensemble fits an additive model.

The forecast of y_{t+h} is given by

$$\hat{y}_{t+h} = \hat{F}_B^{B.Trees}(\boldsymbol{x}_t) = \hat{\beta}_0 + \lambda \sum_{b=1}^{B} \underbrace{\left[\sum_{k=1}^{K} \hat{\beta}_{k,b} \mathbb{I}_{R_{k,b}}(\boldsymbol{x}_t)\right]}_{T_b(\boldsymbol{x}_t, \hat{\boldsymbol{\theta}}_b)}, \quad (2.27)$$

where, for all b, $K_b = K = d + 1$ and $\lambda \in (0, 1]$.

2.2.4 Factors Methods

The following methods avoid high-dimensional problems by using the common factors $\hat{G}_t$ estimated from the dataset $\{\boldsymbol{w}_t\}_{t=1}^{T-h}$ by the method of principal components, where $\boldsymbol{w}_t = (w_{t,1}, ..., w_{t,q})^T$. Consider the following model

$$w_{t,j} = \lambda_j^T G_t + e_{t,j} \quad (2.28)$$

where G_t is the vector of common factors, λ_j is a vector of loadings associated with G_t, and $e_{t,j}$ is the idiosyncratic component of $w_{t,j}$, where $t = 1, ..., T-h$ and $j = 1, ..., q$. The product $\lambda_j^T G_t$ is the common component of $w_{t,j}$ (BAI; NG, 2002).

2.2.4.1 Factors

The factor approach to an h period-ahead forecast is to estimate the forecasting equation using data for $t = 1, ..., T-h$:

$$y_{t+h} = \gamma^T z_t + \delta^T \tilde{g}_t + u_{t+h}, \quad (2.29)$$

where z_t is a vector of predetermined variables, as a constant or lags of y_{t+h}, $\tilde{g}_t$ includes lags of $\hat{g}_t$, $\hat{g}_t \subseteq \hat{G}_t$, and $\hat{G}_t$ are the principal components estimates of the vector G_t in the factors model. The key task in factors based methods is the correct specification of the number of factors. Bai and Ng (2002) set up the determination of factors as a model selection problem employing information criteria to select the number of factors and lags. In our implementation bellow, we follow Medeiros *et al.* (2019) and use the first four principal components of w_t and select the number of lags using the BIC. The forecast is given by

$$\hat{y}_{t+h} = \hat{\beta}_{BIC}^T \boldsymbol{x}_t, \tag{2.30}$$

where $\boldsymbol{x}_t = (z_t, \tilde{g}_t)$, z_t include lags of y_{t+h}, $\tilde{g}_t$ include lags of $\hat{g}_t$ and $\hat{\beta}_{BIC} = (\hat{\gamma}_{BIC}^T, \hat{\delta}_{BIC}^T)^T$ is selected using the BIC from a set coefficients $\{\hat{\beta}^{OLS}(l)\colon l = 1,..,L\}$ estimated by OLS for each lag.

2.2.4.2 Boosting Factors (B.Factors)

To select predictors from a large set of candidates, where they have no natural ordering, Bai and Ng (2009) propose the use of *boosting* to select predictors in factor-augmented autoregressions. The *boosting* algorithm employed is the same presented for the L_2Boost method, where the BIC value is used as a stopping rule to prevent overffiting. The forecast is given by

$$\hat{y}_{t+h} = (\hat{\beta}^{L_2Boost})^T \boldsymbol{x}_t, \tag{2.31}$$

where $\boldsymbol{x}_t = (z_t, \tilde{g}_t)$, z_t include lags of y_{t+h}, $\tilde{g}_t$ include lags of $\hat{g}_t$ and $\hat{\beta}^{L_2Boost}$ is estimated through L_2Boost method.

2.3 METHODOLOGY

In order to evaluate the forecasting performance of the high-dimensional statistical learning methods presented in the preceding section in a time series context, we carry out several numerical exercises, including Monte Carlo simulations and empirical data analyses. The simulation exercise consists of generating time series from classes of known models and then evaluating/comparing the forecasting performance of the different methods. In the empirical applications we study a large set of economic and financial variables from Brazil and the U.S. to forecast inflation h months ahead, where $h = 1,..,12$, and the accumulated inflation for 3, 6 and 12 months, comparing the performance of the high-dimensional statistical learning methods discussed above.

Following Garcia *et al.* (2017) and Medeiros *et al.* (2019) in this work we employ a direct forecast approach such that the h periods ahead response variable, y_{t+h} is modeled as a function of a set of predictors measured at up to time t, considering the following general model:

$$y_{t+h} = F_h(\boldsymbol{x}_t) + u_{t+h}, \quad h = 1,...,H, \quad t = 1,...,T, \tag{2.32}$$

where y_{t+h} is the dependent variable in period $t+h$ and $\boldsymbol{x}_t = (x_{t,1},...,x_{t,q})^T \in \mathbb{X} \subseteq \mathbb{R}^q$ is a set of covariates which contains only variables observed and available at time t. $F_h(.)$ is the mapping between covariates and y_{t+h} and u_{t+h} is the forecasting error. There is a different mapping $F_h(.)$ for each forecasting horizon h and for each method. We use for all methods a rolling-window scheme of fixed length. Using a notation similar to Medeiros *et al.* (2019), the direct forecast equation is given by

$$\hat{y}_{t+h|t} = \hat{F}_{h,t-T_h^w+1:t}(\boldsymbol{x}_t), \tag{2.33}$$

where $\hat{F}_{h,t-T_h^w+1:t}$ is the estimated target function based on data from time $t - T_h^w + 1$ up to t and T_h^w is the window size. The window size varies depending on the forecasting horizon h and the number of lagged variables used in the method.

All methods are evaluated based on a fixed number $T_{PF} = T - T_0$ of point forecasts. For each forecast horizon $h = 1,...,H$ (including the accumulated horizons for the empirical applications) the methods are compared according to two different statistics, namely, the root mean square error (RMSE) and the mean absolute error (MAE), which are defined as follows:

$$\text{RMSE} = \sqrt{T_{PF}^{-1} \sum_{t=T_0+1}^{T} (\hat{u}_{t+h})^2} \quad \text{and} \quad \text{MAE} = T_{PF}^{-1} \sum_{t=T_0+1}^{T} |\hat{u}_{t+h}|, \tag{2.34}$$

where $\hat{u}_{t+h} = y_{t+h} - \hat{y}_{t+h}$.

To test whether the forecasts produced by the set of high-dimensional methods are significantly different from the benchmarks and also which are the best methods, two approaches are considered: Superior Predictive Ability and Model Confidence Set (HANSEN *et al.*, 2011). The first test for Superior Predictive Ability (SPA) is proposed by Hansen (2005) and compares the forecast alternatives with the forecast benchmark. The null hypothesis of this test is that the benchmark method is as least as good as all of the alternatives. The second test is the test for multi-horizon Superior Predictive Ability of Quaedvlieg (2019). It compares the performance of two given methods considering multiples forecasting horizons, where the null hypothesis is that the benchmark is superior than the alternative method. The objective of the MCS is to determine a set of models $\mathcal{M}^*$, containing the best ones from a collection $\mathcal{M}^0$, such that the true best model(s) in $\mathcal{M}^0$ lies in $\mathcal{M}^*$ with a given confidence level, analogously to confidence intervals or regions. When informative data is available the result is a MCS that contains only a few models (or possibly the best one). In turn, uninformative data yield a MCS with many (or possibly all) models.

2.3.1 Methods Implementation and Parameters

All methods are implemented in R. For all shrinkage methods we used the package *glmnet*. The RF method is implemented using the package *randomForestSRC* and B.Trees using the package *gbm*. For the shrinkage methods the parameter λ and the parameter $\alpha \in \{0, 0.5, 1, ..., 10\}$ of WL methods are selected according to the BIC. The parameter ρ of ENet and its adaptive versions is set to 1/2 (1/3 in *glmnet* function). For the adaptive methods we employ Ridge Regression in the first stage to compute the weights ω_j. For the CSR method we fix $\tilde{K} = 20$ and $k = 4$, following Medeiros *et al.* (2019) and using the first four lags of y_{t+h} as fixed controls. The method L_2Boost employs the minimum BIC as the stopping rule, where we use $\lambda = 0.2$ and the maximum number of iterations $B = 10q$. For RF and B.Trees we use the number of trees $B = 500$ and the minimum number of observations by leaf $n_{min} = 15$. The number of splits of B.Trees is fixed at $d = 2$ while for the RF this parameter is not fixed. RF is implemented employing non-overlapping blocks bootstrap where the block size is 4. In the factors methods we use the first four Principal Components.

The parameters of the AR model are estimated by Ordinary Least Squares (OLS) and the order p is determined by the BIC. A set of AR models $\{AR(p) : p \in \{1, ..., L\} \subset \mathbb{N}\}$ is estimated and the selected model is the one which has minimum BIC. For the empirical applications the number of lags used is $L = 4$.

2.4 SIMULATION

In this section we analyze and compare the performance of the statistical learning methods presented[5] in section 2.2 through a Monte Carlo simulation study. Similarly to Konzen and Ziegelmann (2016), we perform Monte Carlo simulations with 1000 replications, simulating $n = 10$ independent time series with an AR(1) structure

$$x_{t,j} = \phi x_{t,j} + \varepsilon_{t,j}, \tag{2.35}$$

where $\phi = 0.5$ and $\varepsilon_{t,j} \sim \mathcal{N}(0,1)$, $j = 1, ..., n$ and all $\varepsilon_{t,j}$ are independent.

We consider three different data-generating processes (DGP). The first one has a specification similar to that used in Konzen and Ziegelmann (2016), being the most sparse model between the three:

$$\begin{aligned} \text{DGP 1: } y_t = 0.8y_{t-1} &+ 0.6x_{t-1,1} + 0.3x_{t-2,1} \\ &- 0.5x_{t-1,2} - 0.2x_{t-2,2} \\ &+ 0.4x_{t-1,3} + 0.3x_{t-2,3} \\ &+ 0.4x_{t-1,4} \\ &- 0.3x_{t-1,5} \\ &+ 0.2x_{t-1,6} + u_t, \quad t = 1, ..., T, \end{aligned} \tag{2.36}$$

where $u_t \sim \mathcal{N}(0,1)$ and all u_t are mutually independent.

The second DGP is given by:

$$\text{DGP 2: } y_t = \sum_{\ell=1}^{L} a_\ell \left(y_{t-\ell}\right) + \sum_{\ell=1}^{L} \sum_{j=1}^{n} b_{\ell,j} \left(x_{t-\ell,j}\right) + u_t, \quad t = 1, ..., T, \tag{2.37}$$

[5] As our simulated data do not have a factor structure we do not report results for factors methods in this section.

where $a_\ell = 0.8\,(-0.5)^{\ell-1}$, $b_{\ell,j} = (0.5)^{\ell}\,(-1)^{\ell+j-1}$, $u_t \sim \mathcal{N}(0,1)$ and all u_t are mutually independent. In this case (DGP 2), for each value of L, we generate and analyse the simulated data employing all lagged variables from 1 to L, thus all coefficients are different from zero.

Finally, our third DGP is a nonlinear model

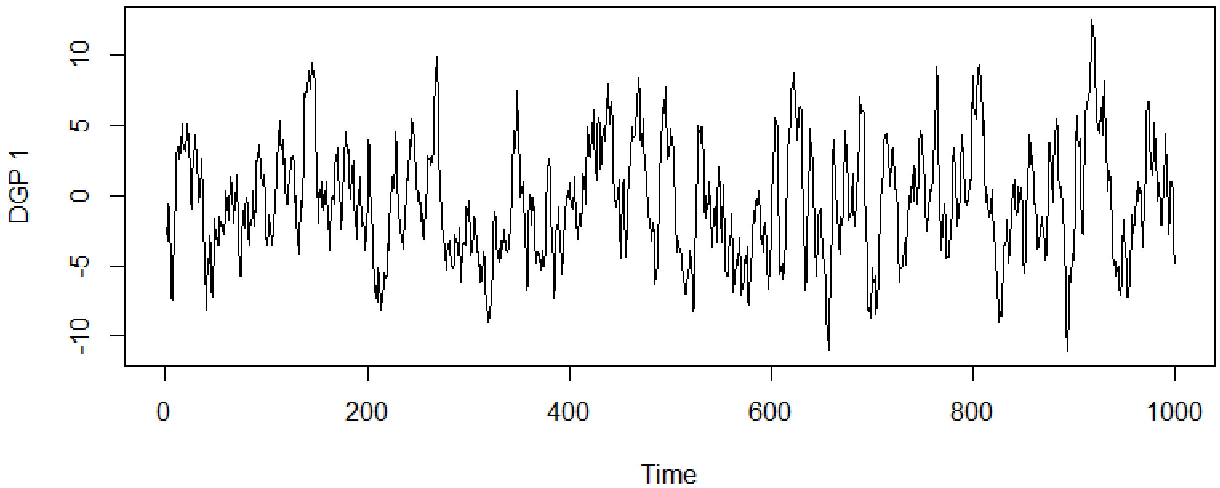
$$\text{DGP 3: } y_t = \sum_{\ell=1}^{L}\{a_\ell\,[g\,(y_{t-\ell};a_\ell)]\} + \sum_{l=1}^{L}\sum_{j=1}^{n}\left\{b_{\ell,j}\left[g\left(x_{t-\ell,j};b_{\ell,j}\right)\right]\right\} + u_t, \quad t = 1,...,T, \quad (2.38)$$

where $g(z;c) = z/(1+|c|z^2)$, $u_t \sim \mathcal{N}(0,1)$ and all u_t are mutually independent. In terms of Taylor's expansion $g(z;c) = z - |c|z^3 + |c|^2z^5 - |c|^3z^7 + |c|^4z^9...$, such that the DGP 3 is similar to the DGP 2 plus a nonlinear part. We remove the last 10 observations of the simulated data end employ all the methods (except for the factors based) to perform the one-step-ahead pseudo out-of-sample forecast for these observations using a rolling-window scheme, where the window has size $T-10$. We analyze situations where $L \in \{4, 12\}$ and $T \in \{150, 500, 1000\}$. Figure 1 presents the time series plots for variables y_t with 1000 observations for the three DGPs.

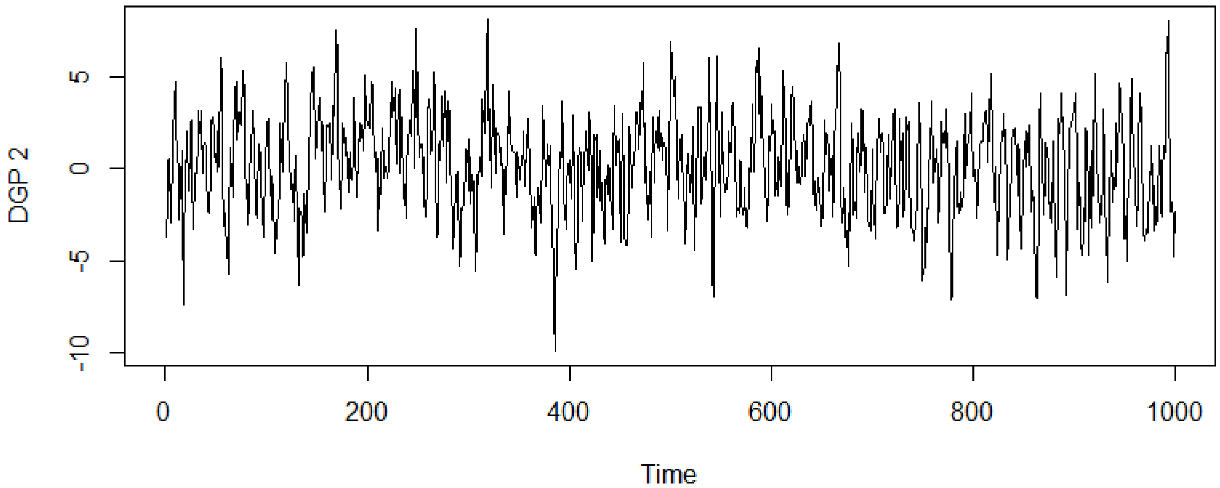
For the first two specifications, which are linear models, we report in Table 2 some statistics related to model/variable selection. Although the Ridge Regression is not a variable selection method, we included its statistics as a reference (when all variables are selected) to compare to the performance of other methods. For each method, sample size and lag order, we report: (1) the average fraction of variables correctly identified (FVCI); (2) the fraction of replications where the true model is included (TMI); (3) the average fraction of relevant variables included (FRVI); (4) the fraction of irrelevant variables excluded (FIVE) and (5) the number of included variables (NIV). While Ridge Regression, by construction, always select all variables and does not exclude any, the CSR always selects $\tilde{K} = 20$ variables and excludes the remaining ones. For each specification, the value of statistic for the best performing method is highlighted in bold.

Considering DGP 1, the adaptive methods, especially WLadaENet, have the best performance for the statistics FVCI, FIVE and NIV. When we consider the statistics TMI and FRV, the methods ENet and L_2Boost perform better than the other methods, while CSR rarely includes the true model (TMI near or equal to zero). For the second DGP, as all coefficients are different from zero, the statistic FIVE is zero for all methods and for each specification. For the TMI statistic (besides Ridge) only LASSO and ENet present values different from zero but very small, only for $L = 4$ and $T \geq 500$. ENet, L_2Boost (for $T = 150$ and 500) and LASSO (for $T = 1000$) present the best performance in terms of the FVCI, FRVI and NIV statistics when $L = 4$. For $L = 12$, L_2Boost has the best performance (excluding Ridge) for FVCI, FRVI and NIV.

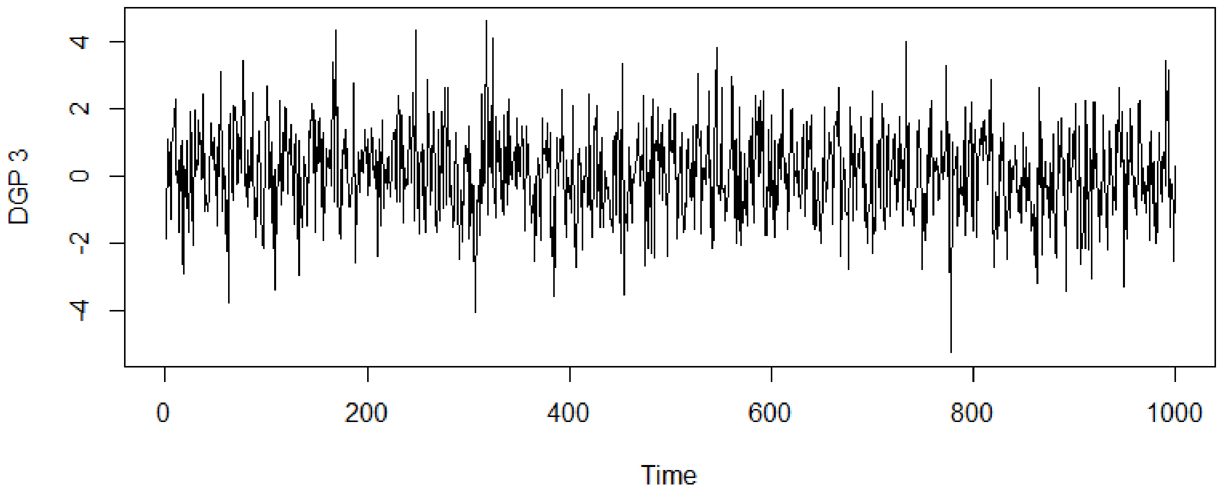
Table 4 shows the results for the one-step-ahead forecasts. We report the mean values of RMSE and MAE across replications, and indicate in bold the method with the lowest forecasting error for each specification. The cells in gray/blue indicate that the method is included in the 50% Model Confidence Set (MCS) using the squared/absolute error as loss function. For DGP 1 and 2 the methods with best performance in terms of errors are WLadaLASSO and WLadaENet,



(a) DGP 1



(b) DGP 2



(c) DGP 3

Figure 1 – Time series plots of variable y_t.

except for DGP 2 when $L = 4$ and $T = 1000$, where ENet has the lowest errors and is the only method included in the MCS. As we use all point forecasts of all replications and, consequently, this data is very informative, we have at most three methods included in the MCS and one or two methods in most cases. For the third DGP, which is nonlinear, when we have four lags the methods RF, B.Trees and L_2 have the best performance for the smallest simulated sample, and are the only methods included in the MCS. For the moderate and the largest sample sizes, Ridge has the lowest errors and is the only method in the MCS. Finally, for 12 lags, the model becomes approximately sparse, then WLadaLASSO and WLadaENet present the lowest errors, being the best performing methods.

In our simulations, when we have a model with a linear sparse structure, the shrinkage methods which perform variable selection, especially those who penalize differently each variable and further penalize the variables with longer lags (WLadaLASSO and WLadaENet), have a better predictive performance. The same occurs when the number of lags increases. When a nonlinear structure is introduced, the nonlinear methods have a slight advantage in terms of predictive performance for small samples and moderate number of lags. In this case, when the sample size increases Ridge Regression improves its performance. When the lag order increases, and the coefficients of variables with higher lags decreases, WLadaLASSO and WLadaENet present the lowest forecasting errors.

