Pesquisa Operacional (2021) 41: e245564 p.1-27
doi: 10.1590/0101-7438.2021.041.00245564
© 2021 Brazilian Operations Research Society
Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
ARTICLES

HORIZON-OPTIMIZED WEIGHTS FOR FORECAST COMBINATION WITH CROSS-LEARNING

Rafael de O. Valle dos Santos[1,2*], Celso F. Araujo F.[2],
Ricardo M. S. Accioly[3] and Fernando Luiz Cyrino Oliveira[4]

Received November 13, 2020 / Accepted September 19, 2021

ABSTRACT. Recent empirical results show that forecast combinations and cross-learning schemes are winning approaches in the time series field. Although many competition-winning combination methods – with cross-learning or not – use static weights along the forecasting horizon, we could not find extensive work about the effects of using horizon-optimized weights. This paper proposes a forecast combination framework and provides a considerably sizeable empirical investigation into the use of horizon-optimized weights, i.e., weights that may vary over the forecasting horizon. We build on cross-learning, time series clustering and cross-validation to form Horizon-Optimized Convex Combinations (HOC2) of forecasts from five methods: Automated exponential smoothing, Automated ARIMA, Theta, TBATS, and Seasonal naïve. Our combinations were tested with data from the previous M1, M3 and M4 forecast competitions, comprising 104,004 time series with different frequencies and lengths. The results shall be helpful to support future research on how horizon-optimized weights can be used interchangeably with static ones.

Keywords: forecast combinations; convex combinations; cross-learning; time series clustering; cross-validation; M competitions.

1 INTRODUCTION

The time series forecasting field is known to be vast and relevant (Petropoulos et al., 2021), with applications ranging from solar radiation prediction (Teixeira Junior et al., 2015), energy optimization (Oliveira et al., 2015), (Souza et al. 2012), (da Silva, Cyrino Oliveira & Souza, 2019) to optimal planning of intensive medical care medical units (Angelo et al., 2017). Among

*Corresponding author

[1] Industrial Engineering Department, Pontifical Catholic University of Rio de Janeiro, Rua Marquês de São Vicente, 225, Gávea, 22451-900 Rio de Janeiro-RJ, Brazil – E-mail: rvsantos2000@yahoo.com.br; rvsantos@petrobras.com.br – https://orcid.org/0000-0003-4103-6600

[2] Petrobras, Av. República do Chile, 65, Centro, 20031-912 Rio de Janeiro-RJ, Brazil – E-mail: celsoaf@petrobras.com.br – https://orcid.org/0000-0003-3987-8905

[3] Institute of Mathematics and Statistics, State University of Rio de Janeiro, Rua São Francisco Xavier, 524, 6o Andar, Bloco B, Maracanã, 20.550-013 Rio de Janeiro-RJ, Brazil – E-mail: raccioly@ime.uerj.br – https://orcid.org/0000-0001-6513-3443

[4] Industrial Engineering Department, Pontifical Catholic University of Rio de Janeiro, Rua Marquês de São Vicente, 225, Gávea, 22451-900 Rio de Janeiro-RJ, Brazil – E-mail: cyrino@puc-rio.br – https://orcid.org/0000-0003-1870-9440

this rather important topic in statistics and machine learning, this paper focus on the study and development of novel forecast combination techniques.

Forecast combinations have been around for nearly 50 years now, since the seminal work of Bates and Granger (1969). Up-to-date results from the latest M4 competition (Makridakis, Spiliotis & Assimakopoulos, 2020) confirm its practical success: out of the six best results – a cluster of results that were statistically distinguished in the leaderboard – 5 used combinations. Moreover, out of the 17 methods that were better than the benchmarks, 12 used combinations. Makridakis et al. (2020) concluded:

> "The higher numerical accuracy of combining, coupled with the poor performances of pure statistical/ML methods, confirms the findings of the previous three M Competitions, as well as those of other competitions/empirical studies. It implies that no single method can capture the time series patterns adequately, whereas a combination of methods, each of which captures a different component of such patterns, is more accurate because it cancels the errors of the individual models through averaging."

Other than reassuring that combining time series forecasts is a winning approach (as it is less risky than selecting a single method), the M4 results also show that cross-learning – the use of information derived from some cluster of time series to forecast the individual time series – is a novelty from the Machine Learning field that is worth studying, as both 1st and 2nd best-ranked methods used this concept.

When combining forecasts from many models – forming ensembles – the success of the approach will not only rely on the quality of the pool of forecasts being combined (Atya, 2020), but also on the combination weights (Timmermman, 2006). Based on our experience, we here classify the weighting methods in two broad types:

- With no optimization procedure: mean, trimmed mean, median etc. E.g.: Petropoulos and Svetunkov (2020), Jaganathan and Prakash (2020);
- With some sort of (in-sample) optimization. E.g.: Pawlikowski, Chorowska, and Yanchuk (2020), Fiorucci and Louzada (2020); Montero-Manso, Athanasopoulos, Hyndman, and Talagala (2020).

Regardless of their nature, most of the published weighting methods generate and use static weights along the forecast horizon. That is why we decided to provide a considerably extensive empirical investigation into the use of horizon-optimized weights, i.e., weights that may vary over the prediction steps.

We propose a forecast combination framework (Section 2) joining cross-learning, time series clustering and cross-validation. For each time series, it selects the most suitable (in-sample) horizon-optimized weighting matrix and performs an out-of-sample convex combination of forecasts derived from a pre-defined pool of methods.

Our empirical investigation comprises 104,004 time series with several frequencies and lengths, extracted from the previous M1 (Makridakis et al., 1982), M3 (Makridakis & Hibon, 2000) and M4 (Makridakis, Spiliotis & Assimakopoulos, 2020) forecast competitions.

It is worth saying that many attempts to determine optimal combination weights have ended up worse than simply using static-equal weights (i.e., simple average) – that fact has become known as the "forecast combination puzzle" (Smith & Wallis, 2009). In other words, the simple (arithmetic) average of forecasts is often "hard to beat" (Timmermman, 2006). Nevertheless, recent results and discussions in the field point out that more sophisticated combination schemes, eventually applying new concepts like cross-learning and up-to-date hybrid methods, may lead to better forecasting accuracies (Makridakis, Spiliotis & Assimakopoulos, 2020; Fry & Brundage, 2020). The use of horizon-optimized weights may also benefit from such advances.

The remainder of the paper is organized as follows. Section 2 describes the proposed methodology. Section 3 presents the setup of experiments and test results using the M Competitions datasets. Section 4 concludes the work with final comments and suggestions for future research.

2 METHODOLOGY

The ensemble framework proposed here – Horizon-Optimized Convex Combinations (HOC2) – is based on the convex combination of forecasts (Section 2.1), with optimized weights that may vary over the forecasting horizon.

In practice, there is no guarantee that horizon-optimized weights outperform static-equal ones, but "some time-variation or adaptive adjustment in the combination weights (or perhaps in the underlying models being combined) can often improve forecasting performance" (Timmermann, 2006). In Valle dos Santos and Vellasco (2015) the authors tested a horizon-optimized weights approach with data from the late NN-3 competition (Crone, Hibon & Nikolopoulos, 2011), achieving good results, but for a minimal number of series.

For each time series being analyzed, HOC2 strategies firstly rely on the individual forecasts produced by the pre-defined pool of methods – in our case, five forecasting methods implemented in the R programming language (Section 2.2).

After that, comes the idea of cross-learning, which is a standard machine learning procedure but a trending topic in the time series field: to use information derived from a large dataset of time series to aid the forecasting of the individual time series. The name "cross-learning" was recently used in the context of time series by Smyl (2020) and then by Makridakis, Spiliotis, and Assimakopoulos (2020), but a synonym could be "global models" (as opposed to "local models"), as used by Fry and Brundage (2020). Bandara, Bergmeir, and Smyl (2020) applied the same idea, using the term "cross-series information".

Cross-learning presumes two significant steps: (i) a training (learning) phase, when information is learned from a set of (training) time series, and (ii) a test phase, when each individual series (in the whole set of available time series) is forecasted, not only using its own in-sample information, but also the (in-sample) information learned from the training phase. It is desirable that the time series in a cross-learning process are somewhat similar, in the sense that they can learn meaningful information from each other. That is why most cross-learning frameworks present some sort of time series clustering (aggregation) before its training phase – at least, the series are aggregated by frequency: yearly, quarterly, monthly, etc. Bandara, Bergmeir, and Smyl (2020) explore the concept of time series clustering before the training phase.

The HOC2 framework performs a cross-learning process with a previous clustering step (Section 2.3). In the training phase (Section 2.4), we define a cluster-wise training set, i.e., a set of (training) time series that evenly represents the predefined clusters. As it will be seen, the training set is a variable portion of the whole time series dataset. The learning process consists of a cross-validation procedure that considers in-sample predictions from the established pool of methods to determine horizon-optimized weights for each time series in the training set. As a final step in this learning phase, a mean weighting matrix (Section 2.1) for each cluster is calculated. Moving on to the test phase (Section 2.5), for each time series being analyzed, the framework establishes the most suitable weighting matrix learned from the training phase, by means of some selection/inference strategy, e.g.: if the time series was in the training set, uses its optimized convex weights; if not, use the mean weighting matrix for the series' cluster (which is also convex). Each time, the selected weighting matrix is used to perform the convex combination (weighted average) of the individual forecasts available, and the results are measured by performance metrics.

