Boosting and Predictability of Macroeconomic Variables: Evidence from Brazil

Guilherme Schultz Lindenmeyer[1] · Hudson da Silva Torrent[2]

Abstract

This paper aims to elaborate a treated data set and apply the boosting methodology to monthly Brazilian macroeconomic variables to check its predictability. The forecasting performed here consists in using linear and nonlinear base-learners, as well as a third type of model that has both linear and nonlinear components in the estimation of the variables using the history itself with lag up to 12 periods. We want to investigate which models and for which forecast horizons we have the strongest performance. The results obtained here through different evaluation approaches point out that, on average, the performance of boosting models using P-Splines as base-learner are the ones that have the best results, especially the methodology with two components: two-stage boosting. In addition, we conducted an analysis on a subgroup of variables with data available until 2022 to verify the validity of our conclusions. We also compared the performance of boosted trees with other models and evaluated model parameters using both cross-validation and Akaike Information Criteria in order to check the robustness of the results.

1 Introduction

Within the universe of machine learning, this work proposes further exploration and application with the component-wise boosting method. This is accomplished for L_2Boosting, the targeted model, by incorporating linear learners in its parameters. It is a machine learning model that is increasingly gaining space in the economic literature. The work of Buchen and Wohlrabe (2011) is one of

✉ Guilherme Schultz Lindenmeyer
    gslindenmeyer@gmail.com

1   Department of Economics, University of Mannheim, Mannheim, Germany

2   Departamento de Estatística, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil

the first papers to consider the boosting methodology as an alternative model to be used in econometric work, in this case, to forecast US industrial production growth. Later, the performance of the method in a high-dimensional macroeconomic context was verified by Wohlrabe and Buchen (2014). Also, more empirical work was done with the model for the case of Germany (Robinzonov et al., 2012; Lehmann & Wohlrabe, 2016), GDP of Japan (Yoon, 2021) and the case of economic variables from the United States (Medeiros et al., 2019; Zeng, 2017; Kauppi & Virtanen, 2021). Finally, the authors Lindenmeyer et al. (2021) were one of the first to apply the methodology to the Brazilian case, verifying the model's performance in forecasting monthly electricity consumption in a Brazilian state. We aim to improve the application of the method in the Brazilian scenario.

The boosting methodology is a machine learning technique that seeks to improve the performance of a given model by combining several base-learners to form a strong one. In the case of L_2Boosting, the method is targeted towards minimizing the mean squared error by iteratively adding new base-learners to the model. The algorithm works by initially fitting a simple model, such as linear regression, to each of the regressors. Then at each iteration it adds the best one, which is the one that minimizes the sum of squared residuals of a simple regression. The next step is to get the residuals of this updated model, which in turn are used to fit a new model. The process is repeated until the maximum number of desired iterations is achieved. Overall, the boosting methodology, particularly L_2Boosting, has proven to be a powerful tool for variable selection in the field of machine learning. The boosting methodology recently has become popular in the field of econometrics due to its ability to improve the accuracy of a model by combining multiple base-learners without losing interpretability. Specifically, in the context of finance, where macroeconomic and financial data can exhibit nonlinear behavior, boosting can offer advantages over traditional linear models such as autoregressive (AR) models, making it a promising technique for predicting complex time series data.

The objective of this paper is the application of the boosting methodology to 140 Brazilian macroeconomic variables typically used for economic forecasting in order to analyze, in fact, how predictable it is within its own history, without taking into account other variables. In this way, it is possible to assign a better prediction model instead of using the same model without having any indication of whether it would be the best. Additionally, we do a comparative study where we compare the forecast obtained from boosting and AR models. This investigation follows the logic coming from the development of boosting methodology when it comes specifically to economic variables and time series. Therefore, we follow the research method from Kauppi and Virtanen (2021) where they specifically study the nonlinearities of macroeconomic time series from the United States. With this paper, our aim is to bring this analysis to the Brazilian case, as well as expand the view to apply the linear and component-wise L_2Boosting algorithm and the version of the algorithm obtained from both splines and boosted trees as gradient descent functions. We wish to investigate both the predictability of Brazilian macroeconomic variables and verify whether they are in general linear or nonlinear and to extend the literature on applications of the algorithm in economics.

We aim to study Brazilian macroeconomic variables and evaluate their predictability using various forecasting models and evaluation tools. To achieve this, we have collected a data set of up-to-date variables that are important for economic research in Brazil, following the selection logic of previous studies. We then estimated our models using a pseudo-out-of-sample forecasting approach, with forecast horizons ranging from 1 to 12 months. We evaluated the performance of all studied models, including the traditional linear model, linear boosting model, and nonlinear boosting model. We also tested an intermediate model that combines linear and nonlinear components, which performed the best on average across our entire data set according to most of the evaluation indicators used. In order to ensure the robustness of our results, we conducted some checks. First, we compared our models with alternative scenarios to assess their performance under different conditions. Additionally, we evaluated two methods for selecting the stopping criterion of the algorithm. Finally, we performed an analysis on a subgroup of variables with data available up to 2022 to further confirm the validity of our conclusions.

The paper is organized as follows: In Sect. 2 we review the literature, in Sect. 3 we present all the methodology used here in this paper; in Sect. 4, we provide details on data collection, as well as its treatment, and explain how we performed the prediction; in Sect. 5 we discuss our results; and finally, in Sect. 6 we conclude this paper with a general discussion of the results obtained.

2 Literature Review

Machine learning methods are gaining popularity in the economic forecasting literature due to their ability to handle high-dimensional data sets with many variables and observations. Boosting models, such as L_2Boosting, can work as variable selection and shrinkage methods while maintaining some level of interpretability and producing accurate predictions. This is in contrast to other machine learning models like neural networks or random forests that prioritize predictive power over interpretability. In a study by Bredahl Kock and Teräsvirta (2016), the authors investigated the use of three neural network estimation methods to forecast macroeconomic time series for G7 countries and Scandinavian countries but found difficulty in adapting them to macroeconomic variables due to their strongly nonlinear nature. The authors used QuickNet to convert nonlinear specification and estimation into a linear variable selection problem. However, the AR model was still a strong competitor to the neural network model.

In a study by Medeiros et al. (2019), many machine learning models were used to forecast US monthly general inflation using an extensive dataset. Their study concludes that machine learning methods can improve inflation forecasting and provide more accurate results than standard benchmarks. Models with variable selection mechanisms shed light on potential nonlinear relationships among key macroeconomic variables, which can have important implications for accurate forecasting, particularly during periods of recession and high uncertainty. From their findings, random forest models were systematically more accurate than any competitor model. It is worth noting that they consider two boosting algorithms: boosted trees

(Friedman, 2001) and factor boosting. The former does not have the variable selection property explicitly, and the latter is a linear component-wise boosting applied to factors from a previously estimated factor model. Therefore, we believe that the potential of boosting can still be explored. In a recent study, Yoon (2021) compared the prediction accuracy of boosting and random forest models against benchmark forecast data for macroeconomic time series published by the Bank of Japan (BOJ) and the International Monetary Fund (IMF) from 2001 to 2018. The authors found that both machine learning models outperformed the BOJ and IMF forecasts in terms of accuracy, with the gradient boosting model being the most accurate. This study suggests that machine learning techniques in forecasting macroeconomic data are promising and encourages further research in this area.

The paper of Kim and Swanson (2018) analyzes the usefulness of big data for forecasting macroeconomic variables using factor models and machine learning methods. The authors evaluate various techniques, including principal component analysis, independent component analysis, and sparse PCA, as well as bagging, boosting, ridge regression, and the elastic net. They carry out a forecasting "horse-race" using prediction models based on a variety of model specification approaches, factor estimation methods, and data windowing methods. They find that factor-based dimension reduction techniques are useful for macroeconomic forecasting, and boosting is an important and successful technique for improving forecasts. Additionally, Chu and Qureshi (2022) explores the use of various forecasting methods, including machine learning and deep learning methods, to forecast U.S. GDP growth using multiple sub-periods. The paper highlights that density-based ML methods such as bagging, boosting, and neural networks can outperform sparsity-based methods for short-horizon forecasts. Also, the authors conclude that ensemble methods, such as boosting, benefit from high dimensional data sets and can outperform popular methods. Our study aims to explore and execute applications of this method in verifying the predictability of Brazilian macroeconomic time series, using different base-learners proposed in the literature. As will be observed in the following sections, boosting is a flexible algorithm that can be combined with different base-learners to perform linear and nonlinear estimations, and as explored in our work, a combination of the two methods. In addition, the algorithm allows for the exploration of its black box, enabling analysis of the selected variables and their importance for estimation[1].

The boosting method was first conceived and introduced by Schapire (1990), Freund (1995) and Freund and Schapire (1996) as an Adaptive Boosting algorithm (AdaBoost), and these papers perform the first fundamental step towards the use of feasible boosting algorithms. Later, in the work of Breiman (1998, 1999), the author, based on the AdaBoost approach, observed that the algorithm can be used as gradient descent in function space, and through this observation, allowed the use of boosting beyond classification contexts. Then Friedman (2001) adapted the existing boosting methodology to apply to regression analysis. In this new methodology,

---

[1] More specifically, within the field of causal econometrics, boosting has been used successfully as an instrumental variable selection algorithm (Bakhitov & Singh, 2022)

the algorithm is used to optimize a squared error loss function, and this creates the methodology we use in this paper, namely L_2Boosting.

For large-scale applications, where the number of variables exceeds the number of observations, classical statistical models such as Ordinary Least Squares (OLS) lose their ability to estimate consistent parameters. Bühlmann and Yu (2003) adapt the existing L_2 Boosting methodology until then so that a preselection of variables can be performed to enable correct estimation of a boosted linear regression model. Later, Bühlmann (2006) mathematically proves the consistency of L_2Boosting in applications with large data sets. After all these scientific contributions, we have the model as pointed out by Schmid and Hothorn (2008),

> [...] when the number of covariates p in a data set is large (and when selecting a small number of relevant covariates is desirable), boosting is usually superior to standard estimation techniques for regression models (such as backward step-wise linear regression, which, e.g., cannot be applied if p is larger than the number of observations n). (pg.2)

In that paper by Schmid and Hothorn, the authors start from Bühlmann and Yu (2003) and study the application and consistency of L_2Boosting via two approaches: with the base-learner used by Bühlmann and Yu, called smoothing spline, and compare it with the base-learner P-Spline approach conceived by Eilers and Marx (2010). Schmid and Hothorn conclude that by using P-Spline as base-learners instead of smoothing spline, "[...] the computational effort of component-wise L_2 Boosting can be greatly reduced, while there is only a minor effect on the predictive performance of the boosting algorithm.". The methodology when dealing with macroeconomic data was improved in Bai and Ng (2009), where the authors perform the first formal adaptation and application of L_2Boosting as a variable selection model with time series. The authors perform tests using regressor variables obtained from Principal Component Analysis (PCA) and also point out: "[...] boosting has the advantage that it does not require *a priori* ordering of the predictors or their lags as conventional model selection procedures do". With this, we arrive at temporally L_2 Boosting with its methodology adapted for large data sets, for time series, and with both linear and nonlinear gradient functions. Finally, these authors developed and proved that the L_2 Boosting model can be used empirically and also they have developed a code library for the R programming and statistical language.

Robinzonov et al. (2012) discuss macroeconomic variables and their nonlinearity, as well as the difficulty of selecting lags for forecasting. They offer as a solution the use of boosting. In the paper, the method is used in two ways, linear base-learner, and base-learner with penalized B-Splines. The empirical exercise performed was to forecast monthly German industrial production from 1992 to 2006 with the boosting versus benchmark (AR) methodology. It is concluded that the boosting method has advantages because of its flexibility, performance, and the way variables and lags are selected. In Lehmann and Wohlrabe (2016), boosting was utilized to also forecast German industrial production over a period ranging from 1996 to 2014, and the authors emphasized the significant benefits of the algorithm, as it allows opening the "black box" of the algorithm. One of the key advantages of boosting is that it provides transparency into the model and allows for variable importance identification

which permits to study of patterns among the variables selected by the algorithm. The study by Wohlrabe and Buchen (2014) highlights how the use of boosting has been gaining ground in the macroeconomic forecasting literature. In their exercise, the authors develop a database to forecast several macroeconomic indicators for the United States, the Eurozone, and Germany. Overall, the conclusion of the paper emphasizes that the model outperforms its benchmark in almost all scenarios and handles macroeconomic forecasting very well. Furthermore, two other conclusions proposed are related to the maximum number of boosting iterations: the proposal made by Hastie (2007) to use the number of different selected variables instead of the trace when considering the use of AIC brings improvements to forecasting, and the K-fold cross-validation criterion, in general, is dominant among the results.

