Multi-output Ensembles for Multi-step Forecasting

Vitor Cerqueira[1,2], Luis Torgo[3]

[1] Faculdade de Engenharia da Universidade do Porto, Porto, Portugal
[2] Laboratory for Artificial Intelligence and Computer Science (LIACC), Portugal
vitorc.research@gmail.com
[3] Dalhousie University, Halifax, Canada

Abstract. Ensemble methods combine predictions from multiple models to improve forecasting accuracy. This paper investigates the effectiveness of multi-output ensembles for multi-step time series forecasting problems. While dynamic ensembles have been extensively studied for one-step ahead forecasting, their application to multi-step forecasting remains largely unexplored, particularly regarding how combination rules should be applied across different forecasting horizons. We conducted comprehensive experiments using 3568 time series from diverse domains and an ensemble of 30 multi-output models to address this research gap. Our findings reveal that dynamic ensembles based on arbitrating and windowing techniques achieve the best performance according to average rank. Interestingly, we observed that most dynamic approaches struggle to outperform a simple static ensemble that assigns equal weights to all constituent models, especially as the forecasting horizon increases. The performance advantage of dynamic methods is more pronounced in short-term forecasting scenarios. The experiments are publicly available in a repository.

Keywords: Ensemble methods · Time series forecasting · Multi-output models · Time series

1 Introduction

Multi-step forecasting plays a key role in organizational planning by reducing long-term uncertainty in time series data. Ensemble methods combine the output of several models to make aggregated predictions. These approaches have been shown to improve predictive performance across many tasks [6], including forecasting [7]. One of the main advantages of ensembles is that they reduce the risk of selecting a suboptimal model [13].

Since Bates and Granger's pioneering work [3] on combining different models for time series forecasting, hundreds of studies have explored forecast combination techniques. The simplest approach is to assign equal weights to each model through arithmetic mean, though weighted averages that reflect expected model performance are also common. In non-stationary time series, where data distributions change over time, the combination rules typically need to be dynamic, with weights being adapted to temporal changes in the series.

The literature encompasses several dynamic forecast combination strategies, most based on regret minimization [10], windowing [19], or meta-learning [9]. However, these dynamic approaches are typically designed for one-step ahead forecasting problems, assuming immediate feedback about actual values for error computation and combination rule updates.

Despite the established importance of multi-step forecasting across domains, there remains a significant research gap regarding how dynamic combination methods should be applied in these settings. For instance, should different weights be computed for each horizon, or should weights be estimated jointly for all forecasting horizons? While approaches for multi-step forecasting include recursive, direct, or multi-output methods [20], with multi-output methods demonstrating superior performance [20], the literature on dynamic ensembles specifically for multi-step ahead forecasting remains scarce.

Our work addresses this gap by investigating the performance of ensembles composed of multi-output models for multi-step forecasting problems. Our goal is two-fold:

1. To determine which dynamic combination rule performs best for multi-step ahead problems, extending previous analyses that focused on one-step ahead tasks [9].
2. To identify the optimal approach for computing ensemble weights along the forecasting horizon. To our knowledge, no prior work has addressed this topic.

We conducted a comprehensive empirical study using 3568 univariate time series from multiple domains. Our experiments tested various combination rules with an ensemble of 30 individual multi-output models, all implemented as standard supervised learning regression algorithms trained for auto-regression.

The results indicate that dynamic combination rules based on arbitrating [9] and windowing [15] achieve the best average rank. We also found that the forecasting horizon significantly impacts performance. While dynamic combination works in short-term forecasting, this type of approach deteriorates as the horizon increases. In effect, most approaches struggle to outperform simple equal-weight combinations for multi-step ahead forecasting. Regarding weight computation across horizons, we observed no substantial differences among the tested approaches. All data sets and experimental code are publicly available in an online repository[4].