Table 2 – Simulation results: Descriptive statistics of models selection for DGP 1 and DGP 2

	DGP 1: Sparse model						DGP 2: Dense model					
	4 candidate lags			12 candidate lags			4 candidate lags			12 candidate lags		
T	150	500	1000	150	500	1000	150	500	1000	150	500	1000
	FVCI			FVCI			FVCI			FVCI		
Ridge	0.2273	0.2273	0.2273	0.0758	0.0758	0.0758	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
LASSO	0.8928	0.9342	0.9472	0.9490	0.9744	0.9790	0.4114	0.4625	0.8274	0.1643	0.1300	0.1323
ENet	0.7389	0.7740	0.7749	0.8746	0.9142	0.9176	0.4873	**0.5516**	**0.9073**	0.1932	0.1546	0.1535
adaLASSO	0.9339	0.9771	0.9850	0.9575	0.9906	0.9946	0.3316	0.3567	0.5960	0.1392	0.1013	0.1131
adaENet	0.9225	0.9727	0.9825	0.9459	0.9887	0.9935	0.3469	0.3669	0.6058	0.1518	0.1044	0.1135
WLadaLASSO	0.9562	0.9900	0.9943	**0.9680**	0.9967	0.9981	0.2941	0.3874	0.5320	0.0851	0.1310	0.1674
WLadaENet	**0.9566**	**0.9906**	**0.9952**	0.9668	**0.9968**	**0.9984**	0.2904	0.3874	0.5334	0.0847	0.1317	0.1674
CSR	0.6230	0.6667	0.6763	0.8415	0.8563	0.8588	0.4545	0.4545	0.4545	0.1515	0.1515	0.1515
L_2Boost	0.8329	0.8771	0.8867	0.8202	0.9015	0.9153	**0.4938**	0.5328	0.6928	**0.3612**	**0.2578**	**0.2793**
	TMI			TMI			TMI			TMI		
Ridge	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
LASSO	0.5922	0.9947	**1.0000**	0.2995	0.9757	**1.0000**	0.0000	0.0013	0.0714	0.0000	0.0000	0.0000
ENet	**0.7800**	**0.9990**	**1.0000**	0.4745	0.9907	**1.0000**	0.0000	**0.0037**	**0.1518**	0.0000	0.0000	0.0000
adaLASSO	0.4075	0.9529	0.9990	0.0565	0.9153	0.9987	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
adaENet	0.4397	0.9543	0.9990	0.0643	0.9278	0.9990	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
WLadaLASSO	0.4243	0.9735	**1.0000**	0.0637	0.9717	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
WLadaENet	0.4563	0.9793	**1.0000**	0.0829	0.9786	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
CSR	0.0001	0.0012	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
L_2Boost	0.7574	0.9998	**1.0000**	**0.6485**	**0.9988**	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	FRVI			FRVI			FRVI			FRVI		
Ridge	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
LASSO	0.9451	0.9995	**1.0000**	0.8756	0.9975	**1.0000**	0.4114	0.4625	0.8274	0.1643	0.1300	0.1323
ENet	**0.9766**	0.9999	**1.0000**	0.9300	0.9991	**1.0000**	0.4873	**0.5516**	**0.9073**	0.1932	0.1546	0.1535
adaLASSO	0.9145	0.9952	0.9999	0.7698	0.9913	0.9999	0.3316	0.3567	0.5960	0.1392	0.1013	0.1131
adaENet	0.9246	0.9953	0.9999	0.7924	0.9927	0.9999	0.3469	0.3669	0.6058	0.1518	0.1044	0.1135
WLadaLASSO	0.9009	0.9973	**1.0000**	0.7007	0.9971	**1.0000**	0.2941	0.3874	0.5320	0.0851	0.1310	0.1674
WLadaENet	0.9070	0.9979	**1.0000**	0.7050	0.9978	**1.0000**	0.2904	0.3874	0.5334	0.0847	0.1317	0.1674
CSR	0.6705	0.7667	0.7878	0.4542	0.5516	0.5678	0.4545	0.4545	0.4545	0.1515	0.1515	0.1515
L_2Boost	0.9728	**1.0000**	**1.0000**	**0.9586**	**0.9999**	**1.0000**	**0.4938**	0.5328	0.6928	**0.3612**	**0.2578**	**0.2793**
	FIVE			FIVE			FIVE			FIVE		
Ridge	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-	-	-	-	-	-
LASSO	0.8775	0.9150	0.9317	0.9550	0.9725	0.9773	-	-	-	-	-	-
ENet	0.6690	0.7075	0.7087	0.8700	0.9072	0.9108	-	-	-	-	-	-
adaLASSO	0.9396	0.9717	0.9806	0.9728	0.9905	0.9942	-	-	-	-	-	-
adaENet	0.9219	0.9661	0.9773	0.9585	0.9884	0.9930	-	-	-	-	-	-
WLadaLASSO	**0.9725**	0.9878	0.9926	**0.9899**	0.9966	0.9979	-	-	-	-	-	-
WLadaENet	0.9712	**0.9884**	**0.9938**	0.9883	**0.9968**	**0.9982**	-	-	-	-	-	-
CSR	0.6090	0.6372	0.6435	0.8733	0.8813	0.8826	-	-	-	-	-	-
L_2Boost	0.7917	0.8410	0.8534	0.8089	0.8934	0.9084	-	-	-	-	-	-
	NIV			NIV			NIV			NIV		
Ridge	44.0000	44.0000	44.0000	132.0000	132.0000	132.0000	**44.0000**	**44.0000**	**44.0000**	**132.0000**	**132.0000**	**132.0000**
LASSO	13.6171	12.8855	12.3235	14.2468	13.3297	12.770	18.101	20.3482	36.4060	21.6831	17.162	17.4666
ENet	21.0193	19.944	19.9051	25.1581	21.3117	20.8805	21.4407	**24.2682**	**39.9232**	25.4983	20.4126	20.259
adaLASSO	11.1996	10.9134	10.6582	**11.0108**	11.0684	10.7089	14.5902	15.6959	26.2219	18.3707	13.3685	14.9258
adaENet	11.9012	11.1070	10.7692	12.9883	11.3414	10.8542	15.2643	16.1446	26.6532	20.0439	13.7855	14.985
WLadaLASSO	9.9441	10.3865	10.2527	8.2447	10.3812	10.2553	12.9418	17.0468	23.4098	11.2356	17.2919	22.1009
WLadaENet	**10.0491**	**10.3726**	**10.212**	8.4804	**10.3736**	**10.2171**	12.778	17.0455	23.4694	11.1768	17.3846	22.0988
CSR	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000
L_2Boost	16.8083	15.4067	14.9853	32.9025	23.0002	21.1782	**21.728**	23.4424	30.4811	**47.6759**	**34.0347**	**36.8712**

Source: Own elaboration from research data (2020). Note: The table presents some statistics related to model/variable selection, where FVCI is the average fraction of variables correctly identified; TMI is the fraction of replications where the true model is included; FRVI is the average fraction of relevant variables included; FIVE is the fraction of irrelevant variables excluded and NIV is the number of included variables. For each specification, the value of statistic for the best performing method is highlighted in bold. Ridge Regression always select all variables and does not exclude any, while CSR always selects $\tilde{K} = 20$ variables and excludes the remaining ones.

Table 3 – Simulation results: Errors for the DGP 1, 2 and 3

Mean of RMSEs	DGP 1: Sparse model						DGP 2: Dense model						DGP 3: Nonlinear model					
(Mean of MAEs)	4 candidate lags			12 candidate lags			4 candidate lags			12 candidate lags			4 candidate lags			12 candidate lags		
T	150	500	1000	150	500	1000	150	500	1000	150	500	1000	150	500	1000	150	500	1000
RW	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)	(1.0000)
AR	0.9262	0.9175	0.9159	0.9269	0.9176	0.9159	0.9109	0.8983	0.8967	0.9172	0.9035	0.9027	0.7565	0.7525	0.7504	0.7558	0.7501	0.75
	(0.9275)	(0.9200)	(0.9183)	(0.9282)	(0.9199)	(0.9183)	(0.9105)	(0.8986)	(0.8972)	(0.9205)	(0.9074)	(0.9061)	(0.7671)	(0.7632)	(0.7608)	(0.7695)	(0.7634)	(0.7636)
Ridge	0.7274	0.6486	0.6342	1.8334	0.7171	0.6655	0.6007	0.523	0.5126	1.0237	0.5739	0.5338	0.7619	**0.6616**	**0.6479**	0.7334	0.7266	0.6733
	(0.7254)	(0.6451)	(0.6323)	(1.8488)	(0.7189)	(0.6652)	(0.5998)	(0.5225)	(0.5126)	(1.0286)	(0.5722)	(0.5337)	(0.7748)	**(0.6723)**	**(0.6588)**	(0.7444)	(0.7352)	(0.684)
LASSO	0.6456	0.5912	0.5809	0.6990	0.6023	0.5855	0.5906	0.5302	0.4952	0.6521	0.5520	0.5331	0.7491	0.6887	0.6683	0.7494	0.708	0.679
	(0.6443)	(0.5914)	(0.5809)	(0.6962)	(0.6021)	(0.5856)	(0.5883)	(0.529)	(0.4959)	(0.6513)	(0.5539)	(0.5339)	(0.759)	(0.6984)	(0.6782)	(0.7649)	(0.7187)	(0.6897)
ENet	0.6641	0.5989	0.5842	0.7572	0.6201	0.5954	0.5972	0.5282	**0.4921**	0.6964	0.5637	0.5404	0.7553	0.6938	0.6697	0.7535	0.7193	0.6829
	(0.6622)	(0.5970)	(0.5831)	(0.7530)	(0.6182)	(0.5942)	(0.5946)	(0.5264)	**(0.4934)**	(0.6947)	(0.5655)	(0.5413)	(0.7652)	(0.7039)	(0.6794)	(0.7691)	(0.7313)	(0.6937)
adaLASSO	0.6301	0.5836	0.5758	0.7115	0.586	0.5762	0.5696	0.5229	0.5002	0.6342	0.5259	0.5159	0.7406	0.681	0.6626	0.7441	0.6866	0.6649
	(0.6290)	(0.5840)	(0.5763)	(0.7099)	(0.5869)	(0.5766)	(0.5676)	(0.5219)	(0.5011)	(0.6328)	(0.5269)	(0.5169)	(0.7521)	(0.6914)	(0.6732)	(0.7592)	(0.6963)	(0.6753)
adaENet	0.6310	0.5842	0.5760	0.728	0.5867	0.5764	0.5722	0.5235	0.5000	0.6482	0.5275	0.5167	0.7441	0.6811	0.6624	0.7449	0.6891	0.6664
	(0.6297)	(0.5844)	(0.5765)	(0.7254)	(0.5872)	(0.5768)	(0.5699)	(0.5223)	(0.5006)	(0.6475)	(0.5286)	(0.5175)	(0.7552)	(0.6912)	(0.6726)	(0.7606)	(0.6989)	(0.6764)
WLadaLASSO	**0.6168**	**0.5799**	**0.5746**	0.6634	**0.5805**	**0.5747**	0.5548	**0.5155**	0.4966	0.5476	0.5093	0.4905	0.7263	0.6700	0.6591	0.7241	**0.6662**	0.6554
	(0.6151)	**(0.5804)**	(0.5751)	(0.6635)	**(0.5806)**	**(0.5754)**	**(0.5537)**	**(0.5149)**	(0.4972)	**(0.5469)**	(0.5102)	(0.4907)	(0.7368)	(0.6811)	(0.6692)	(0.7366)	**(0.6748)**	(0.6654)
WLadaENet	0.6170	0.5804	**0.5746**	**0.6633**	0.5806	0.5749	**0.5546**	0.5157	0.4969	**0.5474**	**0.5088**	**0.4904**	0.7258	0.6698	0.659	**0.72**	0.6663	**0.6553**
	(0.6156)	(0.5808)	**(0.575)**	**(0.6632)**	**(0.5806)**	**(0.5754)**	**(0.5537)**	**(0.5149)**	(0.4976)	**(0.5469)**	**(0.5097)**	**(0.4906)**	(0.7356)	(0.6806)	(0.6696)	**(0.7326)**	(0.6749)	**(0.665)**
CSR	1.3605	1.3028	1.2857	1.4464	1.3489	1.3245	0.9554	0.9366	0.9308	0.9750	0.9313	0.9259	0.7302	0.7233	0.7223	0.7367	0.7233	0.7207
	(1.3633)	(1.3082)	(1.2897)	(1.4459)	(1.3522)	(1.3261)	(0.9583)	(0.9379)	(0.9323)	(0.9775)	(0.9371)	(0.9310)	(0.7405)	(0.7331)	(0.7323)	(0.7497)	(0.736)	(0.7332)
L_2Boost	0.6265	0.5852	0.5772	0.6660	0.5909	0.5800	0.5762	0.5248	0.5057	0.6230	0.5331	0.5192	0.7202	0.6761	0.6621	0.7449	0.6818	0.6657
	(0.6263)	(0.5856)	(0.5775)	(0.6658)	(0.5921)	(0.5805)	(0.5739)	(0.5236)	(0.5060)	(0.6227)	(0.5342)	(0.5205)	(0.7306)	(0.6859)	(0.6722)	(0.7558)	(0.691)	(0.6758)
RF	1.3050	1.0176	0.9151	1.3858	1.0666	0.9561	0.9367	0.8420	0.8008	0.9805	0.8739	0.8303	0.7196	0.691	0.6776	0.7322	0.706	0.6918
	(1.2686)	(0.9999)	(0.9039)	(1.3495)	(1.0495)	(0.9450)	(0.9349)	(0.8382)	(0.7974)	(0.9823)	(0.8755)	(0.8315)	(0.7295)	(0.7014)	(0.6874)	(0.7452)	(0.7179)	(0.7028)
B.Trees	1.4524	1.0613	0.9660	1.6040	1.0661	0.9676	0.9410	0.8390	0.8177	1.0001	0.8353	0.8110	**0.7194**	0.687	0.675	0.7327	0.6957	0.6769
	(1.4109)	(1.0401)	(0.9504)	(1.5656)	(1.0446)	(0.9518)	(0.9400)	(0.8354)	(0.8140)	(1.0004)	(0.8357)	(0.8121)	**(0.7292)**	(0.6984)	(0.685)	(0.7476)	(0.7075)	(0.6878)

Source: Own elaboration from research data (2020). Note: The table shows the means of root mean squared errors (RMSE) and means of mean absolute errors (MAE) in parenthesis for the forecasts across replications, relative to the Random Walk (RW). The values in bold indicate the method with lowest values of RMSE and MAE for each horizon. Cells in gray/blue indicate that the method is included in the 50% MCS constructed based on the T_{max} statistic using the squared/absolute errors.

Table 4 – Simulation results: Test for Superior Predictive Ability (SPA)

Panel (a): Squared errors

	DGP 1: Sparse model						DGP 2: Dense model						DGP 3: Nonlinear model					
	4 candidate lags			12 candidate lags			4 candidate lags			12 candidate lags			4 candidate lags			12 candidate lags		
T	150	500	1000	150	500	1000	150	500	1000	150	500	1000	150	500	1000	150	500	1000
RW	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
AR	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Ridge	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.500	0.522	0.000	0.000	0.000
LASSO	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ENet	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.509	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
adaLASSO	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
adaENet	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
WLadaLASSO	0.655	0.479	0.492	0.717	0.704	0.903	0.369	0.767	0.000	0.281	0.015	0.320	0.004	0.000	0.000	0.717	0.704	0.903
WLadaENet	0.352	0.010	0.540	0.720	0.285	0.105	0.622	0.236	0.000	0.682	0.515	0.654	0.010	0.000	0.000	0.720	0.285	0.105
CSR	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
L_2Boost	0.000	0.000	0.000	0.233	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.407	0.000	0.000	0.233	0.000	0.000
RF	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.653	0.000	0.000	0.000	0.000	0.000
B.Trees	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.730	0.000	0.000	0.000	0.000	0.000

Panel (b): Absolute errors

	DGP 1: Sparse model						DGP 2: Dense model						DGP 3: Nonlinear model					
	4 candidate lags			12 candidate lags			4 candidate lags			12 candidate lags			4 candidate lags			12 candidate lags		
T	150	500	1000	150	500	1000	150	500	1000	150	500	1000	150	500	1000	150	500	1000
RW	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
AR	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Ridge	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.484	0.498	0.000	0.000	0.000
LASSO	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ENet	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.512	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
adaLASSO	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001
adaENet	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
WLadaLASSO	0.796	0.943	0.229	0.671	0.599	0.465	0.467	0.492	0.000	0.546	0.100	0.292	0.008	0.000	0.000	0.671	0.599	0.465
WLadaENet	0.214	0.048	0.767	0.759	0.423	0.514	0.542	0.490	0.002	0.435	0.902	0.698	0.031	0.004	0.000	0.759	0.423	0.514
CSR	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
L_2Boost	0.000	0.000	0.000	0.296	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.340	0.000	0.000	0.296	0.000	0.000
RF	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.668	0.000	0.000	0.000	0.000	0.000
B.Trees	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.751	0.000	0.000	0.000	0.000	0.000

Source: Own elaboration from research data (2020). Note: The table reports the p-values of the test for Superior Predictive Ability of Hansen (2005) using each method as benchmark for each forecasting horizon, including the accumulated horizons. The Panel (a) presents the p-values for the test using the squared errors and the Panel (b) using the absolute errors. The null hypothesis is that the benchmark is not inferior to any alternative method. The gray cells indicate that the null hypothesis is rejected at the 0.05 significance level.

2.5 EMPIRICAL ANALYSIS

In this section we present two empirical applications of inflation forecasting, where we employ all high-dimensional statistical learning methods presented in section 2.2 to forecast Brazilian and U.S. inflation. Table 5 summarizes all the specifications of the empirical applications presented in this work, while Figure 2 presents the time series plots of Brazilian inflation and core inflation, and U.S. inflation, where the red intervals indicate the period for which we perform the out-of-sample forecasts.

Table 5 – Empirical applications summary

	Variables	Lags	q	In-sample	Sample size	Out-of-Sample	Point forecasts
CPI BR (IPCA)	95	4	380	Jan 1999-Dec 2012	168	Jan 2013-Dec 2018	72
CPI Core BR (IPCA-EX0)	95	4	380	Jan 1999-Dec 2012	168	Jan 2013-Dec 2018	72
CPI U.S. small sample	122	4	488	Jan 1999-Dec 2012	168	Jan 2013-Dec 2018	72
CPI U.S. large sample	122	4	488	Jan 1960-Dec 2012	636	Jan 2013-Dec 2018	72

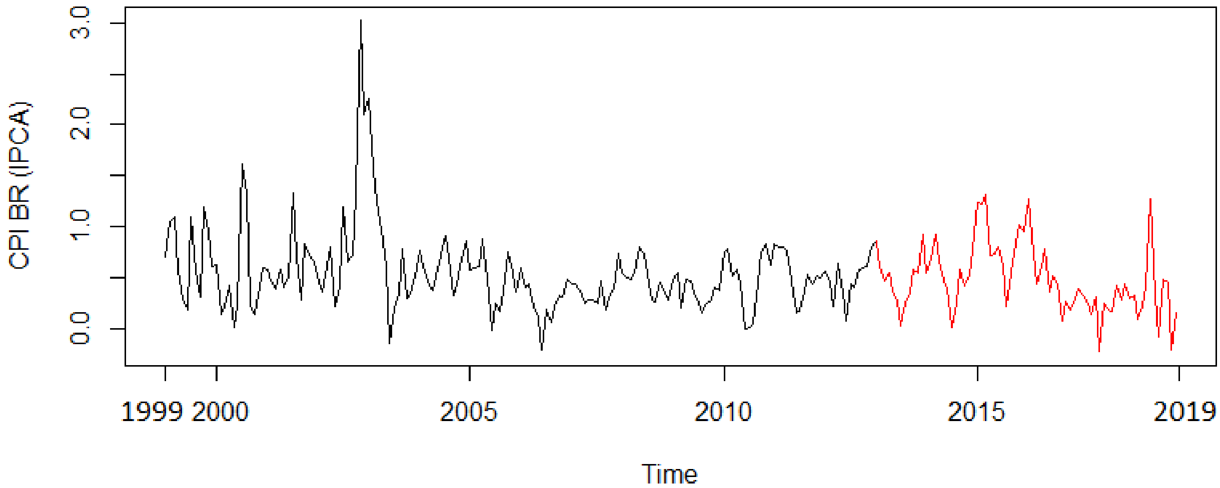
Source: Own elaboration from research data (2020).

2.5.1 Brazilian Inflation

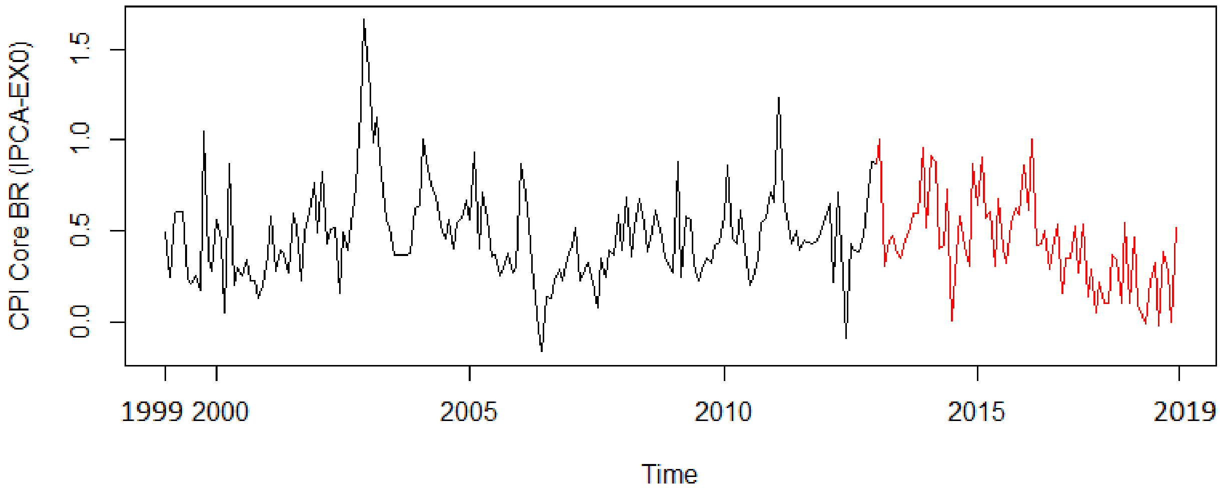
Inflation is measured using the Broad National Consumer Price Index (IPCA), which is the Brazilian official price index. Our dataset consists of macroeconomic and financial variables obtained from the Central Bank of Brazil, the Institute for Applied Economic Research (Ipea) and the Brazilian Institute of Geography and Statistics (IBGE). To choose which variables to include in the data set we followed Medeiros *et al.* (2016). The dataset covers the period from January 1999 to December 2018, having 240 observations and 95 variables. The variables are classified in the same 8 groups used in Medeiros *et al.* (2016), namely, Group 1 - Prices; Group 2 - Employment and Wages; Group 3 - International Transactions and Government Debt; Group 4 - Economic Activity and Production; Group 5: Taxes and Government Income; Group 6 - Exchange Rates and Finance; Group 7 - Money and Group 8 - Economic Confidence.

