Machine learning methods for inflation forecasting in Brazil: new contenders versus classical models*

Gustavo Silva Araujo †

Wagner Piazza Gaglianone ‡

March 4, 2022

Abstract

In this paper, we explore machine learning (ML) methods to improve inflation forecasting in Brazil. An extensive out-of-sample forecasting exercise is designed with multiple horizons, a large database of 501 series, and 50 forecasting methods, including new machine learning techniques proposed here, traditional econometric models and forecast combination methods. We also provide tools to identify the key variables to predict inflation, thus help opening the ML black box. Despite the evidence of no universal best model, the results indicate that machine learning methods can, in numerous cases, outperform traditional econometric models in terms of mean-squared error. Moreover, the results indicate the existence of nonlinearities in the inflation dynamics, that are relevant to forecast inflation. The set of top forecasts often include forecast combinations, tree-based methods (such as random forest and xgboost), breakeven inflation, and survey-based expectations. Altogether, these findings offer a valuable contribution to macroeconomic forecasting, especially, focused on Brazilian inflation.

Keywords: Machine Learning; Big Data; Inflation Forecasting.

JEL Classification: C14; C15; C22; C53; C55; E17; E31.

*The views expressed in the paper are those of the authors and do not necessarily reflect those of the Banco Central do Brasil. We are especially grateful for the helpful comments given by Marcelo Aragão, Jorge Henrique Barbosa, Rafael Cusinato, Serafín Martínez-Jaramillo, Fabio Kanczuk, Vicente Machado, Marcelo Medeiros, Euler de Mello, and André Minella. We also benefited from comments given by the seminar participants at the 18th ESTE - Time Series and Econometrics Meeting (Gramado, September 2019), IV Workshop da Rede de Pesquisa do Banco Central do Brasil (Brasília, November 2019), CEMLA XXV Meeting of the Central Bank Researchers Network (October 2020), and Experiências em Data Science no Coaf e no BCB (March 2021).

†Research Department, Banco Central do Brasil, and Ibmec-RJ. E-mail: gustavo.araujo@bcb.gov.br

‡Corresponding author. Research Department, Banco Central do Brasil, and FGV Crescimento & Desenvolvimento. E-mail: wagner.gaglianone@bcb.gov.br

1 Introduction

Producing reliable inflation forecasts is a constant challenge for policymakers and of greatest importance to economic agents and their investment decisions. Inflation adds uncertainty to investment decisions and shortens the investment horizon, especially in emerging markets, making the construction of accurate forecasts a relevant issue in such economies.

Building accurate forecasts is generally not an easy task, since it requires an approach complex enough to incorporate relevant variables but also focused on excluding irrelevant data. In this sense, machine learning (ML) methods, in general, are able to identify nonlinear patterns in the data, hidden to standard linear models, thus offering an alternative (and compelling) approach to traditional econometric models.

Moreover, despite the low frequency of data in macroeconomics, where the usual variables of interest are collected on an annual, quarterly or monthly basis (leading to much less data accumulation compared, for instance, to an intraday high-frequency database) and the usual split of data into training and test sets (in-sample and out-of-sample, reducing still further the amount of data used for model estimation), there is high incentive to use ML methods in applied macroeconomics, since a lot of these methods can deal with large amounts of data (*big data*), in contrast to linear econometric models usually based on a few variables, besides the continuous improvement in computer technology, allowing to run ML algorithms at much faster speed.

The objective of this paper is to forecast Brazilian inflation based on a large number of macroeconomic and financial variables. Our goal is to assess whether machine learning approaches can indeed offer improvement to forecast accuracy in applied macroeconomics and make a contribution to the standard statistical toolkit used in macro forecasting.

To do so, we conduct an extensive *horse-race* (pseudo out-of-sample forecasting exercise) across 50 models (or methods) to forecast inflation in Brazil at multiple horizons (ranging from 1 up to 18 months). The list of competing methods includes several machine learning methods, based on regularization approaches (elastic net, lasso, adaptive lasso, ridge) or regression trees (random forest, quantile regression forest, xgboost), as well as traditional econometric approaches (ARMA, VAR, factor models), reduced-form structural models (Phillips curves), survey-based forecasts (Focus), breakeven inflation (BEI) from financial market data, among many others. Our database covers 501 time series, coming from 167 macroeconomic and financial variables used to build high-dimensional models.

The literature on macroeconomic forecasting using machine learning methods is relatively new. For instance, see Medeiros et al. (2016) and Garcia et al. (2017) for applications with Brazilian data; Cheng et al. (2019) for aggregating individual survey-based forecasts, using machine learning tools, to improve forecasting of the U.S. inflation; Kohlscheen (2021) for an investigation of the drivers of inflation in 20 advanced countries using random forest; and Costa et al. (2021) for oil price point and density forecasting using machine learning methods.

Our research contributes to this fast-growing literature in five ways: (i) the first original contribution is to propose a new *quantile-combination method*, based on the quantile regression forest model of Meinshausen (2006). The idea is to use information of the conditional distribution from a set of estimated conditional quantiles to build an improved conditional mean forecast; (ii) the second contribution is to employ a *hybrid machine learning* approach, inspired by the work of Medeiros et al. (2021), to build new ML methods. The goal is to disentangle the forecast accuracy due to variable selection from possible nonlinearities in the data-generating process; (iii) the third contribution is to use machine learning not only as forecast method, but also as a *forecast combination* device. The idea is to check whether ML methods can beat traditional forecast combination approaches when combining a given set of point forecasts; (iv) the fourth contribution is to provide a simple way to build *fan charts* from ML-based inflation forecasts, where a measure of uncertainty can be attached to the forecasted inflation-path based on past forecast errors; and (v) the final contribution is to help opening the ML *black box*,[1] by providing a set of auxiliary graphs[2] to further analyze the performance of competing methods far beyond the usual accuracy analysis based on mean-squared forecast error.

The outline of the paper is as follows. In Section 2, we present the methodology comprising machine learning methods and traditional econometric models to predict inflation. Section 3 presents an out-of-sample empirical exercise, where competing methods are used to forecast Brazilian inflation, in the same spirit of Medeiros et al. (2016).[3] Section 4 concludes. The Appendix presents further details on the methodology and additional empirical results.

[1] The *black box* term applied to ML techniques has been around for years now. It is often employed to critisize neural networks' lack of explainability.

[2] The set of auxiliary graphs comprises: bias-variance decomposition, cumulative squared prediction error, word clouds, and variable (or feature) importance.

[3] Compared to previous papers focused on Brazilian inflation forecasting, we considered: (i) a broader set of models (proposing new ML methods and forecast combination tools); (ii) a larger database of macroeconomic and financial variables, including new potential predictors; (iii) longer forecast horizons (from one up to 18 months); and (iv) inflation rates measured either by monthly or twelve-month accumulated rates.

2 Methodology

2.1 Machine learning in a nutshell

Machine learning is a branch of artificial intelligence, often described as the art and science of pattern recognition. It is essentially a data-driven approach, with mild assumptions about the underlying statistical relationships in the data, and entails a large variety of methods. It usually comprises two core elements, a learning method and an algorithm, enabling one to automate as many of the modeling choices as possible in a manner that is not subject to the discretion of the forecaster (Hall, 2018).

Most traditional forecasting methods rely on fitting data to a pre-specified relationship between dependent and independent variables, thus assuming a specific functional and stochastic process. In contrast, a different approach to statistical analysis and forecasting, in particular, is offered by machine learning, which is to a great extent a data-driven approach, since it makes almost no assumption about the underlying statistical relationship in the data.

According to Samuel (1959), machine learning is the ability of computers to learn from experience without being explicitly programmed. Cerulli and Drago (2021) points out that ML places itself at the intersection between statistics, computer science, and artificial intelligence. According to these authors: *"The primary objective of ML is turning information into knowledge and value by 'letting the data speak'"*. Hansen (2019) explains that ML is: *"a new and somewhat vague term, but typically is taken to mean procedures which are primarily used for point prediction in settings with unknown structure. Machine learning methods generally allow for large sample sizes, large number of variables, and unknown structural form."*

In fact, machine learning encompasses a wide variety of models, but often comprises two core elements: a *learning method*, where data is used to determine the best fit for the input variables, and an *algorithm* which captures the relationship between the input and output. In general, ML can be categorized into three types (see Jung et al., 2018):

(i) *supervised learning*, where the dependent variables are clearly identified, even if the specific relationships in the data are not known (e.g., linear regression, logistic regression);

(ii) *unsupervised learning*, where there is no specific output defined beforehand, and the goal is to recognize data patterns and determine output classification categories (e.g., cluster analysis, principal components); and

(iii) *reinforcement learning*, which iteratively search for an optimal location of the input variables that yield the highest reward, that is, maximize a reward function using no training

set (e.g., sarsa, Q-learning).

According to Varian (2014), the growing amounts of data and ever complex-growing relationships warrant the usage of machine learning approaches in economics. In this paper, we build inflation forecasts using different machine learning (*supervised*) algorithms: based either on penalized-regression models (e.g., ridge regression, lasso, adaptive lasso, and elastic net) or on tree-based methods (e.g., random forest, quantile regression forest, and XGBoost).

The first approach entails regularization techniques that introduce penalties for *overfitting* the data.[4,5] For example, the elastic net model mixes two different kinds of regularization, by penalizing both the number of variables in the model and the extent to which any given variable contributes to the model's forecast. By applying these penalties, the elastic net *learns* which variables are most important, eliminating the need for researchers to make discretionary choices about which variables to include.

The second (tree-based) approach is nonparametric, based on the recursive binary partitioning of the covariate space, which can deal with very large number of explanatory variables, thus producing highly nonlinear predicted models.

2.2 Models (or methods) to forecast inflation

There is a variety of approaches in the literature to model the inflation dynamics. According to Ang et al. (2007), economists use four main methods to forecast inflation: time-series models, structural models (e.g., Phillips curve), asset price models (e.g., term-structure of interest rates), and methods that employ survey-based measures (e.g., survey of professional forecasters).

In this paper, inflation forecasts come from 50 forecasting methods listed in Table 1. Besides some traditional econometric approaches to forecast inflation, such as ARMA and VAR models, this paper considers Phillips curves, well-known in the macro literature (e.g., Stock and Watson, 1999), survey-based inflation expectations and inflation forecasts embedded in financial market data (breakeven inflation). The set of forecasting methods also includes many nonlinear machine learning methods, based on regularization procedures or regression trees, and several forecast combination techniques.

[4] In statistics, *overfitting* denotes the production of an analysis, which is assumed to be valid for the entire population (for instance, an estimated input-output relationship), that corresponds too closely to a particular set of data, but it may fail to fit additional data, or reliably forecast future observations. In other words, when the model learned too well the training sample and showed low prediction capability out-of-sample, it is said that this overfitted.

[5] According to Hall (2018), ML algorithms usually deliver a model complex enough to avoid underfitting the data but not so complex to overfit it.

Table 1 - Models (or methods) to forecast inflation

1	Random Walk	26	Hybrid Random Forest - Adalasso
2	Random Walk (Atkeson-Ohanian)	27	Hybrid Random Forest - XGBoost
3	ARMA	28	Inflation Expectations (Breakeven)
4	VAR	29	Inflation Expectations (Focus Survey)
5	Phillips Curve (Backward)	30	Combination 1 (Mean)
6	Phillips Curve (Hybrid)	31	Combination 1 (Median)
7	Factor Model 1	32	Combination 1 (Granger-Ramanathan)
8	Factor Model 2	33	Combination 1 (Constrained Least Squares)
9	Factor Model 3	34	Combination 1 (Complete Subset Regression)
10	Factor Model 4	35	Combination 1 (Adalasso)
11	Elastic Net	36	Combination 1 (Random Forest)
12	LASSO	37	Combination 2 (Mean)
13	Adaptive LASSO (Adalasso)	38	Combination 2 (Median)
14	Ridge Regression	39	Combination 2 (Granger-Ramanathan)
15	Random Forest	40	Combination 2 (Constrained Least Squares)
16	Quantile Regression Forest	41	Combination 2 (Complete Subset Regression)
17	XGBoost	42	Combination 2 (Adalasso)
18	Recurrent Neural Network (RNN)	43	Combination 2 (Random Forest)
19	Disaggregated Inflation (ARMA)	44	Combination 3 (Mean)
20	Disaggregated Inflation (Adalasso)	45	Combination 3 (Median)
21	Disaggregated Infl. (Random Forest)	46	Combination 3 (Granger-Ramanathan)
22	Hybrid Adalasso - OLS	47	Combination 3 (Constrained Least Squares)
23	Hybrid Adalasso - Random Forest	48	Combination 3 (Complete Subset Regression)
24	Hybrid Adalasso - XGBoost	49	Combination 3 (Adalasso)
25	Hybrid Random Forest - OLS	50	Combination 3 (Random Forest)

Notes: Combination 1 is based on models 1-27. Combinations 2 and 3 are based on the superior models of the *model confidence set* of Hansen et al. (2011), considering models 1-27 or 1-29, respectively.

The list of models, of course, is not an exhaustive list, since more complex models could always be included. Nonetheless, the set of inflation forecasting methods listed in Table 1 is a good starting point to compare the accuracy of traditional econometric approaches with some new competing machine learning techniques.

Our main goal here is to forecast the inflation rate y_{t+h} at period $t+h$ using the information set available at period t. In this sense, inflation is modeled as a function of a set of predictors $\widetilde{x}_t$, measured at time t, as follows:

$$y_{t+h} = \Upsilon_h\left(\widetilde{x}_t\right) + \varepsilon_{t+h}, \tag{1}$$

where $\Upsilon_h\left(\cdot\right)$ is a possibly nonlinear mapping of a set of predictors (a single model or an ensemble of different specifications), ε_{t+h} is the forecasting error, and predictors $\widetilde{x}_t$ may include weakly exogenous predictors, lagged values of inflation and a number of factors computed from a large number of potential covariates; see Garcia et al. (2017).

Here, we consider $\widetilde{x}_t = \{\mathbf{1}_t, x_t, x_{t-1}, \ldots, x_{t-s}\}'$, where $\mathbf{1}_t$ is a constant term, $x_t = \{x_{1,t}, \ldots, x_{n,t}\}$ is a set of n predictors, and s is the maximum lag adopted for the set of variables x_t when forming the vector of variables $\widetilde{x}_t$.

To build our forecasting exercise, we split the sample in three consecutive time sub-periods, where time is indexed by $t = 1, 2, ..., T_1, ..., T_2, ..., T$. The first sub-period ($t = 1, ..., T_1$), usually called *estimation sample*, is used for model estimation and forecast inflation y_t in the subsequent periods.[6]

In the second sub-period ($t = T_1 + 1, ..., T_2$), realizations of y_t are confronted with forecasts produced in the estimation sample, and forecast combination weights are estimated, if that is the case. The first and second sub-periods, together, are labeled as *training set*. The final sub-period, also known as *test set*, is where genuine out-of-sample forecast is entertained, comprising the last P observations of the sample ($t = T_2+1, ..., T$). Thus, we have $P = T - T_2$ observations to compare forecasts and compute accuracy measures.

For the regularization approaches considered in this paper (e.g., elastic net), the mapping $\Upsilon_h\left(\cdot\right)$ is linear, such that:

$$y_{t+h} = \widetilde{x}_t'\beta_h + \varepsilon_{t+h}, \tag{2}$$

where β_h is a vector of unknown parameters. The inflation forecast from the linear ML ap-

[6] As will be discussed in section 3.1, we use a recursive estimation scheme (i.e., increasing sample size).

proach, $f^{ML}_{y_{T_2+h}}$, using a sample of $t = 1, ..., T_2$ observations, is given by:

$$f^{ML}_{y_{T_2+h}} = \widetilde{x}'_{T_2}\widehat{\beta_h}, \quad \text{for } h = 1, ..., H. \tag{3}$$

To evaluate forecast $f^{ML}_{y_{T_2+h}}$, we compute the respective mean-squared error as follows: $MSE_h = \frac{1}{P}\sum_{t=T_2+1}^{T}\left(y_t - f^{ML}_{y_{T_2+h}}\right)^2$. Note that we adopt the *direct forecast* approach, where the inflation h periods ahead (y_{T_2+h}) is modeled as a function of a set of predictors $\widetilde{x}'_{T_2}$ measured at time T_2. In other words, for each horizon h, we estimate a different vector of unknown parameters β_h (in contrast to the iterated multistep approach; see Marcellino, Stock and Watson, 2006). Thus, we avoid the necessity of estimating a model for the time-evolution of $\widetilde{x}_t$.

2.2.1 Elastic Net, Lasso, Adaptive Lasso and Ridge Regression

Elastic net: It is a regularization method proposed by Zou and Hastie (2005),[7] which simultaneously does automatic variable selection and continuous shrinkage, and it can select groups of correlated variables. The elastic net encourages a grouping-effect, where highly correlated regressors tend to be jointly included (or excluded) from the model, and it can be particularly useful when the number of predictors k is high compared to the number of observations T. For a nonnegative shrinkage parameter λ, and a combination parameter α strictly between 0 and 1, the elastic net solves the following problem:

$$\widehat{\beta} = \underset{\{\beta_1,...,\beta_k\}}{\arg\min}\left(\frac{1}{T}\sum_{t=1}^{T}\left(y_t - \sum_{j=1}^{k}x'_{j,t}\beta_j\right)^2 + \lambda P_\alpha(\beta)\right), \tag{4}$$

where

$$P_\alpha(\beta) = \sum_{j=1}^{k}\alpha\left|\beta_j\right| + \frac{(1-\alpha)}{2}\beta_j^2, \tag{5}$$

and β is the $k \times 1$ vector of parameters, y_t is the dependent variable, and $\{x_{1,t}, \ldots, x_{k,t}\}$ is the $k \times 1$ vector of regressors. The tuning parameter λ controls the overall strength of the penalty term $P_\alpha(\beta)$, which interpolates between the l_1-norm of β and the squared l_2-norm of β. Note that by setting $\lambda = 0$, elastic net becomes the ordinary least squares (OLS) regression. Also note the objective function has no-closed form solution, but it is convex and can be minimized using any convex optimization method such as gradient or coordinate descent.[8]

[7] According to the authors: *"It is like a stretchable fishing net that retains 'all the big fish'."*

[8] Although we defined the elastic net by using (λ, α), this is not the only choice as the tuning parameters; see Zou and Hastie (2005). For example, one could use the l_1-norm of the coefficients or the fraction of the l_1-norm

Lasso: The least absolute shrinkage and selection operator (*lasso*) was proposed by Tibshirani (1996). The core idea is to shrink to zero the irrelevant coefficients. The lasso is a penalized least squares method imposing an l_1-penalty on the regression coefficients, which allows lasso to do continuous shrinkage and automatic variable selection simultaneously. Also, it is a particular case of the elastic net estimator (4), considering $\alpha = 1$ in penalty term (5).

According to Cheng et al. (2019), lasso is "*the most intensively studied statistical method in the past 15 years*". Indeed, it has shown success in many practical situations, since it can handle more variables than observations. Nonetheless, it has limitations and might even become an inappropriate variable selection method in some cases. Zou and Hastie (2005) lists a few examples: (i) when the number of predictors k is greater than the number of observations T, lasso selects at most T variables before it saturates, due to the nature of the convex optimization problem; (ii) in the case of *grouping effect*[9] lasso tends to select only one variable from the group; (iii) when $T > k$ and with high correlated predictors, the ridge regression tends to perform better than lasso.

Adaptive Lasso: Zou (2006) shows that lasso is inconsistent for variable selection under certain circumstances, and proposes the adaptive lasso (or *adalasso*), where adaptive weights are used for penalizing different coefficients in the l_1-penalty. According to the author, the adaptive lasso enjoys the oracle properties (i.e., it performs as well as if the true underlying model were known) and does not select useless variables that may damage the forecasting accuracy. The core idea behind the model is to use previously known information to select the variables more efficiently.[10]

In practice, it is a two-step estimation that first generates different weights w_j for each candidate variable $x_{j,t}$, which are used in the second-step (lasso estimation) as additional information. The *adalasso* estimator is defined as:

$$\widehat{\beta} = \underset{\{\beta_1,\ldots,\beta_k\}}{\arg\min} \left(\frac{1}{T}\sum_{t=1}^{T}\left(y_t - \sum_{j=1}^{k} x'_{j,t}\beta_j \right)^2 + \lambda \sum_{j=1}^{k} w_j \left|\beta_j\right| \right), \tag{6}$$

where $w_j = \left|\widehat{\beta}_j^*\right|^{-\tau}$ represents the weights; $\widehat{\beta}_j^*$is a parameter estimated in the first-step, and $\tau > 0$ is an additional tuning parameter that determines how much one wants to emphasize the

to parameterize the elastic net. See Appendix A for further details on the choice of tuning parameters (λ, α).

[9] The grouping effect occurs if the regression coefficients of a group of highly correlated variables tend to be equal (up to a change of sign if negatively correlated).

[10] According to Medeiros and Mendes (2016), the conditions required by the *adalasso* estimator are very general, and the model works even when the errors are non-Gaussian, heteroskedastic, and the number of variables increases faster than the number of observations.

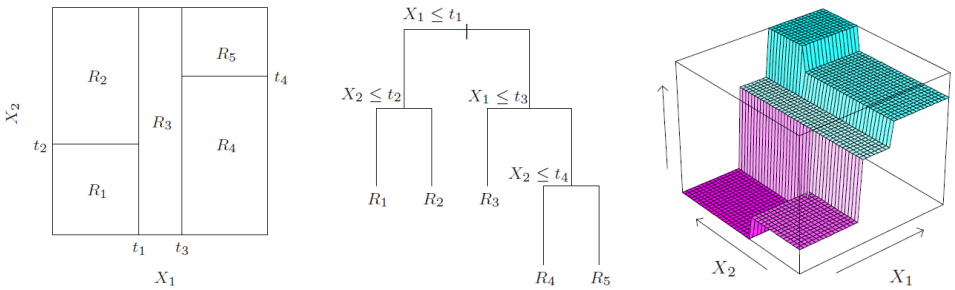
difference in the weights. In general, τ is set to unity and $\widehat{\beta}_j^*$ is the respective lasso coefficient estimated in the first-step.[11]

Ridge Regression: In contrast to lasso, the ridge regression (Hoerl and Kennard, 1970) minimizes the squared sum of the residuals subject to a bound on the l_2-norm of the parameters. It is a particular case of the elastic net estimator (4), considering $\alpha = 0$ in penalty term (5). Since ridge is a continuous shrinkage method, in some cases it can achieve better out-of-sample performance through a *bias-variance* trade-off (that is, using regularization to balance the forecast errors due to bias and variance). In particular, ridge is good at improving the OLS counterpart when multicollinearity is present. However, ridge cannot be used for variable selection (and to produce a parsimonious model), since it retains all regressors in the model, that is, it only shrinks the coefficients close (but never equal) to zero.

2.2.2 Random Forest

A random forest (RF) is a collection of decision trees, introduced as a machine learning tool in Breiman (2001). It is a very popular and powerful method used in high-dimensional regression or classification. The main idea is to reduce the forecast variance by using bootstrap aggregation (*bagging*) of randomly constructed trees. A *regression tree* is a nonparametric model based on the recursive binary partitioning of the covariate space X. According to Garcia et al. (2017), the model is usually displayed in a graph, which has the format of a binary decision tree with parent nodes (or split nodes) and terminal nodes (also called leaves; which represent different partitions of X).[12] Figure 1 shows an example of a regression tree with two covariates.

Figure 1 - Example of a recursive binary splitting in a regression tree



Notes: The left graph shows the partition of a two-dimensional covariate space. The center graph displays the corresponding tree, whereas the right graph shows the prediction surface. Source: Hastie et al. (2009).

[11] In our empirical exercise, we adopt $\tau = 0.3$, following Medeiros et al. (2016). Alternative adalasso models with $\tau = 1$ or 2, in general, generate less accurate forecasts in terms of MSE.

[12] According to the authors, the partitions are often defined by a set of hyperplanes, each of which is orthogonal to the axis of a given predictor variable (also called the split variable).