The rest of the paper is organized as follows. In the next section (Section 2), we provide a background to our work. We start by defining the time series predictive task. We also overview the literature related to our work. We focus on the topics of dynamic ensembles and multi-step forecasting. In Section 3, we present the materials and methods used in this paper. We describe the case study

[4] https://github.com/vcerqueira/experiments-multioutput_ensembles

which contains time series from several application domains. We also describe the methods used in detail. Moreover, we explain the experimental design used in the experiments. Then, we present the experiments in Section 4. In that section, we outline the research questions and provide an empirical answer to them. We discuss the results obtained in Section 5, pointing out some future directions. Finally, the paper is concluded in Section **??**.

2 Background

2.1 Time Series Forecasting

We define a univariate time series as a sequence of values $Y = \{y_1, y_2, \ldots, y_n\}$, where $y_i \in \mathcal{Y} \subset \mathbb{R}$ is the numeric value of the time series at time i and n is the length of Y. We assume that the observations of the series are captured at regular periods (e.g. every hour). This work addresses multi-step ahead forecasting (also referred to as long-term forecasting [20]) problems in univariate time series. In practice, we aim at predicting the value of the H upcoming observations of the time series, $y_{n+1}, \ldots, y_{n+H}$, where H denotes the forecasting horizon.

The predictive task is formalized as an auto-regressive problem. Thus, each observation is represented based on the past and recent values before it. This is accomplished by reconstructing the time series using time delay embedding. This method is used to transform the time series from a sequence to a tabular format. The transformation based on time delay embedding leads to a data set $\mathcal{D} = \{(X, y)\}$. As described before, each observation y_i is modeled according to past q values before it: $X_i = \{y_{i-1}, y_{i-2}, \ldots, y_{i-q}\}$, where $y_i \in \mathcal{Y} \subset \mathbb{R}$. In effect, y represents the target variable which represents the observation we want to predict, and $X_i \in \mathcal{X} \subset \mathbb{R}^q$ represents the i-th embedding vector. Then, we train a multiple regression model f that can be written as $y_i = f(X_i)$. This definition is designed for one-step ahead forecasting. In Section 2.4, we describe how this formalization is extended to the multi-step ahead case.

2.2 Ensemble Methods

The *No Free Lunch* theorem for supervised learning [23] postulates that no learning algorithm is the most appropriate for all problems. All methods have strengths and limitations. This is the key motivation for ensemble methods [6]. While ensemble methods are also amenable to this problem, they reduce the risk of selecting a wrong model by combining multiple ones [13].

Ensemble methods aim at combining the output of several different predictive models. These methods have been shown to perform better than individual models in many tasks and domains of applications. A key aspect of developing accurate ensembles is the diversity among individual models. The models composing the ensemble should be accurate but different from each other. Brown et al. [5] survey several methods to achieve this. In this paper, we focus on heterogeneous ensembles [14]. These ensembles are composed of individual models which are trained with distinct learning algorithms, which is a typical way of encouraging diversity.

2.3 Dynamic Ensembles for Forecasting

Time series forecasting is one of the tasks in which ensembles have been shown to provide state-of-the-art performance [3, 9]. Time series data is amenable to change due to sources of non-stationarity. Consequently, different forecasting models usually show varying relative performance [1]. Aiolfi and Timmermann study this phenomenon. They discovered that some models perform better than others in some periods over a time series. This is the main motivation for using dynamic ensembles for forecasting.

A dynamic ensemble combines the predictions of individual models using weighted averages, where the weights change over time. The weights change to adapt to the current process generating the time series. The main problem when using dynamic ensembles is defining how to compute these weights at each time-step.

Windowing and Regret-based Approaches. The static forecast combination approach which assigns equal weights to all available models is a robust combination method [11] (`Simple`). Another static combination approach is a weighted average, in which the weights are set according to the performance of models in the training data (`LossTrain`). A variant of this approach is to select the model with the best performance on the training data (`Best`). Notwithstanding, dynamic combination rules are also typically used in forecasting problems.

Using past recent performance is a typical way of dynamically combining ensembles. The idea is to give a higher weight to models that performed better in recent observations (`Window`). This approach has shown promising results in different works [15, 7, 19].

