**Working
Paper**

498

**CEMAP - Nº13
CEQEF - Nº49
Working Paper Series**
Fevereiro de 2019

**SAO PAULO SCHOOL
OF ECONOMICS**

Cross-Validation based Forecasting Method: A Machine Learning Approach

**Jeronymo Marcondes Pinto
Emerson Fernandes Marçal**

Os artigos dos *Textos para Discussão da Escola de Economia de São Paulo da Fundação Getúlio Vargas* são de inteira responsabilidade dos autores e não refletem necessariamente a opinião da FGV-EESP. É permitida a reprodução total ou parcial dos artigos, desde que creditada a fonte.

Escola de Economia de São Paulo da Fundação Getúlio Vargas FGV-EESP
www.eesp.fgv.br

Cross-Validation based Forecasting Method: A Machine Learning Approach

Jeronymo Marcondes Pinto¹ and Emerson Fernandes Marçal²

¹Brazilian Ministry of Labor and CEMAP-EESP-FGV.

Phone: 55-11-3799-3228

Email: jeronymomp@gmail.com

²Sao Paulo School of Economics and CEMAP-EESP-FGV.

Rua Itapeva, 286. 10^o floor.

Phone: 55-11-3799-3228.

Email: emerson.marcal@fgv.br

February 6, 2019

This Draft

Abstract

Our paper aims to evaluate two novel methods on selecting the best forecasting model or its combination based on a Machine Learning approach. The methods are based on the selection of the "best" model, or combination of models, by cross-validation technique, from a set of possible models. The first one is based on the seminal paper of Granger-Bates (1969) but weights are estimated by a process of cross-validation applied on the training set. The second one selects the model with the best forecasting performance in the process described above, which we called CvML (Cross-Validation Machine Learning Method). The following models are used: exponential smoothing, SARIMA, artificial neural networks and Threshold autoregression (TAR). Model specification is chosen by R packages: forecast and TSA. Both methods – CvML and MGB - are applied to these models to generate forecasts from one up to twelve periods ahead. Frequency of data is monthly. We run the forecasts exercise to the following to monthly series of Industrial Product Indices for seven countries: Canada, Brazil, Belgium, Germany, Portugal, UK and USA. The data was collected at OECD data, with 504 observations. We choose

Average Forecast Combination, Granger Bates Method, MCS model, Naive and Seasonal Naive Model as benchmarks. Our results suggest that MGB did not performed well. However, CvML had a lower mean absolute error for most of countries and forecast horizons, particularly at longer horizons, surpassing all the proposed benchmarks. Similar results hold for absolute mean forecast error.

Keywords— Economic Forecasting, Cross-Validation, Machine Learning, Model Combination, Model Selection.

JEL codes: C53, C40, C20.

1 Introduction

Advances in information sciences are allowing substantial improvements regarding statistical analysis, with special emphasis on time series forecasting. Current computational resources allow analysts to estimate a plethora of models, raising the challenge of comparing and ranking different forecasting methods. On the other hand, a number of new techniques that are computationally intensive have been proposed and evaluated.

In this scenario, automatic algorithms are emerging as useful tools to generate forecasts (Hyndman and Khandakar (2008), Cryer and Kellet (1991), and Hendry and Doornik (2014)). It is not clear which method is going to prevail. Examples of recent studies that use computational methods to improve forecasting model performance are Duchêne et al. (2016), Roberts et al. (2017), and Bergmeir et al. (2015). A similar scenario holds for model selection in terms of forecasting performance (Jung et al. (2015), Alzahrani et al. (2018), and Hansen et al. (2011)).

The seminal work of Bates and Granger (1969) suggested that a simple forecast combination, such as simple or weighted averages, can perform better than individual models. Timmermann (2006) provides a review of the literature on forecast combinations. The importance of model combinations has been highlighted in recent papers, such as Hsiao and Wan (2014) and Chan and Pauwels (2018).

Our paper evaluates two different strategies to generate good forecasts using forecast combinations and time series econometric models. A machine learning framework is used to construct forecasts by applying cross-validation techniques on a training set. The following time series models are used to generate forecasts to be combined or selected: exponential smoothing, SARIMA, artificial neural networks (ANNs), and threshold autoregression (TAR). All of these models' functional specifications are automatically provided by their *R* packages.

We run a pseudo real time forecast exercise to evaluate the forecasting performance of our strategy by applying it to the Industrial Product Indexes of the following countries: Canada, Brazil, Belgium, Germany, Portugal, UK, and USA. We compare our forecast to naive benchmarks, such as naive and seasonal naive models and other forecast combination methods, as well as the one proposed by Bates and Granger (1969),

a simple mean of forecasts, and a combination approach strategy proposed by Garcia et al. (2017) that uses a model confidence set (MCS) developed by Hansen et al. (2011) to select which models are going to be combined.

This paper is organized as follows. Section 1 discusses our proposed strategy to generate forecasts. Section 2 reports our results. Section 3 discusses the merits and pitfalls of our strategy. Finally some concluding remarks are drawn.

2 Material and Methods

Arlot et al. (2010) highlight the cross-validation method, Cv hereafter, as one of the best techniques to evaluate predictive performance of a model involving large samples. Bergmeir et al. (2015) highlight the use of Cv to improve the performance of time series models and deals with issues such as serial correlation and/or non-stationarity in data.

Arlot et al. (2010), Bergmeir and Benítez (2012), and Bergmeir et al. (2014) reported good performance using Cv on real-world time series data and on artificial data based on a data generating process usually assumed to hold in time series analysis. Our present paper evaluates two strategies to generate forecasts based on the Cv technique.

The first strategy aims to improve the Granger and Bates method (Bates and Granger (1969)) by using Cv to better estimate the weights that are used in forecast combinations. Weights are estimated by Cv application on the training set. We name this procedure as Modified Granger and Bates (*MGB*). The second strategy selects the best model to be used for forecasting based on the Cv technique, which we call the Cross-Validation Machine Learning Method (*CvML*). We use the mean absolute error (MAE) norm to evaluate the models in the training set.

2.1 Modified Granger and Bates Strategy

Bates and Granger (1969), GB hereafter, combines two different models based on the following equations:

$$f_{GB,t} = \omega_1 f_{1,t} + (1 - \omega_1) f_{2,t}, \quad (1)$$

$$\omega_{1,t} = \frac{\sum_{T-v}^{T-1} \hat{e}_{1,t}^2}{\sum_{T-v}^{T-1} \hat{e}_{1,t}^2 + \sum_{T-v}^{T-1} \hat{e}_{2,t}^2}, \quad (2)$$

where $\hat{e}_{t-i} = y_{t-i} - f_{t-i}$. For the case where n models are used, the *MGB* method is specified by the following equations:

$$f_{MGB,t} = \omega_{1,t} f_{1,t} + \omega_{2,t} f_{2,t} + \dots + \omega_{n,t} f_{n,t}, \quad (3)$$

$$\omega_{h,t} = \frac{\sum_{i=0}^{\tau-1} \frac{|\hat{e}_{t-i-1|t-i-2}^h|}{\tau}}{\sum_{h=1}^n \sum_{i=0}^{\tau-1} \frac{|\hat{e}_{t-i-1|t-i-2}^h|}{\tau}} \quad \forall t = 0, 1, 2, \dots, T, \quad (4)$$

where $\hat{e}_{t-i|t-i-1}^h = y_{t-i+1} - f_{t-i|t-i-1}^h$, and $f_{t-i|t-i-1}^h$ is defined as the forecast for $t - i$ using all information from the beginning of the sample until $t - i - 1$ for model h . τ is the growing window used to calculate the weights. For seasonal data we suggest a multiple of the seasons to minimize any possible effects of seasonality on weights.