Note that we first split the covariate space into two regions ($X_1 \leq t_1$ and $X_1 > t_1$)[13] and model the dependent variable by the mean of Y in each region. The selected variable (X_1) and the corresponding split-point (t_1) are chosen in order to achieve the best fit. Then, one (or both) of these regions are split into two more regions, and this process is continued until some stopping rule is applied. In the example shown in Figure 1, the regression tree model predicts Y with the constant c_m in region R_m, $m = 1, ..., 5$, as follows (see Appendix B for more details):

$$\mathbb{E}_{\text{regression tree}}\left(Y \mid (X_1, X_2)\right) = \sum_{m=1}^{5} c_m \mathbf{1}_{\{(X_1,X_2)\in R_m\}}. \tag{7}$$

In practice, one major problem with regression trees is their high prediction variance. Usually, a small change in the data leads to a very different sequences of splits. The main reason for such instability is the hierarchical nature of the algorithm (the effect of a big error in the top split is propagated down to all of the splits below it).[14]

To overcome this issue, one can employ the *bagging* (bootstrap aggregation) method, which consists on fitting the same tree several times to bootstrap-sampled versions of the training data and, then, average the result. This approach often improves model performance because it decreases the forecast variance without increasing too much the bias.[15]

Random forest uses a modified-bagging method (*random subspace projection*) that selects a random *subset* of covariates at each candidate split. The reason for doing this is the correlation of trees in the ordinary bootstrap: if few covariates are strong predictors for the dependent variable, such covariates will be selected in many of the bootstrapped trees, causing them to be correlated. According to Hansen (2019), the modification adopted in RF aims to *decorrelate* the bootstrap trees by introducing extra randomness.[16]

[13] Rather than splitting each node into just two groups, one might consider multiple splits at each stage. However, according to Hastie et al. (2009, p.311), this is not a good strategy, since multiple splits fragment the data too quickly, leaving insufficient data at the next level down.

[14] According to Hastie et al. (2009), regression trees tend to learn highly irregular data patterns and overfit their training sets, thus producing low bias but high variance. To reduce variance, RF averages multiple trees, trained on different parts of the training set. This often generates a small forecast bias, but generally improves the forecast accuracy.

[15] Training many trees on a single training set would give strongly correlated trees, whereas bootstrap sampling helps decorrelating the trees by showing them different training sets.

[16] This way, forecast variance can be reduced by two ways: (i) in each node, the variable being split is selected from a random subset of variables (instead of the full set); and (ii) each tree is learned on a bootstrapped subsample. See Hastie et al. (2009) for further details.

2.2.3 Quantile-combination method based on QRF

Random forest approximates the conditional mean of Y by constructing a weighted average over the sample observations of Y. Nonetheless, random forests can also provide information about the full conditional distribution of the response variable, not only about the conditional mean. This information can be used, for instance, to build prediction intervals and account for outliers in the data. This way, conditional quantiles can be inferred with quantile regression forests (QRF), a generalization of random forest proposed by Meinshausen (2006).[17]

The idea here is to provide a nonparametric way of estimating conditional quantiles for a high-dimensional set of predictor variables. According to the author, the QRF algorithm is shown to be consistent and competitive in terms of predictive power. First, recall the conditional quantile of Y, given X, at quantile level τ, is defined by:

$$Q_\tau\left(Y \mid X\right) = \inf\{y : F(y \mid X) \geq \tau\} \tag{8}$$

or, equivalently,

$$F\left(y \mid X\right) = \Pr\left(Y \leq y \mid X\right) = \mathbb{E}\left(I_{\{Y \leq y\}} \mid X\right), \tag{9}$$

where $F(y \mid X)$ is the conditional cumulative distribution function (CDF) and $I_{\{Y \leq y\}}$ is an indicator function. Note the probability of Y being smaller than $Q_\tau\left(\cdot\right)$ is equal to τ. Next, we approximate the CDF by the weighted average of $I_{\{Y \leq y\}}$ over n observations, as follows:

$$\widehat{F}\left(y \mid X\right) = \sum_{i=1}^{n} w_i(x) I_{\{Y_i \leq y\}}, \tag{10}$$

using the same weights $w_i(x)$ defined in Appendix B for random forest.[18] This way, estimates of the conditional quantiles $\widehat{Q}_\tau\left(\cdot\right)$ can be obtained by plugging $\widehat{F}\left(\cdot\right)$, instead of $F\left(\cdot\right)$ into (8).

Now, we go one step further, by relating the conditional quantiles with the conditional mean of Y. This could be accomplished by integrating the conditional quantile function of Y over the entire domain $\tau \in [0, 1]$; see Koenker (2005, p.302). The conditional mean $\mathbb{E}\left(Y \mid X\right)$ can,

[17] The main difference between QRF and RF is that for each node, RF keeps only the mean of the observations that fall into this node (and neglects all other information). In contrast, QRF keeps the value of all observations in this node and assesses the conditional distribution based on this full information.

[18] See Appendix C for a summary of the algorithm used to compute the CDF estimate in QRF.

thus, be approximated[19] by a sum of estimated conditional quantiles, as follows:[20]

$$\mathbb{E}\left(Y \mid X\right) = \int_0^1 Q_\tau\left(Y \mid X\right) d\tau = \lim_{P\to\infty}\left(\sum_{p=1}^{q} \widehat{Q}_{\tau_p}\left(Y \mid X\right)\Delta\tau_p\right). \tag{11}$$

The idea is to aggregate information from different conditional quantiles in order to achieve an improved conditional mean. The approximation of the conditional mean by a combination of conditional quantiles is not a novel approach in the literature. Indeed, it has a long tradition in statistics (see Judge et al., 1988) and has been previously applied in the forecasting literature. Nonetheless, our original contribution is to propose a new quantile-combination approach, based on quantile regression forest, to build conditional mean forecasts through equations (8), (10) and (11).

The approach proposed here follows the spirit of the averaging scheme applied to quantiles conditional on predictors selected by *lasso*, proposed by Lima and Meng (2017), and of Jiang et al. (2020) which shows that aggregating information over different quantiles can produce superior forecasts for stock return prediction.

The advantage of such approaches relies on the fact that quantiles are robust to outliers (in our case, extreme unanticipated inflationary shocks), which potentially improves forecast accuracy and likely impact the performance of standard models, usually designed to account for average responses. To sum it up, we propose the following three-step algorithm:

1. Choose a finite (equidistant) grid of quantile levels. For example: $\Gamma \equiv [0.05, 0.10, \ldots, 0.95]$;

2. For each $\tau \in \Gamma$, period t, forecast horizon h, and information set $\mathcal{F}_t$, estimate the conditional quantile $\widehat{Q}_\tau\left(y_{t+h} \mid \mathcal{F}_t\right)$ using the QRF method of Meinshausen (2006);

3. Compute the average of $\widehat{Q}_\tau\left(y_{t+h} \mid \mathcal{F}_t\right)$ across all $\tau \in \Gamma$, and consider it as proxy for $\mathbb{E}\left(y_{t+h} \mid \mathcal{F}_t\right)$, that is, as our (QRF-based) inflation forecast.

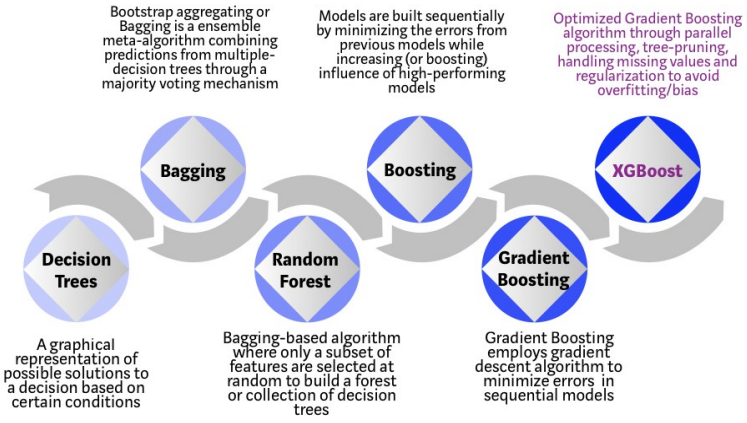
[19] By applying the second fundamental theorem of calculus (or the Newton-Leibniz axiom) on the sum of quantiles, the Riemann integral is obtained in the limit $P \to \infty$ (see Apostol, 1967) and the partitions $\Delta\tau_p = \frac{1}{P+1}$ get finer (i.e., $\Delta\tau_p \to 0$ as long as $P \to \infty$).

[20] We rely on the fact that the conditional quantiles are consistenly estimated using the QRF approach.

2.2.4 XGBoost

Extreme Gradient Boosting (or XGBoost) is a decision tree-based ensemble algorithm that uses a gradient boosting setup, as proposed by Chen and Guestrin (2016). It improves upon the previous gradient boosting setups through systems optimization and algorithmic enhancements. According to Morde and Setty (2019), the XGBoost algorithm has the best combination of prediction performance and processing time compared to other algorithms. Figure 2 shows a brief comparison of the most common decision tree algorithms.

Figure 2 - Algorithms for decision trees



Source: Morde and Setty (2019). Boosting is an *ensemble* technique (that is, makes an average of the predictions of a group of models) that constructs models sequentially, and each subsequent model corrects the errors of the previous one, whereas *bagging* constructs models independently.

Thus, XGBoost is a bagging-based algorithm with a key difference wherein only a subset of features is selected at random. Compared to Random Forest, XGBoost is normally used to train gradient-boosted decision trees and other gradient boosted models, whereas RF uses the same model representation and inference (as gradient-boosted decision trees), but a different training algorithm. Moreover, XGBoost supports *missing values*, since branch directions for missing values are learned during training.

In practice, it requires the right configuration of the algorithm for a dataset by tuning the *hyper-parameters*. Most of them are related to the bias-variance trade-off. When one allows the model to get more complicated (e.g., more depth), the model has better ability to fit the training data (in-sample), resulting in a less biased model. However, such complicated model requires more data. The best model should trade the model complexity with its predictive power carefully. See Chen and Guestrin (2016) for further details.[21]

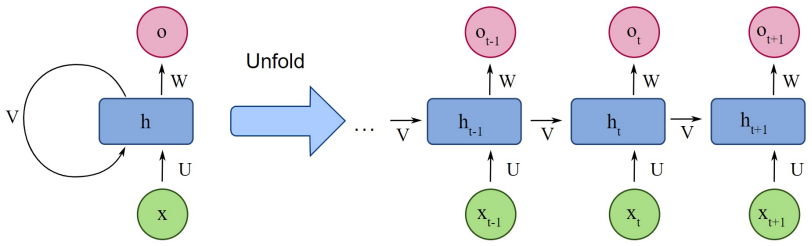
[21] For more details, see also https://towardsdatascience.com/a-guide-to-xgboost-hyperparameters-87980c7f44a9 and https://xgboost.readthedocs.io/en/latest/index.html

2.2.5 Recurrent Neural Networks (RNN)

A recurrent neural network (Elman, 1990) is a class of artificial neural networks commonly used for time series data. RNNs are highly *nonlinear* models that use training data to learn and represent complex dynamic relationships between variables. They are distinguished by their internal state (memory), as they make use of sequential information to capture long-term temporal dependencies between input variables and the output (dependent variable). While traditional deep neural networks assume inputs and outputs are independent of each other, the RNN output depends on the prior elements within the sequence and, thus, can exhibit dynamic temporal behavior.[22] See Tealab (2018) and Dupond (2019) for further details.

Figure 3 shows a basic scheme of the RNN, where x_t is the input vector of predictors, o_t is the output vector (dependent variable), h_t is the hidden layer (a set of neurons) and W and U are model parameter matrices. The left graph illustrates the recurrent connections as the arc labeled "V". The graph on the right unfolds the RNN across time, pointing out that RNN is a class of neural network that exhibits temporal dynamic behavior.

Figure 3 - Basic representation of the RNN



Source: https://en.wikipedia.org/wiki/Recurrent_neural_network

The RNN architecture is essentially driven by the number of *hidden layers* h_t, which controls the overall model complexity[23], and the *activation function* (or basis function), that determines whether a given neuron in the network should be activated. These nonlinear functions typically convert the output of a given neuron to a value between 0 and 1 (or -1 and 1). We adopt here the logistic (or sigmoid) function, given by $g(x) = 1/(1 + e^{-x})$, as activation function.[24]

[22] RNNs can have additional storage states, which may incorporate feedback loops. Such extra states are referred in the literature to as gated states (or gated memory), and are part of long short-term memory networks (LSTMs) and gated recurrent units (GRUs). Here, we employ the standard RNN setup to save computational time and avoid unnecessary model complexity (overfitting).

[23] The so-called *fully* recurrent neural networks connect the outputs to the inputs of all neurons. This is the most general topology, since all other can be represented by setting some connection weights to zero to simulate the lack of connections between neurons. Also, by considering more than one layer, RNNs are part of a machine learning field called *deep learning*.

[24] Neural networks have been found to be very successful in complex settings, with large number of features. However, they often require a substantial amount of data in order to work well in practice.

2.2.6 Hybrid Machine Learning

The use of hybrid models in machine learning to forecast macroeconomic variables is relatively recent (e.g., Smyl, 2020). Here, the idea is to mix *methodologies* to produce a hybrid inflation forecasting method, which allows investigating whether the forecasting accuracy of a given ML model is primarily due to variable selection or to potential nonlinearities in the inflation data-generating process. Inspired by the hybrid method of Medeiros et al. (2021), we propose the following three-step approach:

1. Pre-selection of variables using the *target predictor* procedure of Bai and Ng (2008). First, we separately run individual OLS regressions (i.e., we run the dependent variable onto the intercept and a given individual predictor). Then, each predictor is retained in the set of selected variables only if it is statistically significant at a 5% level, based on its respective OLS regression.

2. Choice of the most relevant variables (e.g., Top20), according to either the Adaptive Lasso or the Random Forest (impurity corrected) approaches; see Appendix D for details on how to compute the variable importance.

3. Use the set of top variables as input (e.g., regressors) in the following models: OLS, Adaptive Lasso, Random Forest or XGBoost in order to forecast the inflation rate.

This way, we build six hybrid models, labelled by Ada-OLS, Ada-RF, Ada-XGB, RF-OLS, RF-Ada or RF-XGB, where *OLS* denotes Ordinary Least Squares, *Ada* means Adaptive Lasso, *RF* denotes Random Forest, and *XGB* means XGBoost. The first nickname stands for the variable selection method, whereas the second one denotes the estimation/forecasting approach. For example, the hybrid model Ada-OLS employs the Adaptive Lasso to choose the top predictors, which are then used as regressors in the OLS model to forecast inflation.

Although the set of nonlinear approaches could be further enlarged, we believe the six hybrid models designed here can help us check the importance of variable selection, as part of many ML methodologies, and the role nonlinearities play when modeling the inflation dynamics. Also, we bring to machine learning a flavor of model interpretation, by first revealing the top features and, then, using this superior dataset as input in a given ML forecasting method.

2.2.7 Traditional Inflation Forecasting

We next analyze more traditional forecasting methods, often used by economic agents when producing inflation forecasts.

Random Walk: The standard random walk (RW) model assumes the $h-$period inflation change is an unforecastable martingale difference sequence (MDS), that is $\mathbb{E}\left(y_{t+h} - y_t \mid \mathcal{F}_t\right) = 0$. The out-of-sample inflation forecast, for all $h = 1, ..., H$, is then $f^{rw}_{y_{T_2+h}} = y_{T_2}$.

RW-AO: This is the variant of the random walk model, considered by Atkeson and Ohanian (2001), which takes the average inflation over the previous twelve months[25] as the forecast for y_{T_2+h}, as follows: $f^{rw-ao}_{y_{T_2+h}} = \frac{1}{12}\sum_{j=0}^{11} y_{T_2-j}$.

ARMA: One of the most common statistical models used in time-series forecasting is the autoregressive moving average (ARMA) model, which assumes that future observations are driven essentially by recent observations. Inflation, which often exhibits persistent behavior, is largely consistent with this assumption.[26]

VAR: The vector autoregression (VAR) model is also a traditional method based on a backward-looking approach. We use here one lag[27] and the following endogenous variables: market price inflation, administered price inflation, one-year real interest rate, log-difference of nominal exchange rate (R$/US$) and output gap (proxied by the HP-filtered IBC-BR series).[28] The choice of variables recognizes different time dynamics of the two main components of inflation in Brazil[29] and incorporates the pass-through of imported inflation to domestic inflation. The forecast of headline inflation is built by aggregating the h-step ahead forecasts of the two inflation components using respective weights.[30]

Inflation expectations (Focus survey): The Focus survey of professional forecasts is a panel database put together by the Central Bank of Brazil (*Banco Central do Brasil – BCB*),

[25] According to Atkeson and Ohanian (2001, p.10): *"...economists have not produced a version of the Phillips curve that makes more accurate inflation forecasts than those from a naive model that presumes inflation over the next four quarters will be equal to inflation over the last four quarters."*

[26] The best ARMA(p,q) model is recursively selected using the Schwarz information criterion.

[27] According to the Schwarz information criterion and diagnostic testing.

[28] See variables 2, 3, 27, 51 and 120 in Appendix E.

[29] The administered price inflation is in some way regulated by a public agency or set by contracts (often including backward indexation clauses), rather than by the interaction between domestic demand and supply conditions. According to Minella et al. (2003), the dynamics of such prices differ from the market prices in three ways: (i) dependence on international prices in the case of refined petroleum products; (ii) greater pass-through from the exchange rate; and (iii) stronger backward-looking behavior.

[30] For details on IPCA weights, see https://www.bcb.gov.br/content/ri/inflationreport/201912/ri201912b7i.pdf

which collects daily information since 1999, after the implementation of the inflation-targeting regime in Brazil. The survey covers more than 100 professional forecasters (e.g., banks, asset management firms, consulting firms and some relevant non-financial institutions), which are followed throughout time with a reasonable turnover. The Focus survey is constantly used by market agents, specialized media and the Central Bank of Brazil itself to monitor inflation expectations. The forecasts are supplied over different horizons and for a large array of macroeconomic series; see Gaglianone et al. (2021). Here, we consider the median of individual inflation forecasts (IPCA) across all survey participants.

Inflation expectations (BEI): Another key source of inflation expectations in Brazil is financial data. For instance, using inflation-linked treasury bond market data,[31] and the parametric model of Svensson (1994), one can extract the so-called *Breakeven Inflation* (BEI), which is available on a daily basis and focused on actual financial market agents' decisions; see Val and Araujo (2019). However, the usage of such data usually embodies risk premium issues and maturities with different market liquidity. Since there is no consensus in the finance literature on how to properly compute risk premium, we do not extract it from the BEI series. Besides, it is often neglected for short horizons, although it might become relevant for longer horizons.[32]

PC-backward: The Phillips curve model (PC) has a long tradition in forecasting inflation (Stock and Watson, 1999). We consider here a backward-looking version of the curve, only including past inflation (inertia), imported inflation (pass-through channel)[33] and output gap (traditional monetary policy channel via aggregate demand).[34] Following the VAR approach, we also disaggregate inflation in two components (market price inflation and administered price inflation), which are modeled separately. First, we estimate a Phillips curve for inflation of market prices. Then, we estimate an ARMA(p, q) model for the administered price inflation. Finally, the headline inflation forecast is built by aggregating the forecasts of the two inflation components using respective weights.

[31] We use data from NTN-B, which is the acronym for *Nota do Tesouro Nacional*, type B, similar to the Treasury Inflation-Protected Securities (TIPS) in the U.S.

[32] We use end-of-month data when considering inflation expectations at monthly frequency. For example, when the goal is to forecast the IPCA of June 2021, for $h = 1$, we use the inflation expectations available on May 31, 2021. Similarly, in order to forecast the IPCA of June 2021, for $h = 2$, we use the Focus and BEI data available on April 30, 2021.

[33] Defined as the sum of the nominal exchange rate (R$/US$) monthly percentage variation and the U.S. inflation (assumed, for simplicity, 2.0% per year).

[34] The output gap is based on the seasonally adjusted IBC-BR index of economic activity. The Hodrick-Prescott (HP) filter is employed to generate the output gap in a recursive estimation scheme, that is, we re-construct the entire output gap series for each new observation added to the estimation sample along the out-of-sample exercise (and, then, re-estimate the Phillips curve to build new forecasts).

PC-hybrid: This approach considers a hybrid (New Keynesian) version of the Phillips curve, which includes backward and forward looking terms, imported inflation and output gap; see Arruda et al. (2011) and Gaglianone, Issler and Matos (2017). The extra term is expected inflation, proxied here by the *Focus* survey. We impose the usual coefficient restriction[35] to guarantee a vertical long-run Phillips curve. The forecasts for administered price inflation and headline inflation follow the same procedures described in the previous approach.

Factor model 1 (direct forecast): The idea that time variations in a large number of variables can be summarized by a small number of factors is empirically attractive and it is employed in a large number of studies in economics and finance; see Forni et al. (2000) and Stock and Watson (2002). Let $x_{i,t} \in \widetilde{x}_t$ be the observed data for the i-th cross-section unit at time t, for $i = 1, ..., N$ and $t = 1, ..., T_2$, and consider the following factor representation of the data:

$$x_{i,t} = \lambda_i' F_t + e_{i,t}, \tag{12}$$

where F_t is a vector of common factors, λ_i is a vector of factor loadings associated with F_t and $e_{i,t}$ is the idiosyncratic component of $x_{i,t}$. Note that λ_i, F_t and $e_{i,t}$ are unknown since only $x_{i,t}$ is observable. Here, we estimate the factors and respective loadings using principal components analysis (PCA). The number of components is determined by the Bai and Ng (2002) criterion. After the PCA estimation of the common factors F_t, we employ the *direct forecast* approach, to model the inflation rate at time $t + h$, as follows:

$$y_{t+h} = \beta_h F_t + \varepsilon_{t+h}. \tag{13}$$

Therefore, the inflation forecast from the (direct) factor model above, using a sample of $t = 1, ..., T_2$ observations, is given by $f_{y_{T_2+h}}^{fm-direct} = \widehat{\beta_h}\widehat{F_{T_2}}$, for $h = 1, ..., H$.

Factor model 2 (iterated forecast): This approach is a variant of the previous one, but using an iterated forecast method instead of the direct forecast approach. The idea is again to employ common factors, but to model the inflation rate in a contemporaneous way in respect to the factors, that is:

$$y_t = \gamma F_t + v_t. \tag{14}$$

Following the literature (e.g., Bańbura et al., 2013), we specify the factors as following a VAR process, that is, $F_t = \Phi(L)F_t + u_t$. The inflation forecast from this *iterated* factor model,

[35] The sum of coefficients on past inflation, expected inflation and imported inflation must be equal to one.

using a sample of $t = 1, ..., T_2$ observations, is given by $f_{y_{T_2+h}}^{fm-iterated} = \widehat{\gamma}\widehat{F_{T_2+h|T_2}}$, for $h = 1, ..., H$, where $\widehat{F_{T_2+h|T_2}}$ are the $h-$step ahead (out-of-sample) forecasts of the common factors, using the VAR model estimated in a recursive scheme.

Factor models 3 and 4 (with targeted predictors, direct or iterated): These are the previous factor models, but now based on a *subset* of predictors that are selected by taking into account that our variable of interest is the inflation rate. Here, we follow the idea of Bai and Ng (2008), who showed the factor model's out-of-sample forecasting performance could be improved by previously selecting (or targeting) the predictors.

The core idea is that irrelevant predictors employed to build a factor model only add noise into the analysis, and thus produce factors with a poor predictive performance. In this sense, we use in the factor model only pre-selected variables, as follows:

(i) in the direct forecast case, we first regress the inflation rate y_{t+h} (or y_t in the iterated case) on the intercept and the candidate variable $x_{i,t} \in \widetilde{x}_t$, for all $i = 1, ..., N$;

(ii) calculate the $t-$statistics for the coefficient associated to $x_{i,t}$;

(iii) include $x_{i,t}$ in the set of predictors (used to extract the factors) only if it is statistically significant at a 5% level;

(iv) proceed as before, in the direct or iterated factor model cases, to build the respective inflation forecasts.

2.2.8 Disaggregated Forecasts

According to a recent *Inflation Report* released by the Central Bank of Brazil (BCB, 2021), the three main groups of market prices (services, industrial goods, and food at home) show important differences in terms of average inflation level, dynamics and determinants.[36] Taking into account such differences in the data-generating process of the inflation subgroups can (potentially) lead to accuracy gains when forecasting the aggregated inflation.[37]

This motivates a *bottom-up* approach, in which we separately model and forecast the inflation dynamics for each one of the main three subgroups of market prices, besides administered prices. To do so, we employ three different models: ARMA, Adalasso and Random Forest.

[36] The report mentions that: *"...disaggregated approaches are useful for extending the scope of the analysis and broadening the understanding of inflation developments and its prospects."* According to the same report, the greatest inertial component is related to the services sector, which is not directly impacted by exchange rate changes or by commodity prices. On the other hand, these factors are relevant in the sectors of industrial goods and food at home.

[37] Also, in the equity-premium literature, Ferreira and Santa-Clara (2011) reports that forecasting separately the three components of stock market returns (dividend yield, earnings growth, and price-earnings ratio growth) yields huge forecasting gain.

Then, we aggregate[38] the four individual forecasts (administered prices, services, industrial goods, and food at home) to form the forecast of headline inflation (IPCA).

