UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
FACULDADE DE CIÊNCIAS ECONÔMICAS
DEPARTAMENTO DE ECONOMIA E RELAÇÕES INTERNACIONAIS

NATHALIA SCHOLLES OREDA

KEY FEATURES FOR IMPROVING MACROECONOMIC FORECASTING: EVIDENCE FROM BRAZIL

Porto Alegre
2024

NATHALIA SCHOLLES OREDA

KEY FEATURES FOR IMPROVING MACROECONOMIC FORECASTING: EVIDENCE FROM BRAZIL

Work submitted to Faculty of Economics at UFRGS as partial requirement for obtaining the Bachelor's Degree in Economics.

Supervisor: Prof. Dr. Hudson da Silva Torrent

Porto Alegre
2024

CIP - Catalogação na Publicação

Oreda, Nathalia Scholles
Key features for improving Macroeconomic Forecasting: Evidence from Brazil / Nathalia Scholles Oreda. -- 2024.
43 f.
Orientador: Hudson da Silva Torrent.

Trabalho de conclusão de curso (Graduação) -- Universidade Federal do Rio Grande do Sul, Faculdade de Ciências Econômicas, Curso de Ciências Econômicas, Porto Alegre, BR-RS, 2024.

1. Data-rich. 2. Aprendizado de máquina. 3. Previsão. 4. Séries Temporais Macroeconômicas. I. Torrent, Hudson da Silva, orient. II. Título.

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

NATHALIA SCHOLLES OREDA

KEY FEATURES FOR IMPROVING MACROECONOMIC FORECASTING: EVIDENCE FROM BRAZIL

Work submitted to Faculty of Economics at UFRGS as partial requirement for obtaining the Bachelor's Degree in Economics.

Approved in: Porto Alegre, 8th of February, 2024.

THESIS DEFENSE COMMITTEE:

Prof. Dr. Hudson da Silva Torrent – Advisor
UFRGS

Prof. Dr. Cristiano Lima Hackmann
UFRGS

Prof. Dr. Fernando Augusto Boeira Sabino da Silva
UFRGS

ACKNOWLEDGEMENTS

I would like to thank my mother for all of the love, support and guidance throughout my life and education journey. Her encouragement and assistance, especially in the past few months, were indispensable for the conclusion of this paper. I am truly grateful for having her as my mother and as my best friend.

A special thanks to the best partner that I could possibly have asked for, Fernando, who helped me so many times with coding questions and encouraged me to improve my programming skills. He gives me love and support every day and I am lucky to have him in my life.

I am also immensely thankful for my advisor, Dr. Hudson Torrent, for all of the assistance and mentorship throughout the development of this paper. Working with him has been an enriching experience. I learned so much with him and I am grateful for all of his suggestions and contributions that helped to improve this paper.

I would also like to express my appreciation to all of my family and friends who have played a role in my academic journey, offering encouragement, assistance and contributions in moments of need.

Lastly, I dedicate this paper to the memory of my beloved father, whose encouragement led me to pursue a degree in Economics. His strength and belief in my abilities have been a guiding force throughout my educational path. Though he is no longer with us, his impact on my life and on this achievement is immeasurable.

ABSTRACT

This study aims to compare the performance of linear and nonlinear models in data-poor and data-rich environments with different regularization and loss functions to understand which characteristics of machine learning are useful for macroeconomic forecasting when using Brazilian macroeconomic variables. For that, we predict three macroeconomic variables that are indicators of the Brazilian economy: the open unemployment rate of the Metropolitan Region of São Paulo, the Brazilian inflation rate using the IPCA index, and the spread with the indicator Emerging Markets Bond Index Plus (EMBI+) for Brazil using 34 linear and nonlinear models that differ in the hyperparameters, the regularizations and the loss function used. We make the predictions considering observations of 139 variables for almost 23 years. The main conclusions show that the best models are the ones with lasso and Elastic Net penalties, and that the data-rich is the best environment, especially when considering shrinkage methods using the elastic-net estimator.

Keywords: Data-rich. Machine Learning. Forecasting. Macroeconomic Time Series.

RESUMO

Este artigo tem como objetivo comparar vários modelos diferentes em diferentes cenários, a fim de compreender quais características do aprendizado de máquina são úteis para previsões macroeconômicas. Este estudo consiste em comparar o desempenho de modelos lineares e não lineares em ambientes pobres e ricos em dados para diferentes janelas e horizontes para previsão de diferentes variáveis macroeconômicas brasileiras. Para isso, prevemos três variáveis macroeconômicas que são indicadores da economia brasileira: a taxa de desemprego aberta da Região Metropolitana de São Paulo, a taxa de inflação brasileira, usando o índice IPCA, e o spread com o indicador Emerging Markets Bond Index Plus (EMBI+) para o Brasil usando 34 modelos lineares e não lineares que diferem nos hiper parâmetros, nas regularizações e na função de perda utilizada. Fazemos as previsões considerando observações de 139 variáveis durante quase 23 anos. As principais conclusões mostram que os melhores modelos são aqueles com penalidade de lasso e elastic-net e o ambiente rico em dados (data-rich) é o melhor, especialmente quando se consideram métodos de shrinkage usando o estimador do elastic-net.

Palavras-chave: Data-rich. Aprendizado de máquina. Previsão. Séries Temporais Macroeconômicas.

CONTENTS

1 INTRODUCTION

Machine learning is a powerful tool to learn from data and it is useful for the problem of modeling and prediction of y given X. Machine learning techniques are increasingly being used due to their benefits and because of the development and availability of powerful and free software, such as Python and R that make it very easy to use them. According to the authors in Mullainathan and Spiess (2017), the success of machine learning at intelligence tasks is mainly because of its ability to discover complex structure that was not specified in advance and to fit well in-sample with complex and very flexible functional forms to the data that also perform well out-of-sample.

This way, machine learning is increasingly being used to forecast macroeconomic variables, and there are already many papers that have successfully used it to predict US macroeconomic variables over the years, for example, Moshiri and Cameron (2000) used artificial neural network modelling to forecast the inflation rate, Sermpinis et al. (2014) used a hybrid genetic algorithm–Support Vector Regression (GA-SVR) model to forecast US inflation and unemployment, and Medeiros et al. (2019) showed that Random Forest outperforms other models when forecasting the US inflation rate. For the Brazilian case, the macroeconomic forecasting literature using machine learning models is still very recent. For the Brazilian inflation prediction, see Araujo and Gaglianone (2023), Garcia, Medeiros, and Vasconcelos (2017) and Medeiros, Vasconcelos, and Freitas (2016). For the prediction of other Brazilian macroeconomic variables, see Lindenmeyer and Torrent (2023).

There are many studies that contribute with evidence to prove that machine learning is useful for macroeconomic forecasting. However, there are not many papers that help to understand what are the main characteristics of machine learning that make a good forecasting model. The authors in Goulet Coulombe et al. (2022) sought to answer this question by conducting a meta-analysis of many machine learning products using US and Canadian data, in order to understand how machine learning is useful for macroeconomic predictability. In their analysis, the authors identified four main features of machine learning, which are described below:

a) the function that forms the prediction;

b) the regularization penalty;

c) the set of hyperparameters;

d) the loss function.

It is important to highlight that such study is necessarily restrictive because there are many possibilities regarding the selection of the regression function, the density of regressors, the correlation between them, the error density, the sample size, the type of polynomial regression, the kernel function, the smoothing and regularization parameters, the type of squared error criterion function, among other factors. Considering this, the authors concluded that nonlinearity

is the true game changer for macroeconomic prediction, that the standard factor model remains the best regularization, that K-fold cross-validation is the best practice and that the squared error loss is preferred to the $\bar{\epsilon}$-insensitive in-sample loss.

With the paper "Key features for improving macroeconomic forecasting - Evidence from Brazil" in section 2, we seek to construct a similar experiment for the Brazilian case by conducting an exercise of an extensive pseudo-out-of-sample forecasting horse race between many models that differ with respect to these features in order to see if we reach different conclusions when analyzing macroeconomic variables for an emerging country. For that, we forecast the open unemployment rate of the Metropolitan Region of São Paulo (UNRATE), the Brazilian inflation rate using the Brazilian Consumer Price Index called *Índice de Preços ao Consumidor Ampliado* (IPCA) and the Brazilian spread using the Emerging Markets Bond Index Plus (EMBI+). We chose to make the predictions both in data-poor and data-rich environments using the model Autoregressive direct (AR) as benchmark, and the models Random Forest, boosting, Support Vector Regression, factor models and models with ridge, lasso, and Elastic Net penalties. We use Ipeadata database to conduct the analysis using observations of 139 variables for almost 23 years and considering five forecasting horizons. We contribute to the literature, by being, as far as we know, the first paper to use these machine learning models to forecast the open unemployment rate of the Metropolitan Region of São Paulo (UNRATE), and the EMBI+. We are also the first paper to analyze the main characteristics of machine learning for an emerging country. The code used in this paper was developed in R programming language and can be found in our digital repository on GitHub [1].

[1] https://github.com/nathaliaoreda/thesis_UFRGS.

2 KEY FEATURES FOR IMPROVING MACROECONOMIC FORECASTING: EVIDENCE FROM BRAZIL

PRINCIPAIS ATRIBUTOS PARA MELHORAR A PREVISIBILIDADE MACROECONÔMICA: EVIDÊNCIAS PARA O BRASIL

ABSTRACT

This study aims to compare the performance of linear and nonlinear models in data-poor and data-rich environments with different regularization and loss functions to understand which characteristics of machine learning are useful for macroeconomic forecasting when using Brazilian macroeconomic variables. For that, we predict three macroeconomic variables that are indicators of the Brazilian economy: the open unemployment rate of the Metropolitan Region of São Paulo, the Brazilian inflation rate using the IPCA index, and the spread with the indicator Emerging Markets Bond Index Plus (EMBI+) for Brazil using 34 linear and nonlinear models that differ in the hyperparameters, the regularizations and the loss function used. We make the predictions considering observations of 139 variables for almost 23 years. The main conclusions show that the best models are the ones with lasso and Elastic Net penalties, and that the data-rich is the best environment, especially when considering shrinkage methods using the Elastic Net estimator.

Keywords: Data-rich. Machine Learning. Forecasting. Macroeconomic Time Series.

RESUMO

Este artigo tem como objetivo comparar vários modelos diferentes em diferentes cenários, a fim de compreender quais características do aprendizado de máquina são úteis para previsões macroeconômicas. Este estudo consiste em comparar o desempenho de modelos lineares e não lineares em ambientes pobres e ricos em dados para diferentes janelas e horizontes para previsão de diferentes variáveis macroeconômicas brasileiras. Para isso, prevemos três variáveis macroeconômicas que são indicadores da economia brasileira: a taxa de desemprego aberta da Região Metropolitana de São Paulo, a taxa de inflação brasileira, usando o índice IPCA, e o spread com o indicador Emerging Markets Bond Index Plus (EMBI+) para o Brasil usando 34 modelos lineares e não lineares que diferem nos hiper parâmetros, nas regularizações e na função de perda utilizada. Fazemos as previsões considerando observações de 139 variáveis durante quase 23 anos. As principais conclusões mostram que os melhores modelos são aqueles com penalidade de lasso e elastic-net e o ambiente rico em dados (data-rich) é o melhor, especialmente quando se consideram métodos de shrinkage usando o estimador do elastic-net.

Palavras-chave: Data-rich. Aprendizado de máquina. Previsão. Séries Temporais Macroeconômicas.

2.1 INTRODUCTION

Machine learning is a powerful tool to learn from data and it is increasingly being used to forecast macroeconomic variables due to its ability to uncover complex patterns and fit flexible forms to the data without overfitting, as described by the authors in Mullainathan and Spiess (2017). There are already many papers that have successfully used machine learning to predict US macroeconomic variables over the years, for example, Moshiri and Cameron (2000), Sermpinis et al. (2014), Medeiros et al. (2019). For the Brazilian case, the macroeconomic forecasting literature using machine learning models is still very recent. For the Brazilian inflation prediction, see Araujo and Gaglianone (2023), Garcia, Medeiros, and Vasconcelos (2017) and Medeiros, Vasconcelos, and Freitas (2016). For the prediction of other Brazilian macroeconomic variables, see Lindenmeyer and Torrent (2023).

There are many studies that contribute with evidence to prove that machine learning is useful for macroeconomic forecasting. However, there are not many papers that help to understand what are the main characteristics of machine learning that make a good forecasting model. The authors in Goulet Coulombe et al. (2022) sought to answer this question by conducting a meta-analysis of many machine learning products using US and Canadian data, in order to understand how machine learning is useful for macroeconomic predictability. In their analysis, the authors identified four main features of machine learning, which are the function that forms the prediction, the regularization penalty, the set of hyperparameters and the loss function used. It is important to highlight that such study is necessarily restrictive because there are many possibilities regarding the selection of the regression function, the density of regressors, the correlation between them, the error density, the sample size, the type of polynomial regression, the kernel function, the smoothing and regularization parameters, the type of squared error criterion function, among other factors. Considering this, the authors concluded that nonlinearity is the true game changer for macroeconomic prediction, that the standard factor model remains the best regularization, that K-fold cross-validation is the best practice and that the squared error loss is preferred to the $\bar{\epsilon}$-insensitive in-sample loss.

