BANCO CENTRAL DE COSTA RICA

DOCUMENTOS DE TRABAJO
N.° 02 | 2020

Forecasting Costa Rican Inflation with Machine Learning Methods

Adolfo Rodríguez Vargas

Frontpage photograph: “Presentes”, bronze sculptures set, 1983, by Costa Rican artist Fernando Calvo Sánchez. From the collection of the Central Bank of Costa Rica.

Forecasting Costa Rican Inflation with Machine Learning Methods

Adolfo Rodríguez Vargas ‡

The ideas expressed in this paper are those of the author and not necessarily represent the view of the Central Bank of Costa Rica.

Abstract

We present a first assessment of the predictive ability of machine learning methods for inflation forecasting in Costa Rica. We compute forecasts using two variants of K-Nearest Neighbours, random forests, extreme gradient boosting and a long short-term memory (LSTM) network. We evaluate their properties according to criteria from the optimal forecast literature, and we compare their performance with that of an average of univariate inflation forecasts currently used by the Central Bank of Costa Rica. We find that the best-performing forecasts are those of LSTM, univariate KNN and in lesser extent random forests. Furthermore, a combination performs better than the individual forecasts included in it and the average of the univariate forecasts. This combination is unbiased, its forecast errors show appropriate properties, and it improves the forecast accuracy at all horizons, both for the level of inflation and for the direction of its changes.

Key words: inflation, forecasting, machine learning, forecast evaluation.
JEL codes: E31, C45, C49, C53.

‡ Department of Economic Research, Economic Division, BCCR. rodriguezva@bccr.fi.cr.

Pronóstico de inflación en Costa Rica mediante métodos de aprendizaje automático

Adolfo Rodríguez Vargas[‡]

Resumen

Se presenta una primera evaluación de la capacidad de métodos de aprendizaje automático para predecir la inflación en Costa Rica. Se calculan pronósticos mediante dos variantes de K-Nearest Neighbours (KNN), bosques aleatorios, extreme gradient boosting y un modelo de tipo long short-term memory (LSTM). Sus propiedades se evalúan de acuerdo con criterios sugeridos en la literatura sobre pronósticos óptimos, se compara su desempeño con el del promedio de los pronósticos univariados actualmente en uso en el Banco Central de Costa Rica. Los resultados son promisorios. Se encontró que los pronósticos con el mejor desempeño son los resultantes de aplicar LSTM, KNN univariado y en menor medida bosques aleatorios. Además, una combinación de los pronósticos obtenidos mediante estos métodos mejora el desempeño con respecto a los pronósticos individuales a todos los horizontes, y también supera en desempeño al promedio de los pronósticos univariados. La combinación resulta insesgada, sus errores de pronóstico no muestran patrones de correlación indeseables, y mejora la capacidad de pronóstico a todos los horizontes, tanto para el nivel de la inflación como para la dirección de sus cambios.

Palabras clave: inflación, pronóstico, aprendizaje automático, evaluación.
Clasificación JEL: E31, C45, C49, C53.

[‡] Department of Economic Research, Economic Division, BCCR. rodriguezva@bccr.fi.cr.

Contents

Forecasting Costa Rican Inflation with Machine Learning Methods

1. Introduction

Article 2 of the Organic Law of the Central Bank of Costa Rica (BCCR) states as its fundamental goal to the ensure internal stability of the currency, which has been understood as keeping a low and stable inflation. As part of an ongoing effort to fulfill that goal, in 2005 the BCCR begun a process to migrate to an inflation targeting scheme that finished in 2018 with the formal adoption of such regime.

One of the main features of this policy scheme is its prospective nature: since monetary policy operates with a lag, policy decisions are made taking into account the expected trajectory of inflation and other relevant economic variables during the policy horizon. Hence, for an inflation-targeting central bank it is crucial to have adequate forecasts for the main economic variables in order to make adequate policy decisions, and among them, inflation forecasts are of particular importance. Currently, the BCCR forecasts inflation with an ensemble of univariate models (Fuentes and Rodríguez, 2016), an ensemble of Bayesian models (Chavarría and Chaverri, 2015) and a semi-structural econometric model (Muñoz and Tenorio, 2008).

All of these forecast come from relatively traditional econometric methods. However, machine learning methods have become increasingly popular as a forecasting tool due to growing availability of big databases and computing power, and to greater access to specialized software. Their use is widespread in classification problems where the variable of interest is discrete, like prediction of delinquency in loans or consumer purchasing decisions, where they have often outperformed more traditional methods. Most applications, hence, use cross-sectional data in classification problems. However, machine learning methods can also be adapted for prediction of continuous time-series data, like inflation or GDP growth.

Precisely, the goal of this study is to perform a first evaluation of the performance of machine learning methods in forecasting Costa Rican inflation. The idea is to verify if forecasts from these methods comply with properties of optimal forecasts under quadratic loss, and if their performance is superior to that of their univariate counterparts currently in use at the BCCR.

The rest of this document includes a very brief review of the methods applied, in section 2; the methodological details on data, implementation of the methods and evaluation criteria, in section 3; and the discussion of the main results in section 4, followed by a succinct concluding section.

2. Machine learning forecasting methods applied in the study

In this section we present a brief review of machine learning methods applied in this study, beginning with a general characterization of the machine learning paradigm as opposed to traditional econometrics. We do not attempt to present neither a formal nor exhaustive review of these methods, for that we suggest to go to the sources cited.

2.1. Machine learning and econometrics

A general definition of machine learning (ML) is the development and application of algorithms to allow machines to improve their performance in a particular task when presented with new information. There is no single unified framework for estimation and analysis, and generally the goal of these algorithms is to identify patterns in the data that can be used to predict values of random variables[1].

Breiman (2001b) states that the main differences of machine learning from traditional statistics and econometrics are the lesser emphasis that the former gives to traditional statistic inference, the lesser emphasis in the idea of a "true model" that generates the data, and greater emphasis in prediction and method optimization. Athey and Imbens (2019) present a useful review of the differences in several dimensions, which is the main source of the discussion below.

Goals

In the traditional econometric approach, an estimation object is defined that is a function of a joint data density, and which generally is a parameter in a statistical model describing the distribution of a set of variables in terms of a set of parameters. Those parameters are estimated with data from a random sample of the population, using objective functions like the sum of squared residuals or the likelihood function. The emphasis is usually on the quality of these estimators (unbiasedness, consistency, efficiency).

On the other hand, the primary goal of the ML approach is to obtain algorithms that allow the prediction of values of a variable from a set of information of other variables. A loss function is defined based on the comparison of the predicted values with the true values, and the parameters of the model are set so that they have desirable properties in terms of that loss function.

[1] The terms machine learning (ML), artificial intelligence (AI) and data science (DS) are often confused. AI is the most general concept: it refers to "…the science and engineering of making intelligent machines…" (McCarthy, 2007). ML can be characterized as a central field in AI, whose main goal is the automatic acquisition of knowledge from data. Data science is an inter-disciplinary field whose aim is to analyze information, obtain information from it and make predictions, for which ML techniques are generally used.

Validation and cross-validation

In traditional econometrics there is no particular emphasis in model validation, whose form is generally supposed as externally given, for example by economic theory. The discussion on model selection is usually centered on hypothesis testing for parameters of a particular model, always under the assumption that there is an underlying true model.

On the contrary, out-of-sample cross-validation in ML aims to improve the predictive ability rather than estimating a causal or structural model. It is a non-parametric approach which is more computationally costly, based in the comparison with out-of-sample data, rather than in-sample goodness-of-fit measures.

The two most widely used variants of cross-validation are *K-fold* cross-validation and *leave-one-out* cross-validation. In the former, data are randomly split into k groups of roughly the same size. On the first iteration, the first group (k=1) is taken as the comparison group for the predicted values, obtained from the estimated model with data from the other K-1 groups, and a first prediction error measure is computed (M_1). On the second iteration, the second group is taken as the comparison group and a second prediction error measure is computed (M_2). The process is repeated K times and the final prediction error measure is the mean of the K error measures computed in the iterations. Leave-one-out cross-validation is simply validation with K iterations where K is equal to the number of available observations (one observation is left out of the estimation sample each time).

Overfitting and regularization

There is a greater emphasis in ML on avoiding overfitting than in traditional statistics and econometrics, so that the desired model should be flexible and fit the data well, but not at a cost in terms of predictive ability. To deal with overfitting, in ML the concept of regularization takes center stage. The idea is that for model selection, starting from a set of models that differ in complexity (for example in the number of parameters), rather than optimizing directly an objective function, a term is added to it penalizing model complexity. Obviously this practice is familiar in traditional econometrics due to the use of information criteria like the Akaike, Schwarz or Hannan-Quinn. The difference is that in ML regularization is guided by the data in a greater way than in the traditional approach, and it is determined explicitly by out-of-sample predictive performance.

Variable selection

In econometrics the number of explanatory variables is generally set taking into account relevant economic theory under the idea of an underlying true model. In ML, on the other hand, the process is more data-driven. It is common that in ML problems the number of variables be very large, even when a priori it could be

considered that many of those variables will not improve prediction. Given that it is not know which of those variables would indeed be useful, a frequently used guide is the *sparsity principle*. In a statistical model that comply with this principle there is a relatively small number of parameters of importance. Thus, this assumption is used in ML techniques to extract the underlying signal from a potentially large dataset. See Hartie, Tibshirani and Wainright (2015) for a more general discussion.

Scalability

ML methods give great importance to computational issues. It is of particular interest that methods are scalable, that is, that they can be applied with relative ease when the size of the dataset increases. Thus, it is possible that with the ML approach methods that might have desirable properties from the point of view of traditional statistics are discarded because they do not "scale well" to big datasets.

Terminology

An additional, although minor, difference, is the terminology often used. Some frequently used terms in ML have an exact counterpart in statistics and econometrics. Some of the most frequent are:

- *Training*, instead of estimation.
- *Features*, instead of regressors, covarates or explanatory variables.
- *Weights*, instead of coefficients.
- *Example or instance,* instead of data point.

An important distinction is between supervised and unsupervised learning. Suupervised learning is traditionally associated with regression analysis, performed with a dataset for a target variable *y* and regressors *x*. When *y* is continuous, the supervised learning problem is a regression problem, and when it is categorical is it a classification one. When only data on regressors is used, but no response variable *y* is defined, the approach is of the unsupervised nature, of which cluster analysis is an example.

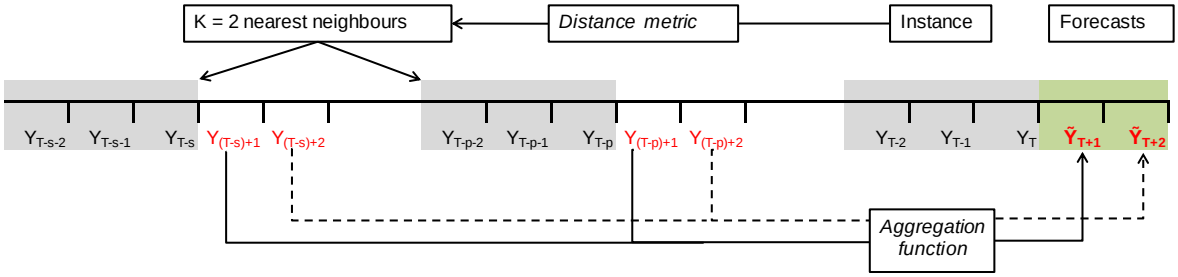
In this study we will used the traditional terminology, unless it is strictly necessary to use the ML terminology.