In Lehmann and Wohlrabe (2017), the aim of the paper is to use the component-wise boosting model to perform regional GDP forecasting from a database previously assembled by the authors in another paper (Lehmann & Wohlrabe, 2015) that has data from three German regions: Saxony, Baden-Württemberg and West Germany. The database consists of macroeconomic indicators, price indicators, consumer survey results, international data, and regional data. With this, L_2Boosting is applied, and then its results are verified by comparing it to a benchmark and analyzing the variables selected by the model. Finally, the authors end the research by highlighting the competitiveness of boosting and its applicability. Another macroeconomic application of the methodology took place in Zeng (2017). The work consisted of continuing a line of research that compares forecasting methods for macroeconomic variables and their comparison between country-specific models and models with international predictors, as well as verifying whether forecasts with aggregate data are superior to those with disaggregated data (Marcellino et al., Feb 2003). In Zeng's paper, the author starts with the conclusion that country-specific forecasts provide better results and tests the feasibility of boosting in the face of the forecasting scenario for macroeconomic variables from 1970 to 2011. Empirical results indicate that using disaggregated data with factor analysis or by selection through boosting results in higher forecasting performances, and the research also considers boosting as a competitive model compared to factor analysis when faced with high-dimensional data.

As the research has progressed, a recent line of study within boosting has emerged that focuses on applications of the methodology in the verification of nonlinear predictability in macroeconomic time series, which is the one we will extend here with this present paper: applications of the methodology in the verification of nonlinear predictability in macroeconomic time series. In Kauppi and Virtanen (2021) there is a study of these nonlinearities in macroeconomic variables in the United States. The importance of the subject is highlighted by the authors,

> While it is often argued that nonlinearity is an inherent feature of macroeconomic time series, linear forecasts have mostly been found to perform better than forecasts based on various nonlinear models. There are cases where nonlinear models have yielded more accurate forecasts than linear models, but it generally remains unclear to what extent and when nonlinear forecasts are likely to be useful in macroeconomic forecasting. (pg.1)

The work consisted in using different boosting approaches to check the predictability of 128 macroeconomic indicators from 1959 to 2016 in the US. With this, the authors show that for a good portion of the selected database, the method can improve the accuracy of the forecast over a linear forecasting approach. They also identified a category of variables where nonlinear modeling is more likely to produce the best results.

Recent scientific literature has explored various methodologies for forecasting in the Brazilian context. For example, Cepni et al. (2020) examines the importance of economic policy uncertainty and data surprises in forecasting real GDP growth in Brazil, along with four other emerging market economies. The study employs dynamic factor models to construct GDP predictions, using both local and global economic variables to assess the relevance of uncertainty and data surprises. The paper also evaluates benchmark linear autoregressive models and estimates factors using a variety of data shrinkage methods. The paper of Ribeiro and dos Santos Coelho (2020) explores the use of regression ensembles (bagging, boosting and stacking) to forecast short-term agricultural commodities prices in Brazil. The study compares the performance of these approaches to single models, specifically K-Nearest Neighbor, Neural Networks, and Support Vector Regression. The paper conducts a case study for soybean and for wheat, and concludes the usefulness of ensemble approaches, such as boosting, which combine predictions from different models to achieve more accurate results. On the other hand, the application of variable selection such as L_2Boosting for forecasting in Brazil is a more recent development. In view of this, Lindenmeyer et al. (2021) perform the application of the methodology, from a database with 822 regressors from 2002 to 2017, to predict the electricity consumption in the state Rio Grande do Sul at the time of the Brazilian energy crisis. They compare the L_2Boosting algorithm with an autoregressive benchmark and perform the prediction up to 3 horizons ahead. And they conclude by considering the boosting methodology as valid and competitive in the face of short-term forecast scenarios (1 month ahead) since the results were strong compared to the benchmark.

With the development and growing importance of the boosting methodology, the adaptation of this methodology to deal with time series, as well as in the analysis of the predictability of macroeconomic variables, and together with the little attention given to Brazil in the literature, we aimed the development of this paper. That is, we will extend another example of the application of the methodology, as well as study and obtain evidence on the Brazilian case. Also, we want to verify whether the conclusions obtained by the authors Kauppi and Virtanen (2021) applied to the case of the United States are also applicable to the case of an emerging country, in this case, Brazil.

3 Methodology

This section presents the models chosen, as well as explains the methodology behind each of them. Boosting models require the selection of some macro parameters in order to perform the forecast. Here we also explain these parameters and the

reasoning behind them. Later, in Sect. 4, we comment in detail on how our forecasting strategy and the application of the models explained here were done.

3.1 Linear Model

The linear modeling is done in a straightforward manner and its main purpose here is to serve as a benchmark for the other models. As our goal is to check the predictability of each time series from its own history, the linear model is autoregressive. Since our data is monthly, we assume a period of up to 12 lags to be considered in the model, so the autoregressive model AR(p) has $p = 0, 1, ..., 12$. Let y_t be the time series we would like to model, then the linear model AR(p) has the following format for a chosen p:

$$y_t = \beta_0 + \sum_{i=1}^{p} \beta_i y_{t-i} + \varepsilon_t \tag{1}$$

where β_0 is a constant, β_i are the coefficients for each lag and ε_t is the error term. From Eq. 1 we can see that we obtain 13 models. Since we want to keep only one as a benchmark, we select the best AR(p) model from commonly used model selection strategies: Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) (Akaike, 1973; Schwarz, 1978).

3.2 Boosting

In this section, we will comment in a general way on how the Boosting algorithm works, and in the following subsections, we will talk about the specifics of each case. For simplicity, we leave aside the variable h for forecast horizons, as their role in the algorithm will be explained later. The algorithm consists in building a model, be it linear or nonlinear, iteratively, and additively (component-wise). Let y_t be some time series in question, x_t a vector of regressor variables (in our case, $x_t(p)$ are the lags from y_t to p, where $p = 1, ..., 12$) and M the stopping criterion of the algorithm, the $\hat{y}_t$ boosting estimation is the result of a sum of M distinct parts plus a constant, having the following form:

$$\hat{y}_t = \hat{f}(x_t) = \hat{f}^{(0)} + v \sum_{m=1}^{M} \hat{g}^{(m)}(x_t; \hat{\beta}_m), \tag{2}$$

where v is a learning rate parameter (usually $0 < v \leq 1$ according to the literature Friedman (2001)) and $\hat{g}^{(m)}(x_t; \hat{\beta}_m)$ is the learner, where $\hat{\beta}_m$ is the set of coefficients obtained from a fitting procedure and a loss function. For a given m,

$$\hat{g}^{(m)}(x_t; \hat{\beta}_m) = \underset{\hat{h}(\cdot)}{\operatorname{argmin}} L(y_t, \hat{h}(x_t)), \tag{3}$$

where $L(\cdot)$ is the chosen loss function, e.g. mean square error and the function $\hat{h}$ is the fitting procedure which can vary depending on the approach. The stopping

criterion M can be chosen by the researcher or also can be retrieved through AIC, BIC, or cross-validation. For our research, we fixed an upper bound $M = 300$ and used cross-validation to choose the optimal M, $M^* \leq 300$, in each estimation. Following this, we can describe generically the algorithm to forecast once as follows:

Step 1. We start with $m = 0$ and define $\hat{f}^{(0)} = \bar{y}_t$, where $\bar{y}_t$ is the average.

Step 2. For $m = 1$ to M:

1. Compute the residuals, defined as $\varepsilon_t = y_t - \hat{f}^{(m-1)}$.
2. Do regression on the residuals ε_t on each predictor $x_{(p)}$, with $p = 1, 2, ..., 12$, and compute the sum of squared residuals (SSR).
3. Select the predictor $x_{(p^*)}$ which has the smallest SSR.
4. Define $\hat{g}^{(m)} = \hat{\beta}_{(p^*)} x_{(p^*)}$.
5. Lastly, update the estimation $\hat{f}^{(m)} = \hat{f}^{(m-1)} + v\hat{g}^{(m)}$.

As stated in Park et al. (2009), "L_2 boosting is simply repeated least-squares fitting of residuals.". In our study, we are testing the predictability of macroeconomic variables using only their own history. Therefore, in Step 2 where the regression occurs, we regress only to the $p = 12$ lags of the objective variable. However, the model is easily expandable to use external regressors. The papers in Lehmann and Wohlrabe (2017) and Lindenmeyer et al. (2021) show examples of using the boosting methodology using dozens of external regressors and their lags. To understand the magnitude of the prediction performed here in this paper, we applied the algorithm specified above to each of all variables in our data set, multiplied by the number of observations used in the test set, and multiplied by the number of models used.

3.2.1 Linear Boosting

Linear boosting is the use of the algorithm specified in the previous section (see Sect. 3.2) with the fit of linear regression in part 2.2 of the algorithm instructions. Let y_t be the time series to be modeled and x_t be its regressors, the model equation for each estimate can be written in the same way as Eq. 1. The difference between the linear model and linear boosting is that here we are selecting the best p for the estimation for each iteration from $m = 1$ to M.

3.2.2 Boosting with Splines

In order to understand how boosting with splines works, we must first understand the concept of splines. We can define splines as a continuous piecewise curve. Subsequently, the application of splines was extended to be used as a "smoothing spline", that is, an estimation of a function $\hat{f}(x_i)$ given a data set z_i and a number of knots to estimate, which together will form an additive estimate (Green & Silverman, 1993).

The use of smoothing splines in boosting methodology started with the paper by Bühlmann and Yu (2003), where they were used as the learner of the algorithm. However, according to Schmid and Hothorn (2008), for the boosting methodology, "smoothing splines are clearly less efficient [computationally]

than other smooth base-learners." In that paper, the authors investigate the likelihood of using a modified splines method - P-splines, formulated by Eilers and Marx (1996). In this method, there is also the use of a penalty, but a discrete one. The authors state, as follows,

> P-splines have been used successfully in regression as an approximation of smoothing splines. We have shown that this approximation is also successful in a boosting context: By using P-spline base-learners instead of smoothing spline base-learners, the computational effort of component-wise L2Boosting can be greatly reduced, while there is only a minor effect on the predictive performance of the boosting algorithm. (pg.18)

Therefore, this concludes that the smoothing splines method can be replaced by P-splines without loss of quality and with gains in efficiency in the case of boosting. For our scenario, instead of applying a linear regression in step 2.2 of the algorithm, a curve approximation of the data made by the P-spline methodology will be used, which was implemented in the R programming language by the authors of the package "mboost" (Hothorn et al., 2011). Therefore, we make use of the package and we use the R within the RStudio free software development environment (R Core Team:, 2019; Rstudio, 2020).

3.2.3 Two-Stage Boosting

There is discussion as to whether macroeconomic and financial variables are predominantly linear or nonlinear, and, regardless, whether the most reasonable model to fit is linear or nonlinear (Stock & Watson, 1998). Recapping the discussion made in Sect. 2 on time series nonlinearity, the authors Kauppi and Virtanen (2021) adapt the model from Taieb and Hyndman (2014) and propose it as a direct forecast procedure, and hybrid model between nonlinear and linear: two-stage boosting.

Given a time series y_t, the model starts with the convectional estimation of the linear methodology explained in Sect. 3.1 for each h. After that, a regression is estimated again, but this time with nonlinear boosting (with splines) on the residuals of the estimated series and original series. The result is a two-component estimate that retains information from both linear and nonlinear estimations. Its advantage is that it is expected, on average, to have smaller forecast errors, because if the time series in question has predominantly linear behavior, the nonlinear component will be small. The opposite is also expected to be valid, i.e. a series with nonlinear behavior will have a smaller linear component in the total weight of the estimate. We have the following equation:

$$\hat{g}_h^{TSBoost}(x_t) = \hat{g}_h^{Linear}(x_t) + \hat{r}_h^{BSpline}(x_t) \tag{4}$$

where $\hat{g}_h^{TSBoost}$ is the two-stage boosting, $\hat{g}_h^{Linear}$ is the linear estimation and $\hat{r}_h^{BSpline}$ is the estimation on the residuals of the linear fit.

4 Data and Forecasting Approach

In this section, we explain how we acquired our data set with 140 variables and 288 observations ranging from January 1996 to December 2019. Furthermore, here we discuss important decision points regarding the chosen forecasting strategies, as well as define our strategy used. Finally, we also explain the indicators and evaluation methods chosen to analyze the performance of the models.

4.1 Data

We gathered a base of 140 Brazilian macroeconomic time series. All variables were limited to start from January 1996, since from then on we already have a certain relative stability because it is post Real Plan and the population is relatively more used to a stable currency. Most of our variables were still being updated until the end of 2021 or the beginning of 2022 since we aim to use current data that can be used as regressors in other research. But to standardize the database and leave aside the effect of the pandemic, we consider the data until December 2019. This sums up to mostly 288 monthly observations for each variable. This decision was also made in order to have a more complete set since at the time of collection there were time series that have not yet been updated for 2020. We divided the 288 observations into a training set, where the model will be developed, and a test set, where we check the effectiveness of the models. As common in many empirical works, we split 75% train set and 25% test set for almost all of our selected variables.[2]

To analyze the predictability of Brazilian macroeconomic variables is the construction of a credible data set of utmost importance. For this, the base is grounded from another data set, built by the authors Barbosa et al. (2020), where they forecast unemployment, industrial production index, IPCA, and IPC (Brazilian CPI indexes). For this, they gathered 117 variables and used factorial models to predict them. Since our focus is on Brazilian variables, we discarded the variables that are not national. Also, since our objective is to analyze the predictability of national variables of interest, we also follow the logic for selecting the variables from Kauppi and Virtanen (2021) and McCracken and Ng (2016).