Forecast errors $\hat{e}_{t-i-1|t-i-2}^h$ are generated using information up to $t - i - 2$. The MAE is calculated using the last τ observations. For example, for monthly data, $\tau=12$ can be a possible choice.

Our procedure is applied on an expanding window basis. The initial point is fixed; models are estimated from a sample covering the period that starts at 0 and ends at $t - i - 2$. Any set of time series can be chosen, but we opt to work with the following set: exponential smoothing (ETS), seasonal autoregressive integrated moving average (SARIMA), artificial neural networks (ANNs) and threshold autoregression (TAR). All of these models' functional specifications are given by their *R* packages, known as *forecast* and *TSA* packages. Those packages are well established in the *R* community, and have been described in Hyndman and Khandakar (2008) and Cryer and Kellet (1991), respectively.

Therefore, our strategy is to generate good forecasts for a period of time t , with $\tau = 12$, because we are working with monthly data:

1. Evaluate each model's performance from 0 until $t - 12$, increasing by one unit after each step until reaching $t - 1$.
2. Calculate the MAE for each model.
3. Generate $\omega_{h,t}$ based on the relative MAE for each model with respect to all others.

2.2 Cross-validation machine learning strategy

CvML follows a similar logic as *MGB* but no weights are generated. *CvML* picks the model with lowest MAE generated from each information set. The procedure is defined in the following equation:

$$\arg \min \Theta_t(h), \quad (5)$$

where $\Theta_t(h)$ is the chosen norm used to pick the "best" model. In our case we opt to use the mean absolute error $MAE = \sum_{i=0}^{\tau-1} \frac{|\hat{e}_{t-i-1|t-i-2}^h|}{\tau}$.

2.3 Benchmarks

To evaluate any strategy, it is important to choose proper benchmarks. If a strategy is unable to outperform forecasts obtained from simple benchmarks, it should be abandoned. Simple benchmarks serve as a lower bound to assess any strategy. For example, if the analyst wants to forecast an exchange rate, random walk is a difficult benchmark to be

beaten (Rossi (2013) and Meese and Rogoff (1983)). An autoregressive model of order 1 is a difficult benchmark when forecasting a consumer price index (Castle et al. (2013) and Stock and Watson (2002)). A forecast obtained from a double difference model can be a difficult benchmark in data where the data generator process faces structural breaks (Clements and Hendry (2001)). We now describe the benchmarks used in this paper.

Average Forecast Combination

One simple way of combining model forecasts is to calculate their average. Arlot et al. (2010) indicate that, among several possible models to forecast a time series, an aggregate version of these models, as an arithmetic mean of these predictions, will outperform any individual estimates. Thus, to forecast a given period t carried out by k different models (y_t^i), the arithmetic mean of forecasts (y_{MA}) would be given by $y_{MA} = \frac{1}{k} \sum_{i=1}^k y_t^i$.

Granger Bates Method

Given that we are using a strategy that modifies how weights are calculated in the GB method, a natural benchmark is the original approach proposed by Bates and Granger (1969). The combined forecast can be calculated from the forecast of k different models: $C_t = w_1 f_{1t} + w_2 f_{2t} + \dots + w_k f_{kt}$, where C_t is the forecast weighted average for time t and w_i is the weight associated with each model forecast (f_i). As discussed in Elliott and Timmermann (2008), a generalization of the Bates and Granger (1969) weight formula is:

$$w_{ih} = \frac{\frac{1}{\sum_{i=1}^n e_{ih}}}{\frac{1}{\sum_{i=1}^n e_{i1}} + \frac{1}{\sum_{i=1}^n e_{i2}} + \dots + \frac{1}{\sum_{i=1}^n e_{ik}}}.$$

Seasonal Naive Model

The forecast k steps ahead of a variable (y_t) is given by the value it assumed in the previous seasonal period. As an illustration, the forecast of an observation in February 2017 would be given by the value that this variable assumed in February 2016, as this example works with monthly data. Analytically, the forecast of a variable (y_t) for a period k will be given by $y_t = y_{t-\tau}$, where τ is the number of seasons. This study works with monthly data, so $\tau = 12$.

Naive Model

It is a process characterized by the following equation: $y_t = y_{t-1}$.

3 Results

3.1 Data

We opt to run our pseudo real time exercise using the Index of Industrial Products for seven different countries: Brazil, United Kingdom, United States, Belgium, Germany, Canada, and Portugal. This dataset was obtained from OECD data ¹.

Industrial production refers to the volume of output generated by production units classified under the industrial sectors, i.e. B (Mining), C (Manufacturing), and D and E (Electricity, gas and water) of the International Standard Industrial Classification of all economic activities provided by OECD. The data is seasonally adjusted by OECD. Our full sample contains 504 observations in a monthly frequency beginning in November of 1975 and ending in October of 2017. We opt to use the Napierian logarithm transformation. The levels contain one unit root as indicated by the Dickey-Fuller and Phillips-Perron tests. We are unable to reject the null hypothesis of a unit root at 1% for all countries.

3.2 Findings

Packages

The experiment was developed by application of all cited models and methods to the Index series. *MGB* and *CvML* aim to select the best model or combination of models from a set of possible choices. Specifications for each model are selected using algorithms from the following *R* packages:

1. *SARIMA* - *forecast* package
2. Exponential Smoothing (*ETS*)- *forecast* package
3. Artificial Neural Network (*ANN*) - *forecast* package
4. *TAR* - *TSA* package

The *forecast* package is described in detail in Hyndman and Khandakar (2008) whereas the *TSA* package is based on the works of Tong (1990) and Cryer and Kellet (1991). For more details regarding the aforementioned methodologies, see Hamilton (1994) and James et al. (2013). Each package chooses a particular specification based on the information available. Model performance may vary throughout the sample. Our strategy allows that our algorithms may pick different models and specifications in each period, true to the spirit of machine learning approaches.

Pseudo real time experiment

---

¹<https://data.oecd.org/>

The data gathered for the countries is used to create many variants of models to forecast industrial production. The sample is split into two parts. The first half of the sample is used to estimate the models, and the second half is used to evaluate the forecast performance of the models over various horizons. In the exercise, we attempt to simulate a real-time operation. We use an information set that reflects, as closely as possible, the one available to agents at the time of the forecast. In other words, the models are re-estimated at each point to incorporate any new information arriving at each instant of time.

For each model, forecasts are generated for one, two, three, and twelve steps ahead. Industrial production index series are revised from their initial publication in their original sources. Therefore, our initial training set includes the first 350 observations, with a test set composed of one observation that can be 1, 2, 3, or 12 steps ahead of the last one used by the model. Hence, we are using Leave-One-Out Cross-validation (Arlot et al. (2010)).