2.2. K-Nearest neighbours

The *K-Nearest Neighbours* algorithm (KNN) is a classification and regression algorithm that basically searches a set of periods similar to the most recent history of the data, and makes the forecast based on the subsequent evolution of the variable of interest. In the case of regression, the algorithm starts with a vector that includes the variable of interest and the explanatory variables, and using a distance

metric (Euclidean, Mahalanobis, etc) finds the K most similar periods (i.e. the nearest neighbours). The value of the target variable is predicted by aggregating the values after the periods where the nearest neighbours are located, typically using an average (simple or weighting by distance). This is one of the simplest ML algorithms, of the type "lazy learning", where the forecast computation is done at the moment of the classification. The method can be used in its univariate version, in which the explanatory variables are lags of the target variable, or multivariate, when the vector of variables includes additional explanatory variables. Figure 1 illustrates the method for the univariate case with 2 lags of the target variable and 2 nearest neighbours. It can be seen that the forecasts for horizons 1 and 2 are aggregations of the observations for Y located in the following 1 or 2 periods after each of the nearest neighbours, and thus for forecasts at any horizon, with any number of neighbours K.

Figure 1. Univariate KNN algorithm



Source: Own elaboration.

The principle of the multivariate KNN algorithm is the same, only considering the lags of the other variables besides those of Y.

KNN regression originates in Mack (1981), and its use for forecasting time series is popularized by Yakowitz and Karlsson (1987), and Yakowitz (1987), who shows that this non parametric regression converges in squared mean at the optimal rate of Stone. Diebold and Nason (1990), Mizrach (1992), Lisi and Medio (1997), Lisi and Schiavo (1999), Meade (2002) and Fernandez-Rodriguez et al (1999) forecast exchange rates using KNN; Barkoulas, Baum and Chakraborty (1996) and Nowman and Saltoglu (2003) forecast interest rates; and Agnon et al (1999) apply this type of methods to the projection of commodity prices. More recently, Nikolopoulos et al (2015) use KNN to forecast sporadic demand in a supply chain setting.

2.3. Random forests

The random forests (RF) algorithm was proposed by Breiman (2001a). This non parametric classification system is also applicable to regression, and consists in a combination of predictive trees (*classification and regression trees*, CART) in which every tree depends of a random vector sampled independently from the distribution

of all trees. Basically, the algorithm builds many decision trees in the estimation sample and outputs the mode of the classes (in classification problems) or the mean of the predictions (for regression). Random forests is based in the bagging principle applied to regression and classification trees. The review below follows closely Cameron (2017).

The idea in a regression tree is sequentially split a set of regressors *X* in rectangular regions so that the sum of squared residuals (SSR) be reduced. The determination both of the regressor *j* to split and the splitting point *s* is done in the following way:

- For each regressor *j* and splitting point *s*, the following sets are defined

$$R1(j,s)= \{X / X_j < s\} \text{ and } R2(j,s)= \{X / X_j \geq s\}$$

- The values of *j* and *s* that minimize

$$\sum_{i:x_i \in \mathrm{R1(j,s)}} (y_i - \bar{y}_{R1})^2 + \sum_{i:x_i \in \mathrm{R2(j,s)}} (y_i - \bar{y}_{R2})^2$$

  are found, where $\bar{y}_{R1}$ is the average of the observations of *y* in region 1 and $\bar{y}_{R2}$ that of region 2.

- Once this first split is found, the preceding criterion is used to further split R1 and R2.

- The splitting of regions stops when a predetermined criterion is met, called the terminal node size. For example, the algorithm stops when there is less than *m* observations in each region.

This method might result in overfitting because each split is done so as to optimize that particular step rather than deciding in a prospective way, choosing the split that would lead to the best future tree.

The decision trees method can result in high variance if the trees in each split are very different. One way to deal with this is the *bagging* method*:* to average over the results of the trees for many possible bootstrapped samples. For each sample, an error measure is computed using the observations out of the bootstrap sample (*out-of-bag error*). Since the interpretation of the trees across samples is complicated, in order to determine the importance of each regressor it is common to compute the average for all of the trees of the amounts in which the SSR falls due to splits in that regressor. A high value indicates an important predictor.

Now, forecasts from bagging will be correlated because if a regressor is important it is likely to appear close to the beginning of the tree in each bootstrap sample. The random forest method deals with this problem using only a subset of $m < p$ predictors in each division within each bootstrap sample, with $m \cong \sqrt{p}$. This method reduces

correlation and variance, but results in a complex model, in which it is usual to evaluate the individual importance of the regressors following the procedure outlined in the previous paragraph.

Applications of random forests have become popular in medicine and other biological sciences, but less so in economics. A notable example is Biau and D'Elia (2010), who apply the method to GDP forecasting for the Euro area from a dataset containing 172 indicators and find that it compares favourably both with autoregressive forecasts and with those of the *Eurozone Economic Outlook*. Furthermore, Bajari et al (2015) include random forests in their ML toolkit for the estimation of demand for groceries, and David (2017) apply them for the projection of growth and cycles in the Euro zone.

2.4. Boosting

Boosting was proposed by Freund and Schapire (1995, 1996), and introduced to regression problems by Friedman et al (2000) and Friedman (2001). This method aims at improving the forecasting ability of simple ML methods. Boosting methods do not estimate a single model, but begin with a linear regression to which the regressor with the largest contribution to overall fit is iteratively added (this is the boosting), according to in-sample performance, but without adjusting the coefficients already existing.

This study applies the *gradient boosting* variant. While classification and regression trees use an algorithm that usually leads to overfitting, boosting methods use a slower algorithm to generate the sequence of trees. Crucially, boosting does not require bootstrapping, because each tree is generated from information of already existing trees, and is adjusted using a modified version of the original dataset. In particular, in *gradient boosting* for the current model *m*:

- A new tree is obtained for the residuals of *m*.
- Current information is updated as $\hat{f}(x) = \hat{f}^m(x) + \lambda \hat{f}^{res\_m}(x)$, where λ is a penalizing parameter.
- Residuals are updated as $res_i = res^m - \lambda \hat{f}^{res\_m}(x_i)$
- Finally, the *boosted* model is given by

$$\hat{f}(x) = \sum_{m=1}^{B} \lambda \hat{f}^{res\_m}(x_i)$$

Variants of the boosting method have proved useful to forecast with large datasets in a computationally efficient way. Some examples are Wohlrabe and Buchen (2014), who evaluate the performance of these methods in forecasting economic variables for the Euro zone and the USA; Lehmann and Wohlrabe (2016), who use

Germany data to assess the type of indicators usually selected by the method; and Zeng (2017), who successfully uses the method to select disaggregate variables to forecast aggregate variables.

2.5. Long Short-Term Memory Models

Hochreiter and Schmidhuber (1997) introduce Long Short-Term Memory (LTSM) networks, which have been shown to improve precision over traditional neural network models. These models are a type of recurrent neural network, different from traditional neural networks in that they include a feedback loop between past decisions and the current outcome. Thus, their functional architecture allows to solve the vanishing gradient problem in the updating rule, which makes possible to handle longer-run dependencies.

Wang and Raj (2017) offer a succinct presentation of these models, which is summarized below. A recurrent neural network is a type of net with connections of units forming a directed loop, which allows them to work with time-series data. The architecture of LSTM was introduced because traditional recurrent neural networks could not handle long run dependencies (see Bengio et al, 1994). The elements that form an LSTM cell are:

- ***States*.** These are the values used to give the output information. They include:
    - *Input data*: called here *X*.
    - *Hidden state*: called *h*, this is the values of the previous hidden layers, and is the same as in traditional recurrent neural networks.
    - *Input state*: these values are defined as a linear combination of the hidden state and the input data on the current period. It is given by:

$$i_t = \sigma(W_{ix} X_t + W_{ih} h_{t-1})$$

      where W_{ix} is a weighting matrix and W_{ih} is a transition matrix between the hidden states. These matrices can be thought of as filters that determine the importance given to the data and to the previous hidden state.

    - *Internal state*: denoted as *m* these are the values that act as memory for the model.
- ***Gates.*** These are the values used to decide on the flow of information between states. They include:

- *Input gate:* it "decides" whether the input state enters internal state. It is defined as:

$$g_t = \sigma(W_{gi} i_t)$$

- *Forget gate:* this gate decides whether the internal state "forgets" the preceding internal state. It is defined as:

$$f_t = \sigma(W_{fi} i_t)$$

- *Output gate:* it decides whether the internal state passes its value to the output and the internal state of the following period. It is defined as:

$$o_t = \sigma(W_{oi} i_t)$$

- The following two equations show how the gates determine the flow of information of the states:

$$m_t = g_t \odot i_t + f_t \odot m_{t-1}$$
$$h_t = o_t \odot m_t$$

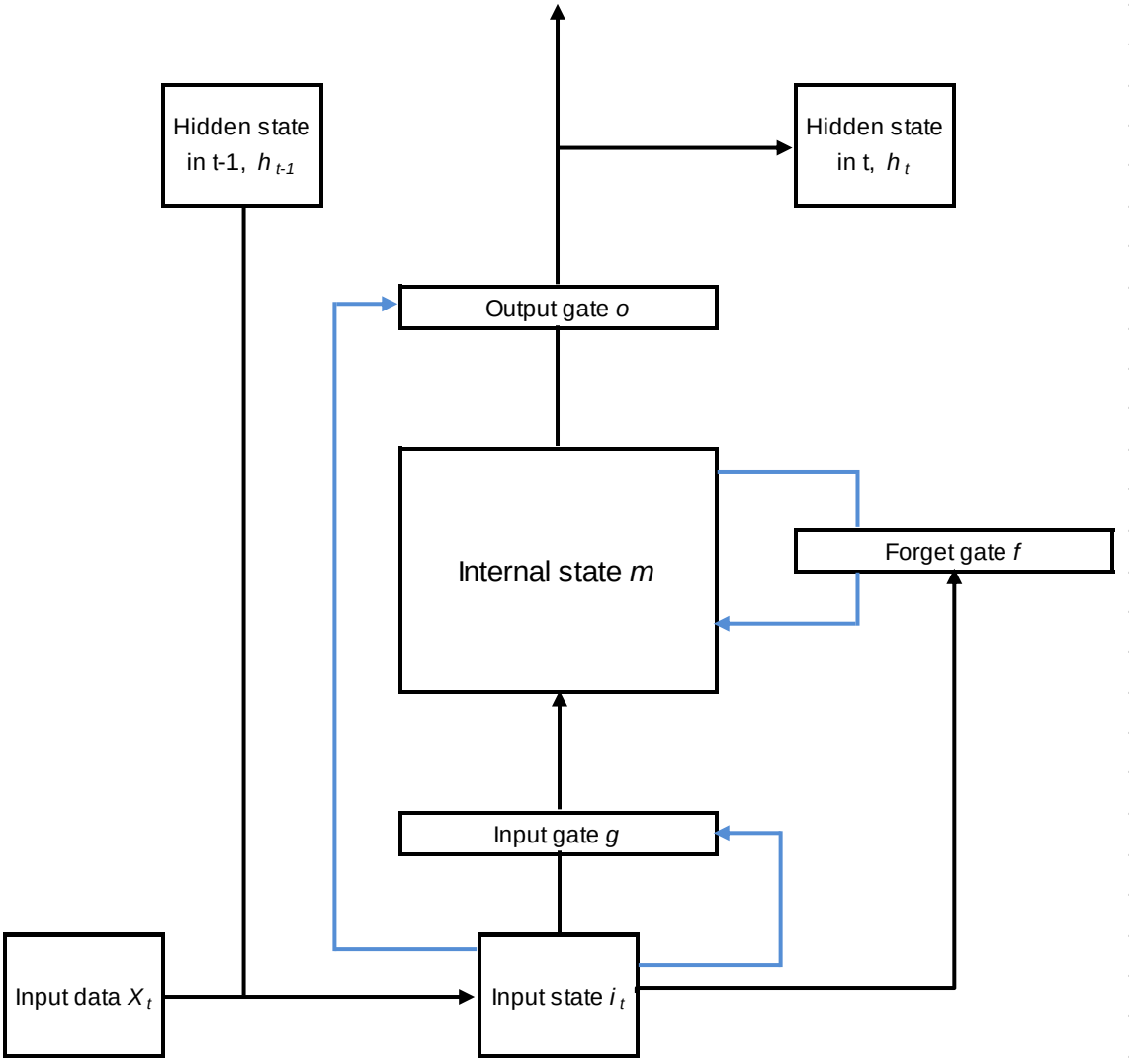
where $\odot$ indicates element-wise multiplication.