2.2.9 Forecast Combination

Since the seminal work of Bates and Granger (1969), it has been observed that combining forecasts across multiple models often produces better forecasts compared to a single model. Nowadays, the accuracy gains of forecast combination over individual forecasts are well-documented in the literature. According to Elliott et al. (2015): *"...forecast combination offers one approach for dealing with the effects of estimation error, model uncertainty, and instability in the underlying data generating process. By diversifying across multiple models, combinations typically deliver more stable forecasts than those associated with individual models."*

Here, we employ different forecast combination methods based on three main sets of individual forecasts. The first set of forecasts (set1) entails individual forecasts from models 1-27. The second and third sets of forecasts (set2 and set3) include only the superior models of the *Model Confidence Set* proposed by Hansen et al. (2011), considering models 1-27 or 1-29, respectively.

The Model Confidence Set (MCS) consists of a sequence of tests allowing the construction of a set of superior models, where the null hypothesis of equal predictive ability is not rejected at a given confidence level. The test statistic can be evaluated for several loss functions, such as mean squared error (MSE) or mean absolute error (MAE). The MCS is a sequential testing procedure, which eliminates the worst model at each step until the hypothesis of equal predictive ability is accepted for all the models belonging to a set of superior models. The MCS method is focused not on the selection of optimal weights, but on the selection of superior models. In other words, it trims out the worse performing models based on a statistical significance test. We implement the MCS procedure considering the MSE loss function and the 95% confidence level. See Shang and Haberman (2018) for further details.

This way, based either on set1, set2 or set3 of forecasts, we use the following forecast combination approaches: Mean, Median, Adalasso, Random Forest, Granger and Ramanathan (1984), Constrained Least Squares (CLS), and Complete Subset Regression (CSR). The first two combination approaches are simply the mean and median forecasts, computed across a given set of individual forecasts (set1, set2 or set3). The Adalasso and Random Forest approaches are used here as forecast combination devices (i.e., instead of using a set of *predictors*, they are now based on a given set of individual *forecasts*).

[38] Using real-time weights of each IPCA subgroup.

In turn, Granger and Ramanathan (1984) set out the foundations of optimal forecast combinations under symmetric and quadratic loss functions. The authors show that under MSE loss the optimal weights can be estimated through an ordinary least squares (OLS) regression of the target variable (inflation, in our case) on a given set of forecasts, plus an intercept to account for possible model bias. However, if the loss function differs from MSE, then the computation of optimal weights may require methods other than a simple OLS.

Although the OLS combination of Granger and Ramanathan (1984) enables us to correct for bias through its intercept term, Nowotarski et al. (2014) points out such unbiasedness comes at the expense of a poorer performance for highly correlated regressors. The Constrained Least Squares (CLS) approach advocated by Nowotarski et al. (2014) is a variant of the previous OLS combination with additional constraints: no intercept term is imposed and the coefficients have to be non-negative and be summed up to 1.

In a distinct approach, Elliott et al. (2013, 2015) propose a method for combining forecasts based on Complete Subset Regressions (CSR). The method combines forecasts based on predictive regressions with k number of predictors (in our setup, a given set of individual forecasts).[39] Hence, assuming $k = 1$ corresponds to an equal-weighted average of all possible forecasts from univariate prediction models, whereas $k = 2$ corresponds to equal-weighted averages of all possible forecasts from bivariate prediction models. The CSR approach has the computational advantage that it can be applied even when the number of predictors exceeds the sample size.[40]

2.3 Fan Chart

Providing confidence intervals is a relevant issue when building point forecasts. Our setup of competing forecasts can be easily adapted to provide a measure of uncertainty around future predictions of inflation. To summarize the idea, we first compute the forecast errors of a given method of interest for a set of horizons $h = 1, ..., H$. Next, we estimate the forecast variance (for each h) and smooth out these variances using a *spline* function to obtain a *smooth* term-structure of variances along the horizons h. Finally, we generate the out-of-sample conditional

[39] The optimal value of k can be determined from the covariance matrix of the potential regressors and, thus, can be selected recursively in time.

[40] According to Elliott et al. (2015), Monte Carlo simulations show that CSR offers a favorable bias-variance trade-off in the presence of many weak predictors. However, one drawback is that the number of regressions to be estimated increases very quickly for large datasets. In such cases, Garcia et al. (2017) adopts a pre-testing procedure, similar to the *targeting predictors* approach of Bai and Ng (2008). In this paper, since the number of predictors is very large, we employ the CSR method with $k = 2$ as a forecast combination device, based on a given set of individual forecasts, instead of considering subset regressions on all candidate variables.

quantiles for a grid of quantile levels τ assuming a Gaussian distribution.[41]

Thus, in order to produce density forecasts of the inflation rate y_{t+h}, using the information set $\mathcal{F}_t$ available at period t, we assume the conditional distribution of y_{t+h} is Gaussian, with conditional mean $\mu_{t+h|t}$ and conditional variance $\sigma^2_{t+h|t}$, that is $(y_{t+h} \mid \mathcal{F}_t) \sim N(\mu_{t+h|t}, \sigma^2_{t+h|t})$. The conditional quantile of y_{t+h}, evaluated at quantile level $\tau \in (0,1)$, is given as follows:

$$Q_\tau(y_{t+h} \mid \mathcal{F}_t) = \mu_{t+h|t} + \sigma_{t+h|t}\Phi^{-1}(\tau)\,. \tag{15}$$

Now, let $f^m_{t+h|t}$ be the model m estimate of the conditional mean of y_{t+h}. Thus, $f^m_{t+h|t} = \widehat{\mu_{t+h|t}}$, where $\mu_{t+h|t} = \mathbb{E}(y_{t+h} \mid \mathcal{F}_t)$. Also, let $\widehat{\sigma^2_{t+h|t}}$ be the model m estimate of the conditional variance of y_{t+h}, that is $\sigma^2_{t+h|t}$, computed using the Newey and West (1987)'s HAC covariance matrix estimator, from a regression of the forecast error of $f^m_{t+h|t}$ on the intercept.[42]

Provided that $[\widehat{\mu_{t+h|t}}, \widehat{\sigma^2_{t+h|t}}]$ are consistent estimates of $[\mu_{t+h|t}, \sigma^2_{t+h|t}]$, one can obtain consistent estimates of the conditional quantiles of y_{t+h}, along a grid of quantile levels $\tau \in \Gamma$, using equation (15). Therefore, the multi-step ahead density forecasts of y_{t+h} can be summarized by a *fan chart* graph, based on the estimated conditional quantiles, over the horizons $h = 1, ..., H$, and the grid of quantile levels $\tau \in \Gamma$.[43]

3 Empirical Exercise

3.1 Data

We focus the analysis on the IPCA, which is the consumer price index (CPI) measured by the Brazilian Institute of Geography and Statistics (IBGE), used to compute the official inflation measure and the target of monetary policy in Brazil. The dependent variable is either the monthly percentage change of the IPCA index, or this measure accumulated over the last twelve months. Forecast horizon (h) varies from 1 to 18 months. The sample period spans over 17 years of data, from January 2004 to August 2021 ($T = 212$ observations).[44] Figure

[41] The objective here is not to produce a density forecast based on a more complex approach (e.g., allowing for asymmetry and fat tails), but only to attach a simple measure of uncertainty to the path of future inflation according to each model's past forecast errors.

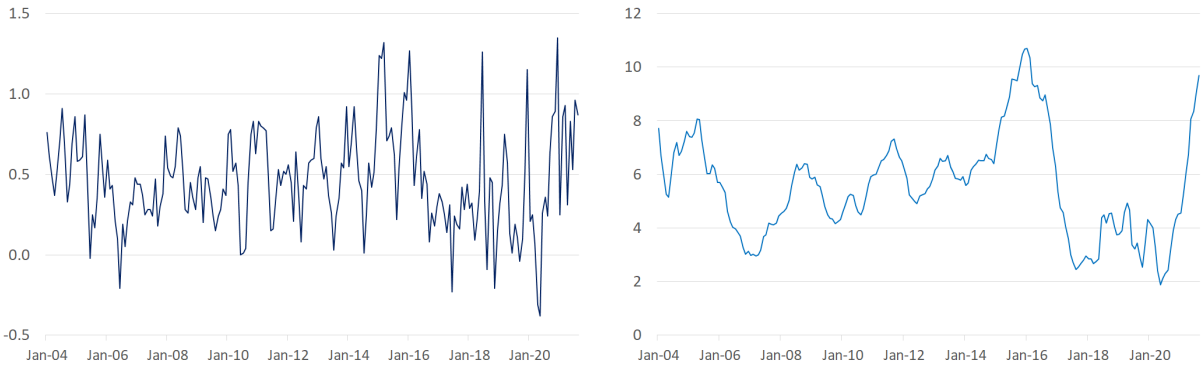
[42] Recall the forecast error $\left(f^m_{t+h|t} - y_{t+h}\right)$ is computed along a (pseudo) out-of-sample forecasting exercise, that is, considering $t = t_{oos1}, ..., t_{oosT}$ and a given h.

[43] See Costa et al. (2021) for a similar exercise of density forecasting the future oil prices.

[44] According to Machado and Portugal (2014), the limited sample problem is a well-known constraint for inference in Brazilian studies, particularly in inflation dynamics where different policy regimes have been the case. In this sense, selecting the sample since 2004 helps us avoid large structural regime breaks.

4 shows the IPCA inflation in our sample period, which starts a decade after the Brazilian monetary stabilization plan in mid-1994. Note that inflation rate levels are compatible to the ones observed in many inflation-targeting and emerging countries.

Figure 4 - Inflation rates (IPCA), % per month (left), % accumulated in 12 months (right)



One of the key features driving the inflation dynamics in emerging economies is the degree of persistence (or inertia).[45] Besides past inflation, other predictors suggested in the literature to forecast inflation often include economic slack measures (e.g., unemployment rate, in a Phillips curve), variables related to production (Stock and Watson, 1999), financial variables (Forni et al., 2003), surveys of expectations (Ang et al., 2007; Faust and Wright, 2013), among others.

In this paper, we use a diverse set of macroeconomic and financial variables drawn from a number of categories.[46] Our database consists of $n = 167$ contemporaneous monthly variables, including (for instance): price indexes, interest rates, financial markets variables, economic activity, labor market variables, government debt, import and export of goods and services, international variables that are potentially related to the Brazilian economy. The main data sources are Anbima, BCB, EPU (Baker et al., 2015), FGV, Funcex, IBGE, Inmet, IpeaData, and Reuters (Refinitiv Eikon Datastream). Appendix E presents the full list of variables used as potential predictors for the inflation rate in Brazil.

In order to ensure stationarity, we conduct individual time series transformation, following the procedure adopted in the FRED-MD database of McCracken and Ng (2015). We consider

[45] In Brazil, the relevance of past inflation has been vastly documented. For instance, Kohlscheen (2012) suggests that models in which past inflation have greater weight in the expectations formation process are more accurate than others purely based on the rational expectations assumption. In turn, Gaglianone, Guillén and Figueiredo (2018) points out the relevance of considering a time-varying inertia when building more accurate inflation forecasting models.

[46] Besides the usual macro series, we included many financial variables, which are shown in the literature (Forni et al., 2003) to be predictors that help forecast inflation. For example, financial market-based implied (breakeven) inflation, which provides a closer monitoring of inflation expectations (since they are updated on a continuously intra-day basis) and are competitive in terms of short-run predictive ability compared to survey expectations (Araujo and Vicente, 2017). We also included in the database many non-standard variables, for instance, to capture supply shocks arising from climate factors (e.g., amount of rainfall in Brazilian cities or even Pacific Ocean temperatures, to capture El Niño or La Niña effects).

six possibilities, as follows: (1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $\ln(x_t)$; (5) $\Delta\ln(x_t)$; and (6) $\Delta^2\ln(x_t)$. The transformation adopted for each series is presented in Appendix E.

After transformations, the stationarity of each time series is checked using the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, considering a 5% significance level. The $n = 167$ contemporaneous variables are then lagged $s = 3$ periods,[47] forming a final database containing 501 series. This way, besides an intercept, equation (1) entails $dim\,(\widetilde{x}_t') = 501$ variables, used as potential predictors for inflation in Brazil.[48]

Recall the first part of the sample ($t = 1, ..., T_1$) is used to estimate the models, whereas the second part of the sample ($t = T_1 + 1, ..., T_2$) is used for estimation of the forecast combination weights (where applicable). The remaining observations ($t = T_2 + 1, ..., T$) are reserved for genuine out-of-sample forecast comparison. We consider $T_1 = 72$ months (6 years), $T_2 = 120$ months (10 years), and $P = T - T_2 = 212 - 120 = 92$ out-of-sample observations. Thus, the evaluation period for $h = 1$ ranges from January 2014 to August 2021 (92 forecasts), whereas for $h = 18$ ranges from June 2015 to August 2021 (75 forecasts).[49]

All models are re-estimated every month in a *recursive* estimation scheme (i.e., expanding sample size), as we incorporate every new time-series observation, one at a time. In this context, each model is initially estimated using the first T_1 (or T_2 in the case of forecast combinations) observations and the out-of-sample point forecasts are generated. We, then, add an additional observation at the end of the *training set*, re-estimate the models and generate again out-of-sample forecasts. This process is repeated along the remaining data (*test set*).[50]

We also conduct an extensive *robustness* analysis by considering two alternative training/test sets: (i) $T_1 = 96$ months (8 years) and $T_2 = 144$ months (12 years); and (ii) $T_1 = 72$ months (6 years) and $T_2 = 144$ months (12 years). The results are presented in Appendix F.

[47] Inflation models usually comprise a rich lag structure (particularly in emerging countries, more prone to inflation inertia). Such structure should capture the dynamic relationship between inflation, past inflation and key macroeconomic variables. Here, we adopt 3 lags to avoid overfitting (besides, forecasting exercises with more lags generally produced higher MSEs).

[48] We standardize data (zero mean and unity variance) in the penalized-regression models (elastic net, lasso, adaptive lasso and ridge), factor models, recurrent neural network and hybrid models. In turn, tree-based methods (random forest, QRF and XGBoost) do not require feature scaling (since common practice in such methods is not to standardize features, we follow this approach here).

[49] Our forecasting exercise is not implemented on a *strict* real-time basis, to avoid unnecessary complications in the execution of the exercise. Practical limitations arise due to data revisions and/or different release delays of features in real time. The literature on inflation forecasting is generally focused on *pseudo* real-time exercises, such as the one conducted in this paper. Besides, the real-time issue looses relevance when considering longer horizons, such as 12 or 18 months.

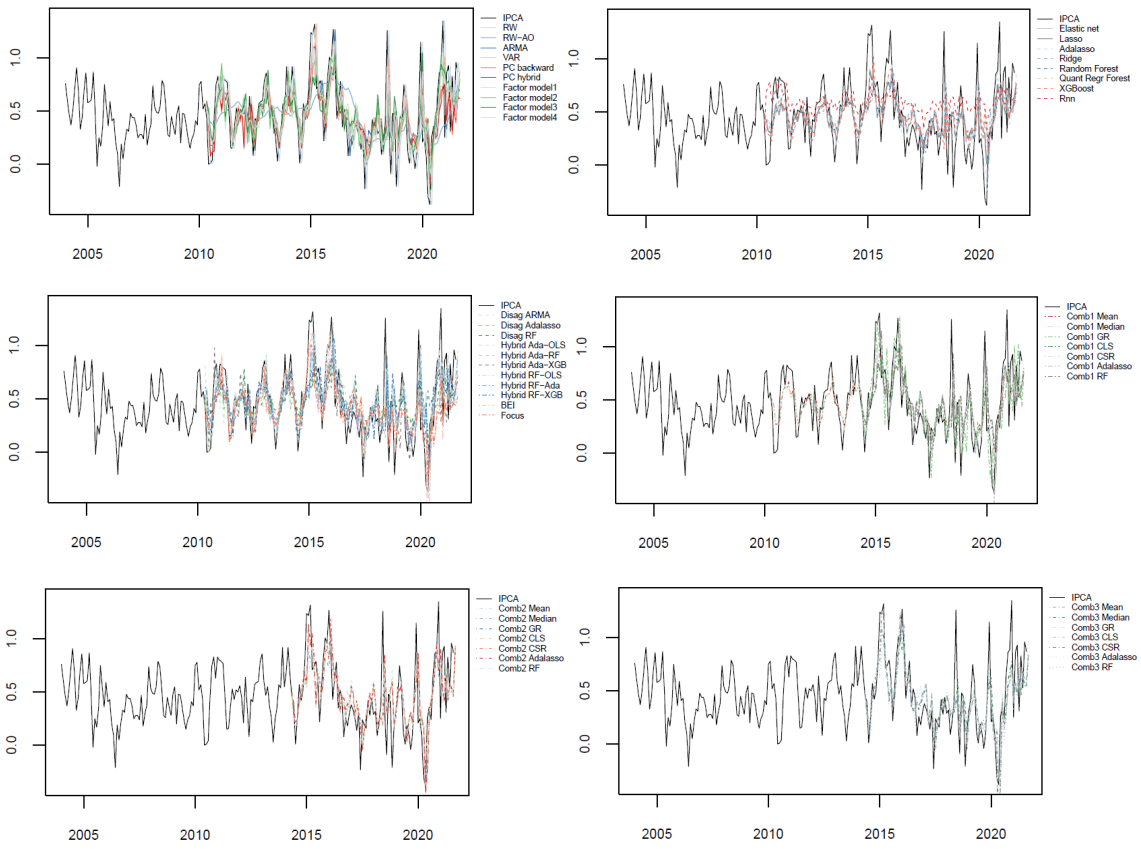
[50] We adopt such an estimation scheme due to the greater efficiency, in general, of recursive regressions compared to rolling-window estimations. However, the latter approach could be justified under a framework with the possibility of structural changes. See Morales-Arias and Moura (2013) for a good discussion on this issue.

The empirical exercise is implemented using the R software (version 4.1.0, 64-bit). The ridge regression, lasso and elastic net models are estimated using the R package *glmnet*, which fits a generalized linear model via penalized maximum likelihood. The adalasso model is implemented using the R package *HDeconometrics*. The same R package is used to compute the BIC information criterion for selection of hyper-parameters. In order to implement the random forest and the quantile regression forest methods,[51] we use the R package *ranger*, whereas the XGBoost approach is based on the R package *xgboost*. Finally, the R package *rnn* is used to implement the recurrent neural networks, and the R package *MCS* is used to implement the *model confidence set* procedure of Hansen et al. (2011).

3.2 Results

The observed inflation rate (% per month, or simply % p.m.) and the respective out-of-sample forecasts ($h = 1$) of the 50 approaches covered in this paper are shown in Figure 5. Appendix G presents forecasts of the inflation rate accumulated in twelve months.

Figure 5 - Inflation (% p.m.) and forecasts, $h = 1$



[51] We used 1,000 trees in both the random forest and the quantile regression forest. In the latter method, we adopt the following grid of quantile levels: $\tau \in (0.05, 0.10, ..., 0.95)$.

For each horizon, the forecast errors are squared and averaged to form the out-of-sample MSE. In addition, we compute the p-value of the Diebold and Mariano (1995) test for non-nested models,[52] using the forecasts from the ARMA model as *benchmark*. Besides the MSE, we compare the best model with the benchmark, at each horizon, in terms of the R^2 out-of-sample statistics (Rapach et al., 2010).[53] Table 2 presents the results.

Table 2 - Mean Squared Error (MSE)

dep. var. = IPCA % p.m.	h = 1	h = 2	h = 3	h = 6	h = 9	h = 12	h = 15	h = 18
(1) RW	0.129	0.187	0.200	0.278	0.204	0.255	0.305	0.279
(2) RW-AO	0.124	0.137	0.146	0.164	0.185	0.194	0.197	0.200
(3) ARMA	0.104	0.137	0.138	0.157	0.168	0.157	0.172	0.173
(4) VAR	0.107	0.138	0.141	0.157	0.156	0.145	0.147*	0.150
(5) PC backward	0.102	0.136	0.143	0.162	0.153	0.139*	0.149	0.172
(6) PC hybrid	0.082***	0.115**	0.123	0.142	0.146*	0.146	0.146	0.159
(7) Factor model1	0.088**	0.113**	0.139	0.156	0.164	0.145	0.148*	0.150
(8) Factor model2	0.085***	0.117*	0.133	0.154	0.154	0.139*	0.142*	0.146*
(9) Factor model3	0.088*	0.108***	0.133	0.155	0.148*	0.150	0.141**	0.158
(10) Factor model4	0.094	0.120**	0.140	0.154	0.153	0.141	0.141**	0.146*
(11) Elastic net	0.092**	0.129	0.149	0.152	0.153	0.145	0.141*	0.150
(12) Lasso	0.092**	0.128	0.151	0.152	0.153	0.146	0.142*	0.150
(13) Adalasso	0.088***	0.119*	0.152	0.153	0.157	0.153	0.148	0.149
(14) Ridge	0.100	0.122**	0.139	0.149	0.161	0.147	0.153	0.155
(15) Random Forest	0.105	0.124	0.141	0.151	0.154	0.139*	0.139*	0.155
(16) Quant Regr. Forest	0.106	0.126	0.143	0.150	0.153	0.138**	0.136**	0.150
(17) XGBoost	0.098	0.116**	0.143	0.161	0.159	0.146	0.133**	0.172
(18) RNN	0.134	0.134	0.144	0.152	0.143*	0.136*	0.128**	0.146
(19) Disag. ARMA	0.110	0.144	0.153	0.168	0.174	0.168	0.169	0.162
(20) Disag. Adalasso	0.097	0.135	0.161	0.143	0.161	0.146	0.146*	0.141*
(21) Disag. RF	0.114	0.133	0.144	0.156	0.152	0.141**	0.148	0.164
(22) Hybrid Ada-OLS	0.090***	0.120*	0.139	0.155	0.172	0.174	0.127**	0.167
(23) Hybrid Ada-RF	0.100	0.112**	0.135	0.168	0.170	0.157	0.127**	0.165
(24) Hybrid Ada-XGB	0.107	0.120	0.154	0.219	0.201	0.172	0.145	0.185
(25) Hybrid RF-OLS	0.101	0.120*	0.213	0.203	0.196	0.208	0.208	0.172
(26) Hybrid RF-Ada	0.092*	0.117**	0.187	0.173	0.160	0.157	0.190	0.157
(27) Hybrid RF-XGB	0.093	0.112**	0.147	0.158	0.174	0.153	0.125**	0.175
(28) BEI	0.067**	0.140	0.126	0.150	0.161	0.154	0.147	0.157
(29) Focus	0.073***	0.116	0.128	0.141	0.152	0.141	0.138	0.141
(30) Comb1 Mean	0.089**	0.114***	0.134	0.147	0.151	0.145	0.137**	0.147
(31) Comb1 Median	0.090**	0.116**	0.136	0.149	0.153	0.141*	0.136**	0.149
(32) Comb1 GR	0.084**	0.144	0.157	0.239	0.238	0.393	0.285	0.350
(33) Comb1 CLS	0.089***	0.114***	0.136	0.147	0.152	0.144	0.139**	0.148
(34) Comb1 CSR	0.090**	0.118**	0.140	0.165	0.192	0.272	0.197	0.224
(35) Comb1 Adalasso	0.081***	0.118	0.133	0.174	0.198	0.319	0.291	0.284
(36) Comb1 RF	0.095	0.126	0.151	0.190	0.197	0.205	0.227	0.217
(37) Comb2 Mean	0.092**	0.115***	0.144	0.146	0.154	0.147	0.138**	0.160
(38) Comb2 Median	0.092**	0.115***	0.139	0.149	0.155	0.147	0.138**	0.158
(39) Comb2 GR	0.093**	0.122	0.150	0.171	0.223	0.184	0.235	0.227
(40) Comb2 CLS	0.090***	0.117**	0.138	0.151	0.154	0.149	0.135**	0.168
(41) Comb2 CSR	0.094**	0.122	0.146	0.165	0.181	0.184	0.204	0.227
(42) Comb2 Adalasso	0.092**	0.123	0.152	0.172	0.207	0.174	0.234	0.223
(43) Comb2 RF	0.101	0.118*	0.153	0.173	0.192	0.201	0.200	0.205
(44) Comb3 Mean	0.067***	0.120	0.124	0.142	0.155	0.139	0.136**	0.146
(45) Comb3 Median	0.067***	0.120	0.124	0.146	0.155	0.139	0.136**	0.146
(46) Comb3 GR	0.055***	0.121	0.128	0.180	0.161	0.166	0.215	0.162
(47) Comb3 CLS	0.068**	0.132	0.130	0.146	0.155	0.141	0.136*	0.146
(48) Comb3 CSR	0.071*	0.123	0.129	0.156	0.161	0.166	0.178	0.162
(49) Comb3 Adalasso	0.070*	0.121	0.129	0.160	0.163	0.166	0.185	0.162
(50) Comb3 RF	0.076***	0.124	0.131	0.158	0.190	0.182	0.175	0.174
number of observations	*88*	*87*	*86*	*83*	*80*	*77*	*74*	*71*
best model	*46*	*9*	*6*	*29*	*18*	*18*	*27*	*29*
R2 oos (%)	*47*	*21*	*10*	*10*	*14*	*13*	*27*	*18*