The values we use to run our projections are not exactly the same as those that were available to agents at that time. We run projections in our pseudo real-time experiment with a slightly better information set. This may result in better forecasting accuracy compared to the projections generated in real time. Tests using vintage data are left for future research.

To assess predictive performance of the proposed models, a comparison of the mean absolute error (MAE) generated for each method is made. The forecasting models performance is primarily based on MAE analysis; however a mean square error ($MSFE$) analysis is also performed. To assure the robustness of the results, we evaluate the forecasting performance of the automated models, *forecast* and *TSA* packages, in addition to the aforementioned benchmarks.

Tables 1 and 2 present a summary of the best models in terms of MAE and $MSFE$ for one to twelve steps ahead. Our results suggest that *CvML* outperforms other models in most of the cases in terms of MAE . *CvML* performs better for Belgium, Canada, UK, and USA. Similar results hold for the $MSFE$ case. Tables 7 and 8 provided in the Appendix show that *CvML* is the second best choice in many cases as well.

On the other hand, the *MGB* strategy performance is disappointing for both MAE and $MSFE$ criteria.

To take a closer look at the results, we opt to run the Diebold and Mariano (DM) equal forecast performance test (Diebold and Mariano (2002)). We compare all best models with the benchmarks. Table 3 contains a summary of the results of the Diebold and Mariano tests.

Two observations can be made from results of 3. The *CvML* strategy outperforms naive benchmarks for most of the cases, with the only exception being Germany. A tougher benchmark is the Average Forecast Combination method. Best results are obtained for Belgium, Canada, Brazil, and the UK. Concerning the Portuguese case, the

CvML strategy outperforms simple benchmarks but it has equivalent performance in statistical terms to a forecast combination. Finally, the *CvML* strategy performance is quite good for twelve steps ahead forecasts.

4 Discussion

4.1 Unconditional forecast performance analysis

In this section, we investigate whether the *CvML* strategy performance is linked to economic cycles.

To do this, we have to date peaks and troughs in our industrial production indexes series. The Harding and Pagan (2002) algorithm was developed for quarterly data. Given that our data is monthly, we apply Bry and Boschan (1971)'s algorithm. The literature on the subject of economic cycle dating is extensive, so we only discuss the algorithm we chose as follows.

Bry and Boschan (1971) classify each observation as ascending or descending and identifies turning points. With this information, we construct a series of dummies for each country whose values depend on whether the economy is growing or contracting. The following equation defines the dummies.

$$DummyCycle = \left\{ \begin{array}{l} 1 \rightarrow expansion \\ 0 \rightarrow contraction \end{array} \right\}. \quad (6)$$

We collect the forecast errors of the *CvML* model and regress these on a constant and a cycle of dummies. If there is evidence that the constant is significant, then there is a bias on the predictions. On the other hand, if a dummy cycle proves to be significant, we conclude that the model performs differently according to business cycles, so there is conditional predictability. Table 4 shows the results of the dating algorithm applied to our series, indicating the percentage of times an observation was marked as an expansion or contraction by the Bry and Boschan (1971) algorithm (BB).

Table 5 reports the results of the aforementioned regression for all countries and steps. It is possible to conclude that there is no evidence for most cases and weak evidence for some cases that either the constant or the dummies are important. This indicates the strength of the *CvML* results. The cases where there is evidence that cycle matters or where there is bias are those where the *CvML* strategy was not among the best models.

4.2 *CvML* model selection algorithm

Table 6 reports the frequency for which each model was selected to provide the forecast by the *CvML* strategy. There is no model that is systematically chosen by the algo-

rithm, suggesting that the Data Generating Mechanism is possibly unstable and may be experiencing some sort of structural change, under this environment it is expected that our strategy switches to copy other scenarios. The *CvML* strategy selects TAR and ETS frequently and they are the most frequent choices for expansion and contraction scenarios, respectively.

4.3 Limitations and possible extensions

Our work has some limitations that should be pointed out. We evaluated only one type of cross-validation method, and it is important to evaluate other types of Cv methods that may surpass some of the problems associated with this method, such as auto-correlation and unit root, as discussed in Bergmeir et al. (2015) and Racine (2000).

It is also important to verify the strategic capability of the model's generality. This can be accomplished in various ways, such as with applications to Monte Carlo simulation series and/or mathematical analysis of its asymptotic properties. In this sense, our results are limited to the series of Industrial Product Indexes.

It is also important to incorporate other models in the analysis. In our study we only included four models (SARIMA, TAR, ETS, and ANN). Therefore, we can include another set of models as a way of generalizing our results.

A future track of research can be set as incorporating the Castle et al. (2015) algorithm to forecasting under structural breaks. That approach can generalize *CvML* to have better performance over different scenarios.

5 Concluding Remarks

This work aims to evaluate a new strategy to construct forecasts from a set of models that are automatically estimated by *R* packages. In a nutshell, we evaluate two strategies. The first is based on a modified version of the Bates and Granger (1969) model that uses a cross-validation process on the training set. The second is based on selecting the model with the lowest MAE in the training set. Both strategies are based on a machine learning approach. *CvML* and *MGB* are applied to model the industrial production indexes of seven countries: Canada, Brazil, Belgium, Germany, Portugal, UK, and USA using OECD data.

Our results suggest that the *MGB* method proposed is not a good choice. On the other hand, the proposed *CvML* method achieved good performance when applied to this dataset for most time horizons and countries, outperforming simple benchmarks (naive and seasonal naive) and more complex ones, such as the simple average forecast combination technique, the traditional Granger and Bates model, and the MCS strategy (Garcia et al. (2017)).

As a robustness check, we compared *CvML* to benchmarks using the Diebold Mariano test and also investigated whether the *CvML* performance was related to economic cycles. Our conclusions are that *CvML* statistically outperforms benchmarks with few exceptions, and that forecast performance does not depend on economic cycles.

References

- Alzahrani, N., Neal, P., Spencer, S. E., McKinley, T. J., and Touloupou, P. (2018). Model selection for time series of count data. *Computational Statistics & Data Analysis*.
- Arlot, S., Celisse, A., et al. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79.
- Bates, J. M. and Granger, C. W. (1969). The combination of forecasts. *Journal of the Operational Research Society*, 20(4):451–468.
- Bergmeir, C. and Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213.
- Bergmeir, C., Costantini, M., and Benítez, J. M. (2014). On the usefulness of cross-validation for directional forecast evaluation. *Computational Statistics & Data Analysis*, 76:132–143.
- Bergmeir, C., Hyndman, R. J., Koo, B., et al. (2015). A note on the validity of cross-validation for evaluating time series prediction. *Monash University Department of Econometrics and Business Statistics Working Paper*, 10:15.
- Bry, G. and Boschan, C. (1971). Programmed selection of cyclical turning points. In *Cyclical Analysis of Time Series: Selected Procedures and Computer Programs*, pages 7–63. NBER.
- Castle, J. L., Clements, M. P., and Hendry, D. F. (2013). Forecasting by factors, by variables, by both or neither? *Journal of Econometrics*, 177(2):305–319.
- Castle, J. L., Clements, M. P., and Hendry, D. F. (2015). Robust approaches to forecasting. *International Journal of Forecasting*, 31(1):99–112.
- Chan, F. and Pauwels, L. L. (2018). Some theoretical results on forecast combinations. *International Journal of Forecasting*, 34(1):64–74.
- Clements, M. P. and Hendry, D. F. (2001). *Forecasting non-stationary economic time series*. mit Press.