In order to conduct our analysis, we collected the historical time series following a package that integrates the Ipeadata API with the R programming language (Gomes, 2022). The result of this data collection made us take variables from Ipeadata with several different sources that will be shown in the Appendix to this work (see Appendix A). In order to have an interesting representation of the variables selected, we present in Table 1 the division by subject of the data collected and in Table 2 the source of each series.

---

[2] There are unemployment variables that had not been updated for the full year 2019 at the time of database development, so the cutoff in those cases between test and training set was 80% and 20% respectively out of 276 observations.

Table 1 Themes of the data set

Theme	Number of series
Balance of payments	4
Capital stock	1
Consumption and sales	16
Currency and credit	2
Employment	8
Exchange	14
Financial	7
Foreign trade	12
National accounts	15
Perception and expectation	2
Prices	35
Production	4
Public finance	15
Salary and income	5
Total	140

Table 2 Sources for the data set

Source of the data	Number of series
Agência Nacional do Petróleo	8
ANBIMA	2
Banco Central do Brasil	28
Confederação Nacional da Indústria	3
Eletrobras	6
Federação das Indústrias do Estado de Minas Gerais	1
Federação das Indústrias do Estado do Rio de Janeiro	3
Federação do Comércio do Estado de São Paulo	3
Federação e Centro das Indústrias do Estado de São Paulo	3
Fundação Centro de Estudos do Comércio Exterior	10
Fundação Getulio Vargas	8
Fundação Instituto de Pesquisas Econômicas	1
Fundação Sistema Estadual de Análise de Dados	5
Instituto Brasileiro de Geografia e Estatística	19
Instituto de Pesquisa Econômica Aplicada	30
Ministério da Economia	8
Ministério do Desenvolvimento, Indústria e Comércio Exterior	2
Total	140

4.1.1 Data Treatment

For the purpose of using the macroeconomic variables obtained in econometric models, we must first treat them. That is, we must perform transformations on each series that leave it with a constant mean and variance over time, i.e. stationarized series.

For this, we apply two tests: the first is the Augmented Dickey-Fuller (ADF) test, which was developed by Dickey and Fuller (1979) and developed in R by Pfaff et al. (2016), and the second is the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test developed by Kwiatkowski et al. (1992), while the coding in R was performed by Trapletti and Hornik (2021). The time series is only considered treated when we interpret the information from both tests as being stationary. For the ADF test, let y_t be the time series in question, and is presented as follows,

$$y_t = \phi y_{t-1} + \sum_{i=1}^{p-1} \phi_i \Delta y_{t-i} + e_t \tag{5}$$

$$\Delta y_t = \underbrace{(\phi - 1)}_{\delta} y_{t-1} + \sum_{i=1}^{p-1} \phi_i \Delta y_{t-i} + e_t \tag{6}$$

where α, ϕ and ϕ_i are coefficients, e_t is the error term, $\delta = \phi - 1$ and p is the number of lags used in the test to bring more robustness. As the test is part of the Unit Root class of tests, the null hypothesis assumes the presence of a unit root ($\alpha = 1$ for Eq. 5 or $\delta = 0$ for Eq. 6) and therefore would be stationary if accepted. The test statistic is given by $\frac{\hat{\delta}}{SE(\hat{\delta})}$ and its critical value is taken from Dickey-Fuller t-distribution (Fuller, 2009). We reject the null hypothesis if the test statistic is smaller than the critical value for 5% significance.

The absence of a unit root does not necessarily mean that the time series in question is stationary. The KPSS test tests the hypothesis that the series is stationary in trend as the null hypothesis or has a unit root as the alternative hypothesis. We decided to bring in this additional test in order to get more robustness to the analysis and treatment of our data. To do so, consider

$$y_t = \beta_t + (r_t + \alpha) + e_t, \tag{7}$$

where y_t is the time series, α is the drift parameter (our case $\alpha = 0$), $r_t = r_{t-1} + u_t$ is a random walk process and both e_t and u_t are independent and identically distributed (i.i.d) error terms with mean equal to zero and constant variance. The KPSS test statistic is given by KPSS $= \frac{\sum_{t=1}^{N} S_t^2}{N^2 \lambda^2}$, where S_t is the sum of squared accumulated errors, λ is the standard deviation and N is the number of observations. The critical values can be found in the original article by the authors Kwiatkowski et al. (1992). Again we are using 5% significance. If after applying both tests, the series is still interpreted as nonstationary by at least one of them, then we apply (1) difference of series, (2) difference of log of series, or (3) direct growth $\left(\frac{y_{t+1} - y_t}{y_t} \right)$. If we applied the

difference once and then the series is stationary, we consider the series as integration order $I(1)$, if only for the second difference, we consider it as $I(2)$. For none of the series it was necessary to run more than two differences to make it stationary.

4.2 Forecasting Approach

For single-step forecasting for time series, we usually use all the available data up to period t and perform direct estimation for $t + 1$. But when we are dealing with multi-step forecasting, i.e. forecasting 2 or more steps ahead, there is a discussion about which method to use: direct forecasting or recursive forecasting. Direct prediction means training a model that always estimates h steps ahead. Therefore, for every h that has to be done in the research, a new model has to be estimated, which can be computationally heavy, and we might also fall into the risk of not using all the available observations at the time of prediction. Recursive forecasting, on the other hand, can be obtained from h different forecasts from one-step ahead models. The advantage of this method is the use of all available information at the time of prediction, but the downside is that we make predictions on top of the predictions, which can lead to carrying errors from $h = 1$ to larger h. In this work, as in the work of Kauppi and Virtanen (2021), we decided to do it by direct prediction. The justification is due to the goal of making forecasts with $h = 1, ..., 12$, and avoiding increasing errors by estimation with low performance in one-step ahead models. Moreover, the forecasting literature on the two strategy differences slightly favors direct estimation (Ji et al., 2005; Marcellino et al., 2006; Hamzaçebi et al., 2009). According to the forecast of simulated time series using the boosting methodology carried out by Kauppi and Virtanen (2021), "In the simulations, we find no significant difference between the direct and indirect procedures, while the direct method is on average more accurate than the indirect approach in terms of empirical comparisons".

All series in our data set are stationary (see Sect. 4.1), and usually, our series are in logarithmic value. Also, the order of integration of the series can be $I(0)$, $I(1)$, and $I(2)$. The estimation here is done as follows, consider y_t to be a series we want to model using the vector of regressors $x_t(p)$, which can be defined as $x_t(p) = y_t, ..., y_{t-p}$, $x_t(p) = \Delta y_t, ..., \Delta y_{t-p}$ or $x_t(p) = \Delta^2 y_t, ..., \Delta^2 y_{t-p}$, for $I(0)$, $I(1)$ and $I(2)$, respectively. In all our prediction exercise we consider $p = 12$. When making the prediction for the observation z_t, we train a model that predicts h steps ahead and that depends on the order of integration of the series. Given h, we can define that variable to be predicted, z_{t+h}, as follows:

$$
\begin{aligned}
I(0) &: z_{t+h} = y_{t+h} \\
I(1) &: z_{t+h} = y_{t+h} - y_t \\
I(2) &: z_{t+h} = y_{t+h} - y_t - h\Delta y_t
\end{aligned}
\tag{8}
$$

Hence, in this way, the estimation of z_{t+h} will be equal to

$$
z_{t+h} = f_h(x_t(p)) + \varepsilon_{t+h}
\tag{9}
$$

Where $f_h(x_t(p))$ is the estimation given h and using $x_t(p)$ as regressors (e.g. boosting). Also, ε_{t+h} is the prediction error. Thus, using all the information until t and the direct forecast approach for a given h, the boosting method shown in Eq. 2 to perform the prediction exercise on a series y_t of, for example, $I(0)$, is:

$$y_{t+h} = f_h(x_t(p)) + \varepsilon_{t+h} = f_h^{(0)} + v \sum_{m=1}^{M} g^{(m)}(x_t(p); \beta_m) + \varepsilon_{t+h}, \qquad (10)$$

$$f_h^{(0)} = \frac{\sum_{i=1}^{t} y_i}{N} = \bar{y}_t. \qquad (11)$$

For the parameters, we set $v = 0.1$ as is generally suggested in several papers (Friedman, 2001; Bühlmann & Hothorn, 2007). Also, our chosen loss function is the minimization of the traditional Mean Squared Error (MSE), defined as the squared difference of the forecast errors. Now the question of interest is how to acquire the optimal stopping criterion, M^*. We could fix a value for all iterations, but nothing guarantees that it would be the best value considered for the data. Therefore, the literature commonly assigns a maximum value, in our case $M_{max} = 300$, and allows some model selection method to define which is the best model with $1 \leq M \leq 300$. To perform pseudo-out-of-sample forecasting, we have to select the best model without knowing the forecast errors. For this, the most commonly used methods are AIC and k-fold cross-validation. Given recent research highlighting the use of k-fold cross-validation over AIC for the boosting methodology, such as (Wohlrabe & Buchen, 2014), we opted for cross-validation. The k-fold method separates the training set into k different parts, trains the model on $k-1$ parts, and tests the performance on the remaining part (Stone, 1974).

One decision point about the forecasting procedure is whether to use an expanding window or a rolling window. Both expanding and rolling window estimates increase the last index of the train set in one for each new estimate to be performed in the iteration. The difference is that the expanding window keeps the first index of the train set fixed, while rolling window also increases it by one for each new estimate to be performed, in order to always use a fixed number of observations for all estimations. As we are using the boosting methodology that performs variable selection, we follow the line of other articles such as Lehmann and Wohlrabe (2017) and we perform an expanding window on all our forecasts. For this, consider T to be the total number of observations collected (most cases $T = 288$), T_1 to be the t index for the first observation in the test set, and T_2 to be the t index for the last observation also in the test set. For most of our series, T_1 represents January 2014 and T_2 December 2019. The advantage of this method is that we always use all the information available until the t-th period of the estimation. It will expand by unity at each i-th iteration of the estimation, to always use all information available at time t, i.e periods $1, ..., (T_1 - 1) + i$.

Finally, to avoid high computational processing time, we chose to perform direct prediction by rounds, where we update the model parameters only once every 12 months. We performed the prediction in 6 rounds since we have

72 observations in our test set. With this, we start estimating at iteration $i = 1$ using all available information from $t = 1$ to $t = T_1 - 1$ to fit the model. With the estimated parameters, we perform the prediction for steps $h = 1, ..., 12$. After the 12 predictions, we expand the training set by 12 observations and re-estimate a new model, with information from $t = 1$ (since we are using expanding window) to $t = T_1 + 11$, and perform the predictions for the next periods $t = T_1 + 13, T_1 + 14, ..., T_1 + 24$. We keep doing this until we cover the entire test set, thus until T_2. Our code is available upon request, and part of the code was adapted from the code provided by Kauppi and Virtanen (2021).

4.3 Evaluation

We aim to check the performance of boosting methods, especially the nonlinear and two-stage boosting methods. Since our interest is to see how well these models perform, we selected the linear estimation presented in Sect. 3.1 as a benchmark. This means that, when possible, we seek to compare the errors and predictive performance of the other proposed models with the linear model. To do this, we need consistent indicators that we can use to compare different models. Among the range of indicators, we have chosen one that is commonly accepted, the root mean squared error (RMSE) because it shows the average deviation between the predicted values and the actual values. From Sect. 4.2, we know that $T_1 < T_2$, then we can define RMSE as follows:

$$\text{RMSE}_h^{\text{model}} = \sqrt{\left(T_2 - T_1 + 1\right)^{-1} \sum_{i=T_1}^{T_2} \left(\text{FE}_{t+h,i}^{\text{model}}\right)^2}, \tag{12}$$

where $\text{FE}_{t+h,k}^{\text{model}} = y_{t+h,i} - \hat{y}_{t+h,i}^{model}$ represents the multi-step forecast error for a given i observation. The smaller the RMSE, the lower the forecast error and the better the model. For the sake of simplifying the indicator and the comparison with the benchmark, consider the following relative indicator (rRMSE):

$$\text{rRMSE}_h = \frac{\text{RMSE}_h^{\text{others}}}{\text{RMSE}_h^{\text{linear}}}. \tag{13}$$

In this modified indicator, when, for a given model other than the linear one, we have rRMSE < 1, it means that the $\text{RMSE}_h^{\text{model}} < \text{RMSE}_h^{\text{linear}}$, so the RMSE indicator of the model to be compared performs better than the linear model. The opposite is true, hence rRMSE > 1 means that the linear model performs better than the selected model.

A proxy to estimate the predictability of the models performed is the coefficient of determination. It is through this indicator that we can have an assessment of the performance of the models used in the prediction of each of the variables

collected using only their history. The authors Kauppi and Virtanen (2021) define the empirical coefficient of determination that we use for given p in this paper as:

$$\widehat{R}^2(\widehat{g}_h^{\text{model}}(x_t)) = 1 - \frac{\widehat{\text{MSE}}(\widehat{g}_h^{\text{model}}(x_t))}{\widehat{\text{var}}(y_{t+h})} \tag{14}$$

where R^2 is the coefficient of determination and $\widehat{\text{var}}(y_{t+h})$ is the variance of the sample from the actual values outside the sample. Also, t varies from T_1 to T_2, where T_1 is the index for the beginning of the test set, and T_2 is the last index of the test set in each variable. If the estimated coefficient $\widehat{R}^2\left(\widehat{g}_h^{\text{model}}(x_t)\right)$ is less than 0, we replace the value by 0.