In all cases σ s an activation function, which generally is a logistic sigmoid or a hyperbolic tangent (*tanh*). The interaction of these elements within the LSTM cell is presented in Figure 2.

In these models all weights are parameters to be estimated, and hence LSTM can be capable of memorizing long time dependencies or of "forgetting" the past as needed. The errors generated by the weighting matrices W produce errors that return through a *backpropagation* process to adjust those weights so that the error is progressively reduced. This process allows that a fraction of the error be assigned to adjust the weights according to partial derivatives used by a learning rule to that end (the *gradient descent* optimization algorithm)[2]. Precisely, LSTMs solve the vanishing gradient descent problem, which impairs the learning from distant information, by preserving the error that can be propagated through time and through the memory cells used.

[2] The simplest version of gradient descent for optimization follows $\theta_{t+1} = \theta_t - \eta \nabla F(\theta_t)$, where η is the learning rate and $\nabla F(.)$ is the gradient of the function to be optimized.

Figure 2. Architecture of an LSTM cell



Source: Wang and Raj (2017).

The widespread use of neural networks for forecasting of macroeconomic variables started in the 1990s, although the first developments on the topic date from the 1940s (McCullogh and Pitts, 1943; Hebb, 1949). Among the first examples of their application for this type of problems are Swanson and White (1995, 1997), in finance; Tkacz and Hu (1999), to forecast Canada's GDP; Stock and Watson (1998), who find that neural networks perform poorly in comparison to other univariate methods; Refenes and White (1998) and Fernández-Rodríguez et al (2000), also in finance; and Moshiri and Cameron (2000), who forecast inflation. Nakamura (2006) shows that applying early stopping close to local minima improves forecasting ability, something already suggested by González (2000). For Costa Rica, Solera (2005) and Esquivel (2007) apply simple neural network models with several specifications for inflation forecasting. Cook and Hall (2017), forecast employment indicators using

several deep neural network architectures, among them LSTM and show that these improve precision with respect to simpler configurations.

3. Forecasting scheme and evaluation criteria

3.1. Data

The variable to forecast is the interannual variation rate of the Consumer Price Index of Costa Rica (with base June-2015). We used a dataset comprising monthly data for the variables detailed in Table 1. We included real sector variables, monetary and internal and external price variables, financial and exchange rate data, and labour market data. Furthermore, we included 12 lags of each of these variables, as well as a set of seasonal dummies, totaling 258 possible explanatory variables. Most of them correspond to interannual variation rates, except for interest rates and the Monetary Conditions Index. Data covers the period January-2003 to February-2019.

3.2. Forecasting procedure

The first set of 12 forecast is done with data from January 2003 to December 2016, the following with data up to January 2017, and so on, increasing the estimation sample one month at a time. From these multi-horizon forecasts, fixed-horizon series for 1, 3, 6 y 12-month forecasts were extracted. These were the series included in the evaluation. Details of estimation and forecasting for each method are presented in the next sections. Table 14 of the annex presents software details.

Table 1. Variables used in the forecasts

Name	Description	Source
IPC	Interannual variation rate of the Consumer Price Index (IPC), base June 2015	National Institute of Statistics and Censuses (INEC)
EXPINF12	12-month inflation expectations	Central Bank of Costa Rica (BCCR)
TCN	Interannual variation rate of the nominal exchange rate	BCCR
TCR_M	Interannual variation rate of the multilateral Index of Real Exchange Rate	BCCR
IMAETC	Interannual variation rate of the Monthy Index of Economic Activity (IMAE), trend-cycle	BCCR
ICFNIV	Financial Conditions Index	BCCR
CREDPRIVSF	Interannual variation rate of total credit of the national financial system to the private sector, local currency.	BCCR
BASEM	Interannual variation rate of the monetary base	BCCR
M1	Interannual variation rate of M1	BCCR
TPM	Monetary policy rate	BCCR
TBP	Basic rate (Tasa Básica Pasiva)	BCCR
PRIMERATE	Prime Rate	BCCR
PETRO	Interannual variation rate of the oil barrel price, average.	Pink Sheet, World Bank
GRANOS	Interannual variation rate of the Grains index	Pink Sheet, World Bank
INFSOC	Inflation of trade partners (interannual variation)	BCCR
ISMNNIV	Interannual variation rate of the Index of Minimum Nominal Wages	BCCR
ISMRNIV	Interannual variation rate of the Index of Minimum Real Wages	BCCR
RESPIB	Financial result of the Central Government as a share of GDP	BCCR, with data from the Ministry of Finance
DEUDAPIB	Total internal debt as a share of GDP	BCCR, with data from the Ministry of Finance

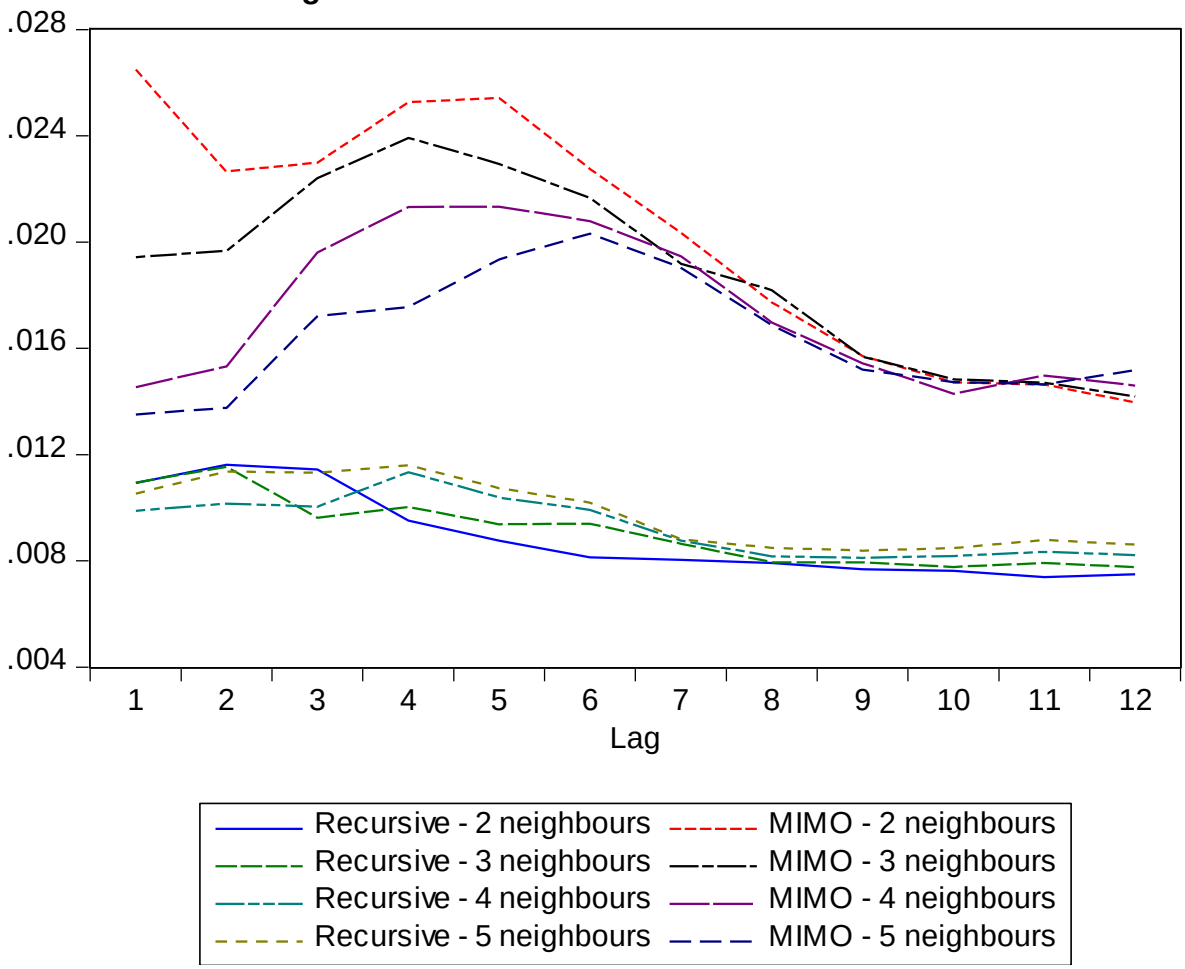
3.2.1. Univariate KNN

The number of nearest neighbours to search, the number of lags of the target variable and the method to perform the multi-horizon forecast were chosen to minimize the average SSR of the forecast series. Forecasts were computed considering from 2 to 5 nearest neighbours, from 1 to 12 lags of the target variable and both a Multiple Input Multiple Output (MIMO) forecasting strategy and a recursive forecasting strategy. With the MIMO strategy, for each nearest neighbour a vector of target values is defined, with size equal to that of the number of periods to forecast. The forecast is then performed by searching the values of inflation most similar to the last 12 values of the time series, and by aggregating the target vectors that follow each nearest neighbour. The recursive strategy is the usual procedure in autoregressive models, in which past predictions of the target variable are used when there is no more historical data for it. In all cases Euclidean distance was used, and the aggregation function to generate the forecast was the arithmetic mean.

Figure 3 shows the average of the roots of the mean squared errors (RMSE) for each estimated model. It is clear that the recursive forecasts are more precise in all cases, with a more homogeneous performance between models with different

number of neighbours and different number of lags. The forecasts with the lowest RMSE are those computed with a recursive strategy, 2 neighbours and 11 lags of the target variable. These will be the forecasts included in the evaluation.