- Cryer, J. D. and Kellet, N. (1991). *Time series analysis*. Springer.
- Diebold, F. X. and Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1):134–144.
- Duchêne, S., Duchêne, D. A., Di Giallonardo, F., Eden, J.-S., Geoghegan, J. L., Holt, K. E., Ho, S. Y., and Holmes, E. C. (2016). Cross-validation to select bayesian hierarchical models in phylogenetics. *BMC evolutionary biology*, 16(1):115.
- Elliott, G. and Timmermann, A. (2008). Economic forecasting. *Journal of Economic Literature*, 46(1):3–56.
- Garcia, M. G., Medeiros, M. C., and Vasconcelos, G. F. (2017). Real-time inflation forecasting with high-dimensional models: The case of brazil. *International Journal of Forecasting*, 33(3):679–693.
- Hamilton, J. D. (1994). *Time series analysis*.
- Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The model confidence set. *Econometrica*, 79(2):453–497.
- Harding, D. and Pagan, A. (2002). Dissecting the cycle: a methodological investigation. *Journal of monetary economics*, 49(2):365–381.
- Hendry, D. F. and Doornik, J. A. (2014). *Empirical model discovery and theory evaluation: automatic selection methods in econometrics*. MIT Press.
- Hsiao, C. and Wan, S. K. (2014). Is there an optimal forecast combination? *Journal of Econometrics*, 178:294–309.
- Hyndman, R. and Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software, Articles*, 27(3):1–22.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.
- Jung, A., Hannak, G., and Goertz, N. (2015). Graphical lasso based model selection for time series. *IEEE Signal Processing Letters*, 22(10):1781–1785.
- Meese, R. A. and Rogoff, K. (1983). Do they fit out of sample? *Journal of international economics*, 14:3–24.
- Racine, J. (2000). Consistent cross-validators model-selection for dependent data: hv-block cross-validation. *Journal of econometrics*, 99(1):39–61.

- Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J. J., Schröder, B., Thuiller, W., et al. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40(8):913–929.
- Rossi, B. (2013). Exchange rate predictability. *Journal of economic literature*, 51(4):1063–1119.
- Stock, J. H. and Watson, M. W. (2002). Macroeconomic forecasting using diffusion indexes. *Journal of Business and Economic Statistics*, 20(2):147–162.
- Timmermann, A. (2006). Forecast combinations. *Handbook of economic forecasting*, 1:135–196.
- Tong, H. (1990). *Non-linear time series: a dynamical system approach*. Oxford University Press.

6 Tables and Appendix

Table 1: Models with the lowest mean absolute error for 1, 2, 3, and 12 steps ahead

	Brazil	Canada	Belgium	Germany	USA	Portugal	UK
1 Step Ahead	<i>CvML</i>	<i>CvML</i>	<i>CvML</i>	<i>SARIMA</i>	<i>TAR</i>	<i>SARIMA</i>	<i>CvML</i>
2 Steps Ahead	<i>Seasonal</i>	<i>CvML</i>	<i>CvML</i>	<i>TAR</i>	<i>CvML</i>	<i>TAR</i>	<i>CvML</i>
3 Steps Ahead	<i>Seasonal</i>	<i>CvML</i>	<i>ETL</i>	<i>CvML</i>	<i>CvML</i>	<i>ETL</i>	<i>CvML</i>
12 Steps Ahead	Naive	<i>CvML</i>	<i>CvML</i>	<i>CvML</i>	<i>CvML</i>	<i>CvML</i>	<i>CvML</i>

Table 2: Models with the lowest mean square error for 1, 2, 3, and 12 steps ahead

	Brazil	Canada	Belgium	Germany	USA	Portugal	UK
1 Step Ahead	<i>CvML</i>	<i>CvML</i>	<i>CvML</i>	<i>ETL</i>	<i>TAR</i>	<i>ANN</i>	<i>ETL</i>
2 Steps Ahead	Seasonal	<i>CvML</i>	<i>CvML</i>	<i>GB</i>	<i>GB</i>	<i>TAR</i>	<i>CvML</i>
3 Steps Ahead	Seasonal	<i>CvML</i>	<i>ETL</i>	<i>ETL</i>	<i>GB</i>	<i>ETL</i>	<i>CvML</i>
12 Steps Ahead	Naive	<i>CvML</i>	<i>CvML</i>	<i>CvML</i>	<i>CvML</i>	<i>CvML</i>	<i>ETL</i>

Table 3: DM test p-values

Belgium					
	Average Combination	MCS	Granger Bates	Naive	Seasonal Naive
1 Step Ahead	0.000	4.423×10^{-1}	1.000×10^{-5}	0.000	3.000×10^{-5}
2 Steps Ahead	1.141×10^{-2}	1.310×10^{-3}	1.129×10^{-2}	4.400×10^{-4}	6.700×10^{-4}
3 Steps Ahead	2.673×10^{-1}	6.924×10^{-2}	3.229×10^{-1}	3.900×10^{-4}	9.800×10^{-4}
12 Steps Ahead	6.460×10^{-3}	1.597×10^{-2}	6.370×10^{-3}	1.610×10^{-3}	1.610×10^{-3}
Canada					
	Average Combination	MCS	Granger Bates	Naive	Seasonal Naive
1 Step Ahead	5.040×10^{-3}	4.920×10^{-3}	5.300×10^{-3}	1.420×10^{-3}	0.000
2 Steps Ahead	4.549×10^{-2}	1.190×10^{-3}	4.653×10^{-2}	1.170×10^{-3}	3.000×10^{-5}
3 Steps Ahead	5.972×10^{-2}	1.208×10^{-2}	6.190×10^{-2}	1.200×10^{-4}	5.000×10^{-5}
12 Steps Ahead	8.902×10^{-2}	8.615×10^{-2}	8.755×10^{-2}	1.000×10^{-5}	1.000×10^{-5}
Brazil					
	Average Combination	MCS	Granger Bates	Naive	Seasonal Naive
1 Step Ahead	1.800×10^{-4}	2.468×10^{-1}	1.595×10^{-2}	0.000	1.000×10^{-5}
2 Steps Ahead	4.000×10^{-5}	0.000	1.400×10^{-4}	0.000	8.000×10^{-5}
3 Steps Ahead	7.587×10^{-1}	1.200×10^{-3}	9.920×10^{-1}	0.000	1.400×10^{-4}
12 Steps Ahead	2.000×10^{-5}	0.000	9.000×10^{-5}	7.000×10^{-5}	7.000×10^{-5}
Germany					
	Average Combination	MCS	Granger Bates	Naive	Seasonal Naive
1 Step Ahead	8.868×10^{-1}	5.191×10^{-1}	8.620×10^{-1}	4.170×10^{-3}	3.000×10^{-4}
2 Steps Ahead	4.893×10^{-1}	0.000	4.512×10^{-1}	4.510×10^{-1}	1.333×10^{-2}
3 Steps Ahead	2.412×10^{-1}	1.287×10^{-2}	2.544×10^{-1}	1.636×10^{-1}	9.390×10^{-3}
12 Steps Ahead	3.144×10^{-1}	3.000×10^{-5}	3.046×10^{-1}	2.799×10^{-2}	2.799×10^{-2}
Portugal					
	Average Combination	MCS	Granger Bates	Naive	Seasonal Naive
1 Step Ahead	8.488×10^{-1}	7.418×10^{-2}	7.817×10^{-1}	1.000×10^{-5}	1.000×10^{-4}
2 Steps Ahead	8.595×10^{-1}	0.000	7.626×10^{-1}	6.000×10^{-4}	2.900×10^{-4}
3 Steps Ahead	3.368×10^{-1}	1.926×10^{-1}	3.616×10^{-1}	7.680×10^{-3}	3.900×10^{-4}
12 Steps Ahead	7.827×10^{-2}	1.000×10^{-4}	9.328×10^{-2}	3.820×10^{-3}	3.820×10^{-3}
UK					
	Average Combination	MCS	Granger Bates	Naive	Seasonal Naive
1 Step Ahead	1.644×10^{-2}	1.556×10^{-1}	1.679×10^{-2}	2.100×10^{-4}	1.000×10^{-5}
2 Steps Ahead	4.052×10^{-2}	3.260×10^{-3}	4.389×10^{-2}	2.687×10^{-2}	7.300×10^{-4}
3 Steps Ahead	4.800×10^{-3}	3.100×10^{-4}	5.310×10^{-3}	1.591×10^{-2}	3.300×10^{-4}
12 Steps Ahead	9.054×10^{-1}	3.329×10^{-1}	9.261×10^{-1}	4.800×10^{-4}	4.800×10^{-4}
USA					
	Average Combination	MCS	Granger Bates	Naive	Seasonal Naive
1 Step Ahead	9.317×10^{-1}	3.708×10^{-2}	9.524×10^{-1}	9.552×10^{-2}	5.200×10^{-4}
2 Steps Ahead	6.702×10^{-1}	1.178×10^{-1}	6.686×10^{-1}	5.716×10^{-2}	4.970×10^{-3}
3 Steps Ahead	7.291×10^{-1}	7.378×10^{-1}	7.714×10^{-1}	2.283×10^{-1}	3.540×10^{-3}
12 Steps Ahead	2.217×10^{-1}	4.702×10^{-1}	2.267×10^{-1}	2.830×10^{-3}	2.830×10^{-3}