Combining the output of multiple models is a well-studied topic in the online learning literature [10]. Example methods include the exponentially weighted average (`EWA`), the polynomially weighted average (`MLpol`), and the fixed share aggregation (`FS`). These approaches are designed to minimize regret. Regret is the average error suffered relative to the best error we could have obtained. We refer the reader to the second chapter of the seminal work by Cesa-Bianchi and Lugosi [10].

Meta-learning. Meta-learning is another type of strategy that can be used to combine the output of multiple models. Arguably, the most popular meta-learning approach is `Stacking` [22]. Cerqueira et al. [9] proposed a meta-learning approach called Arbitrated Dynamic Ensemble (`ADE`) for dynamic forecast combinations. `ADE` works by building a meta-model for each model in the ensemble. Each meta-model is designed to model and forecast the error of the corresponding (base) model. Then, the models in the ensemble are weighted according to the error forecasts provided by meta-models.

2.4 Multi-step Forecasting

Multi-step ahead forecasting denotes the process of predicting multiple instances of a time series. This task reduces the long-term uncertainty of time series, which is desirable across many application domains. Several works have been devoted to multi-step ahead forecasting problems [20, 21, 12].

In this section, we review several state-of-the-art approaches for multi-step ahead forecasting. We split these into two types: single-output methods (Section 2.4), and multi-output methods (Section 2.4). Taieb et al. [20] are followed closely to describe some of these methods.

Single Output Methods One of the most popular approaches to multi-step ahead forecasting is the Recursive method (also known as the Iterative). Recursive works by fitting a model f for one-step ahead forecasting (c.f. Section 2.1):

$$y_{i+1} = f(\{y_i, y_{i-1}, \dots, y_{i-q+1}\})$$

To get predictions for the next H observations, the model f is iterated H. To be more precise, in the i-th time-step, $\{y_i, y_{i-1}, \dots, y_{i-q+1}\}$ is the input used to forecast the value of y_{i+1}. Let $\hat{y}_{i+1}$ denote that prediction. Then, $\{\hat{y}_{i+1}, y_i, \dots, y_{i-q}\}$ is the input vector for forecasting $\hat{y}_{i+2}$ in the same time-step. The Recursive strategy is known to propagate errors along the forecasting horizon. Thus, this approach requires an accurate model specification to work well.

Another popular multi-step ahead forecasting method is the Direct approach. This strategy works by building a forecasting model for each horizon:

$$y_{i+h} = f_h(\{y_i, y_{i-1}, \dots, y_{i-q+1}\}),$$

where $h \in \{1, \dots, H\}$. Since no model is iterated along the horizon, the Direct approach does not suffer from error propagation. Notwithstanding, this method leads to greater computational costs because it trains a model for each horizon. Besides, it assumes that each horizon is independent, which is not generally true.

The method DirRec attempts to bridge the best aspects of Recursive and Direct. Similarly to Direct, this approach builds one model for each forecasting horizon. Moreover, for each successive horizon, the set of inputs is augmented with the predictions from previous steps (following Recursive). This approach is also referred to as classifier (or regression) chains in the machine learning literature [18].

Multiple Output Methods As the name implied, single output approaches model one horizon at a time. Thus, they do not account for the stochastic dependency along the forecasting horizon. This aspect is taken into account by multi-input multi-output (MIMO) models [20]. In this work, we also refer to these approaches by multi-output models.

A multi-output model is trained on the complete forecasting horizon jointly:

$$[y_{i+1}, y_{i+2}, \dots, y_{i+H}] = F(\{y_i, y_{i-1}, \dots, y_{i-q+1}\}),$$

where F denotes the multi-output model. Essentially, there are H target variables instead of one, and these are modeled jointly by a single model.

Taieb et al. [20] compared several approaches for multi-step ahead forecasting, including the ones described above. They used the 111 time series from the NN5 forecasting competition. The main conclusion is that multi-output strategies perform better than single-output ones. For this reason, we focus on multi-output methods in this work. Notwithstanding, the dynamic ensembles used in this paper can also be applied with other multi-step ahead forecasting strategies.