The main components of the framework are summarized in Figure 1. Sections 2.2 to 2.5 bring further details.

2.1 Convex combinations

The convex combination of K forecasts at time t+h, estimated using the available data at time t:

$$y^{C}_{t+h|t} = \sum_{k=1}^{K} \widehat{w}_{t+h|t,k}\widehat{y}_{t+h|t,k}, \tag{1}$$

subject to

$$\sum_{k=1}^{K} \widehat{w}_{t+h|t,k} = 1 \quad \text{and} \quad \widehat{w}_{t+h|t,k}{\geq}0, \tag{2}$$

where $\widehat{y}_{t+h|t,k}$ is the k-th forecast being combined and $\widehat{w}_{t+h|t,k}$ its respective weight. The constraints in Equation (2) turn the unconstrained linear combination in Equation 1 into a (constrained) convex combination. Convex combinations have great practical interest for two reasons: (i) guarantee that the combined forecast is unbiased if the underlying forecasts are unbiased and (ii) make weight interpretations straightforward, as weights can be seen as ordinary

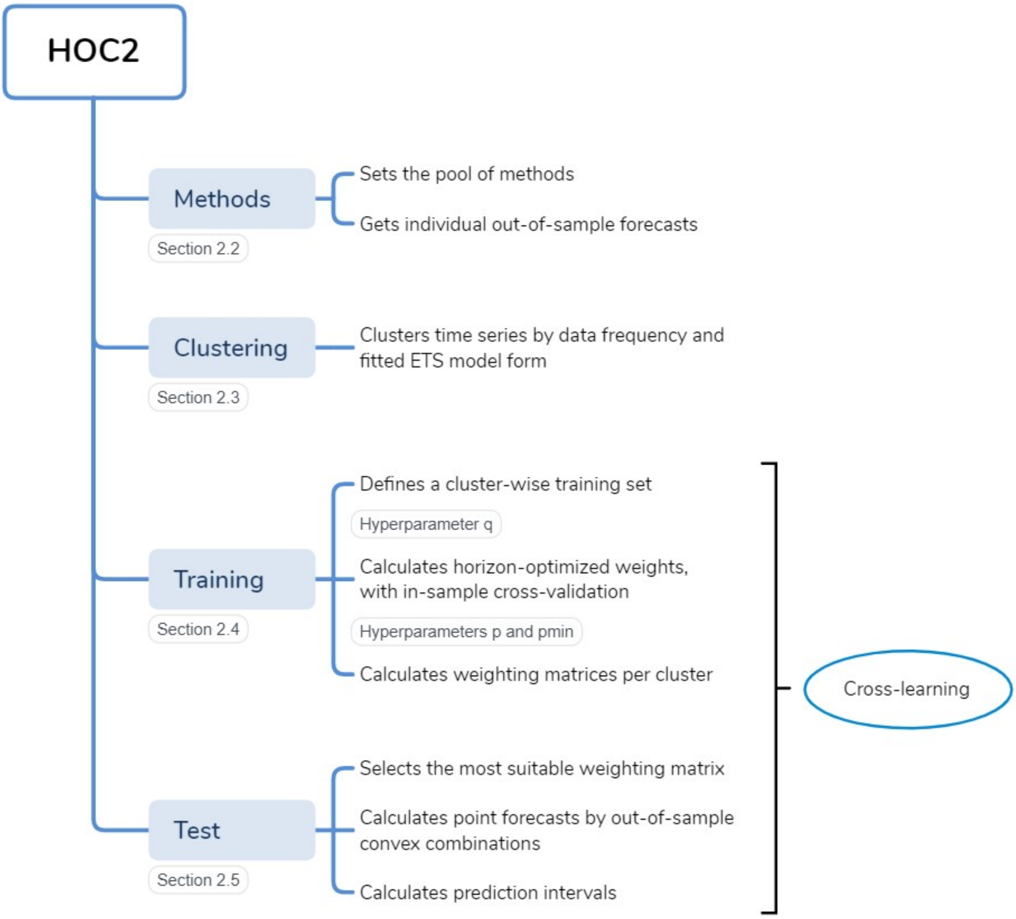
HOC2
Methods
Section 2.2
Sets the pool of methods
Gets individual out-of-sample forecasts
Clustering
Section 2.3
Clusters time series by data frequency and fitted ETS model form
Training
Section 2.4
Defines a cluster-wise training set
Hyperparameter q
Calculates horizon-optimized weights, with in-sample cross-validation
Hyperparameters p and pmin
Calculates weighting matrices per cluster
Test
Section 2.5
Selects the most suitable weighting matrix
Calculates point forecasts by out-of-sample convex combinations
Calculates prediction intervals
Cross-learning

Figure 1 – HOC2 framework conceptual diagram.

percentages (Valle dos Santos & Vellasco, 2015; Timmermman, 2006; Diebold, 1988; Granger & Ramanathan, 1984).

Given a pool of K methods, the set of convex weights estimated at time t for the forecasting horizon t+1, t+2, ..., t+H can be organized as a weighting matrix $\widehat{w}_t^{H,K}$ of size H x K**,** where each line-vector sums to one and has positive components:

$$\widehat{w}_t^{H,K} = \begin{bmatrix} \widehat{w}_{t+1|t,1} & \widehat{w}_{t+1|t,2} & \cdots & \widehat{w}_{t+1|t,K} \\ \widehat{w}_{t+2|t,1} & \widehat{w}_{t+2|t,2} & \cdots & \widehat{w}_{t+2|t,K} \\ \vdots & \vdots & \cdots & \vdots \\ \widehat{w}_{t+H|t,1} & \widehat{w}_{t+H|t,2} & \cdots & \widehat{w}_{t+H|t,K} \end{bmatrix} \tag{3}$$

In the most general case, when the weights are horizon-optimized, the weighting (line-) vectors at t+h vary over the forecasting horizon (h = 1, 2, ..., H).

2.2 Pool of methods

The forecasts to be combined rise from the 5 methods listed below, with their respective R programming language functions and bibliographic references given in parentheses. All functions are implemented over the forecast package in R (Hyndman et al., 2018). As an assumption for automation, all methods were used in their default form and no further treatment was applied.

- Automated exponential smoothing – ETS (*ets*) (Hyndman & Khandakar, 2008);
- Automated ARIMA – AutoARIMA (*auto.arima*) (Hyndman & Khandakar, 2008);
- Theta (*thetaf*) (Assimakopoulos & Nikolopoulos, 2000; Hyndman & Billah, 2003);
- TBATS (*tbats*) (De Livera, Hyndman & Snyder, 2011);
- Seasonal naïve (*snaive*): an ARIMA $(0,0,0)(0,1,0)_m$ model where m is the seasonal period.

2.3 Clustering

In the clustering phase, we first aggregate the time series per frequency (yearly, quarterly, monthly, etc.) and then consider an extra level of aggregation: by fitted ETS model form.

As thoroughly discussed by Petropoulos, Hyndman, and Bergmeir (2018) and Meira, Cyrino Oliveira, and Jeon (2021), the output of the *ets* function is a model form consisting of three terms: *E*rror, *T*rend and *S*easonality – abbreviated as ETS. The error term can be additive (A) or multiplicative (M). The trend and seasonal terms can be none (N), additive (A) or multiplicative (M). At the same time, if the trend component exists, this can be damped (d) or not – for example, ETS(M,Ad,N) refers to a model form with multiplicative error, additive damped trend and no seasonality. By default, the *ets* function excludes models with multiplicative trends from the search of an optimal model, and models with multiplicative seasonality must have multiplicative errors in order to avoid numerical instability. As so, by default, there are 15 different possible model forms and, thus, 15 possible time series clusters per frequency.

We established the aggregation of time series per fitted ETS model form as a proxy to clustering time series by its common features. There are 36 classes of features that can be extracted from a time series (Hyndman, Wang, Kang & Talagala, 2018), leading to many possible choices of aggregations (which may be explored in future works).

2.4 Training

Training is the core phase in the HOC2 framework. Its first step is to determine a training (sub)set of time series, by taking a $q\%$ sample out the complete set of time series available for the data frequency being studied. The percentage q, $0 < q \leq 100$, is a framework´s hyperparameter.

In learning processes, it is highly desirable that the training set is representative of the full scope of analysis. On that matter, it is essential that the training set is cluster-wise, i.e., evenly represents the 15 clusters of the previous clustering phase: this is done in a straightforward way, by forming the training set as the union of $q\%$ samples over each cluster´s subset of time series.

For each time series in the training set, the learning process considers a set of in-sample predictions derived from the pool of methods and calculates horizon-optimized weights for the pool´s convex combination. In our case, the optimization procedure is to set the time series weights as the inverse of some pre-defined error function (e.g., symmetric absolute percentage error), based on a rolling origin cross-validation scheme. We use a cross-validation scheme derived from the GROE method proposed by Fiorucci, Pellegrini, Louzada, and Petropoulos (2015). (The GROE method stands for Generalized Rolling Origin Evaluation and is a general process for cross-validation on time series forecasting methods.)