Figure 3. RMSE for univariate KNN models



Source: Own elaboration

3.2.2. KNN with explanatory variables

We decided to include 11 lags of inflation, as in the univariate case. It would be impractical to consider all 258 exogenous variables in the search for nearest neighbours, hence a selection process was implemented. We began by considering the variables included in the inflation equation of the quarterly model of the BCCR (Muñoz and Tenorio, 2008), along with monetary and external prices variables, and we computed correlations of inflation with lags and leads of each one of them. For each variable, we selected the lag that resulted in the highest correlation with inflation. Finally, we decided to include the contemporaneous value of 12-month ahead inflation expectations and of the interannual variation rate of nominal

exchange rate, the fourth lag of the interannual variation rate of the monetary base, the ninth lag of the policy rate, the eight lag of interannual variation rate of the oil Price and the sixth lag of the interannual variation rate of the price of grains. The number of neighbours was determined as in the univariate case: by computing the average of the RMSEs of each set of forecasts for 2 to 5 neighbours. The lowest RMSE was reached with 5 neighbours. We used Euclidean distance and aggregation by inverse-distance weighted average[3].

3.2.3. Random forests

Implementation of random forests requires to set the number of trees to generate, the minimum number of terminal observations admissible in each node and the number of variables to include in each split. It is generally accepted that the implementation of this method requires relatively little calibration of these parameters to obtain acceptable predictions (Segal, 2004; Boelaert y Ollion, 2018; Athey e Imbens, 2019). The parameter that is most frequently optimally-calibrated is the number of candidate variables entering each split. Regarding the number of trees to generate, Breiman (2001a) proved convergence of the mean squared generalization error in random forest regression, which has been taken as an argument to use a high number of trees. Recently, research by Probst y Boulesteix (2018) on this issue showed that a high but computationally feasible number of trees can be recommended, as long as classic mean loss error measures are used.

Taking into account these considerations, for the forecasting exercise the parameters were set in the following way:

- *Number of trees*: it was set at 100 for all forecasting exercises.
- *Size of terminal nodes*: 5 observations (usual value in most applications).
- *Number of variables in each tree:* this parameter was calibrated for each forecasting exercise, by finding the number of variables that minimized *out-of-bag* error.

3.2.4. Extreme gradient boosting

The version of boosting applied in this study is extreme gradient boosting, developed by Chen and Guestrin (2016) as a regularized adaptation of gradient boosting, whose end is to control overfitting. The difference with other implementations of the algorithm is technical: it is an efficient and, crucially, *scalable* application of the gradient boosting approach, which is optimized to receive *sparse data*, and allows

[3] With this method, the forecast is given by $\hat{P} = \sum_{i=1}^{K} w_i(x) P_i \Big/ \sum_{i=1}^{K} w_i(x)$, where $w_i(x)$ is the inverse of the distance of each neighbour *i*.

for parallel computing, which makes it faster than other versions of the algorithm. For this study we used all variables described in section 3.1, with the RMSE as evaluation metric and cyclical selection of variables (deterministic selection, cyclically considering one variable at a time). The parameters to calibrate are:

- *nrounds*: it controls the maximum number of iterations needed for the underlying gradient descent algorithm to converge.

- *lambda*: this parameter aims to avoid overfitting, by controlling the L2 regularization process on the weights (variable coefficients), in a manner equivalent to a Ridge regression.

- *alpha*: this parameter also aims to limit overfitting, by controlling the L1 regularization process, similar to a Lasso regression.

3.2.5. LSTM model

For estimation and forecasting we used the *Keras* package*,* an application programming interface (API) for R, launched in mid-2017. This package runs on *Tensorflow* in Python, which serves as the "backend" engine (data access).

The metric for evaluation during training was precision as measured by the MSE. We used the Adam optimizer by Kingma and Ba (2017), with the default learning rate of 0.02, default learning rate decay (1e-6), and hyperbolic tangent as the activation function.

3.2.6. Evaluation criteria

In this section we present a summary of the evaluation criteria applied in this study. These are derived from a long literature that includes Mincer and Zarnowitz (1969), Granger and Newbold (1973), Stekler (1991), and Diebold and López (1996), who present four properties that must be met by optimal forecasts under quadratic loss. West (2006) presents a useful review of the relevant literature. Presentation of the criteria follows closely Fuentes and Rodríguez (2016).

a- Unbiasedness

If a forecast does not systematically under of overestimate the true value of the target variable, forecast errors should have zero mean. Mincer and Zarnowitz (1969) propose an unbiasedness test based on the regression in levels given by:

$$y_{t+s} = \alpha + \beta \hat{y}_{t+s} + \varepsilon_{t+s}$$

where $\hat{y}_{t+s}$ is the forecast with horizon *s* for the target variable, and y_{t+s} is its real value. The null hypothesis of unbiasedness is α =0 and β=1. In this study we use Wald tests.

b- Error correlation

Diebold and López (1996) show that forecast errors e_{t+h} for an optimal forecast $\hat{y}_{t+h}$ are White noise for horizon h=1 and at most follow an MA(h-1) process for h>1. They recommend the modified Wilcoxon signed-rank test proposed by Dufour (1981) for autocorrelation in the case h =1, and for h >1 the Cumby and Huizinga (1992) test, whose null hypothesis is that errors follow an MA(q) process, with 0 ≤ q ≤ h-1, and whose alternative hypothesis is that q > h.

c- Precision

We present RMSE and Theil inequality coefficients, and the statistical significance of differences in forecasting ability under quadratic loss for pairs of models are tested using the modified Diebold-Mariano test by Harvey, Leybourne and Newbold (1997)[4]. Additionally, we compute the percentage of changes in direction of the inflation that were correctly predicted.

d- Forecast error variances

Diebold and López (1996) state that optimal forecasts under quadratic loss should have errors whose variance is non-decreasing as the forecast horizon increases. This basically reflects that the uncertainty of the forecast should decrease, not increase, when more information is available (as is the case for shorter horizons). To verify this property we conduct F tests for the difference of variances.

e- Forecast encompassing

Tests of forecast encompassing are used to determine whether a forecast contains all of the information used in one or several alternative forecasts. This is useful, for example, to decide if there is an information gain in combining several of them. We use the Chong and Hendry (1986) test for encompassing of point forecasts under quadratic loss function. From the regression:

---

[4] The statistics are $DM = \frac{\bar{d}}{s_d}$ y $HLN = \left(\frac{T+1-2k+k(k-1)/T}{T} \right)^{1/2} DM$, where $\bar{d}$ is the mean of the squared difference of forecast errors, s_d is a consistent estimate of its standard deviation, k is the forecast horizon and T the number of observations.

$$y_{t+s} = \beta_0 + \beta_1 \hat{y}_{t+s,1} + \beta_2 \hat{y}_{t+s,2} + \varepsilon_{t+s}$$

we conclude that forecast 1 encompasses forecast 2 if the join null $(\beta_0 \quad \beta_1 \quad \beta_2) = (0 \quad 1 \quad 0)$ is not rejected, whereas forecast 2 encompasses forecast 1 if $(\beta_0 \quad \beta_1 \quad \beta_2) = (0 \quad 0 \quad 1)$ is not rejected. Otherwise, both forecasts contain useful information.

4. Evaluation results and discussion

Means and standard deviations of the forecasts are presented in the following table. The results of the evaluation are presented in the following subsections.

Table 2. Descriptive statistics for forecasts

	Mean				Standard deviation			
	h =1	h =3	h =6	h =12	h=1	h =3	h =6	h =12
Univariate KNN	0.0185	0.0181	0.0181	0.0161	0.0057	0.0058	0.0063	0.0064
KNN with exogenous variables	0.0434	0.0433	0.0417	0.0314	0.0130	0.0158	0.0163	0.0213
Random forests	0.0236	0.0278	0.0346	0.0407	0.0052	0.0046	0.0040	0.0033
Extreme gradient boosting	0.0185	0.0181	0.0181	0.0161	0.0058	0.0068	0.0066	0.0110
LSTM model	0.0184	0.0192	0.0193	0.0212	0.0055	0.0045	0.0043	0.0019
Average of univariate methods	0.0190	0.0197	0.0201	0.0216	0.0057	0.0060	0.0066	0.0076

Source: Own elaboration

4.1. Unbiasedness

Table 3 shows that of the forecasts computed, only for those of the LSTM and the average of the univariate forecasts can the hypothesis of unbiasedness be maintained in at least one case. For the former, the null of unbiasedness is not rejected for the forecasts with horizons of 3, 6 and 12 months, while for the latter it is not rejected only for horizon h=1.

Table 3. Mincer and Zarnowitz (1969) unbiasedness test
P-value for the joint Wald test

	h = 1	h = 3	h = 6	h = 12
Univariate KNN	0.0299	0.0000	0.0000	0.0000
KNN with exogenous variables	0.0000	0.0000	0.0000	0.0000
Random forests	0.0000	0.0000	0.0000	0.0000
Extreme gradient boosting	0.0097	0.0000	0.0000	0.0000
LSTM model	0.0454	**0.0554**	**0.0503**	**0.7245**
Average of univariate methods	**0.0631**	0.0000	0.0000	0.0000

Source: Own elaboration

4.2. Forecast error corrrelations

In most cases forecast errors met the desirable properties set by Diebold and López (1996). For horizons h>1, Cumby and Huizinga tests show than in all cases the null that the errors follow at most an MA(h-1) process is not rejected. Additionally, for the errors of the univariate KNN, extreme, gradient boosting, the LSTM model and the average of the univariate forecasts there is no statistical evidence of autocorrelation for the errors at 1-month horizon.

Table 4. Autocorrelation tests for forecast errors
P-values [1/]

	h = 1		h = 3	h = 6	h = 12
	Wilcoxon/ Dufour	Cumby & Huizinga (1992)			
Univariate KNN	**0.1618**	**0.0740**	**0.3841**	**0.9331**	**0.3173**
KNN with exogenous variables	0.0000	0.0412	**0.4250**	**0.7268**	**0.3173**
Random forests	0.0000	**0.0977**	**0.1797**	**0.4401**	**0.4167**
Extreme gradient boosting	**0.3746**	**0.1433**	**0.4072**	**0.1934**	**0.3173**
LSTM model	**0.3603**	**0.2642**	**0.1524**	**0.5528**	**0.3173**
Average of univariate methods	**1.0000**	**0.4717**	**0.1933**	**0.5658**	**0.3173**

1/ P-values for Cumby and Huizinga (1992) statistics for the null of errors following a MA(h-1) versus alternative of MA(h). For h =1 the Wilcoxon test proposed by Dufour (1981) is used.
Source: Own elaboration

4.3. Precision

Table 5 shows the RSME and the Theil coefficient for the forecasts. At all horizons, the LSTM model show the highest precision, followed by the average of the univariate methods, the forecasts of the univariate KNN and those of extreme gradient boosting and random forests. The forecasts computed by KNN with exogenous variables show poor precision.