There are currently no econometric tests yet to compare in a generic way the predictive ability of one model compared to another regardless of the model specifications. The closest to this was the Giacomini-White test, which measures the statistical significance of the difference in the predictive ability of two models (Giacomini & White, Nov 2006). According to the authors,

> We implement this different focus by conducting inference about conditional, rather than unconditional, moments of forecasts and forecast errors. Recognizing that even a good model may produce bad forecasts due to estimation uncertainty or model instability, we make the object of evaluation the entire forecasting method (including the model, the estimation procedure, and the size of the estimation window), whereas the existing literature concentrates solely on the model. In so doing, we are also able to handle more general data assumptions (heterogeneity rather than stationarity) and estimation methods, as well as provide a unified framework for comparing forecasts based on nested or non-nested models, which were not previously available. (Giacomini & White, Nov 2006, pg.23)

We implement this statistical test from the "afmtools" package in R (Contreras-Reyes et al., 2013). The null hypothesis of the Giacomini-White test states that the predictive ability of two models is equal, while the alternative hypothesis when using the one-sided statistic, considers that one model has greater predictive power than the other. A point worth noting regarding the use of this test in our work is the same one raised by the authors Kauppi and Virtanen (2021),

> One problem with the GW test is that its validity rests on the assumption of "nonvanishing estimation errors" and it is thus designed for situations where the underlying simulated out-of-sample prediction errors are obtained by using a fixed (or a rolling) window rather than an expanding window estimation scheme applied here." (pg.13)

Even if the scenario where we apply the Giacomini-White test is not ideal, we believe that it can be used as a reference because we calculate other indicators specified here in this section, which also serve to ascertain the predictive power. Through them, we can get an idea about the performance of each of the models in the 140 verified series, and we can have a way to compare the models among
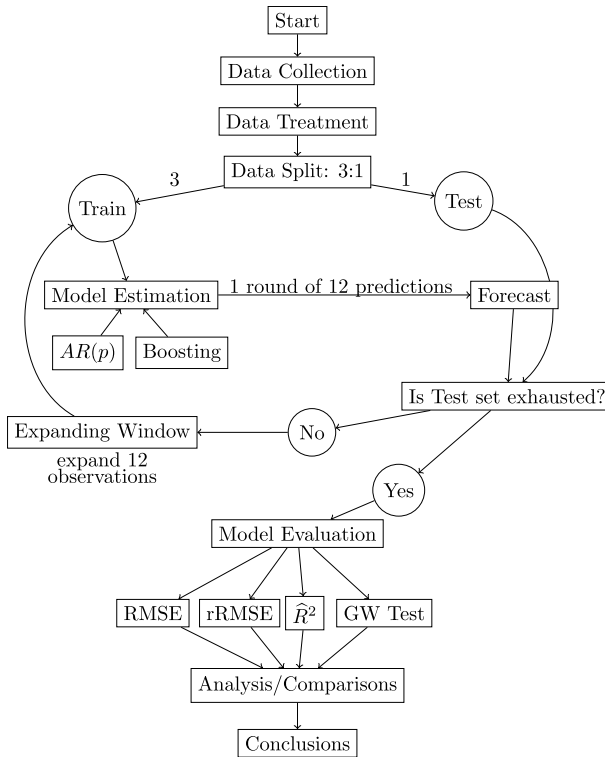
Fig. 1 Methodology flowchart

themselves. Figure 1 provides a visual representation of the research steps and summarizes the methodology applied.

5 Empirical Findings

Here we analyze the performance of each of the models applied in forecasting the 140 variables collected for the test period, which is from January 2014 to December 2019. The models used are explained in the methodology section (see Sect. 3), i.e. simple linear estimation (Linear), linear boosting (BOLS), boosting with splines as base-learner (BSpline) and two-stage boosting (TSBoost). Tables 3, 4 and 5 contain other models: BSpline* and TSBoost*, which refer, respectively, to boosting with splines without extrapolation and two-stage boosting without extrapolation, and finally Tree, which is boosting with regression trees as base-learner (more details in Sects. 5.1 and 5.2).

A result that better evokes the predictive performance of each model in its own history is the empirical coefficient of determination R^2 calculated from the prediction error and variance of the test set sample, as defined in Eq. 14. According to the R^2
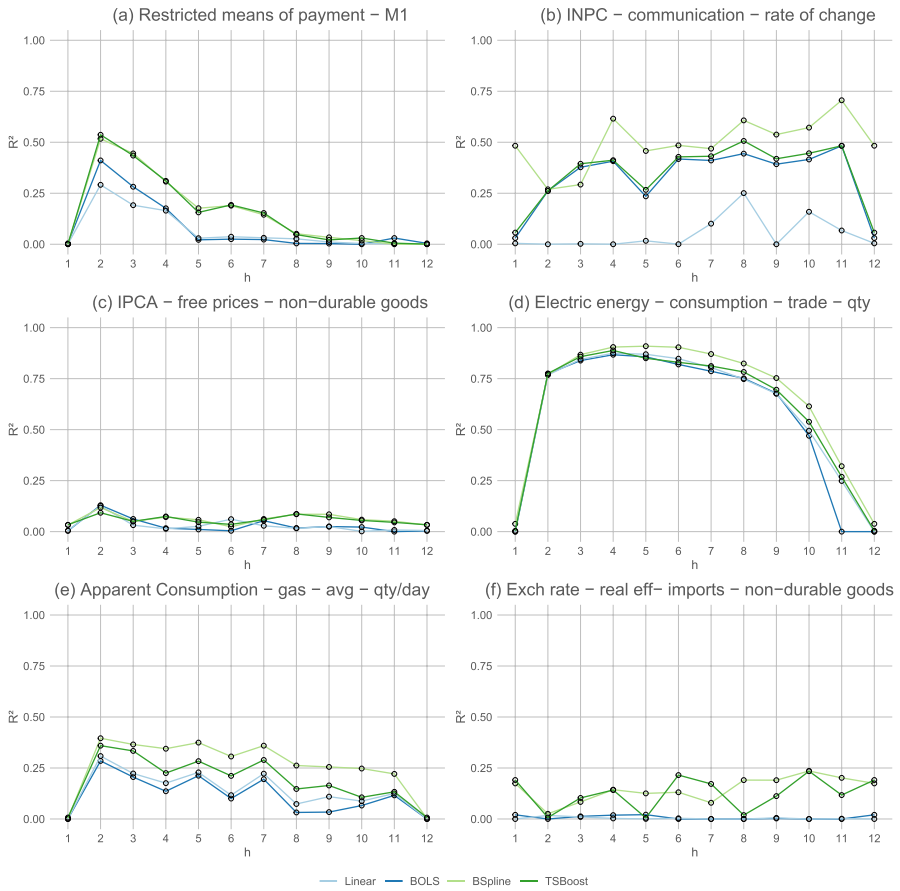
Fig. 2 Selected plots 1—period 2014–2019—R^2

methodology, it is assumed that the higher the variance of a variable, the more difficult it is to predict. Therefore, for highly volatile series, there is a certain compensation at the moment of coefficient calculation (see details in Sect. 4.3). Since we did the estimation for all variables in our database, there are 140 different plots. To allow us to select some of them to present here, we made a special condition for choosing them.[3] Figs. 2 and 3 contain some plots where the *y*-axis is the estimated coefficient R^2 and the *x*-axis is the forecast horizon $h = 1, \ldots, 12$.

From the figures, we can see that there seems to be a combined movement between the linear models (Linear and BOLS) and the nonlinear models (BSpline and TSBoost). Furthermore, since our goal is to test the validity and applicability

---

[3] The selected graphs are obtained for variables where the logical condition below is realized for at least 9 or more different h's. Let h be in $\{1, \ldots, 12\}$ and P be the models with P-Spline as base-learner, then

$$(R^2_{P|h} > R^2_{\text{Linear}|h}) \cap (R^2_{P|h} > R^2_{\text{BOLS}|h}).$$
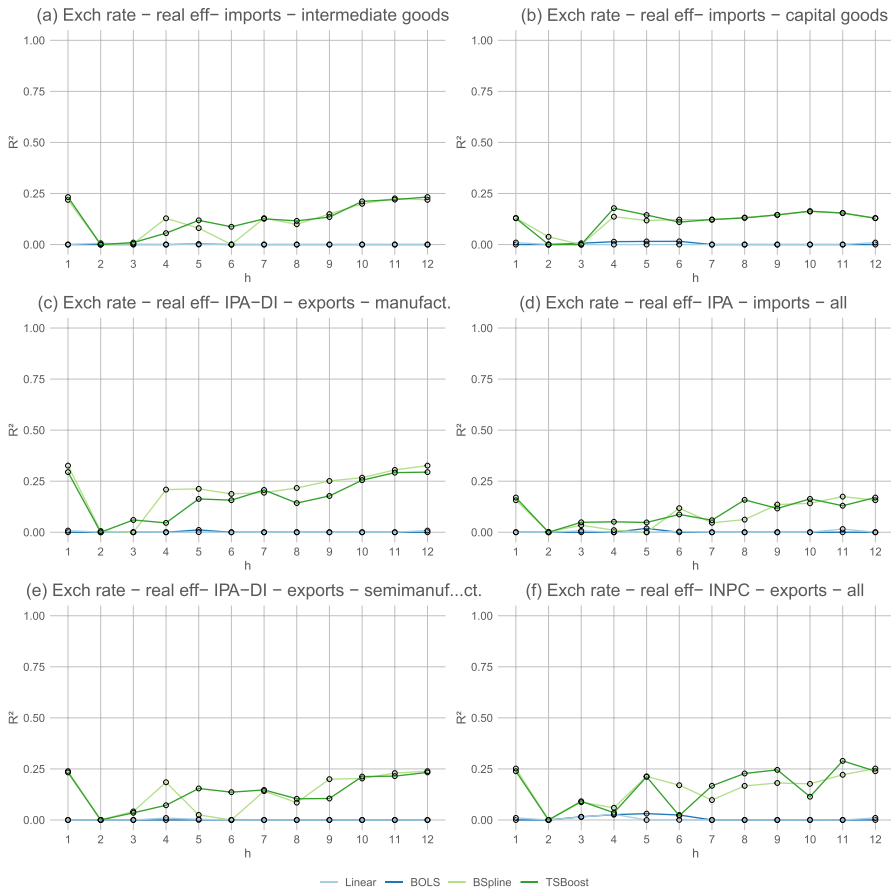
Fig. 3 Selected plots 2—period 2014–2019—R^2

of boosting methodology, especially with nonlinear base-learners (P-Spline) to forecast macroeconomic variables, we are focusing the analysis on the cases where the performance of these models is better. With this, we can visually see if there are differences in the curves estimated by the different models for the selected variables. Both Figs. 2 and 3 show graphs where there is a clear difference between the estimation with the linear base-learner and with splines as base-learner.

In particular, in Fig. 2, we can see that, for example, the curves in graphs (a), (d), and (e) present the greatest difference between the estimation methods. The nature of the variables is about the monetary base, apparent gas consumption, and electricity consumption, hence, from different types of economic data. Nevertheless, these are scenarios where estimation with splines has a clear advantage. Curve (d) presents the best determination coefficient, reaching values above 0.80 depending on the forecast horizon. Another point of emphasis is the low value