dep. var. = IPCA % 12 months	h = 1	h = 2	h = 3	h = 6	h = 9	h = 12	h = 15	h = 18
(1) RW	0.259	0.795	1.448	3.854	6.551	10.059	11.966	13.254
(2) RW-AO	3.238	4.106	4.983	7.513	9.925	11.948	13.169	13.821
(3) ARMA	0.189	0.674	1.268	3.829	6.579	11.005	14.012	17.600
(4) VAR	0.268	0.844	1.574	4.460	7.755	11.917	14.411	15.237
(5) PC backward	0.233	0.734	1.334	3.669	5.664	8.904	12.134	15.115
(6) PC hybrid	0.228	0.697	1.240	3.558	5.897	9.053	12.642	16.194
(7) Factor model1	0.768	1.306	1.922	4.081	6.890	9.305	8.832	8.476
(8) Factor model2	0.515	0.997	1.574	3.814	6.305	9.131	10.262	10.384
(9) Factor model3	0.439	0.993	1.747	4.720	8.157	8.707	7.887	7.066
(10) Factor model4	0.408	0.899	1.456	3.470	6.205	9.960	11.569	11.382
(11) Elastic net	0.253	0.854	1.662	4.474	9.741	11.590	11.326	7.789
(12) Lasso	0.251	0.814	1.648	4.334	9.492	11.603	11.627	7.734
(13) Adalasso	0.220	0.785	1.568	5.090	10.209	14.386	17.436	12.252
(14) Ridge	1.211	1.748	2.442	4.902	7.448	8.386	8.195	7.154
(15) Random Forest	0.936	1.667	2.432	4.807	7.158	8.106	7.731	6.928
(16) Quant Regr. Forest	0.929	1.677	2.459	4.812	7.099	8.022	7.569	6.738
(17) XGBoost	0.291	0.873	1.657	4.681	7.110	7.751	6.867	5.881
(18) RNN	3.079	4.492	3.190	8.099	5.560	9.873	10.701	8.591
(19) Disag. ARMA	0.186	0.654	1.256	3.774	6.392	10.531	12.405	14.232
(20) Disag. Adalasso	0.234	0.833	1.970	5.245	10.187	10.668	12.970	12.201
(21) Disag. RF	0.982	1.781	2.623	5.132	7.140	7.967	8.092	7.879
(22) Hybrid Ada-OLS	0.209	0.729	1.509	4.513	11.412	14.469	14.719	9.173
(23) Hybrid Ada-RF	0.561	1.244	1.974	4.647	7.444	9.297	9.089	8.279
(24) Hybrid Ada-XGB	0.379	1.043	1.918	5.126	8.049	9.478	8.775	6.883
(25) Hybrid RF-OLS	0.182	0.746	1.438	5.200	16.010	17.836	20.383	10.608
(26) Hybrid RF-Ada	0.188	0.790	1.658	4.788	16.140	18.435	18.502	10.201
(27) Hybrid RF-XGB	0.319	1.116	2.166	4.932	7.886	8.412	7.207	6.626
(28) BEI	0.072***	0.396**	0.636**	2.412*	4.848*	7.239	7.120	7.290
(29) Focus	0.080***	0.367***	0.748**	2.381*	4.349*	6.306	6.305	6.023
(30) Comb1 Mean	0.331	0.880	1.520	3.939	6.851	8.172	7.689	6.004
(31) Comb1 Median	0.241	0.824	1.521	4.068	7.084	8.447	8.041	6.625
(32) Comb1 GR	0.244	0.824	1.896	5.346	10.638	21.204	20.188	15.492
(33) Comb1 CLS	0.316	0.893	1.610	4.513	6.774	8.360	7.687	6.026
(34) Comb1 CSR	0.343	0.852	1.645	5.372	10.008	12.913	14.097	14.011
(35) Comb1 Adalasso	0.236	0.785	1.541	6.691	14.700	28.671	27.811	16.241
(36) Comb1 RF	0.420	1.057	2.084	5.653	9.517	12.677	14.247	12.175
(37) Comb2 Mean	0.182	0.783	1.459	4.158	6.543	11.062	12.655	12.650
(38) Comb2 Median	0.182	0.773	1.430	4.314	6.721	11.062	12.655	12.650
(39) Comb2 GR	0.206	0.748	1.552	4.096	12.533	19.296	22.484	24.721
(40) Comb2 CLS	0.186	0.715	1.441	4.523	6.952	13.451	15.088	12.206
(41) Comb2 CSR	0.192	0.805	1.620	5.619	10.698	19.296	22.619	27.822
(42) Comb2 Adalasso	0.193	0.754	1.435	5.168	13.552	17.963	20.351	28.134
(43) Comb2 RF	0.294	1.106	2.061	6.143	9.288	10.815	10.402	13.972
(44) Comb3 Mean	0.070***	0.362***	0.684**	3.606	6.252	9.025	9.517	9.962
(45) Comb3 Median	0.070***	0.362***	0.684**	3.992	6.574	9.025	9.517	9.962
(46) Comb3 GR	0.084***	0.405**	0.794*	3.073	8.126	17.610	21.024	18.308
(47) Comb3 CLS	0.073***	0.409**	0.646**	3.044	5.759	11.262	9.682	10.483
(48) Comb3 CSR	0.085***	0.413**	0.698**	4.287	9.300	18.846	27.321	25.635
(49) Comb3 Adalasso	0.071***	0.424*	0.702**	3.636	9.908	18.472	26.671	23.528
(50) Comb3 RF	0.180	0.629	1.241	5.042	8.396	11.425	13.284	15.828
number of observations	*88*	*87*	*86*	*83*	*80*	*77*	*74*	*71*
best model	*45*	*45*	*28*	*29*	*29*	*29*	*29*	*17*
R2 oos (%)	*62*	*46*	*49*	*37*	*33*	*42*	*55*	*66*

Notes: Yellow cells denote Top10 models (lowest MSEs) in each horizon. ***, **, and * indicate rejection at 1%, 5%, and 10% levels, respectively, using the Diebold and Mariano (1995) test and considering model 3 as benchmark. The R2 out-of-sample statistics (R2 oos) refers to the best model in each horizon.

Considering the monthly inflation rate (left panel in Table 2), at the shortest horizon ($h = 1$), the best model is model 46 (comb3 GR), which provides an accuracy gain of 47%, in terms

[52] The null hypothesis assumes equal forecasting accuracy of two competing forecasts. The variances entering the test statistics use here the Newey and West (1987) HAC covariance estimator.

[53] It is defined as follows: $R^2oos = 100 \times \left(1 - \left(\sum_{t=T_2+1}^{T} \left(y_{t+h} - \widehat{f}^{i}_{t+h|t}\right)^2\right) / \left(\sum_{t=T_2+1}^{T} \left(y_{t+h} - \widehat{f}^{BMK}_{t+h|t}\right)^2\right)\right)$, where $\widehat{f}^{i}_{t+h|t}$ is the forecast of y_{t+h} from model i using information up to period t and $\widehat{f}^{BMK}_{t+h|t}$ is the benchmark forecast. Positive (negative) values for the R^2oos means forecast $\widehat{f}^{i}_{t+h|t}$ beats (is beaten by) $\widehat{f}^{BMK}_{t+h|t}$.

of the R^2oos statistic, compared to the ARMA model. Such expressive result is statistically significant (at 1% level, using the Diebold-Mariano test), and it is achieved by using the *Model Confidence Set* (MCS) of Hansen et al. (2011) on the largest set of forecasts (that is, including BEI and Focus), and the Granger and Ramanathan (1984) method to estimate the weights used to combine the MCS *superior* forecasts. The top forecasts at $h = 1$ also include BEI and Focus, besides other forecast combinations using the third set of forecasts (comb3).

For longer horizons, the accuracy gains of the best models over the benchmark decrease, ranging from 10% to 27%. Among the *machine learning* methods, it is worth highlighting the good performance of the regression tree-based methods (random forest, quantile regression forest[54] and xgboost), in particular, for longer horizons ($h \geq 12$). The recurrent neural network (RNN) also shows a good result at longer horizons.[55] These results reflect the importance of nonlinear methods when modeling the inflation dynamics in Brazil.

On the other hand, traditional inflation forecasting (linear) models, such as ARMA and VAR, never enter the set of Top10 forecasts (yellow cells in Table 2) at any horizon. Results for the *disaggregated* forecasts are a bit disappointing,[56] since they beat the benchmark (ARMA) forecast in just a few cases. The results for the *hybrid* models seem to be a little more promising at some horizons.[57]

In turn, the Phillips curves and the factor models (especially for $h > 9$) enter more often into the set of Top10 forecasts. Focus and BEI also belong to the set of best forecasts, in several cases. Regarding combinations, the third set of forecasts (set3, including Focus and BEI in the pool of forecasts) clearly provides a superior[58] information set, when compared to set1 and set2. In this sense, the quality of the information set embodied in the pool of forecasts seems to be more important here than the forecast combination method[59] used to weight the individual forecasts (i.e., there is no clear winner method in comb3 to be used in all horizons).

[54] For medium/long horizons, note QRF shows a non-neglible accuracy improvement over RF, which is due to the role quantiles play at improving conditional mean forecasts.

[55] Deep learning models usually stand out when based on a large database, which is not necessarily the case in our monthly series setup.

[56] The gains of separetely forecasting inflation components seems not to offset here model misspecification and parameter uncertainty, among others, when estimating multiple individual models.

[57] In both inflation rates, Ada-OLS usually performs worser than Adalasso (and RF-OLS worser than RF), thus suggesting nonlinearities are important to forecast inflation. By comparing Adalasso with RF, there is no clear indication of the best variable selection method (Adalasso dominates RF-Ada in many horizons, whereas RF usually dominates Ada-RF for $h \geq 6$). However, when including XGBoost, the results become crystal clear for the 12-month accumulated inflation: XGBoost dominates both Ada-XGB and RF-XGB in all horizons (similar results are obtained, in many cases, with monthly inflation), thus suggesting that XGBoost is a strong variable selection method.

[58] Recall that market professionals devote considerable resources to inflation forecasting.

[59] Regarding the use of machine learning methods as forecast combination devices, results are not so encouraging when compared to more traditional combination approaches.

Now considering the inflation rate in twelve months (right panel in Table 2), note the excellent performance of BEI and Focus, which belong to the set of Top10 forecasts in almost all horizons. Besides, both forecasts statistically beat the benchmark (at least) at 10% level, using the Diebold-Mariano test, from $h = 1$ to 9 months. The other forecasts that can also beat the benchmark (but only for horizons up to three months) are the forecast combinations (comb3, excepting model 50) based on the third set of forecasts. Again, comb3 dominates the other two sets of combinations at short/medium horizons, which can be attributed to quality of the information set (e.g., BEI and Focus) coupled with the MCS method.

Again, stands out the superior accuracy of the machine learning forecasts, compared to great part of the traditional forecasting approaches. For instance, for horizons from $h = 12$ to 18 months, the tree-based methods dominate the other competing methods (with a few exceptions) in set1 of forecasts (models 1 to 27). In particular, for the longest horizon ($h = 18$), the XGBoost method is the best among all the 50 candidates, providing an accuracy gain of 66%, in terms of the R^2oos statistic, compared to the benchmark (ARMA) model.

The MSEs from the robustness exercises, presented in Appendix F, in general, lead to similar conclusions. For instance, factor models show the best results among traditional models, machine learning methods perform better at medium/long horizons, and forecast combination including all individual forecasts, and using the MCS approach, delivers an improved accuracy when compared to the other sets of forecasts.

Altogether, these results complement the ones previously reported by the inflation forecasting literature. For instance, regarding Brazilian inflation, Medeiros et al. (2016) reports that machine learning (lasso-based) methods show the smallest errors at short-horizons, whereas the AR is the best model for long horizons, followed by the factor model, in some cases. Garcia et al. (2017) documents the superiority of the CSR method, compared to alternative ML methods, and argues that forecast combination based on model confidence sets can achieve superior predictive performances. Regarding U.S. inflation, Medeiros et al. (2021) points out that random forest is the ML method that deserves more attention, since it dominates all other models.

Next, we deepen the forecast error analysis[60] by investigating the trade-off between bias and variance.[61] Following Lima and Meng (2017), we decompose the MSE into two parts: the forecast variance and the squared forecast bias. To do so, we calculate the MSE of any forecast

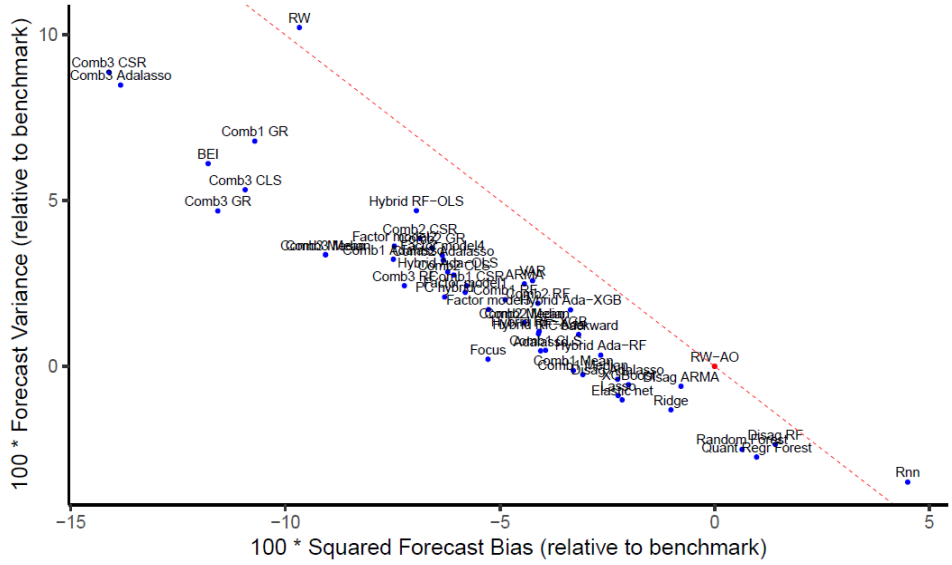
[60] According to Ng (2015), time spent coming up with diagnostics for learning algorithms is time well spent.

[61] Underfitting usually occurs when a model is too simple (e.g., few predictors), showing low forecast variance but more bias towards wrong outcomes. In turn, models that are more complex are often able to reduce the bias, but at the cost of a higher forecast variance. This trade-off between models too simple (high bias) versus too complex (high variance) is a key issue in statistics and affects all supervised ML methods.

$\widehat{f_{y_{t+h}}}$ as $\frac{1}{T^*}\sum_t \left(y_{t+h} - \widehat{f_{y_{t+h}}}\right)^2$ and the forecast variance as $\frac{1}{T^*}\sum_t \left(\widehat{f_{y_{t+h}}} - \frac{1}{T^*}\sum_t \widehat{f_{y_{t+h}}}\right)^2$, where T^* is the number of out-of-sample observations. The squared forecast bias is, then, computed as the difference between MSE and the forecast variance.

Figure 6 shows the relative forecast variance and squared forecast bias of all 50 forecasting methods considering the monthly inflation rate as dependent variable. This analysis is particularly important in model selection and helps understanding why some methods display a better forecast accuracy compared to others. The relative forecast variance (squared bias) is calculated as the difference between the forecast variance (squared bias) of the i-th model and the forecast variance (squared bias) of the moving-average approach RW-AO. Thus, the value of relative forecast variance (squared bias) for the RW-AO is necessarily equal to zero. Moreover, each point on the red line represents a forecast with the same MSE as the RW-AO. Points to the right of the line are forecasts outperformed by the RW-AO, and points to the left represent forecasts that outperform the RW-AO. Since the RW-AO is a simple moving average of inflation, it will have a low forecast variance but will likely be biased.

Figure 6 - Scatterplot of relative forecast variance and squared forecast bias ($h = 1$)



Notes: Each point on the red dotted line represents a forecast with the same MSE as the RW-AO; points to the right are forecasts outperformed by the RW-AO, and points to the left represent forecasts that outperform the RW-AO.

Note that for $h = 1$ all forecasts (excepting RW and RNN) outperform the RW-AO. Since forecast variances are generally greater than RW-AO's variance (note the blue dots are usually above zero in the vertical axis of Figure 6), the higher accuracy compared to RW-AO relies almost exclusively on a method's ability to lower the bias relative to the RW-AO. Also note

that tree-based methods deliver a lower forecast variance, essentially, due to data bootstrapping in multiple tree-learning, besides a random subset of variables used in each tree.

On the other hand, forecast combinations seem to increase the forecast variance, when compared to individual forecasting methods. In particular, the best method for $h = 1$ (Comb3 GR) delivered the lowest MSE by reducing the bias (e.g., compared to Focus) without increasing too much the variance. The main message is that the forecasting methods that yield a sizeable reduction in the forecast bias, while keeping variance under control, are able to improve forecasting accuracy over the lowest-variance approach (RW-AO).

The MSE decomposition for other cases are shown in Appendix G. For the twelve-month accumulated inflation ($h = 12$ or 18), the tree-based methods again show a lower variance, whereas forecast combinations in general exhibit a sizeable bias (one possible reason is multicollinearity at long horizons, due to highly correlated forecasts).[62]

The previous analysis enabled a discussion on relative (average) forecast accuracy. However, such measures alone do not convey any information on how the performance of the competing methods evolves *over time*. To tackle this issue, we compute the Cumulative Squared Prediction Error (CSPE) of each method, compared to the benchmark, along the pseudo out-of-sample exercise; see Rapach et al. (2010) and Lima and Meng (2017).

Figure 7 shows the differences over time between the CSPEs of the benchmark (ARMA) and each competing method. When the curve in each graph increases, the considered method outperforms the benchmark, while the opposite holds when the curve decreases. In addition, if the curve is higher at the end of the evaluation period, the method has a lower MSE compared to the benchmark, considering all out-of-sample observations.

Note in Figure 7 the best forecast (model 46) consistently outperformed the benchmark along the out-of-sample observations (smooth decline of the blue line). On the other hand, for several methods (e.g., BEI and Comb3 CSR) there is a concentrated accuracy loss, compared to benchmark (sharp increase of the blue line), at the beginning of 2020 (probably due to the covid-19 pandemic, and the higher uncertainty about the future of economy).[63] In other cases, such as RF and QRF, note the blue line fluctuates slightly above the zero horizontal line, thus indicating, for $h = 1$, a bit worse performance of such methods in comparison to the benchmark.

[62] Recall that benefits of combination come from a set of forecasts containing low pairwise correlations.

[63] Appendix G shows the CSPE curves for other cases. In Figure G4, note the top forecasts improve accuracy over the benchmark, essentially, by the end of 2016 until 2018, keeping a stable performance outside this period.

Figure 7 - Cumulative Square Prediction Error (CSPE) for $h = 1$ (IPCA % p.m.)

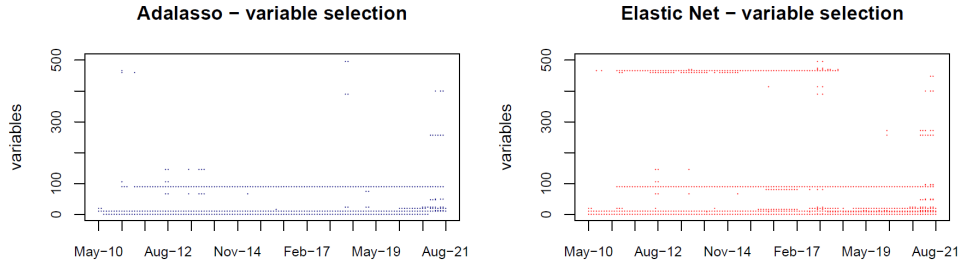
Notes: A positively sloped curve in each panel indicates that the conditional model is outperformed by the benchmark, while the opposite holds for a downward sloping curve. Moreover, if the curve is positive (negative) at the end of the period, then the competing method has a higher (lower) MSE than the benchmark over the evaluation period.

Another interesting analysis is the identification of the most important variables chosen by the machine learning methods to predict inflation. A first approach is to track the number of variables selected (or not) over time, along the pseudo out-of-sample forecasting exercise. Figure 8 reveals, for illustrative purposes, among the 501 potential predictors for inflation, which ones were indeed selected (and when), according to the Adalasso and Elastic Net methods.

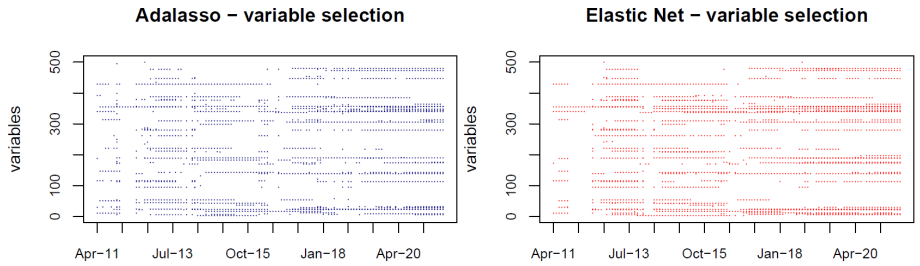
Figure 8 shows how the selection procedure works over time. The horizontal axis represents the end of the estimation sample, along the out-of-sample forecasting exercise, and the vertical axis denotes all the 501 regressors. A blue dot indicates that variable i has a non-zero coefficient in the adalasso estimation (a red dot, in the elastic net) with sample ending at period t, used to build forecasts for y_{t+h}. This allows us to discover how the models change in response to different economic conditions over time.[64] In other words, Figure 8 shows that the statistical significance of the coefficients vary considerably over time for some variables, while staying relatively stable for several others. For instance, note for $h = 1$ that adalasso quite regularly selects 3 variables along the forecasting exercise. As later revealed in Figure 9, these variables are the IPCA headline and IPC-Fipe inflation rates, lagged one month, both representing the inertial component of inflation, besides the commercial consumption of electricity, also lagged one month, which is a variable directly related to economic activity and aggregate demand.

Figure 8 - Variable selection over time

$h = 1$ (IPCA % p.m.)



$h = 12$ (IPCA % 12 months)



[64] In Appendix G, Figure G5 shows the average number of variables selected by lasso, adalasso and elastic net. Overall, elastic net selects more variables than lasso and adalasso (probably due to the grouping effect; see Zou and Hastie, 2005). In turn, adalasso is more parsimonious, compared to the other two methods. On the other hand, fewer variables are selected when the dependent variable is the monthly inflation rate, in comparison to the 12-month accumulated rate (naturally more autocorrelated compared to the monthly rate).

Now, we identify which variables are more frequently chosen by some machine learning algorithms. Although we do not attempt to economically interpret the driving-forces behind the ML forecasts here, further inspecting these models allows us to better understand how they are making forecasts, which may reveal new statistical relationships in the data previously overlooked by standard linear models.

Figure 9 shows *word cloud* graphs, which are images composed by the names of variables[65] most frequently selected by a given method, where the size of each word indicates the frequency a variable is selected (e.g., non-zero coefficient) along the recursive estimations in the out-of-sample forecasting exercise. Thus, most frequent variables have larger font size, whereas variables with the same frequency of selection have the same size and color.

Figure 9 - Word cloud (*frequency*), adalasso (left) and elastic net (right)

$h = 1$ (IPCA % p.m.)

$h = 12$ (IPCA % 12 months)

Note: The size of each word indicates the frequency a variable is selected along the out-of-sample forecasting exercise.

Variables with the same frequency of selection have the same size and color.

The variables identified by adalasso and elastic net (for $h = 1$) include past inflation (inertial inflationary dynamics) and variables related to the real economy (e.g., commercial electricity consumption, agriculture exports). Regarding the 12-month inflation rate (for $h = 12$), the set of most frequent variables now also include interest rates, fiscal variables, external sector variables (current account, dollar index, CRB food, oil price), and also new variables (not

[65] The names displayed in the word clouds are the nicknames listed in Appendix E.

traditionally used to forecast inflation), such as the temperature of the Pacific Ocean (oceanic nino index),[66] due to the role that the *El Niño* and *La Niña* might play in food inflation.

Another way to inspect the ML results is to build the so-called *variable importance* (or feature importance) graphs. The idea is again to build a rank of variables, but now based on their usefulness in predicting inflation at a given horizon.[67] Word clouds can again be used to summarize the results.

For the penalized-regression models (elastic net, lasso, adalasso and ridge), we compute the rank of variable importance based on the absolute value of the estimated coefficients (adjusted for the original scale of each variable) multiplied by the standard deviation of the respective variable. For tree-based methods, the rank can be computed using either the methods of permutation or impurity (used here); see Appendix D for further details.

Figure 10 presents variable importance results for monthly inflation (selecting the best methods in Table 2, among models 11-17), whereas Appendix G shows some results for the 12-month accumulated inflation.

Figure 10 - Word cloud (*importance*), selected models, IPCA % p.m.

$h = 1$, elastic net (left), random forest (right)

$h = 3$, ridge regression (left), random forest (right)

Note: The size of each word indicates the variable importance. Most relevant variables have larger font size and variables with the same importance have the same size and color.