We seek to construct a similar experiment for the Brazilian case by conducting an exercise of an extensive pseudo-out-of-sample forecasting horse race between many models that differ with respect to these features in order to see if we reach different conclusions when analyzing macroeconomic variables for an emerging country. For that, we forecast the open unemployment rate of the Metropolitan Region of São Paulo (UNRATE), the Brazilian inflation rate using the Brazilian Consumer Price Index called *Índice de Preços ao Consumidor Ampliado* (IPCA) and the Brazilian spread using the Emerging Markets Bond Index Plus (EMBI+). We chose to make the predictions both in data-poor and data-rich environments using 34 linear and nonlinear models that differ in the hyperparameters, the regularizations and the loss function used. We use Ipeadata database to conduct the analysis using observations of 139 variables for almost 23 years and considering five forecasting horizons. As far as we know, this is the first paper to

use these machine learning models to forecast the open unemployment rate of the Metropolitan Region of São Paulo (UNRATE), and the EMBI+. It is also the first paper to analyze the main characteristics of machine learning for an emerging country.

The rest of the paper is organized as follows. Section 2.2 describes the methodology used for the forecasting exercise. Section 2.3 describes the forecasting setup. Section 2.4 details the main results of this study. Lastly, section 2.5 concludes.

2.2 METHODOLOGY

This section presents the methodology followed in this paper, with details of the features studied, the considered data environments, as well as the models chosen.

Consider an observed value y that we want to analyze. We assume that some predictors X are related to the variable y in some form. We can write this scenario as following:

$$y = f(X) + \epsilon, \tag{1}$$

where y is the response variable, X is the set of p predictors available for the prediction $(X_1, X_2, ..., X_p)$ and ϵ is a random error term. When y is unknown, it is possible to predict it h steps ahead using

$$\hat{y}_{t+h} = \hat{f}(X_t) + \epsilon, \tag{2}$$

where $\hat{f}$ is the estimate for f, and $\hat{y}_{t+h}$ is the resulting prediction for y, h steps ahead. With the assumption that $\hat{f}$ and X_t are fixed, it is possible to show that

$$E(y_{t+h} - \hat{y}_{t+h})^2 = [f(X) - \hat{f}(X)]^2 + Var(\epsilon), \tag{3}$$

where $E(y_{t+h} - \hat{y}_{t+h})^2$ represents the expected value of the squared difference between the predicted and actual value of y and $Var(\epsilon)$ represents the variance associated with the error term ϵ (James et al., 2013). The goal of machine learning methods is to help estimating $f(X)$ as precisely as possible by minimizing the reducible error ($[f(X) - \hat{f}(X)]^2$).

As described by Goulet Coulombe et al. (2022), the general prediction setup by Hastie, Tibshirani, and Friedman (2009) can be described as

$$\min_{g \in \mathcal{G}} \{\hat{L}(y_{t+h}, g(X_t)) + pen(g; \tau)\}, \quad t = 1, \ldots, T. \tag{4}$$

The four main features for this setup are the following:

a) $\mathcal{G}$ is the space of possible functions g that combine the data to form the prediction. The goal is to discover how nonlinearity can contribute to minimize the reducible error. For that, we compare the performance of both linear and nonlinear models.

b) $pen()$ is the regularization penalty that controls large values for the function g and prevents overfitting. We discuss the regularization penalties used in this paper in subsection 2.2.1.

c) τ is the set of hyperparameters including those in the penalty and the approximator g. For the τ optimization, we consider both in-sample and out-of-sample methods.

- Akaike Information Criterion (AIC) introduced by AKAIKE (1973) is an in-sample measure that tries to select the model that is closest to the high-dimensional truth.
- Bayesian Information Criterion (BIC) from Schwarz (1978) is also an in-sample measure that considers the truth as contained in the set of models when n tends to infinity. Especially for large sample sizes, BIC penalizes more heavily than AIC.
- K-fold cross validation (CV) is an out-of sample measure defined by

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i, \tag{5}$$

where k are the number of folds in which the set of observations is divided and MSE_i is the mean squared error of the model on the ith fold. The method is fit on $k-1$ folds and the remaining folder is used to compute the mean squared error (MSE). The mean is then computed from the k estimates of the test error results (James et al., 2013). For our study, we consider $k = 5$, since it is widely used in the forecasting literature and it was also used in Goulet Coulombe et al. (2022), making it easier to compare the results.

d) $\hat{L}$ is the loss function that penalizes errors in prediction. The most common loss function is the squared error loss that can be defined as following, according to Hastie, Tibshirani, and Friedman (2009):

$$L(Y, f(X)) = (Y - f(X))^2. \tag{6}$$

The squared error loss is the most popular loss function since the expected squared error is minimized by the conditional mean. However for this study following Goulet Coulombe et al. (2022), we will also consider the $\bar{\epsilon}$-insensitive loss function for the Support Vector Regression (SVR) models, defined in subsection 2.2.2.

2.2.1 Environments

Many papers have studied the performance of macroeconomic forecasting using data-poor and data-rich environments. Machine learning is often associated with big data, since using a larger dataset with many predictors can help to identify patterns in the data, perform variable selection and improve the prediction. However, using a large number of predictors can sometimes lead to worse results by overfitting, i.e., considering random noise as real signal in the data, causing larger variance and deteriorating the model's performance. This is known in the literature as the bias-variance tradeoff. In this paper we follow Goulet Coulombe et al. (2022) to study different regularization schemes by comparing several models in the following data environments: *Data-Poor* H_t^- using the autoregressive direct (AR) model, *Data-Rich* H_t^+ using

the autoregression augmented with diffusion indices (ARDI) from Stock and Watson (2002) and *Elastic Net (EN)* using special cases of the Elastic Net (EN) problem as alternative shrinkage methods.

a) The *Data-Poor* H_t^- model is the *autoregressive direct* (AR) model, which is specified as:

$$y_{t+h} = c + \rho(L)y_t + e_{t+h}, \quad t = 1, \ldots, T, \tag{7}$$

where c is a constant, $h \geq 1$ is the forecasting horizon, p_y is the hyperparameter in this model, which is the order used for the lag polynomial $\rho(L)$ and e_{t+h} is the error term. Here the predictors in X are the variable that is being forecasted and its p_y lags.

b) The *Data-Rich* H_t^+ model is the autoregression augmented with diffusion indices (ARDI) from Stock and Watson (2002):

$$y_{t+h} = c + \rho(L)y_t + \beta(L)F_t + e_{t+h}, \quad t = 1, \ldots, T \tag{8}$$

$$X_t = \Lambda F_t + u_t, \tag{9}$$

where c is a constant, F_t are K *consecutive static* factors, $\rho(L)$ and $\beta(L)$ are lag polynomials of orders p_y and p_f respectively and e_{t+h} and u_t are error terms. The estimate of F_t is obtained by principal component analysis (PCA) and the result is used as data in an Autoregressive Distributed Lag model (ARDL) model. p_y, p_f and K are the hyperparameters of the model. In this model, the predictors X are the variable that is being forecasted with its p_y lags and the K *consecutive static* factors with its p_f lags. It is important to highlight that p_y and p_f always have the same value in our study.

c) The *Elastic Net (EN)* is used to generate alternative shrinkage methods to provide variation for the data-rich environment. The Elastic Net estimator by Zou and Hastie (2005) considers the lasso and ridge estimators as two extreme cases. The Elastic Net selects variables like the lasso and shrinks together the coefficients of correlated predictors like ridge (Hastie; Tibshirani; Friedman, 2009). The EN problem is defined as following, as can be seen in Goulet Coulombe et al. (2022):

$$\min_{\beta} \sum_{t=1}^{T} \left(y_{t+h} - Z_t\beta\right)^2 + \lambda \sum_{k=1}^{K} \left(\alpha|\beta_k| + (1-\alpha)\beta_k^2\right), \tag{10}$$

where $Z_t = B(H_t)$ is a transformation of the original predictive set X_t and λ is a complexity parameter that controls the amount of shrinkage. We use different B operators and α to generate variations across shrinkage schemes. Also, by setting α to either 1 or 0 we generate lasso and Ridge Regression respectively.

Following Goulet Coulombe et al. (2022), we consider the following variation of B for a fixed λ:

- B_1: taking all observables H_t^+: For B_1, we consider the whole untransformed high-dimensional dataset (H_t^+). In this variation, the predictors in Z_t are all the untransformed 139 variables of the dataset and its p_y lags. The hyperparameters in this model are the λ which is determined by CV and the lag order for the whole untransformed high-dimensional dataset that we set to $p_y = 12$, based on Goulet Coulombe et al. (2022), that also considered the value 12 as the maximum lag order for the models.

- B_2: taking all principal components of X_t: Here $B_2()$ rotates the whole dataset X_t into factors estimated by principal components, which then constitute Z_t to be used in (10). The hyperparameters in this model are the λ which is determined by CV and the lag order for the factors estimated by principal components (p_f) that we also set to 12.

- B_3: taking all principal components of H_t^+: Finally, $B_3()$ is similar to B_2 and rotates the whole dataset X_t into factors estimated by principal components. The result is then combined to the variable that is being predicted, which then constitute Z_t to be used in (10). The hyperparameters in this model are the λ, which is determined by CV, the lag order of the variable to be estimated (p_y) and the lag order for the factors estimated by principal components (p_f). In this case, we also choose to set $p_y = 12$ and $p_f = 12$.

We also compare the Elastic Net variations to the linear boosting model for B_1 (b1_cv_boost), B_2 (b2_cv_boost) and B_3 (b3_cv_boost). The linear boosting is explained in subsection 2.2.2. Here the lag orders considered are the same as the Elastic Net variations and we also use k-fold cross-validation to select the number of iterations *mstop* for our model.

2.2.2 Models

In this subsection, we detail all linear and nonlinear models used in this paper. A summary of the models used can be seen in table 1.

Table 1 – Models

	Model	Model Name	Function	Regularization	Loss Fuction
Data-poor models					
1	ar_bic/ ar_aic/ ar_cv	Autoregressive direct (AR)	Linear		Squared Error
2	shrink_poor_cv_ridge	Ridge Regression- AR	Linear	ridge	Squared Error
3	shrink_poor_cv_lasso	Lasso- AR	Linear	lasso	Squared Error
4	shrink_poor_cv_en	Elastic Net- AR	Linear	en	Squared Error
5	rf_poor_cv	Random Forest- AR	Nonlinear		Squared Error
6	bols_poor_cv	Linear Boosting- AR	Linear		Squared Error
7	bbs_poor_cv	Boosting with Splines- AR	Nonlinear		Squared Error
8	svr_linear_poor_cv	Support Vector Regression-Linear kernel- AR	Linear		$\bar{\epsilon}$-insensitive
9	svr_rbf_poor_cv	Support Vector Regression-RBF kernel- AR	Nonlinear		$\bar{\epsilon}$-insensitive
Data-rich models					
10	ardi_bic/ ardi_aic/ ardi_cv	Autoregression augmented with diffusion indices (ARDI)	Linear	PCA	Squared Error
11	shrink_rich_cv_ridge	Ridge Regression- ARDI	Linear	ridge- PCA	Squared Error
12	shrink_rich_cv_lasso	Lasso- ARDI	Linear	lasso-PCA	Squared Error
13	shrink_rich_cv_en	Elastic Net- ARDI	Linear	en-PCA	Squared Error
14	rf_rich_cv	Random Forest- ARDI	Nonlinear	PCA	Squared Error
15	bols_rich_cv	Linear Boosting- ARDI	Linear	PCA	Squared Error
16	bbs_rich_cv	Boosting with Splines- ARDI	Nonlinear	PCA	Squared Error
17	svr_rich_poor_cv	Support Vector Regression-Linear kernel- ARDI	Linear	PCA	$\bar{\epsilon}$-insensitive
18	svr_rbf_rich_cv	Support Vector Regression-RBF kernel- ARDI	Nonlinear	PCA	$\bar{\epsilon}$-insensitive
19	b1_cv_boost	Linear Boosting-B1	Linear		Squared Error
20	b1_cv_ridge	Ridge Regression-B1	Linear	ridge	Squared Error
21	b1_cv_en	Elastic Net-B1	Linear	en	Squared Error
22	b1_cv_lasso	Lasso-B1	Linear	lasso	Squared Error
23	b2_cv_boost	Linear Boosting-B2	Linear	PCA	Squared Error
24	b2_cv_ridge	Ridge Regression-B2	Linear	ridge-PCA	Squared Error
25	b2_cv_en	Elastic Net-B2	Linear	en-PCA	Squared Error
26	b2_cv_lasso	Lasso-B2	Linear	lasso-PCA	Squared Error
27	b3_cv_boost	Linear Boosting-B3	Linear	PCR	Squared Error
28	b3_cv_ridge	Ridge Regression-B3	Linear	ridge-PCR	Squared Error
29	b3_cv_en	Elastic Net-B3	Linear	en-PCR	Squared Error
30	b3_cv_lasso	Lasso-B3	Linear	lasso-PCR	Squared Error

Source: Author.

a) Autoregressive direct (AR) model:

This is the benchmark model used for this paper and it is specified as in (7). We select the hyperparameter p_y, which is the order used for the lag polynomial using BIC, AIC, and CV. The lag order is selected from the subset $p_y = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$, following Goulet Coulombe et al. (2022). For computational purposes, the hyperparameter is estimated every 12 months.

b) Autoregression augmented with diffusion indices (ARDI) model:

The ARDI model is a data-rich model specified as in (8). The number of *consecutive static* factors K is selected from the subset {3, 6, 10}, as in Goulet Coulombe et al. (2022). In addition, the lag orders are also selected from the subset $p_y, p_f = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$. All the hyperparameters of this model are selected using BIC, AIC, and CV and are estimated every 12 months.

c) Shrinkage models (shrink):

The OLS estimator does not take into consideration the trade-off between bias and variance and it is subject to overfitting. It considers all coefficients as having equal weights without doing variable selection or regularization. To deal with this issue of

the OLS estimator and working to improve the prediction accuracy, shrinkage methods can be used. This way, we estimate a penalized version of the AR model (shrink_poor) and ARDI model (shrink_rich) that allows potentially more lagged predictors. For the shrinkage, we consider three different penalties: Ridge Regression, Lasso, and Elastic Net. For all of these methods, the hyperparameter p_y is estimated from the subset $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$ for the shrink_poor models using CV and for the shrink_rich models the number of *consecutive static* factors K is selected from the subset {3, 6, 10}, as in Goulet Coulombe et al. (2022) and the hyperparameters p_y and p_f are selected from the subset $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$ using CV. The hyperparameter λ is also cross validated. For computational purposes, the hyperparameters are estimated every 12 months. To estimate the shrinkage models, we use package *glmnet*.