Note: The table shows p-values of the Diebold Mariano test applied to *CvML* versus each method described in the columns.

Table 4: Percentage of observations marked as Expansions and Contractions by the BB algorithm

Models selected by CvML during economic contraction and expansion							
Method	USA	UK	Portugal	Germany	Brazil	Canada	Belgium
Expansion	5.362×10^{-1}	5.652×10^{-1}	5.290×10^{-1}	6.087×10^{-1}	3.986×10^{-1}	6.522×10^{-1}	2.246×10^{-1}
Contraction	4.638×10^{-1}	4.348×10^{-1}	4.710×10^{-1}	3.913×10^{-1}	6.015×10^{-1}	3.478×10^{-1}	7.754×10^{-1}

Table 5: Unconditional forecast performance analysis

		<i>Dependent variable: CvML Errors</i>			
		1 Step	2 Step	3 Step	4 Step
Belgium	Dummy cycle	2.053×10^{-3} (4.521×10^{-1})	1.758×10^{-3} (5.999×10^{-1})	2.857×10^{-3} (4.448×10^{-1})	3.276×10^{-3} (2.169×10^{-1})
	Constant	-8.780×10^{-4} (6.655×10^{-1})	-1.880×10^{-4} (9.184×10^{-1})	-4.580×10^{-4} (7.997×10^{-1})	-1.257×10^{-3} (3.711×10^{-1})
Canada	Dummy cycle	1.339×10^{-3} (4.842×10^{-1})	1.879×10^{-3} (3.556×10^{-1})	1.970×10^{-3} (3.147×10^{-1})	1.017×10^{-3} (6.722×10^{-1})
	Constant	-1.703×10^{-3} (3.799×10^{-1})	-2.011×10^{-3} (3.158×10^{-1})	-2.630×10^{-3} (1.488×10^{-1})	-1.473×10^{-3} (5.261×10^{-1})
Brazil	Dummy cycle	1.647×10^{-3} (7.464×10^{-1})	1.686×10^{-2} (1.025×10^{-1})	$2.540 \times 10^{-2***}$ (4.106×10^{-3})	1.762×10^{-2} (1.113×10^{-1})
	Constant	-4.248×10^{-3} (2.420×10^{-1})	$-1.648 \times 10^{-2**}$ (1.821×10^{-2})	$-1.246 \times 10^{-2*}$ (5.263×10^{-2})	$-1.604 \times 10^{-2**}$ (4.393×10^{-2})
Germany	Dummy cycle	4.230×10^{-3} (3.240×10^{-1})	-2.313×10^{-3} (4.822×10^{-1})	5.421×10^{-3} (1.461×10^{-1})	3.984×10^{-3} (2.588×10^{-1})
	Constant	-3.104×10^{-3} (4.699×10^{-1})	3.918×10^{-3} (2.212×10^{-1})	-3.741×10^{-3} (3.091×10^{-1})	-2.961×10^{-3} (3.804×10^{-1})
Portugal	Dummy cycle	3.676×10^{-3} (4.334×10^{-1})	5.507×10^{-3} (1.481×10^{-1})	3.584×10^{-3} (3.877×10^{-1})	1.321×10^{-3} (6.979×10^{-1})
	Constant	-3.208×10^{-3} (2.246×10^{-1})	$-3.835 \times 10^{-3*}$ (6.676×10^{-2})	$-4.990 \times 10^{-3*}$ (5.684×10^{-2})	-5.160×10^{-4} (7.896×10^{-1})
UK	Dummy cycle	$2.403 \times 10^{-3*}$ (6.420×10^{-2})	1.967×10^{-3} (1.493×10^{-1})	$2.063 \times 10^{-3*}$ (8.186×10^{-2})	1.350×10^{-3} (3.628×10^{-1})
	Constant	$-2.524 \times 10^{-3*}$ (5.460×10^{-2})	$-2.053 \times 10^{-3*}$ (5.949×10^{-2})	$-2.554 \times 10^{-3**}$ (1.755×10^{-2})	$-2.272 \times 10^{-3**}$ (9.783×10^{-2})
USA	Dummy cycle	$2.506 \times 10^{-3*}$ (6.583×10^{-2})	$3.246 \times 10^{-3**}$ (3.613×10^{-2})	$3.921 \times 10^{-3**}$ (2.120×10^{-2})	3.366×10^{-3} (1.368×10^{-1})
	Constant	$-1.546 \times 10^{-3*}$ (8.287×10^{-2})	$-2.072 \times 10^{-3**}$ (2.261×10^{-2})	$-2.394 \times 10^{-3**}$ (1.981×10^{-2})	$-2.918 \times 10^{-3**}$ (1.369×10^{-2})

Note 1: Sig. Levels - *** p - value < .01, ** p - value < .05, * p - value < .1.

Note 2: Heteroskedasticity and auto-correlation robust errors - Newey-West.