Are these differences in forecasting ability statistically significant? Table 10 of the annex shows the p-values for the Harvey, Leybourne y Newbold tests for each pair of forecasts at each horizon. At 5% significance, the tests show that the forecasts of LSTM and univariate KNN have the best performance, since those forecasts have the most significant differences in comparison with the rest of methods, and few significant differences between them. In particular, for horizons of 1 to 6 months, we cannot conclude that these two forecasts are significantly different.

For the shortest horizon, univariate forecasts are significantly more precise than all methods, except univariate KNN and extreme gradient boosting. Next are LSTM model forecasts, that outperform 2 methods. Furthermore, at this horizon the forecast obtained by random forests, extreme gradient boosting and univariate KNN are not significantly different. At longer horizons the performance of LSTM model forecasts is very good: for horizons of 3 and 6 months they outperform all methods except univariate KNN and extreme gradient boosting, and for h=12 they are significantly more precise than all the others. Extreme gradient boosting forecasts deteriorate sharply at h =12. Overall, the LSTM model shows the best performance, followed by univariate KNN and extreme gradient boosting, which are very similar between them, and random forests. KNN with exogenous variables is the worst performer, as it is significantly less precise than all methods at most horizons.

Table 5. RMSE and Theil inequality coefficient for forecasts

	RMSE				Theil [1/]			
	h = 1	h = 3	h = 6	h = 12	h = 1	h = 3	h = 6	h = 12
Univariate KNN	0.0039	0.0062	0.0067	0.0100	0.1006	0.1587	0.1676	0.25743
KNN with exogenous variables	0.0267	0.0277	0.0256	0.0244	0.4111	0.4174	0.3910	0.41104
Random forests	0.0057	0.0089	0.0145	0.0193	0.1295	0.1834	0.2612	0.30792
Extreme gradient boosting	0.0044	0.0072	0.0072	0.0144	0.1130	0.1851	0.1819	0.38501
LSTM model	0.0039	0.0036	0.0042	0.0024	0.1003	0.0892	0.1027	0.05648
Average of univariate methods	0.0032	0.0061	0.0065	0.0094	0.0805	0.1487	0.1552	0.21178

[1/] Values closer to zero indicate more precision.
Source: Own elaboration.

Precision in the prediction of the direction of changes in inflation is summarized in Table 6. It is noticeable the good performance of the random forests forecasts, since at all horizons the direction of the change in inflation is correctly predicted. This can be assessed from Figure 6: the series of random forests forecasts follow closely the movements of inflation, although generally at different levels. Univariate KNN and the LSTM model also show good performance, but for horizons of 1, 3 and 6 months, while extreme gradient boosting has a modest performance over the 3 and 6-month horizons.

Table 6. Prediction of the direction of changes in inflation
In bold if greater than 50%

	h = 1	h = 3	h = 6	h = 12
Univariate KNN	**0.5600**	**0.6087**	**0.5500**	0.4286
KNN with exogenous variables	**0.6400**	**0.6522**	0.3500	0.3571
Random forests	**0.6800**	**0.6522**	**0.6000**	**0.7143**
Extreme gradient boosting	0.4800	**0.6522**	**0.5500**	0.4286
LSTM model	**0.6000**	**0.5652**	**0.5500**	0.5000
Average of univariate methods	0.4800	**0.6087**	0.4500	0.2857

Source: Own elaboration

4.4. Encompassing tests

Tables Table ***11*** and Table ***12*** of the annex present the results of Chong and Hendry encompassing tests for each pair of forecasts at each horizon. At the shortest horizon, univariate-methods forecasts encompass those of KNN and boosting. At higher horizons the LSTM model forecasts again show a better performance, since they are the only ones to encompass other forecasts, particularly for h=3 and h=12, when they encompass 4 and 5 of the 5 forecasts in comparison, respectively.

4.5. Forecast error variance

Most of the forecasts evaluated do not have variances that increase with the horizon. This property is verified only for univariate KNN forecasts and extreme gradient boosting. However, as seen in Table 13 of the annex, in most cases it cannot be concluded that the variances at different forecast horizons are significantly different from each other. Only the variances for the forecast errors of univariate-method forecasts and univariate KNN at horizon h=1 can be said to be different from those at the rest of horizons.

Table 7. Variances of forecast errors

	h = 1	h = 3	h = 6	h = 12
Univariate KNN	0.00002	0.00004	0.00004	0.00007
KNN with exogenous variables	0.00012	0.00023	0.00021	0.00054
Random forests	0.00001	0.00002	0.00001	0.00001
Extreme gradient boosting	0.00002	0.00005	0.00005	0.00011
LSTM model	0.00002	0.00001	0.00002	0.00001
Average of univariate methods	0.00001	0.00004	0.00004	0.00010

Source: Own elaboration

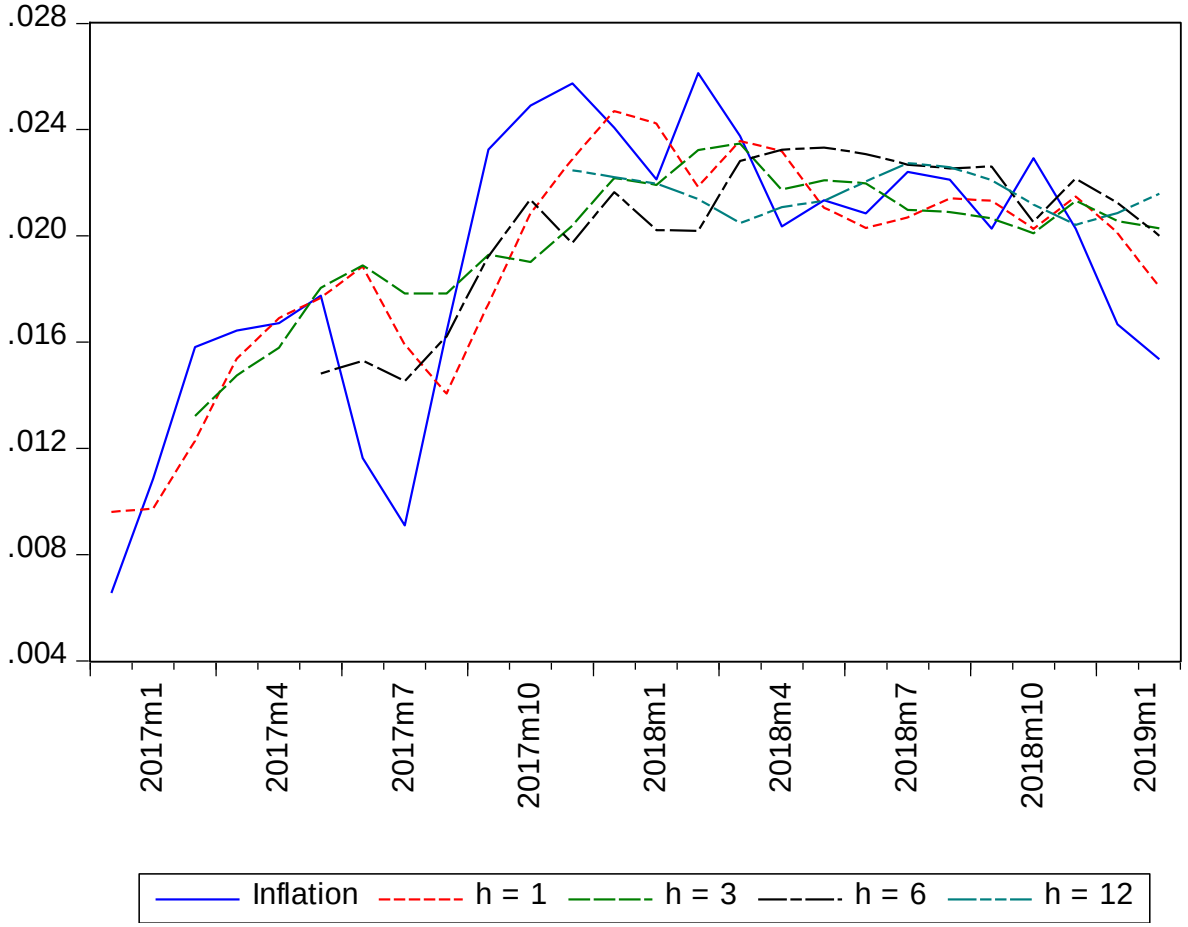
4.6. Combination of forecasts

In this section we evaluate a combination of the forecasts with the best individual performance. Ideally, the forecasts resulting from the combination should exhibit superior, or at least similar, properties as the individual forecasts. In particular, it must be noted that of the methods assessed only LSTM model produces unbiased forecasts, hence it would be desirable to obtain a combination that shows this property while keeping the other desirable properties of the forecasts included in its calculation.

From the analysis in sections 4.1 to 4.5 it can be concluded that the methods that consistently show desirable properties are LSTM and univariate KNN, although their ability to predict the direction of changes in inflation, while adequate, is lower than that of random forests. We decided to combine the forecasts of these three methods with the procedure suggested by Capistrán and Timmermann (CT)[5]. The results of the combination are shown in Figure 4 and those of its evaluation in Table 8 and Cuadro ***9***.

[5] See Capistrán and Timmermann (2009). The combination resulting from this method are the fitted values of a regression of the true values of the target variable on a constant and the simple average of the individual forecasts.

Figure 4. Combination of forecast at several horizons



Source: Own elaboration

In general, the results of the combination are very satisfactory. It is unbiased at all horizons, their forecast errors are not auto correlated at the h=1 horizon and at most follow a MA(h-1) process for h>1, and it has an ability to predict the direction of changes in inflation equal or higher than 50% at all horizons, but particularly those that are longer. Furthermore, at horizons of 3, 6 and 12 months the combination is significantly more precise than the individual forecasts with the more notable exception of the LSTM model. For the closest horizon, however, the combination is not more precise than the univariate KNN forecast, extreme gradient boosting nor the average of the univariate methods[6].

[6] The mean of these three individual forecasts was also evaluated, but their properties were not satisfactory. In particular, it is not unbiased at all horizons, it does not improve on the precision of individual forecast and is considerably less precise than the CT combination presented. Several CT combinations including extreme gradient boosting were also considered, resulting in undesirable properties (bias at longer horizons, poor prediction of direction of changes).

Table 8. Evaluation of the forecast combination
Several tests

	P-value for null hypothesis of unbiasedness	Prediction of direction of changes in inflation (% success)	P-values for autocorrelation tets [1/]	Forecast error variances
h=1	1.00	0.6400	0.2940	0.000010
h=3	1.00	0.6957	0.2832	0.000013
h=6	1.00	0.5000	0.5932	0.000012
h=12	1.00	0.5714	0.3173	0.000008

[1/] h=1 : Wilcoxon, others: Cumby and Huizinga
Source: Own elaboration

Cuadro 9. P-values for the Harvey, Leybourne and Newbold (1997) test
Combination vs individual methods
In bold if the forecast error of the column is significantly lower than that of the row

	Combination			
	h = 1	**h = 3**	**h = 6**	**h = 12**
Univariate KNN	0.0847	**0.0490**	**0.0234**	**0.0457**
KNN with exogenous variables	**0.0000**	**0.0000**	**0.0003**	**0.0013**
Random forests	**0.0091**	**0.0015**	**0.0000**	**0.0000**
Extreme gradient boosting	0.0999	**0.0037**	**0.0226**	0.0526
LSTM model	**0.0112**	0.4333	0.1966	0.6729
Average of univariate methods	0.4132	**0.0327**	**0.0029**	**0.0288**

Source: Own elaboration

5. Conclusions

The goal of this study was to perform a first evaluation of the capacity to forecast inflation of machine learning methods in Costa Rica. Forecasts were computed with 5 methods: two variants of KNN, random forests, extreme gradient boosting and a *long short-term memory* (LSTM) model. Their properties were evaluated according to the properties suggested by the literature on optimal forecasts, comparing their performance with that of the average of univariate methods described in Fuentes and Rodríguez (2016) and currently in use at the Central Bank of Costa Rica.