- Ridge Regression (ridge): The parameters for the models using Ridge Regression are estimated using ridge penalty. In equation (10), the ridge penalty corresponds to $\alpha = 0$, that is, $\lambda \sum_{k=1}^{K} (\beta_k^2)$. The penalty keeps the β coefficients small to reduce the variance. It is important to highlight that the ridge penalty does not help with variable selection since all variables are included in the model even if they have small β coefficients.

- Least Absolute Shrinkage and Selection Operator (lasso): The lasso penalty is similar to Ridge Regression; however it may drive some of the β coefficients to 0, potentially helping with variable selection. In equation (10), the lasso penalty corresponds to $\alpha = 1$, i.e., $\lambda \sum_{k=1}^{K} (|\beta_k|)$. The lasso penalty forces some of the coefficient estimates to be exactly 0 when the tuning parameter λ is sufficiently large.

- Elastic Net (en): As described in subsection (2.2.1), the Elastic Net penalty combines the benefits from both ridge and lasso. The en model is specified as in (10). For the Elastic Net estimation we consider $\alpha = 0.5$, which is the same value used in Medeiros et al. (2019).

d) Random Forest(rf):

A popular way of introducing nonlinearity to the forecast is by using tree-based methods. This is because trees are intuitive and have natural interpretation. However, they are often not competitive in terms of predictive accuracy. The inferior prediction accuracy of a standard tree regression is related to the fact that small changes in the training data lead to large changes in prediction. As described by Hastie, Tibshirani, and Friedman (2009), Random Forest is a method to de-correlate the individual tree predictions and can often perform better. The idea behind Random Forest is to build a number of decision trees on bootstrapped training samples like the bagging method. However, each time the tree is split, instead of considering all predictors, a random selection of m predictors is chosen as split candidates from the full set of p predictors. m is a tuning parameter and we adopt the

default choice for regression that is $m = p/3$. For the depth of the tree B, we also adopt the default value of $B = 500$. Random Forest is in between parametric and non-parametric models because it has some structure, but it is not as strict as the structure of the linear model, for example. Here we don't do cross-validation to select hyperparameters since the Random Forest method already does variable selection internally. As also done in the other models, we consider $p_y = 12$ in rf_poor. In addition, for the rf_rich model, we consider $p_f = 12$ and $K = 10$. To estimate the Random Forest models, we use package *randomForest*.

e) Boosting (bols and bbs):

We also use boosting methods for the forecasting which is a popular and powerful approach. According to Kauppi and Virtanen (2020), the model is gradually "learned" from the available data on a term-by-term basis. The generic boosting estimator can be defined as following:

$$\hat{f}(.) = \hat{f}^{(0)} + v \sum_{m=1}^{M} \hat{g}^{(m)}(.), \tag{11}$$

where M is the stopping criterion of the algorithm, v is a learning rate parameter, $\hat{f}^{(0)}$ is a constant that represents the starting point for the boosting process and $\hat{g}^{m)}(.)$ is the learner. If we consider a specific m, we can define

$$\hat{g}^{(m)}(.) = \underset{\hat{h}(\cdot)}{\operatorname{argmin}} L(f, \hat{h}(.)), \tag{12}$$

where the function $\hat{h}$ is the fitting procedure, and $L(\cdot)$ is the chosen loss function. In this study, we consider the MSE as the loss function. For the fitting procedure we use a linear regression with the linear boosting (bols) and the P-spline methodology with the boosting with splines (bbs) following Schmid and Hothorn (2008), which allows to introduce nonlinearity to the boosting method and therefore more flexibility to the linear structure with the use of a nonparametric base.

The *component-wise* boosting procedure used in this paper starts with the simple average as the $\hat{f}^{(0)}$. The weak learners are then added to the model sequentially. For each iteration m, a single predictor is considered and the baseline learner is fitted for each of the predictor's lags at a time. A regression is then done to the residuals and the fit with the smallest residual sum of squares is selected and used to update the estimation $\hat{f}^{(m)} = \hat{f}^{(m-1)} + v\hat{g}^{(m)}$. This procedure is repeated M times.

The boosting idea is to boost weak learners to a strong learner. The algorithm learns slowly from previous residuals errors and this way can handle complex relationships in the data, thus improving the accuracy of the model. The *component-wise* boosting procedure allows

to optimize each component individually to improve the model's performance and does variable selection by not including weak predictors to the model. This way, the boosting method does not require pre-ordering of the predictors or their lags, as described by Bai and Ng (2009). Considering this, we chose not to use CV to select the optimal p_y, p_f and K hyperparameters. For bols_poor and bbs_poor we consider $p_y = 12$ and for bols_rich and bbs_rich we also consider $p_f = 12$ and $K = 10$. Also, for bbs_poor and bbs_rich, we adopt the degrees of freedom $df = 4$, following the authors in Lindenmeyer and Torrent (2023). The upper bound was fixed in $M = 300$ and we use k-fold cross-validation to select the number of iterations *mstop*, which is the major tuning parameter of boosting, as stated in Hofner et al. (2012). To estimate the boosting models, we use package *mboost*.

f) Support Vector Regression (svr):

Support Vector Regression (svr) is the regression for Support Vector Machines (svm) discussed initially by Vapnik (1996). Support Vector Machines are based on the idea of constructing a hyperplane that almost separates all the training observations correctly according to their class labels and that a test data will be classified according to its location side in this separating hyperplane (James et al., 2013). This is known as a support vector classifier that uses soft margins to avoid overfitting. As stated in James et al. (2013), "The *support vector machine* (SVM) is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using *kernels*." This way, SVM is a nonparametric technique. The Support Vector Regression consists on implementing a linear epsilon-insensitive SVM (ϵ-SVM) regression which is known as L1 loss. As described in Goulet Coulombe et al. (2022), the ϵ-SVR can be defined as

$$\begin{aligned} &\min_{\gamma} \frac{1}{2}\gamma'\gamma + C\left[\sum_{t=1}^{T}(\xi_t + \xi_t^*)\right] \\ s.t. &\begin{cases} y_{t+h} - \gamma'\phi(Z_t) - c \leq \bar{\epsilon} + \xi_t \\ \gamma'\phi(Z_t) + c - y_{t+h} \leq \bar{\epsilon} + \xi_t^* \\ \xi_t, \xi_t^* \geq 0, \end{cases} \end{aligned} \tag{13}$$

where ξ_t, ξ_t^* are slack variables, $\phi()$ is the basis function of the feature space implicitly defined by the kernel, C is the cost of a violation to the margin, $\bar{\epsilon}$ is the insensitive-loss function, γ are the related weights, c is a constant and T is the size of the sample used for estimation. The loss function associated with the $\epsilon-$SVR treats errors that are within $\bar{\epsilon}$ distance of the observed value as zero. The low error points are the ones that have small residuals. This way, the loss function associated with the $\epsilon-$SVR is

$$P_{\bar{\epsilon}}(\epsilon_{t+h|t}) := \begin{cases} 0 & if \quad |e_{t+h}| \leq \bar{\epsilon} \\ |e_{t+h}| - \bar{\epsilon} & otherwise \end{cases} . \tag{14}$$

For $\epsilon-$SVR the penalty increases at a constant rate once errors are sufficiently large, and it is different from the squared error loss, where the penalty increases with the size of the forecasting error.

In this paper, following Goulet Coulombe et al. (2022) we consider both Support Vector Regression with linear kernel (svr_linear) and with nonlinear RBF kernel (svr_rbf). For both svr_linear_poor and svr_rbf_poor, p_y is once again estimated from the subset $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$, and for svr_linear_rich and svr_rbf_rich the hyperparameters p_f and K are also cross-validated with $p_f = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$ and $K = (3, 6, 10)$.

The other hyperparameters in these models are the cost (C) selected from the subset $C = (0.1, 0.5, 1, 2, 5)$ and the epsilon ($\bar{\epsilon}$) selected from the subset $\bar{\epsilon} = (0.1, 0.2, 0.3, 0.4, 0.5)$ to offer variation from the default values. They are selected every 12 months and by using CV. In addition for the svr_rbf model a scale parameter σ needed for the kernel is also a hyperparameter that is cross-validated from the subset $\sigma = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ every 12 months. To estimate the Support Vector Regression models, we use package *e1071*.

2.3 FORECASTING SETUP

In this section we describe the empirical setup for constructing the forecasting exercise. We detail the database used, the forecasting exercise and finally the evaluation methods used to measure the performance of the methods.

2.3.1 Database

The database used in this paper corresponds to monthly data in the period between January 1996 and May 2019, using the data available in Ipeadata, a macroeconomic, financial, and regional database of Brazil maintained by Ipea. We decided to start the database in January 1996, because the country's economic scenario was very unstable before the implementation of the Real Plan, in February 1994, which could compromise the results of this paper. We decided to end our analysis in May 2019 since this is the last date available for the open unemployment rate of the Metropolitan Region of São Paulo used for the forecasting, when this data series was deactivated.

We construct our dataset based on another dataset, built by the authors Lindenmeyer and Torrent (2023), where they gathered a base of 140 Brazilian macroeconomic time series from the following themes: balance of payments, capital stock, consumption and sales, currency and credit, employment, exchange, financial, foreign trade, national accounts, perception and

expectation, prices, production, public finance and salary and income.

We used the same dataset as the authors but removed three time series that presented missing values during the period analyzed. We also included two time series to the dataset that we also consider for prediction: the Brazilian Consumer Price Index called *Índice de Preços ao Consumidor Ampliado* (IPCA); and the Emerging Markets Bond Index Plus for Brazil that is the difference between the rates of return on bonds from Brazil and those offered by bonds issued by the US Treasury (EMBI+). This way, our final dataset consists of 139 variables for Brazilian macroeconomic time series.

To make the data stationary, we follow Lindenmeyer and Torrent (2023) and apply two tests: the Augmented Dickey Fuller (ADF) test, which was developed by Dickey and Fuller (1979), and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test developed by Kwiatkowski et al. (1992). We consider the dataset as treated when the information from both tests can be interpreted as being stationary.

2.3.2 Forecasting exercise

We make predictions considering the forecasting horizons $h = 1, 3, 6, 9, 12$ months using both rolling window and expanding window. To train our models, we use data from January 1996 to December 2008 and to test the results out-of-sample we consider the period from January 2009 to May 2019. Before estimating the forecasting, the data was standardized to ensure that all variables are on the same scale. This helps with the methods comparison and yields to better interpretation of the results.

Apart from the boosting and the alternative Elastic Net environment models, where we optimize hyperparameters every month, we re-optimize hyperparameters every year, i.e., every 12 months to improve computational performance.

In this study, we focus on the prediction of the following three macroeconomic variables that are indicators of the Brazilian economy using the data in Ipeadata:

a) open unemployment rate of the Metropolitan Region of São Paulo (UNRATE);

b) inflation rate using the Brazilian Consumer Price Index called *Índice de Preços ao Consumidor Ampliado* (IPCA);

c) spread using the Emerging Markets Bond Index Plus for Brazil, that is the difference between the rates of return on bonds from Brazil and those offered by bonds issued by the US Treasury (EMBI+).

2.3.3 Evaluation

We follow a standard practice in the forecasting literature and use the root Mean Squared Error (RMSE) to evaluate our forecasting methods. The root Mean Squared Error can be defined as

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{y}_{t+h|t,m} - y_{t+h})^2}, \tag{15}$$

where T is the total number of out-of-sample forecasts, $\hat{y}_{t+h|t,m}$ is the forecast of model m for time $t+h$, and y_{t+h} is the actual value in time $t+h$. The RMSE is measured in reference to the benchmark ar_bic in the following way:

$$relativeRMSPE_h^m = \frac{RMSPE_h^m}{RMSPE_h^{bench}}. \tag{16}$$

This way, when the relative RMSPE < 1, the alternative model performed better than the benchmark for a horizon h. When the relative RMSPE > 1, the benchmark performed better.

We also perform the Diebold-Mariano test, following Diebold and Mariano (1995) to test whether difference in expected loss is significantly different from zero for each model against the reference ar_bic, i.e.

$$H_0 : E\left(L(\hat{Y}_A, Y) - L(\hat{Y}_B, Y)\right) = 0 \tag{17}$$

This way if we reject H_0, we reject the hypothesis that both models perform equally well. We evaluate Diebold-Mariano test for 1%, 5% and 10% significance.

We also implement the Model Confidence Set (MCS) from Hansen and Nason (2011) that compares a given set of models to determine the best models for a given variable and horizon. The MCS is constructed from a collection of competing models where a criterion is used to evaluate these models empirically. An equivalence test is applied to the set and if this equivalence test is rejected, there is evidence that one model is better than the other and the null hypothesis of Equal Predictive Ability is rejected at a certain confidence level. In this case, an elimination rule is used to discard the models with poor sample performances from M until only the best models for the criterion used remain in the set M for a given level of confidence. In this paper, the loss function used for the MCS is the squared error and the test statistic used is the Tmax, as defined in Hansen and Nason (2011).