The forecasts are based on a rolling-window scheme of 168 observations, such that we have more variables ($q = 380$, considering 95 variables and 4 lags) than observations ($T_0 = 168$). All methods are evaluated based on 72 point forecasts, where the out of sample period goes from January 2013 to December 2018. We assume that the variables in the "Prices" group (in percentage change) are stationary. The variables in the other groups are tested for stationarity by using both the Augmented Dick-Fuller test (ADF) and the Kwiatkowski–Phillips–Schmidt–Shin (KPSS). The list of variables and the transformations used in order achieve stationarity are presented in Appendix A.

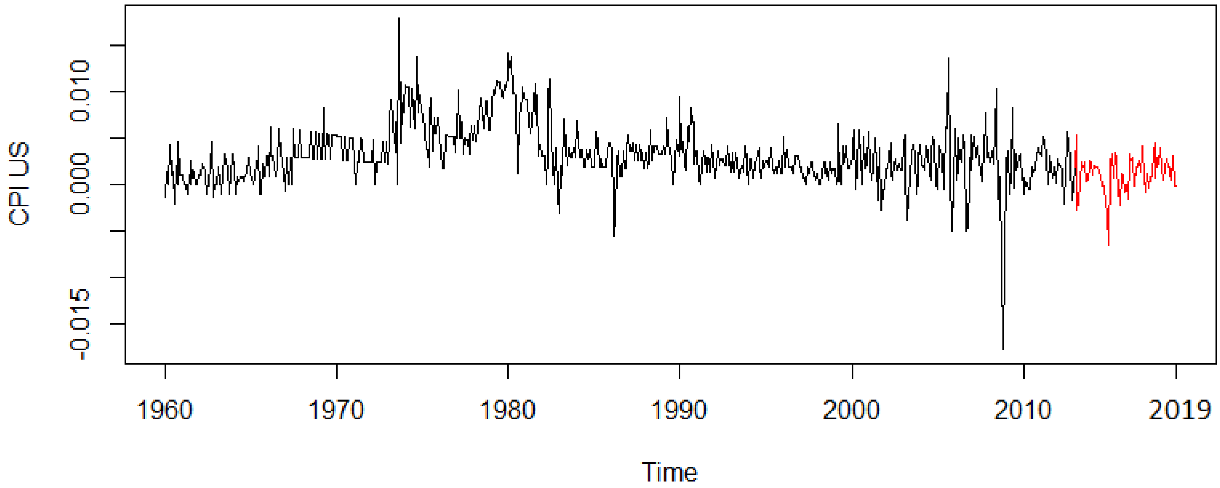
Table 6 shows the root mean squared error (RMSE) and the mean absolute error (MAE) for all forecasting methods including the mean and the median of all forecasts. The forecasts for 1 to 12 months ahead are reported as well as the accumulated forecasting inflation for 3, 6 and 12 months. The values of h-month accumulated inflation are computed using the forecasts from 1 to h steps ahead, $h \in \{3, 6, 12\}$, except for the RW method where the forecast for the h-months



(a) CPI BR (% change)



(b) CPI Core BR (% change)



(c) CPI U.S. (log change)

Figure 2 – Time series plots of BR and U.S. inflation.

accumulated inflation is computed using the accumulated inflation of the h previous months. The smallest errors (RMSE and MAE) are indicated in bold for each forecasting horizon. The gray/blue cells indicate that the method is included in the 50% Model Confidence Set (MCS) using the squared /absolute error as loss function.

Three methods are included in the MCS for all forecasting horizons (considering both squared and absolute errors), namely, Ridge Regression, CSR and B.Trees. Considering the RMSE, Ridge Regression is the most accurate method for all forecasting horizons, including accumulated except for $h = 1$, 3 and 4. For $h = 1$ and 3 the best method is B.Factors while for $h = 4$, B.Trees is the most accurate. For accumulated inflation Ridge Regression presents the lowest errors while L_2Boost is the second best method for 12-months accumulated inflation.

We also performed forecasts for the core inflation IPCA-EX0[6] using the same dataset and for the same periods in-sample and out-of-sample. Table 7 presents the results for the forecasts of the core inflation. As it can be seen now the MCS includes fewer methods, mainly for $h = 6,7$ and 8 and for the 12-months accumulated inflation. Ridge Regression is the best method for most horizons, while for the 12 months accumulates inflation L_2Boost presented the lowest errors and adaLASSO for $h = 1$.

Tables 8 and 9 present the results for Superior Predictive Ability (SPA). For the test for SPA of Hansen (2005) we use each method as benchmark and compare to the others, where the null hypothesis is that the benchmark is not inferior to any forecasting alternative. For the uniform and average multi-horizon tests for SPA of Quaedvlieg (2019) we use the RW and AR methods as benchmarks and compare to each methods. The null hypothesis is that the benchmark is better than the alternative method. For both forecasts of inflation and core inflation the p-values of the muti-horizon test for SPA indicates that the benchmark RW is not superior to any alternative method at a significance level of 5% considering the squared error function. For inflation forecasts the AR benchmark is not only statistically superior to Ridge, CSR, RF and B.Trees while for the core inflation forecasts, the AR method is not superior to Ridge, CSR and RF. The test for SPA of Hansen (2005) shows that Ridge, CSR, RF and B.Trees are not inferior to any alternative for inflation forecasts for all horizons while for the core inflation forecasts only Ridge, CSR and RF are not inferior to the alternatives.

Figures 3 to 6, inspired by Medeiros *et al.* (2019), show the plots of relative variable importance (aggregated by variable groups) for all twelve forecasting horizons and for all methods except the univariate and factors based methods, for inflation and core inflation forecasts. Similarly to Medeiros *et al.* (2019), for methods based on the linear model (shrinkage methods, CSR and L_2Boost) the relative importance is computed as the average coefficient[7] size. For RF the Out-of-Bag (OOB) samples[8] are used to compute the variable importance, while for B.Trees all samples are employed. The samples are passed down the b_{th} tree when it is grown and the

[6] IPCA excluding food at home and administered prices.
[7] All variables are standardized having mean zero and standard deviation one.
[8] The OOB samples are the observations which are not selected in the bootstrap sample process, for each $b = 1,...,B$.

accuracy is recorded, then the values for the variable j are permuted at random, and the accuracy is computed once more. Then the decrease in accuracy is averaged over all trees and is used as a measure of the importance of variable j in the forest (HASTIE *et al.*, 2009). A missing bar means that no variable is selected.

For Ridge Regression the relative variable importance is stable across the forecasting horizons and does not change much between the inflation and the core inflation forecasts. The variable importance is on average about 25% for Autoregressive terms plus the Group 1 (Prices) and for Group 4 (Economic Activity and Production) in both cases. For CSR method the Autoregressive terms (used as controls) plus the variables of Group 1 sum on average about 75% for the inflation forecasts and about 65% for the core. RF, B.Trees and L_2Boost present a pattern of variable importance near to Ridge but less stable across the horizons, where the importance of Autorregressive terms plus Group 1 represent on average about 45%. For LASSO and ENet the importance of Prices is about 45% for inflation and 35% for the core the Group 2 (Employment and Wages) is the second most important for inflation while for the core is Group 5 (Taxes and Government Income). When we consider the adaptive methods the importance of Prices is about 50% for inflation and 40% for the core. Group 5 is clearly the second most important for the core, while for inflation are Groups 2 and 6 (Exchange Rates and Finance).

Table 6 – Forecasting results: Errors for the CPI (Brazil) from 2013 to 2018

	Consumer price index (Brazil) 2013-2018 Forecast horizon														
RMSE/(MAE)	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m
RW	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)
AR	0.89	0.84	0.83	0.78	0.71	0.72	0.74	0.76	0.83	0.84	0.89	0.88	0.80	0.77	0.75
	(0.89)	(0.84)	(0.79)	(0.75)	(0.75)	(0.72)	(0.74)	(0.73)	(0.80)	(0.85)	(0.88)	(0.92)	(0.76)	(0.75)	(0.80)
Ridge	0.89	**0.80**	0.81	0.74	**0.67**	**0.64**	**0.65**	**0.68**	**0.74**	**0.78**	**0.84**	**0.82**	**0.73**	**0.62**	**0.65**
	(0.84)	**(0.78)**	(0.77)	**(0.70)**	**(0.68)**	**(0.61)**	**(0.61)**	**(0.65)**	**(0.73)**	(0.80)	(0.82)	**(0.86)**	**(0.70)**	**(0.60)**	**(0.66)**
LASSO	0.92	0.86	0.86	0.80	0.71	0.69	0.71	0.74	0.81	0.85	0.88	0.93	0.81	0.76	0.78
	(0.86)	(0.86)	(0.87)	(0.79)	(0.75)	(0.68)	(0.72)	(0.73)	(0.80)	(0.87)	(0.87)	(0.99)	(0.80)	(0.75)	(0.81)
ENet	0.93	0.84	0.85	0.78	0.70	0.71	0.72	0.75	0.81	0.85	0.88	0.88	0.81	0.77	0.79
	(0.87)	(0.87)	(0.85)	(0.76)	(0.74)	(0.70)	(0.73)	(0.73)	(0.80)	(0.85)	(0.87)	(0.93)	(0.79)	(0.75)	(0.84)
adaLASSO	0.96	0.87	0.93	0.86	0.71	0.69	0.71	0.76	0.86	0.87	0.88	0.89	0.85	0.77	0.75
	(0.92)	(0.87)	(0.94)	(0.86)	(0.76)	(0.68)	(0.71)	(0.75)	(0.84)	(0.90)	(0.86)	(0.95)	(0.82)	(0.75)	(0.76)
adaENet	0.91	0.87	0.87	0.81	0.70	0.69	0.68	0.74	0.82	0.87	0.88	0.88	0.81	0.75	0.76
	(0.89)	(0.87)	(0.87)	(0.81)	(0.75)	(0.68)	(0.68)	(0.72)	(0.81)	(0.88)	(0.87)	(0.94)	(0.79)	(0.74)	(0.78)
WLadaLASSO	0.96	0.91	0.90	0.84	0.80	0.71	0.69	0.76	0.87	0.84	0.86	0.90	0.84	0.78	0.75
	(0.93)	(0.90)	(0.92)	(0.82)	(0.82)	(0.70)	(0.70)	(0.76)	(0.85)	(0.86)	(0.86)	(0.96)	(0.80)	(0.74)	(0.76)
WLadaENet	0.93	0.86	0.89	0.80	0.80	0.71	0.68	0.73	0.82	0.84	0.87	0.88	0.81	0.77	0.76
	(0.90)	(0.88)	(0.90)	(0.80)	(0.82)	(0.72)	(0.69)	(0.73)	(0.81)	(0.86)	(0.86)	(0.94)	(0.77)	(0.75)	(0.77)
CSR	0.89	0.84	**0.80**	0.75	0.69	0.67	0.66	0.70	0.76	0.80	0.85	0.85	0.76	0.71	0.70
	(0.84)	(0.85)	(0.80)	(0.73)	(0.74)	(0.65)	(0.65)	(0.68)	**(0.73)**	**(0.79)**	(0.82)	(0.88)	(0.72)	(0.66)	(0.70)
L_2Boost	1.01	0.90	0.94	0.93	0.76	0.72	0.72	0.76	0.85	0.85	0.88	0.89	0.84	0.76	0.67
	(1.00)	(0.88)	(0.90)	(0.85)	(0.74)	(0.69)	(0.68)	(0.69)	(0.82)	(0.87)	(0.85)	(0.89)	(0.78)	(0.71)	(0.69)
RF	0.89	0.84	0.84	0.78	0.69	0.67	0.67	0.71	0.76	0.81	0.85	0.85	0.79	0.73	0.78
	(0.84)	(0.83)	(0.80)	(0.76)	(0.73)	(0.67)	(0.67)	(0.70)	(0.76)	(0.82)	(0.85)	(0.90)	(0.76)	(0.71)	(0.80)
B.Trees	0.91	0.82	**0.80**	**0.73**	**0.67**	0.68	0.68	0.70	0.79	0.82	**0.84**	0.83	0.77	0.69	0.76
	(0.88)	(0.81)	**(0.76)**	**(0.70)**	(0.70)	(0.67)	(0.68)	(0.69)	(0.78)	(0.80)	**(0.81)**	(0.90)	(0.75)	(0.68)	(0.80)
Factors	0.89	0.83	0.83	0.80	0.72	0.72	0.73	0.75	0.84	0.87	0.90	0.86	0.77	0.71	0.75
	(0.88)	(0.81)	(0.78)	(0.75)	(0.74)	(0.72)	(0.72)	(0.76)	(0.84)	(0.88)	(0.90)	(0.93)	(0.73)	(0.68)	(0.73)
B.Factors	0.89	0.82	**0.80**	0.74	0.71	0.73	0.74	0.76	0.81	0.86	0.89	0.86	0.74	0.68	0.73
	(0.83)	(0.79)	(0.78)	(0.71)	(0.75)	(0.74)	(0.76)	(0.74)	(0.80)	(0.85)	(0.88)	(0.92)	(0.71)	(0.68)	(0.77)
Mean	**0.87**	0.81	**0.80**	0.75	0.69	0.67	0.67	0.71	0.78	0.81	0.85	0.84	0.75	0.69	0.73
	(0.83)	(0.80)	(0.80)	(0.72)	(0.72)	(0.65)	(0.67)	(0.69)	(0.77)	(0.82)	(0.84)	(0.90)	(0.74)	(0.68)	(0.73)
Median	0.88	0.82	0.81	0.76	0.69	0.67	0.68	0.72	0.80	0.82	0.86	0.87	0.77	0.71	0.72
	(0.83)	(0.81)	(0.81)	(0.75)	(0.73)	(0.66)	(0.69)	(0.71)	(0.78)	(0.84)	(0.85)	(0.93)	(0.75)	(0.70)	(0.73)

Source: Own elaboration from research data (2020). Note: The table shows the root mean squared errors (RMSE) and mean absolute errors (MAE) in parenthesis for the forecasts, relative to the Random Walk (RW). The values in bold indicate the method with lowest values of RMSE and MAE for each horizon. Cells in gray/blue indicate that the method is included in the 50% MCS constructed based on the T_{max} statistic using the squared/absolute errors.

Table 7 – Forecasting results: Errors for the CPI Core (Brazil) from 2013 to 2018

	Consumer price index - Core (Brazil) 2013-2018														
	Forecast horizon														
RMSE/(MAE)	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m
RW	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)
AR	0.81	0.86	0.85	0.80	0.74	0.81	0.67	0.68	0.64	0.81	0.67	0.93	0.88	0.91	1.02
	(0.82)	(0.89)	(0.85)	(0.79)	(0.72)	(0.79)	(0.70)	(0.67)	(0.69)	(0.85)	(0.69)	(0.97)	(0.87)	(0.93)	(1.14)
Ridge	0.69	0.79	0.76	**0.69**	**0.60**	**0.63**	**0.54**	**0.54**	**0.58**	0.74	0.61	0.86	**0.71**	**0.67**	0.82
	(0.67)	**(0.75)**	(0.77)	**(0.69)**	**(0.60)**	**(0.64)**	(0.58)	**(0.55)**	**(0.62)**	(0.75)	**(0.60)**	(0.87)	**(0.72)**	**(0.67)**	(0.93)
LASSO	0.71	0.82	0.78	0.77	0.68	0.78	0.65	0.68	0.71	0.85	0.68	0.96	0.80	0.89	1.10
	(0.69)	(0.81)	(0.79)	(0.77)	(0.67)	(0.78)	(0.69)	(0.68)	(0.74)	(0.88)	(0.70)	(1.00)	(0.85)	(0.87)	(1.22)
ENet	0.76	0.96	0.85	0.78	0.69	0.80	0.70	0.74	0.79	0.96	0.76	1.04	0.94	0.95	1.18
	(0.76)	(0.93)	(0.87)	(0.78)	(0.68)	(0.79)	(0.73)	(0.74)	(0.81)	(0.98)	(0.77)	(1.07)	(1.02)	(0.94)	(1.29)
adaLASSO	**0.67**	0.79	0.79	0.79	0.70	0.75	0.64	0.66	0.67	0.81	0.66	0.92	0.78	0.88	1.06
	(0.67)	(0.80)	(0.81)	(0.80)	(0.68)	(0.75)	(0.66)	(0.68)	(0.69)	(0.83)	(0.69)	(0.96)	(0.77)	(0.83)	(1.18)
adaENet	0.68	0.80	0.80	0.79	0.69	0.81	0.65	0.68	0.69	0.84	0.68	0.96	0.80	0.91	1.10
	(0.68)	(0.80)	(0.80)	(0.80)	(0.67)	(0.82)	(0.69)	(0.69)	(0.71)	(0.86)	(0.70)	(0.99)	(0.80)	(0.88)	(1.22)
WLadaLASSO	0.68	0.80	0.79	0.81	0.72	0.78	0.63	0.66	0.69	0.85	0.67	0.90	0.79	0.88	1.02
	(0.68)	(0.80)	(0.81)	(0.82)	(0.69)	(0.77)	(0.65)	(0.68)	(0.71)	(0.88)	(0.69)	(0.94)	(0.79)	(0.83)	(1.15)
WLadaENet	0.68	0.86	0.78	0.78	0.70	0.80	0.66	0.68	0.71	0.85	0.67	0.95	0.80	0.89	1.07
	(0.68)	(0.88)	(0.78)	(0.78)	(0.67)	(0.79)	(0.70)	(0.69)	(0.74)	(0.89)	(0.70)	(0.99)	(0.80)	(0.86)	(1.20)
CSR	0.71	0.77	0.80	0.75	0.67	0.70	0.55	0.59	**0.58**	**0.71**	0.60	**0.82**	0.77	0.81	0.94
	(0.70)	(0.78)	(0.80)	(0.76)	(0.67)	(0.69)	**(0.57)**	(0.59)	**(0.62)**	**(0.73)**	(0.61)	**(0.83)**	(0.79)	(0.81)	(1.07)
L_2Boost	**0.67**	0.82	0.78	0.82	0.69	0.71	0.65	0.63	0.63	0.86	0.63	0.88	0.73	0.75	**0.81**
	(0.71)	(0.86)	(0.82)	(0.84)	(0.69)	(0.72)	(0.65)	(0.63)	(0.66)	(0.87)	(0.66)	(0.93)	(0.74)	(0.71)	**(0.83)**
RF	0.72	0.82	0.81	0.73	0.64	0.65	0.56	0.58	0.61	0.75	**0.59**	0.83	0.81	0.78	0.93
	(0.73)	(0.81)	(0.81)	(0.73)	(0.63)	**(0.64)**	(0.58)	(0.58)	(0.63)	(0.78)	(0.61)	(0.85)	(0.84)	(0.78)	(1.02)
B.Trees	0.80	0.93	0.87	0.77	0.71	0.75	0.65	0.68	0.71	0.89	0.72	1.05	0.90	0.86	1.07
	(0.80)	(0.89)	(0.86)	(0.77)	(0.68)	(0.73)	(0.67)	(0.69)	(0.74)	(0.93)	(0.73)	(1.08)	(0.97)	(0.88)	(1.13)
Factors	0.76	0.81	0.88	0.85	0.79	0.84	0.67	0.75	0.77	0.84	0.69	0.94	0.83	0.92	1.07
	(0.74)	(0.78)	(0.88)	(0.86)	(0.74)	(0.86)	(0.70)	(0.73)	(0.80)	(0.84)	(0.69)	(0.96)	(0.87)	(0.92)	(1.18)
B.Factors	0.72	0.84	0.87	0.78	0.71	0.77	0.65	0.64	0.66	0.80	0.67	0.90	0.82	0.84	1.04
	(0.74)	(0.84)	(0.88)	(0.79)	(0.68)	(0.76)	(0.67)	(0.64)	(0.70)	(0.81)	(0.67)	(0.93)	(0.84)	(0.84)	(1.15)
Mean	0.68	0.77	**0.75**	0.72	0.65	0.71	0.60	0.63	0.64	0.77	0.64	0.88	0.74	0.79	0.98
	(0.67)	(0.77)	(0.76)	(0.71)	(0.64)	(0.70)	(0.62)	(0.63)	(0.66)	(0.79)	(0.65)	(0.91)	(0.77)	(0.79)	(1.08)
Median	0.68	**0.76**	**0.75**	0.75	0.67	0.75	0.61	0.62	0.63	0.77	0.64	0.89	0.75	0.82	1.00
	(0.68)	(0.76)	**(0.75)**	(0.74)	(0.65)	(0.74)	(0.63)	(0.62)	(0.66)	(0.80)	(0.66)	(0.92)	(0.79)	(0.81)	(1.11)

Source: Own elaboration from research data (2020). Note: The table shows the root mean squared errors (RMSE) and mean absolute errors (MAE) in parenthesis for the forecasts, relative to the Random Walk (RW). The values in bold indicate the method with lowest values of RMSE and MAE for each horizon. Cells in gray/blue indicate that the method is included in the 50% MCS constructed based on the T_{max} statistic using the squared/absolute errors.