We found that the forecast with the best performance are those from the LSTM model, univariate KNN and to a lesser extent random forests and extreme gradient boosting. In particular, LSTM model forecasts are unbiased for horizons longer than 1 month, show more precision than the rest of the forecasts and they encompass most other forecasts.

A combination of forecasts improves the performance in comparison with individual forecasts at all horizons, and crucially, also outperforms the forecasts from univariate methods. The combination is unbiased, their errors do not show undesirable correlation patterns, and it improves forecasting ability at all horizons, both for the level of inflation and for the direction of its changes.

Given these results, we consider that the implementation of machine learning methods for forecasting at the BCCR is a promissory endeavor. A first line of work could be the improvement in the application of methods that underperformed in the study, as well as the potential extension of the work to include additional ML methods.

6. References

Agnon, Y., Golan, A. and Shearer, M. (1999). Nonparametric, nonlinear, short-term forecasting: theory and evidence for nonlinearities in the commodity markets. *Economics Letters*, *65*(3), pp. 293-299.

Athey, S. and Imbens, G.W. (2019). *Machine Learning Methods Economists Should Know About*. Eprint en arXiv:1903.10075.

Bajari, P., Nekipelov, D., Ryan, S.P. and Yang, M (2015). Machine Learning Methods for Demand Estimation. *American Economic Review, 105*(5), pp. 481-85.

Barkoulas, J., Baum, C.F. and Chakraborty, A. (1996). *Nearest-Neighbor Forecasts of U.S. Interest Rates*. Boston College Working Papers in Economics 313, Boston College Department of Economics, revisado en 2003.

Bengio, Y., Simard, P. and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks, 5*(2), pp.157-166.

Biau, O. and D'Elia, A. (2010). *Euro area GDP forecasting using large survey datasets. A random forest approach*. Euroindicators working papers EWP 2011/002.

Boelaert, J. and Ollion, É. (2018). The Great Regression: Machine Learning, Econometrics, and the Future of Quantitative Social Sciences. *Revue française de sociologie, 59*(3), pp.475-506. doi:10.3917/rfs.593.0475.

Breiman, L. (2001a). Random Forests. *Machine Learning*, *45*, pp. 5–32.

Breiman, L. (2001b). Statistical Modeling: The Two Cultures. *Statistical Science, 16*(3), pp. 199-215.

Breiman, L., Cutler, A., Liaw, A. and Wiener, M. (2018). *Paquete "randomForest" para R para la regresión y clasificación mediante bosques aleatorios*, versión 4.6-14, 22 de marzo de 2018.

Cameron, C. (2017). *Machine Learning for Microeconometrics* [presentación]. Presented at the School of Economics, University of Sydney, on April 12, 2017 and based on the 2016 course ECN240F at the Department of Economics, University of California - Davis.

Chen, T and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '16). ACM, New York, NY, USA, 785-794. DOI: https://doi.org/10.1145/2939672.2939785

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Li, Y. (2019*). R package "xgboost" for extreme gradient boosting*, version 0.82.1, March 11, 2019.

Chong, Y. Y., and Hendry, D. (1986). Econometric Evaluation of Linear Macro-Economic Models. *Review of Economic Studies*, 53, pp. 671-690.

Cook, T and Hall, A.S. (2017). *Macroeconomic Indicator Forecasting with Deep Neural Networks*. RWP-17-11. Federal Reserve Bureau of Kansas City.

Cumby, R. E., and Huizinga, J. (1992). Testing the autocorrelation structure of disturbances in ordinary least squares and instrumental variables regression. *Econometrica*, *60*, pp.185-195.

Diebold, F. and Nason, J. (1990). Nonparametric exchange rate prediction? *Journal of International Economics, 28*, pp.315-332.

Diebold, F. X., and López, J. A. (1996). Forecast Evaluation and Combination. In G. S. Maddala, & C. R. Rao, *Handbook of Statistics* (pp. 241-268). Amsterdam: NorthHolland.

Esquivel Monge, M. (2007). *Performance of Artificial Neural Networks in Forecasting Costa Rican Inflation*. DEC-DIE-029-2009-DI. Department of Economic Research, Central Bank of Costa Rica.

Fernández-Rodríguez, F., Sosvilla-Rivero, S. and Andrada-Felix, J. (1999). Exchange-rate forecasts with simultaneous nearest-neighbour methods: Evidence from the EMS. *International Journal of Forecasting, 15*, pp.383-392.

Fernández-Rodríguez, F., González-Martel, C. and Sosvilla-Rivero, S. (2000). On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid stock market. *Economics Letters, 69*(1), pp. 89-94.

Freund, Y., and Schapire, R. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. En *Computational Learning Theory,* Paul V (ed). Springer: Berlin, pp. 23–37.

Freund, Y. and Schapire, R. (1996). Experiments with a new boosting algorithm, *Proceedings of the Thirteenth International Conference on Machine Learning*: Morgan Kaufmann: Burlington, MA, 1996, pp.148–156.

Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics, 29*, pp.1189–1232.

Friedman, J., Hastie T, and Tibshirani R. (2000). Additive logistic regression: a statistical view of boosting. *Annals of Statistics, 28*, pp.337–407.

Fuentes Fuentes, M. and Rodríguez-Vargas, A. (2016). *Pronósticos univariados de inflación para Costa Rica con volatilidad estocástica y efectos GARCH* (Documento de Investigación No. 04-2016). Department of Economic Research, Central Bank of Costa Rica.

Gonzalez, S. (2000). *Neural Networks for Macroeconomic Forecasting: A Complementary Approach to Linear Regression Models.* Finance Canada Working Paper 2000-07.

Grothendieck, G. (2018). *Paquete "dyn" para R para regresión con series de tiempo*, versión 0.2-9.6, 18 de marzo de 2018.

Harvey, D., Leybourne, S., and Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, *13*, pp. 281-291.

Hastie, T., Tibshirani, R. and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC.

Hebb, D. (1949). *The Organization of Behavior*. New York: Wiley.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), pp.1735–1780.

Hyndman, R., Athanasopoulos, G., Bergmeir, C., Cáceres, G., Chhay, L., O'Hara-Wild, M., …, Yasmeen, F. (2019). *R package "forecast" for Time Series and Linear Models*, version 8.7, April 29, 2019.

Kingma, D. and Ba, J. (2017). *Adam: A Method for Stochastic Optimization*. Eprint en arXiv:1412.6980 [cs.LG]

Lehmann, R. and Wohlrabe, K. (2016). Looking into the black box of boosting: the case of Germany. *Applied Economics Letters*, *23*(17), pp. 1229-1233

Lisi, F. and Medio, A. (1997). Is a random walk the best exchange rate predictor? *International Journal of Forecasting, 13*, pp.255-267.

Lisi, F. and Schiavo, R. (1999). A comparison between neural networks and chaotic models for exchange rate prediction. *Computational Statistics & Data Analysis,* 30, pp.87-102.

Martínez, F., Frías, M.P., Pérez, M.D. and Rivera, A.J. (2017). A methodology for applying k-nearest neighbor to time series forecasting. *Artificial Intelligence Review*, pp.1-19. https://doi.org/10.1007/s10462-017-9593-z

Meade, M. (2002). A comparison of the accuracy of short term foreign exchange forecasting methods. *International Journal of Forecasting, 18*, pp. 67-83.

Mack, Y. P. (1981). Local properties of k-NN regression estimates. *SIAM Journal of Algebraic and Discrete Methods, 2*(3), pp.311-323.

McCarthy, J. (2007). *What Is Artificial Intelligence?* Stanford University, available at http://www-formal.stanford.edu/jmc/

McCulloch, W. and Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, *5*(4), pp.115–133.

Mincer, J., and Zarnowitz, V. (1969). The Evaluation of Economic Forecasts. In J. Mincer, *Economic Forecasts and Expectations: Analyses of Forecasting Behavior* (NBER Studies in Business Cycles, Vol. 19). Cambridge, MA: National Bureau of Economic Research.

Mizrach, B. (1992). Multivariate Nearest-Neighbour Forecasts of EMS Exchange Rates*. Journal of Applied Econometrics, 7*, pp. 151-163.

Moshiri, S. and Cameron, N. (2000). Econometrics versus ANN Model Forecasting Inflation. *Journal of Forecasting*, *19*(3), pp. 201-217.

Muñoz, E. and Tenorio, E. (2008). *El modelo macroeconómico de proyección trimestral del Banco Central de Costa Rica en la transición a la flexibilidad del tipo de cambio* (Documento de Investigación DIE-08-2008-DI). Department of Economic Research, Central Bank of Costa Rica.

Nakamura, (2006). Inflation forecasting using a neural network. *Economics Letters, 86*(3), pp.373-378.

Nowman, B. and Saltoglu, B. (2003). Continuous time and nonparametric modelling of U.S. interest rate models, *International Review of Financial Analysis, 12*, pp.25-34.

Nikolopoulos, K.I, Babai, Z. and Bozos, K. (2016). Forecasting supply chain sporadic demand with nearest neighbor approaches. *International Journal of Production Economics, 177*, pp.139-148.

Probst, P. and Boulesteix, A.L. (2018). To Tune or Not to Tune the Number of Trees in Random Forest. *Journal of Machine Learning Research, 18*, pp. 1-18.

Refenes, A. P. and H. White (1998). Neural Networks and Financial Economics. *International Journal of Forecasting, 6*(17), pp. 347-495.

Segal, M. R. (2004). Machine Learning Benchmarks and Random Forest Regression. *UCSF: Center for Bioinformatics and Molecular Biostatistics*. Retrieved from https://escholarship.org/uc/item/35x3v9t4

Solera, A. (2005). Pronóstico de inflación en Costa Rica: una estimación con redes neuronales artificiales. Documento de Trabajo. DCS-155-2005. Department of Social Accounting, Central Bank of Costa Rica.

Stock, J.H. and Watson, M. (1998). *A Comparison of Linear and Nonlinear Univariate Models for Forecasting Macroeconomic Time Series.* NBER Working Paper 6607.

Swanson, N. and White, H. (1995). A Model Selection Approach to Assessing the Information in the Term Structure Using Linear Models and Artificial Neural Networks. *Journal of Business & Economic Statistics*, *13*, pp. 265-275.

Swanson, N. and White, H. (1997), A Model Selection Approach To Real-Time Macroeconomic Forecasting Using Linear Models And Artificial Neural Networks. *The Review of Economics and Statistics, 79(*4), pp. 540-55.