Lastly we use the cumulative squared forecast errors (CSFE) initially proposed by Welch and Goyal (2008) to graphically analyze when the models outperforms or underperforms the benchmark ar_bic. As described by Caldeira and Torrent (2016), the CSFE can be defined as

$$CSFE_{m,T} = \sum_{t=1}^{T}[(\hat{y}_{t+h|t,bench} - y_{t+h})^2 - (\hat{y}_{t+h|t,m} - y_{t+h})^2]. \tag{18}$$

This evaluation method measures the cumulative squared prediction errors of the model minus the cumulative squared prediction error of the benchmark. When a line increases, the alternative prediction is better; whenever it decreases, the benchmark prediction is better.

2.4 RESULTS

In this section, we detail the main findings of this forecasting study. Tables 2, 3 and 4 present the results found for the three variables studied using rolling window. Figures 1, 2 and 3 show the CSFE results for the models that performed better in the Model Confidence Set (MCS). We opted to report only 10 models in each plot to improve the data visualization. In the appendix A, we also present the results using expanding window.

For the prediction of EMBI+, we can see that for $h = 1$ most of the models root mean squared error (RMSE) are better than the benchmark (ar_bic), as can be seen in table 2. The best models in the Model Confidence Set (MCS) were the ones that used the lasso and Elastic Net penalties. The best model in $h = 1$ was b1_cv_en that outperformed the benchmark by approximately 22%. We can see that for all the horizons analyzed, the lasso and Elastic Net penalties performed well. However, we cannot statistically say that these models are better than the benchmark, according to the Diebold-Mariano test. We can also see that for the data-poor environment and for the data-rich B_1, B_2 and B_3 environment the results are very similar and usually the same, which indicates that the lasso and Elastic Net penalties shrink to zero several coefficients in the models and that the most important coefficient is the variable itself that is being predicted. The ridge penalty was outside the MCS for $h = 1$, however we can see that this model also performed well in the different horizons. Considering the Diebold-Mariano test, we can reject the hypothesis that this model and the benchmark performed equally well with 1% of significance for $h = 1$ and 10% of significance for $h = 3$. Analyzing figure 1, it is possible to check that for $h = 6, 9, 12$, the cumulative squared forecast errors (CSFE) of ridge prediction performs better than ar_bic for most of the prediction time.

Table 2 – EMBI+: RMSPE Rolling Window

	model	h=1	h=3	h=6	h=9	h=12
Data-poor models						
1	ar_bic	0.8980	0.7174	0.7081	0.7014	0.7062
2	ar_aic	1.0908***	**1.000**	**1.000**	**1.0454**	**1.0408**
3	ar_cv	0.8986***	**1.003**	**1.0029**	**1.0019**	**1.0253***
4	shrink_poor_cv_ridge	0.8721***	**1.0134***	**1.0029**	**1.0001**	**0.994**
5	shrink_poor_cv_lasso	**0.7841**	**0.9871**	**0.9988**	**1.0062**	**1.0003**
6	shrink_poor_cv_en	**0.7841**	**0.9871**	**0.9988**	**1.0062**	**1.0003**
7	rf_poor_cv	0.9253***	**1.0815****	**1.0445**	**1.0398**	**1.0511**
8	bols_poor_cv	0.9479***	**0.9923**	**0.9944**	**0.9994**	**1.0013**
9	bbs_poor_cv	0.8931***	**0.9947**	**1.0037**	**1.0156**	**1.0112**
10	svr_linear_poor_cv	1.0223***	**0.9968**	**1.026**	**1.0561**	**1.0267**
11	svr_rbf_poor_cv	1.0126***	**0.9936**	**1.1124****	1.2097***	1.1201***
Data-rich models						
12	ardi_bic	1.108***	**2.4782***	87.9048	271.9304	193.1272**
13	ardi_aic	3.6637***	8.4372***	87.9048	271.9304	193.1272**
14	ardi_cv	1.0113***	**1.0784*****	**1.1151***	**1.1064****	**1.1047**
15	shrink_rich_cv_ridge	0.9525***	**1.0309**	**1.1084****	**1.1057****	**1.0562**
16	shrink_rich_cv_lasso	0.785	**0.9866**	**0.9986**	**1.0067**	**1.0011**
17	shrink_rich_cv_en	0.7858	**0.9867**	**0.9974**	**1.0064**	**1.0006**
18	rf_rich_cv	0.8617***	**1.0676*****	1.0799***	**1.0475**	**1.0417**
19	bols_rich_cv	0.9354***	**1.0083**	**1.0656**	**1.0536**	**1.0551**
20	bbs_rich_cv	0.8758***	**0.9889**	**1.0072**	**1.0106**	**1.0015**
21	svr_linear_rich_cv	1.0869***	**0.9806**	**1.0322**	**1.1171****	**1.0879**
22	svr_rbf_rich_cv	0.848**	**0.994**	**0.9927**	**1.0105**	**1.0014**
23	b1_cv_boost	0.8847***	**1.0395****	**1.0292**	**1.0111**	**0.9998**
24	b1_cv_ridge	1.0067***	**1.2582*****	1.2835***	1.2132***	1.2079***
25	b1_cv_en	**0.7839**	**0.9871**	**0.9988**	**1.0062**	**1.0003**
26	b1_cv_lasso	**0.7841**	**0.9871**	**0.9988**	**1.0062**	**1.0003**
27	b2_cv_boost	0.8286**	**1.0115**	**1.0172**	**1.0347**	**1.014**
28	b2_cv_ridge	0.7999	**0.999**	**1.0061**	**1.0127**	**1.001**
29	b2_cv_en	**0.7841**	**0.9871**	**0.9988**	**1.0062**	**1.0003**
30	b2_cv_lasso	**0.7841**	**0.9871**	**0.9988**	**1.0062**	**1.0003**
31	b3_cv_boost	0.8705***	**1.0115**	**1.0138**	**1.0335**	**1.0048**
32	b3_cv_ridge	0.8083**	**1.0006**	**1.0039**	**1.0065**	**0.9937**
33	b3_cv_en	0.7842	**0.9871**	**0.9988**	**1.0062**	**1.0003**
34	b3_cv_lasso	**0.7841**	**0.9871**	**0.9988**	**1.0062**	**1.0003**

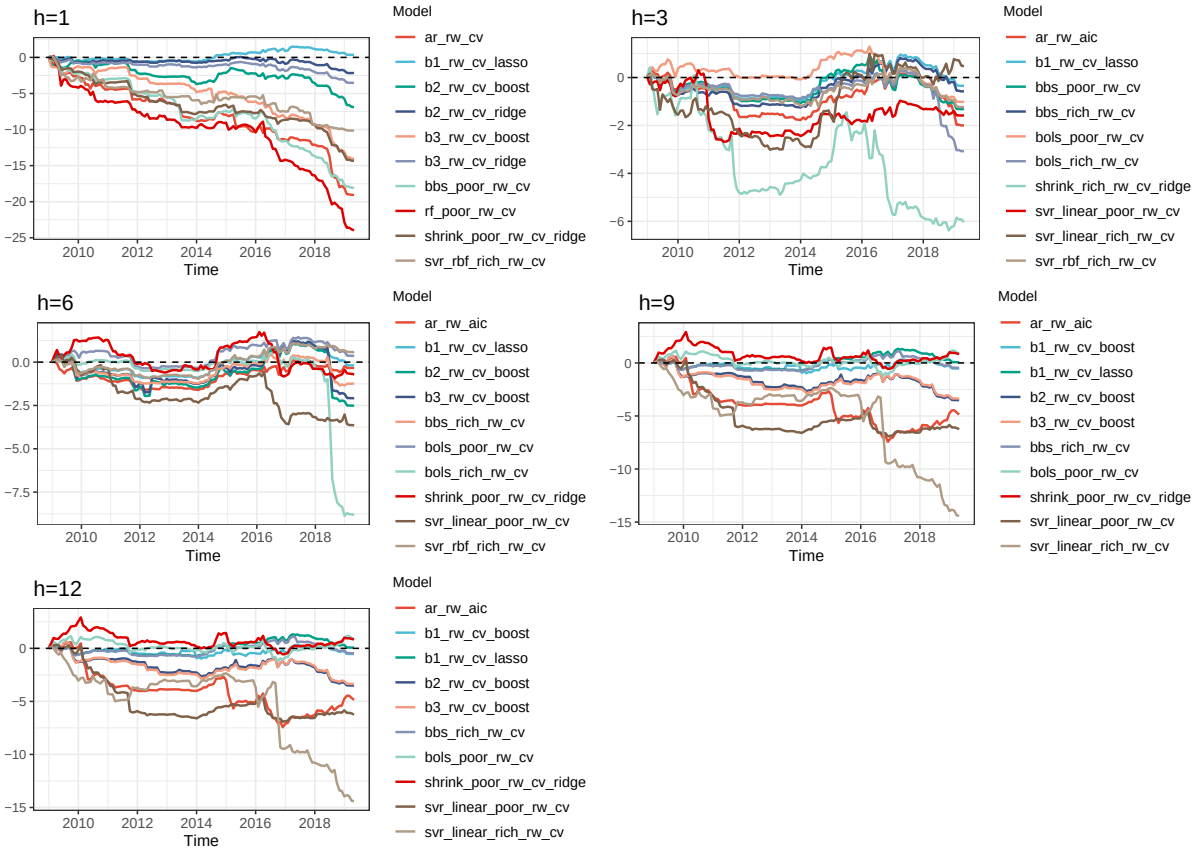
Source: Author.

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

For the other horizons $h = 3, 6, 9, 12$, the benchmark is harder to beat. In $h = 3$, the only model discarded in the MCS was ardi_aic and for $h = 6, 9, 12$, the models ardi_bic, ardi_aic and b1_cv_ridge were eliminated. Analyzing the ardi results, we can conclude that the models were overfit, especially for $h = 6, 9, 12$. This can be explained by the fact that the ardi model does not have any penalty and all coefficients are included with the same weights in the regression.

For the Brazilian inflation rate IPCA, the best RMSE result for all of the horizons was model b2_cv_lasso, that outperforms the benchmark by approximately 32% in $h = 1, 3$, 30.5% in $h = 6$, 31% in $h = 9$ and 25% for $h = 12$, as described in table 3. For $h = 3, 6, 9$, we can

Figure 1 – CSFE EMBI+ Rolling Window



Source: Author.

Note: In the plots, we report only 10 models that were inside the Model Confidence Set (MCS) to improve the data visualization.

also say that the difference in expected loss is significantly different from zero in reference to the model ar_bic. Here the best models were in the data-rich environment, especially in B_1, B_2 and B_3. Once again, the models with better RMSE had lasso or Elastic Net penalty. Other models that performed well with low RMSE and significance in the Diebold-Mariano test were the nonlinear model svr_rbf_rich_cv and the linear models b2_cv_boost and b2_cv_ridge. The ardi models were eliminated from the MCS due to overfitting.

Table 3 – IPCA: Relative RMSE Rolling Window

	model	h=1	h=3	h=6	h=9	h=12
Data-poor models						
1	ar_bic	1.0723	1.0827	1.0544	1.0701	0.9822
2	ar_aic	1.0955***	**1.045****	**1.0801****	**1.0953**	**1.0382**
3	ar_cv	1.0601***	**0.7714****	**0.7632***	**0.7855****	**1.2103**
4	shrink_poor_cv_ridge	**0.9278*****	**0.9123****	**0.9747***	**0.9838**	**1.0046**
5	shrink_poor_cv_lasso	**0.7624****	**0.6926****	**0.7014****	**0.6983****	**0.7577***
6	shrink_poor_cv_en	**0.8521*****	**0.761****	**0.7740****	**0.7626***	**0.8126**
7	rf_poor_cv	1.0003***	**0.9886****	**0.9953****	**1.0119***	**1.1149***
8	bols_poor_cv	1.0457***	**1.0015****	**0.9887***	**1.0046***	**1.0185**
9	bbs_poor_cv	**1.0629*****	**1.2121****	**1.7610**	**1.8225**	**2.033**
10	svr_linear_poor_cv	1.0457***	**0.9946****	**1.0739****	**1.0434***	**1.0495**
11	svr_rbf_poor_cv	0.9744***	**0.8518*****	**0.8223****	**0.8119***	**0.8843***
Data-rich models						
12	ardi_bic	1.0049***	**0.8567****	306.6598	862.5396	785.3151
13	ardi_aic	2.2529***	4.7945***	306.6598	862.5396	785.3151
14	ardi_cv	1.0956***	**1.0419****	**1.1724****	**1.0357***	**1.2261**
15	shrink_rich_cv_ridge	**0.9428*****	**0.8996****	**0.9934***	**1.0381***	**0.9775**
16	shrink_rich_cv_lasso	**0.7721*****	**0.6849****	**0.7023****	**0.6945****	**0.7548***
17	shrink_rich_cv_en	**0.8489*****	**0.7544****	**0.7549****	**0.7562***	**0.8053**
18	rf_rich_cv	**0.8899*****	**0.8631*****	**0.9201*****	**0.9294****	**0.9878***
19	bols_rich_cv	**0.9755*****	**0.9075****	**0.9062****	**0.8973***	**0.9066**
20	bbs_rich_cv	**0.982*****	**1.0323****	**1.1205**	**1.0937**	**1.1920**
21	svr_linear_rich_cv	1.0164***	**0.9164*****	1.0051**	**0.9331***	**0.9394***
22	svr_rbf_rich_cv	**0.7457****	**0.6848****	**0.7063*****	**0.6917****	**0.7788***
23	b1_cv_boost	**0.8889*****	**0.8383****	**0.8773****	**0.8567***	**0.8596**
24	b1_cv_ridge	**0.8485*****	**0.8913*****	**0.942*****	**0.9249*****	**1.0034****
25	b1_cv_en	**0.8021*****	**0.7308****	**0.7413****	**0.7327***	**0.7725***
26	b1_cv_lasso	**0.7301****	**0.6803****	**0.6947****	**0.687***	**0.7493**
27	b2_cv_boost	**0.7500*****	**0.7017***	**0.6989***	**0.7014****	**0.7649***
28	b2_cv_ridge	**0.6901**	**0.6800****	**0.7035****	**0.6952****	**0.7546***
29	b2_cv_en	**0.6886**	**0.6774****	**0.6946***	**0.6868***	**0.7505***
30	b2_cv_lasso	**0.6821**	**0.6760****	**0.6946***	**0.6860***	**0.7493**
31	b3_cv_boost	**0.9303*****	**0.8488****	**0.8514***	**0.8468***	**0.8797**
32	b3_cv_ridge	**0.7029***	**0.7018****	**0.7262*****	**0.7181****	**0.7760****
33	b3_cv_en	**0.8150*****	**0.7367****	**0.7393****	**0.7315***	**0.7885***
34	b3_cv_lasso	**0.7270****	**0.6826****	**0.6948****	**0.6873***	**0.7496**