[66] This series is the Oceanic Niño Index (ONI), provided by the Climate Prediction Center, linked to the National Oceanic and Atmospheric Administration (NOAA, USA).

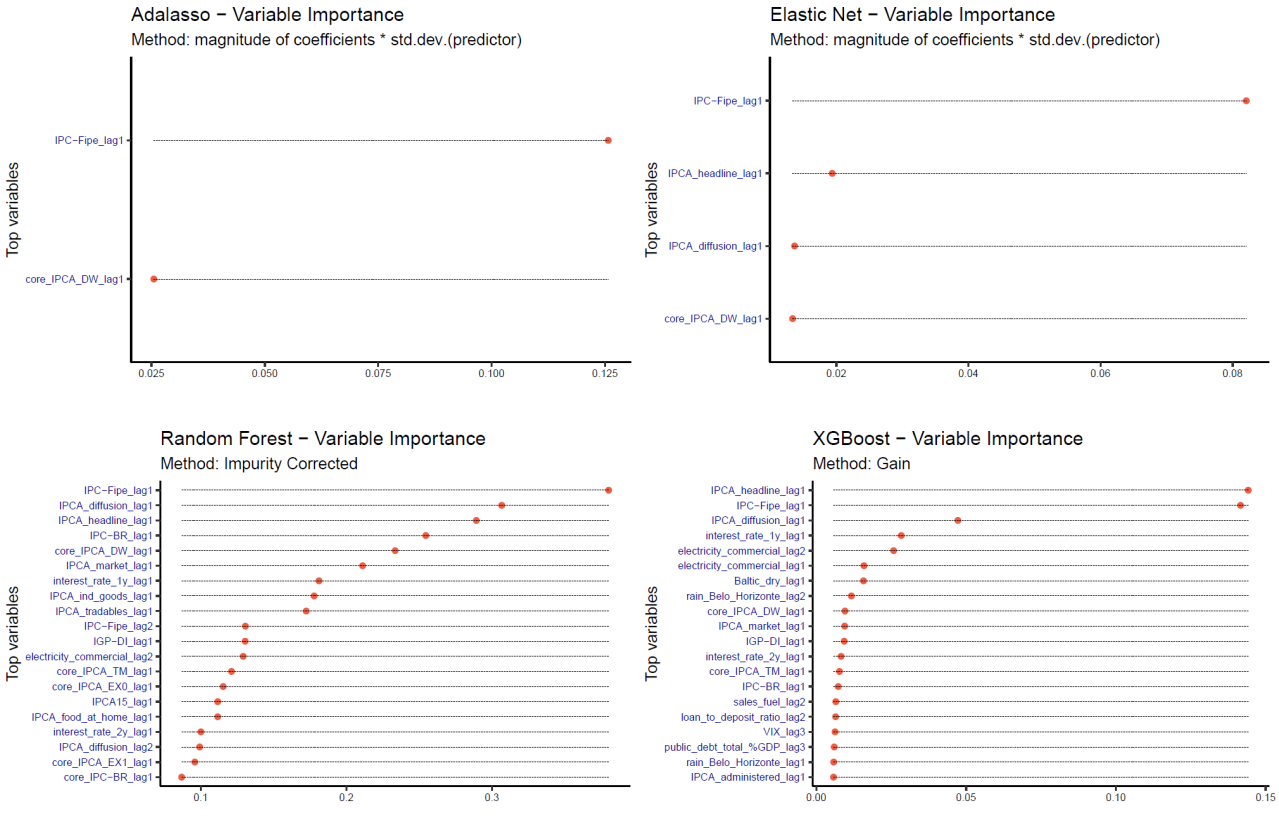
[67] Variable (or feature) importance refers here to techniques that assign *scores* to variables based on how useful they are at improving the forecast accuracy of a given model.

Figures 11-12 also show variable importance results, but only considering the Top 20 most important variables. Note that in some cases (e.g., adalasso, $h = 1$), there are just a few relevant predictors employed by the forecasting method.

In Figures 10 or 11, considering $h = 1$ and monthly inflation rate, the lagged inflation measured by the IPC–Fipe stands out as the most relevant predictor according to three methods (adalasso, elastic net and random forest), being second-place in XGBoost. For $h = 3$ (Figure 10), the one-year interest rate gains relevance, together with variables related to the real economy and the external sector, besides unusual predictors of Brazilian inflation, such as the amount of rainfall in some Brazilian cities. Also note in Figure 10 that penalized-regressions (elastic net and ridge) give more importance to fewer predictors, compared to random forest.

On the other hand, Figure 12 shows the results for the twelve-month inflation rate, and $h = 12$, confirming that the usual variables often employed to forecast inflation in Brazil (e.g., past inflation, lagged interest rates, exchange rate and fiscal variables) are indeed the most relevant in our empirical exercise.

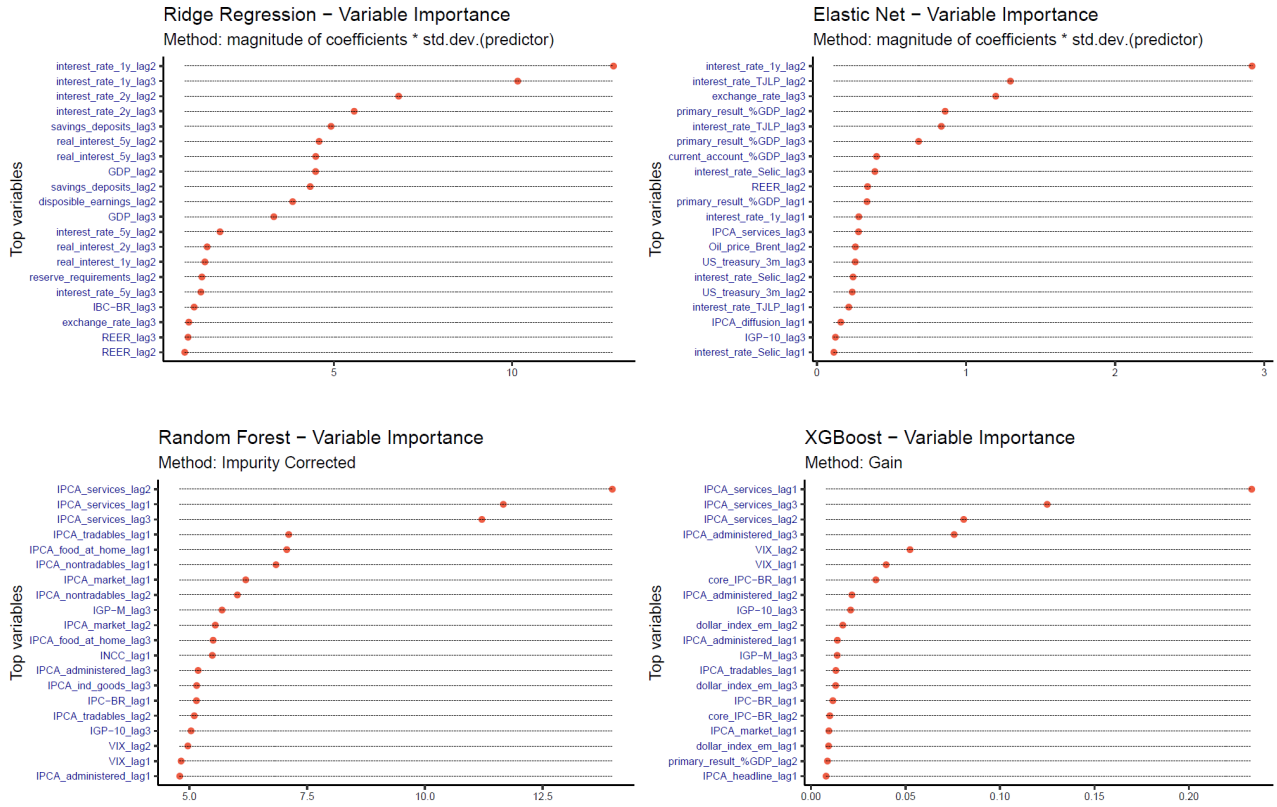
Figure 11 - Variable importance, top variables, $h = 1$, IPCA % p.m.



Note: The horizontal axis denotes the variable importance scale based on a given method.

Higher figures represent features more relevant to predict inflation.

Figure 12 - Variable importance, top variables, $h = 12$, IPCA % 12 months



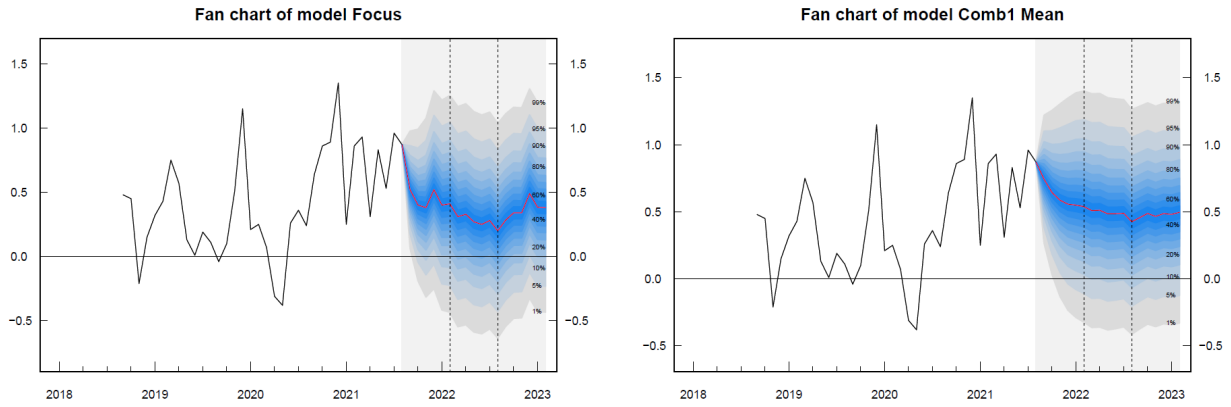
Note: The horizontal axis denotes the variable importance scale based on a given method.

Higher figures represent features more relevant to predict inflation.

Finally, we depart from the point forecast setup and build simple density forecasts associated to the point forecasts at multiple horizons. Figure 13 shows such density forecasts (fan charts) for selected methods, which allows us to conduct *risk management* analysis.[68] For example, according to model 30 (comb1 mean), the probability of the monthly inflation rate to be greater than 0.8% p.m. (per month) in December 2022 is of 20% (and to be above 1.0% p.m. is of 9%). In turn, the chance of the 12-month rate, for instance, being above 5.0% p.y. (per year) in December 2022 is of 36% according to BEI (and of 29% according to Focus); see Figure G7 in Appendix G. Such analysis can be useful when evaluating the chance of future inflation to be above/below the inflation-target.

[68] Recall that, for simplicity, we assumed a Gaussian distribution, where the conditional mean is the point forecast of a given method and the conditional variance comes from past forecast errors of such method.

Figure 13 - Fan charts (IPCA % p.m.)



4 Conclusions

Machine learning is constantly evolving with new methods being developed every day. Progress has been made in macroeconomics over the recent years on the usage of such methods coupled with big data. However, model interpretability is usually lost in such approaches.

According to Occam's razor, models should be simple and explainable. Nonetheless, outcomes from machine learning methods are not easily interpreted, for instance, due to a highly nonlinear setup or a large set of inputs.

In this paper, we tackle this issue and take a step towards transparency (turning the *black box* into a *gray box*), by providing complementary tools (e.g., word clouds, variable importance graphs and fan charts) to better understand the ML outcomes. The tools we provide for identifying the most important variables to predict inflation also allows shifting the discussion from *big data* to *good data*, in the sense that finding high-quality data is more important than the quantity of data (Ng, 2015, 2021).

In this context, we study the inflation forecasting accuracy of 50 competing methods; including some new machine learning techniques proposed here (e.g., hybrid models and a quantile-combination method based on quantile regression forest), traditional econometric models (e.g., ARMA, VAR), reduced-form structural models (Phillips curves), factor models, survey-based forecasts, regularization procedures (e.g., elastic net, lasso and ridge), and forecast combinations, among others.

The variable of interest is the Brazilian inflation as measured by the IPCA. In order to evaluate the predictive power of each method, we conduct a pseudo out-of-sample empirical exercise (*horse-race*) based on 501 time series, coming from 167 macroeconomic and financial variables, where each method produces point forecasts for horizons $h = 1, ..., 18$ months ahead.

According to Wolpert (1996), there is no universal best model. In other words, the set of assumptions that works well in one domain may work poorly in another framework. The empirical results documented in this paper go in this direction, suggesting that some machine learning algorithms are, indeed, able to consistently outperform traditional econometric models in terms of MSE. However, there is no supreme model for all cases, since the performance depends on the forecast horizon and whether inflation is measured by its monthly rate or accumulated in twelve months.

The main takeaways are the following: (i) ML methods, often designed to work under low *noise-to-signal* ratio setups (e.g., image classification, voice recognition) and large datasets, can do a pretty good job under medium/high noise-to-signal ratio and a dataset not so large in time-dimension (e.g., applied macroeconomics); (ii) ML methods consistently beat the benchmark (ARMA) model and, in many cases, exhibit two-digit accuracy gains in terms of the R^2 out-of-sample statistic; (iii) nonlinearities captured by ML methods (Varian, 2014), such as recurrent neural networks or random forest, are important to forecast inflation in Brazil; (iv) at shorter horizons, forecast combinations are useful, especially when using big data to build individual forecasts and the model confidence set of Hansen et al. (2011) to select the superior forecasts to be combined; (v) at longer horizons, tree-based methods, such as random forest and XGBoost, perform quite well and dominate other models in several cases; (vi) Focus and BEI also belong to the set of top forecasts in many horizons (for monthly inflation, they improve the quality of the information set used as input in combinations, whereas for the 12-month accumulated inflation rate they belong to the set of top forecasts in almost all horizons); (vii) XGBoost is a competing variable selection method; and (viii) fan charts can easily be constructed from point forecasts, which allows estimating the probability of future inflation to be above/below the inflation-target.

To sum it up, we analyze machine learning methods that forecasters should have in their toolkit when predicting inflation in Brazil. Moreover, we reveal the top features that an econometrician should have in mind when building inflation forecast models. These findings represent a valuable contribution to academics, practitioners and policymakers interested in macroeconomic forecasting using machine learning, in particular, focused on Brazilian inflation.

References

[1] Altmann, A., Tolosi, L., Sander, O., Lengauer, T., 2010. Permutation importance: a corrected feature importance measure. *Bioinformatics* 26, 1340-1347.

[2] Ang, A., Bekaert, G., Wei, M., 2007. Do Macro Variables, Asset Markets or Surveys Forecast Inflation Better? *Journal of Monetary Economics* 54, 1163-1212.

[3] Apostol, T.M., 1967. *Calculus, Vol. 1: One-Variable Calculus with an Introduction to Linear Algebra* (2nd Ed.), New York: John Wiley & Sons.

[4] Araujo, G.S., Vicente, J.V.M., 2017. Estimação da Inflação Implícita de Curto Prazo. Working Paper n. 460, Banco Central do Brasil.

[5] Arlot, S., Celisse, A., 2010. A survey of cross-validation procedures for model selection. *Statistics Surveys* 4, 40-79.

[6] Arruda, E., Ferreira, R., Castelar, I., 2011. Modelos lineares e não lineares da curva de phillips para previsão da taxa de inflação no brasil. *Revista Brasileira de Economia* 65, 237-252.

[7] Atkeson, A., Ohanian, L., 2001. Are Phillips curves useful for forecasting inflation? *Federal Reserve Bank of Minneapolis Quarterly Review* 25, 2-11.

[8] Bai, J., Ng, S., 2002. Determining the number of factors in approximate factor models. *Econometrica* 70, 191-221.

[9] Bai, J., Ng, S., 2008. Forecasting economic time series using targeted predictors. *Journal of Econometrics* 146, 304-317.

[10] Baker, S.R., Bloom, N., Davis, S.J., 2015. Measuring Economic Policy Uncertainty. NBER Working Paper 21633, National Bureau of Economic Research.

[11] Bańbura, M., Giannone, D., Modugno, M., Reichlin, L., 2013. Now-casting and the real-time data flow. Working Paper Series n.1564, European Central Bank.

[12] Bates, J.M., Granger, C.W.J., 1969. The combination of forecasts. *Operational Research Quarterly* 20, 451-468.

[13] BCB – Banco Central do Brasil, 2021. New small-scale disaggregate model. Inflation Report – March 2021.

[14] Breiman, L., 2001. Random forests. *Machine Learning* 45, 5-32.

[15] Cerulli, G., Drago, C., 2021. An Introduction to Machine Learning and Text Mining for economists using Stata, Python and R. Mimeo, available at: https://www.side-iea.it/events/courses/introduction-machine-learning-and-text-mining-economists-using-stata-python-and-r

[16] Chen, T., Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System. The 22nd SIGKDD Conference on Knowledge Discovery and Data Mining. Mimeo.

[17] Cheng, K., Huang, N., Shi, Z., 2019. Survey-Based Forecasting: To Average or Not To Average. Mimeo, available at: https://www.researchgate.net/publication/336141912

[18] Costa, A.B.R., Ferreira, P.C.G., Gaglianone, W.P., Guillén, O.T.C., Issler, J.V., Lin, Y., 2021. Machine Learning and Oil Price Point and Density Forecasting. *Energy Economics* 102, 105494.

[19] Diebold, F.X., Mariano, R.S., 1995. Comparing Predictive Accuracy. *Journal of Business and Economic Statistics* 13, 253-263.

[20] Dupond, S., 2019. A thorough review on the current advance of neural network structures. *Annual Reviews in Control* 14, 200-230.

[21] Elliott, G., Gargano, A., Timmermann, A., 2013. Complete subset regressions. *Journal of Econometrics* 177(2), 357-373.

[22] Elliott, G., Gargano, A., Timmermann, A., 2015. Complete subset regressions with large-dimensional sets of predictors. *Journal of Economic Dynamics and Control* 54, 86-110.

[23] Elman, J.L., 1990. Finding structure in time. *Cognitive Science* 14(2), 179-211.

[24] Faust, J., Wright, J.H., 2013. Forecasting Inflation. Handbook of Economic Forecasting, vol. 2A, Chapter 1, 3-56. Ed. Elsevier B.V.

[25] Ferreira, M.A., Santa-Clara, P., 2011. Forecasting stock market returns: The sum of the parts is more than the whole. *Journal of Financial Economics* 100(3), 514-537.

[26] Forni, M., Hallim, M., Lippi, M., Reichlin, L., 2000. The generalized dynamic factor model: Identification and estimation. *Review of Economics and Statistics* 82, 540-554.

[27] Forni, M., Hallin, M., Lippi, M., Reichlin, L., 2003. Do financial variables help forecasting inflation and real activity in the euro area? *Journal of Monetary Economics* 50, 1243-1255.

[28] Gaglianone, W.P., Giacomini, R., Issler, J.V., Skreta, V., 2021. Incentive-driven Inattention. *Journal of Econometrics* (*in press*).

[29] Gaglianone, W.P., Guillén, O.T.C., Figueiredo, F.M.R., 2018. Estimating Inflation Persistence by Quantile Autoregression with Quantile-Specific Unit Roots. *Economic Modelling* 73(C), 407-430.

[30] Gaglianone, W.P., Issler, J.V., Matos, S.M., 2017. Applying a Microfounded-Forecasting Approach to Predict Brazilian Inflation. *Empirical Economics* 53(1), 137-163.

[31] Garcia, M.G.P., Medeiros, M.C., Vasconcelos, G.F.R., 2017. Real-time inflation forecasting with high-dimensional models: The case of Brazil. *International Journal of Forecasting* 33, 679-693.

[32] Granger, C.W.J., Ramanathan, R., 1984. Improved methods of combining forecasting. *Journal of Forecasting* 3, 197-204.

[33] Hall, A.S., 2018. Machine Learning Approaches to Macroeconomic Forecasting. *Federal Reserve Bank of Kansas City Economic Review*, 4th quarter of 2018, 63-81.

[34] Hansen, B.E., 2019. Econometrics. Mimeo, available at: https://www.ssc.wisc.edu/~bhansen/econometrics/Econometrics.pdf

[35] Hansen, P.R., Lunde, A., Nason, J.M., 2011. The model confidence set. *Econometrica* 79(2), 453-497.

[36] Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd edition. Springer-Verlag, New York.

[37] Hoerl, A.E., Kennard, R.W., 1970. Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* 12(1), 55-67.

[38] Janitza, S., Celik, E., Boulesteix, A.-L., 2018. A computationally fast variable importance test for random forests for high-dimensional data. *Advances in Data Analysis and Classification* 12(4), 885-915.

[39] Jiang, C., Maasoumiz, E., Xiao, Z., 2020. Quantile aggregation and combination for stock return prediction. *Econometric Reviews* 39(7), 715-743.

[40] Judge, G.G., Hill, R.C., Griffiths, W.E., Lütkepohl, H., Lee, T.-C., 1988. *Introduction to the Theory and Practice of Econometrics*. New York, Wiley.

[41] Jung, J.K., Patnam, M., Ter-Martirosyan, A., 2018. An Algorithmic Crystal Ball: Forecasts-based on Machine Learning. IMF Working Paper WP/18/230.

[42] Koenker, R., 2005. *Quantile Regression*. Cambridge University Press.

[43] Kohlscheen, E., 2012. Uma nota sobre erros de previsão da inflação de curto prazo. *Revista Brasileira de Economia* 66, 289-297.

[44] Kohlscheen, E., 2021. What does machine learning say about the drivers of inflation? BIS Working Papers n. 980, Bank for International Settlements.

[45] Lima, L.R., Meng, F., 2017. Out-of-sample return predictability: a quantile combination approach. *Journal of Applied Econometrics* 32(4), 877-895.

[46] Lundberg, S., Lee, S., 2017. A unified approach to interpreting model predictions. arXiv:1705.07874.

[47] Machado, V., Portugal, M., 2014. Measuring inflation persistence in Brazil using a multivariate model. *Revista Brasileira de Economia* 68 (2), 225-241.

[48] Marcellino, M., Stock, J., Watson, M., 2006. A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series. *Journal of Econometrics* 135, 499-526.

[49] McCracken, M.W., Ng, S., 2015. FRED-MD: A Monthly Database for Macroeconomic Research. Working Paper 2015-012B, Federal Reserve Bank of St. Louis.

[50] Medeiros, M., Mendes, E., 2016. L1-regularization of high-dimensional time-series models with flexible innovations. *Journal of Econometrics* 191, 255-271.

[51] Medeiros, M., Vasconcelos, G., de Freitas, E.H., 2016. Forecasting Brazilian Inflation with High Dimensional Models. *Brazilian Review of Econometrics* 36(2), 223-254.

[52] Medeiros, M., Vasconcelos, G., Veiga, A., Zilberman, E., 2021. Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods. *Journal of Business & Economic Statistics* 39(1), 98-119.

[53] Meinshausen, N., 2006. Quantile Regression Forests. *Journal of Machine Learning Research* 7, 983-999.

[54] Minella, A., de Freitas, P.S., Goldfajn, I., Muinhos, M.K., 2003. Inflation targeting in Brazil: constructing credibility under exchange rate volatility. *Journal of International Money and Finance* 22 (7), 1015-1040.

[55] Morales-Arias, L., Moura, G.V., 2013. Adaptive forecasting of exchange rates with panel data. *International Journal of Forecasting* 29, 493-509.

[56] Morde, V., Setty, V.A., 2019. XGBoost Algorithm: Long May She Reign! Mimeo.

[57] Nembrini, S., Koenig, I.R., Wright, M.N., 2018. The revival of the Gini Importance? *Bioinformatics* 34(21), 3711-3718.

[58] Newey, W.K., West, K.D., 1987. A Simple, Positive Semi-Definite, Het- eroskedasticity and Autocorrelation Consistent Covariance Matrix. *Econometrica* 55, 703-708.

[59] Ng, A., 2015. Advice for applying machine learning. Mimeo, available at: https://see.stanford.edu/materials/aimlcs229/ML-advice.pdf.

[60] Ng, A., 2021. Introduction to machine learning in production. Mimeo, available at: https://pt.coursera.org/lecture/introduction-to-machine-learning-in-production/from-big-data-to-good-data-qGI5Y

[61] Nowotarski, J., Raviv, E., Trück, S., Weron, R., 2014. An empirical comparison of alternative schemes for combining electricity spot price forecasts. *Energy Economics* 46, 395-412.

[62] Rapach, D. E., Strauss, J. K., & Zhou, G., 2010. Out-of-Sample Equity Premium Prediction: Combination Forecasts and Links to the Real Economy. *The Review of Financial Studies* 23(2), 821-862.

[63] Samuel, A.L., 1959. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development* 3(3), 210-229.

[64] Shang, H.L., Haberman, S., 2018. Model confidence sets and forecast combination: an application to age-specific mortality. *Genus* 74(19), 1-23.