Tkacz, G. and Hu, S. (1999). *Forecasting GDP growth using Artificial Neural Networks.* Working Paper 99-3, Bank of Canada.

Yakowitz, S. (1987). Nearest-Neighbour Methods for Time Series Analysis. *Journal of Time Series Analysis*, 8, pp. 10-26.

Yakowitz, S. and Karlsson, M. (1987). Nearest neighbour methods for time series, with application to rainfall/runoff prediction, in *Stochastic Hydrology*, Macneill, J. B. y Umphrey, G. J. (eds), D. Reidel Publishing Co., pp. 149-160.

Wang, H. and Raj, B. (2017). On the Origin of Deep Learning. Eprint en arXiv:1702.07800 [cs.LG], actualizado al 3 de marzo de 2017.

Wohlrabe, K. and Buchen. T. (2014). Assessing the Macroeconomic Forecasting Performance of Boosting: Evidence for the United States, the Euro Area and Germany. *Journal of Forecasting, 33*, pp. 231–242.

Zeng, J. (2017). Forecasting Aggregates with Disaggregate Variables: Does Boosting Help to Select the Most Relevant Predictors? *Journal of Forecasting*, *36*, pp. 74–90.

7. Annex

Figure 5. Variables used

Figure 5 (cont.). Variables used

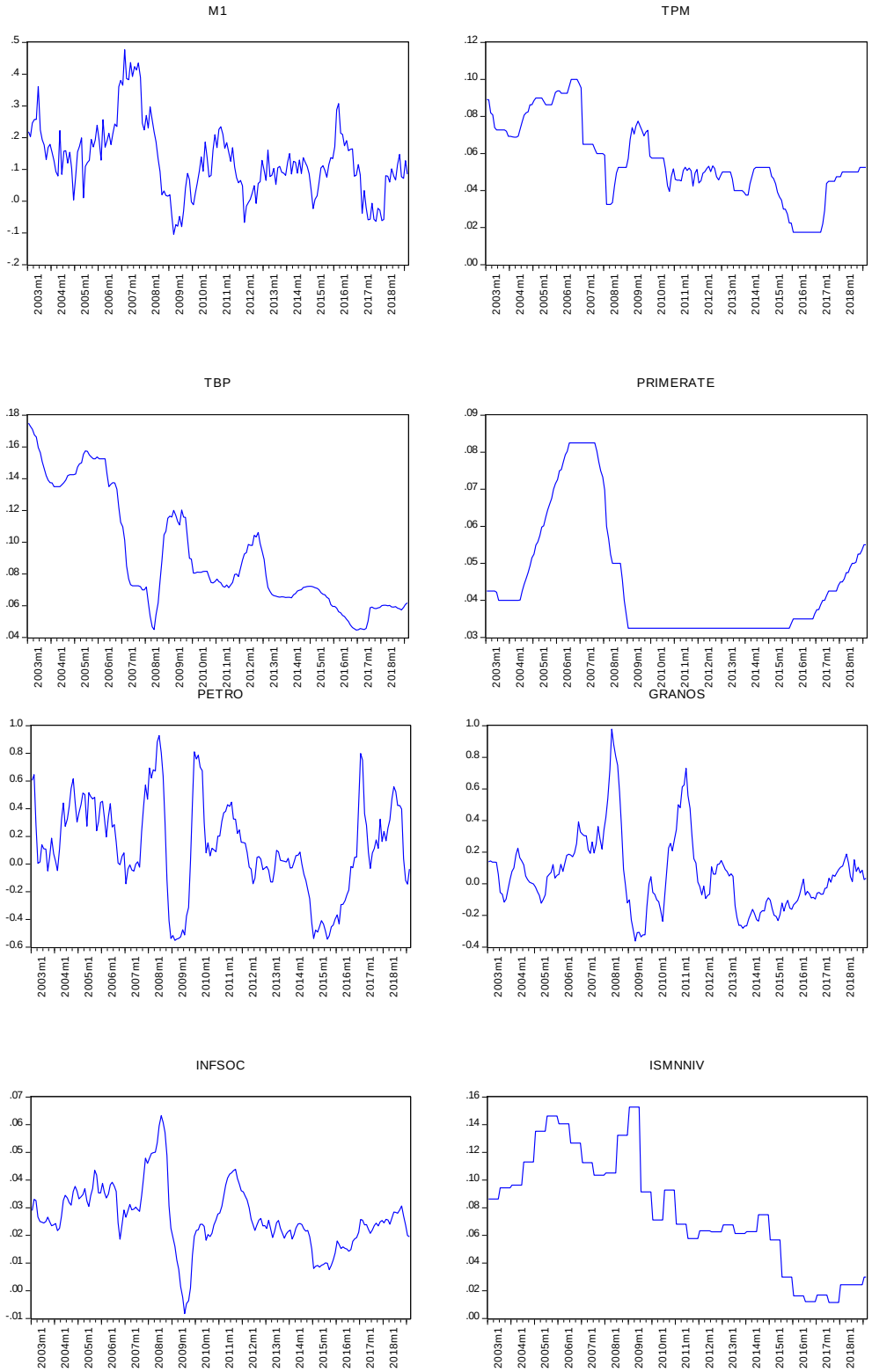
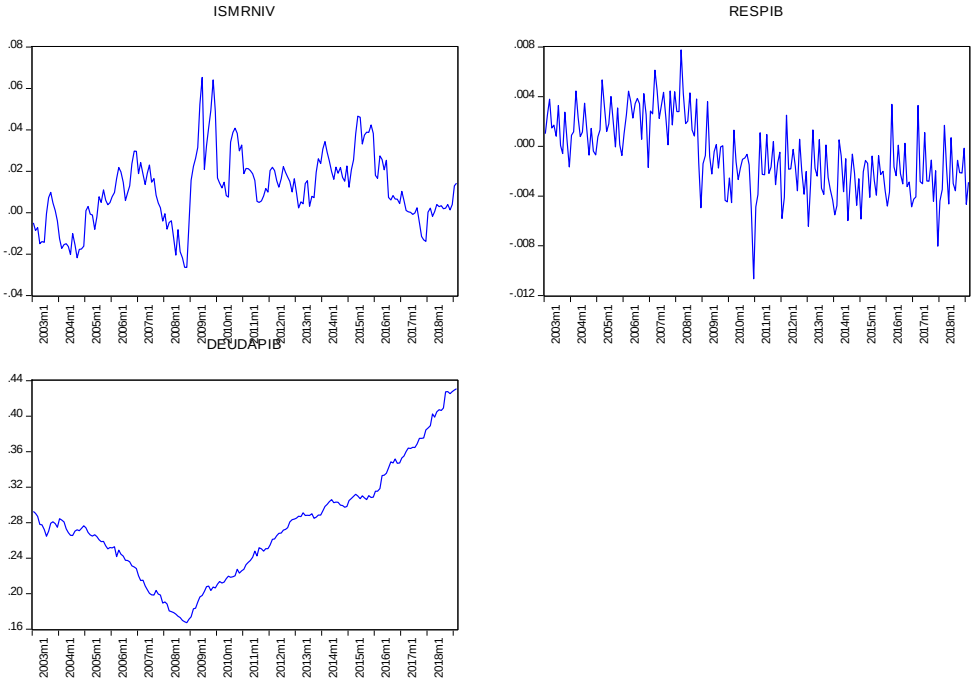
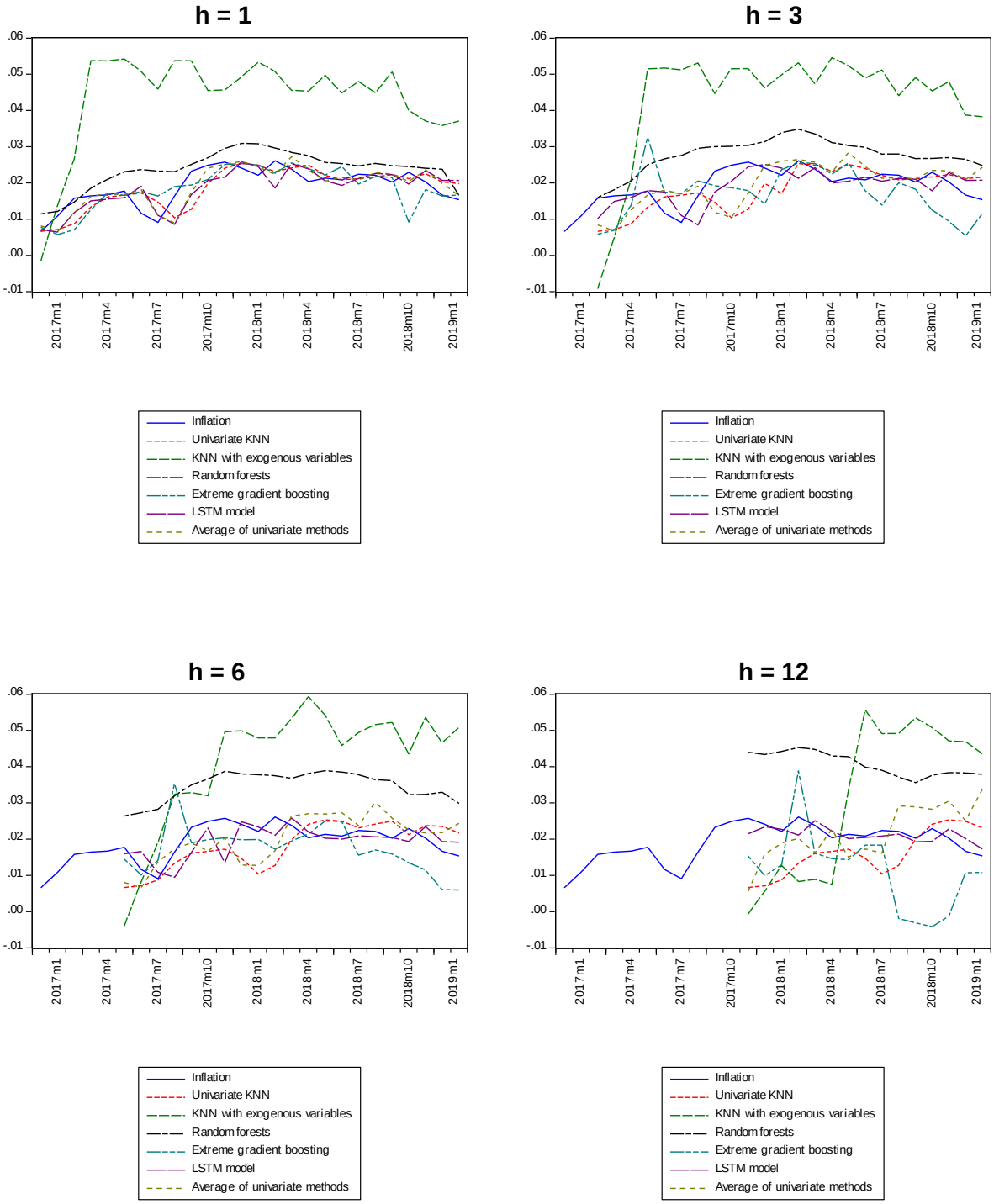