Table 6: Models selected by CvML

Models selected by CvML during economic contraction								
Method	USA	UK	Portugal	Germany	Brazil	Canada	Belgium	Avg
ANN	1.328×10^{-1}	3.097×10^{-2}	3.543×10^{-2}	8.612×10^{-2}	1.993×10^{-1}	6.742×10^{-2}	5.314×10^{-2}	8.646×10^{-2}
ARIMA	2.188×10^{-1}	2.655×10^{-1}	2.008×10^{-1}	2.345×10^{-1}	1.329×10^{-1}	3.596×10^{-1}	2.754×10^{-1}	2.410×10^{-1}
ETS	2.852×10^{-1}	4.867×10^{-1}	5.512×10^{-1}	3.828×10^{-1}	8.970×10^{-2}	2.584×10^{-1}	5.507×10^{-1}	3.721×10^{-1}
TAR	3.633×10^{-1}	2.168×10^{-1}	2.126×10^{-1}	2.967×10^{-1}	5.781×10^{-1}	3.944×10^{-1}	1.208×10^{-1}	3.118×10^{-1}
Models selected by CvML during economic expansion								
Method	USA	UK	Portugal	Germany	Brazil	Canada	Belgium	Avg
ANN	1.950×10^{-1}	4.487×10^{-2}	5.634×10^{-2}	1.003×10^{-1}	9.545×10^{-2}	1.611×10^{-1}	5.645×10^{-2}	1.014×10^{-1}
ARIMA	2.270×10^{-1}	2.500×10^{-1}	1.479×10^{-1}	7.295×10^{-2}	1.136×10^{-1}	2.861×10^{-1}	3.226×10^{-1}	2.029×10^{-1}
ETS	1.454×10^{-1}	4.038×10^{-1}	6.268×10^{-1}	4.590×10^{-1}	1.273×10^{-1}	1.583×10^{-1}	4.516×10^{-1}	3.389×10^{-1}
TAR	4.326×10^{-1}	3.013×10^{-1}	1.690×10^{-1}	3.678×10^{-1}	6.636×10^{-1}	3.944×10^{-1}	1.694×10^{-1}	3.569×10^{-1}

Note 1: The table shows in each cell the percentage of times each model was selected by the *CvML* method during the machine learning process. This result aggregates all steps.

Table 7: Mean Absolute Error for each method used

Method	Germany	Canada	Brazil	Belgium	Portugal	UK	USA
Arithmetic Mean 1 step	1.295×10^{-2}	8.361×10^{-3}	2.778×10^{-2}	1.851×10^{-2}	2.003×10^{-2}	7.047×10^{-3}	4.078×10^{-3}
Arithmetic Mean 12 steps	1.122×10^{-1}	8.881×10^{-3}	7.108×10^{-2}	1.807×10^{-2}	4.035×10^{-2}	7.026×10^{-3}	5.818×10^{-3}
Arithmetic Mean 2 steps	1.231×10^{-2}	8.660×10^{-3}	7.128×10^{-2}	1.819×10^{-2}	1.908×10^{-2}	6.864×10^{-3}	4.266×10^{-3}
Arithmetic Mean 3 steps	1.412×10^{-2}	8.534×10^{-3}	5.281×10^{-2}	1.967×10^{-2}	2.293×10^{-2}	6.995×10^{-3}	4.492×10^{-3}
Artificial Neural Network 1 step	1.386×10^{-2}	9.211×10^{-3}	2.235×10^{-2}	1.808×10^{-2}	1.977×10^{-2}	8.405×10^{-3}	4.892×10^{-3}
Artificial Neural Network 12 steps	1.512×10^{-2}	9.369×10^{-3}	8.259×10^{-2}	1.985×10^{-2}	2.560×10^{-2}	8.704×10^{-3}	5.840×10^{-3}
Artificial Neural Network 2 steps	1.671×10^{-2}	9.652×10^{-3}	8.216×10^{-2}	2.100×10^{-2}	2.672×10^{-2}	8.819×10^{-3}	4.953×10^{-3}
Artificial Neural Network 3 steps	1.534×10^{-2}	9.216×10^{-3}	6.606×10^{-2}	2.273×10^{-2}	2.651×10^{-2}	8.834×10^{-3}	5.100×10^{-3}
Exponential Smoothing 1 step	1.229×10^{-2}	8.433×10^{-3}	2.667×10^{-2}	1.762×10^{-2}	1.946×10^{-2}	6.647×10^{-3}	4.484×10^{-3}
Exponential Smoothing 12 steps	1.258×10^{-2}	9.177×10^{-3}	7.838×10^{-2}	1.803×10^{-2}	1.920×10^{-2}	6.826×10^{-3}	6.394×10^{-3}
Exponential Smoothing 2 steps	1.237×10^{-2}	8.683×10^{-3}	7.768×10^{-2}	1.757×10^{-2}	1.952×10^{-2}	6.693×10^{-3}	4.506×10^{-3}
Exponential Smoothing 3 steps	1.243×10^{-2}	8.685×10^{-3}	5.779×10^{-2}	1.759×10^{-2}	1.937×10^{-2}	6.699×10^{-3}	4.677×10^{-3}
Granger Bates 1 step	1.276×10^{-2}	8.315×10^{-3}	2.366×10^{-2}	1.788×10^{-2}	1.934×10^{-2}	6.931×10^{-3}	4.012×10^{-3}
Granger Bates 12 steps	1.226×10^{-2}	8.853×10^{-3}	6.900×10^{-2}	1.798×10^{-2}	2.490×10^{-2}	6.968×10^{-3}	5.769×10^{-3}
Granger Bates 2 steps	1.188×10^{-2}	8.621×10^{-3}	6.893×10^{-2}	1.796×10^{-2}	1.653×10^{-2}	6.752×10^{-3}	4.225×10^{-3}