[65] Smyl, S., 2020. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting* 36, 75-85.

[66] Stock, J., Watson, M., 1999. Forecasting inflation. *Journal of Monetary Economics* 44, 293-335.

[67] Stock, J., Watson, M., 2002. Forecasting Using Principal Components from a Large Number of Predictors. *Journal of the American Statistical Association* 97(460), 1167-1179.

[68] Svensson, L., 1994. Monetary policy with flexible exchange rates and forward interest rates as indicators. Institute for International Economic Studies, Stockholm University. Mimeo.

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

[70] Tealab, A., 2018. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal* 3(2), 334-340.

[71] Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society* 58(1), 267-288.

[72] Val, F.F., Araujo, G.S., 2019. Breakeven Inflation Rate Estimation: an alternative approach considering indexation lag and seasonality. Working Paper n. 493, Banco Central do Brasil.

[73] Varian, H.R., 2014. Big data: New tricks for econometrics. *Journal of Economic Perspectives* 28(2), 3-28.

[74] Wolpert, D., 1996. The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation* 8, 1341-1390.

[75] Zou, H., 2006. The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association* 101 (476), 1418-1429.

[76] Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society* 67(2), 301-320.

[77] Zou, H., Hastie, T., Tibshirani, R., 2007. On the degrees of freedom of the lasso. *The Annals of Statistics* 35, 2173-2192.

– Appendix –

"Machine learning methods for inflation forecasting in Brazil: new contenders versus classical models"

Gustavo Silva Araujo [69]

Wagner Piazza Gaglianone [70]

March 2022

Contents:

[69] Research Department, Banco Central do Brasil, and Ibmec-RJ. E-mail: gustavo.araujo@bcb.gov.br

[70] Corresponding author. Research Department, Banco Central do Brasil, and FGV Crescimento & Desenvolvimento. E-mail: wagner.gaglianone@bcb.gov.br

Appendix A. Further details on *elastic net*

There are well-established methods for choosing tuning parameters (λ, α). For instance, K-fold *cross-validation* (CV) is a popular method for computing the prediction error and comparing different models using training data. The loss often used for cross-validation is the mean squared-error (MSE). The goal is to produce the so-called cross-validation curve, built by computing the MSE as a function of the tuning parameter λ chosen over a pre-selected grid.[71]

Note that in the elastic net there are two tuning parameters, so one would need to cross-validate the model on a two-dimensional surface. The minimum MSE, thus, provides the pair (λ, α) to be used in the final model estimation. Parameters can be estimated using the penalized maximum likelihood, in which the regularization path (i.e., the path of each coefficient β_j against, for instance, the l_1-norm of the whole coefficient vector as λ varies) can be computed.

Another way to choose the tuning parameters is to employ *information criteria*. For example, Zou et al. (2007) show that one can consistently estimate the degrees of freedom of the lasso model using information criteria as alternative to cross-validation. An advantage of such procedure is that selecting the model using information criterion is much faster than using cross-validation.

More importantly, performing cross-validation in a time-series context is quite challenging, since data are (usually) not independent and identically distributed (i.i.d.). In other words, traditional cross-validation methods are not appropriate for time series dataset, since temporal dependency imposes correlation in the time dimension, meanwhile K-fold cross validation assumes i.i.d. amid samples (Arlot and Celisse, 2010). Moreover, the temporal dependencies in the CV approach would split the dataset randomly, loosing the chronological order of observations, which is troublesome in forecasting, because one would be using the future to predict the past (Tashman, 2000). See also Medeiros et al. (2016) for further details.

In this paper, since we are mainly interested in a forecasting exercise using time series data, we select the best lasso, adalasso and elastic net models using an information criterion procedure (BIC – Bayesian Information Criterion).

[71] The CV algorithm splits the training set of observations in two parts: *training fold* (used for the estimation of parameters) and *test fold* (based on the remaining observations, used for model predictions). Then, forecast errors are computed and used to calculate the MSE over the entire set of predictions using K-folds. See Jung et al. (2018) for further details.

Appendix B. Further details on *random forest*

First, we investigate how to properly grow a *regression tree*.[72] The algorithm needs to automatically decide on both the splitting variables and split points. In the example shown in Figure 1, if one assumes a mean-squared error loss function, the optimal $\widehat{c_m}$ is simply the average of the response Y in the region R_m. However, finding the best partition in terms of overall MSE, according to Hastie et al. (2009), is usually computationally infeasible. In this sense, the authors propose the following approach, focused on the implementation of CART (classification and regression tree) models:

(i) consider a splitting variable j and split point s, and define the pair of half-planes:

$$R_1(j,s) = \{X \mid X_j \leq s\} \quad \text{and} \quad R_2(j,s) = \{X \mid X_j > s\}, \tag{16}$$

(ii) find the splitting variable j and split point s that solve the minimization problem:

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right], \tag{17}$$

where the previous inner minimizations, for any choice j and s, can be solved by:

$$\widehat{c_1} = \mathbb{E}(y_i \mid x_i \in R_1(j,s)) \quad \text{and} \quad \widehat{c_2} = \mathbb{E}(y_i \mid x_i \in R_2(j,s)). \tag{18}$$

Note that for a given splitting variable, the calculation of the optimal split point s can be easily done. Thus, by searching through all covariates, the determination of the best pair (j,s) is feasible. Then, based on the best split, we divide the data into the two resulting regions R_1 and R_2 and repeat the splitting process on each of the two regions. This process is repeated on all of the resulting regions. To sum it up, the regression tree can be estimated by repeating the three steps below, for each terminal node of the tree, until the minimum number of observations at each node is achieved:

(1) randomly select m out of p covariates as possible split variables;[73]

[72] According to Hansen (2019): *"The literature on regression trees has developed some colorful language to describe the tools, based on the metaphor of a living tree. 1. A split point is node. 2. A subsample is a branch. 3. Increasing the set of nodes is growing a tree. 4. Decreasing the set of nodes is pruning a tree."*

[73] The size of a tree is a tuning parameter governing the model's complexity, and the optimal size should be adaptively chosen from the data. The preferred strategy is to stop the splitting process when some minimum node size is reached. Typically, for regression problems with p predictors, the literature recommends to use $m = p/3$ (rounded down) in each split, with a minimum node size of 5 as the default. Note the reduction of the tuning parameter m will, in general, reduce the correlation between any pair of trees. See Hastie et al. (2009, chapter 15.3) for more details.

(2) select the best variable/split point among the m candidates;

(3) split the node into two child nodes.

Next, we represent mathematically the *random forest* model, following the discussion in Meinshausen (2006): consider n independent observations (Y_i, X_i), for $i = 1, ..., n$, and let θ be the random parameter vector that determines how a tree $T(\theta)$ is grown, that is, characterizes the tree in terms of split variables, cut-points at each node, and terminal-node values. Also, let $\Im$ be the space in which X lives, that is $X : \Omega \to \Im$, where $\Im \subseteq \mathbb{R}^p$ and $p \in \mathbb{N}_+$ is the dimensionality of the set of covariates X.

Every leaf of a tree (terminal node) $l = 1, ..., L$ corresponds to a subspace of $\Im$, that is $R_l \subseteq \Im$. For every $x \in \Im$, there is one (and only one) leaf l such that $x \in R_l$ (corresponding to the leaf that is obtained when dropping x down the tree). Denote this leaf by $l(x, \theta)$ for tree $T(\theta)$. The prediction of a single tree $T(\theta)$ conditioned on $X = x$ is obtained by averaging over the observed values in leaf $l(x, \theta)$. Let the weight vector $w_i(x, \theta)$ be given by a positive constant if observation X_i is part of leaf $l(x, \theta)$ and 0 if it is not. The weights sum to one, such that:

$$w_i(x, \theta) = \frac{\mathbf{1}_{\{X_i \in R_{l(x,\theta)}\}}}{\sum_{j=1}^{n} \mathbf{1}_{\{X_j \in R_{l(x,\theta)}\}}}. \tag{19}$$

The forecasting model based on a single *regression tree*, conditioned on a covariate $X = x$, is then the weighted average of the original observations Y_i, for all $i = 1, ..., n$, that is:

$$\mathbb{E}_{\text{regression tree}}(Y \mid X = x) = \sum_{i=1}^{n} w_i(x, \theta) Y_i. \tag{20}$$

Note that conditional on the knowledge of the subregions R_l, for $l = 1, ..., L$, the relationship between inflation Y and the set of covariates X in equation (1) is approximated here by a piecewise constant model, where each leaf represents a distinct regime (see Garcia et al., 2017).

Now, using *random forests*, the conditional mean above is approximated by the averaged prediction of K single trees, each constructed with a parameter vector θ_k, $k = 1, ..., K$. Let $w_i(x)$ be the average of $w_i(x, \theta_k)$ over this collection of trees, as follows:

$$w_i(x) = \frac{1}{K} \sum_{k=1}^{K} w_i(x, \theta_k). \tag{21}$$

The prediction of random forests is, thus, the averaged response of all trees, as follows:

$$\mathbb{E}_{\text{random forest}}\left(Y \mid X = x\right) = \sum_{i=1}^{n} w_i(x) Y_i. \tag{22}$$

Note that the approximation of the conditional mean of Y conditioned on $X = x$ is given by a weighted sum over all observations. The weights vary with the covariate and tend to be large for those observations $i \in \{1, ..., n\}$ where the conditional distribution of Y, given $X = X_i$, is similar to the conditional distribution of Y given $X = x$.

Appendix C. Further details on *quantile regression forest*

The QRF algorithm proposed by Meinshausen (2006) for computing the estimate of the conditional distribution function can be summarized as follows:

(a) grow trees $T(\theta_k)$, for $k = 1, ..., K$, as in random forests. However, for every leaf (on each tree) consider all observations in the leaf, not just their average.

(b) for a given $X = x$, drop x down all trees. Compute the weight $w_i(x, \theta_k)$ of observation $i \in \{1, ..., n\}$ for every tree as in (19). Compute weight $w_i(x)$ for every observation $i \in \{1, ..., n\}$ as an average over $w_i(x, \theta_k)$, for all $k = 1, ..., K$, as in (21).

(c) compute the estimate of the distribution function as in (10) for all $y \in \mathbb{R}$, using the weights from the previous step (b).

Appendix D. Variable importance

The relative importance of individual features can be a crucial aspect to understand the outcome of a given model. The main idea is to assess the relative importance of a variable based on the amount by which the inclusion/exclusion of such variable improves/deteriorates the model's performance.

In a *big data* context, the variable importance analysis aims to provide a deeper insight into the underlying processes that generated the data. The goal is to identify the best subset of features to predict the target variable. Identifying this set of relevant predictors helps opening the so-called *black box*, thus resulting in a more interpretable model (and the model analysis efforts can be concentrated on the few most informative features).

In linear models, if one employs standardized regressors (e.g., with zero mean and unit variance), the higher the absolute value of a given coefficient, the more of an impact that feature has on the dependent variable. In other words, to show which features are the most important in a linear model, one can simply rank the standardized features according to the modulus of their coefficients.[74]

However, this simple comparison does not hold outside the linear setup. The difficulty on decomposing the importance of variables in more general models is essentially due to the nonlinear nature of the ML methods. In such cases, there are different approaches to identify and rank the most important features.

Importance on regression trees

Random forests are among the most popular machine learning methods due to their good forecasting accuracy, robustness and ease of use. In contrast to parametric methods, random forests are fully nonparametric and can deal with nonlinear effects, thus offering a great model flexibility in practical applications. Furthermore, RF can even be applied in the statistically challenging setting where the number of variables is higher than the number of observations. This makes random forests especially attractive for complex high-dimensional data applications; see Janitza et al. (2018).

Nonetheless, a suitable understanding of the *black box* mechanism behind the random forest method is of greatest importance. Nowadays, ML models are often deployed to production without a proper understanding of why exactly the algorithms make the decisions they do. As these new tools become more relevant in everyday life, model interpretability becomes one of the most important problems in machine learning these days. In particular, regarding the use of RF as a forecasting device, it is critical to comprehend the key variable interactions that are providing the predictive accuracy.

One attempt to tackle this issue is to build the so-called *variable importance* measures, by attributing scores to the variables, which reflect their relative importance in the overall model accuracy. Such measures can be used to identify relevant features, perform variable selection and quantify the prediction strength of each variable, allowing one to rank the variables according to their predictive abilities. See Hastie et al. (2009, chapter 15) for further details.[75]

[74] In this paper, we rank the modulus of the coefficients (from lasso, adalasso, ridge and elastic net) multiplied by the standard deviation of the respective variable. See https://stats.stackexchange.com/questions/14853/variable-importance-from-glmnet

[75] There are many other ways on the lookout for opening the ML black box. Just to mention a few examples: (i) Partial Dependence Plots (PDP), which show the marginal effect of a given predictor on the outcome of a ML

There are two main variable importance measures: (i) the *permutation* approach of Altmann et al. (2010); and (ii) the *impurity-corrected* method of Nembrini et al. (2018). Also, one can carry out the Janitza et al. (2018) hypothesis test of no association between the predictor and the dependent variable in both measures.

Method 1

The permutation method, also known as the mean decrease in accuracy, is one of the most common variable importance measures, and it is computed from the change in prediction accuracy when removing any association between the dependent variable and a given predictor, with large changes indicating that the predictor is important.[76] One disadvantage of the permutation approach is to produce biased outcomes when predictors are highly correlated. In addition, adding a correlated variable to the RF model can decrease the importance of another variable. Furthermore, the permutation importance is very computationally intensive in the case of high-dimensional data.

Method 2

Alternative importance measures based on impurity (i.e., how well the regression trees split the variables) are popular because they are simple, fast to compute and can be more robust to data perturbations compared with those based on permutation.[77] However, the impurity importance is known to be biased towards variables with more categories or more possible split points. Also, when the dataset has two (or more) correlated variables, any of them can be selected as predictor. Nevertheless, once one of these (correlated) variables is used as predictor, the importance of others is significantly reduced, since the impurity these other variables can decrease is already reduced by the first selected variable.[78] In this sense, Nembrini et al. (2018)

model; (ii) Surrogate Models (SM), which are auxiliary interpretable models (e.g., linear regression), built to approximate the predictions of a ML model in order to understand the (black box) outcomes by analyzing and interpreting the surrogate model's responses; and (iii) Shapley and SHAP values, which are auxiliary measures of feature importance (Lundberg and Lee, 2017).

[76] According to Nembrini et al. (2018): *"To calculate the permutation importance of the variable xi, its original association with the response y is broken by randomly permuting the values of all individuals for xi. With this permuted data, the tree-wise out-of-bag (OOB) estimate of the prediction error is computed. The difference between this estimate and the OOB error without permutation, averaged over all trees, is the permutation importance of the variable xi. This procedures is repeated for all variables of interest x1,. . . ,xp. The larger the permutation importance of a variable, the more relevant the variable is for the overall prediction accuracy."*

[77] Recall that random forest consists of a number of decision trees. Every node in the trees is a condition on a given variable, and it is designed to optimally split the dataset into two parts so that overall model accuracy can be improved. The measure based on which the (locally) optimal condition is chosen is called impurity (or variance, in the case of the regression trees). This way, one can compute how much each variable reduces the weighted impurity in a tree. For a forest, the impurity reduction from each variable can be averaged and a ranking of variables can be constructed according to this importance measure.

[78] This is not an issue in respect to model forecasting, but regarding model interpretation, it can lead to the

propose the "corrected impurity" importance measure, which is unbiased in terms of the number of categories and category frequencies and is computationally efficient (i.e., almost as fast as the standard impurity importance and much faster than the permutation importance).

In the case of XGBoost, similarly to the impurity approach of random forest, the importance of features is based on gains, that represent fractional contribution of each feature to the model, based on the total gain of the tree node splits of this feature. Higher percentage means a more important predictive feature.

Hypothesis test

Besides building a ranking of importance, it is also crucial to statistically check whether a given predictor is important (or not) in respect to the depend variable of the RF model. According to Janitza et al. (2018), the variable importance depends on many different factors, including aspects related to the data (e.g., correlations, signal-to-noise ratio or the total number of variables) as well as on the random forest specific factors (such as the choice of the number of randomly drawn candidate predictor variables for each split node). Therefore, there is no universally applicable threshold that can be used to statistically discriminate between important and non-important variables. Nonetheless, several hypothesis-testing approaches have been developed. The permutation-based tests entail the repeated computation of random forests. While for low-dimensional settings those approaches might be computationally tractable, for high-dimensional models (e.g., including thousands of predictors), computing time might become enormous. In this sense, Janitza et al. (2018) proposes a variable importance test that is appropriate for high-dimensional data where many variables do not carry any information related to the dependent variable. According to the authors, the testing approach, based on cross-validation procedures, shows at least comparable power and a substantially smaller computation time.

incorrect conclusion that one of the variables is a strong predictor while the others (correlated variables) are not important, while, in reality, they are all close in respect to their statistical relationship with the dependent variable. This effect can be attenuated by using random variable selection at each node (instead of using all possible variables) when growing a tree within the random forest setup.

Appendix E. Database

Table E1 - List of macroeconomic and financial variables

	Category	Name	Source	Original unit	Nickname	tcode
1	Inflation	IPCA (consumer price index)	IBGE	%p.m.	IPCA_headline	1
2	Inflation	IPCA (consumer price index, market prices)	IBGE	%p.m.	IPCA_market	1
3	Inflation	IPCA (consumer price index, administered prices)	IBGE	%p.m.	IPCA_administered	1
4	Inflation	IPCA (consumer price index, tradables)	BCB	%p.m.	IPCA_tradables	1
5	Inflation	IPCA (consumer price index, nontradables)	BCB	%p.m.	IPCA_nontradables	1
6	Inflation	IPCA (consumer price index, services)	BCB	%p.m.	IPCA_services	1
7	Inflation	IPCA (consumer price index, industrial goods)	BCB	%p.m.	IPCA_ind_goods	1
8	Inflation	IPCA (consumer price index, food at home)	BCB	%p.m.	IPCA_food_at_home	1
9	Inflation	IPCA diffusion index	BCB	%	IPCA_diffusion	1
10	Inflation	IPCA-15 (consumer price index-extended 15)	IBGE	%p.m.	IPCA15	1
11	Inflation	IPC-Fipe (consumer price index)	Fipe	%p.m.	IPC-Fipe	1
12	Inflation	IPC-Br (consumer price index)	FGV	%p.m.	IPC-BR	1
13	Inflation	IGP-DI (general price index)	FGV	%p.m.	IGP-DI	1
14	Inflation	IGP-M (general price index)	FGV	%p.m.	IGP-M	1
15	Inflation	IGP-10 (general price index)	FGV	%p.m.	IGP-10	1
16	Inflation	INCC (national index of building costs)	FGV	%p.m.	INCC	1
17	Inflation	Core IPC-Br (core inflation)	FGV	%p.m.	core_IPC-BR	1
18	Inflation	Core IPCA - Exclusion EX0 (core inflation)	BCB	%p.m.	core_IPCA_EX0	1
19	Inflation	Core IPCA - Exclusion EX1(core inflation)	BCB	%p.m.	core_IPCA_EX1	1
20	Inflation	Core IPCA - Double Weight (core inflation)	BCB	%p.m.	core_IPCA_DW	1
21	Inflation	Core IPCA - Trimmed Means Smoothed (core inflation)	BCB	%p.m.	core_IPCA_TM	1
22	Interest rates	Nominal policy interest rate (Selic)	BCB	%p.a.	interest_rate_Selic	2
23	Interest rates	Nominal policy interest rate (long-term interest rate, TJLP)	BCB	%p.a.	interest_rate_TJLP	2
24	Interest rates	Nominal market interest rate (prefixed, 1year)	Anbima	%p.a.	interest_rate_1y	2
25	Interest rates	Nominal market interest rate (prefixed, 2 years)	Anbima	%p.a.	interest_rate_2y	2
26	Interest rates	Nominal market interest rate (prefixed, 5 years)	Anbima	%p.a.	interest_rate_5y	2
27	Interest rates	Real market interest rate (indexed IPCA, 1year)	Anbima	%p.a.	real_interest_1y	2
28	Interest rates	Real market interest rate (indexed IPCA, 2 years)	Anbima	%p.a.	real_interest_2y	2
29	Interest rates	Real market interest rate (indexed IPCA, 5 years)	Anbima	%p.a.	real_interest_5y	2
30	Interest rates	U.S. Treasury 3 months nominal yield	Reuters	%p.a.	US_treasury_3m	2
31	Interest rates	U.S. Treasury 2 years nominal yield	Reuters	%p.a.	US_treasury_2y	2
32	Interest rates	U.S. Treasury 10 years nominal yield	Reuters	%p.a.	US_treasury_10y	2
33	Interest rates	U.S. Treasury 5 years TIPS (Treasury Inflation-Protected Securities)	Reuters	%p.a.	US_treasury_5y_tips	2
34	Money	Monetary base	BCB	R$ thousand	monetary_base	5
35	Money	Money supply (currency outside banks)	BCB	R$ thousand	money_supply	5
36	Money	Demand deposits	BCB	R$ thousand	demand_deposits	5
37	Money	Savings deposits	BCB	R$ thousand	savings_deposits	5
38	Money	M1	BCB	R$ thousand	M1	5
39	Money	M2	BCB	R$ thousand	M2	6
40	Money	M3	BCB	R$ thousand	M3	6
41	Money	M4	BCB	R$ thousand	M4	6
42	Banking	Credit spread (nonearmarked credit rate - Selic rate)	BCB, authors	basis points	credit_spread	2
43	Banking	Non-Performing Loans (NPL) of total credit	BCB, authors	%	non_performing_loans	2
44	Banking	Loan-to-Deposit ratio (LTD)	BCB, authors	Units	loan_to_deposit_ratio	3
45	Banking	Reserve requirements ratio (financial inst. reserve requirements / total deposits)	BCB, authors	Units	reserve_requirements	2
46	Banking	Nonearmarked credit operations outstanding	BCB, authors	R$ million	credit_outstanding	6
47	Capital markets	Ibovespa (Brazil)	Reuters	Index	Ibovespa	5
48	Capital markets	Euro Stoxx 50 price index	Reuters	Index	euro_stoxx50	5
49	Capital markets	MSCI emerging countries (EM, US$)	Reuters	Index	MSCI_emerging	5
50	Capital markets	MSCI developed countries (World, US$)	Reuters	Index	MSCI_developed	5
51	FX and risk	FX-rate (nominal exchange rate, R$/US$)	Reuters	Units	exchange_rate	5
52	FX and risk	REER (Real effective exchange rate, IPA-13 currencies)	Reuters	Index	REER	5
53	FX and risk	U.S. dollar index (DXY, geometric average of 6 currencies in respect to US$)	Reuters	Index	dollar_index	5
54	FX and risk	U.S. dollar emerging market index (Federal Reserve, 19 countries)	Reuters	Index	dollar_index_em	5
55	FX and risk	Embi+Br (Emerging Markets Bond Index Plus Brazil, spread)	Reuters	basis points	embi+br	5
56	FX and risk	CDS (Credit Default Swap, Brazil 5 years)	Reuters	basis points	CDS_5y	5
57	FX and risk	U.S. corporate bonds Moodys seasoned BAA	Reuters	%p.a.	US_corp_bonds	2
58	FX and risk	VIX CBOE volatility index (30-day expected volatility of the S&P500)	Reuters	Index	VIX	1

Note: The column "tcode" denotes the following data transformations:
(1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $\ln(x_t)$; (5) $\Delta \ln(x_t)$; (6) $\Delta^2 \ln(x_t)$.

Table E1 - List of macroeconomic and financial variables (cont.)