Figure 5 (cont.). Variables used



Source: Own elaboration

Figure 6. Forecasts at several horizons



Source: Own elaboration

Table 10. P-values for Leybourne y Newbold (1997) test

In bold if the forecast error of the column is significantly lower than that of the row

h = 1

	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		1.0000	0.9252	0.6721	0.4751	0.0743
KNN with exogenous variables	**0.0000**		**0.0000**	**0.0000**	**0.0000**	**0.0000**
Random forests	0.0748	1.0000		0.1709	**0.0484**	**0.0172**
Extreme gradient boosting	0.3279	1.0000	0.8291		0.3043	0.0998
LSTM model	0.5249	1.0000	0.9516	0.6957		**0.0057**
Average of univariate methods	0.9257	1.0000	0.9828	0.9002	0.9943	
Cases where column error is lower than row error:	1	0	1	1	2	3

h = 3

	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		1.0000	0.8898	0.7423	0.0579	0.3927
KNN with exogenous variables	**0.0000**		**0.0000**	**0.0000**	**0.0000**	**0.0000**
Random forests	0.1102	1.0000		0.1781	**0.0031**	0.0762
Extreme gradient boosting	0.2577	1.0000	0.8219		**0.0055**	0.2259
LSTM model	0.9421	1.0000	0.9969	0.9945		0.9645
Average of univariate methods	0.6073	1.0000	0.9238	0.7741	**0.0355**	
Cases where column error is lower than row error:	1	0	1	1	4	1

h = 6

	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		0.9995	1.0000	0.5882	0.0535	0.3980
KNN with exogenous variables	**0.0005**		**0.0041**	**0.0007**	**0.0004**	**0.0004**
Random forests	**0.0000**	0.9959		**0.0000**	**0.0000**	**0.0000**
Extreme gradient boosting	0.4118	0.9993	1.0000		0.0506	0.3657
LSTM model	0.9465	0.9996	1.0000	0.9494		0.9721
Average of univariate methods	0.6020	0.9996	1.0000	0.6343	**0.0279**	
Cases where column error is lower than row error:	2	0	1	2	3	2

h = 12

	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		0.9903	0.9995	0.7712	**0.0420**	0.4078
KNN with exogenous variables	**0.0097**		0.1374	**0.0046**	**0.0013**	**0.0031**
Random forests	**0.0005**	0.8626		0.1543	**0.0000**	**0.0001**
Extreme gradient boosting	0.2288	0.9954	0.8457		**0.0495**	0.1800
LSTM model	0.9580	0.9987	1.0000	0.9505		0.9690
Average of univariate methods	0.5923	0.9969	0.9999	0.8200	**0.0310**	
Cases where column error is lower than row error:	2	0	0	1	5	2

Source: Own elaboration

Table 11. Results of Chong y Hendry (1986) encompassing tests

For each pair of forecasts the table shows which forecast encompasses the other

h = 1

	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN						Column
KNN with exogenous variables						Column
Random forests						
Extreme gradient boosting						Column
LSTM model						
Average of univariate methods	Row	Row		Row		
Cases where column forecast encompasses row forecast:	0	0	0	0	0	3

h = 3

	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN					Column	
KNN with exogenous variables					Column	
Random forests						
Extreme gradient boosting					Column	
LSTM model	Row	Row		Row		Row
Average of univariate methods					Column	
Cases where column forecast encompasses row forecast:	0	0	0	0	4	0

h = 6

	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN					Column	
KNN with exogenous variables						
Random forests						
Extreme gradient boosting						
LSTM model	Row					Row
Average of univariate methods					Column	
Cases where column forecast encompasses row forecast:	0	0	0	0	2	0

h = 12

	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN					Column	
KNN with exogenous variables					Column	
Random forests					Column	
Extreme gradient boosting					Column	
LSTM model	Row	Row	Row	Row		Row
Average of univariate methods					Column	
Cases where column forecast encompasses row forecast:	0	0	0	0	5	0

Source: Own elaboration

Table 12. P-values of Chong y Hendry (1986) encompassing tests

h = 1

Ho: $(\beta_0 \quad \beta_1 \quad \beta_2) = (0 \quad 1 \quad 0)$	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		0.0214	0.0014	0.0374	0.0550	0.0001
KNN with exogenous variables	0.0000		0.0000	0.0000	0.0000	0.0000
Random forests	0.0000	0.0000		0.0000	0.0000	0.0000
Extreme gradient boosting	0.0009	0.0178	0.0000		0.0005	0.0000
LSTM model	0.0672	0.0233	0.0006	0.0318		0.0000
Average of univariate methods	0.0910	0.0544	0.0338	0.1365	0.0150	0.0868

Ho: $(\beta_0 \quad \beta_1 \quad \beta_2) = (0 \quad 0 \quad 1)$	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		0.0000	0.0000	0.0009	0.0672	0.0910
KNN with exogenous variables	0.0214		0.0000	0.0178	0.0233	0.0544
Random forests	0.0014	0.0000		0.0000	0.0006	0.0338
Extreme gradient boosting	0.0374	0.0000	0.0000		0.0318	0.1365
LSTM model	0.0550	0.0000	0.0000	0.0005		0.0150
Average of univariate methods	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000

h = 3

Ho: $(\beta_0 \quad \beta_1 \quad \beta_2) = (0 \quad 1 \quad 0)$	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		0.0000	0.0000	0.0000	0.0000	0.0000
KNN with exogenous variables	0.0000		0.0000	0.0000	0.0000	0.0000
Random forests	0.0000	0.0000		0.0000	0.0000	0.0000
Extreme gradient boosting	0.0000	0.0000	0.0000		0.0000	0.0000
LSTM model	0.1324	0.1361	0.0371	0.0726		0.1152
Average of univariate methods	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Ho: $(\beta_0 \quad \beta_1 \quad \beta_2) = (0 \quad 0 \quad 1)$	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		0.0000	0.0000	0.0000	0.1324	0.0000
KNN with exogenous variables	0.0000		0.0000	0.0000	0.1361	0.0000
Random forests	0.0000	0.0000		0.0000	0.0371	0.0000
Extreme gradient boosting	0.0000	0.0000	0.0000		0.0726	0.0000
LSTM model	0.0000	0.0000	0.0000	0.0000		0.0000
Average of univariate methods	0.0000	0.0000	0.0000	0.0000	0.1152	

Source: Own elaboration

Table 12 (cont.). P-values of Chong y Hendry (1986) encompassing tests

h = 6

Ho: $(\beta_0 \quad \beta_1 \quad \beta_2) = (0 \quad 1 \quad 0)$	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		0.0000	0.0000	0.0000	0.0000	0.0000
KNN with exogenous variables	0.0000		0.0000	0.0000	0.0000	0.0000
Random forests	0.0000	0.0000		0.0000	0.0000	0.0000
Extreme gradient boosting	0.0000	0.0000	0.0000		0.0000	0.0000
LSTM model	0.1105	0.0358	0.0000	0.0035		0.0868
Average of univariate methods	0.0000	0.0000	0.0000	0.0000	0.0000	

Ho: $(\beta_0 \quad \beta_1 \quad \beta_2) = (0 \quad 0 \quad 1)$	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		0.0000	0.0000	0.0000	0.1105	0.0000
KNN with exogenous variables	0.0000		0.0000	0.0000	0.0358	0.0000
Random forests	0.0000	0.0000		0.0000	0.0000	0.0000
Extreme gradient boosting	0.0000	0.0000	0.0000		0.0035	0.0000
LSTM model	0.0000	0.0000	0.0000	0.0000		0.0000
Average of univariate methods	0.0000	0.0000	0.0000	0.0000	0.0868	

h = 12

Ho: $(\beta_0 \quad \beta_1 \quad \beta_2) = (0 \quad 1 \quad 0)$	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		0.0000	0.0000	0.0000	0.0000	0.0000
KNN with exogenous variables	0.0000		0.0000	0.0000	0.0000	0.0000
Random forests	0.0000	0.0000		0.0000	0.0000	0.0000
Extreme gradient boosting	0.0000	0.0000	0.0000		0.0000	0.0000
LSTM model	0.0693	0.4695	0.3764	0.0001		0.1260
Average of univariate methods	0.0000	0.0000	0.0000	0.0000	0.0000	

Ho: $(\beta_0 \quad \beta_1 \quad \beta_2) = (0 \quad 0 \quad 1)$	Univariate KNN	KNN with exogenous variables	Random forests	Extreme gradient boosting	LSTM model	Average of univariate methods
Univariate KNN		0.0000	0.0000	0.0000	0.0693	0.0000
KNN with exogenous variables	0.0000		0.0000	0.0000	0.4695	0.0000
Random forests	0.0000	0.0000		0.0000	0.3764	0.0000
Extreme gradient boosting	0.0000	0.0000	0.0000		0.0001	0.0000
LSTM model	0.0000	0.0000	0.0000	0.0000		0.0000
Average of univariate methods	0.0000	0.0000	0.0000	0.0000	0.1260	

Source: Own elaboration

Table 13. Tests of equality of variances

P-values of tests for forecasts at horizons indicated in row/column

Univariate KNN	1	3	6	12
1		**0.0398**	**0.0230**	**0.0007**
3	**0.0398**		0.7835	0.1320
6	**0.0230**	0.7835		0.2265
12	**0.0007**	0.1320	0.2265	

KNN with exogenous variables	1	3	6	12
1		0.1357	0.2013	**0.0013**
3	0.1357		0.8693	0.0629
6	0.2013	0.8693		0.0531
12	**0.0013**	0.0629	0.0531	

Random forests	1	3	6	12
1		0.3090	0.4251	0.4631
3	0.3090		0.0864	0.1250
6	0.4251	0.0864		0.9788
12	0.4631	0.1250	0.9788	

Extreme gradient boosting	1	3	6	12
1		**0.0360**	0.0520	**0.0002**
3	**0.0360**		0.9243	0.0774
6	0.0520	0.9243		0.0744
12	**0.0002**	0.0774	0.0744	

LSTM model	1	3	6	12
1		0.6502	0.7874	0.0759
3	0.6502		0.4876	0.1593
6	0.7874	0.4876		0.0534
12	0.0759	0.1593	0.0534	

Average of univariate methods	1	3	6	12
1		**0.0021**	**0.0009**	**0.0000**
3	**0.0021**		0.7385	0.0513
6	**0.0009**	0.7385		0.1137
12	**0.0000**	0.0513	0.1137	

Source: Own elaboration

Table 14. Software used

Method	Software	Details
Univariate KNN	R	*Package tsfknn* by Martínez, Frías, Pérez and Rivera (2017) for the application of KNN to time series.
KNN with exogenous variables	MATLAB	Function *knnsearch* was used to find the neareast neighbours
Random forests	R	Package *randomForest* by Breiman et al. (2018), for the generation of random forests for regression and classification. Package *dyn* by Grothendieck (2018), that allows the creation of interfaces with several functions that do regression, *randomForests* among them. This package makes possible to use those functions with time series data, including specifications with lags and differences.
Extreme gradient boosting	R	Package *xgboost* by Chen et al (2015).
LSTM model	R	*Keras* package, requires TensorFlow and Python.
Average of univariate methods	MATLAB, Eviews	

Source: Own elaboration