Table 3 R^2 from the data set

R^2 min	Method	h=1	h=2	h=3	h=4	h=5	h=6	h=7	h=8	h=9	h=10	h=11	h=12
		(a) All cases											
0	Linear	0.33	0.35	0.35	0.34	0.32	0.29	0.30	0.30	0.28	0.26	0.24	0.11
	BOLS	0.30	0.33	0.32	0.33	0.31	0.27	0.29	0.29	0.27	0.26	0.23	0.11
	BSpline	0.30	0.31	0.31	0.32	0.30	0.27	0.29	0.29	0.28	0.28	0.25	0.13
	TSBoost	0.33	0.34	0.34	0.34	0.32	0.29	0.30	0.30	0.29	0.28	0.26	0.13
	BSpline*	0.30	0.32	0.32	0.33	0.31	0.28	0.30	0.30	0.29	0.28	0.26	0.13
	TSBoost*	0.33	0.34	0.34	0.34	0.32	0.29	0.31	0.30	0.29	0.28	0.26	0.13
	Tree	0.25	0.26	0.26	0.26	0.24	0.21	0.23	0.22	0.21	0.22	0.20	0.09
	N	140	140	140	140	140	140	140	140	140	140	140	140
0.1	Linear	0.43	0.46	0.49	0.52	0.52	0.52	0.53	0.54	0.48	0.49	0.47	0.50
	BOLS	0.40	0.45	0.47	0.48	0.50	0.50	0.51	0.50	0.47	0.49	0.46	0.50
	BSpline	0.41	0.44	0.45	0.45	0.47	0.42	0.44	0.46	0.44	0.42	0.40	0.38
	TSBoost	0.44	0.48	0.51	0.52	0.50	0.51	0.49	0.48	0.45	0.45	0.41	0.41
	BSpline*	0.41	0.44	0.46	0.45	0.47	0.42	0.45	0.47	0.43	0.42	0.40	0.36
	TSBoost*	0.44	0.48	0.50	0.51	0.50	0.51	0.50	0.48	0.46	0.46	0.41	0.40
	Tree	0.38	0.42	0.42	0.42	0.40	0.39	0.40	0.39	0.40	0.37	0.34	0.32
	N	101	98	92	89	88	79	85	85	88	85	85	43
		(b) Splines more accurate on average											
0	Linear	0.35	0.33	0.31	0.32	0.31	0.26	0.29	0.28	0.26	0.25	0.24	0.14
	BOLS	0.33	0.32	0.30	0.31	0.31	0.25	0.29	0.28	0.26	0.26	0.25	0.14
	BSpline	0.35	0.34	0.33	0.35	0.33	0.28	0.31	0.32	0.30	0.32	0.32	0.19
	TSBoost	0.36	0.35	0.33	0.34	0.33	0.29	0.31	0.31	0.29	0.31	0.31	0.19
	BSpline*	0.36	0.34	0.33	0.35	0.33	0.28	0.32	0.32	0.30	0.32	0.31	0.20
	TSBoost*	0.36	0.35	0.33	0.34	0.33	0.29	0.31	0.31	0.29	0.30	0.31	0.19
	Tree	0.30	0.30	0.27	0.28	0.26	0.20	0.25	0.26	0.24	0.24	0.24	0.14
	N	75	68	67	78	84	80	76	88	82	82	74	71
0.1	Linear	0.46	0.54	0.50	0.52	0.59	0.52	0.54	0.55	0.47	0.51	0.46	0.49
	BOLS	0.43	0.52	0.49	0.51	0.57	0.50	0.53	0.56	0.47	0.52	0.47	0.50
	BSpline	0.46	0.53	0.51	0.52	0.57	0.48	0.52	0.55	0.49	0.53	0.47	0.47
	TSBoost	0.47	0.56	0.52	0.53	0.59	0.53	0.54	0.56	0.49	0.53	0.48	0.52
	BSpline*	0.46	0.53	0.51	0.52	0.57	0.49	0.53	0.56	0.49	0.53	0.47	0.48
	TSBoost*	0.47	0.56	0.51	0.53	0.59	0.54	0.54	0.56	0.49	0.53	0.48	0.51
	Tree	0.40	0.47	0.44	0.45	0.48	0.37	0.43	0.46	0.40	0.42	0.40	0.39
	N	55	40	39	45	40	36	38	42	41	37	36	19
		(c) Linear more accurate on average											
0	Linear	0.30	0.36	0.38	0.37	0.34	0.34	0.33	0.33	0.32	0.28	0.23	0.09
	BOLS	0.26	0.33	0.35	0.35	0.31	0.31	0.30	0.29	0.30	0.27	0.21	0.08
	BSpline	0.23	0.29	0.29	0.28	0.26	0.26	0.25	0.25	0.25	0.22	0.18	0.06
	TSBoost	0.28	0.34	0.35	0.34	0.31	0.30	0.29	0.29	0.29	0.25	0.21	0.07
	BSpline*	0.24	0.30	0.30	0.30	0.28	0.28	0.27	0.26	0.26	0.23	0.19	0.06
	TSBoost*	0.28	0.34	0.36	0.34	0.31	0.30	0.30	0.30	0.29	0.26	0.21	0.08
	Tree	0.18	0.23	0.25	0.23	0.22	0.22	0.21	0.21	0.21	0.20	0.15	0.04
	N	65	72	73	62	56	60	64	52	58	58	66	69
0.1	Linear	0.45	0.50	0.55	0.58	0.60	0.70	0.62	0.64	0.70	0.58	0.51	0.63
	BOLS	0.40	0.47	0.51	0.54	0.56	0.64	0.58	0.59	0.66	0.56	0.49	0.56
	BSpline	0.38	0.42	0.44	0.46	0.48	0.57	0.51	0.54	0.58	0.48	0.43	0.47
	TSBoost	0.43	0.48	0.53	0.54	0.58	0.66	0.58	0.61	0.67	0.55	0.48	0.57
	BSpline*	0.38	0.43	0.46	0.47	0.50	0.60	0.53	0.54	0.60	0.49	0.44	0.47
	TSBoost*	0.43	0.48	0.53	0.54	0.58	0.66	0.58	0.60	0.67	0.55	0.48	0.57
	Tree	0.30	0.34	0.36	0.37	0.39	0.48	0.42	0.43	0.45	0.38	0.35	0.28
	N	39	47	48	37	29	26	31	23	24	26	27	8

Automatic colors, where greener is relatively better and redder worse

*Splines prediction without extrapolation

of R^2 when $h = 1$. Mispecifications when $h = 1$ show one of the reasons why we chose to perform directly as opposed to recursive forecasting. For subsequent h's, we have, for most cases, a higher value for R^2.

In Fig. 3, we have a selection of variables where all are specifications of the real effective exchange rate. According to the methodology of the time series presented in Nonnenberg (2015), we can define the real effective exchange rate as the weighted

average of Brazil's bilateral real exchange rates against each of its major trading partners. The weighting of each country is given by the share of each country in Brazilian exports or imports. The Brazilian inflation index used is the INPC/IBGE. This indicator brings information about the level of prices, inflation, and Brazil's main trading partners into one index. According to the figure, we can see that there is a considerable advantage for nonlinear methods over linear methods concerning the R^2 value in almost all horizons. Moreover, an interesting point is that they are indicators with low predictive power if using linear methods since we can see that, for example, curves (a) and (e) have $R^2 = 0$ for almost all h's. In contrast, boosting with splines and two-stage boosting have relatively high values for the coefficient, especially at $h = 1$ and $h = 12$ in all scenarios.

Table 3 shows the calculated R^2 as the average of all selected series given the forecast horizon and model. As done in Kauppi and Virtanen (2021), we separated in minimum R^2, which can be 0 or 0.1. We did this because we can distinguish series where forecasting using the series' own history is difficult with the models used, and that means the value for R^2 is closer to 0. We apply colors to the tables to make it easier to visualize the results. The closer the coefficient is to 1, the better the prediction given h and the model considered, so the greener the cell in the table. The worse it is, the redder the cell is. In part (a), considering all cases (number of observations $N = 140$), we find that the best models for all series are the TSBoost (mean 0.295) and the linear model (mean 0.290), followed by the BSpline model (mean 0.285). What is worth noting is that the TSBoost average is higher than the Linear's and the BSpline's, which shows that it is a powerful model to use when one does not know the best model to estimate for a specific series, because on average it is the most accurate. Part (b) of Fig. 3 shows us the selection for the subset where series with splines (either BSpline or TSBoost) are more accurate on average. We can see that N has dropped from 140 to 77 on average, i.e., 55% of the series used. As expected, we can see that as much as for the R^2 equal to 0 or 0.1, we have that the performance of models with splines is considerably better, having means equal to 0.31 while linear models have means equal to 0.27. In case (c), we select the series whose indicators for the linear models are superior. We then obtained 45% of all the series on N average, but we can see that the performance of the Linear model is superior to all the others, including the BOLS model. We comment on the results for the models without extrapolation and the Tree model in subsequent sections. One thing to note here in part (c) of the table, where we have the selection of series where the Linear model is better, the performance of the TSBoost model is almost as good as the linear model. So, with these results, for selection (b), the TSBoost model is as good as the BSPline, and for part (c), the TSBoost model is almost as good as the linear one, making it a suitable model in general.

Another way to visualize the predictive performance is from the estimated rRMSE of the series. Table 4 presents the indicator values compared to the Linear model, which we consider as a benchmark. The table shows the average rRMSE of N observations considering the model and the forecast step. In general, part (a) shows us that the Linear model is, on average, superior to all models from $h = 1$ to $h = 7$. Thereafter, we have marginal superiority to the TSBoost model. Note that BSpline is not better on average in any case compared to the Linear model. In scenario (b), on

Table 4 rRMSE relative to linear model

Method	h=1	h=2	h=3	h=4	h=5	h=6	h=7	h=8	h=9	h=10	h=11	h=12
(a) All cases												
BOLS	1.029	1.028	1.039	1.064	1.085	1.100	1.082	1.058	1.039	1.029	1.031	1.020
BSpline	1.023	1.038	1.064	1.056	1.056	1.063	1.056	1.050	1.048	1.039	1.035	1.026
TSBoost	1.004	1.009	1.002	1.012	1.010	1.015	1.003	0.993	0.996	0.992	0.989	0.981
BSpline*	1.016	1.027	1.042	1.038	1.043	1.046	1.044	1.040	1.040	1.032	1.029	1.020
TSBoost*	1.002	1.001	0.998	1.003	1.000	1.002	0.996	0.991	0.991	0.988	0.984	0.979
Tree	1.104	1.127	1.149	1.156	1.161	1.179	1.175	1.161	1.167	1.171	1.150	1.070
N	140	140	140	140	140	140	140	140	140	140	140	140
(b) Splines more accurate on average												
BOLS	1.016	1.013	1.029	1.054	1.042	1.024	1.024	1.005	0.998	1.005	1.028	1.006
BSpline	0.987	0.988	1.003	1.001	0.997	0.986	0.981	0.981	0.972	0.973	0.994	0.986
TSBoost	0.990	0.993	0.980	0.983	0.984	0.978	0.979	0.971	0.969	0.959	0.954	0.947
BSpline*	0.983	0.986	0.998	0.999	0.992	0.982	0.979	0.974	0.968	0.972	0.993	0.981
TSBoost*	0.990	0.980	0.978	0.983	0.984	0.977	0.971	0.969	0.968	0.960	0.952	0.946
Tree	1.037	1.039	1.072	1.088	1.068	1.058	1.061	1.050	1.030	1.058	1.073	1.034
N	80	73	72	85	94	91	89	100	95	93	89	94
(c) Linear more accurate on average												
BOLS	1.047	1.045	1.050	1.079	1.174	1.242	1.183	1.188	1.124	1.077	1.036	1.048
BSpline	1.072	1.094	1.128	1.143	1.178	1.207	1.187	1.223	1.209	1.169	1.108	1.109
TSBoost	1.022	1.027	1.026	1.057	1.064	1.085	1.044	1.048	1.054	1.057	1.051	1.052
BSpline*	1.061	1.071	1.089	1.100	1.146	1.166	1.159	1.203	1.190	1.152	1.093	1.101
TSBoost*	1.019	1.025	1.020	1.033	1.034	1.048	1.040	1.046	1.041	1.045	1.039	1.045
Tree	1.193	1.222	1.230	1.261	1.350	1.403	1.375	1.441	1.456	1.394	1.285	1.143
N	60	67	68	55	46	49	51	40	45	47	51	46

Automatic colors, where greener is relatively better and redder worse

*Splines prediction without extrapolation

Table 5 Giacomini-white test—quantity of series statistically superior to the linear model

p-value	Method	h=1	h=2	h=3	h=4	h=5	h=6	h=7	h=8	h=9	h=10	h=11	h=12
0.90	BOLS	19	19	22	21	24	26	22	31	28	26	21	36
	BSpline	12	13	15	17	20	22	20	29	26	29	25	37
	TSBoost	16	23	21	19	25	23	23	27	30	34	40	40
	BSpline*	14	17	21	22	24	25	21	29	30	32	30	40
	TSBoost*	20	23	22	21	24	23	21	29	30	37	44	40
	Tree	42	48	47	47	40	36	35	33	32	26	24	16
0.95	BOLS	15	12	14	12	13	14	13	19	17	18	16	21
	BSpline	7	7	9	10	13	14	12	20	19	22	21	19
	TSBoost	10	14	12	10	12	14	14	11	23	23	30	22
	BSpline*	9	11	12	14	14	16	12	20	20	22	22	22
	TSBoost*	11	14	15	11	11	14	13	11	23	23	31	24
	Tree	27	34	39	35	29	30	28	27	27	21	23	13

Automatic colors, where greener is relatively better and redder worse

*Splines prediction without extrapolation

the other hand, we are selecting only those series where the estimation with splines is superior on average. For 63% of the total series, the performance for the spline models is better than the linear models for all h's. Scenario (c) brings us the series where the Linear model is better, but we have only 37% of the series. In this case, all other models for all h's have the value of rRMSE > 1, but still, the TSBoost model is superior to the BOLS model on average. Here, the same thing that occurs in the previous table is true again: the TSBoost model is as good as BSpline in part (b), and the TSBoost model is almost as good as the Linear model in part (c). Also, we expose the rRMSE indicator in full tables for each of the variables for the forecast horizons $h = 1, 6, 12$ in Appendix A.