The cross-validation/optimization procedure considers p successive in-sample forecasts for each lag h in the desirable forecasting horizon (h = 1, 2, ..., H). It works as follows: for each time series in the training set, for each method k in the pool of methods (k = 1, 2, ..., K), compute p different in-sample forecasts $\widehat{y}_{t+h|t,k}$ with rolling origins t = T-h, T-h-1, ..., T-h-p, where T is the series´ last in-sample point (in other words, compute $\widehat{y}_{T|T-h,k}$, $\widehat{y}_{T-1|T-h-1,k}, \ldots, \widehat{y}_{T-p|T-h-p,k}$). After that, each out-of-sample weight at point h can be estimated by Equation 4:

$$\widehat{w}_{T+h|T,k} = [p/\sum_{p} G(\widehat{y}_{t+h|t,k})]/S \tag{4}$$

where:

$$G(\widehat{y}_{t+h|t,k}) = 2\frac{\left|y_{t+h} - \widehat{y}_{t+h|t,k}\right|}{(\left|y_{t+h}\right| + \left|\widehat{y}_{t+h|t,k}\right|)}.100 \tag{5}$$

$$S = \sum_{k=1}^{N} [p/\sum_{p} G(\widehat{y}_{t+h|t,k})] \tag{6}$$

$y_{t+h|t,k}$ is the actual value of the series at time t+h and the function $G()$ is set here to be the symmetric Absolute Percentage Error (sAPE), the building block of the sMAPE performance metric (Section 3.1). The denominator S guarantees that the weighting vector at $T+h$ is convex (Section 2.1).

The desirable number of rolling origins, p, is a framework´s hyperparameter. Practically speaking, it is crucial to notice that the training series may not always be long enough for the computation of p different in-sample forecasts for the requested H (maximum horizon length). To deal with cases like that, we may consider another hyperparameter: the minimum number of rolling origins to be accepted, p_{min}. For instance, if p_{min} cannot be reached, the series is considered small, and a simple average (static-equal) weighting matrix may be linked to the series. For performance matters, we observe that this learning procedure may generate up to p x H x K different predictions per training series.

The final step in the training process is to take the mean weighting matrices for each cluster, based on the per-series individual horizon-optimized matrices from the cross-validation procedure. This is a very straightforward computation: for each cluster, its mean weighting matrix is the mean of the weighting matrices of its valid (rolling origins $\geq p_{\text{min}}$) components time series – the mean of all convex matrices is also convex. The clusters' weighting matrices will be useful in the test phase, to infer combining weights for time series that were not in the training set (in the machine learning jargon, inference is also called generalization.)

2.5 Test

The test phase is the final step in the framework, where all time series must generate out-of-sample forecasts and go through performance metrics, both for point forecasts and prediction intervals. It is interesting to notice that, as opposed to the usual (cross-section) regression/classification applications in machine learning, here the test and training sets are not disjoint. On the contrary, the test set encompasses all the available time series, including the ones in the training set, and the training set can also be a 100% sample from the whole set of series. This is so because in the time series context the difference between training and test lies in the fact the training set deals with in-sample forecasts, while the test set deals with out-of-sample forecasts.

For each time series to be tested, different strategies can be set to select or infer its out-of-sample weighting matrix, always based on what was learnt in the training phase. We propose the following unified three-way strategy:

1. If the time series was in the training set and had an acceptable number rolling origins r ($r \geq p_{min}$): use the series own weighting matrix, as optimized in the training phase;
2. If the series was not in the training set or had a limited number of rolling origins r ($1 \leq r < p_{min}$): use the mean weighting matrix for the series cluster;
3. If the series was too small (e.g., $< 2.H$) for reliable calculations (r = 0): use a simple average (static-equal) weighting matrix (in our experiments, this case only happens over the M1 competition dataset).

Once the weighting matrices are defined, the computation of out-of-sample point forecasts is straightforward, by the convex combination of the available forecasts from the pool of methods.

The framework also computes prediction intervals in a rather straightforward way, solely based on the prediction intervals generated by the ETS procedure (Hyndman, Koehler, Ord & Snyder, 2008, p. 22-23 and 88). For each time series, we take the absolute deviations between the ETS forecasts and each of its upper and lower prediction bounds. Both deviations are then applied accordingly to the HOC2 point forecasts, forming its own prediction bounds. To ensure even comparisons, the same prediction-interval procedure is applied to the simple average combination experiments, placed in this work as benchmarks for HOC2 combinations.

3 EMPIRICAL INVESTIGATION

The M competitions – time series forecasting competitions organized by Spyros Makridakis since the early eighties – have been gradually challenging academic researchers, software vendors and business companies as a benchmark to test new models and methods (Hyndman, 2020). As so, we tested the methodology presented in Section 2 with time series data from the three most cited competitions to date: M1 (Makridakis et al., 1982), M3 (Makridakis & Hibon, 2000) and M4 (Makridakis, Spiliotis & Assimakopoulos, 2020) competitions.

Table 1 depicts the complete set of time series used in this paper. It shows the number of series per competition and frequency, and points out the required forecasting horizon (H) in each case. Together, our experiments comprise 104,004 time series.

Table 1 – The complete set of time series used in this paper.

Frequency	*M1-Competition*	*M3-Competition*	*M4-Competition*	*Total*	*H*
yearly	181	645	23,000	23,826	6
quarterly	203	756	24,000	24,959	8
monthly	617	1,428	48,000	50,045	18
weekly	-	-	359	359	13
daily	-	-	4227	4,227	14
hourly	-	-	414	414	48
other	-	174	-	174	8
Total	1,001	3,003	100,000	104,004	-

H is the forecasting horizon.

Throughout the experiments, HOC2 predictions are mostly measured against two benchmarks: (i) the simple average of the forecasts being combined (AVG) and (ii) the individual forecasts alone. The performance comparisons were all carried out following the M4 competition's way: direct accuracy metric confrontation up to the third decimal digit.

3.1 Performance metrics for point forecasts

Two accuracy metrics are used here to score point forecasts (PFs) performances: the symmetric mean absolute percentage error (sMAPE) and the mean absolute scaled error (MASE). Those metrics have been commonly used over recent publications in the field, as discussed by Makridakis, Spiliotis, and Assimakopoulos (2020). The lower they are (and the closer to zero), the better:

$$sMAPE = \frac{2}{h}\sum_{t=n+1}^{n+h} \frac{\left|Y_t - \widehat{Y}_t\right|}{\left(|Y_t| + \left|\widehat{Y}_t\right|\right)}.100 \tag{7}$$

$$MASE = \frac{1}{h}\frac{\sum_{t=n+1}^{n+h}\left|Y_t - \widehat{Y}_t\right|}{\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t - Y_{t-m}|} \tag{8}$$

where Y_t is the true value of the time series at point t, $\hat{Y}_t$ is the forecasted value at point t, h is the forecasting horizon, n is the length of the sample, and m is the time interval between successive observations considered for each data frequency (or the number of periods within a season): 12 for monthly data, 4 for quarterly data, 24 for hourly data, and 1 for yearly, weekly and daily data. The numerator in Equation 8 is scaled by dividing its value with the mean absolute seasonal difference of the series.

It is worth highlighting that sMAPE and MASE are originally computed per time series. Thus, practically speaking, when considering a set of time series – as in the forecast competitions mentioned here – the overall performance of a method will be the average metric calculated over the entire set.

A third valuable metric, introduced by the M4 competition, is the overall weighted average (OWA):

$$OWA = \frac{1}{2}\left(\frac{sMAPE}{sMAPE_b} + \frac{MASE}{MASE_b}\right), \tag{9}$$

where sMAPE$_\text{b}$ and MASE$_\text{b}$ are benchmarking performances from some pre-defined method.

The OWA metric was the official ranking measure for PFs in the M4 competition, and “Näive 2” – a random walk model applied to seasonally adjusted data – was set by the organizers as the benchmarking method for the relative performance computations (Makridakis, Spiliotis & Assimakopoulos, 2020). As stated by the competition team: “(...) if Method X displays a MASE of 1.6 and an sMAPE of 12.5% across the 100,000 series of M4, while Naïve 2 displays a MASE of 1.9 and an sMAPE of 13.7%, the relative MASE and sMAPE of Method X would be equal to 1.6/1.9 = 0.84 and 12.5/13.7 = 0.91, respectively, resulting in an OWA of (0.84 + 0.91)/2 = 0.88, which indicates that, on average, the method examined is about 12% more accurate than Naïve 2, taking into account both MASE and sMAPE. Note that sMAPE and MASE are first estimated for each series by averaging the error computed for each forecasting horizon, then averaged again across all time series to compute the average value for the entire dataset. On the other hand, OWA is computed only once at the end of the evaluation process for the whole sample of series”.