Granger Bates 3 steps	1.376×10^{-2}	8.507×10^{-3}	5.236×10^{-2}	1.940×10^{-2}	2.255×10^{-2}	6.912×10^{-3}	4.488×10^{-3}
CvML 1 step	1.309×10^{-2}	7.639×10^{-3}	2.094×10^{-2}	1.567×10^{-2}	2.011×10^{-2}	6.643×10^{-3}	3.945×10^{-3}
CvML 12 steps	1.217×10^{-2}	8.198×10^{-3}	5.685×10^{-2}	1.669×10^{-2}	1.676×10^{-2}	6.719×10^{-3}	5.021×10^{-3}
CvML 2 steps	1.203×10^{-2}	8.084×10^{-3}	5.485×10^{-2}	1.678×10^{-2}	1.880×10^{-2}	6.483×10^{-3}	4.115×10^{-3}
CvML 3 steps	1.222×10^{-2}	8.212×10^{-3}	5.355×10^{-2}	1.829×10^{-2}	1.999×10^{-2}	6.525×10^{-3}	4.401×10^{-3}
MCS 1 step	1.245×10^{-2}	8.570×10^{-3}	2.191×10^{-2}	1.662×10^{-2}	1.820×10^{-2}	7.190×10^{-3}	4.458×10^{-3}
MCS 12 steps	1.430×10^{-2}	9.130×10^{-3}	7.957×10^{-2}	1.876×10^{-2}	2.391×10^{-2}	7.743×10^{-3}	5.844×10^{-3}
MCS 2 steps	1.671×10^{-2}	9.652×10^{-3}	8.216×10^{-2}	2.100×10^{-2}	2.672×10^{-2}	8.819×10^{-3}	4.953×10^{-3}
MCS 3 steps	1.471×10^{-2}	8.811×10^{-3}	6.014×10^{-2}	2.124×10^{-2}	2.522×10^{-2}	7.780×10^{-3}	4.654×10^{-3}
Naive 1 step	1.858×10^{-2}	1.041×10^{-2}	8.438×10^{-2}	2.962×10^{-2}	3.206×10^{-2}	9.986×10^{-3}	5.729×10^{-3}
Naive 12 steps	1.867×10^{-2}	1.266×10^{-2}	3.210×10^{-2}	2.613×10^{-2}	2.707×10^{-2}	1.050×10^{-2}	7.667×10^{-3}
Naive 2 steps	1.570×10^{-2}	1.061×10^{-2}	6.534×10^{-2}	2.529×10^{-2}	2.835×10^{-2}	9.479×10^{-3}	5.662×10^{-3}
Naive 3 steps	1.656×10^{-2}	1.130×10^{-2}	7.434×10^{-2}	2.513×10^{-2}	2.566×10^{-2}	9.306×10^{-3}	5.668×10^{-3}
MGB 1 step	1.296×10^{-2}	8.349×10^{-3}	2.435×10^{-2}	1.837×10^{-2}	2.000×10^{-2}	7.033×10^{-3}	4.079×10^{-3}
MGB 12 steps	1.926×10^{-2}	8.887×10^{-3}	6.894×10^{-2}	1.806×10^{-2}	3.076×10^{-2}	6.996×10^{-3}	5.804×10^{-3}
MGB 2 steps	1.223×10^{-2}	8.648×10^{-3}	6.892×10^{-2}	1.810×10^{-2}	1.900×10^{-2}	6.781×10^{-3}	4.266×10^{-3}
MGB 3 steps	1.412×10^{-2}	8.523×10^{-3}	5.234×10^{-2}	1.950×10^{-2}	2.273×10^{-2}	6.951×10^{-3}	4.510×10^{-3}
SARIMA 1 step	1.220×10^{-2}	8.188×10^{-3}	2.679×10^{-2}	1.606×10^{-2}	1.797×10^{-2}	6.695×10^{-3}	4.491×10^{-3}
SARIMA 12 steps	1.451×10^{-2}	9.198×10^{-3}	7.730×10^{-2}	1.832×10^{-2}	2.301×10^{-2}	7.456×10^{-3}	6.193×10^{-3}
SARIMA 2 steps	1.461×10^{-2}	8.786×10^{-3}	7.784×10^{-2}	2.005×10^{-2}	2.393×10^{-2}	7.134×10^{-3}	4.364×10^{-3}
SARIMA 3 steps	1.476×10^{-2}	8.653×10^{-3}	5.568×10^{-2}	2.037×10^{-2}	2.430×10^{-2}	7.369×10^{-3}	4.593×10^{-3}
Seasonal Naive 1 step	1.906×10^{-2}	1.242×10^{-2}	3.088×10^{-2}	2.600×10^{-2}	2.898×10^{-2}	1.047×10^{-2}	7.513×10^{-3}
Seasonal Naive 12 steps	1.867×10^{-2}	1.266×10^{-2}	3.210×10^{-2}	2.613×10^{-2}	2.707×10^{-2}	1.050×10^{-2}	7.667×10^{-3}
Seasonal Naive 2 steps	1.900×10^{-2}	1.247×10^{-2}	3.108×10^{-2}	2.606×10^{-2}	2.898×10^{-2}	1.051×10^{-2}	7.527×10^{-3}
Seasonal Naive 3 steps	1.898×10^{-2}	1.243×10^{-2}	3.127×10^{-2}	2.582×10^{-2}	2.868×10^{-2}	1.053×10^{-2}	7.554×10^{-3}
TAR 1 step	1.619×10^{-2}	8.542×10^{-3}	6.264×10^{-2}	2.580×10^{-2}	2.768×10^{-2}	8.420×10^{-3}	3.517×10^{-3}
TAR 12 steps	4.200×10^{-1}	8.887×10^{-3}	5.685×10^{-2}	2.006×10^{-2}	1.050×10^{-1}	7.264×10^{-3}	5.502×10^{-3}
TAR 2 steps	1.090×10^{-2}	8.584×10^{-3}	5.485×10^{-2}	1.685×10^{-2}	1.205×10^{-2}	6.764×10^{-3}	4.433×10^{-3}
TAR 3 steps	1.647×10^{-2}	8.621×10^{-3}	5.722×10^{-2}	2.089×10^{-2}	2.538×10^{-2}	7.184×10^{-3}	4.751×10^{-3}