Table 8 – Forecasting results: Superior predictive ability test (CPI - Brazil, 2013-2018)

Panel (a): Squared errors

	Hansen‘s test – Forecasting horizon															Quaedvlieg test			
Benchmark	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m	Unif.(RW)	Avg.(RW)	Unif.(AR)	Avg.(AR)
RW	0.156	0.051	0.050	0.010	0.000	0.000	0.002	0.006	0.028	0.062	0.118	0.131	0.010	0.003	0.053	-	-	0.944	0.991
AR	0.755	0.521	0.637	0.509	0.538	0.094	0.109	0.110	0.160	0.487	0.451	0.601	0.316	0.073	0.388	0.000	0.007	-	-
Ridge	0.797	0.916	0.687	0.799	0.821	0.978	0.975	0.961	0.974	0.970	0.916	0.988	0.840	0.751	0.818	0.000	0.006	0.088	0.041
LASSO	0.633	0.450	0.423	0.366	0.548	0.205	0.163	0.184	0.282	0.379	0.477	0.292	0.201	0.037	0.306	0.000	0.004	0.357	0.586
ENet	0.525	0.518	0.514	0.487	0.574	0.111	0.143	0.165	0.265	0.437	0.470	0.525	0.262	0.063	0.286	0.000	0.005	0.405	0.351
adaLASSO	0.264	0.332	0.148	0.160	0.532	0.253	0.243	0.075	0.178	0.261	0.470	0.459	0.175	0.018	0.382	0.000	0.003	0.648	0.892
adaENet	0.690	0.363	0.450	0.319	0.590	0.252	0.594	0.234	0.198	0.319	0.451	0.496	0.206	0.057	0.353	0.000	0.002	0.422	0.465
WLadaLASSO	0.259	0.162	0.270	0.166	0.078	0.128	0.455	0.064	0.171	0.509	0.757	0.381	0.180	0.024	0.377	0.000	0.010	0.566	0.893
WLadaENet	0.586	0.402	0.285	0.400	0.079	0.086	0.627	0.370	0.235	0.536	0.611	0.509	0.194	0.021	0.352	0.000	0.005	0.571	0.728
CSR	0.788	0.500	0.898	0.784	0.675	0.603	0.761	0.658	0.818	0.740	0.933	0.736	0.666	0.210	0.406	0.000	0.006	0.156	0.037
L_2Boost	0.090	0.242	0.170	0.093	0.187	0.106	0.184	0.186	0.280	0.465	0.397	0.345	0.252	0.116	0.588	0.018	0.004	0.768	0.837
RF	0.838	0.606	0.689	0.626	0.827	0.547	0.720	0.572	0.733	0.705	0.864	0.871	0.395	0.135	0.292	0.000	0.004	0.139	0.049
B.Trees	0.664	0.741	0.986	0.981	0.918	0.414	0.488	0.544	0.484	0.597	0.910	0.822	0.627	0.318	0.337	0.000	0.005	0.076	0.020
Factors	0.786	0.594	0.716	0.394	0.482	0.070	0.111	0.134	0.140	0.322	0.420	0.733	0.608	0.244	0.408	0.000	0.004	0.618	0.541
B.Factors	0.774	0.713	0.940	0.853	0.538	0.045	0.074	0.124	0.236	0.343	0.401	0.729	0.875	0.409	0.473	0.000	0.002	0.414	0.126
Mean	1.000	0.914	0.986	0.852	0.845	0.507	0.785	0.479	0.547	0.688	0.991	0.924	0.888	0.220	0.096	0.000	0.003	0.000	0.011
Median	0.953	0.790	0.933	0.736	0.675	0.497	0.559	0.418	0.311	0.670	0.764	0.672	0.571	0.150	0.459	0.000	0.003	0.000	0.019

Panel (b): Absolute errors

	Hansen‘s test – Forecasting horizon															Quaedvlieg test			
Benchmark	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m	Unif.(RW)	Avg.(RW)	Unif.(AR)	Avg.(AR)
RW	0.084	0.031	0.044	0.009	0.001	0.000	0.001	0.001	0.021	0.067	0.081	0.214	0.012	0.002	0.046	-	-	0.887	0.985
AR	0.496	0.412	0.600	0.478	0.227	0.049	0.059	0.119	0.248	0.523	0.388	0.608	0.441	0.063	0.210	0.000	0.012	-	-
Ridge	0.838	0.893	0.706	0.902	0.992	0.970	0.956	0.995	0.892	0.795	0.786	0.980	0.970	0.963	0.949	0.000	0.003	0.038	0.035
LASSO	0.817	0.335	0.244	0.182	0.245	0.152	0.029	0.081	0.177	0.339	0.446	0.241	0.158	0.025	0.195	0.012	0.010	0.653	0.839
ENet	0.683	0.311	0.372	0.418	0.315	0.074	0.064	0.084	0.385	0.444	0.419	0.461	0.265	0.052	0.148	0.000	0.012	0.658	0.715
adaLASSO	0.302	0.264	0.064	0.081	0.107	0.156	0.093	0.019	0.194	0.170	0.451	0.371	0.185	0.017	0.378	0.000	0.009	0.717	0.899
adaENet	0.523	0.257	0.255	0.094	0.253	0.180	0.258	0.094	0.302	0.257	0.396	0.396	0.243	0.040	0.300	0.000	0.016	0.563	0.703
WLadaLASSO	0.254	0.146	0.113	0.184	0.099	0.053	0.121	0.016	0.185	0.417	0.589	0.311	0.252	0.039	0.381	0.000	0.016	0.715	0.906
WLadaENet	0.439	0.236	0.148	0.135	0.037	0.038	0.179	0.108	0.316	0.456	0.495	0.399	0.376	0.017	0.314	0.000	0.003	0.684	0.858
CSR	0.890	0.336	0.698	0.682	0.377	0.455	0.418	0.515	0.874	0.914	0.839	0.937	0.831	0.438	0.519	0.000	0.006	0.176	0.053
L_2Boost	0.061	0.278	0.198	0.188	0.326	0.144	0.265	0.388	0.319	0.381	0.439	0.709	0.409	0.200	0.568	0.005	0.019	0.659	0.647
RF	0.900	0.504	0.679	0.492	0.349	0.220	0.300	0.348	0.608	0.739	0.652	0.731	0.456	0.091	0.183	0.000	0.007	0.158	0.107
B.Trees	0.622	0.655	0.971	0.907	0.735	0.208	0.232	0.373	0.363	0.808	0.944	0.720	0.521	0.260	0.177	0.000	0.009	0.006	0.031
Factors	0.550	0.628	0.805	0.573	0.308	0.027	0.069	0.029	0.201	0.279	0.248	0.584	0.697	0.272	0.246	0.000	0.011	0.658	0.635
B.Factors	0.880	0.875	0.869	0.825	0.254	0.008	0.031	0.056	0.236	0.437	0.372	0.650	0.883	0.259	0.310	0.000	0.012	0.515	0.220
Mean	0.995	0.902	0.745	0.879	0.425	0.344	0.266	0.292	0.558	0.752	0.770	0.879	0.764	0.192	0.256	0.000	0.009	0.074	0.037
Median	0.986	0.757	0.521	0.493	0.272	0.287	0.084	0.140	0.291	0.638	0.646	0.558	0.500	0.111	0.232	0.000	0.004	0.176	0.052

Source: Own elaboration from research data (2020). Note: The table reports the p-values of the test for Superior Predictive Ability of Hansen (2005) (left) using each method as benchmark for each forecasting horizon, including the accumulated horizons. The p-values of the uniform and average multi-horizon SPA test of Quaedvlieg (2019) are also reported (right) using RW ans AR as benchmarks. The Panel (a) presents the p-values for the test using the squared errors and the Panel (b) using the absolute errors. The null hypothesis of the single-horizon test is that the bench of the is that the bench mark is not inferior to any alternative method. The null hypothesis of the multi-horizon test is that the benchmark is superior to the alternative method. The gray cells indicate that the null hypothesis is rejected at the 0.05 significance level.

Table 9 – Forecasting results: Superior predictive ability test (CPI Core - Brazil, 2013-2018)

Panel (a): Squared errors

	Hansen‘s test – Forecasting horizon															Quaedvlieg test			
Benchmark	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m	Unif.(RW)	Avg.(RW)	Unif.(AR)	Avg.(AR)
RW	0.000	0.035	0.009	0.003	0.000	0.000	0.000	0.001	0.000	0.009	0.000	0.083	0.050	0.065	0.324	-	-	0.952	0.997
AR	0.013	0.145	0.075	0.085	0.020	0.007	0.036	0.015	0.354	0.225	0.052	0.116	0.138	0.104	0.162	0.000	0.006	-	-
Ridge	0.639	0.710	0.650	0.859	0.590	0.708	0.722	0.994	0.901	0.711	0.693	0.382	0.885	0.947	0.949	0.000	0.003	0.000	0.011
LASSO	0.411	0.382	0.492	0.194	0.073	0.017	0.025	0.006	0.013	0.080	0.020	0.065	0.362	0.154	0.062	0.000	0.007	0.859	0.074
ENet	0.105	0.055	0.175	0.136	0.084	0.011	0.018	0.006	0.007	0.024	0.002	0.033	0.097	0.113	0.067	0.173	0.007	0.915	0.990
adaLASSO	0.895	0.659	0.423	0.141	0.062	0.066	0.126	0.046	0.181	0.271	0.107	0.157	0.472	0.174	0.085	0.000	0.001	0.271	0.070
adaENet	0.778	0.577	0.376	0.123	0.085	0.004	0.035	0.011	0.065	0.104	0.057	0.089	0.408	0.142	0.049	0.000	0.007	0.569	0.146
WLadaLASSO	0.754	0.576	0.411	0.085	0.054	0.008	0.138	0.048	0.149	0.185	0.108	0.254	0.469	0.178	0.179	0.000	0.003	0.545	0.179
WLadaENet	0.804	0.260	0.504	0.169	0.095	0.007	0.021	0.010	0.048	0.161	0.096	0.083	0.400	0.162	0.067	0.000	0.005	0.723	0.187
CSR	0.408	0.891	0.320	0.332	0.163	0.234	0.586	0.341	0.840	0.912	0.674	0.803	0.427	0.270	0.447	0.000	0.001	0.000	0.004
L_2Boost	0.879	0.391	0.529	0.089	0.176	0.299	0.107	0.204	0.369	0.230	0.398	0.285	0.792	0.424	0.700	0.000	0.004	0.445	0.091
RF	0.335	0.385	0.182	0.569	0.388	0.485	0.538	0.378	0.604	0.620	0.797	0.661	0.352	0.384	0.422	0.000	0.002	0.000	0.008
B.Trees	0.011	0.090	0.113	0.180	0.038	0.066	0.073	0.025	0.071	0.086	0.003	0.024	0.129	0.176	0.123	0.242	0.005	0.885	0.885
Factors	0.152	0.443	0.059	0.007	0.002	0.001	0.044	0.002	0.000	0.123	0.032	0.093	0.364	0.068	0.085	0.000	0.005	0.943	0.974
B.Factors	0.283	0.267	0.102	0.139	0.058	0.017	0.054	0.067	0.172	0.272	0.076	0.233	0.330	0.203	0.130	0.000	0.006	0.454	0.052
Mean	0.935	0.950	0.994	0.474	0.195	0.135	0.189	0.056	0.302	0.426	0.114	0.274	0.756	0.285	0.244	0.000	0.003	0.039	0.010
Median	0.853	0.997	0.864	0.298	0.125	0.037	0.164	0.086	0.312	0.400	0.138	0.284	0.709	0.231	0.176	0.000	0.005	0.001	0.006

Panel (b): Absolute errors

	Hansen‘s test – Forecasting horizon															Quaedvlieg test			
Benchmark	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m	Unif.(RW)	Avg.(RW)	Unif.(AR)	Avg.(AR)
RW	0.000	0.035	0.024	0.008	0.000	0.001	0.000	0.000	0.000	0.019	0.000	0.138	0.093	0.084	0.354	-	-	0.922	0.995
AR	0.017	0.107	0.149	0.176	0.082	0.036	0.037	0.032	0.239	0.081	0.028	0.064	0.239	0.090	0.048	0.004	0.009	-	-
Ridge	0.841	0.856	0.756	0.836	0.970	0.692	0.648	0.969	0.934	0.580	0.722	0.631	0.898	0.735	0.504	0.000	0.002	0.000	0.011
LASSO	0.689	0.383	0.555	0.310	0.201	0.032	0.014	0.006	0.028	0.117	0.013	0.057	0.292	0.189	0.014	0.029	0.010	0.761	0.117
ENet	0.177	0.086	0.176	0.203	0.140	0.036	0.018	0.010	0.020	0.026	0.004	0.041	0.063	0.128	0.022	0.315	0.018	0.895	0.973
adaLASSO	0.942	0.483	0.402	0.261	0.170	0.084	0.115	0.026	0.222	0.322	0.054	0.109	0.610	0.251	0.019	0.000	0.010	0.081	0.098
adaENet	0.886	0.530	0.486	0.270	0.274	0.007	0.026	0.004	0.089	0.170	0.031	0.082	0.467	0.168	0.011	0.006	0.010	0.540	0.169
WLadaLASSO	0.820	0.502	0.360	0.182	0.142	0.015	0.163	0.027	0.171	0.084	0.064	0.150	0.554	0.261	0.035	0.000	0.007	0.337	0.166
WLadaENet	0.950	0.168	0.615	0.260	0.286	0.033	0.015	0.004	0.056	0.074	0.064	0.064	0.482	0.208	0.012	0.003	0.006	0.556	0.215
CSR	0.677	0.681	0.510	0.329	0.226	0.333	0.833	0.431	0.810	0.857	0.796	0.847	0.607	0.279	0.104	0.000	0.004	0.000	0.007
L_2Boost	0.540	0.267	0.316	0.159	0.197	0.262	0.198	0.246	0.482	0.215	0.287	0.241	0.774	0.561	0.915	0.000	0.003	0.363	0.165
RF	0.304	0.413	0.344	0.636	0.534	0.735	0.664	0.408	0.702	0.525	0.647	0.689	0.276	0.352	0.260	0.000	0.005	0.000	0.008
B.Trees	0.030	0.138	0.179	0.301	0.167	0.164	0.069	0.013	0.080	0.048	0.008	0.028	0.092	0.172	0.104	0.305	0.010	0.871	0.724
Factors	0.315	0.676	0.104	0.046	0.033	0.016	0.045	0.006	0.002	0.143	0.098	0.105	0.271	0.077	0.031	0.001	0.011	0.896	0.912
B.Factors	0.312	0.272	0.108	0.172	0.190	0.033	0.072	0.097	0.138	0.368	0.132	0.251	0.344	0.220	0.039	0.000	0.010	0.474	0.047
Mean	0.978	0.870	0.930	0.621	0.449	0.245	0.274	0.074	0.366	0.329	0.186	0.244	0.779	0.306	0.096	0.000	0.004	0.000	0.010
Median	0.923	0.924	0.961	0.456	0.338	0.085	0.147	0.122	0.423	0.246	0.116	0.207	0.620	0.260	0.055	0.000	0.003	0.000	0.007

Source: Own elaboration from research data (2020). Note: The table reports the p-values of the test for Superior Predictive Ability of Hansen (2005) (left) using each method as benchmark for each forecasting horizon, including the accumulated horizons. The p-values of the uniform and average multi-horizon SPA test of Quaedvlieg (2019) are also reported (right) using RW ans AR as benchmarks. The Panel (a) presents the p-values for the test using the squared errors and the Panel (b) using the absolute errors. The null hypothesis of the single-horizon test is that the bench of the is that the bench mark is not inferior to any alternative method. The null hypothesis of the multi-horizon test is that the benchmark is superior to the alternative method. The gray cells indicate that the null hypothesis is rejected at the 0.05 significance level.

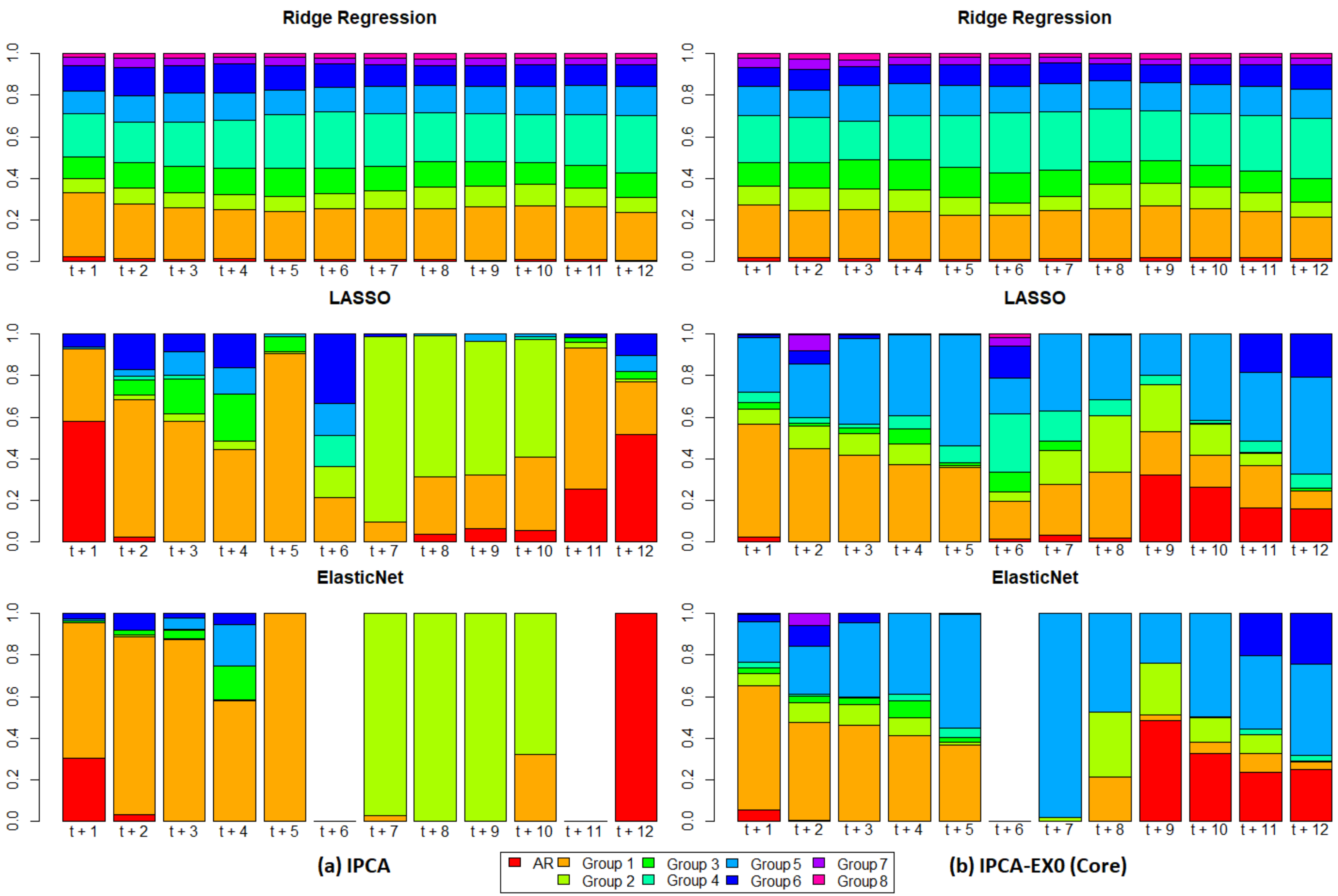


Figure 3 – Variable importance of methods Ridge, LASSO and ENet for Brazilian IPCA (a) and IPCA-EX0 (b).

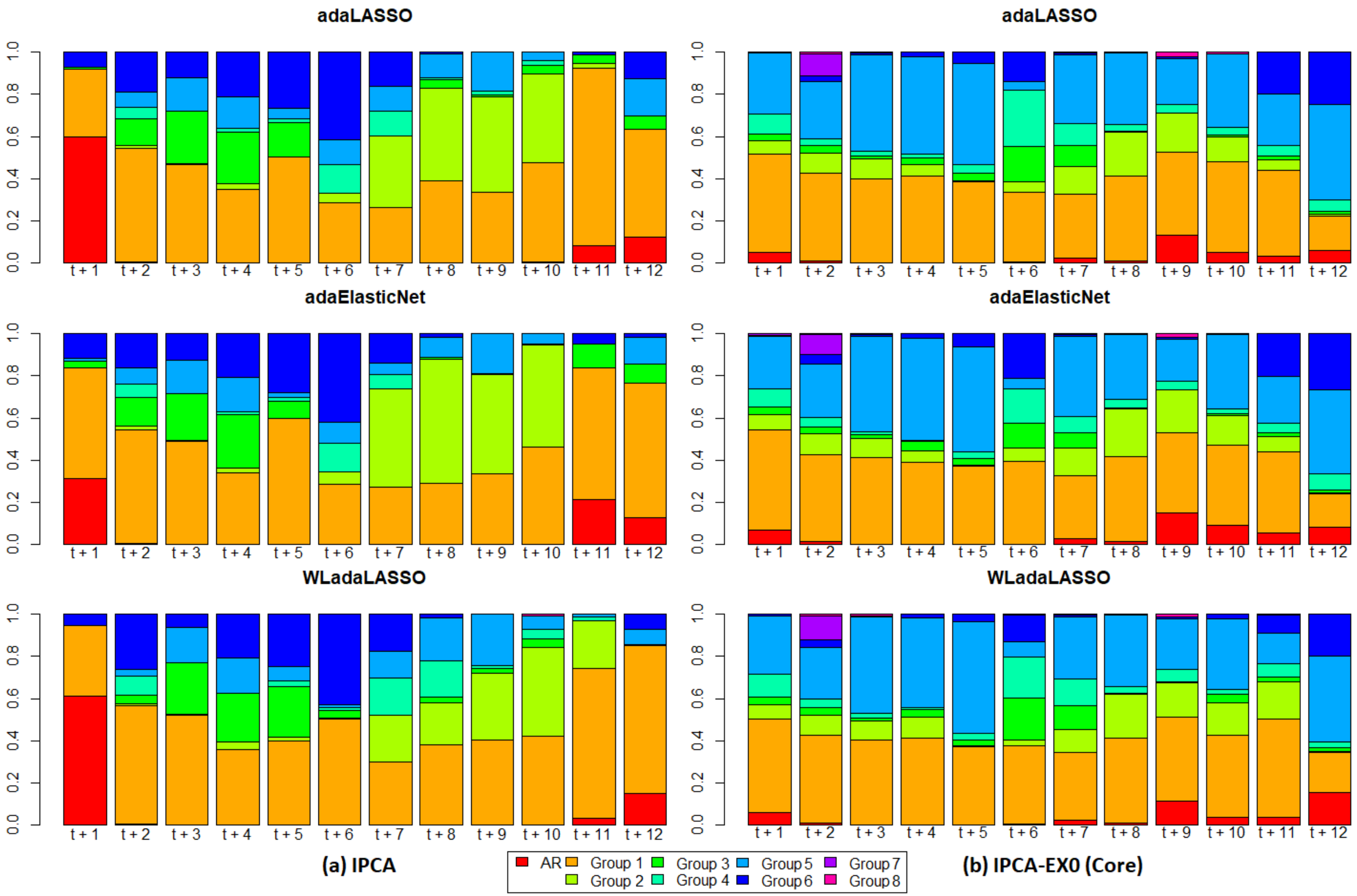


Figure 4 – Variable importance of methods adaLASSO, adaENet and WLadaLASSO for Brazilian IPCA (a) and IPCA-EX0 (b).