3.2 Performance metric for prediction intervals

The first M competition to (optionally) ask for predictions intervals (PIs) – other than just point forecasts – was the M4 competition. Following this path, we here calculate PIs for all our experiments.

To evaluate PIs performances, we use the mean scaled interval score (MSIS):

$$MSIS = \frac{1}{h}\frac{\sum_{t=n+1}^{n+h}(U_t - L_t) + \frac{2}{\alpha}(L_t - Y_t)I(Y_t < L_t) + \frac{2}{\alpha}(Y_t - U_t)I(Y_t > U_t)}{\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t - Y_{t-m}|}, \tag{10}$$

where U_t and L_t are respectively the upper and lower bounds for the prediction intervals, Y_t are the future observations (true values), α is the significance level (*0.05* here) and I is the indicator function, its value being 1 if its argument is TRUE and 0 otherwise. The denominator in Equation 10 is the same scale element as in Equation 8; n is the length of the training set and m is the number of periods within a season.

The MSIS metric is an average that penalizes the width of the prediction interval and the points there are outside the specified bounds, considering how far they are of the bound. The numerator, called Interval Score (IS), is scaled by dividing its value with the mean absolute seasonal difference of the series.

Like other metrics, MSIS is originally computed per time series. Thus, when considering forecast competitions, the overall performance of a method will be the average metric calculated over the complete set of series. The lower the metric, the better.

3.3 HOC2 hyperparameters

Table 2 shows the frameworks´ hyperparameters (Section 2) used in our experiments, aggregated by competition dataset. It also shows the total number of time series in each (cluster-wise) training set (*Nt*).

Table 2 – User-defined HOC2 hyperparameters.

Dataset	*q%*	*p*	p_{min}	*Nt*
M1	100%	10	5	1,001
M3	100%	10	5	3,003
M4	10%	10	5	10,030

H is the forecasting horizon. *Nt* is the number of time series in the training set.

There are a few observations to be made at this point. First, as discussed in Section 2.4, our framework may produce up to p x H x K different predictions per training series, i.e., p x H x K x Nt predictions per competition dataset. Although these computations may be directly parallelized by time series cluster – as the training set is cluster-wise – they can still be very time-consuming (Section 3.5). That is why we set the training set for the M4 competition to be only 10% (q = 10) of the available data (which still led to more than 2x the number of time series for both M1 and M3 datasets together). This q number was defined after some preliminary tests and brought results that we considered fair to our purposes. Of course, future work may deal with different values of q for the M4 dataset.

Finally, we call attention to that the number 10,030 is not exactly 10% of 100,000 series of the M4 competition. This explained by the effect of rounding in the cluster-wise formation of the training sets.

3.4 Results

Tables 3-5 show the individual datasets results, including HOC2 and selected benchmarks – the simple average of the forecasts being combined (AVG) and the individual forecasts alone. Results are grouped by performance metric: average sMAPE and average MASE for point forecasts (PFs), and average MSIS for prediction intervals (PIs).

Particularly for the M4 dataset, Table 6 shows values for the OWA performance metric (Section 3.1), the official ranking measure for point forecasts in the M4 competition. The table also indicates sMAPE_b and MASE_b for “Naïve 2”, the pre-defined benchmarking method for the OWA computation – the M4´s performance data is available in a spreadsheet file (“Evaluation and Ranks.xlsx”) that can be downloaded from the M competitions repository at GitHub (https://github.com/Mcompetitions/M4-methods).

Our first comment on the results is that, concerning total performances for every experiment and accuracy metric (including OWA), the HOC2 strategy consistently outperform the benchmarks, with only 1 exception out of 10 possible winning positions: MSIS for the M3 dataset, where AVG wins (Table 4). Breaking the overall results by PFs (sMAPE, MASE and OWA metrics) and PIs (MSIS metric), HOC2 has a 7/7 (100%) winning performance for PFs and a 2/3 (67%) winning performance for PIs.

Breaking the results by data frequency, the framework outperforms the benchmarks on 23 out of 32 winning positions for PFs (72%), the 9 exceptions being the following (winning methods are given in parentheses): M1 dataset – MASE (ARIMA) and sMAPE (ARIMA) for yearly data; M3 dataset – MASE for yearly (AVG) and other (ETS) data; M4 dataset – MASE for monthly data (ARIMA), sMAPE for weekly data (TBATS), and sMAPE (TBATS), MASE (ARIMA) and OWA (ARIMA) for hourly data. Concerning PIs, HOC2 has a 7/13 (54%) winning performance, the 6 exceptions being: M3 dataset – MSIS for yearly (AVG) and other (ETS) data; M4 dataset – MSIS for monthly (ETS), weekly (TBATS), daily (TBATS) and hourly (ARIMA) data. Table 7 summarizes the winning methods, showing that HOC2 is the overall less risky approach, particularly when compared to simple average (AVG), which presented only one winning position for PFs and other for PIs (both for yearly series).

Finally, we point out that HOC2 results are especially remarkable for the M4 dataset, not only by the fact that this dataset is much larger than the others, but also due to the training percentage used in HOC2 experiments: just 10% of the total amount of time series (q = 10). Another interesting fact related to the M4 results, the winning OWA results are much aligned with the best results in the original M4 competition, outperforming all the competition´s benchmarks.

3.5 Processing time

The relevant time-consuming phase in the HOC2 framework is the training phase, where we produce up to p x H x K x Nt predictions per competition dataset – p is the number of required rolling origins, H is the maximum forecasting horizon, K is the number of methods in the pool and Nt is the amount of training series in the dataset. Building on the out-of-sample predictions from the pool of methods phase, the framework’s clustering and test phases are somewhat immediate calculations (Section 2).

Table 3 – M1 dataset – average performances.

	Yearly	*Quarterly*	*Monthly*	*Total*
H	6	8	18	-
Nf	181	203	617	1,001
sMAPE				
HOC2	17.974	**15.507**	**14.375**	**15.255**
AVG	18.281	15.718	14.385	15.360
ETS	18.613	17.464	14.971	16.135
ARIMA	**17.230**	17.375	16.026	16.517
THETA	20.174	16.352	16.527	17.151
TBATS	17.418	16.653	15.127	15.851
SNAIVE	22.431	18.944	17.299	18.560
MASE				
HOC2	3.717	**1.540**	**1.027**	**1.617**
AVG	3.790	1.578	1.029	1.640
ETS	3.771	1.657	1.074	1.680
ARIMA	**3.467**	1.706	1.124	1.666
THETA	4.189	1.702	1.091	1.775
TBATS	3.499	1.694	1.117	1.665
SNAIVE	4.893	2.078	1.314	2.116
MSIS				
HOC2	**58.587**	**18.871**	**8.811**	**19.852**
AVG	60.028	19.162	8.882	20.215
ETS	59.784	21.318	9.625	21.066
ARIMA	62.631	24.094	11.333	23.197
THETA	69.263	24.507	9.809	23.540
TBATS	63.970	24.369	11.971	23.887
SNAIVE	87.971	25.082	10.359	27.379

H is the forecasting horizon. *Nf* is the number of time series per data frequency. The scores on the last column – *Total* – are the weighted averages of the previous columns with *Nf* as weight. Best results per frequency and metric are bold-faced.

Table 4 – M3 dataset – average performances.

	Yearly	*Quarterly*	*Monthly*	*Other*	*Total*
H	6	8	18	8	-
Nf	645	756	1428	174	3,003
sMAPE					
HOC2	**15.750**	**9.020**	**13.296**	**4.343**	**12.228**
AVG	15.791	9.054	13.339	4.401	12.269
ETS	17.003	9.684	14.139	4.372	13.067
ARIMA	17.104	10.011	14.904	4.513	13.543
THETA	16.756	9.203	13.856	4.922	12.790
TBATS	17.370	10.223	13.844	4.354	13.140
SNAIVE	17.880	11.065	17.234	6.302	15.186
MASE					
HOC2	2.695	**1.072**	**0.838**	1.824	**1.353**
AVG	**2.686**	1.075	0.840	1.904	1.357
ETS	2.860	1.170	0.865	**1.814**	1.425
ARIMA	2.959	1.189	0.867	1.841	1.454
THETA	2.774	1.117	0.864	2.271	1.419
TBATS	3.127	1.256	0.861	1.848	1.504
SNAIVE	3.172	1.425	1.146	3.089	1.764
MSIS					
HOC2	29.371	**10.158**	**6.276**	13.962	12.659
AVG	**28.907**	10.227	6.289	14.343	**12.605**
ETS	30.616	10.717	6.342	**13.428**	13.068
ARIMA	40.807	12.535	7.052	15.288	16.160
THETA	31.234	10.907	7.195	16.031	13.805
TBATS	44.186	13.502	7.086	14.495	17.099
SNAIVE	39.976	11.906	8.605	21.860	16.942

H is the forecasting horizon. *Nf* is the number of time series per data frequency. The scores on the last column – *Total* – are the weighted averages of the previous columns with *Nf* as weight. Best results per frequency and metric are bold-faced.

Table 5 – M4 dataset – average performances.