	Category	Name	Source	Original unit	Nickname	tcode
59	Labor	Unemployment rate (open)	IBGE	%	unemployment_rate	3
60	Labor	Formal employment created - South	MTb	Units	employment_south	2
61	Labor	Formal employment created - Southeast	MTb	Units	employment_southeast	2
62	Labor	Formal employment created - North	MTb	Units	employment_north	2
63	Labor	Formal employment created - Northeast	MTb	Units	employment_northeast	2
64	Labor	Formal employment created - Central-West	MTb	Units	employment_central_west	2
65	Labor	Minimum wage	MTb	R$	minimum_wage	5
66	Labor	Hours worked in production (Rio Grande do Sul)	Fiergs	Index	hours_worked_prod_RS	5
67	Labor	Disposable overall earnings (accumulated in 12 months)	BCB	R$ million	disposible_earnings	6
68	Industry	Industrial production (mineral extraction)	IBGE	Index	ind_prod_mineral_extract	5
69	Industry	Industrial production (manufacturing industry)	IBGE	Index	ind_prod_manufacturing	5
70	Industry	Industrial production (capital goods)	IBGE	Index	ind_prod_capital_goods	5
71	Industry	Industrial production (intermediate goods)	IBGE	Index	ind_prod_interm_goods	5
72	Industry	Industrial production (consumer goods)	IBGE	Index	ind_prod_consumer_goods	5
73	Industry	Industrial production (durable goods)	IBGE	Index	ind_prod_durable_goods	5
74	Industry	Industrial production (semidurable and nondurable goods)	IBGE	Index	ind_prod_non-durable	5
75	Industry	Installed capacity utilization (Rio Grande do Sul)	Fiergs	%	capacity_utilization_RS	2
76	Industry	Capacity utilization (manufacturing industry, FGV)	FGV	%	capacity_utilization_industry	2
77	Industry	Steel production	BCB	Index	steel_production	5
78	Industry	Vehicles production (total)	Anfavea	Units	vehicles_production	5
79	Industry	Truck production	Anfavea	Units	truck_production	5
80	Industry	Bus production	Anfavea	Units	bus_production	5
81	Industry	Production of agricultural machinery (total)	Anfavea	Units	agricultural_machinery	5
82	Sales	Sales volume index in the retail sector (total)	IBGE	Index	sales_total	5
83	Sales	Sales volume index in the retail sector (fuel and lubricants)	IBGE	Index	sales_fuel	5
84	Sales	Sales volume index in the retail sector (hyperm., superm., food, bever. and tobacco)	IBGE	Index	sales_hypermarket	5
85	Sales	Sales volume index in the retail sector (textiles, clothing and footwear)	IBGE	Index	sales_textiles	5
86	Sales	Sales volume index in the retail sector (furniture and white goods)	IBGE	Index	sales_furniture	5
87	Sales	Sales volume index in the retail sector (vehicles and motorcycles, spare parts)	IBGE	Index	sales_vehicles1	5
88	Sales	Vehicle sales (total)	Anfavea	Units	sales_vehicles2	5
89	Sales	Domestic vehicle sales	Anfavea	Units	domestic_sales_vehicles	5
90	Energy	Electric energy consumption (commercial)	Eletrobras	GWh	electricity_commercial	5
91	Energy	Electric energy consumption (residential)	Eletrobras	GWh	electricity_residential	5
92	Energy	Electric energy consumption (industrial)	Eletrobras	GWh	electricity_industrial	5
93	Energy	Electric energy consumption (other)	Eletrobras	GWh	electricity_other	5
94	Energy	Electric energy consumption (total)	Eletrobras	GWh	electricity_total	5
95	Climate	El Niño-Southern Oscillation, as measured by the Oceanic Niño Index (ONI)	NOAA	Index	oceanic_nino_index	2
96	Climate	Total monthly precipitation (mm) in Belém	INMET	mm	rain_Belem	2
97	Climate	Total monthly precipitation (mm) in Belo Horizonte	INMET	mm	rain_Belo_Horizonte	2
98	Climate	Total monthly precipitation (mm) in Curitiba	INMET	mm	rain_Curitiba	2
99	Climate	Total monthly precipitation (mm) in Florianópolis	INMET	mm	rain_Florianopolis	2
100	Climate	Total monthly precipitation (mm) in Goiânia	INMET	mm	rain_Goiania	2
101	Climate	Total monthly precipitation (mm) in Manaus	INMET	mm	rain_Manaus	2
102	Climate	Total monthly precipitation (mm) in Palmas	INMET	mm	rain_Palmas	2
103	Climate	Total monthly precipitation (mm) in Porto Alegre	INMET	mm	rain_Porto_Alegre	2
104	Climate	Total monthly precipitation (mm) in Recife	INMET	mm	rain_Recife	2
105	Climate	Total monthly precipitation (mm) in Rio Branco	INMET	mm	rain_Rio_Branco	2
106	Climate	Total monthly precipitation (mm) in Rio de Janeiro	INMET	mm	rain_Rio_de_Janeiro	2
107	Climate	Total monthly precipitation (mm) in Salvador	INMET	mm	rain_Salvador	2
108	Climate	Total monthly precipitation (mm) in São Luís	INMET	mm	rain_Sao_Luis	2
109	Climate	Total monthly precipitation (mm) in São Paulo	INMET	mm	rain_Sao_Paulo	2
110	Climate	Total monthly precipitation (mm) in Vitória	INMET	mm	rain_Vitoria	2

Note: The column "tcode" denotes the following data transformations:
(1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $\ln(x_t)$; (5) $\Delta \ln(x_t)$; (6) $\Delta^2 \ln(x_t)$.

Table E1 - List of macroeconomic and financial variables (cont.)

	Category	Name	Source	Original unit	Nickname	tcode
111	Public sector	Primary result of consolidated public sector (current monthly flows)	BCB	R$ million	primary_result	2
112	Public sector	Primary result of consolidated public sector (flows accum. in 12 months)	BCB	R$ million	primary_result_12m	3
113	Public sector	Primary result of consolidated public sector (flow accum. in 12 months, % GDP)	BCB	%	primary_result_%GDP	2
114	Public sector	Net public debt (total, federal government and central bank, % GDP)	BCB	%	public_debt_total_%GDP	3
115	Public sector	Net public debt (internal, federal government and central bank, % GDP)	BCB	%	public_debt_internal_%GDP	3
116	Public sector	Net public debt (external, federal government and central bank, % GDP)	BCB	%	public_debt_external_%GDP	2
117	Public sector	Net public debt (total, consolidated public sector, balances in reais)	BCB	R$ million	public_debt_total	6
118	Public sector	Net public debt (internal, consolidated public sector, balances in reais)	BCB	R$ million	public_debt_internal	6
119	Public sector	Net public debt (external, consolidated public sector, balances in reais)	BCB	R$ million	public_debt_external	2
120	Economic activity	IBC-BR (central bank economic activity index)	BCB	Index	IBC-BR	5
121	Economic activity	GDP (accumulated in the last 12 months, current prices)	BCB	R$ million	GDP	6
122	Economic activity	Consumer confidence index	Fecomercio	Index	consum_confidence	2
123	Exterior	Import price index	Funcex	Index	import_price	6
124	Exterior	Import quantum index	Funcex	Index	import_quantum	5
125	Exterior	Export price index	Funcex	Index	export_price	6
126	Exterior	Export quantum index	Funcex	Index	export_quantum	5
127	Exterior	Imports (agriculture, forestry and fishing)	MDIC/Secex	US$ FOB	imports_agriculture	5
128	Exterior	Imports (mining and quarrying)	MDIC/Secex	US$ FOB	imports_mining	5
129	Exterior	Imports (manufacturing)	MDIC/Secex	US$ FOB	imports_manufacturing	5
130	Exterior	Imports (other products)	MDIC/Secex	US$ FOB	imports_others	5
131	Exterior	Imports (total)	MDIC/Secex	US$ FOB	imports_total	5
132	Exterior	Exports (agriculture, forestry and fishing)	MDIC/Secex	US$ FOB	exports_agriculture	5
133	Exterior	Exports (mining and quarrying)	MDIC/Secex	US$ FOB	exports_mining	5
134	Exterior	Exports (manufacturing)	MDIC/Secex	US$ FOB	exports_manufacturing	5
135	Exterior	Exports (other products)	MDIC/Secex	US$ FOB	exports_others	5
136	Exterior	Exports (total)	MDIC/Secex	US$ FOB	exports_total	5
137	Exterior	International reserves (total)	BCB	US$ million	international_reserves	6
138	Exterior	Current account (monthly, net)	BCB	US$ million	current_account	2
139	Exterior	Current account (accumulated in 12 months, in relation to GDP)	BCB	%	current_account_%GDP	2
140	Exterior	FDI (Foreign Direct Investment, accumulated in 12 months)	BCB, authors	US$ million	FDI	2
141	Exterior	FPI (Foreign Portfolio Investment, accumulated in 12 months)	BCB, authors	US$ million	FPI	2
142	Commodities	CRB all commodities index	Reuters	Index	CRB	5
143	Commodities	CRB foodstuffs index	Reuters	Index	CRB_food	5
144	Commodities	CRB metals index	Reuters	Index	CRB_metals	5
145	Commodities	Baltic exchange dry index	Reuters	Index	Baltic_dry	5
146	Commodities	Oil price (Brent, Europe)	Reuters	US$/barrel	Oil_price_Brent	5
147	Commodities	Oil price (WTI, Oklahoma-USA)	Reuters	US$/barrel	Oil_price_WTI	5
148	Global uncertainty	Economic Policy Uncertainty index for Australia	EPU	Index	EPU_Australia	2
149	Global uncertainty	Economic Policy Uncertainty index for Brazil	EPU	Index	EPU_Brazil	2
150	Global uncertainty	Economic Policy Uncertainty index for Canada	EPU	Index	EPU_Canada	2
151	Global uncertainty	Economic Policy Uncertainty index for Chile	EPU	Index	EPU_Chile	2
152	Global uncertainty	Economic Policy Uncertainty index for China	EPU	Index	EPU_China	2
153	Global uncertainty	Economic Policy Uncertainty index for Colombia	EPU	Index	EPU_Colombia	2
154	Global uncertainty	Economic Policy Uncertainty index for France	EPU	Index	EPU_France	2
155	Global uncertainty	Economic Policy Uncertainty index for Germany	EPU	Index	EPU_Germany	2
156	Global uncertainty	Economic Policy Uncertainty index for Greece	EPU	Index	EPU_Greece	2
157	Global uncertainty	Economic Policy Uncertainty index for India	EPU	Index	EPU_India	2
158	Global uncertainty	Economic Policy Uncertainty index for Ireland	EPU	Index	EPU_Ireland	2
159	Global uncertainty	Economic Policy Uncertainty index for Italy	EPU	Index	EPU_Italy	2
160	Global uncertainty	Economic Policy Uncertainty index for Japan	EPU	Index	EPU_Japan	2
161	Global uncertainty	Economic Policy Uncertainty index for Korea	EPU	Index	EPU_Korea	2
162	Global uncertainty	Economic Policy Uncertainty index for Netherlands	EPU	Index	EPU_Netherlands	2
163	Global uncertainty	Economic Policy Uncertainty index for Russia	EPU	Index	EPU_Russia	2
164	Global uncertainty	Economic Policy Uncertainty index for Spain	EPU	Index	EPU_Spain	2
165	Global uncertainty	Economic Policy Uncertainty index for Singapore	EPU	Index	EPU_Singapore	2
166	Global uncertainty	Economic Policy Uncertainty index for UK	EPU	Index	EPU_UK	2
167	Global uncertainty	Economic Policy Uncertainty index for USA	EPU	Index	EPU_USA	2

Note: The column "tcode" denotes the following data transformations:
(1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $\ln(x_t)$; (5) $\Delta \ln(x_t)$; (6) $\Delta^2 \ln(x_t)$.

Appendix F. Robustness analysis

Table F1 - Mean Squared Error (MSE), $T_1 = 96$ months, $T_2 = 144$ months

dep. var. = IPCA % p.m.	h = 1	h = 2	h = 3	h = 6	h = 9	h = 12	h = 15	h = 18
(1) RW	0.147	0.194	0.193	0.260	0.173	0.276	0.291	0.197
(2) RW-AO	0.128	0.144	0.153	0.171	0.180	0.209	0.187	0.177
(3) ARMA	0.113	0.130	0.129	0.144	0.139	0.155	0.155	0.152
(4) VAR	0.117	0.138	0.137	0.150	0.152	0.167	0.162	0.164
(5) PC backward	0.110	0.131	0.134	0.159	0.136	0.153	0.162	0.204
(6) PC hybrid	0.092***	0.124	0.136	0.157	0.143	0.163	0.170	0.192
(7) Factor model1	0.091**	0.106***	0.137	0.145	0.150	0.155	0.159	0.176
(8) Factor model2	0.091***	0.117	0.131	0.147	0.144	0.155	0.151*	0.154
(9) Factor model3	0.095*	0.106**	0.134	0.145	0.139	0.165	0.151	0.187
(10) Factor model4	0.103	0.125	0.141	0.145	0.142	0.158	0.150**	0.154
(11) Elastic net	0.096**	0.121	0.144	0.142	0.147	0.156	0.158	0.161
(12) Lasso	0.098**	0.120	0.145	0.142	0.146	0.159	0.159	0.162
(13) Adalasso	0.093***	0.113*	0.141	0.143	0.150	0.169	0.163	0.160
(14) Ridge	0.101	0.122	0.138	0.142	0.161	0.169	0.181	0.187
(15) Random Forest	0.105	0.123	0.142	0.150	0.149	0.155	0.153	0.172
(16) Quant Regr. Forest	0.105	0.122	0.141	0.146	0.148	0.154	0.149	0.167
(17) XGBoost	0.102	0.118	0.148	0.168	0.160	0.159	0.150	0.185
(18) RNN	0.142	0.143	0.169	0.234	0.146	0.133*	0.163	0.186
(19) Disag. ARMA	0.118	0.148	0.155	0.169	0.175	0.199	0.181	0.169
(20) Disag. Adalasso	0.099*	0.127	0.147	0.137	0.156	0.150	0.152	0.153
(21) Disag. RF	0.120	0.139	0.148	0.160	0.150	0.156	0.163	0.181
(22) Hybrid Ada-OLS	0.098**	0.124	0.137	0.144	0.157	0.185	0.136**	0.201
(23) Hybrid Ada-RF	0.105	0.114	0.131	0.159	0.147	0.168	0.144	0.195
(24) Hybrid Ada-XGB	0.113	0.122	0.159	0.228	0.183	0.180	0.161	0.204
(25) Hybrid RF-OLS	0.110	0.115	0.160	0.156	0.211	0.250	0.205	0.202
(26) Hybrid RF-Ada	0.100*	0.112*	0.145	0.136*	0.153	0.169	0.173	0.178
(27) Hybrid RF-XGB	0.096	0.116	0.157	0.159	0.163	0.165	0.146	0.208
(28) BEI	0.082*	0.168	0.143	0.158	0.162	0.161	0.162	0.173
(29) Focus	0.083**	0.125	0.138	0.150	0.157	0.161	0.159	0.167
(30) Comb1 Mean	0.095***	0.114**	0.133	0.145	0.145	0.160	0.152	0.164
(31) Comb1 Median	0.095**	0.113**	0.134	0.145	0.146	0.157	0.151	0.167
(32) Comb1 GR	0.099	0.147	0.175	0.273	0.295	0.265	0.354	0.714
(33) Comb1 CLS	0.094***	0.114**	0.137	0.152	0.149	0.175	0.158	0.170
(34) Comb1 CSR	0.099**	0.125	0.155	0.193	0.188	0.202	0.216	0.250
(35) Comb1 Adalasso	0.091**	0.122	0.155	0.195	0.243	0.189	6.423	0.198
(36) Comb1 RF	0.099*	0.135	0.178	0.228	0.195	0.198	0.222	0.216
(37) Comb2 Mean	0.092***	0.116	0.139	0.173	0.160	0.201	0.156	0.175
(38) Comb2 Median	0.092***	0.116	0.139	0.173	0.156	0.201	0.156	0.177
(39) Comb2 GR	0.096***	0.129	0.163	0.196	0.221	0.177	0.200	0.214
(40) Comb2 CLS	0.093***	0.120	0.136	0.166	0.154	0.167	0.162	0.178
(41) Comb2 CSR	0.096***	0.129	0.163	0.196	0.192	0.177	0.200	0.234
(42) Comb2 Adalasso	0.099**	0.127	0.158	0.190	0.181	0.172	0.204	0.214
(43) Comb2 RF	0.105	0.132	0.179	0.218	0.179	0.179	0.202	0.237
(44) Comb3 Mean	0.075**	0.135	0.133	0.151	0.159	0.160	0.163	0.169
(45) Comb3 Median	0.075**	0.135	0.133	0.151	0.159	0.160	0.163	0.169
(46) Comb3 GR	0.066***	0.135	0.146	0.188	0.182	0.172	0.186	0.191
(47) Comb3 CLS	0.083*	0.152	0.138	0.157	0.161	0.167	0.170	0.169
(48) Comb3 CSR	0.096	0.165	0.150	0.188	0.182	0.172	0.186	0.191
(49) Comb3 Adalasso	0.089	0.163	0.152	0.186	0.178	0.171	0.188	0.188
(50) Comb3 RF	0.075**	0.141	0.161	0.198	0.194	0.187	0.198	0.204
number of observations	*64*	*63*	*62*	*59*	*56*	*53*	*50*	*47*
best model	*46*	*7*	*3*	*26*	*5*	*18*	*22*	*3*
R2 oos (%)	*41*	*18*	*0*	*5*	*2*	*13*	*12*	*0*

dep. var. = IPCA % 12 months	h = 1	h = 2	h = 3	h = 6	h = 9	h = 12	h = 15	h = 18
(1) RW	0.301	0.932	1.708	4.552	7.329	10.303	10.471	9.399
(2) RW-AO	3.806	4.836	5.870	8.845	11.415	12.964	13.108	12.423
(3) ARMA	0.216	0.770	1.447	4.382	6.675	9.485	8.673	7.074
(4) VAR	0.282	0.862	1.564	4.160	6.704	9.845	10.358	9.060
(5) PC backward	0.278	0.873	1.587	4.327	6.438	8.826	10.590	11.585
(6) PC hybrid	0.271	0.830	1.481	4.270	6.941	9.553	11.995	13.619
(7) Factor model1	0.642	1.102	1.698	3.512*	5.599	7.091	6.824	6.582
(8) Factor model2	0.365	0.780	1.316	3.479*	5.746	8.081*	8.035	6.817
(9) Factor model3	0.352	0.865	1.561	4.620	5.538	5.619*	4.353	5.215
(10) Factor model4	0.363	0.791	1.245	2.756	4.680	7.584	7.972	6.951
(11) Elastic net	0.260	0.845	1.641	4.284	7.626	8.500	6.842	5.388
(12) Lasso	0.255	0.825	1.663	4.164	7.764	8.708	7.094	5.367
(13) Adalasso	0.230	0.764	1.536	4.546	8.137	8.959	8.594	7.620
(14) Ridge	0.939	1.389	1.997	4.017	5.784	5.950**	5.660	5.398
(15) Random Forest	0.771	1.449	2.087	4.006	5.816	6.533	5.903	5.257
(16) Quant Regr. Forest	0.767	1.452	2.094	3.967	5.671	6.379	5.657	5.010
(17) XGBoost	0.331	0.960	1.799	4.592	6.804	6.548	5.088	4.230
(18) RNN	3.087	8.169	5.318	26.928	1.916	10.601	12.438	2.992
(19) Disag. ARMA	0.214	0.762	1.476	4.522	7.471	10.794	9.828	7.874
(20) Disag. Adalasso	0.268	0.790	1.616	5.400	8.028	7.460*	7.151	6.533
(21) Disag. RF	0.972	1.743	2.510	4.734	6.123	6.739	7.071	6.814
(22) Hybrid Ada-OLS	0.218	0.774	1.669	5.472	11.954	9.838	7.371	6.019
(23) Hybrid Ada-RF	0.391	1.129	1.974	4.297	5.955	7.007	6.951	6.367
(24) Hybrid Ada-XGB	0.340	1.044	2.111	5.271	6.772	7.111	6.182	5.215
(25) Hybrid RF-OLS	0.193	0.821	1.544	6.175	13.578	15.006	12.185	6.459
(26) Hybrid RF-Ada	0.199	0.861	1.784	5.658	14.539	14.741	11.851	6.218
(27) Hybrid RF-XGB	0.372	1.283	2.469	5.110	7.086	6.846	4.780	5.121
(28) BEI	0.088***	0.464*	0.666**	2.322**	4.641***	6.126*	6.192	6.404
(29) Focus	0.090***	0.373***	0.742**	2.193***	3.706**	4.687*	4.564*	4.269
(30) Comb1 Mean	0.329	0.869	1.568	4.473	6.465	7.452	6.690	5.067
(31) Comb1 Median	0.247	0.808	1.514	4.117	6.177	7.156*	6.437	5.520
(32) Comb1 GR	0.292	1.034	1.809	5.305	16.059	31.148	25.465	28.669
(33) Comb1 CLS	0.348	0.797	1.553	5.392	6.601	8.251	6.979	5.235
(34) Comb1 CSR	0.338	0.848	1.609	6.025	11.365	18.254	20.333	21.751
(35) Comb1 Adalasso	0.301	0.927	1.563	9.059	29.811	36.481	31.575	27.743
(36) Comb1 RF	0.477	1.186	2.172	6.225	9.113	12.270	16.555	13.883
(37) Comb2 Mean	0.210	0.776	1.453	4.297	6.872	8.582	9.470	5.994
(38) Comb2 Median	0.210	0.776	1.453	4.297	6.872	8.582	9.470	7.254
(39) Comb2 GR	0.229	0.834	1.608	5.697	12.588	10.144	24.401	24.850
(40) Comb2 CLS	0.214	0.788	1.513	4.387	7.085	8.805	10.766	8.046
(41) Comb2 CSR	0.219	0.818	1.611	5.719	12.678	10.142	33.575	24.581
(42) Comb2 Adalasso	0.224	0.810	1.626	5.353	12.348	10.611	32.409	23.788
(43) Comb2 RF	0.331	1.106	2.130	5.908	10.036	11.546	18.029	12.989
(44) Comb3 Mean	0.082***	0.395***	0.951**	4.297	6.872	8.582	9.470	8.579
(45) Comb3 Median	0.082***	0.395***	0.951**	4.297	6.872	8.582	9.470	8.579
(46) Comb3 GR	0.088***	0.444**	0.829*	5.697	12.588	10.144	24.401	31.376
(47) Comb3 CLS	0.089***	0.476*	0.727**	4.387	7.085	8.805	10.766	10.846
(48) Comb3 CSR	0.105**	0.471*	0.731**	5.719	12.678	10.142	33.575	53.768
(49) Comb3 Adalasso	0.087***	0.506	0.710**	5.353	12.348	10.611	32.409	51.737
(50) Comb3 RF	0.165	0.645	1.388	5.908	10.036	11.546	18.029	19.453
number of observations	*64*	*63*	*62*	*59*	*56*	*53*	*50*	*47*
best model	*45*	*29*	*28*	*29*	*18*	*29*	*9*	*18*
R2 oos (%)	*62*	*51*	*53*	*49*	*71*	*50*	*49*	*57*

Notes: Yellow cells denote Top10 models (lowest MSEs) in each horizon. ***, **, and * indicate rejection at 1%, 5%, and 10% levels, respectively, using the Diebold and Mariano (1995) test, considering model3 (ARMA) as benchmark. The R2 out-of-sample statistics (R2 oos) refers to the best model in each horizon.