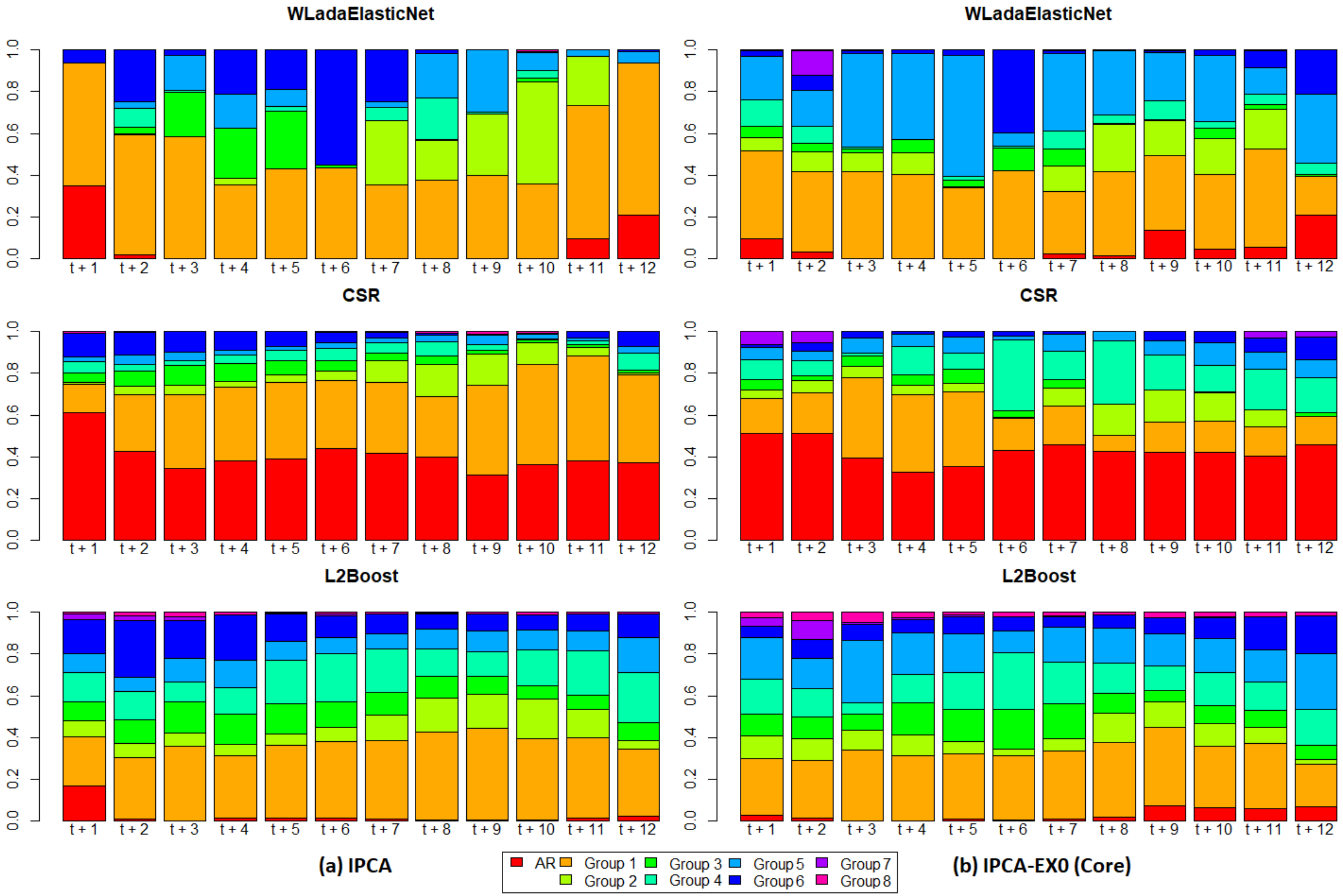


Figure 5 – Variable importance of methods WLadaENet, CSR and L_2Boost for Brazilian IPCA (a) and IPCA-EX0 (b).

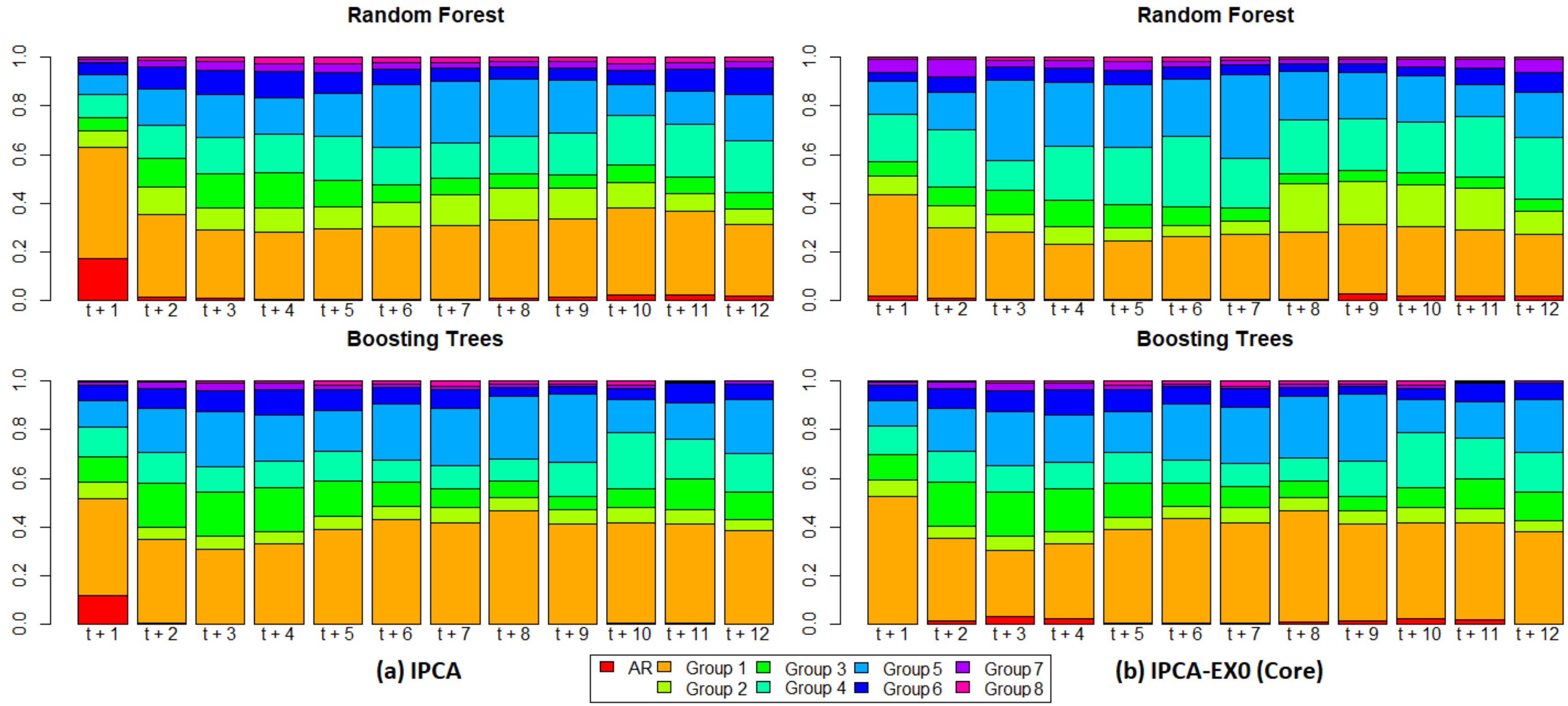


Figure 6 – Variable importance of methods RF and B.Trees for Brazilian IPCA (a) and IPCA-EX0 (b).

2.5.2 U.S. Inflation

For U.S. inflation we use the variables from the FRED-MD[9] database compiled by McCracken and Ng (2016), performing forecasts only for the Consumer Price Index (CPI) in log change. The dataset we use covers the period from January 1960 to December 2018, having 708 observations and 122 variables classified in 8 groups: Group 1 - Output and income; Group 2 - Labor market; Group 3 - Housing; Group 4 - Consumption, orders, and inventories; Group 5 - Money and credit; Group 6 - Interest and exchange rates; Group 7 - Prices and Group 8 - Stock market. In order to achieve stationarity employ the transformations recommended by McCracken and Ng (2016), except for the Group 7 (Prices) where we use the same transformations employed in Medeiros *et al.* (2019). The list of variables and the transformations used are available in Appendix B.

We compute forecasts for the same period out-of-sample used for Brazilian inflation, from January 2013 to December 2018 but varying the period in-sample. where for the first forecast we use the same period we used in the first application, from January 1999 to December 2012 (small sample). We also perform forecasts using the period in-sample from January 1960 to December 2012 (large sample). In the first case we have more variables ($q = 488$, considering 122 variables and 4 lags) than samples ($T_0 = 168$), while in the second case the sample size ($T_0 = 636$) is higher than the number of predictors ($q = 488$).

Table 10 shows the results for the forecasts using the small sample. RW, CSR and L_2Boost are excluded from the MCS for almost all horizons. For $h = 1$ adaENet and WLadaENet present the lowest errors, while for the 12-months accumulated inflation RW is the best method in terms of RMSE, while RF presents the lowest MAE. We do not have a most accurate method for the most horizons, but RF presents the lowest errors in 6 horizons, in cases which RF is not the most accurate method its errors are not more than 10% (or 5% in most cases) higher than the best method. ENet (except for $h = 1$) and the adaptive methods also present a good performance in general.

In Table 11 we report the result for the forecasts using the large sample. As can be seen, on the one hand, the performance of AR and Ridge decrease over all horizons with when we use the large sample and, on the other hand, CSR and L_2Boost increase their performances, such that L_2Boost becomes the most accurate method for the most horizons, while WLadaLASSO and WLadaENet present the lowest errors for h=1. L_2Boost is the most accurate method especially when we consider the 3 accumulated horizons. WLadaENet also presents a good performance having the lowest errors in 7 horizons when we consider both RMSE and MAE. The performance reduction of Ridge is not due to the sample size increasing itself. As expected in theory and empirically observed the performance of Ridge improves as the sample size increases. Moreover, Zou and Hastie (2005) affirm that, for usual situations where $T > q$, the performance of LASSO

[9] The FRED-MD is a large database containing monthly observations of macroeconomic variables, designed for the analysis of *big data* and is updated in real-time thorough the FRED database. The FRED-MD is available at https://research.stlouisfed.org/econ/mccracken/fred-databases/.

is dominated by Ridge if there are high correlations between predictors.

The p-values relative to the SPA tests are presented in Tables 12 and 13. When we consider the forecasts based on the small sample, the multi-horizon test for SPA of RW benchmark indicates that it only is not inferior to LASSO at a significance level of 0.05, while the benchmark AR is only on average inferior to ENet and adaENet. For the forecasts based on the large sample, both benchmarks (RW and AR) are not uniformly inferior to the Ridge, while AR is also not inferior on average. For the single-horizon SPA test and considering the forecasts for the small sample, the methods AR, ENet, adaLASSO, adaENet, WLadaLASSO, WLadaENet, RF, B.Trees, Factors and B.Factors are statistically not inferior to the alternatives for all forecasting horizons. But for the large samples the methods LASSO, ENet, adaLASSO, WLadaLASSO, WLadaENet, CSR, L_2Boost, RF and B.Trees are statistically not inferior to the alternative methods.

Figures 7 to 10 present the relative variable importance for the U.S. inflation forecasts. The values of variable importance are computed as described in the previous section. The values of variable importance are stable across the horizons for Ridge Regression, and do not change much for either forecast based on the small sample or on the large sample, where in both cases the variable importance of Autoregressive terms plus Group 7 (Prices) is about 20%. Group 2 (Labor market) is the group where the variable importance is higher, being about 25%. In the first case ENet do not select any variable in most forecasting horizons.

When we consider the forecasts based on the small sample Group 6 (Interest and exchange rates) presents the highest percentage of variable importance in most cases. For the forecasts based on the large sample, except for Ridge Regression, the relative importance of Autoregressive terms plus the Prices group assumes values between 50% and 80% (on average). When the sample size is large LASSO and ENet present a similar pattern of variable importance and the respective values of RMSE and MAE in Table 11 also do not differ much. The same situation occurs to the adaptive versions of ENet and LASSO.

Table 10 – Forecasting results: Errors for the CPI (U.S.) from 2013 to 2018 - small sample

	Consumer price index (U.S.) 2013-2018 - small sample Forecast horizon														
RMSE/(MAE)	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m
RW	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**
	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)
AR	0.85	0.74	0.75	0.76	0.77	0.68	0.64	0.69	0.70	0.76	0.77	0.74	0.81	0.78	1.12
	(0.87)	(0.72)	(0.68)	(0.71)	(0.74)	(0.61)	(0.57)	(0.62)	(0.63)	(0.70)	(0.70)	(0.66)	(0.78)	(0.71)	(1.03)
Ridge	0.80	0.76	0.75	0.78	0.76	**0.66**	**0.62**	0.68	0.72	0.85	0.90	0.88	0.81	0.77	1.08
	(0.81)	(0.73)	(0.71)	(0.74)	(0.73)	**(0.60)**	**(0.56)**	(0.64)	(0.70)	(0.84)	(0.87)	(0.83)	(0.83)	(0.76)	(1.07)
LASSO	0.73	0.74	0.95	1.18	1.29	0.68	0.71	0.76	0.81	0.77	1.24	1.27	0.80	0.88	1.22
	(0.76)	(0.73)	(0.86)	(1.12)	(1.13)	(0.62)	(0.63)	(0.69)	(0.71)	(0.71)	(1.08)	(1.18)	(0.82)	(0.84)	(1.18)
ENet	0.82	0.74	0.73	0.76	0.76	0.67	0.65	0.69	0.70	0.77	0.76	**0.71**	0.80	0.77	1.09
	(0.81)	(0.72)	(0.68)	(0.71)	(0.73)	(0.61)	(0.57)	(0.61)	(0.64)	(0.71)	(0.68)	**(0.62)**	**(0.77)**	**(0.70)**	(1.00)
adaLASSO	0.73	0.77	0.74	0.77	0.76	0.68	0.66	0.70	0.71	0.78	0.76	0.74	0.78	0.77	1.08
	(0.77)	(0.76)	(0.68)	(0.72)	(0.74)	(0.62)	(0.58)	(0.62)	(0.64)	(0.72)	(0.68)	(0.64)	(0.79)	(0.72)	(0.99)
adaENet	**0.71**	0.76	0.73	0.76	0.76	0.68	0.65	0.69	0.70	0.78	0.76	**0.71**	0.78	0.77	1.07
	(0.74)	(0.75)	(0.68)	(0.72)	(0.73)	(0.61)	(0.57)	(0.61)	(0.64)	(0.72)	(0.68)	**(0.62)**	(0.78)	(0.71)	(0.98)
WLadaLASSO	0.76	0.78	0.74	0.80	0.76	0.68	0.66	0.70	0.73	0.79	0.76	0.74	0.79	0.78	1.10
	(0.81)	(0.78)	(0.70)	(0.74)	(0.74)	(0.62)	(0.58)	(0.63)	(0.65)	(0.72)	(0.68)	(0.65)	(0.82)	(0.73)	(1.00)
WLadaENet	**0.71**	0.76	0.74	0.79	0.76	0.68	0.65	0.69	0.71	0.79	0.76	0.72	0.78	0.77	1.08
	(0.74)	(0.75)	(0.68)	(0.73)	(0.73)	(0.61)	(0.57)	(0.61)	(0.65)	(0.72)	(0.68)	**(0.62)**	(0.78)	(0.71)	(0.98)
CSR	0.82	0.81	0.83	0.85	0.86	0.76	0.70	0.74	0.77	0.86	0.85	0.82	0.89	0.95	1.45
	(0.88)	(0.78)	(0.78)	(0.80)	(0.84)	(0.71)	(0.62)	(0.66)	(0.73)	(0.83)	(0.78)	(0.75)	(0.87)	(0.89)	(1.33)
L_2Boost	0.91	0.85	1.00	0.91	0.92	0.75	0.68	0.74	0.86	0.93	0.90	0.87	0.94	0.91	1.19
	(0.97)	(0.86)	(1.00)	(0.91)	(0.91)	(0.68)	(0.61)	(0.68)	(0.78)	(0.92)	(0.84)	(0.81)	(0.98)	(0.90)	(1.15)
RF	0.78	**0.72**	0.73	0.77	0.76	0.69	0.64	**0.67**	**0.69**	0.77	0.78	0.74	0.78	0.77	1.07
	(0.80)	**(0.70)**	**(0.67)**	(0.72)	(0.74)	(0.63)	**(0.56)**	**(0.60)**	**(0.62)**	(0.71)	(0.69)	(0.66)	(0.78)	(0.71)	**(0.95)**
B.Trees	0.81	0.74	0.75	**0.75**	**0.75**	0.67	0.64	**0.67**	0.72	0.79	0.77	0.75	0.82	**0.76**	1.10
	(0.84)	(0.73)	(0.69)	(0.72)	**(0.72)**	(0.61)	(0.57)	**(0.60)**	(0.68)	(0.73)	(0.70)	(0.67)	(0.81)	**(0.70)**	(1.01)
Factors	0.78	0.78	0.79	0.80	0.79	0.70	0.67	0.71	0.70	**0.74**	**0.73**	0.73	0.84	0.84	1.14
	(0.80)	(0.77)	(0.73)	(0.75)	(0.75)	(0.64)	(0.59)	(0.64)	(0.65)	**(0.69)**	**(0.65)**	(0.66)	(0.84)	(0.78)	(1.03)
B.Factors	0.84	0.78	0.74	0.76	0.78	0.70	0.65	0.69	0.70	0.79	0.78	0.74	0.84	0.82	1.18
	(0.86)	(0.74)	**(0.67)**	(0.71)	(0.76)	(0.64)	(0.57)	(0.61)	(0.65)	(0.74)	(0.70)	(0.64)	(0.81)	(0.75)	(1.08)
Mean	0.74	0.74	**0.72**	**0.75**	**0.75**	0.68	0.65	0.68	0.70	0.77	0.77	0.74	0.78	0.77	1.06
	(0.75)	(0.72)	**(0.67)**	**(0.68)**	**(0.72)**	(0.61)	**(0.56)**	(0.61)	(0.64)	(0.70)	(0.68)	(0.66)	(0.79)	**(0.70)**	(0.96)
Median	0.74	0.75	0.73	0.76	**0.75**	0.67	0.65	0.68	0.71	0.78	0.76	0.72	**0.77**	**0.76**	1.08
	(0.74)	(0.73)	**(0.67)**	(0.71)	(0.73)	(0.61)	(0.57)	(0.61)	(0.64)	(0.71)	(0.68)	(0.63)	(0.78)	**(0.70)**	(0.99)

Source: Own elaboration from research data (2020). Note: The table shows the root mean squared errors (RMSE) and mean absolute errors (MAE) in parenthesis for the forecasts, relative to the Random Walk (RW). The values in bold indicate the method with lowest values of RMSE and MAE for each horizon. Cells in gray/blue indicate that the method is included in the 50% MCS constructed based on the T_{max} statistic using the squared/absolute errors.