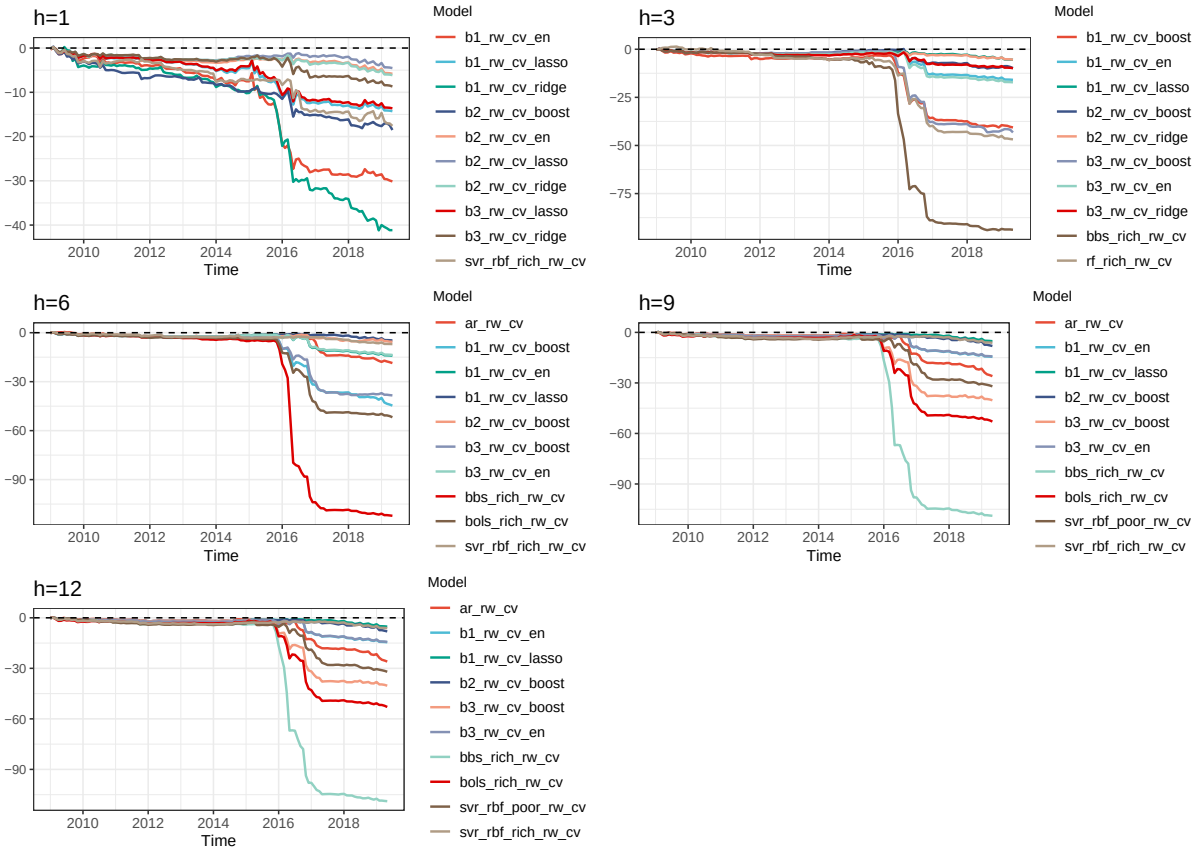
Source: Author.

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

When we analyze the CSFE results in figure 2, we can see that most models performed close to the benchmark for most of the prediction time. In 2016, we can see that many models have a significant loss of performance, and after 2017, the prediction stabilizes again. This loss of performance can be related to the Brazilian severe economic recession of 2016, that resulted in a sharp decline of GDP, high unemployment rates, and an increase of the inflation rate during this period.

By analyzing the results for the unemployment rate of the Metropolitan Region of São Paulo (UNRATE) in table 4, we can see once again that the models with better RMSE are the

Figure 2 – CSFE IPCA Rolling Window



Source: Author.

Note: In the plots, we report only 10 models that were inside the Model Confidence Set (MCS) to improve the data visualization.

ones with lasso and Elastic Net penalties and these models are also included as best models in the MCS. In $h = 1$, the model with best RMSE is the shrink_poor_cv_lasso which is in the data-poor environment and outperfomed the benchmark by 36%. In all other horizons, the best models have lasso or Elastic Net penalty in the data-rich environment. In $h = 3$, the model with the best RMSE was b1_cv_lasso, outperforming the benchmark by approximately 32% and with 10% significance in Diebold-Mariano test. The other models included in the MCS are the models with ridge penalty (b2_cv_ridge, b3_cv_ridge and shrink_poor_cv_ridge in $h = (6, 9, 12)$, rf_rich_cv in $h = (1, 3, 6)$, rf_poor_cv in $h = 6$ and svr_rbf_rich_ CV in all of the horizons. The boosting models were also included in MCS (bbs_poor_cv in $h = (6, 9, 12)$, b2_cv_boost in $h = (3, 6, 9)$ and b3_cv_boost in $h = 6$). Once again, the ardi models were overfit. Analyzing figure 3, we can also conclude that the models that consistently outperform the benchmark ar_bic are mainly the ones with lasso and Elastic Net penalties.

Table 4 – UNRATE: Relative RMSE Rolling Window

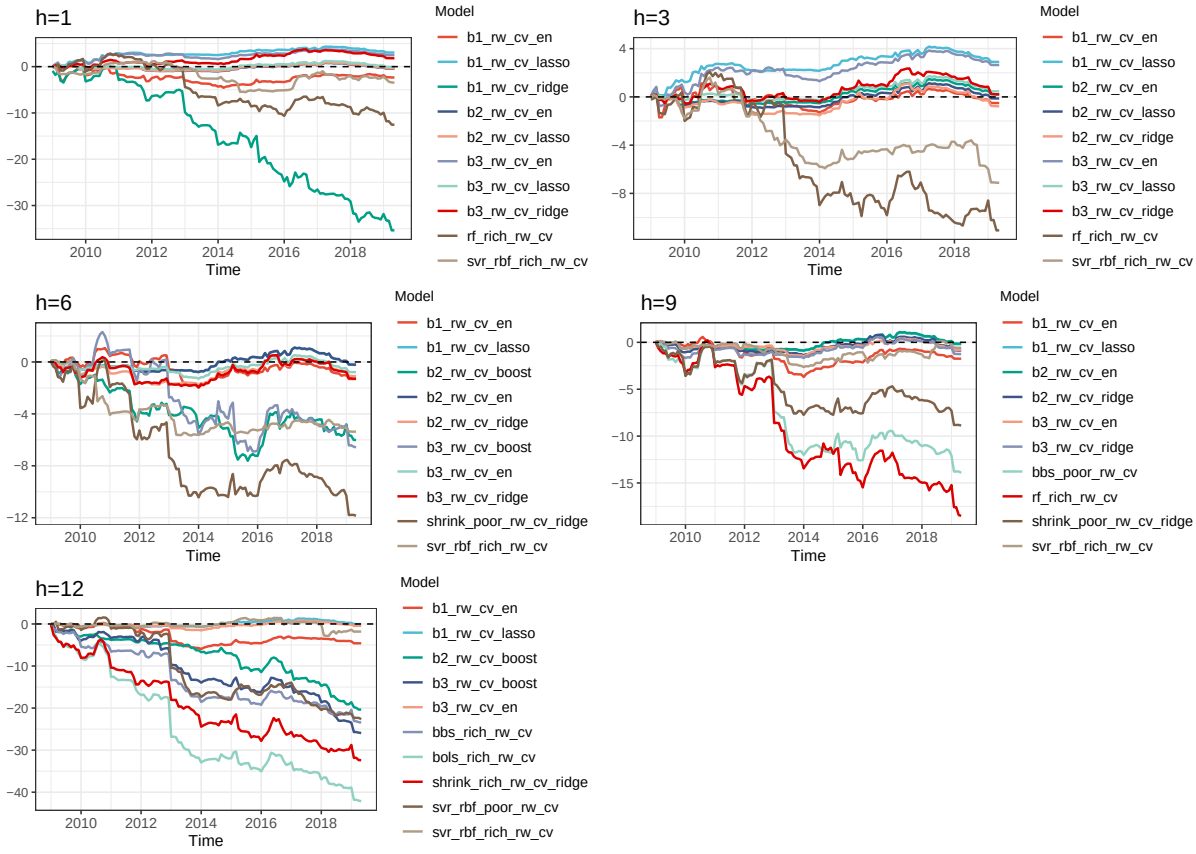
	model	h=1	h=3	h=6	h=9	h=12
Data-poor models						
1	ar_bic	1.0744	1.0277	0.8764	0.881	0.8432
2	ar_aic	1.0006***	1.001***	1.0315***	0.9805**	1.0151**
3	ar_cv	0.9277***	0.9831***	1.0374***	1.0158***	1.0494***
4	shrink_poor_cv_ridge	0.7224	0.7499	**0.8790**	**0.8567**	**0.9045**
5	shrink_poor_cv_lasso	**0.6400**	**0.6729**	**0.8072**	**0.8028**	**0.8376**
6	shrink_poor_cv_en	**0.6547**	**0.6870**	**0.8064**	**0.8053**	**0.8398**
7	rf_poor_cv	0.7360*	0.7840**	**0.9156***	0.9124*	0.9588**
8	bols_poor_cv	0.8666***	0.9121***	0.9171*	0.9177**	0.9747**
9	bbs_poor_cv	0.8462***	0.8635***	**0.908***	**0.8866**	**0.9426**
10	svr_linear_poor_cv	0.9844***	0.9674***	1.0164***	1.0256***	1.0044***
11	svr_rbf_poor_cv	0.7881***	0.8152***	0.9185***	0.9393**	0.9778*
Data-rich models						
12	ardi_bic	1.0284***	2.5647***	64.1994	151.8136	326.5195
13	ardi_aic	2.9764***	5.4569***	64.1994	151.8136	326.5195
14	ardi_cv	1.0736***	1.1488***	1.2276***	1.4296***	1.3647***
15	shrink_rich_cv_ridge	0.7416*	0.8351***	1.0223***	1.0256***	1.0324***
16	shrink_rich_cv_lasso	**0.6415**	**0.6735**	**0.8068**	**0.8030**	**0.8363**
17	shrink_rich_cv_en	**0.6585**	**0.6782**	**0.8051**	**0.8051**	**0.8375**
18	rf_rich_cv	**0.7204**	**0.7457**	**0.9079***	0.9128**	0.9459**
19	bols_rich_cv	0.8434***	0.858***	0.9573**	1.027***	1.0843***
20	bbs_rich_cv	0.7973***	0.8493***	0.9391**	0.9256**	0.9824**
21	svr_linear_rich_cv	0.9625***	1.0141***	1.0219***	1.1246***	1.0882***
22	svr_rbf_rich_cv	**0.6753**	**0.7254**	**0.8397**	**0.8053**	**0.8496**
23	b1_cv_boost	0.8937***	0.9138***	1.0225***	1.0476***	1.1797***
24	b1_cv_ridge	0.8229***	0.8827***	0.9985**	1.0293***	1.0623***
25	b1_cv_en	**0.6695**	**0.6900**	**0.8142**	**0.8127**	**0.8678**
26	b1_cv_lasso	**0.6410**	**0.6710***	**0.8072**	**0.8028**	**0.8377**
27	b2_cv_boost	0.7557***	**0.7426**	**0.8437**	**0.8751***	0.9646***
28	b2_cv_ridge	**0.6540**	**0.6914**	**0.8128**	**0.8071**	**0.8368**
29	b2_cv_en	**0.6597**	**0.6860**	**0.8070**	**0.8024**	**0.8376**
30	b2_cv_lasso	**0.6592**	**0.6879**	**0.8072**	**0.8028**	**0.8376**
31	b3_cv_boost	0.7530**	0.7976**	**0.8472**	0.9235***	0.9966***
32	b3_cv_ridge	**0.6476**	**0.6860**	**0.8140**	**0.8096**	**0.8382**
33	b3_cv_en	**0.6441**	**0.6725**	**0.8108**	**0.8066**	**0.8405**
34	b3_cv_lasso	**0.6576**	**0.6846**	**0.8072**	**0.8028**	**0.8376**

Source: Author.