The remaining strategy to analyze the results we report is the Giacomini-White test. This test measures the statistical difference in prediction between the two models. But, as discussed in Sect. 4.3, we are not sure of the validity of the test, since we are expected to do this with a rolling window of estimation. Since the number of observations in the test set is 36, we have a relatively low n, so we believe the results of the test are not dismissible and we chose to show them here. Table 5 shows the results of applying the test. We iteratively ran and applied the test for each of the 140 series, and compared them to the Linear model. The null hypothesis of the test is that there is statistically no difference in prediction between the models, and the alternative hypothesis used is that the tested model is superior to the Linear model. The results show, given the model and given h, the number of series, out of 140, which are statistically superior in forecasting in comparison to the Linear model. We separate them into two significance levels, namely 10%, and 5%. We can easily see that for most series, the test indicts no difference in prediction between the models. But for the series whose superiority is statistically superior, we have, on average, better predictions for the Tree model (we will explain in more detail in Sect. 5.2) and the TSBoost model. As seen from this and the other tables, we can see that the predictive performance of the two-stage boosting model is always high on average, being an excellent candidate to be used in all cases, especially when the nature of the series is unknown (linear or nonlinear).

5.1 Splines Without Extrapolation

From the methodology explained for Sect. 3.2.2, whenever we use the boosting algorithm considering the P-Spline as a base-learner, we are smoothing the noisy data. However, one of the problems we may have when doing smoothing is the consequence of extrapolation. The extrapolation does not necessarily lead to bad results, but for some cases, mostly when estimation with extrapolation occurs, it generates very different values in comparison to when there is an interpolation, causing an increase in the variance of the estimates and consequently reducing the overall performance of the model with splines. One of the corrections, presented by the authors Kauppi and Virtanen (2021) as "Hybrid Model", is whenever extrapolation occurs, to replace the extrapolated estimation with the Linear model estimation. The results of these hybrid models are the models shown in Tables 3, 4, and 5 such as BSpline* and TSBoost*.

From the results, we can see in Table 3 that the models without extrapolation, on average, have marginal gains in predictive power when compared to the models with extrapolation. Even in scenario (b), we always have the BSpline* with superior performance compared to the conventional BSpline in both R^2 scenarios. In Table 4, the TSBoost* model is the most superior model on average in (a) all cases and in (b) splines better on average. Finally, both BSpline* and TSBoost* are better than their counterparts with extrapolation in Table 5, showing that the strategy of removing extrapolation can generate marginal gains in predictive power, and also supporting the view that using the two-stage boosting model is superior on average. Visually, we chose not to show them because qualitatively the curves are very similar to the curves of the splines with extrapolation.
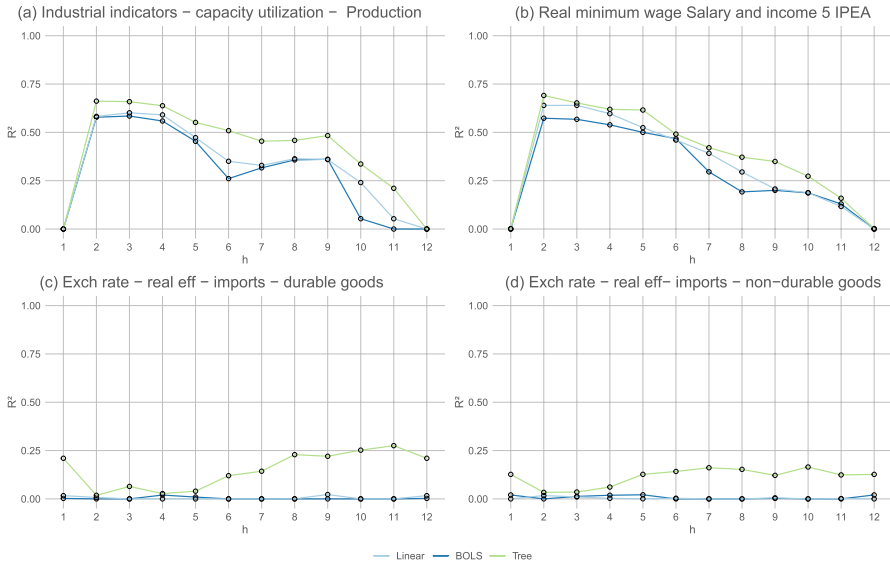
Fig. 4 Selected plots 3—period 2014–2019—R^2

5.2 Robustness Check: Boosting Regression Trees

We perform the estimation with an alternative base-learner that is commonly applied in the boosting methodology: regression trees. The boosting method was augmented and the "mboost" statistical package was improved from the contributions on regression trees of the paper Hothorn et al. (2006). With this, we were able to apply the boosting algorithm and construct a piecewise and recursive estimation of the binary partition model. This is a non-parametric model and potentially estimates a nonlinear relationship between the predicted variable and the regressors. However, according to the authors of the Hothorn et al. (2011) statistical package, "The regression fit is a black box prediction machine and thus hardly interpretable." We chose to apply this method as a robustness check due to its wide applicability in the forecasting literature.

The performance of the Tree model in all scenarios in Tables 3 and 4 is, for all h and compared to all other models, the worst. Our interpretation of the results is, therefore, that the Tree model realizes such a nonlinear relationship between the regressors and the regressed, that when the variable to be estimated is not of nonlinear intrinsic behavior, the Tree model has low predictive power. But on the other hand, Table 5 shows the Tree model as the model that has the most statistically superior predictions to the Linear model, by a relatively large margin. Figure 4 visually shows the performance of the Tree model compared to the linear models for some selected variables, from which we can see that there is a relatively wide visual difference between the models.

Table 6 R^2 from the data set—AIC

R^2 min	Method	h=1	h=2	h=3	h=4	h=5	h=6	h=7	h=8	h=9	h=10	h=11	h=12
		(a) All cases											
0	Linear	0.33	0.35	0.35	0.34	0.32	0.29	0.30	0.30	0.28	0.26	0.24	0.11
	BOLS	0.31	0.33	0.33	0.34	0.32	0.28	0.30	0.30	0.28	0.27	0.24	0.11
	BSpline	0.31	0.32	0.33	0.34	0.32	0.29	0.30	0.31	0.30	0.29	0.26	0.13
	TSBoost	0.33	0.34	0.34	0.34	0.32	0.29	0.30	0.30	0.29	0.28	0.26	0.13
	BSpline*	0.32	0.33	0.34	0.35	0.33	0.30	0.32	0.32	0.30	0.29	0.26	0.13
	TSBoost*	0.33	0.35	0.35	0.34	0.32	0.29	0.31	0.30	0.29	0.28	0.26	0.13
	N	140	140	140	140	140	140	140	140	140	140	140	140
0.1	Linear	0.43	0.46	0.49	0.52	0.52	0.52	0.53	0.54	0.48	0.49	0.47	0.50
	BOLS	0.40	0.44	0.48	0.48	0.48	0.48	0.51	0.49	0.48	0.48	0.47	0.49
	BSpline	0.41	0.45	0.46	0.46	0.45	0.41	0.45	0.45	0.43	0.44	0.40	0.38
	TSBoost	0.44	0.48	0.51	0.52	0.50	0.50	0.49	0.49	0.46	0.45	0.41	0.41
	BSpline*	0.41	0.45	0.47	0.45	0.46	0.42	0.45	0.45	0.43	0.44	0.39	0.37
	TSBoost*	0.44	0.48	0.50	0.52	0.50	0.51	0.50	0.49	0.46	0.45	0.40	0.40
	N	101	98	92	89	88	79	85	85	88	85	85	43
		(b) Splines more accurate on average											
0	Linear	0.36	0.36	0.30	0.32	0.32	0.25	0.28	0.29	0.26	0.26	0.25	0.12
	BOLS	0.35	0.36	0.30	0.32	0.32	0.25	0.29	0.30	0.27	0.27	0.26	0.13
	BSpline	0.37	0.38	0.33	0.36	0.35	0.29	0.33	0.34	0.32	0.33	0.32	0.18
	TSBoost	0.37	0.38	0.31	0.33	0.33	0.27	0.31	0.32	0.29	0.30	0.30	0.17
	BSpline*	0.38	0.38	0.34	0.36	0.36	0.29	0.33	0.35	0.32	0.33	0.32	0.18
	TSBoost*	0.37	0.38	0.31	0.33	0.33	0.27	0.31	0.32	0.29	0.30	0.30	0.17
	N	75	68	67	78	84	80	76	88	82	82	74	71
0.1	Linear	0.46	0.51	0.49	0.52	0.55	0.50	0.53	0.54	0.45	0.49	0.47	0.51
	BOLS	0.44	0.49	0.48	0.52	0.54	0.49	0.53	0.55	0.46	0.50	0.47	0.52
	BSpline	0.46	0.52	0.51	0.55	0.55	0.49	0.53	0.56	0.48	0.53	0.49	0.49
	TSBoost	0.47	0.52	0.50	0.53	0.55	0.52	0.54	0.55	0.46	0.51	0.48	0.54
	BSpline*	0.47	0.52	0.51	0.55	0.56	0.50	0.54	0.57	0.48	0.52	0.48	0.51
	TSBoost*	0.47	0.52	0.50	0.53	0.55	0.52	0.54	0.55	0.46	0.50	0.48	0.53
	N	55	40	39	45	40	36	38	42	41	37	36	19
		(c) Linear more accurate on average											
0	Linear	0.29	0.33	0.40	0.39	0.34	0.36	0.33	0.31	0.32	0.27	0.22	0.10
	BOLS	0.26	0.31	0.37	0.36	0.31	0.33	0.31	0.28	0.30	0.26	0.21	0.09
	BSpline	0.23	0.27	0.32	0.30	0.27	0.29	0.27	0.24	0.26	0.22	0.18	0.07
	TSBoost	0.27	0.31	0.38	0.36	0.31	0.32	0.30	0.27	0.29	0.25	0.20	0.08
	BSpline*	0.24	0.29	0.34	0.32	0.29	0.31	0.29	0.26	0.28	0.23	0.19	0.07
	TSBoost*	0.27	0.31	0.38	0.36	0.31	0.33	0.31	0.28	0.30	0.25	0.20	0.08
	N	65	72	73	62	56	60	64	52	58	58	66	69
0.1	Linear	0.45	0.50	0.57	0.58	0.61	0.69	0.62	0.65	0.73	0.61	0.50	0.62
	BOLS	0.41	0.47	0.53	0.54	0.56	0.63	0.58	0.60	0.69	0.59	0.48	0.55
	BSpline	0.38	0.42	0.48	0.48	0.51	0.57	0.52	0.55	0.64	0.54	0.44	0.48
	TSBoost	0.42	0.48	0.55	0.55	0.59	0.64	0.58	0.61	0.70	0.58	0.46	0.56
	BSpline*	0.38	0.44	0.50	0.49	0.53	0.60	0.55	0.56	0.65	0.54	0.45	0.49
	TSBoost*	0.42	0.48	0.55	0.55	0.59	0.64	0.59	0.61	0.71	0.58	0.47	0.56
	N	39	47	48	37	29	26	31	23	24	26	27	8

Automatic colors, where greener is relatively better and redder worse

*Splines prediction without extrapolation

5.3 Robustness Check: Comparing AIC with Cross-Validation

To make another robustness check of the results, we chose to elaborate and present the table of the empirical coefficient of determination R^2 calculated for the estimations, where the $M_{stop} = M^*$ of the boosting algorithm was acquired through the AIC methodology, as opposed to k-fold cross-validation. Using Table 6, we can view the results in a manner analogous to Table 3. We do not present the AIC method for selecting M^* for the model boosting regression trees.

Starting from scenario (a) and comparing the results between the two tables, we can see that, in general, the R^2 results obtained a marginal gain, but the average for

TSBoost remained the same. The model that obtained the most gains in scenario (a) was BSpline, but in scenario (a) with $R^2\text{min} = 0.1$, the results are almost identical. In part (b), again the BSpline model obtained some marginal gains at $R^2\text{min} = 0$, but at $R^2\text{min} = 0.1$ most models had a small relative loss. Finally, in part (c), all models on average had a small gain. The use of AIC found approximately 58% of the database to have more accurate splines on average, considering all h's. Overall, both results from using AIC or cross-validation to select the $\text{M}_{stop} = M^*$ for each estimation are bringing qualitatively similar and numerically very close results.

5.4 Robustness Check: Data Ranging Until 2022

Finally, we chose to focus the research on data within the interval from 1996 to 2019, but the period from 2020 onwards is particularly special as it encompasses the coronavirus pandemic and adds a challenge for forecasting. That said, the main loss of studying the Brazilian variables with the data through 2022 is that we would lose several variables whose values are not yet available. On the other hand, as a robustness test of our findings, we chose to conduct the same analysis for the subset of the data that have their values updated monthly through December 2022. By doing so, the data set dropped from 140 variables to 78. We set the beginning of the training set as the same year (2014) for ease of comparison. We include two tables in Appendix A exposing the average values per model per forecast horizon similar to the tables already presented. The first table (Table 7) presents the R^2 scores for the estimations from 2014 through 2022, and Table 8 presents the R^2 scores for the estimations focusing only on the additional period from 2020 through 2022. Overall, the results obtained are similar to those found at the core of the research. The main difference is that all models suffer a proportional loss of accuracy when forecasting 2020 through 2022, which is expected due to the crisis. Specifically, the TSBoost model remains, on average, the best model in predictability. When analyzing the cases where BSpline is more accurate, TSBoost is as accurate as using only BSpline. When filtering out the cases where the linear model is more accurate, TSBoost is the second-best model. We can conclude that the uncertainty generated by the pandemic and its derived crisis has not changed the conclusions obtained by our research.