Table 11 – Forecasting errors for the CPI (U.S) from 2013 to 2018 - large sample

Consumer price index (U.S.) 2013-2018 - large sample															
	Forecast horizon														
RMSE/(MAE)	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m
RW	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)	(1.00)
AR	0.87	0.79	0.81	0.86	0.86	0.78	0.72	0.73	0.75	0.84	0.87	0.78	0.90	0.97	1.43
	(0.92)	(0.81)	(0.82)	(0.85)	(0.89)	(0.75)	(0.66)	(0.67)	(0.72)	(0.81)	(0.78)	(0.68)	(0.94)	(0.95)	(1.38)
Ridge	0.96	0.91	0.83	0.96	0.98	0.85	0.87	0.92	0.95	1.08	1.03	0.89	0.85	0.82	1.21
	(1.00)	(0.89)	(0.84)	(0.96)	(1.03)	(0.87)	(0.83)	(0.89)	(0.96)	(1.11)	(0.98)	(0.81)	(0.87)	(0.84)	(1.28)
LASSO	0.78	0.76	0.75	0.81	0.79	0.71	0.65	0.67	0.69	0.77	0.73	0.74	0.79	0.80	1.08
	(0.80)	(0.76)	(0.70)	(0.78)	(0.79)	(0.67)	(0.59)	(0.61)	(0.65)	(0.71)	(0.66)	(0.66)	(0.80)	(0.75)	(1.02)
ENet	0.79	0.76	0.76	0.80	0.83	0.70	0.65	0.67	0.70	0.78	0.75	0.74	0.81	0.82	1.12
	(0.81)	(0.77)	(0.72)	(0.77)	(0.80)	(0.66)	(0.59)	(0.61)	(0.65)	(0.73)	(0.68)	(0.64)	(0.83)	(0.77)	(1.04)
adaLASSO	0.80	**0.72**	0.75	0.79	0.80	0.72	0.68	0.69	0.71	0.76	0.72	0.71	0.77	0.76	1.00
	(0.80)	(0.73)	(0.71)	(0.75)	(0.78)	(0.68)	(0.62)	(0.62)	(0.67)	(0.73)	(0.66)	(0.67)	(0.75)	(0.69)	(0.91)
adaENet	0.79	0.73	0.75	0.79	0.80	0.73	0.67	0.69	0.68	0.75	0.72	0.70	0.78	0.78	1.01
	(0.79)	(0.73)	(0.71)	(0.75)	(0.79)	(0.70)	(0.61)	(0.62)	(0.64)	(0.72)	(0.65)	(0.66)	(0.77)	(0.72)	(0.94)
WLadaLASSO	**0.74**	0.74	**0.72**	0.78	0.78	0.72	0.64	0.69	0.68	0.76	**0.71**	0.70	0.76	0.75	0.96
	(0.73)	(0.74)	(0.69)	(0.73)	(0.76)	(0.68)	(0.59)	(0.62)	(0.64)	(0.73)	(0.66)	(0.66)	(0.75)	(0.68)	(0.90)
WLadaENet	**0.74**	0.74	0.75	**0.75**	0.79	0.73	0.64	0.69	**0.67**	0.74	**0.71**	**0.68**	0.78	0.75	0.96
	(0.74)	(0.76)	(0.72)	**(0.71)**	(0.76)	(0.69)	**(0.56)**	**(0.60)**	(0.63)	(0.71)	**(0.64)**	(0.63)	(0.78)	(0.68)	(0.90)
CSR	0.80	0.73	0.74	0.77	0.77	0.69	0.66	0.67	**0.67**	**0.73**	0.73	0.70	0.77	0.75	1.00
	(0.84)	(0.74)	(0.70)	(0.76)	(0.78)	(0.67)	(0.62)	(0.64)	(0.64)	**(0.70)**	(0.68)	(0.63)	(0.78)	(0.70)	(0.94)
L_2Boost	0.78	0.73	**0.72**	0.79	**0.74**	**0.67**	**0.62**	**0.64**	**0.67**	0.75	0.74	0.72	**0.73**	**0.70**	**0.93**
	(0.77)	**(0.72)**	**(0.68)**	(0.75)	**(0.72)**	(0.64)	**(0.56)**	(0.61)	(0.65)	(0.73)	(0.69)	(0.65)	**(0.72)**	**(0.64)**	**(0.87)**
RF	0.78	0.73	0.73	0.78	0.77	0.69	0.64	0.67	0.69	0.76	0.75	0.71	0.78	0.77	1.02
	(0.81)	(0.73)	(0.69)	(0.73)	(0.76)	(0.64)	**(0.56)**	**(0.60)**	(0.63)	**(0.70)**	(0.68)	**(0.62)**	(0.78)	(0.70)	(0.91)
B.Trees	0.81	0.75	0.76	0.78	0.77	0.68	0.64	0.69	0.70	0.78	0.78	0.73	0.84	0.81	1.13
	(0.86)	(0.73)	(0.72)	(0.74)	(0.76)	**(0.63)**	(0.58)	**(0.60)**	**(0.61)**	(0.71)	(0.69)	(0.63)	(0.85)	(0.76)	(1.05)
Factors	0.83	0.80	0.80	0.82	0.77	0.73	0.71	0.70	0.71	0.76	0.75	0.73	0.86	0.90	1.23
	(0.86)	(0.81)	(0.78)	(0.81)	(0.78)	(0.70)	(0.67)	(0.65)	(0.67)	(0.71)	(0.68)	(0.65)	(0.87)	(0.85)	(1.21)
B.Factors	0.81	0.76	0.76	0.81	0.80	0.73	0.68	0.70	0.71	0.78	0.77	0.73	0.82	0.85	1.19
	(0.85)	(0.77)	(0.76)	(0.81)	(0.82)	(0.70)	(0.61)	(0.64)	(0.67)	(0.72)	(0.68)	(0.64)	(0.83)	(0.82)	(1.15)
Mean	0.77	0.73	0.73	0.77	0.76	0.70	0.65	0.66	0.68	0.75	0.73	**0.68**	0.77	0.75	0.95
	(0.78)	(0.73)	(0.70)	(0.73)	(0.76)	(0.67)	(0.59)	(0.62)	(0.65)	(0.71)	(0.67)	**(0.62)**	(0.77)	(0.69)	**(0.87)**
Median	0.78	0.73	0.74	0.78	0.77	0.70	0.65	0.67	0.68	0.75	0.72	0.70	0.77	0.76	1.00
	(0.79)	(0.73)	(0.70)	(0.75)	(0.77)	(0.67)	(0.59)	(0.61)	(0.64)	**(0.70)**	(0.65)	**(0.62)**	(0.77)	(0.71)	(0.93)

Source: Own elaboration from research data (2020). Note: The table shows the root mean squared errors (RMSE) and mean absolute errors (MAE) in parenthesis for the forecasts, relative to the Random Walk (RW). The values in bold indicate the method with lowest values of RMSE and MAE for each horizon. Cells in gray/blue indicate that the method is included in the 50% MCS constructed based on the T_{max} statistic using the squared/absolute errors.

Table 12 – Forecasting results: Superior predictive ability test (CPI - U.S., 2013-2018, small sample)

Panel (a): Squared errors

	Hansen's test – Forecasting horizon															Quaedvlieg's test			
Benchmark	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m	Unif.(RW)	Avg.(RW)	Unif.(AR)	Avg.(AR)
RW	0.019	0.028	0.027	0.126	0.248	0.004	0.001	0.001	0.001	0.008	0.090	0.096	0.115	0.099	0.861	-	-	0.970	0.996
AR	0.079	0.556	0.320	0.702	0.617	0.473	0.661	0.594	0.772	0.559	0.155	0.343	0.781	0.701	0.343	0.000	0.003	-	-
Ridge	0.240	0.292	0.423	0.669	0.567	0.909	0.955	0.738	0.557	0.010	0.020	0.027	0.749	0.644	0.510	0.000	0.004	0.922	0.946
LASSO	0.762	0.494	0.063	0.004	0.042	0.482	0.146	0.123	0.146	0.280	0.016	0.004	0.716	0.336	0.225	0.675	0.079	0.925	0.999
ENet	0.275	0.579	0.684	0.731	0.628	0.589	0.645	0.775	0.860	0.318	0.433	0.985	0.715	0.674	0.425	0.000	0.004	0.586	0.019
adaLASSO	0.609	0.127	0.418	0.623	0.676	0.447	0.506	0.633	0.662	0.175	0.440	0.375	0.974	0.822	0.500	0.000	0.005	0.810	0.352
adaENet	0.953	0.150	0.689	0.728	0.624	0.545	0.637	0.776	0.724	0.212	0.430	0.986	0.921	0.878	0.481	0.000	0.001	0.679	0.025
WLadaLASSO	0.404	0.085	0.422	0.502	0.663	0.437	0.510	0.543	0.484	0.132	0.300	0.342	0.939	0.734	0.397	0.000	0.003	0.836	0.878
WLadaENet	0.932	0.250	0.614	0.566	0.632	0.542	0.631	0.779	0.552	0.139	0.410	0.728	0.988	0.818	0.531	0.000	0.008	0.788	0.053
CSR	0.130	0.279	0.330	0.312	0.269	0.028	0.128	0.160	0.167	0.126	0.018	0.025	0.318	0.173	0.041	0.000	0.007	0.932	0.993
L_2Boost	0.074	0.199	0.006	0.150	0.123	0.053	0.314	0.121	0.032	0.022	0.029	0.037	0.157	0.236	0.229	0.004	0.009	0.949	0.994
RF	0.307	0.977	0.671	0.473	0.502	0.303	0.789	0.843	0.946	0.407	0.236	0.414	0.916	0.857	0.597	0.000	0.005	0.409	0.135
B.Trees	0.179	0.610	0.458	0.831	0.977	0.612	0.752	0.868	0.550	0.132	0.231	0.194	0.734	0.942	0.399	0.000	0.007	0.754	0.193
Factors	0.271	0.155	0.355	0.454	0.486	0.231	0.399	0.344	0.780	0.963	0.894	0.476	0.571	0.451	0.257	0.000	0.006	0.785	0.916
B.Factors	0.109	0.187	0.488	0.728	0.617	0.258	0.548	0.618	0.839	0.170	0.168	0.390	0.503	0.532	0.215	0.000	0.002	0.795	0.914
Mean	0.711	0.348	0.999	0.999	0.844	0.558	0.656	0.782	0.805	0.367	0.184	0.298	1.000	0.975	0.522	0.000	0.004	0.504	0.025
Median	0.702	0.286	0.823	0.757	0.682	0.606	0.598	0.792	0.690	0.234	0.303	0.958	0.999	0.997	0.432	0.000	0.004	0.661	0.015

Panel (b): Absolute errors

	Hansen's test – Forecasting horizon															Quaedvlieg's test			
Benchmark	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m	Unif.(RW)	Avg.(RW)	Unif.(AR)	Avg.(AR)
RW	0.018	0.008	0.003	0.026	0.062	0.000	0.000	0.000	0.000	0.002	0.002	0.007	0.104	0.041	0.641	-	-	0.969	0.997
AR	0.133	0.631	0.637	0.350	0.623	0.736	0.944	0.723	0.803	0.950	0.146	0.247	0.899	0.750	0.580	0.000	0.005	-	-
Ridge	0.359	0.468	0.391	0.246	0.617	0.853	0.886	0.354	0.093	0.066	0.010	0.018	0.554	0.483	0.576	0.000	0.005	0.940	0.954
LASSO	0.783	0.534	0.075	0.002	0.018	0.730	0.231	0.117	0.128	0.767	0.004	0.003	0.645	0.274	0.259	0.490	0.014	0.945	0.997
ENet	0.381	0.604	0.764	0.352	0.805	0.814	0.947	0.679	0.732	0.703	0.564	0.957	0.982	0.894	0.821	0.000	0.001	0.561	0.036
adaLASSO	0.568	0.121	0.774	0.295	0.701	0.686	0.779	0.501	0.638	0.451	0.560	0.622	0.887	0.738	0.908	0.000	0.000	0.790	0.341
adaENet	0.978	0.226	0.774	0.333	0.805	0.783	0.920	0.669	0.712	0.538	0.548	0.955	0.954	0.858	0.851	0.000	0.003	0.762	0.044
WLadaLASSO	0.325	0.104	0.445	0.187	0.695	0.687	0.772	0.499	0.518	0.438	0.505	0.480	0.780	0.678	0.841	0.000	0.003	0.846	0.900
WLadaENet	0.934	0.269	0.631	0.234	0.801	0.784	0.921	0.665	0.602	0.435	0.556	0.882	0.956	0.855	0.918	0.000	0.002	0.682	0.091
CSR	0.126	0.106	0.150	0.222	0.229	0.026	0.202	0.207	0.049	0.112	0.013	0.015	0.398	0.180	0.117	0.000	0.011	0.904	0.992
L_2Boost	0.044	0.085	0.000	0.030	0.143	0.124	0.416	0.109	0.009	0.012	0.001	0.001	0.098	0.124	0.299	0.015	0.013	0.943	0.991
RF	0.368	0.980	0.865	0.397	0.618	0.386	0.955	0.932	0.902	0.565	0.356	0.377	0.922	0.878	0.915	0.000	0.003	0.549	0.228
B.Trees	0.228	0.557	0.529	0.274	0.934	0.703	0.920	0.774	0.256	0.403	0.189	0.100	0.749	0.941	0.775	0.000	0.002	0.739	0.741
Factors	0.424	0.165	0.352	0.118	0.408	0.340	0.597	0.356	0.585	0.871	0.971	0.257	0.580	0.319	0.580	0.000	0.001	0.809	0.929
B.Factors	0.210	0.392	0.862	0.394	0.313	0.324	0.776	0.700	0.463	0.243	0.218	0.461	0.729	0.475	0.441	0.000	0.003	0.666	0.899
Mean	0.897	0.468	0.955	0.990	0.899	0.854	0.985	0.808	0.799	0.910	0.473	0.226	0.968	0.991	0.989	0.000	0.005	0.173	0.034
Median	0.961	0.358	0.851	0.361	0.904	0.840	0.927	0.714	0.706	0.680	0.425	0.919	0.991	0.988	0.895	0.000	0.001	0.539	0.025

Source: Own elaboration from research data (2020). Note: The table reports the p-values of the test for Superior Predictive Ability of Hansen (2005) (left) using each method as benchmark for each forecasting horizon, including the accumulated horizons. The p-values of the uniform and average multi-horizon SPA test of Quaedvlieg (2019) are also reported (right) using RW ans AR as benchmarks. The Panel (a) presents the p-values for the test using the squared errors and the Panel (b) using the absolute errors. The null hypothesis of the single-horizon test is that the bench of the is that the bench mark is not inferior to any alternative method. The null hypothesis of the multi-horizon test is that the benchmark is superior to the alternative method. The gray cells indicate that the null hypothesis is rejected at the 0.05 significance level.

Table 13 – Forecasting results: Superior predictive ability test (CPI - U.S., 2013-2018, large sample)

Panel (a): Squared errors

	Hansen‘s test – Forecasting horizon															Quaedvlieg‘s test			
Benchmark	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m	Unif.(RW)	Avg.(RW)	Unif.(AR)	Avg.(AR)
RW	0.012	0.017	0.008	0.059	0.062	0.005	0.003	0.003	0.001	0.013	0.004	0.004	0.046	0.020	0.723	-	-	0.930	0.992
AR	0.042	0.105	0.089	0.222	0.224	0.072	0.199	0.286	0.296	0.046	0.087	0.232	0.080	0.020	0.055	0.000	0.008	-	-
Ridge	0.005	0.049	0.044	0.069	0.027	0.004	0.020	0.024	0.017	0.002	0.004	0.068	0.253	0.320	0.245	0.292	0.045	0.879	0.969
LASSO	0.290	0.414	0.509	0.379	0.425	0.160	0.167	0.271	0.528	0.240	0.642	0.376	0.496	0.261	0.298	0.000	0.003	0.000	0.005
ENet	0.185	0.325	0.341	0.429	0.336	0.313	0.235	0.394	0.376	0.193	0.311	0.373	0.370	0.272	0.271	0.000	0.003	0.000	0.007
adaLASSO	0.226	0.971	0.466	0.499	0.450	0.055	0.151	0.122	0.212	0.480	0.757	0.536	0.542	0.334	0.646	0.000	0.001	0.000	0.011
adaENet	0.221	0.889	0.437	0.492	0.433	0.043	0.163	0.138	0.837	0.640	0.910	0.712	0.450	0.278	0.418	0.000	0.004	0.000	0.010
WLadaLASSO	0.928	0.745	0.876	0.395	0.388	0.078	0.518	0.115	0.779	0.493	0.888	0.674	0.735	0.551	0.760	0.000	0.000	0.000	0.009
WLadaENet	0.806	0.649	0.457	0.931	0.529	0.053	0.523	0.179	0.988	0.786	0.945	0.890	0.556	0.512	0.759	0.000	0.003	0.000	0.005
CSR	0.134	0.918	0.486	0.558	0.608	0.451	0.272	0.474	0.907	0.990	0.610	0.746	0.466	0.446	0.722	0.000	0.003	0.000	0.009
L_2Boost	0.341	0.861	0.844	0.328	0.943	0.820	0.948	0.938	0.751	0.550	0.453	0.471	0.964	0.985	0.972	0.000	0.000	0.000	0.012
RF	0.322	0.801	0.756	0.503	0.511	0.439	0.640	0.329	0.640	0.461	0.343	0.555	0.437	0.371	0.392	0.000	0.002	0.000	0.008
B.Trees	0.143	0.429	0.345	0.538	0.442	0.634	0.404	0.171	0.376	0.201	0.144	0.456	0.317	0.316	0.256	0.000	0.001	0.000	0.005
Factors	0.083	0.119	0.110	0.122	0.428	0.081	0.045	0.081	0.187	0.414	0.378	0.439	0.181	0.051	0.099	0.000	0.002	0.136	0.009
B.Factors	0.122	0.370	0.342	0.102	0.157	0.023	0.053	0.079	0.188	0.245	0.143	0.420	0.184	0.100	0.153	0.000	0.002	0.000	0.004
Mean	0.331	0.903	0.834	0.539	0.651	0.192	0.230	0.414	0.882	0.820	0.595	0.999	0.582	0.414	0.613	0.000	0.002	0.000	0.004
Median	0.372	0.963	0.587	0.388	0.355	0.163	0.299	0.268	0.828	0.721	0.872	0.831	0.491	0.308	0.442	0.000	0.004	0.000	0.005

Panel (b): Absolute errors

	Hansen’s test – Forecasting horizon															Quaedvlieg‘s test			
Benchmark	1	2	3	4	5	6	7	8	9	10	11	12	3m	6m	12m	Unif.(RW)	Avg.(RW)	Unif.(AR)	Avg.(AR)
RW	0.002	0.007	0.002	0.011	0.024	0.000	0.000	0.000	0.000	0.004	0.000	0.000	0.024	0.004	0.534	-	-	0.941	0.994
AR	0.022	0.142	0.035	0.134	0.080	0.064	0.154	0.300	0.038	0.077	0.110	0.354	0.025	0.013	0.045	0.000	0.005	-	-
Ridge	0.006	0.060	0.029	0.013	0.004	0.006	0.006	0.024	0.004	0.002	0.002	0.040	0.161	0.119	0.112	0.395	0.037	0.848	0.966
LASSO	0.181	0.434	0.516	0.087	0.146	0.203	0.340	0.846	0.390	0.782	0.722	0.474	0.295	0.225	0.346	0.000	0.001	0.000	0.007
ENet	0.079	0.493	0.449	0.262	0.316	0.455	0.328	0.864	0.316	0.590	0.362	0.594	0.270	0.188	0.320	0.000	0.000	0.000	0.002
adaLASSO	0.170	0.863	0.505	0.429	0.180	0.107	0.100	0.538	0.160	0.585	0.634	0.294	0.674	0.542	0.798	0.000	0.001	0.004	0.014
adaENet	0.207	0.819	0.418	0.442	0.132	0.065	0.135	0.671	0.516	0.736	0.823	0.384	0.572	0.331	0.677	0.000	0.004	0.002	0.008
WLadaLASSO	0.920	0.742	0.712	0.781	0.452	0.150	0.249	0.522	0.496	0.586	0.679	0.355	0.765	0.696	0.894	0.000	0.002	0.004	0.007
WLadaENet	0.700	0.540	0.328	0.945	0.406	0.146	0.718	0.846	0.655	0.849	0.991	0.722	0.481	0.654	0.874	0.000	0.001	0.000	0.012
CSR	0.025	0.877	0.621	0.345	0.192	0.276	0.061	0.398	0.485	0.882	0.375	0.846	0.399	0.462	0.731	0.000	0.002	0.000	0.010
L_2Boost	0.422	0.918	0.893	0.531	0.982	0.692	0.744	0.687	0.351	0.474	0.331	0.437	0.957	0.937	0.811	0.000	0.002	0.008	0.018
RF	0.145	0.812	0.807	0.641	0.515	0.705	0.860	0.920	0.747	0.903	0.389	0.944	0.512	0.494	0.637	0.000	0.005	0.000	0.007
B.Trees	0.119	0.755	0.349	0.529	0.500	0.819	0.497	0.833	0.927	0.705	0.275	0.812	0.221	0.225	0.316	0.000	0.006	0.000	0.002
Factors	0.083	0.179	0.091	0.048	0.277	0.130	0.127	0.178	0.181	0.726	0.352	0.614	0.082	0.034	0.088	0.000	0.003	0.102	0.012
B.Factors	0.038	0.386	0.192	0.036	0.017	0.033	0.076	0.309	0.161	0.588	0.319	0.683	0.186	0.082	0.141	0.000	0.000	0.000	0.006
Mean	0.167	0.962	0.674	0.851	0.522	0.298	0.224	0.797	0.281	0.891	0.601	0.998	0.576	0.547	0.974	0.000	0.003	0.000	0.009
Median	0.129	0.944	0.653	0.450	0.297	0.251	0.299	0.939	0.521	0.996	0.911	0.995	0.527	0.387	0.747	0.000	0.000	0.000	0.009

Source: Own elaboration from research data (2020). Note: The table reports the p-values of the test for Superior Predictive Ability of Hansen (2005) (left) using each method as benchmark for each forecasting horizon, including the accumulated horizons. The p-values of the uniform and average multi-horizon SPA test of Quaedvlieg (2019) are also reported (right) using RW ans AR as benchmarks. The Panel (a) presents the p-values for the test using the squared errors and the Panel (b) using the absolute errors. The null hypothesis of the single-horizon test is that the bench of the is that the bench mark is not inferior to any alternative method. The null hypothesis of the multi-horizon test is that the benchmark is superior to the alternative method. The gray cells indicate that the null hypothesis is rejected at the 0.05 significance level.