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

By analyzing the results for the three variables, we can conclude that the best models are the ones with lasso and Elastic Net penalties, especially in the data-rich environment. This result is different than the analysis in Goulet Coulombe et al. (2022). In their study, the authors concluded: "[...] nonlinearities are the true game changer for the data-rich environment, as they improve substantially the forecasting accuracy for all macroeconomic variables in our exercise and especially when predicting at long horizons." For the Brazilian macroeconomic forecasting, the nonlinear models didn't have the best performance and had trouble beating the benchmark ar_bic. This difference can be related to the smaller number of observations

Figure 3 – CSFE UNRATE Rolling Window



Source: Author.

Note: In the plots, we report only 10 models that were inside the Model Confidence Set (MCS) to improve the data visualization.

available for Brazilian macroeconomic forecasting and therefore less years available for the training sample. Since nonparametric estimation assumes no structure on the data, it is necessary a large training sample to guarantee a smaller variance. Also, as discussed in Clements, Franses, and Swanson (2004), a model that captures nonlinearities in the dataset and generates a good in-sample fit does not necessarily mean that it will also generate a good out-of-sample forecast as there are many unknowns and the economic system is extremely complex.

For the regularization, we can conclude that the data-rich environment leads to better results than the data-poor environment, however we found that the Elastic Net regularizations with B_1, B_2 and B_3 usually contribute to generating better models, especially lasso and Elastic Net. In addition, the predictions for ridge penalty and boosting models also improve in these environments. Regarding the loss function, we reach similar conclusions as the authors in Goulet Coulombe et al. (2022) that the main benefits from the use of SVR is the nonlinearity of the RBF kernel and the squared error is preferred to the $\bar{\epsilon}$-insensitive loss function.

The results for the Brazilian inflation rate are consistent with Garcia, Medeiros, and Vasconcelos (2017) and Medeiros, Vasconcelos, and Freitas (2016) that showed that models with the lasso penalty, such as lasso and the Flex-adaLasso are the best models for short horizons. On the other hand, the results found in this paper are different for longer horizons. In Medeiros,

Vasconcelos, and Freitas (2016), the author showed that for longer horizons, the AR and the factor models performed better. Also, in Araujo and Gaglianone (2023), the authors showed that the nonlinear models such as Random Forest play an important role to forecast the Brazilian inflation rate at longer horizons. For our study, we have showed that the models with lasso and Elastic Net penalties performed better than the nonlinear models even for longer horizons, however it is possible to see that the nonlinear models become more competitive in longer horizons.

In the appendix A, we show the results using an expanding window. By analyzing the results, we can reach the same conclusions as the rolling window analysis.

2.5 CONCLUSION

This paper's goal was to understand what are the key characteristics of machine learning that are useful for macroeconomic forecasting. We studied a long-term comprehensive dataset of Brazilian macroeconomic variables from Ipeadata, spanning almost 23 years, and conducted a forecasting horse-race between 34 models at five different horizons to identify what makes a good forecasting model for the Brazilian macroeconomic variables, considering the four main features of machine learning. By analyzing the results for the three Brazilian variables unemployment rate, inflation and spread, we can conclude that:

a) The ar_bic model is not so easy to beat, especially when considering a big lag order for the model. Even so, there were some models that consistently outperformed the benchmark for the three variables considered.

b) The nonlinear models didn't perform the best as one might expect; however, they became more competitive in longer horizons. Considering the trade-off between bias and variance, even though the nonlinear models can help to decrease bias, they increase the variance of the prediction, and can harm the forecasting performance.

c) The lasso and Elastic Net penalties models have the ability to perform variable selection and overall had the best relative RMSE, outperforming the benchmark ar_bic as much as 36%. Also, they were always contained in the set of best models of the MCS. This is in line with the findings of the authors for the Brazilian inflation rate Garcia, Medeiros, and Vasconcelos (2017) and Medeiros, Vasconcelos, and Freitas (2016) that showed that models with the lasso penalty are the best for short horizons.

d) We can also affirm that the data-rich environment is the best and the alternative Elastic Net penalty can contribute to improving the predictability and making the models more competitive, for example the ridge penalty and the boosting models.

e) Lastly, we can conclude that the squared error loss function remains as being the best for forecasting, as also affirmed by Goulet Coulombe et al. (2022).

Overall, by comparing the results of this paper with the ones found by Goulet Coulombe et al. (2022), we can also say that the key features of machine learning can vary depending on the variables and on the country that is being analyzed, and the results can change in different contexts. We suggest future research on studying the main characteristics of machine learning for other emerging countries and to extend the analysis to other macroeconomic variables. In addition, including different machine learning models is desirable for understanding how the characteristics may change depending on the context of the macroeconomic forecasting analysis.

2.6 REFERENCES

AKAIKE, Htrotugu. Maximum likelihood identification of Gaussian autoregressive moving average models. **Biometrika**, United Kingdom, v. 60, p. 255–265, Aug. 1973. Available from: https://doi.org/10.1093/biomet/60.2.255. Visited on: 9 Jan. 2024.

ARAUJO, Gustavo Silva; GAGLIANONE, Wagner Piazza. Machine learning methods for inflation forecasting in Brazil: New contenders versus classical models. **Latin American Journal of Central Banking**, Netherlands, v. 4, p. 100087, June 2023. Available from: https://doi.org/10.1016/j.latcb.2023.100087. Visited on: 5 Jan. 2024.

BAI, Jushan; NG, Serena. Boosting diffusion indices. **Journal of Applied Econometrics**, United Kingdom, v. 24, p. 607–629, Mar. 2009. Available from: https://doi.org/10.1002/jae.1063. Visited on: 24 Oct. 2022.

CALDEIRA, João; TORRENT, Hudson. Forecasting the US Term Structure of Interest Rates Using Nonparametric Functional Data Analysis. **Journal of Forecasting**, United Kingdom, v. 36, p. 56–73, Apr. 2016. Available from: https://doi.org/10.1002/for.2414. Visited on: 8 Jan. 2024.

CLEMENTS, Michael P.; FRANSES, Philip Hans; SWANSON, Norman R. Forecasting economic and financial time-series with non-linear models. **International Journal of Forecasting**, Netherlands, v. 20, p. 169–183, Apr. 2004. Available from: https://doi.org/10.1016/j.ijforecast.2003.10.004. Visited on: 7 Jan. 2024.

DIEBOLD, Francis X.; MARIANO, Roberto S. Comparing Predictive Accuracy. **Journal of Business Economic Statistics**, United States, v. 13, p. 253–263, July 1995. Available from: https://doi.org/10.1080/07350015.1995.10524599. Visited on: 27 Dec. 2023.

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**, Netherlands, v. 33, p. 679–693, July 2017. Available from: https://doi.org/10.1016/j.ijforecast.2017.02.002. Visited on: 9 Dec. 2019.

GOULET COULOMBE, Philippe; LEROUX, Maxime; STEVANOVIC, Dalibor; SURPRENANT, Stéphane. How is machine learning useful for macroeconomic forecasting? **Journal of Applied Econometrics**, United Kingdom, v. 37, p. 920–964, June 2022. Available from: https://doi.org/10.1002/jae.2910. Visited on: 27 Dec. 2023.

HANSEN, Peter R.; NASON, James M. The Model Confidence Set. **Econometrica**, United States, v. 79, p. 453–497, Mar. 2011. Available from: https://doi.org/10.3982/ECTA5771. Visited on: 3 Jan. 2024.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome H. **The Elements of Statistical Learning:** Data Mining, Inference, and Prediction. 2nd ed. New York: Springer, 2009.

HOFNER, Benjamin; MAYR, Andreas; ROBINZONOV, Nikolay; SCHMID, Matthias. Model-based boosting in R: a hands-on tutorial using the R package mboost. **Computational Statistics**, Germany, v. 29, p. 3–35, Dec. 2012. Available from: https://doi.org/10.1007/s00180-012-0382-5. Visited on: 30 Dec. 2023.

JAMES, Gareth; WITTEN, Daniela; HASTIE, Trevor; TIBSHIRANI, Robert. **An introduction to statistical learning:** with applications in R. 2nd ed. New York: Springer, 2013.

KAUPPI, Heikki; VIRTANEN, Timo. Boosting nonlinear predictability of macroeconomic time series. **International Journal of Forecasting**, Netherlands, v. 37, p. 151–170, May 2020. Available from: https://doi.org/10.1016/j.ijforecast.2020.03.008. Visited on: 9 Jan. 2024.

LINDENMEYER, Guilherme Schultz; TORRENT, Hudson da Silva. Boosting and Predictability of Macroeconomic Variables: Evidence from Brazil. **Computational Economics**, Springer Science+Business Media, Netherlands, Aug. 2023. Available from: https://doi.org/10.1007/s10614-023-10421-3. Visited on: 3 Jan. 2024.

MEDEIROS, Marcelo C; VASCONCELOS, Gabriel; FREITAS, Eduardo. Forecasting Brazilian Inflation with High-Dimensional Models. **Brazilian Review of Econometrics**, Brazil, v. 36, p. 223–254, Nov. 2016. Available from: https://doi.org/10.12660/bre.v99n992016.52273. Visited on: 5 Jan. 2024.

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**, United States, v. 39, p. 98–119, Aug. 2019. Available from: https://doi.org/10.1080/07350015.2019.1637745. Visited on: 9 Jan. 2024.

MOSHIRI, Saeed; CAMERON, Norman. Neural network versus econometric models in forecasting inflation. **Journal of Forecasting**, United Kingdom, v. 19, p. 201–217, Apr. 2000. Available from: https://doi.org/10.1002/(sici)1099-131x(200004)19:3%3C201::aid-for753%3E3.0.co;2-4. Visited on: 8 Jan. 2024.

MULLAINATHAN, Sendhil; SPIESS, Jann. Machine Learning: an Applied Econometric Approach. **Journal of Economic Perspectives**, United States, v. 31, p. 87–106, May 2017. Available from: https://doi.org/10.1257/jep.31.2.87. Visited on: 8 Jan. 2024.

SCHMID, Matthias; HOTHORN, Torsten. Boosting additive models using component-wise P-Splines. **Computational Statistics Data Analysis**, Netherlands, v. 53, p. 298–311, Dec. 2008. Available from: https://doi.org/10.1016/j.csda.2008.09.009. Visited on: 24 Sept. 2021.

SCHWARZ, Gideon. Estimating the Dimension of a Model. **The Annals of Statistics**, United States, v. 6, p. 461–464, Mar. 1978. Available from: https://www.jstor.org/stable/2958889. Visited on: 9 Jan. 2024.

SERMPINIS, Georgios; STASINAKIS, Charalampos; THEOFILATOS, Konstantinos; KARATHANASOPOULOS, Andreas. Inflation and Unemployment Forecasting with Genetic Support Vector Regression. **Journal of Forecasting**, United Kingdom, v. 33, p. 471–487, May 2014. Available from: https://doi.org/10.1002/for.2296. Visited on: 8 Jan. 2024.

STOCK, James H; WATSON, Mark W. Macroeconomic Forecasting Using Diffusion Indexes. **Journal of Business Economic Statistics**, United States, v. 20, p. 147–162, Apr. 2002. Available from: https://doi.org/10.1198/073500102317351921. Visited on: 27 Dec. 2023.

VAPNIK, Vladimir Naoumovitch. **The nature of statistical learning theory**. Springer, Cop, 1996.

WELCH, Ivo; GOYAL, Amit. A Comprehensive Look at The Empirical Performance of Equity Premium Prediction. **Review of Financial Studies**, v. 21, p. 1455–1508, Mar. 2008. Available from: https://doi.org/10.1093/rfs/hhm014. Visited on: 8 Jan. 2024.

ZOU, Hui; HASTIE, Trevor. Regularization and variable selection via the elastic net. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, United Kingdom, v. 67,

p. 301–320, Apr. 2005. Available from: https://doi.org/10.1111/j.1467-9868.2005.00503.x. Visited on: 27 Dec. 2023.

3 CONCLUSION

This work aimed to understand what are the key characteristics of machine learning that are useful for macroeconomic forecasting. . We studied a long-term comprehensive dataset of Brazilian macroeconomic variables from Ipeadata, spanning almost 23 years, and conducted a forecasting horse-race between 34 models at five different horizons to identify what makes a good forecasting model for the Brazilian macroeconomic variables, considering the four main features of machine learning: the function considered for prediction, the regularization penalty, the set of hyperparameters and the loss function used. To analyze the predictions made, we followed the known practice to calculate the relative RMSE and performed the Diebold-Mariano test and the Model Confidence Set (MCS). Finally, we also analyzed the results graphically by using the cumulative squared forecast errors (CSFE).

By analyzing the results for the three Brazilian variables unemployment rate, inflation and spread, we can arrive to the following conclusions:

a) The ar_bic model is not so easy to beat, especially when we consider a big lag order for the model. Even so, there were some models that consistently outperformed the benchmark for the three variables considered.

b) The lasso and Elastic Net penalties models had overall the best relative RMSE, outperforming the benchmark ar_bic as much as 36%. Also, they were always contained in the best models of the MCS. This is in line with the findings of the authors for the Brazilian inflation rate Garcia, Medeiros, and Vasconcelos (2017) and Medeiros, Vasconcelos, and Freitas (2016) that showed that models with the lasso penalty are the best for short horizons. The lasso has the benefit to do variable selection, while the Elastic Net combines the benefits from ridge and lasso penalties. According to the authors in Zou and Hastie (2005): "Similar to the lasso, the Elastic Net simultaneously does automatic variable selection and continuous shrinkage, and it can select groups of correlated variables." By analyzing the nonlinear models, they didn't have the best performance, however they became more competitive in longer horizons.

c) We can also affirm that the data-rich environment is the best and the alternative Elastic Net penalty can contribute to improving the predictability and making the models more competitive, for example the ridge penalty and the boosting models.

d) We can conclude that the squared error loss function remains as being the best for forecasting, as also affirmed by Goulet Coulombe et al. (2022).