	Yearly	*Quarterly*	*Monthly*	*Weekly*	*Daily*	*Hourly*	*Total*
H	6	8	18	13	14	48	-
Nf	23,000	24,000	48,000	359	4,227	414	100,000
sMAPE							
HOC2	**13.715**	**9.926**	**12.590**	8.431	**2.969**	13.106	**11.790**
AVG	13.754	9.997	12.680	8.408	2.971	13.440	11.861
ETS	15.356	10.291	13.525	8.727	3.046	17.307	12.725
ARIMA	15.153	10.413	13.496	8.594	3.185	14.088	12.686
THETA	14.564	10.313	13.012	9.089	3.053	18.138	12.307
TBATS	14.918	10.188	12.950	**8.405**	3.003	**12.414**	12.301
SNAIVE	16.342	12.521	15.988	9.161	3.045	13.912	14.657
MASE							
HOC2	**3.066**	**1.149**	0.937	**2.450**	**3.212**	1.154	**1.580**
AVG	3.084	1.165	0.946	2.469	3.213	1.229	1.593
ETS	3.444	1.161	0.948	2.527	3.253	1.824	1.680
ARIMA	3.401	1.166	**0.931**	2.541	3.399	**0.949**	1.665
THETA	3.375	1.231	0.970	2.639	3.262	2.455	1.695
TBATS	3.437	1.186	1.053	2.486	3.274	1.235	1.733
SNAIVE	3.974	1.602	1.260	2.777	3.278	1.193	2.057
MSIS							
HOC2	**31.343**	**9.342**	9.149	20.722	29.269	17.123	**15.225**
AVG	31.607	9.454	9.159	21.062	29.288	16.955	15.319
ETS	34.897	9.452	**8.297**	20.386	29.700	17.487	15.678
ARIMA	45.071	11.090	8.762	19.525	32.312	**7.494**	18.701
THETA	44.451	11.624	9.546	24.096	32.557	21.053	19.145
TBATS	40.263	9.782	13.123	**18.140**	**28.978**	11.552	19.245
SNAIVE	56.554	13.346	10.846	26.358	32.552	9.054	22.925

H is the forecasting horizon. *Nf* is the number of time series per data frequency. The scores on the last column – *Total* – are the weighted averages of the previous columns with *Nf* as weight. Best results per frequency and metric are bold-faced.

Table 6 – M4 dataset – OWA performances.

	Yearly	*Quarterly*	*Monthly*	*Weekly*	*Daily*	*Hourly*	*Total*
H	6	8	18	13	14	48	-
Nf	23,000	24,000	48,000	359	4,227	414	100,000
OWA							
HOC2	**0.805**	**0.870**	**0.877**	**0.901**	**0.977**	0.597	**0.848**
AVG	0.809	0.879	0.884	0.903	0.978	0.622	0.854
ETS	0.903	0.891	0.915	0.931	0.996	0.852	0.908
ARIMA	0.892	0.898	0.905	0.927	1.041	**0.581**	0.903
THETA	0.870	0.917	0.907	0.971	0.999	1.006	0.897
TBATS	0.889	0.895	0.944	0.906	0.993	0.596	0.907
SNAIVE	1.000	1.153	1.147	1.000	1.000	0.628	1.078
sMAPE_b	16.342	11.012	14.427	9.161	3.045	18.383	13.564
MASE_b	3.974	1.371	1.063	2.777	3.278	2.395	1.912

H is the forecasting horizon. *Nf* is the number of time series per data frequency. As initially done by the competition´s organizers, the OWA scores on the last column – *Total* – are NOT the weighted averages of the previous columns: they are computed directly with the total sMAPE and MASE for the methods and the total sMAPE_b and MASE_b for the benchmarking. Best results per frequency are bold-faced.

Table 7 – Number of winning positions per method.

	Yearly	*Quartely*	*Monthly*	*Weekly*	*Daily*	*Hourly*	*Other*	*Total*
Point forecasts								
HOC2	**4**	**7**	**6**	**2**	**3**		**1**	**23**
AVG	1							1
ETS							**1**	1
ARIMA	2		1			**2**		5
THETA								
TBATS				1		1		2
SNAIVE								
Prediction intervals								
HOC2	**2**	**3**	**2**					**7**
AVG	1							1
ETS			1				**1**	2
ARIMA						**1**		1
THETA								
TBATS				**1**	**1**			2
SNAIVE								

Best results per frequency are bold-faced.

The simple average of the forecasts in the pool of methods relies on a total of H x K x Nt predictions per competition dataset. Thus, theoretically speaking, if AVG has a total processing time of τ, HOC2 will present processing time around $p\tau$. However, HOC2 training algorithm may be directly parallelized by time series cluster, which may bring the total processing time down to the largest-cluster processing time.

As practical example, Table 8 shows processing times collected from the M3 dataset experiments, both for HOC2 and AVG. (See Section 3.7 for a complete report about clusters´ lengths.) Considering the parallel version of the training algorithm, HOC2 may take, overall, 40% more processing time than AVG. With the non-parallel version, processing time can take up to 8.9 times the benchmarking (which is in line with our theoretical considerations).

Table 8 – Processing times.

	Yearly	*Quarterly*	*Monthly*	*Other*	*Total*
H	6	8	18	8	-
Nf	645	756	1428	174	3,003
Largest cluster % size (cMax)	32.9%	18.8%	15.0%	34.5%	-
Processing times (hour)					
HOC2_{np} (non-parallel)	0.4	2.7	18.1	0.2	21.4
HOC2_{p} (parallel)	0.1	0.5	2.7	0.1	3.4
AVG	0.1	0.3	2.0	0.0	2.4
HOC2_{np}/AVG	6.0	9.5	8.9	10.1	8.9
HOC2_{p}/AVG	2.0	1.8	1.3	3.5	1.4

H is the forecasting horizon. *Nf* is the number of time series per data frequency. HOC2_{p} = HOC2_{np} x cMax. These experiments were carried out with an Intel Core i7© / 16GB RAM machine running Microsoft Windows© 10.

3.6 Test phase statistics

In this section we provide statistics about the type of weighting matrices selected in the test phases of our experiments. Those matrices can be of 3 types (Section 2.5):

W1. Individually optimized for the time series, if the time series was in the training set and was considered to be of good length to have an acceptable number rolling origins r ($r \geq p_{min}$);

W2. Cluster optimized, if the series was not in the training set or had a limited number of rolling origins ($1 \leq r < p_{min}$);

W3. Static-equal (simple average) matrix, if the series was too small ($< 2.H$) for reliable calculations ($r = 0$).

Tables 9 to 11 show test phase statistics per dataset. Notice that W3-type matrices only happens over the M1 competition dataset.

Table 9 – M1 dataset – test phase statistics.

	Yearly	*Quarterly*	*Monthly*	*Total*
H	6	8	18	-
Nf	181	203	617	1,001
%W1	48%	86%	92%	83%
%W2	46%	1%	2%	10%
%W3	6%	13%	5%	7%

H is the forecasting horizon. *Nf* is the number of time series per data frequency. The percentages are rounded (they sum to 100%).

Table 10 – M3 dataset – test phase statistics.

	Yearly	*Quarterly*	*Monthly*	*Other*	*Total*
H	6	8	18	8	-
Nf	645	756	1428	174	3,003
%W1	70%	93%	100%	100%	92%
%W2	30%	7%	0%	0%	8%
%W3	0%	0%	0%	0%	0%

H is the forecasting horizon. *Nf* is the number of time series per data frequency. The percentages are rounded (they sum to 100%).

Table 11 – M4 dataset – test phase statistics.

	Yearly	*Quarterly*	*Monthly*	*Weekly*	*Daily*	*Hourly*	*Total*
H	6	8	18	13	14	48	-
Nf	23,000	24,000	48,000	359	4,227	414	100,000
%W1	9%	10%	10%	11%	10%	12%	10%
%W2	91%	90%	90%	89%	90%	88%	90%
%W3	0%	0%	0%	0%	0%	0%	0%

H is the forecasting horizon. *Nf* is the number of time series per data frequency. The percentages are rounded (they sum to 100%).

3.7 Additional reports

We end the experiments section with some additional reports about the framework operation. Figures 2 to 7 depict cluster distributions and respective mean weighting matrices (in a time-series fashion). For simplicity's sake, we chose to show only the largest clusters´ weights (but there are, in fact, one possible weighting matrix per cluster).