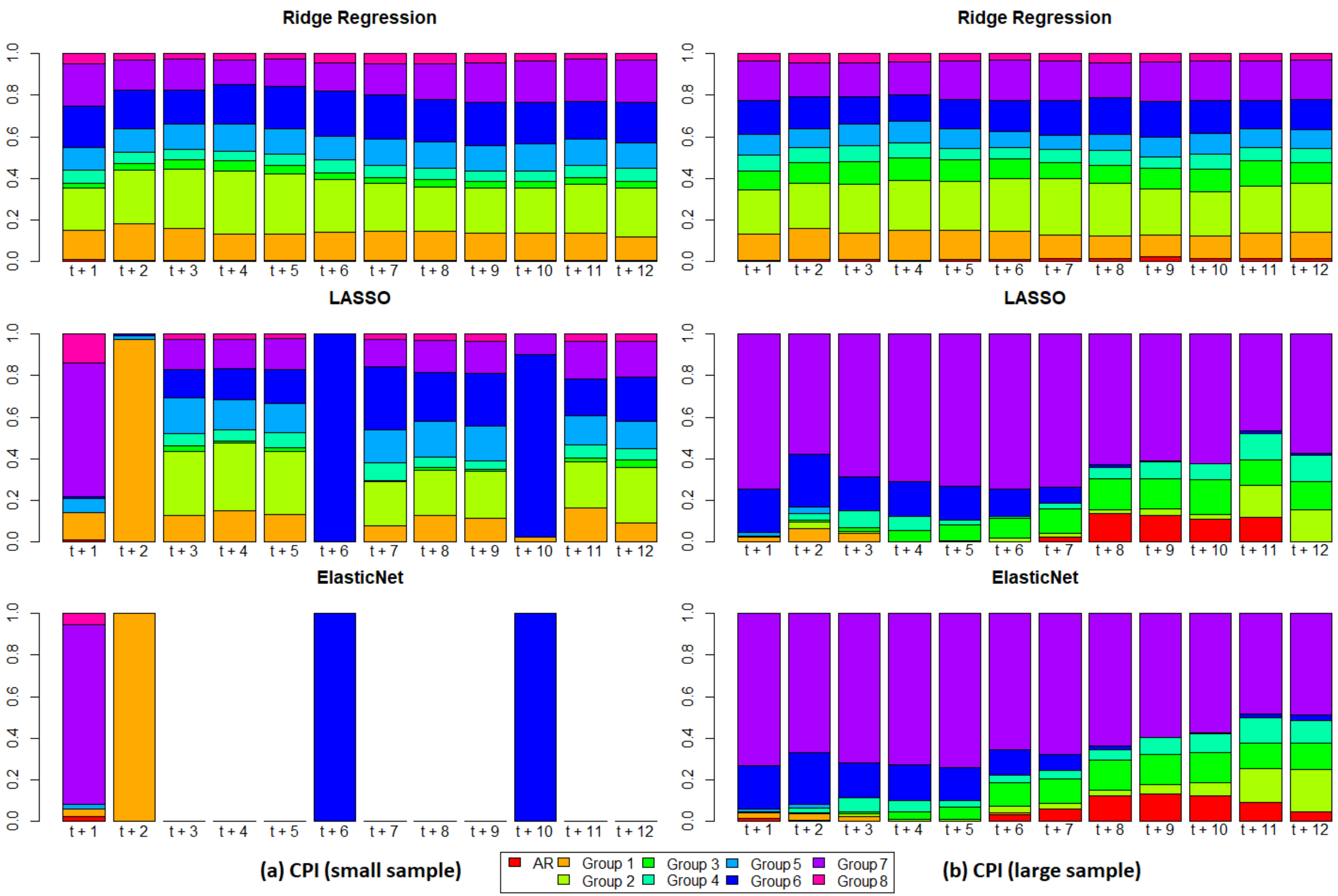


Figure 7 – Variable importance of methods Ridge, LASSO and ENet for U.S. CPI using the small (a) and the large samples (b)

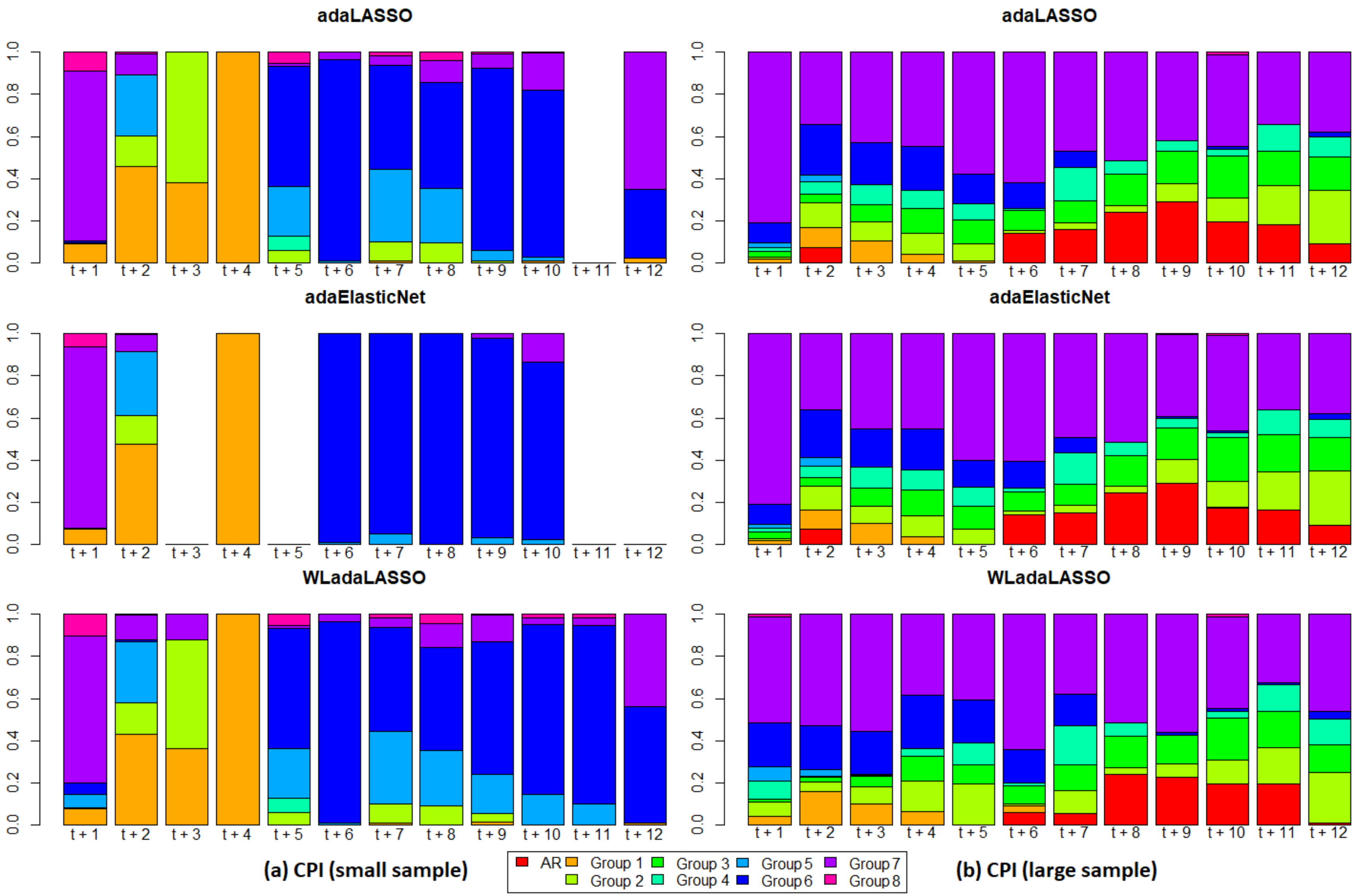


Figure 8 – Variable importance of methods adaLASSO, adaENet and WLadaLASSO for U.S. CPI using the small (a) and the large samples (b)

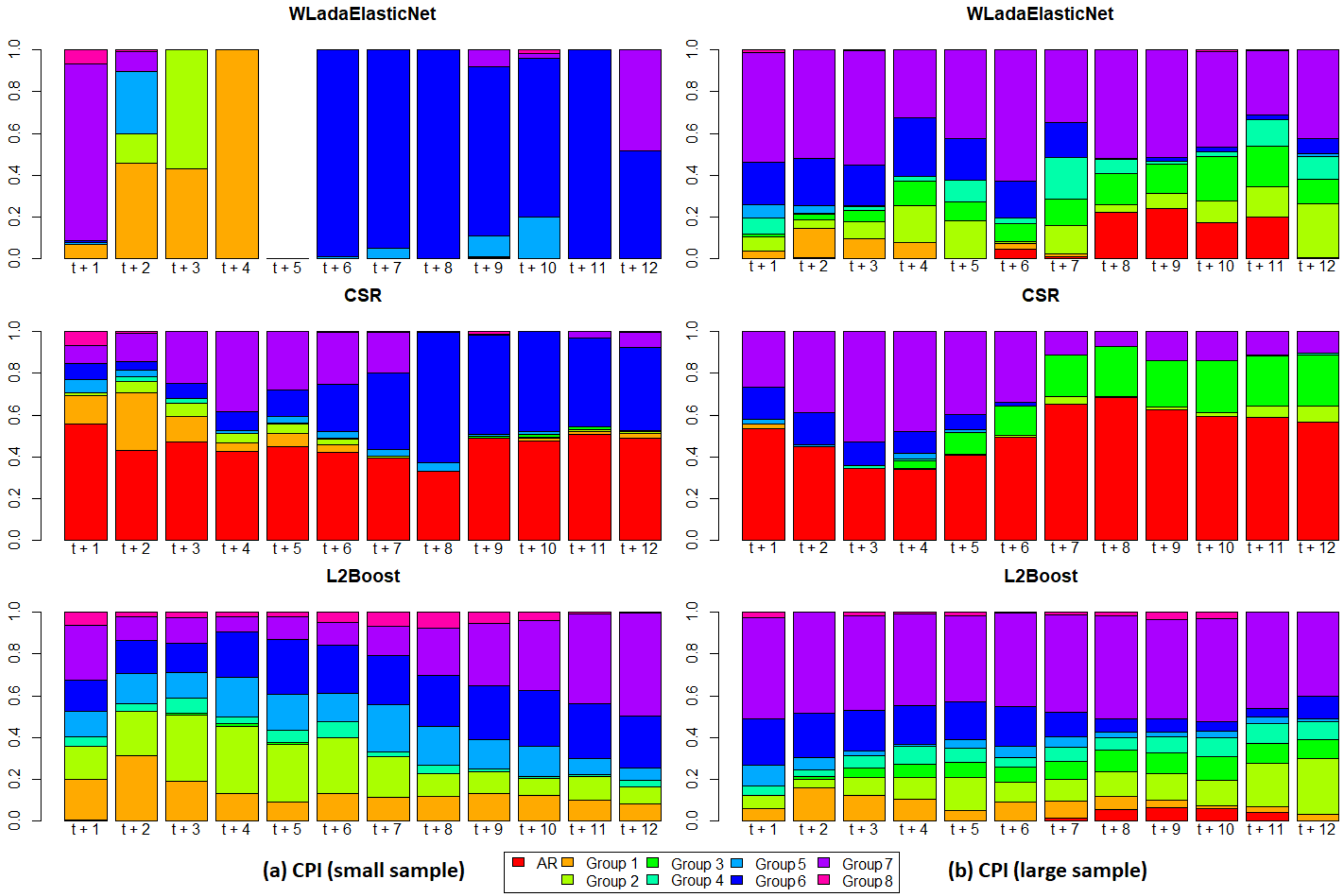


Figure 9 – Variable importance of methods WLadaENet, CSR and L_2Boost for U.S. CPI using the small (a) and the large samples (b)

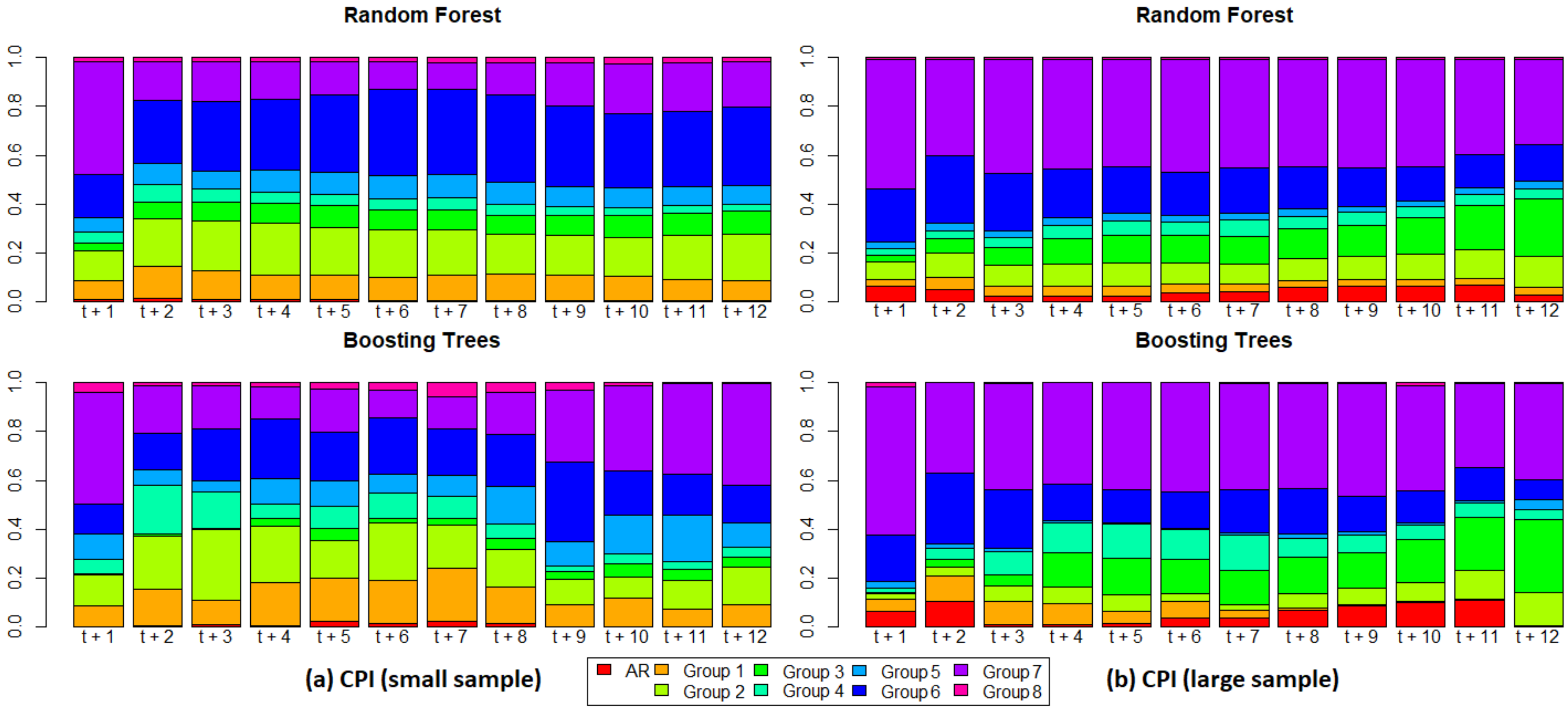


Figure 10 – Variable importance of methods RF and B.Trees for U.S. CPI using the small (a) and the large samples (b)

2.6 CONCLUDING REMARKS

We study a variety of high-dimensional statistical learning methods to perform time series forecasting. In order to evaluate the forecasting performance of these methods we carry out several numerical exercises, including Monte Carlo simulations and empirical data analyses. Through our Monte Carlo implementation, we simulate three different data-generating process, where the first is a sparse model, the second is a dense model, and the third model presents nonlinearities. The methods WLadaLASSO and WLadaENet, which is proposed in the paper, have the best performance in terms of variable selection and forecast in most cases, even when nonlinearities are present. Although these two methods have a good performance in our simulations, they do not perform well to forecast Brazilian inflation and core inflation. In turn, Ridge Regression presents the most accurate forecasts in most horizons both for inflation and core inflation forecasts. For the forecasts of U.S., Ridge Regression has a moderate predictive performance for small sample size (number of variables is larger than the samples size), whereas its performance decreases when the large sample (sample size is larger than the number of variables) is used. Unlike Ridge Regression, L_2Boost improves its performance when the sample size increases, especially when we consider the accumulated forecasting horizons. WLadaENet, our proposed novel method, has a good performance to forecast U.S. inflation when the large sample size is used.

3 CONSIDERAÇÕES FINAIS

No artigo que integra esta dissertação apresentamos diversos métodos de aprendizado estatístico, os quais propõe-se a evitar alguns dos problemas enfrentados pelos métodos tradicionais na presença de dados de alta dimensão, obtendo um bom desempenho em termos de previsão. Utilizamos estes métodos para realizar previsão de séries temporais, avaliando seu desempenho através de diversos exercícios numéricos, incluindo simulação Monte Carlo e análise de dados empíricos. Simulamos três processos geradores de dados, o primeiro deles sendo um modelo linear com estrutura esparsa, o segundo um modelo linear com estrutura densa e o terceiro apresentando não linearidades. Os métodos WLadaLASSO e WLadaENet tiveram um bom desempenho em seleção de variável para o modelo linear esparso e um bom desempenho em previsão na maioria dos casos, mesmo para o modelo com não linearidades.

Apesar destes métodos apresentarem um bom desempenho nas simulações, não tiveram um desempenho tão bom para a previsão da inflação brasileira e do núcleo de inflação. Por sua vez, o método Ridge apresentou um bom desempenho na maioria dos horizontes de previsão, nestes casos. Já para a previsão da inflação norte americana, o método Ridge apresentou um desempenho apenas moderado, utilizando-se conjuntos de dados com tamanhos similares. Neste caso diversos métodos tiveram desempenho semelhante, como ENet, adaLASSO, adaENet, WLadaLASSO, WLadaENet, Random Forest e B.Trees, onde o método Random Forest é o método que apresenta os menores erros em mais horizontes de previsão.

Ao utilizar-se um conjunto de dados cujo tamanho da amostra é consideravelmente maior (tamanho da amostra maior que o número de variáveis), para a inflação norte americana, o desempenho do método Ridge caiu drasticamente, enquanto o método L_2Boost, passou a apresentar os menores erros de previsão. Esta melhora de desempenho é ainda mais perceptível quando se consideram os horizontes de previsão de inflação acumulada. As previsões utilizando o método WLadaENet, proposto no artigo, também tiveram um bom desempenho para a inflação norte americana no segundo caso, levemente superior ao do método WLadaLASSO em alguns horizontes de previsão. Desta forma não temos um método que é universalmente o melhor, mas métodos que funcionam melhor em determinadas situações, onde um método que faz boas previsões para uma determinada estrutura de dados pode não fazer boas previsões para uma estrutura de dados diferente.

REFERÊNCIAS

BAI, Jushan; NG, Serena. Boosting Diffusion Indices. **Journal of Applied Econometrics**, Wiley, v. 24, n. 4, p. 607–629, 2009.

BAI, Jushan; NG, Serena. Determining the Number of Factors in Approximate Factor Models. **Econometrica**, v. 70, n. 1, p. 191–221, 2002.

BREIMAN, Leo. Random Forests. **Machine Learning**, v. 45, n. 1, p. 5–32, Oct. 2001.

BREIMAN, Leo. Bagging Predictors. **Machine Learning**, v. 24, n. 2, p. 123–140, Aug. 1996.

BÜHLMANN, Peter. Boosting for High-Dimensional Linear Models. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 34, n. 2, p. 559–583, 2006.

BÜHLMANN, Peter; YU, Bin. Boosting With the L2 Loss. **Journal of the American Statistical Association**, Taylor & Francis, v. 98, n. 462, p. 324–339, 2003.

ELLIOTT, Graham; GARGANO, Antonio; TIMMERMANN, Allan. Complete subset regressions. **Journal of Econometrics**, v. 177, n. 2, p. 357–373, 2013. Dynamic Econometric Modeling and Forecasting.

ELLIOTT, Graham; GARGANO, Antonio; TIMMERMANN, Allan. Complete subset regressions with large-dimensional sets of predictors. **Journal of Economic Dynamics and Control**, v. 54, p. 86–110, 2015.

FAN, Jianqing. Features of Big Data and sparsest solution in high confidence set. In: LIN, Xihong *et al.* (Eds.). **Past, Present and Future of Statistical Science**. 1. ed. New York: Chapman & Hall, Mar. 2014. chap. 34, p. 507–523.

FRIEDMAN, Jerome H. Greedy Function Approximation: A Gradient Boosting Machine. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 29, n. 5, p. 1189–1232, 2001.

FRIEDMAN, Jerome H. Stochastic gradient boosting. **Computational Statistics & Data Analysis**, v. 38, n. 4, p. 367–378, 2002.

FRIEDMAN, Jerome; HASTIE, Trevor; TIBSHIRANI, Rob. Regularization Paths for Generalized Linear Models via Coordinate Descent. **Journal of Statistical Software, Articles**, v. 33, n. 1, p. 1–22, 2010.

GARCIA, Márcio G. P.; MEDEIROS, Marcelo C.; VASCONCELOS, Gabriel F. R. Real-time inflation forecasting with high-dimensional models: The case of Brazil. **International Journal of Forecasting**, v. 33, n. 3, p. 679–693, 2017.

GU, Shihao; KELLY, Bryan; XIU, Dacheng. Empirical Asset Pricing via Machine Learning. Available at SSRN: https://ssrn.com/abstract=3159577 or http://dx.doi.org/10.2139/ssrn.3159577. Chicago, Sept. 2019.

HANSEN, Peter Reinhard. A Test for Superior Predictive Ability. **Journal of Business & Economic Statistics**, [American Statistical Association, Taylor & Francis, Ltd.], v. 23, n. 4, p. 365–380, 2005.

HANSEN, Peter Reinhard; LUNDE, Asger; NASON, James M. The Model Confidence Set. **Econometrica**, v. 79, n. 2, p. 453–497, 2011.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome H. **The elements of statistical learning: data mining, inference and prediction**. 2. ed. New York: Springer, 2009.

HOERL, Arthur E.; KENNARD, Robert W. Ridge Regression: Biased Estimation for Nonorthogonal Problems. **Technometrics**, Taylor & Francis, v. 12, n. 1, p. 55–67, 1970.

JAMES, Gareth; WITTEN, Daniela; HASTIE, Trevor; TIBSHIRANI, Robert. **An Introduction to Statistical Learning – with Applications in R**. 1. ed. New York: Springer, 2013.

KONZEN, Evandro; ZIEGELMANN, Flavio A. LASSO-Type Penalties for Covariate Selection and Forecasting in Time Series. **Journal of Forecasting**, v. 35, n. 7, p. 592–612, 2016.

MCCRACKEN, Michael W.; NG, Serena. FRED-MD: A Monthly Database for Macroeconomic Research. **Journal of Business & Economic Statistics**, Taylor Francis, v. 34, n. 4, p. 574–589, 2016.