Lastly, by comparing the results of this paper with the ones found by Goulet Coulombe et al. (2022), we can also say that the key features of machine learning can vary depending on the variables and on the country that is being analyzed, and the results can change in different contexts. We suggest future research on studying the main characteristics of machine learning

for other emerging countries and to extend the analysis to other macroeconomic variables. In addition, including different machine learning models is desirable for understanding how the characteristics may change depending on the context of the macroeconomic forecasting analysis.

REFERENCES

ARAUJO, Gustavo Silva; GAGLIANONE, Wagner Piazza. Machine learning methods for inflation forecasting in Brazil: New contenders versus classical models. **Latin American Journal of Central Banking**, Netherlands, v. 4, p. 100087, June 2023. Available from: https://doi.org/10.1016/j.latcb.2023.100087. Visited on: 5 Jan. 2024.

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**, Netherlands, v. 33, p. 679–693, July 2017. Available from: https://doi.org/10.1016/j.ijforecast.2017.02.002. Visited on: 9 Dec. 2019.

GOULET COULOMBE, Philippe; LEROUX, Maxime; STEVANOVIC, Dalibor; SURPRENANT, Stéphane. How is machine learning useful for macroeconomic forecasting? **Journal of Applied Econometrics**, United Kingdom, v. 37, p. 920–964, June 2022. Available from: https://doi.org/10.1002/jae.2910. Visited on: 27 Dec. 2023.

LINDENMEYER, Guilherme Schultz; TORRENT, Hudson da Silva. Boosting and Predictability of Macroeconomic Variables: Evidence from Brazil. **Computational Economics**, Springer Science+Business Media, Netherlands, Aug. 2023. Available from: https://doi.org/10.1007/s10614-023-10421-3. Visited on: 3 Jan. 2024.

MEDEIROS, Marcelo C; VASCONCELOS, Gabriel; FREITAS, Eduardo. Forecasting Brazilian Inflation with High-Dimensional Models. **Brazilian Review of Econometrics**, Brazil, v. 36, p. 223–254, Nov. 2016. Available from: https://doi.org/10.12660/bre.v99n992016.52273. Visited on: 5 Jan. 2024.

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**, United States, v. 39, p. 98–119, Aug. 2019. Available from: https://doi.org/10.1080/07350015.2019.1637745. Visited on: 9 Jan. 2024.

MOSHIRI, Saeed; CAMERON, Norman. Neural network versus econometric models in forecasting inflation. **Journal of Forecasting**, United Kingdom, v. 19, p. 201–217, Apr. 2000. Available from: https://doi.org/10.1002/(sici)1099-131x(200004)19:3%3C201::aid-for753%3E3.0.co;2-4. Visited on: 8 Jan. 2024.

MULLAINATHAN, Sendhil; SPIESS, Jann. Machine Learning: an Applied Econometric Approach. **Journal of Economic Perspectives**, United States, v. 31, p. 87–106, May 2017. Available from: https://doi.org/10.1257/jep.31.2.87. Visited on: 8 Jan. 2024.

SERMPINIS, Georgios; STASINAKIS, Charalampos; THEOFILATOS, Konstantinos; KARATHANASOPOULOS, Andreas. Inflation and Unemployment Forecasting with Genetic Support Vector Regression. **Journal of Forecasting**, United Kingdom, v. 33, p. 471–487, May 2014. Available from: https://doi.org/10.1002/for.2296. Visited on: 8 Jan. 2024.

ZOU, Hui; HASTIE, Trevor. Regularization and variable selection via the elastic net. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, United Kingdom, v. 67, p. 301–320, Apr. 2005. Available from: https://doi.org/10.1111/j.1467-9868.2005.00503.x. Visited on: 27 Dec. 2023.

APPENDIX A – RESULTS USING EXPANDING WINDOW

In this appendix, we present the results using the expanding window for forecasting.

Table 5 – EMBI+: Relative RMSE Expanding Window

	Model	h=1	h=3	h=6	h=9	h=12
Data-poor models						
1	ar_bic	0.8891	0.7088	0.7089	0.7065	0.7070
2	ar_aic	1.093***	**1.000**	**1.000**	**0.9753***	**1.0036**
3	ar_cv	**0.9013*****	**1.0037**	**1.0059**	**1.0091**	**1.003**
4	shrink_poor_cv_ridge	**0.872*****	**1.0076**	**1.0036**	**0.9737**	**0.9869**
5	shrink_poor_cv_lasso	**0.7935**	**0.9980**	**0.9967**	**1.0002**	**0.9996**
6	shrink_poor_cv_en	**0.7935**	**0.9980**	**0.9967**	**1.0002**	**0.9996**
7	rf_poor_cv	**0.9195*****	1.096**	**1.0849***	**1.0207**	**1.0636**
8	bols_poor_cv	**0.965*****	**0.9982**	**0.9951**	**0.9963**	**0.9772**
9	bbs_poor_cv	**0.9025*****	**0.9961**	**1.0095**	**1.0005**	**1.0023**
10	svr_linear_poor_cv	1.0185***	**0.9868**	**0.9954**	**0.9964**	**0.9713**
11	svr_rbf_poor_cv	1.0352***	**1.0073**	**1.011**	**1.011**	**1.0237**
Data-rich models						
12	ardi_bic	1.1107***	**1.0046**	**11.3348**	**7.9001**	**10.1987**
13	ardi_aic	**1.5711*****	3.3188*	11.4972	8.1235	10.3186
14	ardi_cv	1.019***	**1.017**	**1.0279**	**1.0415****	**1.1008**
15	shrink_rich_cv_ridge	**0.9419*****	**1.002**	**1.0095**	**1.0295**	**1.0358**
16	shrink_rich_cv_lasso	**0.7936**	**0.9979**	**0.9967**	**1.0002**	**0.9995**
17	shrink_rich_cv_en	**0.7936**	**0.9980**	**0.9967**	**1.0003**	**0.9996**
18	rf_rich_cv	**0.9119*****	1.0705***	**1.0513****	**1.0353**	**1.059*****
19	bols_rich_cv	**0.9537*****	**0.9982**	**0.9982**	**1.0049**	**1.0039**
20	bbs_rich_cv	**0.9205*****	**1.0005**	**1.0046**	**1.0027**	**0.9984**
21	svr_linear_rich_cv	1.0969***	**0.9945**	**1.0201**	**1.0319**	**1.0429**
22	svr_rbf_rich_cv	**0.8716*****	**0.9999**	**1.0011**	**1.0027**	**1.0013**
23	b1_cv_boost	**0.9296*****	**1.0651**	**1.0472**	**1.0009**	**0.9996**
24	b1_cv_ridge	**0.9916*****	1.2244***	**1.2185*****	**1.1924*****	**1.1985*****
25	b1_cv_en	**0.7938**	**0.9980**	**0.9967**	**1.0002**	**0.9996**
26	b1_cv_lasso	**0.7935**	**0.9980**	**0.9967**	**1.0002**	**0.9996**
27	b2_cv_boost	**0.8436****	**1.0195**	**0.9924**	**0.9925**	**0.9973**
28	b2_cv_ridge	**0.8714*****	1.0853***	**1.0923*****	**1.0647***	**1.0385**
29	b2_cv_en	**0.7935**	**0.9980**	**0.9967**	**1.0002**	**0.9996**
30	b2_cv_lasso	**0.7935**	**0.9980**	**0.9967**	**1.0002**	**0.9996**
31	b3_cv_boost	**0.9108*****	**1.0204**	**0.9911**	**0.9873**	**1.0098**
32	b3_cv_ridge	**0.8782*****	1.0851***	**1.0898*****	**1.0568***	**1.0308**
33	b3_cv_en	**0.7935**	**0.9980**	**0.9967**	**1.0002**	**0.9996**
34	b3_cv_lasso	**0.7935**	**0.9980**	**0.9967**	**1.0002**	**0.9996**

Source: Author.

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

Table 6 – IPCA: Relative RMSE Expanding Window

	model	h=1	h=3	h=6	h=9	h=12
Data-poor models						
1	ar_bic	1.0671	0.9584	0.8746	0.8582	0.8122
2	ar_aic	1.0436***	**1.038*** **	**1.0276** **	**1.0014***	**1.000** **
3	ar_cv	0.9866***	**0.9014*** **	**0.9198*** **	**0.9077*** **	**1.0213***
4	shrink_poor_cv_ridge	0.9322***	**0.922** **	**0.9549** **	**0.9584***	**1.0118***
5	shrink_poor_cv_lasso	**0.7308***	**0.7642***	**0.8371** **	**0.8509***	**0.8991***
6	shrink_poor_cv_en	0.7956***	**0.7756** **	**0.8382** **	**0.8536** **	**0.8996***
7	rf_poor_cv	0.9879***	**1.0347** **	**1.0287** **	**1.0649***	**1.1867***
8	bols_poor_cv	1.0174***	**0.9945*** **	**0.9377** **	**0.9742***	**1.0076**
9	bbs_poor_cv	1.0187***	**0.919** **	**0.9282** **	**0.9515***	**0.9799***
10	svr_linear_poor_cv	1.0207***	**0.9909*** **	**1.0751** **	**0.9732** **	**0.9908** **
11	svr_rbf_poor_cv	0.9633***	**0.9333** **	**0.9575*** **	**0.9248*** **	**0.9492** **
Data-rich models						
12	ardi_bic	0.9998***	**0.9191*** **	**2.2917**	**30.2984**	**3.1345**
13	ardi_aic	1.2353***	**1.349** **	**2.4489***	30.373	**3.3158**
14	ardi_cv	1.0021***	**1.0153*** **	**1.1758*** **	**1.077** **	**1.0316*** **
15	shrink_rich_cv_ridge	**0.8785*** **	**0.8974*** **	**0.9907** **	**1.0402***	**0.9577*** **
16	shrink_rich_cv_lasso	**0.7454** **	**0.7652***	**0.8403** **	**0.856** **	**0.9009***
17	shrink_rich_cv_en	0.8092***	**0.7798** **	**0.8418** **	**0.8578** **	**0.9002***
18	rf_rich_cv	**0.8785*** **	**0.8705** **	**0.9718*** **	**1.0169***	**1.0705***
19	bols_rich_cv	0.9575***	**0.922*** **	**0.9255** **	**0.9402** **	**0.9633** **
20	bbs_rich_cv	0.9532***	**0.8927** **	**0.9003** **	**0.9193** **	**0.9551** **
21	svr_linear_rich_cv	0.9629***	**0.9845*** **	**0.9643*** **	**1.0145** **	**1.0172** **
22	svr_rbf_rich_cv	**0.777** **	**0.7927***	**0.8645*** **	**0.8655** **	**0.9408*** **
23	b1_cv_boost	0.8885***	**0.8718** **	**0.9161** **	**0.9595** **	**0.9667***
24	b1_cv_ridge	**0.7636*** **	**0.9206*** **	**1.0253*** **	**1.0577*** **	**1.1372*** **
25	b1_cv_en	**0.7662*** **	**0.7716** **	**0.8377** **	**0.8544** **	**0.8993***
26	b1_cv_lasso	**0.711**	**0.7642***	**0.8371** **	**0.8509***	**0.8991***
27	b2_cv_boost	**0.7245***	**0.7585**	**0.8468** **	**0.8524***	**0.913***
28	b2_cv_ridge	**0.75*** **	**0.8416*** **	**0.9239*** **	**0.9428** **	**1.0069** **
29	b2_cv_en	**0.6855**	**0.7642***	**0.8371** **	**0.8509***	**0.8991***
30	b2_cv_lasso	**0.6853**	**0.7642***	**0.8371** **	**0.8509***	**0.8991***
31	b3_cv_boost	0.9414***	**0.8517** **	**0.8802** **	**0.8909** **	**0.9309** **
32	b3_cv_ridge	**0.7325** **	**0.8519*** **	**0.9370** **	**0.9607** **	**1.0152** **
33	b3_cv_en	**0.7681*** **	**0.7703** **	**0.8369** **	**0.8518** **	**0.8993***
34	b3_cv_lasso	**0.7106**	**0.7642***	**0.8371** **	**0.8509***	**0.8991***

Source: Author.