Table F2 - Mean Squared Error (MSE), $T_1 = 72$ months, $T_2 = 144$ months

dep. var. = IPCA % p.m.	h = 1	h = 2	h = 3	h = 6	h = 9	h = 12	h = 15	h = 18
(1) RW	0.147	0.194	0.193	0.260	0.173	0.276	0.291	0.197
(2) RW-AO	0.128	0.144	0.153	0.171	0.180	0.209	0.187	0.177
(3) ARMA	0.113	0.130	0.129	0.144	0.139	0.155	0.155	0.152
(4) VAR	0.117	0.138	0.137	0.150	0.152	0.167	0.162	0.164
(5) PC backward	0.110	0.131	0.134	0.159	0.136	0.153	0.162	0.204
(6) PC hybrid	0.092***	0.124	0.136	0.157	0.143	0.163	0.170	0.192
(7) Factor model1	0.091**	0.106***	0.137	0.145	0.150	0.155	0.159	0.176
(8) Factor model2	0.091***	0.117	0.131	0.147	0.144	0.155	0.151*	0.154
(9) Factor model3	0.095*	0.106**	0.134	0.145	0.139	0.165	0.151	0.187
(10) Factor model4	0.103	0.125	0.141	0.145	0.142	0.158	0.150**	0.154
(11) Elastic net	0.096**	0.121	0.144	0.142	0.147	0.156	0.158	0.161
(12) Lasso	0.098**	0.120	0.145	0.142	0.146	0.159	0.159	0.162
(13) Adalasso	0.093***	0.113*	0.141	0.143	0.150	0.169	0.163	0.160
(14) Ridge	0.101	0.122	0.138	0.142	0.161	0.169	0.181	0.187
(15) Random Forest	0.105	0.123	0.142	0.150	0.149	0.155	0.153	0.172
(16) Quant Regr. Forest	0.105	0.122	0.141	0.146	0.148	0.154	0.149	0.167
(17) XGBoost	0.102	0.118	0.148	0.168	0.160	0.159	0.150	0.185
(18) RNN	0.146	0.138	0.148	0.149	0.134	0.135***	0.136*	0.166
(19) Disag. ARMA	0.118	0.148	0.155	0.169	0.175	0.199	0.181	0.169
(20) Disag. Adalasso	0.099*	0.127	0.147	0.137	0.156	0.150	0.152	0.153
(21) Disag. RF	0.120	0.139	0.148	0.160	0.150	0.156	0.163	0.181
(22) Hybrid Ada-OLS	0.098**	0.124	0.137	0.144	0.157	0.185	0.136**	0.201
(23) Hybrid Ada-RF	0.105	0.114	0.131	0.159	0.147	0.168	0.144	0.195
(24) Hybrid Ada-XGB	0.113	0.122	0.159	0.228	0.183	0.180	0.161	0.204
(25) Hybrid RF-OLS	0.110	0.115	0.160	0.156	0.211	0.250	0.205	0.202
(26) Hybrid RF-Ada	0.100*	0.112*	0.145	0.136*	0.153	0.169	0.173	0.178
(27) Hybrid RF-XGB	0.096	0.116	0.157	0.159	0.163	0.165	0.146	0.208
(28) BEI	0.082*	0.168	0.143	0.158	0.162	0.161	0.162	0.173
(29) Focus	0.083**	0.125	0.138	0.150	0.157	0.161	0.159	0.167
(30) Comb1 Mean	0.095**	0.114**	0.133	0.144	0.144	0.160	0.152	0.164
(31) Comb1 Median	0.095**	0.113**	0.134	0.144	0.146	0.156	0.150	0.166
(32) Comb1 GR	0.094*	0.138	0.154	0.251	0.225	0.326	0.374	0.429
(33) Comb1 CLS	0.094***	0.114**	0.135	0.143	0.144	0.160	0.155	0.166
(34) Comb1 CSR	0.098**	0.123	0.149	0.182	0.177	0.196	0.205	0.243
(35) Comb1 Adalasso	0.089***	0.123	0.143	0.196	0.171	0.217	0.241	0.323
(36) Comb1 RF	0.100*	0.140	0.170	0.219	0.187	0.196	0.233	0.237
(37) Comb2 Mean	0.092***	0.120	0.142	0.159	0.159	0.172	0.155	0.175
(38) Comb2 Median	0.092***	0.120	0.142	0.159	0.155	0.165	0.155	0.179
(39) Comb2 GR	0.097***	0.131	0.163	0.199	0.180	0.208	0.191	0.341
(40) Comb2 CLS	0.093***	0.122	0.140	0.157	0.162	0.178	0.154	0.187
(41) Comb2 CSR	0.096***	0.131	0.163	0.199	0.178	0.186	0.191	0.263
(42) Comb2 Adalasso	0.096***	0.132	0.161	0.196	0.178	0.184	0.187	0.359
(43) Comb2 RF	0.102**	0.129	0.171	0.201	0.180	0.180	0.208	0.258
(44) Comb3 Mean	0.075**	0.137	0.137	0.151	0.159	0.160	0.163	0.169
(45) Comb3 Median	0.075**	0.137	0.137	0.151	0.159	0.160	0.163	0.169
(46) Comb3 GR	0.066***	0.143	0.147	0.171	0.178	0.198	0.185	0.187
(47) Comb3 CLS	0.083*	0.154	0.145	0.154	0.160	0.161	0.169	0.169
(48) Comb3 CSR	0.096	0.146	0.148	0.171	0.178	0.198	0.185	0.187
(49) Comb3 Adalasso	0.089	0.144	0.148	0.166	0.177	0.197	0.188	0.185
(50) Comb3 RF	0.075**	0.144	0.147	0.184	0.189	0.178	0.188	0.186
number of observations	*64*	*63*	*62*	*59*	*56*	*53*	*50*	*47*
best model	*46*	*7*	*3*	*26*	*18*	*18*	*18*	*3*
R2 oos (%)	*41*	*18*	*0*	*5*	*3*	*12*	*12*	*0*

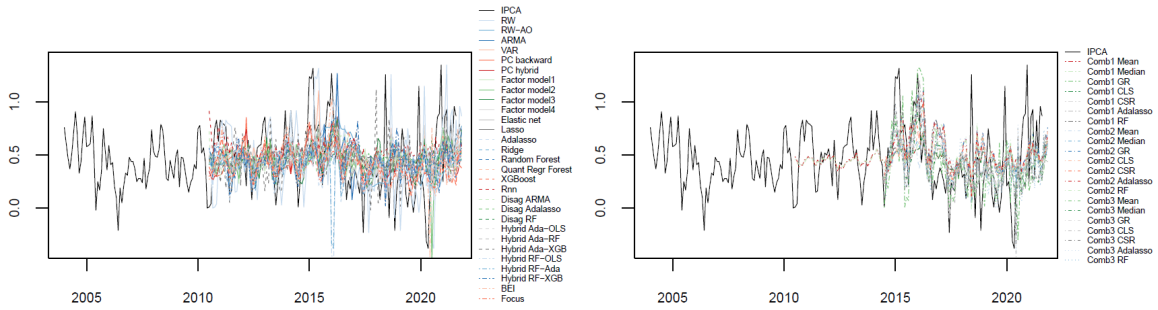
dep. var. = IPCA % 12 months	h = 1	h = 2	h = 3	h = 6	h = 9	h = 12	h = 15	h = 18
(1) RW	0.301	0.932	1.708	4.552	7.329	10.303	10.471	9.399
(2) RW-AO	3.806	4.836	5.870	8.845	11.415	12.964	13.108	12.423
(3) ARMA	0.216	0.770	1.447	4.382	6.675	9.485	8.673	7.074
(4) VAR	0.282	0.862	1.564	4.160	6.704	9.845	10.358	9.060
(5) PC backward	0.278	0.873	1.587	4.327	6.438	8.826	10.590	11.585
(6) PC hybrid	0.271	0.830	1.481	4.270	6.941	9.553	11.995	13.619
(7) Factor model1	0.642	1.102	1.698	3.512*	5.599	7.091	6.824	6.582
(8) Factor model2	0.365	0.780	1.316	3.479*	5.746	8.081*	8.035	6.817
(9) Factor model3	0.352	0.865	1.561	4.620	5.538	5.619*	4.353	5.215
(10) Factor model4	0.363	0.791	1.245	2.756	4.680	7.584	7.972	6.951
(11) Elastic net	0.260	0.845	1.641	4.284	7.626	8.500	6.842	5.388
(12) Lasso	0.255	0.825	1.663	4.164	7.764	8.708	7.094	5.367
(13) Adalasso	0.230	0.764	1.536	4.546	8.137	8.959	8.594	7.620
(14) Ridge	0.939	1.389	1.997	4.017	5.784	5.950**	5.660	5.398
(15) Random Forest	0.771	1.449	2.087	4.006	5.816	6.533	5.903	5.257
(16) Quant Regr. Forest	0.767	1.452	2.094	3.967	5.671	6.379	5.657	5.010
(17) XGBoost	0.331	0.960	1.799	4.592	6.804	6.548	5.088	4.230
(18) RNN	2.420	2.349	1.940	4.051	2.635	3.716	6.300	3.050
(19) Disag. ARMA	0.214	0.762	1.476	4.522	7.471	10.794	9.828	7.874
(20) Disag. Adalasso	0.268	0.790	1.616	5.400	8.028	7.460*	7.151	6.533
(21) Disag. RF	0.972	1.743	2.510	4.734	6.123	6.739	7.071	6.814
(22) Hybrid Ada-OLS	0.218	0.774	1.669	5.472	11.954	9.838	7.371	6.019
(23) Hybrid Ada-RF	0.391	1.129	1.974	4.297	5.955	7.007	6.951	6.367
(24) Hybrid Ada-XGB	0.340	1.044	2.111	5.271	6.772	7.111	6.182	5.215
(25) Hybrid RF-OLS	0.193	0.821	1.544	6.175	13.578	15.006	12.185	6.459
(26) Hybrid RF-Ada	0.199	0.861	1.784	5.658	14.539	14.741	11.851	6.218
(27) Hybrid RF-XGB	0.372	1.283	2.469	5.110	7.086	6.846	4.780	5.121
(28) BEI	0.088***	0.464*	0.666**	2.322**	4.641***	6.126*	6.192	6.404
(29) Focus	0.090***	0.373***	0.742**	2.193***	3.706**	4.687*	4.564*	4.269
(30) Comb1 Mean	0.311	0.864	1.546	4.095	6.488	7.324*	6.541	5.020
(31) Comb1 Median	0.247	0.807	1.497	4.078	6.192	7.060*	6.432	5.524
(32) Comb1 GR	0.265	0.919	2.022	5.911	9.757	25.777	24.184	15.396
(33) Comb1 CLS	0.318	0.882	1.650	4.952	6.428	7.748*	6.671	5.103
(34) Comb1 CSR	0.341	0.851	1.644	5.626	10.313	14.179	16.628	17.264
(35) Comb1 Adalasso	0.274	0.792	1.472	7.815	13.681	31.862	35.690	20.999
(36) Comb1 RF	0.436	1.144	2.216	5.985	9.291	13.587	16.528	13.533
(37) Comb2 Mean	0.219	0.753	1.435	4.501	6.872	8.556	10.809	10.900
(38) Comb2 Median	0.219	0.753	1.435	4.501	6.872	8.556	10.809	10.900
(39) Comb2 GR	0.230	0.801	1.530	4.896	8.957	12.357	23.214	13.804
(40) Comb2 CLS	0.216	0.779	1.518	4.723	6.982	9.342	11.881	10.148
(41) Comb2 CSR	0.220	0.784	1.522	4.898	8.957	12.357	24.944	13.804
(42) Comb2 Adalasso	0.217	0.795	1.585	5.026	7.780	12.339	25.242	13.455
(43) Comb2 RF	0.313	0.950	1.803	7.025	7.752	13.480	15.294	15.459
(44) Comb3 Mean	0.082***	0.395***	0.698**	3.795	6.199	8.556	8.378	10.900
(45) Comb3 Median	0.082***	0.395***	0.698**	4.024	6.188	8.556	8.378	10.900
(46) Comb3 GR	0.086***	0.436**	0.798**	4.161	11.209	12.357	16.751	13.804
(47) Comb3 CLS	0.089***	0.476*	0.666**	3.055**	5.808	9.342	7.330	10.148
(48) Comb3 CSR	0.105**	0.487*	0.707**	3.991	7.982	12.357	16.751	13.804
(49) Comb3 Adalasso	0.086***	0.502	0.718**	4.063	11.143	12.339	13.966	13.455
(50) Comb3 RF	0.150	0.635	1.194	5.202	8.642	13.480	15.786	15.459
number of observations	*64*	*63*	*62*	*59*	*56*	*53*	*50*	*47*
best model	*45*	*29*	*28*	*29*	*18*	*18*	*9*	*18*
R2 oos (%)	*62*	*51*	*53*	*49*	*60*	*60*	*49*	*56*

Notes: Yellow cells denote Top10 models (lowest MSEs) in each horizon. ***, **, and * indicate rejection at 1%, 5%, and 10% levels, respectively, using the Diebold and Mariano (1995) test, considering model3 (ARMA) as benchmark. The R2 out-of-sample statistics (R2 oos) refers to the best model in each horizon.

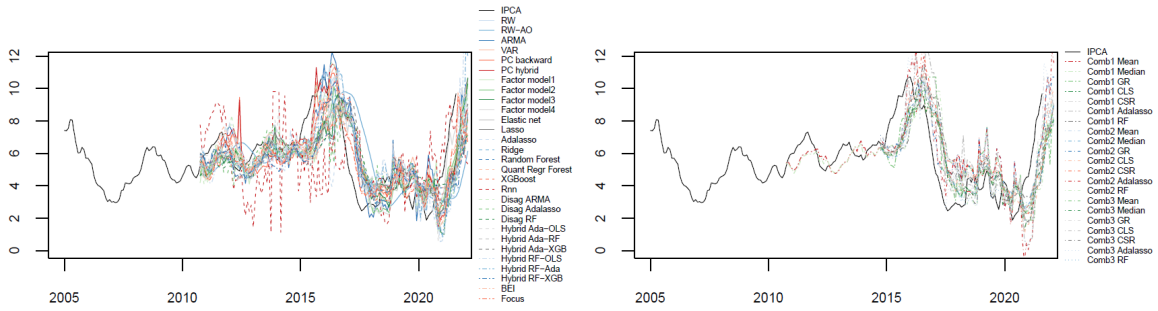
Appendix G. Additional results

Figure G1 - Inflation and forecasts

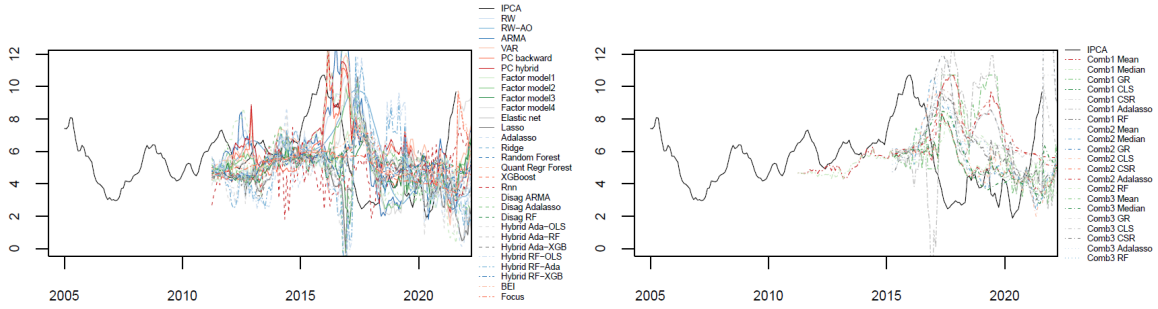
$h = 3$, IPCA % p.m.



$h = 6$, IPCA % 12 months



$h = 12$, IPCA % 12 months



$h = 18$, IPCA % 12 months

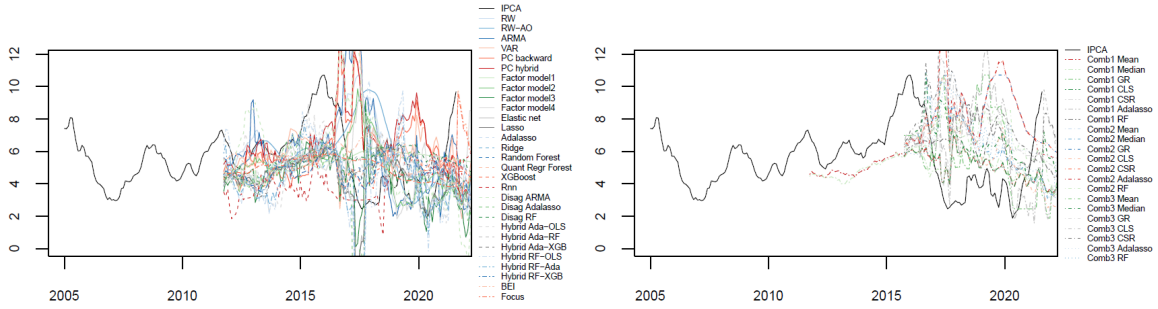
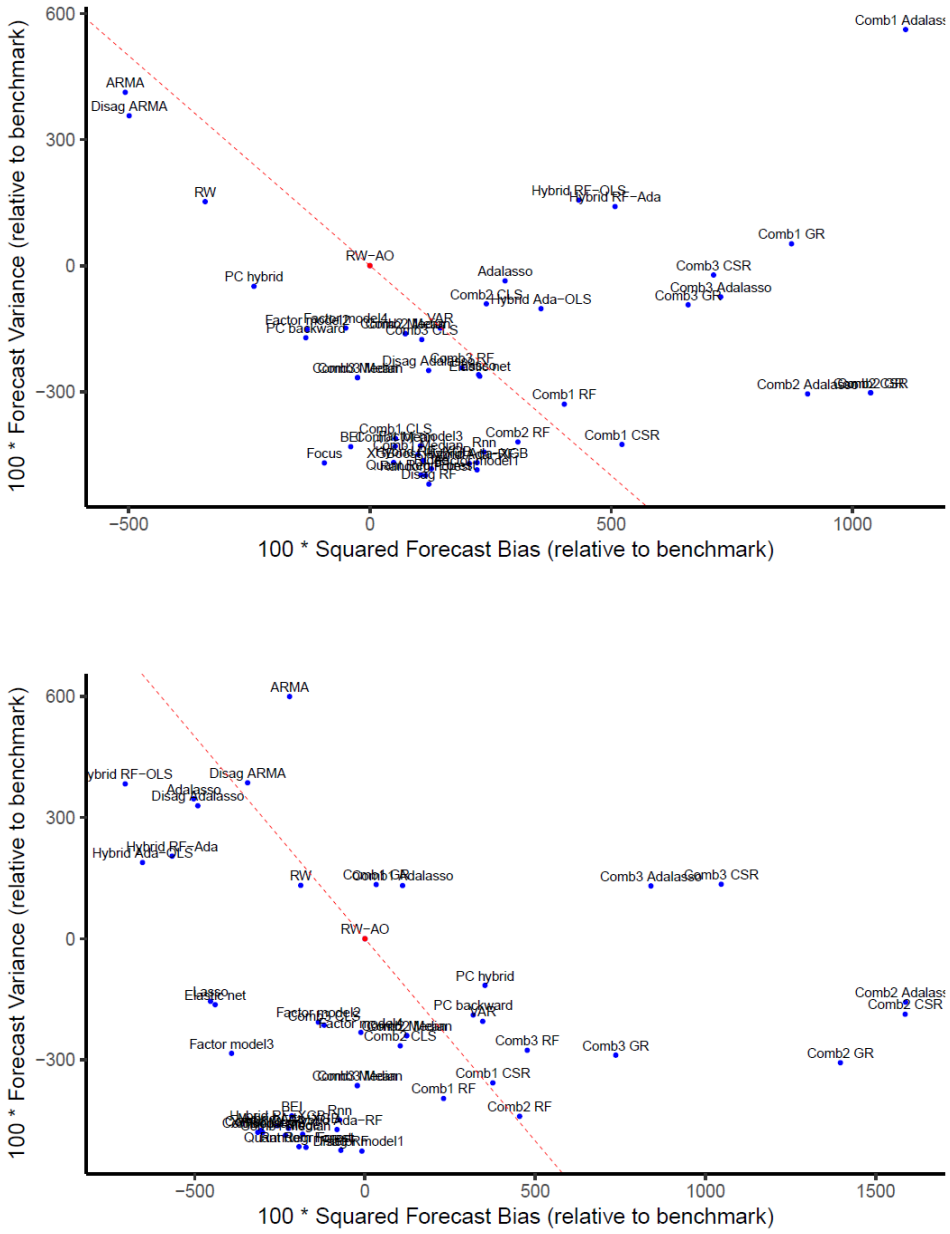
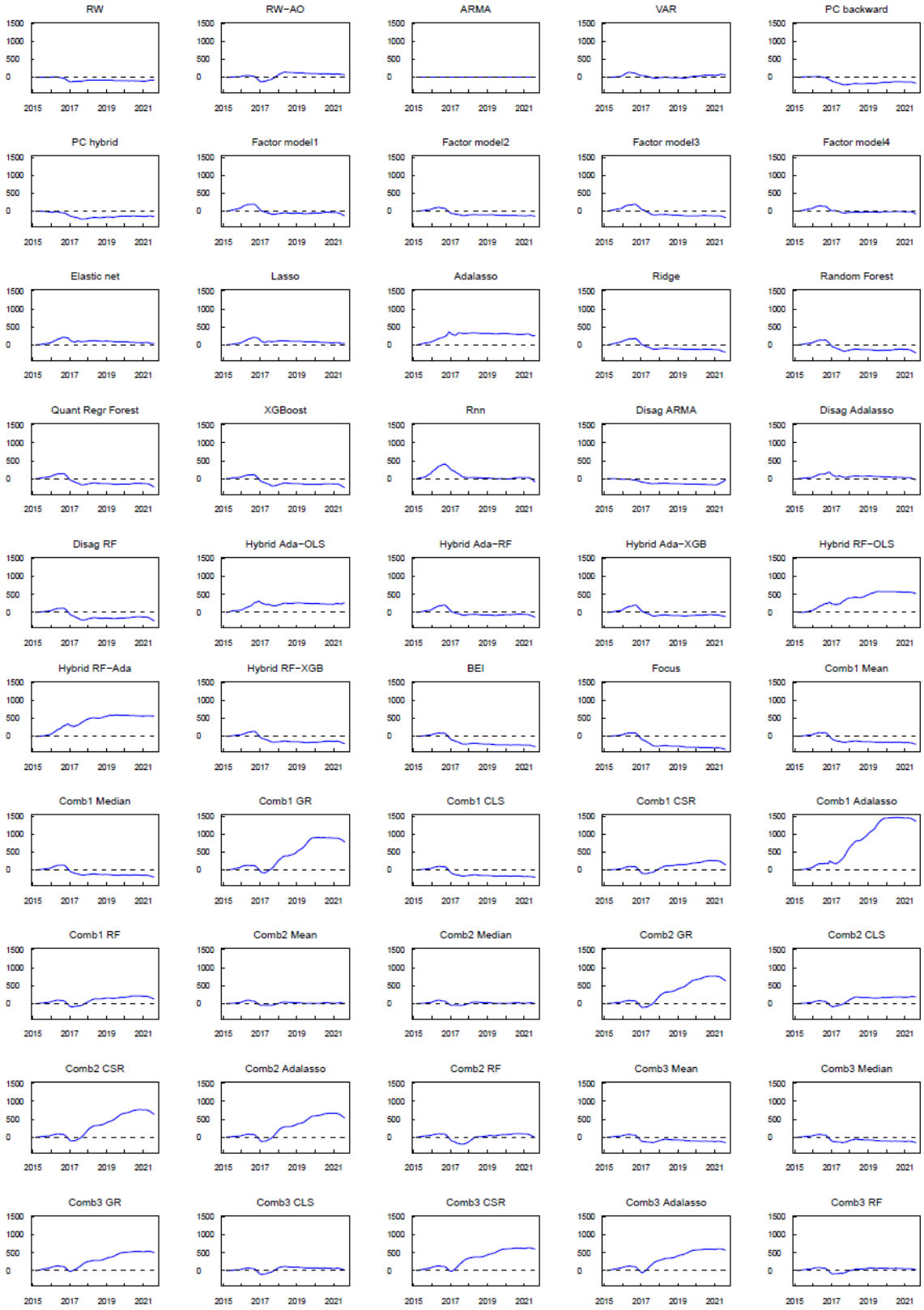


Figure G2 - Scatterplot of relative forecast variance and squared forecast bias for the IPCA (% 12 months), $h = 12$ (top) and $h = 18$ (bottom)



Notes: The y-axis and x-axis represent relative forecast variance and squared forecast bias, computed as the difference between the forecast variance (squared bias) of the considered approach and the forecast variance (squared bias) of the RW-AO. Each point on the red dotted line represents a forecast with the same MSE as the RW-AO; points to the right are forecasts outperformed by the RW-AO, and points to the left represent forecasts that outperform the RW-AO.

Figure G3 - Cumulative Square Prediction Error (CSPE) for $h = 12$ (IPCA % 12 months)

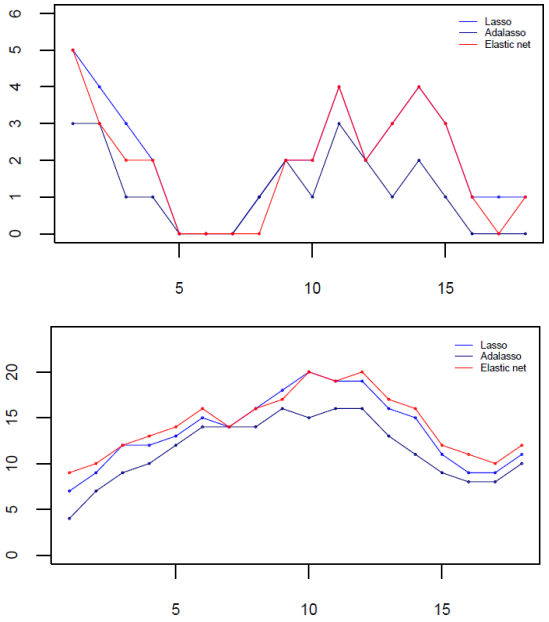


Notes: A positively sloped curve in each panel indicates that the conditional model is outperformed by the benchmark, while the opposite holds for a downward sloping curve. Moreover, if the curve is positive (negative) at the end of the period, then the competing method has a higher (lower) MSE than the benchmark over the evaluation period.

Figure G4 - Cumulative Square Prediction Error (CSPE) for $h = 18$ (IPCA % 12 months)

Notes: A positively sloped curve in each panel indicates that the conditional model is outperformed by the benchmark, while the opposite holds for a downward sloping curve. Moreover, if the curve is positive (negative) at the end of the period, then the competing method has a higher (lower) MSE than the benchmark over the evaluation period.

Figure G5 - Average number of variables selected by lasso, adalasso and elastic net



Notes: Vertical axis is the number of selected variables and the horizontal axis is the forecast horizon. The top graph shows the results for the IPCA (% per month), whereas the bottom graph shows the results for the IPCA (% 12 months).

Figure G6 - Word cloud (*importance*), selected models, IPCA % 12 months

$h = 12$, xgboost (left), random forest (right)

$h = 18$, xgboost (left), random forest (right)

Figure G7 - Fan chart (IPCA % 12 months)