6 Conclusion

This work aimed to elaborate and treat a new and recent Brazilian data set with macroeconomic time series in a way that follows the molds of other economic and econometric research. Then, the application of boosting methodology to verify the linear and nonlinear predictability of the variables studied. And, finally, the application of the two-stage boosting model, a model initially formalized by Kauppi and Virtanen (2021) and having its performance verified by the empirical coefficient of determination indicator, calculated based on the out-of-sample forecasts. As secondary objectives, we had the application of robustness checking of the selected models from the comparison with other models, namely the hybrid

models—splines without extrapolation—and the trees regression as base-learner, as well as the comparison of the applied methodology of selecting the optimal model via cross-validation with the AIC method.

We can say that the application of the models obtained solid results for all the models applied, but, on average, the two-stage boosting modeling had the highest performance. Thus, we can say that this model is the best candidate among the models used here to forecast Brazilian macroeconomic variables with their own history. As presented in the results part of this paper, the two-stage boosting model achieved high performance generally when boosting with splines also performed well, but it also retained as strong predictive power together with the linear model in cases where the linear model was superior to purely nonlinear modeling. Also, regarding boosting with splines, it is a strong base-learner method with good results for our case, especially for series where nonlinear forecasting is a good model. Modifying the modeling to remove the extrapolations caused by splines can marginally increase the predictive power, depending on the linear model prediction, or whichever way the prediction is replaced. Additionally, between the AIC or cross-validation selection methods in time series, we can conclude that their use does not qualitatively change the results, but only marginally the values, both of which can be used to obtain close results. Finally, as a robustness check, we conducted an analysis using data ranging until 2022, which includes the period of the coronavirus pandemic. Despite a loss of accuracy when forecasting 2020 through 2022, the results obtained using this data set are consistent with our main findings, with the two-stage boosting model continuing to have the highest performance. Therefore, we can more confidently conclude that our research has successfully applied and tested various models for forecasting Brazilian macroeconomic variables, with the two-stage boosting model remaining the best candidate among the models used for the exercise.

It is worth noting that the conclusions obtained here are limited to the applied context and that further study of the results of the methodology is therefore desired. However, taking into account the research conducted on Kauppi and Virtanen (2021) regarding macroeconomic variables from the United States and our research on Brazilian macroeconomic variables, we are moving in the direction of consensus between the results. We suggest future research for applying the modeling in other contexts, as well as looking for ways that can be done to further improve the two-stage boosting model.

Appendix A

Tables 7 and 8 present the R^2 score for the 78 Brazilian macroeconomic variables that have data ranging from 1996 through 2022. Table 7 calculates the values with the estimates being from the interval 2014 through 2022 and Table 8 calculates the scores with the interval being only the additional period 2020 through 2022.

Table 7 R^2 from the data until 2022—score 2014–2022

R^2 min	Method	h=1	h=2	h=3	h=4	h=5	h=6	h=7	h=8	h=9	h=10	h=11	h=12
	(a) All cases												
0	**Linear**	0.25	0.25	0.27	0.25	0.25	0.23	0.24	0.24	0.24	0.22	0.21	0.13
	BOLS	0.23	0.23	0.24	0.24	0.23	0.21	0.22	0.23	0.23	0.21	0.20	0.12
	Bspline	0.23	0.22	0.23	0.23	0.22	0.21	0.22	0.23	0.24	0.23	0.23	0.16
	TSBoost	0.25	0.25	0.27	0.26	0.25	0.24	0.25	0.25	0.26	0.25	0.25	0.16
	Bspline*	0.23	0.22	0.24	0.24	0.23	0.21	0.22	0.23	0.24	0.23	0.23	0.16
	TSBoost*	0.25	0.25	0.27	0.26	0.25	0.24	0.25	0.25	0.26	0.25	0.25	0.16
	Tree	0.18	0.18	0.18	0.17	0.18	0.16	0.17	0.18	0.18	0.17	0.17	0.10
	N	78	78	78	78	78	78	78	78	78	78	78	78
0.1	**Linear**	0.35	0.39	0.41	0.45	0.44	0.42	0.42	0.42	0.42	0.38	0.41	0.37
	BOLS	0.34	0.36	0.39	0.40	0.43	0.40	0.41	0.40	0.42	0.39	0.41	0.40
	Bspline	0.35	0.34	0.38	0.38	0.37	0.33	0.35	0.35	0.33	0.32	0.32	0.32
	TSBoost	0.35	0.38	0.42	0.45	0.43	0.38	0.38	0.38	0.37	0.35	0.34	0.34
	Bspline*	0.34	0.34	0.38	0.38	0.38	0.34	0.36	0.34	0.34	0.32	0.32	0.33
	TSBoost*	0.35	0.38	0.42	0.44	0.43	0.38	0.39	0.39	0.37	0.35	0.35	0.34
	Tree	0.33	0.36	0.35	0.32	0.31	0.29	0.29	0.31	0.30	0.29	0.28	0.26
	N	53	51	48	42	43	46	48	50	54	55	55	35
	(b) Splines more accurate on average												
0	**Linear**	0.24	0.25	0.23	0.24	0.24	0.21	0.20	0.19	0.23	0.22	0.19	0.10
	BOLS	0.24	0.24	0.22	0.23	0.23	0.20	0.19	0.19	0.23	0.21	0.18	0.10
	Bspline	0.26	0.24	0.23	0.26	0.26	0.23	0.24	0.24	0.29	0.29	0.26	0.20
	TSBoost	0.26	0.26	0.25	0.26	0.26	0.24	0.24	0.23	0.29	0.30	0.26	0.18
	Bspline*	0.26	0.24	0.23	0.26	0.26	0.23	0.23	0.23	0.28	0.28	0.25	0.19
	TSBoost*	0.26	0.26	0.25	0.26	0.26	0.24	0.23	0.23	0.28	0.29	0.25	0.17
	Tree	0.22	0.21	0.19	0.20	0.22	0.18	0.19	0.19	0.22	0.22	0.18	0.12
	N	45	48	49	52	48	50	47	47	46	44	47	41
0.1	**Linear**	0.39	0.39	0.44	0.48	0.47	0.42	0.39	0.39	0.47	0.43	0.40	0.41
	BOLS	0.38	0.38	0.43	0.46	0.46	0.41	0.40	0.38	0.47	0.44	0.40	0.41
	Bspline	0.41	0.37	0.43	0.46	0.47	0.40	0.39	0.40	0.46	0.43	0.38	0.42
	TSBoost	0.41	0.41	0.46	0.49	0.49	0.43	0.40	0.40	0.49	0.45	0.42	0.45
	Bspline*	0.41	0.38	0.43	0.47	0.47	0.39	0.38	0.38	0.45	0.43	0.39	0.42
	TSBoost*	0.41	0.41	0.46	0.49	0.48	0.42	0.39	0.40	0.48	0.45	0.41	0.44
	Tree	0.35	0.33	0.38	0.39	0.41	0.34	0.32	0.31	0.37	0.35	0.31	0.30
	N	26	28	23	24	23	22	22	23	21	20	19	9
	(c) Linear more accurate on average												
0	**Linear**	0.26	0.26	0.33	0.27	0.26	0.27	0.29	0.30	0.25	0.22	0.25	0.16
	BOLS	0.21	0.22	0.27	0.24	0.24	0.24	0.27	0.28	0.23	0.20	0.23	0.14
	Bspline	0.19	0.17	0.22	0.16	0.16	0.16	0.19	0.20	0.16	0.15	0.19	0.11
	TSBoost	0.24	0.24	0.30	0.24	0.23	0.24	0.26	0.28	0.22	0.20	0.23	0.14
	Bspline*	0.19	0.19	0.25	0.20	0.19	0.18	0.21	0.23	0.18	0.17	0.20	0.11
	TSBoost*	0.24	0.24	0.30	0.25	0.24	0.25	0.27	0.28	0.23	0.20	0.23	0.14
	Tree	0.13	0.12	0.16	0.12	0.11	0.11	0.15	0.16	0.13	0.11	0.14	0.07
	N	33	30	29	26	30	28	31	31	32	34	31	37
0.1	**Linear**	0.38	0.45	0.48	0.51	0.47	0.49	0.53	0.53	0.44	0.41	0.49	0.62
	BOLS	0.33	0.41	0.43	0.46	0.43	0.46	0.50	0.50	0.41	0.39	0.47	0.57
	Bspline	0.30	0.33	0.35	0.36	0.31	0.34	0.38	0.39	0.32	0.32	0.40	0.50
	TSBoost	0.36	0.43	0.46	0.48	0.44	0.46	0.50	0.51	0.42	0.39	0.48	0.60
	Bspline*	0.30	0.35	0.41	0.39	0.34	0.38	0.41	0.42	0.34	0.32	0.41	0.51
	TSBoost*	0.36	0.43	0.47	0.48	0.44	0.47	0.51	0.51	0.42	0.39	0.48	0.60
	Tree	0.22	0.22	0.25	0.24	0.21	0.21	0.28	0.29	0.23	0.21	0.30	0.34
	N	18	14	17	11	15	13	15	15	15	16	14	7

Automatic colors, where greener is relatively better and redder worse

*Splines prediction without extrapolation

Table 8 R^2 from the data until 2022—score 2020–2022

R^2 min	Method	h=1	h=2	h=3	h=4	h=5	h=6	h=7	h=8	h=9	h=10	h=11	h=12
	(a) All cases												
0	Linear	0.18	0.18	0.20	0.20	0.20	0.20	0.20	0.20	0.21	0.19	0.19	0.13
	BOLS	0.18	0.15	0.18	0.18	0.18	0.18	0.18	0.17	0.19	0.18	0.18	0.11
	Bspline	0.17	0.14	0.16	0.17	0.18	0.17	0.18	0.18	0.20	0.19	0.19	0.13
	TSBoost	0.18	0.17	0.20	0.19	0.20	0.20	0.21	0.21	0.23	0.22	0.22	0.16
	Bspline*	0.17	0.15	0.18	0.19	0.19	0.18	0.19	0.18	0.20	0.19	0.20	0.13
	TSBoost*	0.18	0.17	0.20	0.19	0.20	0.20	0.21	0.20	0.23	0.22	0.22	0.16
	Tree	0.16	0.12	0.14	0.14	0.15	0.15	0.15	0.14	0.16	0.14	0.14	0.09
	N	78	78	78	78	78	78	78	78	78	78	78	78
0.1	Linear	0.29	0.31	0.38	0.38	0.40	0.40	0.42	0.40	0.41	0.39	0.38	0.34
	BOLS	0.29	0.29	0.33	0.32	0.34	0.35	0.38	0.38	0.37	0.34	0.33	0.39
	Bspline	0.33	0.34	0.32	0.32	0.32	0.32	0.31	0.32	0.34	0.32	0.30	0.33
	TSBoost	0.30	0.32	0.37	0.35	0.39	0.37	0.39	0.40	0.38	0.37	0.34	0.34
	Bspline*	0.32	0.33	0.34	0.33	0.33	0.32	0.31	0.32	0.33	0.31	0.30	0.33
	TSBoost*	0.30	0.32	0.39	0.36	0.39	0.36	0.39	0.39	0.38	0.37	0.35	0.34
	Tree	0.31	0.31	0.30	0.29	0.30	0.30	0.26	0.27	0.28	0.28	0.27	0.26
	N	45	37	41	39	39	41	41	40	47	45	50	35
	(b) Splines more accurate on average												
0	Linear	0.18	0.22	0.19	0.20	0.19	0.15	0.17	0.20	0.16	0.17	0.16	0.08
	BOLS	0.19	0.22	0.19	0.21	0.20	0.16	0.16	0.20	0.15	0.17	0.16	0.09
	Bspline	0.21	0.24	0.20	0.26	0.26	0.22	0.23	0.26	0.23	0.25	0.24	0.16
	TSBoost	0.20	0.23	0.21	0.23	0.23	0.19	0.23	0.26	0.23	0.25	0.24	0.17
	Bspline*	0.20	0.24	0.20	0.26	0.25	0.20	0.21	0.24	0.21	0.23	0.22	0.15
	TSBoost*	0.20	0.23	0.20	0.22	0.21	0.18	0.20	0.25	0.21	0.23	0.23	0.16
	Tree	0.19	0.22	0.18	0.21	0.22	0.19	0.19	0.20	0.18	0.19	0.18	0.11
	N	48	32	39	34	32	35	36	37	43	41	43	42
0.1	Linear	0.34	0.40	0.41	0.39	0.39	0.39	0.40	0.47	0.38	0.39	0.36	0.36
	BOLS	0.35	0.38	0.36	0.37	0.36	0.36	0.38	0.43	0.36	0.38	0.36	0.38
	Bspline	0.37	0.40	0.36	0.42	0.41	0.41	0.40	0.41	0.39	0.37	0.36	0.39
	TSBoost	0.36	0.42	0.42	0.41	0.43	0.43	0.44	0.51	0.40	0.42	0.39	0.38
	Bspline*	0.36	0.41	0.38	0.42	0.40	0.39	0.37	0.40	0.35	0.38	0.37	0.39
	TSBoost*	0.36	0.42	0.42	0.41	0.41	0.41	0.41	0.49	0.38	0.41	0.38	0.38
	Tree	0.34	0.36	0.30	0.33	0.34	0.35	0.33	0.33	0.30	0.31	0.29	0.30
	N	23	15	15	16	15	13	13	15	15	16	16	7
	(c) Linear more accurate on average												
0	Linear	0.18	0.15	0.22	0.19	0.21	0.24	0.23	0.20	0.28	0.22	0.23	0.18
	BOLS	0.15	0.10	0.16	0.15	0.18	0.19	0.19	0.15	0.25	0.18	0.19	0.13
	Bspline	0.12	0.07	0.13	0.11	0.12	0.13	0.14	0.10	0.16	0.12	0.14	0.10
	TSBoost	0.15	0.12	0.19	0.15	0.18	0.20	0.20	0.16	0.24	0.18	0.20	0.15
	Bspline*	0.13	0.08	0.16	0.13	0.14	0.16	0.16	0.12	0.20	0.15	0.17	0.11
	TSBoost*	0.16	0.12	0.20	0.16	0.19	0.22	0.21	0.16	0.25	0.20	0.21	0.15
	Tree	0.10	0.04	0.10	0.08	0.10	0.12	0.12	0.08	0.14	0.09	0.10	0.07
	N	30	46	39	44	46	43	42	41	35	37	35	36
0.1	Linear	0.34	0.42	0.42	0.41	0.42	0.48	0.48	0.45	0.53	0.45	0.45	0.64
	BOLS	0.30	0.32	0.36	0.35	0.37	0.42	0.41	0.41	0.48	0.39	0.39	0.51
	Bspline	0.27	0.29	0.34	0.29	0.28	0.37	0.34	0.35	0.37	0.31	0.32	0.45
	TSBoost	0.32	0.39	0.41	0.37	0.38	0.47	0.45	0.42	0.50	0.41	0.42	0.62
	Bspline*	0.26	0.29	0.37	0.31	0.31	0.37	0.35	0.35	0.41	0.33	0.34	0.46
	TSBoost*	0.32	0.39	0.42	0.37	0.39	0.47	0.45	0.42	0.50	0.42	0.42	0.62
	Tree	0.23	0.20	0.25	0.21	0.23	0.30	0.26	0.25	0.30	0.21	0.25	0.32
	N	12	8	13	14	19	14	16	11	15	14	14	7