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

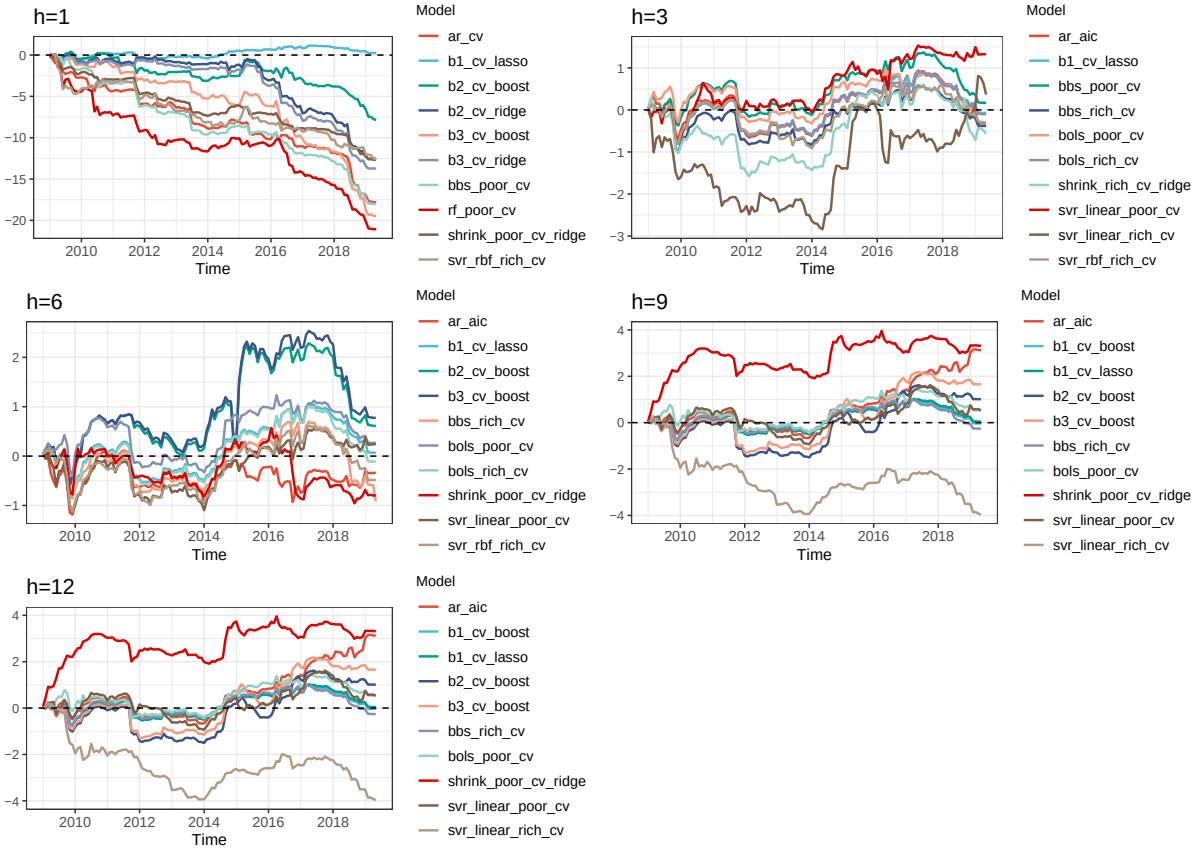
Table 7 – UNRATE: Relative RMSE Expanding Window

	model	h=1	h=3	h=6	h=9	h=12
Data-poor models						
1	ar_bic	1.0531	1.0144	0.8582	0.8669	0.8228
2	ar_aic	1.0041***	0.994***	1.0343***	**0.988*** **	**1.0283** **
3	ar_cv	0.8887***	0.9587***	1.0612***	**1.0223*** **	**1.0839*** **
4	shrink_poor_cv_ridge	0.7419	**0.7645**	**0.891**	**0.888**	**0.8936**
5	shrink_poor_cv_lasso	**0.6581**	**0.6808**	**0.8236**	**0.816**	**0.8582**
6	shrink_poor_cv_en	**0.6707**	**0.6989**	**0.8181**	**0.8127**	**0.8525**
7	rf_poor_cv	0.7675**	0.8008**	**0.9314***	**0.9241***	**0.9788** **
8	bols_poor_cv	0.8756***	0.9267***	**0.9368***	**0.9404** **	**0.9878** **
9	bbs_poor_cv	0.8974***	0.9166***	**0.9188**	**0.9186***	**0.9601**
10	svr_linear_poor_cv	1.006***	0.9616***	1.0361***	**1.0069*** **	**1.0135** **
11	svr_rbf_poor_cv	0.8255***	**0.8238** **	**0.9382***	**0.9934** **	**0.976*** **
Data-rich models						
12	ardi_bic	0.9743***	0.871***	**1.4186**	**2.9937**	**2.8748**
13	ardi_aic	1.201***	1.2916***	2.0418***	3.4212*	3.3586**
14	ardi_cv	0.9896***	1.0741***	1.2865***	**1.2196*** **	**1.0793** **
15	shrink_rich_cv_ridge	0.7526*	0.8058**	**1.0085** **	**1.048*** **	**1.0517*** **
16	shrink_rich_cv_lasso	**0.6581**	**0.6819**	**0.8234**	**0.816**	**0.8582**
17	shrink_rich_cv_en	**0.6619**	**0.7018**	**0.8164**	**0.8086**	**0.8499**
18	rf_rich_cv	**0.7283**	**0.7729***	**0.9288***	**0.928** **	**0.9599** **
19	bols_rich_cv	0.8672***	0.8775***	**0.9623***	**1.0428*** **	**1.1074*** **
20	bbs_rich_cv	0.8375***	0.8398***	**0.9839** **	**0.9613** **	**1.0182** **
21	svr_linear_rich_cv	0.9902***	1.0098***	1.0947***	**1.1906*** **	**1.114*** **
22	svr_rbf_rich_cv	**0.681**	**0.7395**	**0.8275**	**0.816**	**0.8899**
23	b1_cv_boost	0.9126***	0.9125***	**0.992***	**1.0665*** **	**1.1265*** **
24	b1_cv_ridge	**0.789** **	0.832**	**0.9581***	**1.0038*** **	**1.0409*** **
25	b1_cv_en	**0.6793**	**0.7043**	**0.8275**	**0.8238**	**0.8675**
26	b1_cv_lasso	**0.6544**	**0.6814**	**0.8236**	**0.816**	**0.859**
27	b2_cv_boost	0.7319	**0.7682** **	**0.8704**	**0.9292** **	**1.0035*** **
28	b2_cv_ridge	**0.6788**	**0.7105**	**0.8392**	**0.8296**	**0.8651**
29	b2_cv_en	**0.6717**	**0.6961**	**0.8236**	**0.8167**	**0.8584**
30	b2_cv_lasso	**0.6717**	**0.6972**	**0.8236**	**0.816**	**0.8582**
31	b3_cv_boost	0.7426	**0.7983** **	**0.8889**	**0.9417** **	**0.9984** **
32	b3_cv_ridge	0.6718	**0.7054**	**0.8408**	**0.8344**	**0.8696**
33	b3_cv_en	**0.657**	**0.6826**	**0.826**	**0.8199**	**0.8553**
34	b3_cv_lasso	**0.6647**	**0.6943**	**0.8236**	**0.816**	**0.8582**

Source: Author.

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

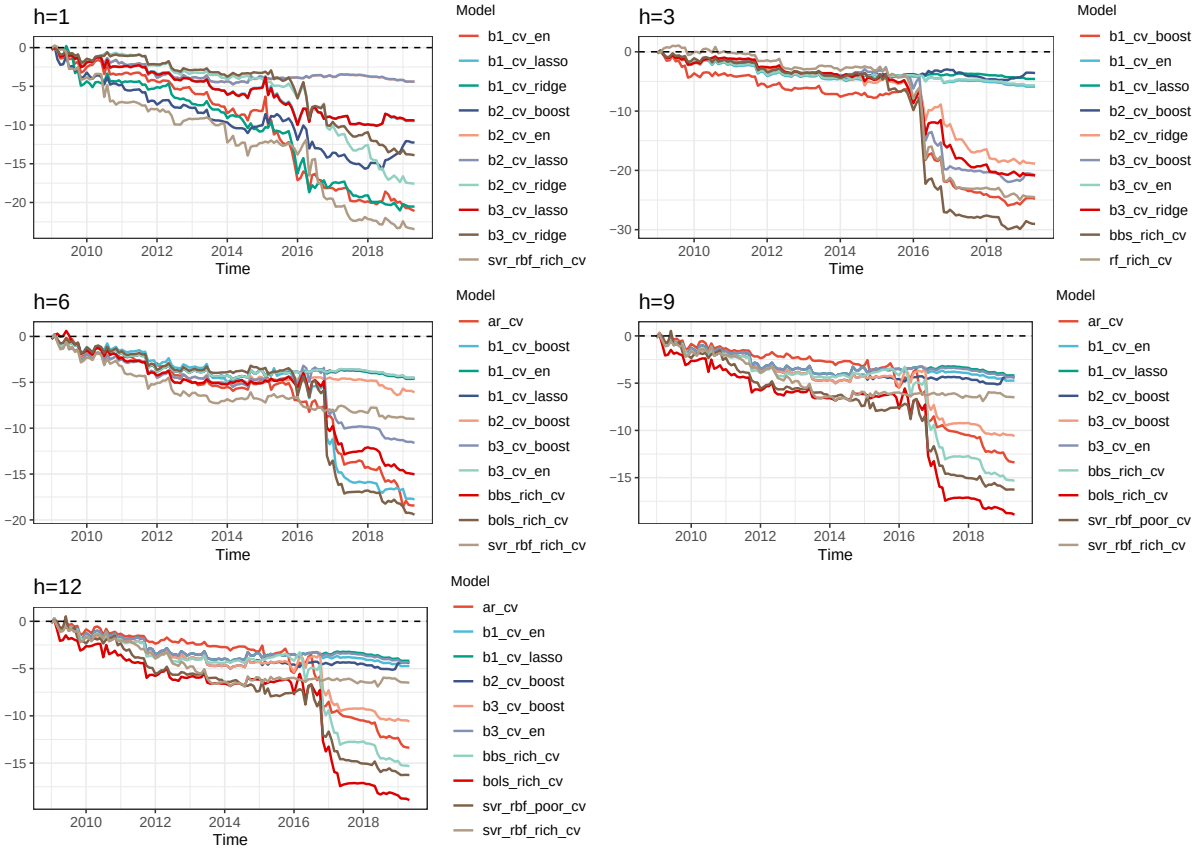
Figure 4 – CSFE EMBI+ Expanding Window



Source: Author.

Note: In the plots, we report only 10 models that were inside the Model Confidence Set (MCS) to improve the data visualization.

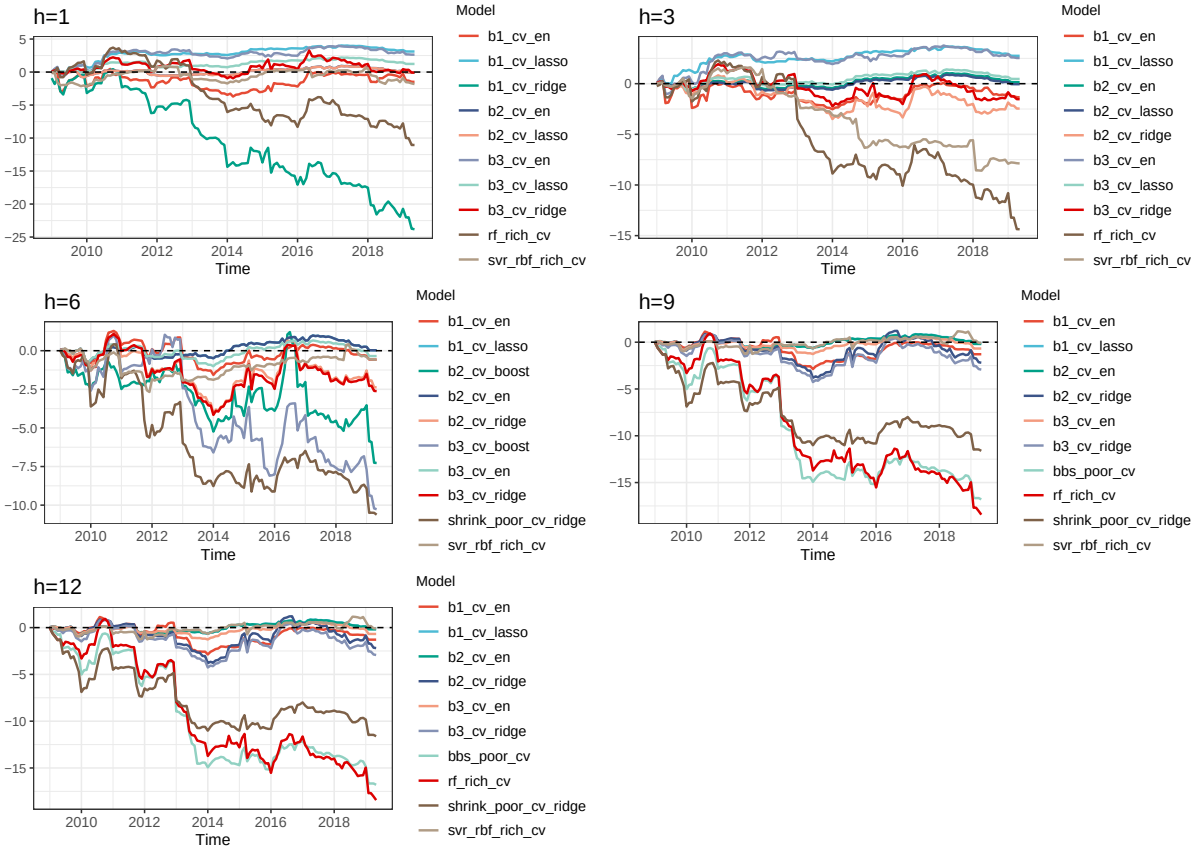
Figure 5 – CSFE IPCA Expanding Window



Source: Author.

Note: In the plots, we report only 10 models that were inside the Model Confidence Set (MCS) to improve the data visualization.

Figure 6 – CSFE UNRATE Expanding Window



Source: Author.

Note: In the plots, we report only 10 models that were inside the Model Confidence Set (MCS) to improve the data visualization.