3 Materials and Methods

This section details the materials and methods used in this work. We describe the data sets used in the experiments in Section 3.1. Then, we detail the methodology carried out to evaluate each approach (Section 3.2). We list all approaches in Section 3.3, including learning algorithms, dynamic combination rules, and different strategies for weighting models along the horizon. Finally, we detail the experimental design in Section 3.4, including the cross-validation procedure and evaluation metric.

3.1 Data

The experiments encompassed 3568 time series. These were collected from the following 7 popular databases `electricity_nips`, `nn5_daily_without_missing`, `solar-energy`, `traffic_nips`, `taxi_30min`, `m4_hourly`, and `m4_weekly`. Table 1 presents a summary of these datasets, including the number of time series and their average length. Their name refers to the corresponding identifier in the Python library *gluonts* [2].

Table 1. Summary of the data sets

Name	Frequency	# Time Series	Avg. Length
`electricity_nips`	Hourly	370	5833
`nn5_daily_without_missing`	Daily	111	735
`solar-energy`	Hourly	137	7009
`traffic_nips`	Hourly	963	4001
`taxi_30min`	Half-hourly	1214	1488
`m4_hourly`	Hourly	414	960
`m4_weekly`	Weekly	359	934

3.2 Methodology

The goal of this paper is to analyze how several dynamic ensembles perform for multi-step ahead forecasting problems. This is accomplished by carrying out a set of experiments. For each time series in the case study (c.f. Section 3.1), we apply the following procedure. The available time series is split into training and test sets. As we describe below in Section 3.4, this process is repeated using a cross-validation procedure.

The models in the ensemble are fit using the training data. As mentioned in Section 2.3, pruning the ensemble usually leads to better forecasting performance. In effect, we prune the ensemble and keep only the best 75% of models in the available pool. The pruning process is carried out using a nested validation procedure. The training set is further split into two parts: an inner training set and a validation set. The inner training set contains 70% of the observations of the complete (outer) training set. The validation set contains the final 30% of the complete training set. All models are fit using the inner training set and evaluated in the validation set. The 25% of models with the worst forecasting performance are discarded. The remaining ones are re-fit using the complete training set.

After the pruning and re-fitting process, the ensemble is applied to the test set. A combination rule is applied to aggregate the predictions of all available models. The list of all combination rules applied in the experiments is described in Section 3.3. Note that we test these combination rules with different approaches concerning the estimation of weights across the horizon. We test four strategies that are listed in Section 3.3. Finally, the performance of each method is evaluated according to its performance in the test set. The evaluation metric is described in Section 3.4.

3.3 Methods

This section details the methods used in the experiments. First, we describe the regression learning algorithms used to create multi-output forecasting models (Section 3.3). Then, we describe the 9 methods used to combine the predictions of those models (Section 3.3). Finally, we detail four different approaches to compute the ensemble weights across the forecasting horizon (Section 3.3).

Learning Algorithms The ensembles used in the experiments are composed of 40 individual multi-output models. These were created using the following learning algorithms: random forest regression, extra trees regression, bagging of decision trees, projection pursuit regression, LASSO regression, ridge regression, elastic-net regression, k-nearest neighbors regression, principal components regression, and partial least squares regression. We used the implementation in the *scikit-learn* [16] Python library to use these methods. Table 2 describes the different parameters used for each learning algorithm. In total, there are 40 different learning approaches.

Table 2. Summary of the parameters of the learning algorithms

ID	Algorithm	Parameter(s)	Value(s)
BAGGING_1	Bagging of decision trees	No. trees	50
BAGGING_2			100
RF_1	Random Forest	{No. trees, max depth}	{50, *default*}
RF_2			{50, 3}
RF_3			{50, 5}
RF_4			{100, *default*}
RF_5			{100, 3}
RF_6			{100, 5}
ET_1	Extra trees regression	{No. trees, max depth}	{50, *default*}
ET_2			{50, 3}
ET_3			{50, 5}
ET_4			{100, *default*}
ET_5			{100, 3}
ET_6			{100, 5}
KNN_1	K-nearest neighbors	{K, weight}	{1, uniform}
KNN_2			{5, uniform}
KNN_3			{10, uniform}
KNN_4			{20, uniform}
KNN_5			{50, uniform}
KNN_6			{1, distance}
KNN_7			{5, distance}
KNN_8			{10, distance}
KNN_9			{20, distance}
KNN_10			{50, distance}
PPR	Projection pursuit regression		*default*
LASSO_1	LASSO regression	{regularization}	{1}
LASSO_2			{0.75}
LASSO_3			{0.5}
LASSO_4			{0.25}
RIDGE_1	Ridge regression	{regularization}	{1}
RIDGE_2			{0.75}
RIDGE_3			{0.5}
RIDGE_4			{0.25}
EN	Elastic-net regression		*default*
PLS_1	Partial least squares regression	No. components	2
PLS_2			3
PLS_3			5
PCR_1	Principal components regression	No. components	2
PCR_2			3
PCR_3			5