MEDEIROS, Marcelo C.; MENDES, Eduardo F. 1-regularization of high-dimensional time-series models with non-Gaussian and heteroskedastic errors. **Journal of Econometrics**, v. 191, n. 1, p. 255–271, 2016.

MEDEIROS, Marcelo C.; VASCONCELOS, Gabriel F. R. Forecasting macroeconomic variables in data-rich environments. **Economics Letters**, v. 138, p. 50–52, 2016.

MEDEIROS, Marcelo C.; VASCONCELOS, Gabriel F. R.; FREITAS, Eduardo. Forecasting Brazilian Inflation with High-Dimensional Models. **Brazilian Review of Econometrics**, v. 36, n. 2, p. 223–254, 2016.

MEDEIROS, Marcelo C.; VASCONCELOS, Gabriel F. R.; VEIGA, Álvaro; ZILBERMAN, Eduardo. Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods. **Journal of Business & Economic Statistics**, Taylor & Francis, p. 1–22, 2019.

MULLAINATHAN, Sendhil; SPIESS, Jann. Machine Learning: An Applied Econometric Approach. **Journal of Economic Perspectives**, v. 31, n. 2, p. 87–106, May 2017.

QUAEDVLIEG, Rogier. Multi-Horizon Forecast Comparison. **Journal of Business & Economic Statistics**, Taylor Francis, v. 0, n. 0, p. 1–14, 2019.

TIBSHIRANI, Robert. Regression Shrinkage and Selection via the Lasso. **Journal of the Royal Statistical Society. Series B (Methodological)**, v. 58, n. 1, p. 267–288, 1996.

ZOU, Hui. The Adaptive Lasso and Its Oracle Properties. **Journal of the American Statistical Association**, Taylor & Francis, v. 101, n. 476, p. 1418–1429, 2006.

ZOU, Hui; HASTIE, Trevor. Regularization and Variable Selection via the Elastic Net. v. 67, n. 2, p. 301–320, 2005.

ZOU, Hui; ZHANG, Hao Helen. On the Adaptive Elastic-Net with a Diverging Number of Parameters. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 37, n. 4, p. 1733–1751, 2009.

APPENDIX A – LIST OF VARIABLES (BRAZIL)

For Tables A.1 - A.8 the column *tcode* denotes the following data transformation for a series x: (1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $\log(x_t)$; (5) $\Delta \log(x_t)$; (6) $\Delta^2 \log(x_t)$; (7) $\Delta(x_t/x_t - 1)$.

Table A.1 – Group 1: Prices

	id	tcode	Group 1: Prices description
1	1	1	CPI: IPCA
2	2	1	GPI: IGP-10
3	3	1	GPI: IGP-DI
4	4	1	GPI: IGPM
5	5	1	CPI: INPC
6	6	1	CPI: IPCA15
7	7	1	CPI: IPCA-DP (Core - Double Weight)
8	8	1	CPI: IPCA-EX0 (Core - Exclusion)
9	9	1	CPI: IPCA-EX1 (Core - Exclusion)
10	10	1	CPI: IPCA-MS (Core - Trimmed Means Smoothed)
11	11	1	CPI: IPCA - Administred Prices
12	12	1	CPI: IPCA - Housing
13	13	1	CPI: IPCA - Personal Expenditures
14	14	1	CPI: IPCA - Gasoline
15	15	1	CPI: IPCA - Health Insurance
16	16	1	CPI: IPCA - Other
17	17	1	CPI: IPCA - Water and Sewage Rates
18	18	1	CPI: IPCA - Durable Goods

Source: Own elaboration from research data (2020).
Note: All prices in percentage change.

Table A.2 – Group 2: Employment and Wages

	id	tcode	Group 2: Employment and Wages description
1	19	2	Unemployment Rate (RMSP)
2	20	2	Unemployment Rate - Open (RMSP)
3	21	2	Unemployment Rate - Hidden Precarious - (RMSP)
4	22	3	Employment Personnel - Industry
5	23	3	Hours Worked - Industry
6	24	2	Minimum Wage - Real
7	25	2	Minimum Wage - PPP

Source: Own elaboration from research data (2020).

Table A.3 – Group 3: International Transactions and Government Debt

	id	tcode	description
		Group 3: International Transactions and Government Debt	
1	26	2	Current Account (US$)
2	27	2	Balance on Goods and Services (US$)
3	28	2	Balance on Goods (US$)
4	29	2	Secondary Income (US$)
5	30	1	Capital Account (US$)
6	31	2	Financial Account (US$)
7	32	2	External Debt - States and Cities
8	33	3	Fiscal Debt - Public Sector
9	34	2	Internal Debt - States and Cities
10	35	2	Internal Debt - Federal Government and Central Bank
11	36	3	Total Net Debt - States and Cities
12	37	3	Total Net Debt - Federal Government and Central Bank

Source: Own elaboration from research data (2020).

Table A.4 – Group 4: Economic Activity and Production

	id	tcode	description
			Group 4: Economic Activity and Production
1	38	2	Apparent Consumption - Alcohol
2	39	2	Apparent Consumption - Oil and Derivatives
3	40	5	Apparent Consumption - Capital Goods
4	41	5	Apparent Consumption - Intermediate Goods
5	42	5	Apparent Consumption - Consumer Goods - All
6	43	5	Apparent Consumption - Durable Consumer Goods
7	44	5	Apparent Consumption - Semi-durable and Non-durable Goods
8	45	5	Apparent Consumption - Industry General
9	46	5	Apparent Consumption - Manufacturing Industry
10	47	5	Real Income - Industry
11	48	2	Electricity - Consumption
12	49	2	Default - Number of Queries
13	50	5	Economic Condition Index
14	51	2	Bad Checks
15	52	2	Capacity Utilization
16	53	5	Industrial Production - Intermediate Goods
17	54	5	Industrial Production - General
18	55	5	Industrial Production - Capital Goods
19	56	5	Industrial Production - Metallurgy
20	57	5	Industrial Production - Cellulose and Paper
21	58	2	Industrial Production - Oil
22	59	2	Industrial Production - Steel
23	60	5	Industrial Production - Motor Vehicles
24	61	2	Slaughter - Cattle
25	62	2	Slaughter - Poultry
26	63	2	Slaughter - Pigs
27	64	5	Construction Index

Source: Own elaboration from research data (2020).

Table A.5 – Group 5: Taxes and Government Income

	Group 5: Taxes and Government Income		
	id	tcode	description
1	65	2	Cofins - total - Gross Income
2	66	3	Social Contribution over Net Profits
3	67	2	PIS/Pasep - total - Gross Income
4	68	3	Gross Collection of Federal Revenues
5	69	2	Taxes on Goods Circulation
6	70	2	Import Taxes
7	71	5	Income Taxes - Individual
8	72	5	Income Taxes - Legal Entity
9	73	6	Total Gross Income Taxes
10	74	1	Taxes on Rural Properties
11	75	5	Tax on Motor Vehicles
12	76	5	Financial Taxes
13	77	5	Industry Taxes
14	78	2	Taxes - Other
15	79	2	Taxes - Fees

Source: Own elaboration from research data (2020).

Table A.6 – Group 6: Exchange Rates and Finance

	Group 6: Exchange Rates and Finance		
	id	tcode	description
1	80	2	Exchange Rate - Dollar Commercial
2	81	2	PPP conversion factor (private consumption)
3	82	5	BOVESPA Stock Index
4	83	5	Dow Jones Stock Index
5	84	2	Return on Savings
6	85	1	Returns on Gold
7	86	2	Interest Rate CDI/Over
8	87	2	Interest Rate TJLP
9	88	2	Interest Rate Over/Selic

Source: Own elaboration from research data (2020).

Table A.7 – Group 7: Money

	Group 7: Money		
	id	tcode	description
1	89	2	M0
2	90	2	M1
3	91	3	M2
4	92	6	M3
5	93	6	M4

Source: Own elaboration from research data (2020).

Table A.8 – Group 8: Economic Confidence

	Group 8: Economic Confidence		
	id	tcode	description
1	94	5	Index - Consumer Confidence
2	95	5	Index - Economic Expectations

Source: Own elaboration from research data (2020).

APPENDIX B – LIST OF VARIABLES (U.S.)

For Tables B.1 - B.8 the column *tcode* denotes the following data transformation for a series x: (1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $log(x_t)$; (5) $\Delta log(x_t)$; (6) $\Delta^2 log(x_t)$; (7) $\Delta(x_t/x_t - 1)$. The FRED column gives mnemonics in FRED followed by a short description. The comparable series in Global Insight is given in the column GSI.

Table B.1 – Group 1: Output and income

				Group 1: Output and income		
	id	tcode	fred	description	gsi	gsi:description
1	1	5	RPI	Real Personal Income	M_14386177	PI
2	2	5	W875RX1	Real personal income ex transfer receipts	M_145256755	PI less transfers
3	6	5	INDPRO	IP Index	M_116460980	IP: total
4	7	5	IPFPNSS	IP: Final Products and Nonindustrial Supplies	M_116460981	IP: products
5	8	5	IPFINAL	IP: Final Products (Market Group)	M_116461268	IP: final prod
6	9	5	IPCONGD	IP: Consumer Goods	M_116460982	IP: cons gds
7	10	5	IPDCONGD	IP: Durable Consumer Goods	M_116460983	IP: cons dble
8	11	5	IPNCONGD	IP: Nondurable Consumer Goods	M_116460988	IP: cons nondble
9	12	5	IPBUSEQ	IP: Business Equipment	M_116460995	IP: bus eqpt
10	13	5	IPMAT	IP: Materials	M_116461002	IP: matls
11	14	5	IPDMAT	IP: Durable Materials	M_116461004	IP: dble matls
12	15	5	IPNMAT	IP: Nondurable Materials	M_116461008	IP: nondble matls
13	16	5	IPMANSICS	IP: Manufacturing (SIC)	M_116461013	IP: mfg
14	17	5	IPB51222s	IP: Residential Utilities	M_116461276	IP: res util
15	18	5	IPFUELS	IP: Fuels	M_116461275	IP: fuels
16	20	2	CUMFNS	Capacity Utilization: Manufacturing	M_116461602	Cap util

Source: McCracken and Ng (2016).

Table B.2 – Group 2: Labor market

	Group 2: Labor market					
	id	tcode	fred	description	gsi	gsi:description
1	21*	2	HWI	Help-Wanted Index for United States		Help wanted indx
2	22*	2	HWIURATIO	Ratio of Help Wanted/No. Unemployed	M_110156531	Help wanted/unemp
3	23	5	CLF16OV	Civilian Labor Force	M_110156467	Emp CPS total
4	24	5	CE16OV	Civilian Employment	M_110156498	Emp CPS nonag
5	25	2	UNRATE	Civilian Unemployment Rate	M_110156541	U: all
6	26	2	UEMPMEAN	Average Duration of Unemployment (Weeks)	M_110156528	U: mean duration
7	27	5	UEMPLT5	Civilians Unemployed - Less Than 5 Weeks	M_110156527	U < 5 wks
8	28	5	UEMP5TO14	Civilians Unemployed for 5-14 Weeks	M_110156523	U 5-14 wks
9	29	5	UEMP15OV	Civilians Unemployed - 15 Weeks & Over	M_110156524	U 15+ wks
10	30	5	UEMP15T26	Civilians Unemployed for 15-26 Weeks	M_110156525	U 15-26 wks
11	31	5	UEMP27OV	Civilians Unemployed for 27 Weeks and Over	M_110156526	U 27+ wks
12	32	5	CLAIMSx	Initial Claims	M_15186204	UI claims
13	33	5	PAYEMS	All Employees: Total nonfarm	M_123109146	Emp: total
14	34	5	USGOOD	All Employees: Goods-Producing Industries	M_123109172	Emp: gds prod
15	35	5	CES1021000001	All Employees: Mining and Logging: Mining	M_123109244	Emp: mining
16	36	5	USCONS	All Employees: Construction	M_123109331	Emp: const
17	37	5	MANEMP	All Employees: Manufacturing	M_123109542	Emp: mfg
18	38	5	DMANEMP	All Employees: Durable goods	M_123109573	Emp: dble gds
19	39	5	NDMANEMP	All Employees: Nondurable goods	M_123110741	Emp: nondbles
20	40	5	SRVPRD	All Employees: Service-Providing Industries	M_123109193	Emp: services
21	41	5	USTPU	All Employees: Trade, Transportation & Utilities	M_123111543	Emp: TTU
22	42	5	USWTRADE	All Employees: Wholesale Trade	M_123111563	Emp: wholesale
23	43	5	USTRADE	All Employees: Retail Trade	M_123111867	Emp: retail
24	44	5	USFIRE	All Employees: Financial Activities	M_123112777	Emp: FIRE
25	45	5	USGOVT	All Employees: Government	M_123114411	Emp: Govt
26	46	1	CES0600000007	Avg Weekly Hours : Goods-Producing	M_140687274	Avg hrs
27	47	2	AWOTMAN	Avg Weekly Overtime Hours : Manufacturing	M_123109554	Overtime: mfg
28	48	1	AWHMAN	Avg Weekly Hours : Manufacturing	M_14386098	Avg hrs: mfg
29	127	6	CES0600000008	Avg Hourly Earnings : Goods-Producing	M_123109182	AHE: goods
30	128	6	CES2000000008	Avg Hourly Earnings : Construction	M_123109341	AHE: const
31	129	6	CES3000000008	Avg Hourly Earnings : Manufacturing	M_123109552	AHE: mfg

Source: McCracken and Ng (2016).

Table B.3 – Group 3: Housing

	Group 3: Housing					
	id	tcode	fred	description	gsi	gsi:description
1	50	4	HOUST	Housing Starts: Total New Privately Owned	M_110155536	Starts: nonfarm
2	51	4	HOUSTNE	Housing Starts, Northeast	M_110155538	Starts: NE
3	52	4	HOUSTMW	Housing Starts, Midwest	M_110155537	Starts: MW
4	53	4	HOUSTS	Housing Starts, South	M_110155543	Starts: South
5	54	4	HOUSTW	Housing Starts, West	M_110155544	Starts: West
6	55	4	PERMIT	New Private Housing Permits (SAAR)	M_110155532	BP: total
7	56	4	PERMITNE	New Private Housing Permits, Northeast (SAAR)	M_110155531	BP: NE
8	57	4	PERMITMW	New Private Housing Permits, Midwest (SAAR)	M_110155530	BP: MW
9	58	4	PERMITS	New Private Housing Permits, South (SAAR)	M_110155533	BP: South
10	59	4	PERMITW	New Private Housing Permits, West (SAAR)	M_110155534	BP: West

Source: McCracken and Ng (2016).

Table B.4 – Group 4: Consumption, orders, and inventories

	id	tcode	fred	description	gsi	gsi:description
				Group 4: Consumption, orders, and inventories		
1	3	5	DPCERA3M086SBEA	Real personal consumption expenditures	M_123008274	Real Consumption
2	4	5	CMRMTSPLx	Real Manu. and Trade Industries Sales	M_110156998	M&T sales
3	5	5	RETAILx	Retail and Food Services Sales	M_130439509	Retail sales
4	65	5	AMDMNOx	New Orders for Durable Goods	M_14386110	Orders: dble gds
5	66	5	ANDENOx	New Orders for Nondefense Capital Goods	M_178554409	Orders: cap gds
6	68	5	BUSINVx	Total Business Inventories	M_15192014	M&T invent
7	69	2	ISRATIOx	Total Business: Inventories to Sales Ratio	M_15191529	M&T invent/sales

Source: McCracken and Ng (2016).

Table B.5 – Group 5: Money and credit

	id	tcode	fred	description	gsi	gsi:description
				Group 5: Money and credit		
1	70	6	M1SL	M1 Money Stock	M_110154984	M1
2	71	6	M2SL	M2 Money Stock	M_110154985	M2
3	72	5	M2REAL	Real M2 Money Stock	M_110154985	M2 (real)
4	73	6	AMBSL	St. Louis Adjusted Monetary Base	M_110154995	MB
5	74	6	TOTRESNS	Total Reserves of Depository Institutions	M_110155011	Reserves tot
6	76	6	BUSLOANS	Commercial and Industrial Loans	BUSLOANS	C&I loan plus
7	77	6	REALLN	Real Estate Loans at All Commercial Banks	BUSLOANS	DC&I loans
8	78	6	NONREVSL	Total Nonrevolving Credit	M_110154564	Cons credit
9	79	2	CONSPI	Nonrevolving consumer credit to Personal Income	M_110154569	Inst cred/PI
10	131	6	MZMSL	MZM Money Stock	N.A.	N.A.
11	132	6	DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding	N.A.	N.A.
12	133	6	DTCTHFNM	Total Consumer Loans and Leases Outstanding	N.A.	N.A.
13	134	6	INVEST	Securities in Bank Credit at All Commercial Banks	N.A.	N.A.

Source: McCracken and Ng (2016).

Table B.6 – Group 6: Interest and exchange rates

	id	tcode	fred	description	gsi	gsi:description
				Group 6: Interest and exchange rates		
1	84	2	FEDFUNDS	Effective Federal Funds Rate	M_110155157	Fed Funds
2	85	2	CP3Mx	3-Month AA Financial Commercial Paper Rate	CPF3M	Comm paper
3	86	2	TB3MS	3-Month Treasury Bill	M_110155165	3 mo T-bill
4	87	2	TB6MS	6-Month Treasury Bill	M_110155166	6 mo T-bill
5	88	2	GS1	1-Year Treasury Rate	M_110155168	1 yr T-bond
6	89	2	GS5	5-Year Treasury Rate	M_110155174	5 yr T-bond
7	90	2	GS10	10-Year Treasury Rate	M_110155169	10 yr T-bond
8	91	2	AAA	Moody‘s Seasoned Aaa Corporate Bond Yield		Aaa bond
9	92	2	BAA	Moody‘s Seasoned Baa Corporate Bond Yield		Baa bond
10	93	1	COMPAPFFx	3-Month Commercial Paper Minus FEDFUNDS		CP-FF spread
11	94	1	TB3SMFFM	3-Month Treasury C Minus FEDFUNDS		3 mo-FF spread
12	95	1	TB6SMFFM	6-Month Treasury C Minus FEDFUNDS		6 mo-FF spread
13	96	1	T1YFFM	1-Year Treasury C Minus FEDFUNDS		1 yr-FF spread
14	97	1	T5YFFM	5-Year Treasury C Minus FEDFUNDS		5 yr-FF spread
15	98	1	T10YFFM	10-Year Treasury C Minus FEDFUNDS		10 yr-FF spread
16	99	1	AAAFFM	Moody‘s Aaa Corporate Bond Minus FEDFUNDS		Aaa-FF spread
17	100	1	BAAFFM	Moody‘s Baa Corporate Bond Minus FEDFUNDS		Baa-FF spread
18	102	5	EXSZUSx	Switzerland / U.S. Foreign Exchange Rate	M_110154768	Ex rate: Switz
19	103	5	EXJPUSx	Japan / U.S. Foreign Exchange Rate	M_110154755	Ex rate: Japan
20	104	5	EXUSUKx	U.S. / U.K. Foreign Exchange Rate	M_110154772	Ex rate: UK
21	105	5	EXCAUSx	Canada / U.S. Foreign Exchange Rate	M_110154744	EX rate: Canada

Source: McCracken and Ng (2016).

Table B.7 – Group 7: Prices

	Group 7: Prices					
	id	tcode	fred	description	gsi	gsi:description
1	106	5	WPSFD49207	PPI: Finished Goods	M110157517	PPI: fin gds
2	107	5	WPSFD49502	PPI: Finished Consumer Goods	M110157508	PPI: cons gds
3	108	5	WPSID61	PPI: Intermediate Materials	M_110157527	PPI: int matls
4	109	5	WPSID62	PPI: Crude Materials	M_110157500	PPI: crude matls
5	110	5	OILPRICEx	Crude Oil, spliced WTI and Cushing	M_110157273	Spot market price
6	111	5	PPICMM	PPI: Metals and metal products	M_110157335	PPI: nonferrous
7	113	5	CPI	CPI : All Items	M_110157323	CPI-U: all
8	114	5	CPIAPPSL	CPI : Apparel	M_110157299	CPI-U: apparel
9	115	5	CPITRNSL	CPI : Transportation	M_110157302	CPI-U: transp
10	116	5	CPIMEDSL	CPI : Medical Care	M_110157304	CPI-U: medical
11	117	5	CUSR0000SAC	CPI : Commodities	M_110157314	CPI-U: comm.
12	118	5	CUSR0000SAD	CPI : Durables	M_110157315	CPI-U: dbles
13	119	5	CUSR0000SAS	CPI : Services	M_110157325	CPI-U: services
14	120	5	CPIULFSL	CPI : All Items Less Food	M_110157328	CPI-U: ex food
15	121	5	CUSR0000SA0L2	CPI : All items less shelter	M_110157329	CPI-U: ex shelter
16	122	5	CUSR0000SA0L5	CPI : All items less medical care	M_110157330	CPI-U: ex med
17	123	5	PCEPI	Personal Cons. Expend.: Chain Index	gmdc	PCE defl
18	124	5	DDURRG3M086SBEA	Personal Cons. Exp: Durable goods	gmdcd	PCE defl: dlbes
19	125	5	DNDGRG3M086SBEA	Personal Cons. Exp: Nondurable goods	gmdcn	PCE defl: nondble
20	126	5	DSERRG3M086SBEA	Personal Cons. Exp: Services	gmdcs	PCE defl: service

Source: McCracken and Ng (2016).

Table B.8 – Group 8: Stock market

	Group 8: Stock market					
	id	tcode	fred	description	gsi	gsi:description
1	80	5	S&P 500	S&P‘s Common Stock Price Index: Composite	M_110155044	S&P 500
2	81	5	S&P: indust	S&P‘s Common Stock Price Index: Industrials	M_110155047	S&P: indust
3	82	2	S&P div yield	S&P‘s Composite Common Stock: Dividend Yield		S&P div yield
4	83	5	S&P PE ratio	S&P‘s Composite Common Stock: Price-Earnings Ratio		S&P PE ratio

Source: McCracken and Ng (2016).