Automatic colors, where greener is relatively better and redder worse

*Splines prediction without extrapolation

Author Contributions All authors contributed to conceptualization, writing-review and editing; methodology and supervision were performed by HdaST; software, validation, data curation and writing were performed by GL. All authors have read and agreed to the submitted version of the manuscript.

Declarations

References

Akaike, H. (1973). Maximum likelihood identification of Gaussian autoregressive moving average models. *Biometrika, 60*(2), 255–265. https://doi.org/10.1093/biomet/60.2.255

Bai, J., & Ng, S. (2009). Boosting diffusion indices. *Journal of Applied Econometrics, 24*(4), 607–629. https://doi.org/10.1002/jae.1063

Bakhitov, E., & Singh, A. (2022). Causal gradient boosting: Boosted instrumental variable regression. In: *Proceedings of the 23rd ACM Conference on Economics and Computation,* pp. 604–605, EC '22, Association for Computing Machinery, New York, NY, USA (2022), ISBN 978-1-4503-9150-4, https://doi.org/10.1145/3490486.3538251.

Barbosa, R. B., Ferreira, R. T., & d Silva, T. M. (2020). Previsão de variáveis macroeconômicas brasileiras usando modelos de séries temporais de alta dimensão. *Estudos Econômicos (São Paulo), 50*(1), 67–98. https://doi.org/10.1590/0101-41615013rrt

Bredahl Kock, A., & Teräsvirta, T. (2016). Forecasting macroeconomic variables using neural network models and three automated model selection techniques. *Econometric Reviews, 35*(8–10), 1753–1779. https://doi.org/10.1080/07474938.2015.1035163

Breiman, L. (1998). Arcing classifier (with discussion and a rejoinder by the author). *The Annals of Statistics, 26*(3), 801–849. https://doi.org/10.1214/aos/1024691079

Breiman, L. (1999). Prediction games and arcing algorithms. *Neural Computation, 11*(7), 1493–1517. https://doi.org/10.1162/089976699300016106

Buchen, T., & Wohlrabe, K. (2011). Forecasting with many predictors: Is boosting a viable alternative? *Economics Letters, 113*(1), 16–18. https://doi.org/10.1016/j.econlet.2011.05.040

Bühlmann, P. (2006). Boosting for high-dimensional linear models. *Annals of Statistics, 34*(2), 559–583. https://doi.org/10.1214/009053606000000092

Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science, 22*(4), 477–505. https://doi.org/10.1214/07-STS242

Bühlmann, P., & Yu, B. (2003). Boosting with the L2 loss: Regression and classification. *Journal of the American Statistical Association, 98*(462), 324–339. https://doi.org/10.1198/016214503000125

Cepni, O., Guney, I. E., & Swanson, N. R. (2020). Forecasting and nowcasting emerging market GDP growth rates: The role of latent global economic policy uncertainty and macroeconomic data surprise factors. *Journal of Forecasting, 39*(1), 18–36. https://doi.org/10.1002/for.2602

Chu, B., & Qureshi, S. (2022). Comparing out-of-sample performance of machine learning methods to forecast U.S. GDP growth. *Computational Economics.* https://doi.org/10.1007/s10614-022-10312-z

Contreras-Reyes, J. E., Georg M. G., & Palma, W. (2013). afmtools: Estimation. In: *Diagnostic and forecasting functions for ARFIMA models (Feb 2013),* http://www2.uaem.mx/r-mirror/web/packages/afmtools/index.html.

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association, 74*(366a), 427–431. https://doi.org/10.1080/01621459.1979.10482531

Eilers, P. H., & Marx, B. D. (2010). Splines, knots, and penalties. *Wiley Interdisciplinary Reviews: Computational Statistics, 2*(6), 637–653. https://doi.org/10.1002/wics.125

Eilers, P. H. C., & Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science, 11*(2), 89–121. https://doi.org/10.1214/ss/1038425655

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation, 121*(2), 256–285. https://doi.org/10.1006/inco.1995.1136

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In: *Proceedings of the 13th international conference on machine learning, 13* (pp. 148–156).

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Fuller, W. A. (2009). Introduction to statistical time series. Wiley, ISBN 978-0-470-31775-4, google-Books-ID: tI6j47m4tVwC.

Giacomini, R., & White, H. (2006). Tests of conditional predictive ability. *Econometrica, 74*(6), 1545–1578. https://doi.org/10.1111/j.1468-0262.2006.00718.x

Gomes, L. E. (2022). An R package for Ipeadata API database. https://github.com/gomesleduardo/ipeadatar, original-date: 2019-01-22T13:38:38Z.

Green, P. J., & Silverman, B. W. (1993). *Nonparametric regression and generalized linear models: A roughness penalty approach*. Chapman and Hall/CRC.

Hamzaçebi, C., Akay, D., & Kutay, F. (2009). Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. *Expert Systems with Applications, 36*(2 part 2), 3839–3844. https://doi.org/10.1016/j.eswa.2008.02.042

Hastie, T. (2007). Comment: Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science, 22*(4), 513–515. https://doi.org/10.1214/07-STS242A

Hothorn, T., Buehlmann, P., Kneib, T., & Schmid, M. (2011). mboost: Model-Based Boosting. R package version 2.0-12.

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics, 15*(3), 651–674. https://doi.org/10.1198/106186006X133933

Ji, Y., Hao, J., Reyhani, N., & Lendasse, A. (2005). Direct and recursive prediction of time series using mutual information selection. In J. Cabestany, A. Prieto, & F. Sandoval (Eds.), *Computational intelligence and bioinspired systems* (pp. 1010–1017). Springer. https://doi.org/10.1007/11494669_124

Kauppi, H., & Virtanen, T. (2021). Boosting nonlinear predictability of macroeconomic time series. *International Journal of Forecasting, 37*(1), 151–170. https://doi.org/10.1016/j.ijforecast.2020.03.008

Kim, H. H., & Swanson, N. R. (2018). Mining big data using parsimonious factor, machine learning, variable selection and shrinkage methods. *International Journal of Forecasting, 34*(2), 339–354. https://doi.org/10.1016/j.ijforecast.2016.02.012

Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics, 54*(1), 159–178. https://doi.org/10.1016/0304-4076(92)90104-Y

Lehmann, R., & Wohlrabe, K. (2015). Forecasting GDP at the regional level with many predictors. *German Economic Review, 16*(2), 226–254. https://doi.org/10.1111/geer.12042

Lehmann, R., & Wohlrabe, K. (2016). Looking into the black box of boosting: The case of Germany. *Applied Economics Letters, 23*(17), 1229–1233. https://doi.org/10.1080/13504851.2016.1148246

Lehmann, R., & Wohlrabe, K. (2017). Boosting and regional economic forecasting: The case of Germany. *Letters in Spatial and Resource Sciences, 10*(2), 161–175. https://doi.org/10.1007/s12076-016-0179-1

Lindenmeyer, G., Skorin, P. P., & Torrent, H. (2021). Using boosting for forecasting electric energy consumption during a recession: A case study for the Brazilian state Rio Grande do Sul. *Letters in Spatial and Resource Sciences, 14*(2), 111–128. https://doi.org/10.1007/s12076-021-00268-3

Marcellino, M., Stock, J. H., & Watson, M. W. (2003). Macroeconomic forecasting in the Euro area: Country specific versus area-wide information. *European Economic Review, 47*(1), 1–18. https://doi.org/10.1016/S0014-2921(02)00206-4

Marcellino, M., Stock, J. H., & Watson, M. W. (2006). A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series. *Journal of Econometrics, 135*(1), 499–526. https://doi.org/10.1016/j.jeconom.2005.07.020

McCracken, M. W., & Ng, S. (2016). FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics, 34*(4), 574–589. https://doi.org/10.1080/07350015.2015.1086655

Medeiros, M. C., Vasconcelos, G. F., Veiga, A., & Zilberman, E. (2019). Forecasting inflation in a data-rich environment: The benefits of machine learning methods. *Journal of Business and Economic Statistics, 39*(1), 98–119. https://doi.org/10.1080/07350015.2019.1637745

Nonnenberg, M. J. B. (2015). Novos Cálculos da Taxa Efetiva Real de Câmbio para o Brasil

Park, B., Lee, Y., & Ha, S. (2009). L2 boosting in kernel regression. *Bernoulli, 15*(3), 599–613.

Pfaff, B., Zivot, E., & Stigler, M. (2016) URCA: Unit root and cointegration tests for time series data (Sep 2016). https://CRAN.R-project.org/package=urca.

R Core Team: R (2019). A language and environment for statistical computing. R Foundation for Statistical Computing.

Ribeiro, M. H. D. M., & dos Santos Coelho, L. (2020). Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. *Applied Soft Computing, 86*, 105837. https://doi.org/10.1016/j.asoc.2019.105837

Robinzonov, N., Tutz, G., & Hothorn, T. (2012). Boosting techniques for nonlinear time series models. *AStA Advances in Statistical Analysis, 96*(1), 99–122. https://doi.org/10.1007/s10182-011-0163-4

Rstudio, T. (2020). RStudio: Integrated development for R. Rstudio Team, PBC, Boston, MA. http://www.rstudio.com/https://doi.org/10.1145/3132847.3132886.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning, 5*(2), 197–227. https://doi.org/10.1007/bf00116037

Schmid, M., & Hothorn, T. (2008). Boosting additive models using component-wise P-Splines. *Computational Statistics and Data Analysis, 53*(2), 298–311. https://doi.org/10.1016/j.csda.2008.09.009

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics, 6*(2), 461–464.

Stock, J. H., & Watson, M. W. (1998). A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series. Working Paper 6607, National Bureau of Economic Research (Jun 1998). https://doi.org/10.3386/w6607, series: Working Paper Series.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological), 36*(2), 111–133. https://doi.org/10.1111/j.2517-6161.1974.tb00994.x

Taieb, S. B., & Hyndman, R. (2014). Boosting multi-step autoregressive forecasts. In: *Proceedings of the 31st international conference on machine learning,* (pp. 109–117). PMLR (Jan 2014), ISSN: 1938-7228.

Trapletti, A., & Hornik, K. (2021) tseries: Time series analysis and computational finance (Nov 2021). https://CRAN.R-project.org/package=tseries.

Wohlrabe, K., & Buchen, T. (2014). Assessing the macroeconomic forecasting performance of boosting: Evidence for the United States, the Euro area and Germany. *Journal of Forecasting, 33*(4), 231–242. https://doi.org/10.1002/for.2293

Yoon, J. (2021). Forecasting of real GDP growth using machine learning models: Gradient boosting and random forest approach. *Computational Economics, 57*(1), 247–265. https://doi.org/10.1007/s10614-020-10054-w

Zeng, J. (2017). Forecasting aggregates with disaggregate variables: Does boosting help to select the most relevant predictors? *Journal of Forecasting, 36*(1), 74–90. https://doi.org/10.1002/for.2415

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.