Forecast Combination Methods In terms of forecast combination methods, we focus on the following approaches.

- `Simple`: Combination rule which assigns equal weights to all models. In practice, the predictions of the available models are combined using the arithmetic mean;
- `Window`: Dynamic weighted average of the predictions of the available models [15]. The weights are computed according to the forecasting performance in the last λ observations;

- `Blast`: A variant of the `Window` approach [19]. Instead of using past recent performance to weigh the available models, the idea is to select the model with the best performance in the last λ observations;
- `ADE`: A dynamic combination approach based on a meta-learning strategy called arbitrating [9]. The idea is to build a meta model (a Random Forest) for each (base) model in the ensemble. Each meta model is designed to predict the error of the corresponding base model. Then, the models in the ensemble are weighted according to the error forecasts. We refer to the work by Cerqueira et al. [9] for a complete read on this method;
- `EWA`: A dynamic combination rule based on an exponentially weighted average. This method follows the popular weighted majority algorithm [10];
- `FS`: The fixed share dynamic combination approach. This method is designed to handle non-stationary time series;
- `MLpol`: A dynamic combination method based on a polynomially weighted average;
- `Best`: A baseline which selects the individual model in the ensemble with the best performance in the training data to predict all the test instances;
- `LossTrain`: Another baseline which weights the available models based on the error on the training set. The weights are static and fixed for all testing observations;

Most of these combination approaches are dynamic to cope with the non-stationarities present in the time series. The exceptions are `LossTrain` and `Simple`. We followed the study by Cerqueira et al. [9] to set the value of the λ parameter to 50 observations.

Weighting Approaches Over the Horizon Dynamic ensemble methods typically assume immediate feedback. They are designed only for one-step ahead forecasting. Thus, it is not clear how the ensemble weights should be computed along the forecasting horizon. We study the following approaches to estimate the weights at each time-step:

- `Complete Horizon (CH)`: The weights of individual models are estimated using their average performance over the complete forecasting horizon;
- `Individual Horizon (IH)`: The ensemble estimates different weights for each horizon;
- `First Horizon Forward (FHF)`: The weights computed for the first horizon are propagated over the rest of the horizon;
- `Last Horizon Backward (LHB)`: For completeness, we include the inverse approach to `FHF`. According to `LHB`, the weights computed for the last horizon are propagated backward to all horizons before it.

We test all these variants with the combination rules presented above. The exception is `Simple`, whose weights are static and not dependent on forecasting performance. This leads to a total of 33 variants for analysis.

3.4 Experimental Design

We estimate the forecasting performance of models using a Monte Carlo cross-validation procedure [17], which is also referred to as repeated holdout [8]. This estimation method is applied with 10 folds. The training and test sizes of each fold are set to 60% and 10% of the size of the input time series, respectively. Monte Carlo cross-validation provides competitive performance estimates relative to other approaches [8].

We preprocess each time series as follows. We take the first differences to remove the trend. Then, we apply time delay embedding to transform the series for auto-regression. We set the parameter q (the number of lags) to 5. This means that the future values of a time series are modeled based on the previous 5 observations. Finally, we set the maximum forecasting horizon to 18 observations ($H = 18$). The mean absolute error (MAE) is used as the evaluation metric. This metric has the limitation of being scale-dependent. However, we will focus on ranks and percentage differences to compare results across multiple time series.