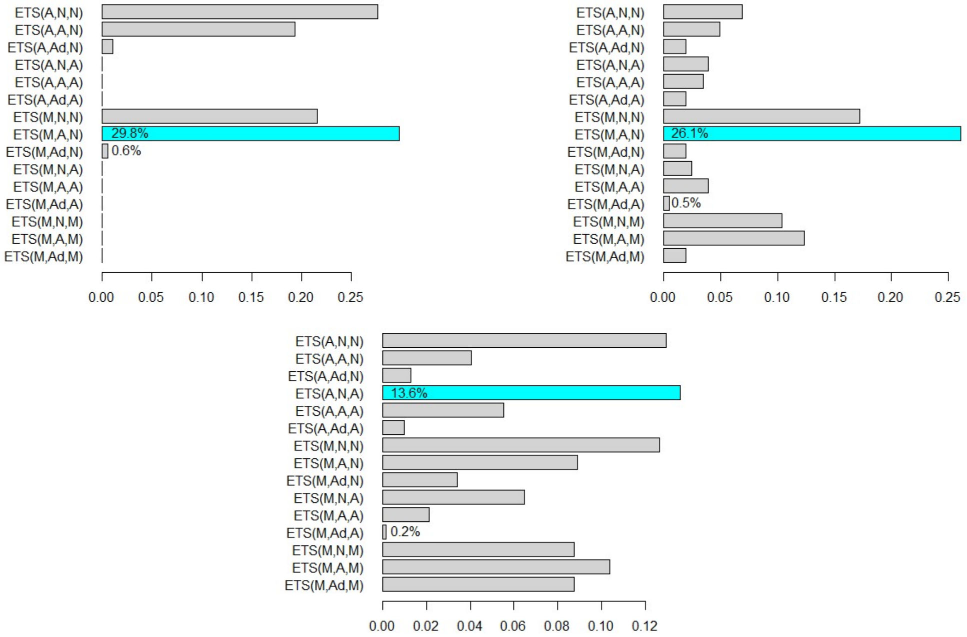

Figure 2 – M1 dataset – ETS model-form clusters distributions. For visual reference, largest and smallest (non-empty) clusters are numbered with their lengths. The largest clusters are highlighted.

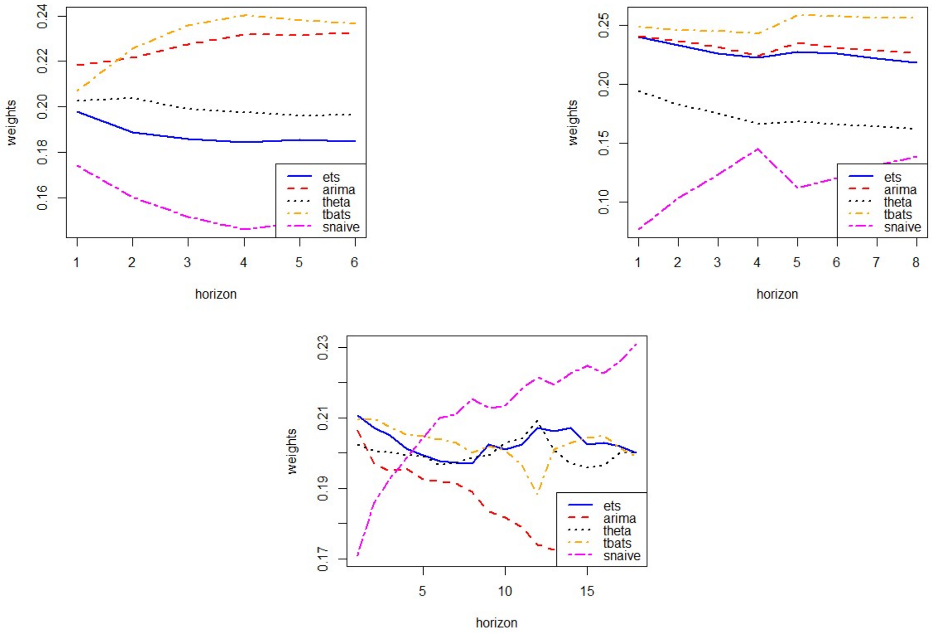

Figure 3 – M1 dataset – Largest clusters´ mean weighting matrices, in a time-series fashion. The graphs show how the method´s weights evolve along the forecasting horizon. The training set here was based on 100% (q = 100) of the total time series dataset.

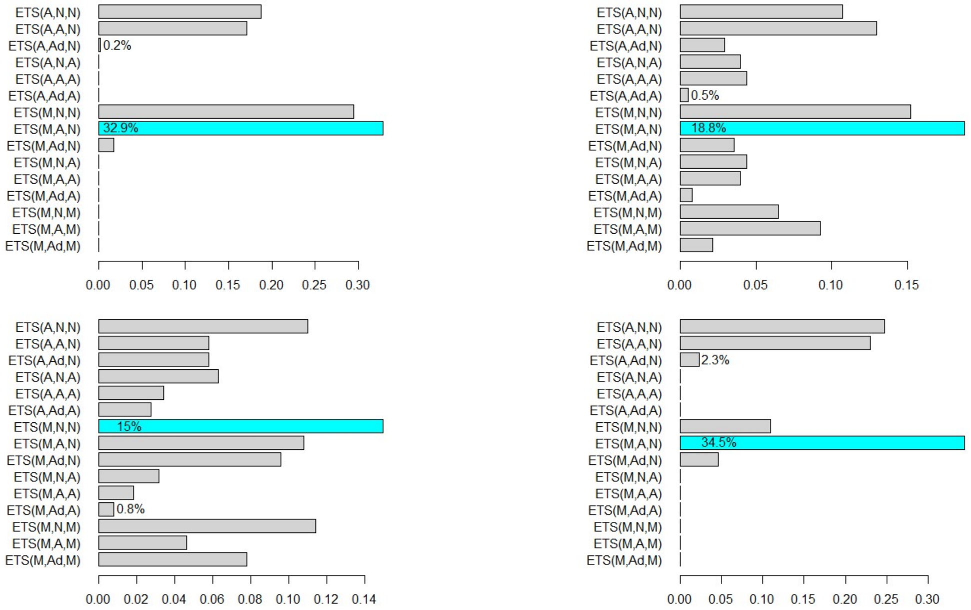

Figure 4 – M3 dataset – ETS model-form clusters distributions. For visual reference, largest and smallest (non-empty) clusters are numbered with their lengths. The largest clusters are highlighted.

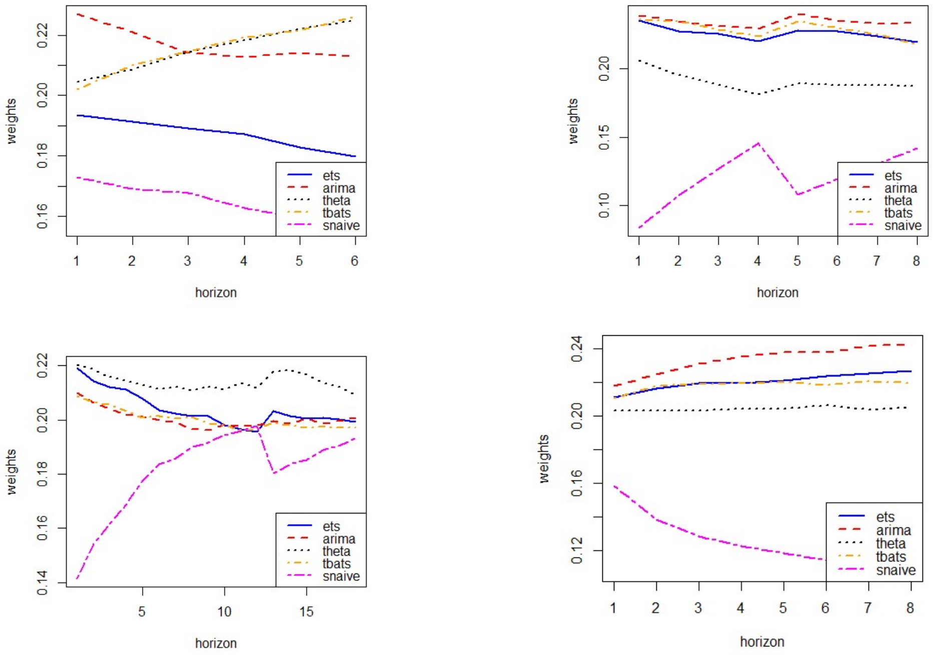

Figure 5 – M3 dataset – Largest clusters' mean weighting matrices, in a time-series fashion. The graphs show how the method´s weights evolve along the forecasting horizon. The training set here was based on 100% (q = 100) of the total time series dataset.