Table 8: Mean Square Error for each method used

Method	Germany	Canada	Brazil	Belgium	Portugal	UK	USA
Arithmetic Mean 1 step	3.470×10^{-4}	1.090×10^{-4}	1.290×10^{-3}	5.780×10^{-4}	6.430×10^{-4}	1.000×10^{-4}	1.000×10^{-4}
Arithmetic Mean 12 steps	3.790×10^{-1}	1.270×10^{-4}	6.929×10^{-3}	5.590×10^{-4}	1.600×10^{-2}	9.600×10^{-5}	9.600×10^{-5}
Arithmetic Mean 2 steps	2.520×10^{-4}	1.200×10^{-4}	7.066×10^{-3}	5.800×10^{-4}	5.780×10^{-4}	9.700×10^{-5}	9.700×10^{-5}
Arithmetic Mean 3 steps	5.040×10^{-4}	1.190×10^{-4}	5.028×10^{-3}	6.390×10^{-4}	8.550×10^{-4}	1.000×10^{-4}	1.000×10^{-4}
Artificial Neural Network 1 step	3.270×10^{-4}	1.060×10^{-4}	1.183×10^{-3}	4.560×10^{-4}	5.130×10^{-4}	9.200×10^{-5}	9.200×10^{-5}
Artificial Neural Network 12 steps	3.770×10^{-4}	1.350×10^{-4}	8.101×10^{-3}	5.570×10^{-4}	8.470×10^{-4}	9.800×10^{-5}	9.800×10^{-5}
Artificial Neural Network 2 steps	4.120×10^{-4}	1.230×10^{-4}	8.270×10^{-3}	6.950×10^{-4}	8.910×10^{-4}	1.030×10^{-4}	1.030×10^{-4}
Artificial Neural Network 3 steps	4.040×10^{-4}	1.230×10^{-4}	5.399×10^{-3}	6.550×10^{-4}	8.950×10^{-4}	1.020×10^{-4}	1.020×10^{-4}
Exponential Smoothing 1 step	2.860×10^{-4}	1.140×10^{-4}	1.047×10^{-3}	5.210×10^{-4}	6.220×10^{-4}	8.700×10^{-5}	8.700×10^{-5}
Exponential Smoothing 12 steps	3.010×10^{-4}	1.330×10^{-4}	8.392×10^{-3}	5.430×10^{-4}	5.930×10^{-4}	9.200×10^{-5}	9.200×10^{-5}
Exponential Smoothing 2 steps	2.880×10^{-4}	1.200×10^{-4}	8.426×10^{-3}	5.200×10^{-4}	6.250×10^{-4}	8.800×10^{-5}	8.800×10^{-5}
Exponential Smoothing 3 steps	2.910×10^{-4}	1.210×10^{-4}	5.542×10^{-3}	5.220×10^{-4}	6.170×10^{-4}	8.800×10^{-5}	8.800×10^{-5}
Granger Bates 1 step	3.350×10^{-4}	1.080×10^{-4}	9.360×10^{-4}	5.430×10^{-4}	5.940×10^{-4}	9.800×10^{-5}	9.800×10^{-5}
Granger Bates 12 steps	2.560×10^{-4}	1.270×10^{-4}	6.599×10^{-3}	5.540×10^{-4}	2.558×10^{-3}	9.500×10^{-5}	9.500×10^{-5}
Granger Bates 2 steps	2.320×10^{-4}	1.190×10^{-4}	6.673×10^{-3}	5.700×10^{-4}	4.500×10^{-4}	9.600×10^{-5}	9.600×10^{-5}
Granger Bates 3 steps	4.340×10^{-4}	1.180×10^{-4}	4.976×10^{-3}	6.260×10^{-4}	8.280×10^{-4}	9.800×10^{-5}	9.800×10^{-5}
CvML 1 step	3.410×10^{-4}	9.700×10^{-5}	7.260×10^{-4}	4.470×10^{-4}	6.560×10^{-4}	8.900×10^{-5}	8.900×10^{-5}
CvML 12 steps	2.620×10^{-4}	1.090×10^{-4}	5.004×10^{-3}	4.810×10^{-4}	4.640×10^{-4}	9.600×10^{-5}	9.600×10^{-5}
CvML 2 steps	2.950×10^{-4}	1.090×10^{-4}	4.706×10^{-3}	5.030×10^{-4}	5.840×10^{-4}	8.300×10^{-5}	8.300×10^{-5}
CvML 3 steps	2.990×10^{-4}	1.080×10^{-4}	4.953×10^{-3}	5.920×10^{-4}	7.350×10^{-4}	8.800×10^{-5}	8.800×10^{-5}
MCS 1 step	3.160×10^{-4}	1.130×10^{-4}	8.300×10^{-4}	4.760×10^{-4}	5.190×10^{-4}	1.010×10^{-4}	1.010×10^{-4}
MCS 12 steps	3.700×10^{-4}	1.360×10^{-4}	8.409×10^{-3}	5.690×10^{-4}	8.770×10^{-4}	1.040×10^{-4}	1.040×10^{-4}
MCS 2 steps	5.000×10^{-4}	1.510×10^{-4}	9.749×10^{-3}	7.520×10^{-4}	1.129×10^{-3}	1.440×10^{-4}	1.440×10^{-4}
MCS 3 steps	3.910×10^{-4}	1.280×10^{-4}	6.110×10^{-3}	7.080×10^{-4}	9.270×10^{-4}	1.140×10^{-4}	1.140×10^{-4}
Naive 1 step	5.750×10^{-4}	1.750×10^{-4}	1.030×10^{-2}	1.415×10^{-3}	1.725×10^{-3}	1.790×10^{-4}	1.790×10^{-4}
Naive 12 steps	6.520×10^{-4}	2.530×10^{-4}	1.717×10^{-3}	1.169×10^{-3}	1.179×10^{-3}	2.020×10^{-4}	2.020×10^{-4}
Naive 2 steps	4.150×10^{-4}	1.930×10^{-4}	6.891×10^{-3}	9.790×10^{-4}	1.217×10^{-3}	1.710×10^{-4}	1.710×10^{-4}
Naive 3 steps	4.580×10^{-4}	1.890×10^{-4}	9.707×10^{-3}	9.730×10^{-4}	1.046×10^{-3}	1.680×10^{-4}	1.680×10^{-4}
MGB 1 step	3.490×10^{-4}	1.080×10^{-4}	9.820×10^{-4}	5.700×10^{-4}	6.370×10^{-4}	1.000×10^{-4}	1.000×10^{-4}
MGB 12 steps	2.572×10^{-3}	1.270×10^{-4}	6.585×10^{-3}	5.590×10^{-4}	5.751×10^{-3}	9.600×10^{-5}	9.600×10^{-5}
MGB 2 steps	2.470×10^{-4}	1.200×10^{-4}	6.641×10^{-3}	5.760×10^{-4}	5.740×10^{-4}	9.600×10^{-5}	9.600×10^{-5}
MGB 3 steps	5.220×10^{-4}	1.180×10^{-4}	4.951×10^{-3}	6.320×10^{-4}	8.470×10^{-4}	9.800×10^{-5}	9.800×10^{-5}
SARIMA 1 step	3.500×10^{-4}	1.300×10^{-4}	8.340×10^{-4}	5.360×10^{-4}	6.050×10^{-4}	1.260×10^{-4}	1.260×10^{-4}
SARIMA 12 steps	4.080×10^{-4}	1.470×10^{-4}	9.069×10^{-3}	6.230×10^{-4}	9.890×10^{-4}	1.270×10^{-4}	1.270×10^{-4}
SARIMA 2 steps	5.000×10^{-4}	1.510×10^{-4}	9.749×10^{-3}	7.520×10^{-4}	1.129×10^{-3}	1.440×10^{-4}	1.440×10^{-4}
SARIMA 3 steps	4.220×10^{-4}	1.420×10^{-4}	7.183×10^{-3}	8.020×10^{-4}	1.038×10^{-3}	1.430×10^{-4}	1.430×10^{-4}
Seasonal Naive 1 step	6.490×10^{-4}	2.430×10^{-4}	1.626×10^{-3}	1.154×10^{-3}	1.360×10^{-3}	2.000×10^{-4}	2.000×10^{-4}
Seasonal Naive 12 steps	6.520×10^{-4}	2.530×10^{-4}	1.717×10^{-3}	1.169×10^{-3}	1.179×10^{-3}	2.020×10^{-4}	2.020×10^{-4}
Seasonal Naive 2 steps	6.490×10^{-4}	2.450×10^{-4}	1.638×10^{-3}	1.160×10^{-3}	1.364×10^{-3}	2.010×10^{-4}	2.010×10^{-4}
Seasonal Naive 3 steps	6.500×10^{-4}	2.440×10^{-4}	1.649×10^{-3}	1.144×10^{-3}	1.339×10^{-3}	2.020×10^{-4}	2.020×10^{-4}
TAR 1 step	5.910×10^{-4}	1.180×10^{-4}	5.974×10^{-3}	1.116×10^{-3}	1.385×10^{-3}	1.440×10^{-4}	1.440×10^{-4}
TAR 12 steps	6.198	1.280×10^{-4}	5.004×10^{-3}	7.410×10^{-4}	2.692×10^{-1}	1.120×10^{-4}	1.120×10^{-4}
TAR 2 steps	2.460×10^{-4}	1.190×10^{-4}	4.706×10^{-3}	5.860×10^{-4}	3.050×10^{-4}	9.800×10^{-5}	9.800×10^{-5}
TAR 3 steps	1.552×10^{-3}	1.210×10^{-4}	5.026×10^{-3}	7.920×10^{-4}	1.412×10^{-3}	1.090×10^{-4}	1.090×10^{-4}