4 Experiments

This section presents the experiments conducted to compared different forecast combination methods. These are designed to address the following research questions:

- RQ1: How do the dynamic ensemble methods compare with each other in terms of their rank across all data sets?
- RQ2: What is the best approach for computing the weights along the forecasting horizon?
- RQ3: How does each dynamic ensemble method compare with a static ensemble that assigns equal weights to all models?
- RQ4: Does the forecasting horizon affect the results obtained?

4.1 Results

RQ1 Figure 1 shows the average (mean) rank, and respective standard deviation, of each method across the 3568 time series. An approach gets a rank of 1 in a given data set if it shows the lowest error. The best approaches are variants of `ADE`, `Window`, and `MLpol`. On the other hand variants of `Blast`, `Best`, `FS`, and `EWA` occupy the bottom positions. All methods show a considerable standard deviation of rank. This corroborates the idea that no method dominates over the rest.

RQ2 Figure 2 illustrates the average rank of all methods but aggregated by weighting strategy. This analysis uncovers interesting outcomes. Varying the weighting strategy (`FHF`, `LHB`, `IH`, `CH`) does not have a significant impact in average rank. On the other hand, different combination approaches show significantly

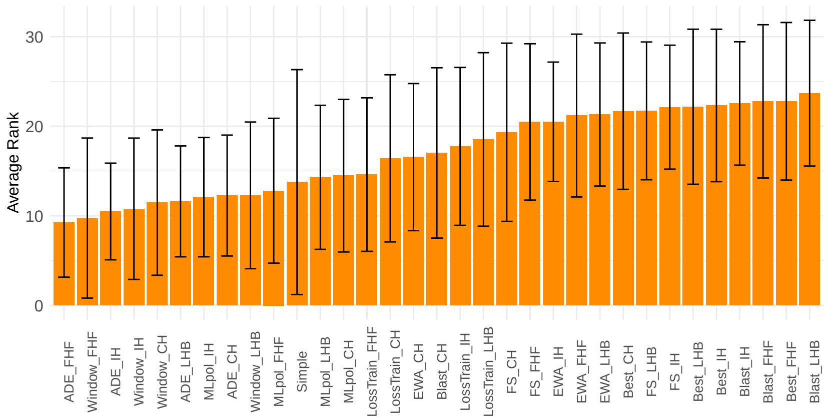

Fig. 1. Average rank, and respective standard deviation, of each method across all time series.

different scores. For example, ADE shows better performance relative to Best irrespective of the weighting strategy. The two best forecast combination methods, ADE and Window, show similar behavior in terms of relative scores for the different weighting strategies. Their best score is achieved with FHF, followed by IH. Conversely, the worst combination approaches, namely Blast, Best, EWA, and FS, have their best score when coupled with CH.

RQ3 So far, the results were analyzed according to the average rank. However, average rank ignores the magnitude of differences in predictive performance [4]. To overcome this limitation, we also study the percentage difference in performance between each method and a reference method. We set Simple as the reference method, which assigns equal weights to the models in the ensemble.

For each method m, the percentage difference is computed as follows.

$$100 \times \frac{\mathrm{MAE}_{\mathtt{m}} - \mathrm{MAE}_{\mathtt{Simple}}}{\mathrm{MAE}_{\mathtt{Simple}}}$$

where $\mathrm{MAE}_{\mathtt{m}}$ and $\mathrm{MAE}_{\mathtt{Simple}}$ represent the MAE of method m and Simple, respectively. Negative values denote better performance by method m.

Figure 3 depicts the distribution of the percentage difference in MAE between each method and Simple. The methods are ordered by decreasing median percentage difference in MAE.

The order of the methods is similar to that obtained in the previous analyzes. Only ADE_FHF and Window_FHF show a median performance difference below zero. This indicates that Simple outperforms the other dynamic combination methods more times than not.