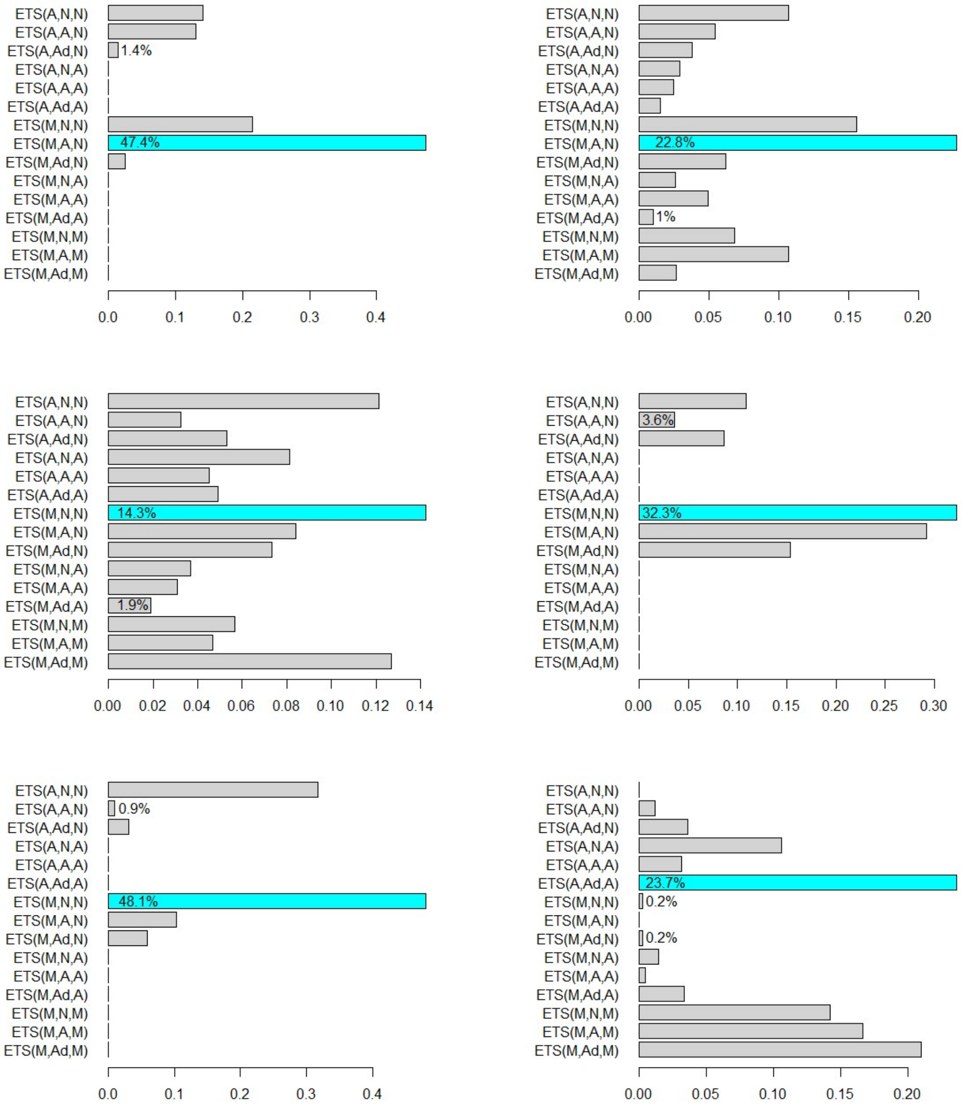

Figure 6 – M4 dataset – ETS model-form clusters distributions. For visual reference, largest and smallest (non-empty) clusters are numbered with their lengths. The largest clusters are highlighted.

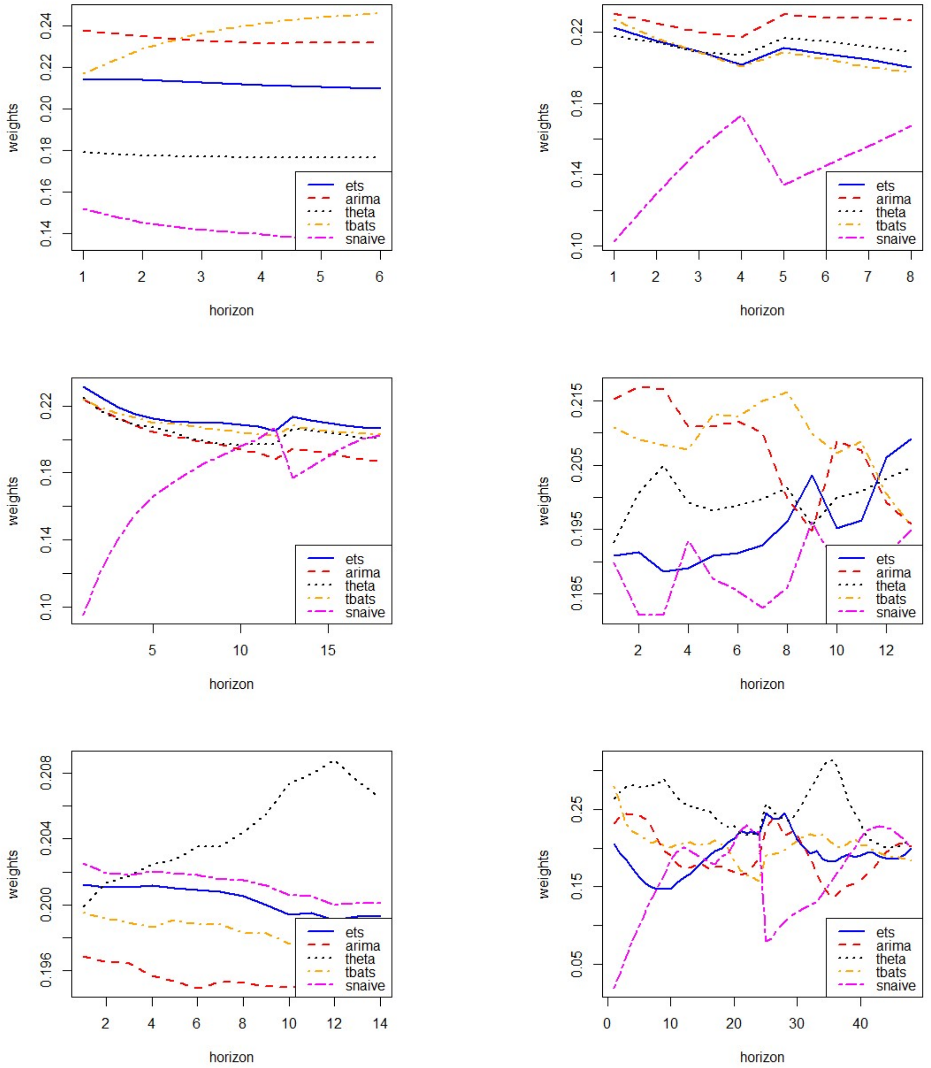

Figure 7 – M4 dataset – Largest clusters´ mean weighting matrices, in a time-series fashion. The graphs show how the method´s weights evolve along the forecasting horizon. The training set here was based on 10% (q = 10) of the total time series dataset.

4 CONCLUSION

This paper proposes a forecast combination framework that considers horizon-optimized weights, i.e., weights that may vary over the forecasting horizon. The framework – named Horizon-Optimized Convex Combinations (HOC2) – builds on cross-learning, time series clustering and cross-validation to form convex combinations of forecasts from 5 methods – Automated exponential smoothing, Automated ARIMA, Theta, TBATS and Seasonal naïve. It was tested with 104,004 time series from the past M1, M3 and M4 competitions.

Concerning overall results both for point forecasts (PFs) and prediction intervals (PIs), HOC2 presented performance gains over its five underlying forecasting methods alone and over their simple average combination (AVG) in 9 out of 10 possible winning positions. Also, the framework presented somewhat competitive results for the M4 dataset, a 100,000 time-series dataset that served the original M4 competition.

Breaking the results by data frequency, the framework outperformed the benchmarks on 23 out of 32 winning positions (72%) for point forecasts (PFs) and had a 7/13 (54%) winning performance for PIs. More important, it is the most winning approach for both PFs (23 x 5 ARIMA winnings) and PIs (7 x 2 ETS/TBATS winnings). In other words, it was the overall less risky approach, particularly when compared to simple average (AVG), which presented only one winning position for PFs and other for PIs.

Considering the M4 dataset alone, we highlight that the presented results were based on a 10% training set, wisely sampled from the original dataset. We see this as an interesting fact in two ways: (i) as an element to reduce processing complexity (by learning and generalization), and (ii) as something that leaves room for future performance improvements, together with the usage of the cluster-parallel training procedure (that leads to much shorter processing times).

Considering the extensive analysis and presented performances, our results shall be helpful to support future research on how horizon-optimized weights can be used interchangeably with static ones. Here are some improvement ideas to follow:

- Deeper analysis of the framework´s operation, with further investigation of its hyperparameters and introduction of other benchmarks;
- Study of alternative methods for time series clustering, before the cross-learning training phase;
- Better care of the pool of methods being combined. For instance, straightforward exercises would be to use pruned and treated models generated by the ETS function (Meira, Cyrino Oliveira, and Jeon, 2020) or to replace the Theta model by its optimized versions (Fiorucci and Louzada, 2020): the Dynamic Optimized Theta model (DOTM) or the Optimized Theta model (OTM). For automation purposes, our approach intentionally relied on the use of methods in their default form, but working on the pool shall bring improvements, as pointed out by Atiya (2020): (i) “Forecast combination should be a winning

strategy if the constituent forecasts are either diverse or comparable in performance” and (ii) “One should exclude forecasts that are considerably worse than the best ones in the pool, unless they are very diverse from the rest”. Those points agree with many previous recommendations discussed in the literature (Armstrong, 2001; Timmermann, 2006; Kourentzes, Barrow & Petropoulos, 2019);

- Association of HOC2´s horizon-optimized weights with more complex cross-learning schemes, possibly joining statistics and machine learning techniques, as done, for instance, by Montero-Manso, Athanasopoulos, Hyndman, and Talagala (2020).

The source code for HOC2 is available at https://github.com/rvsantos2000/hoc2.