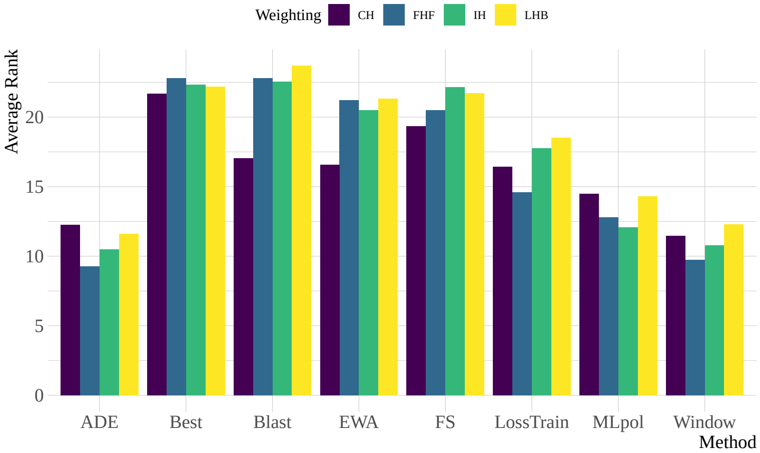

Fig. 2. Average rank of each method, aggregated by weighting strategy.

RQ4 The analysis presented so far quantifies the performance of each method for long-term forecasting. Specifically, predicting 18 step in advance. However, long-term forecasts are typically less accurate than short-term ones. We analyzed the impact of the forecasting horizon in the results obtained.

Figure 4 shows the average (median) percentage difference of each method relative to `Simple` over the forecasting horizon. As before, the average is computed across the 3568 time series.

The figure shows a clear trend which indicates that the methods decrease their performance relative to `Simple` as the forecasting horizon increases. For *t+1* (one-step ahead forecasting), 5 out of 8 combination methods outperform `Simple`. But, for long-term forecasting (*t+18*), only two methods achieve better performance. It is also interesting to note that there are slight changes in the relative performance of methods. For example, for *t+1* `Window_FHF` is only the fourth best approach (`ADE_FHF` shows the best average rank). However, by *t+18*, `Window_FHF` shows the best score.

5 Discussion

This paper investigates the performance of several dynamic ensembles for multi-step ahead forecasting problems. We focus on ensembles composed of multi-output models. Notwithstanding, the combination rules tested in this work can also be applicable to models following a different strategy regarding multi-step ahead prediction (c.f. Section 2.3).

The main motivation for this work is that the literature concerning dynamic ensembles for forecasting is focused on one-step ahead predictions. Most ap-

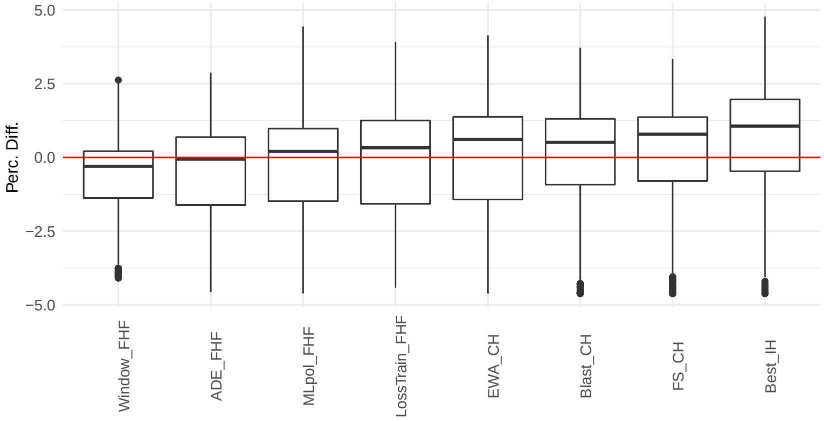

Fig. 3. Distribution of percentage difference in MAE between each method and `Simple` across all time series. Negative values denote better performance of the respective method.

proaches assume immediate feedback from the environment to compute the error of each method. Previous works concerning dynamic ensembles for multi-step ahead forecasting are scarce. This gap raised important questions about how to properly weigh individual models across extended forecasting horizons.