References

[1] ANGELO SA, ARRUDA EF, GOLDWASSER R, LOBO MSC, SALLES A & LAPA E SILVA JR. 2017. Demand forecast and optimal planning of intensive care unit (ICU) capacity. *Pesquisa Operacional*, **37**(2): 229–245.

[2] ARMSTRONG JS. 2001. Combining forecasts. In JS Armstrong (Ed.). *Principles of forecasting: A handbook for researchers and practitioners*. Kluwer Academic Publishers.

[3] ASSIMAKOPOULOS V & NIKOLOPOULOS K. 2000. The theta model: a decomposition approach to forecasting. *International Journal of Forecasting*, **16**(4): 521–530.

[4] ATIYA A. 2020. Why does forecast combination work so well? *International Journal of Forecasting*, **36**(1): 197–200.

[5] BANDARA K, BERGMEIR C & SMYL S. 2020. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, **140**: 112896.

[6] CRONE SF, HIBON M & NIKOLOPOULOS K. 2011. Advances in forecasting with neural networks: Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting*, **27**(3): 635–660.

[7] DA SILVA FLC, CYRINO OLIVEIRA FL & SOUZA RC. 2019. A bottom-up bayesian extension for long-term electricity consumption forecasting. *Energy*, **167**: 1980–210.Available at: https://www.scopus.com/record/display.uri?eid=2-s2.0--85056242461&origin=resultslist

[8] DE LIVERA A.M, HYNDMAN R.J & SNYDER RD. 2011. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, **106**(496): 1513–1527.

[9] DIEBOLD FX. 1988. Serial Correlation and the Combination of Forecasts. *Journal of Business and Economic Statistics*, **6**: 105–111.

[10] FIORUCCI JA & LOUZADA F. 2020. GROEC: combination method via generalized rolling origin evaluation. *International Journal of Forecasting*, **36**(1): 105–109.

[11] FIORUCCI JA, PELLEGRINI TR, LOUZADA F & PETROPOULOS F. 2015. The optimized theta method. arXiv preprint Available at: http://arxiv.org/abs/1503.03529.

[12] FRY S & BRUNDAGE M. 2020. The M4 forecasting competition – A practitioner's view. *International Journal of Forecasting*, **36**(1): 156–160.

[13] GRANGER CWJ & RAMANATHAN R. 1984. Improved Methods of Combining Forecasts. *Journal of Forecasting*, **3**: 197–204.

[14] HYNDMAN RJ. 2020. A brief history of forecasting competitions. *International Journal of Forecasting*, **36**(1): 7–14.

[15] HYNDMAN RJ & BILLAH MB. 2003. Unmasking the theta method. *International Journal of Forecasting*, **19**(2): 287–290.

[16] HYNDMAN RJ & KHANDAKAR Y. 2008. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, **27**(3): 1–22.

[17] HYNDMAN RJ, BERGMEIR C, CACERES G, CHHAY L, O'HARA-WILD M, PETROPOULOS F, RAZBASH S, WANG E & YASMEEN F. 2018. *Forecast: forecasting functions for time series and linear models. R package version 8.3*. Available at: http://pkg.robjhyndman.com/forecast.

[18] HYNDMAN RJ, KOEHLER AB, ORD JK & SNYDER RD. 2008. *Forecasting with exponential smoothing: the state space approach*. Springer-Verlag. Available at: http://www.exponentialsmoothing.net.

[19] HYNDMAN RJ, WANG E, KANG Y & TALAGALA T. 2018. *tsfeatures: Time series feature extraction. R package version 0.1.*

[20] JAGANATHAN S & PRAKASH P. 2020. Combination based forecasting method: M4 competition. *International Journal of Forecasting*, **36**(1), 98–104.

[21] KOURENTZES N, BARROW D & PETROPOULOS F. 2019. Another look at forecast selection and combination: Evidence from forecast pooling. *International Journal of Production Economics*, **209**: 226–235.

[22] MAKRIDAKIS S & HIBON M. 2000. The M3-competition: Results. conclusions and implications. *International Journal of Forecasting*, **16**(4): 451–476.

[23] MAKRIDAKIS S, ANDERSEN A, CARBONE R, FILDES R, HIBON M, LEWANDOWSKI R, JOSEPH NEWTON H, PARZEN E & WINKLER RL. 1982. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, **1**(2): 111–153.

[24] MAKRIDAKIS S, SPILIOTIS E & ASSIMAKOPOULOS V. 2020. The M4 competition: 100.000 time series and 61 forecasting methods. *International Journal of Forecasting*, **36**(1): 54–74.

[25] MEIRA E, CYRINO OLIVEIRA FL & JEON J. 2021. Treating and Pruning: New approaches to forecasting model selection and combination using prediction intervals. *International Journal of Forecasting*, **37**: 547–568.

[26] MONTERO-MANSO P, ATHANASOPOULOS G, HYNDMAN RJ & TALAGALA TS. 2020. FFORMA: feature-based forecast model averaging. *International Journal of Forecasting*, **36**(1): 86–92.

[27] OLIVEIRA FLC, SOUZA RC & MARCATO ALM. 2015. A time series model for building scenarios trees applied to stochastic optimization. *International Journal of Electrical Power and Energy Systems*, **67**: 315–323.

[28] PAWLIKOWSKI M, CHOROWSKA A & YANCHUK O. 2020. Weighted ensemble of statistical models. *International Journal of Forecasting*, **36**(1): 93–97.

[29] PETROPOULOS F & SVETUNKOV I. 2020. A simple combination of univariate models. *International Journal of Forecasting, 36*(1), 110–115.

[30] PETROPOULOS F, APILETTI D, ASSIMAKOPOULOS V, BABAI M.Z, BARROW D.K, BEN TAIEB S, BERGMEIR C, BESSA R.J, BIKAJ J, BOYLAN J.E, BROWELL J, CARNEVALE C, CASTLE JL, CIRILLO P, CLEMENTS MP, CORDEIRO C, CYRINO OLIVEIRA FL, DE BAETS S, DOKUMENTOV A, ELLISON J, FISZEDER P, FRANSES P.H, FRAZIER DT, GILLILAND M, GÖNÜL MS, GOODWIN P, GROSSI L, GRUSHKA-COCKAYNE Y, GUIDOLIN M, GUIDOLIN M, GUNTER U, GUO X, GUSEO R, HARVEY N, HENDRY D.F, HOLLYMAN R, JANUSCHOWSKI T, JEON J, JOSE VRR, KANG Y, KOEHLER AB, KOLASSA S, KOURENTZES N, LEVA S, LI, F, LITSIOU K, MAKRIDAKIS S, MARTIN GM, MARTINEZ AB, MEERAN S, MODIS T, NIKOLOPOULOS K, ÖNKAL D, PACCAGNINI A, PANAGIOTELIS A, PANAPAKIDIS I, PAVÍA JM, PEDIO M, PEDREGAL TERCERO DJ, PINSON P, RAMOS P, RAPACH D, READE JJ, ROSTAMI-TABAR B, RUBASZEK M, SERMPINIS G, SHANG HL, SPILIOTIS E, SYNTETOS AA, TALAGALA PD, TALAGALA TS, TASHMAN L, THOMAKOS D, THORARINSDOTTIR T, TODINI E, TRAPERO ARENAS JR, WANG X, WINKLER RL, YUSUPOVA A & ZIEL F. 2021. *Forecasting: theory and practice*. arXiv 2012.03854.

[31] PETROPOULOS F, HYNDMAN RJ & BERGMEIR C. 2018. Exploring the sources of uncertainty: why does bagging for time series forecasting work? *European Journal of Operational Research,* **268**: 545–554.

[32] SMITH J & WALLIS KF. 2009. A simple explanation of the forecast combination puzzle. *Oxford Bulletin of Economics and Statistics*, **71**(3), 331–355.

[33] SOUZA RC, MARCATO ALM, DIAS BH & OLIVEIRA FLC. 2012. Optimal operation of hydrothermal systems with Hydrological Scenario Generation through Bootstrap and Periodic Autoregressive Models. *European Journal of Operational Research*, **222**(3): 606–615.

[34] SMYL S. 2020. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, **36**(1): 75–85.

[35] TEIXEIRA JUNIOR LA, SOUZA RM, MENEZES ML, CASSIANO KM, PESSANHA JFM & SOUZA RC. 2015. Artificial neural network and wavelet decomposition in the forecast of global horizontal solar radiation. *Pesquisa Operacional*, **35**(1): 73–90.

[36] TIMMERMANN A. 2006. Forecast combinations. In: G Elliott. CWJ Granger & A Timmermann (Eds.). *Handbook of economic forecasting*, p. 135–196. Elsevier.

[37] VALLE DOS SANTOS RO & VELLASCO MMBR. 2015. Neural Expert Weighting: A NEW framework for dynamic forecast combination. *Expert Systems with Applications*, **42**(22): 8625–8636.

How to cite

VALLE DOS SANTOS RO ET AL. 2021. Horizon-optimized weights for forecast combination with cross-learning. *Pesquisa Operacional*, **41**: e245564. doi: 10.1590/0101-7438.2021.041.00245564.