The results from the experiments provide several insights. We found that while all methods demonstrated considerable variability across different time series, variants of `ADE` and `Window` consistently achieved the best average rank (RQ1). This suggests these approaches are more robust when applied to multi-step forecasting tasks. Regarding weight computation across the forecasting horizon (RQ2), our comparison of four different approaches revealed no statistically significant differences in performance. However, the `FHF` strategy (which propagates weights computed for one-step ahead predictions) maximized the effectiveness of the best-performing methods.

When compared to a static ensemble using equal weights for all models (RQ3), only two methods (`ADE_FHF` and `Window_FHF`) showed systematic performance improvements. This finding is particularly noteworthy as it suggests that many dynamic approaches fail to outperform the simple averaging approach in multi-step contexts. We also discovered that the forecasting horizon significantly affects the relative performance of dynamic ensembles. All methods decrease their performance relative to `Simple` as the forecasting horizon increases (RQ4).

These findings have important practical implications. For short-term forecasting applications, dynamic ensembles such as `ADE_FHF` or `Window_FHF` may be advantageous. However, for longer forecasting horizons, competitive results can be obtained with simpler approaches such as equal-weight averages.

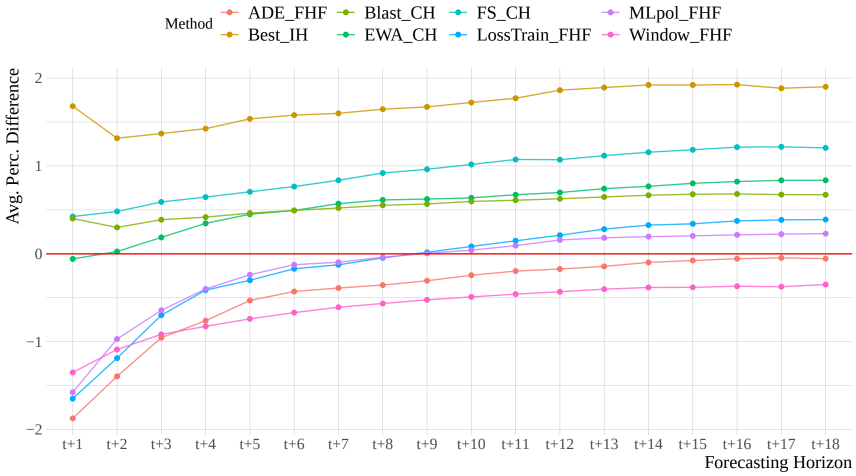

Fig. 4. Median percentage difference of each method relative to `Simple` in each forecasting horizon.

We remark that the results obtained may be dependent on the particular experimental setup used, including hyperparameters (e.g. forecasting horizon) and forecasting methods. We focus on various machine learning multi-output regression algorithms employed using auto-regression. Nonetheless, many other forecasting methods exist in the literature, from classical forecasting approaches such as ARIMA to various neural networks architectures that have been showing state-of-the-art performance in benchmark datasets.

Acknowledgements

This work is a result of Agenda "Center for Responsible AI", nr. C645008882-00000055, investment project nr. 62, financed by the Recovery and Resilience Plan (PRR) and by European Union - NextGeneration EU. Funded by the European Union – NextGenerationEU. Funded by the European Union under the Horizon Europe Framework Programme Grant Agreement Nº: 101095387. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Health and Digital Executive Agency can be held responsible for them; UID/00027 of the LIACC - Artificial Intelligence and Computer Science Laboratory - funded by Fundação para a Ciência e a Tecnologia, I.P./ MCTES through the national funds. The work of L. Torgo was undertaken, in part, thanks to funding from the Canada Research Chairs program.

Declaration: The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Aiolfi, M., Timmermann, A.: Persistence in forecasting performance and conditional combination strategies. Journal of Econometrics **135**(1), 31–53 (2006)
2. Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D.C., Rangapuram, S., Salinas, D., Schulz, J., et al.: Gluonts: Probabilistic and neural time series modeling in python. Journal of Machine Learning Research **21**(116), 1–6 (2020)
3. Bates, J.M., Granger, C.W.: The combination of forecasts. Journal of the Operational Research Society **20**(4), 451–468 (1969)
4. Benavoli, A., Corani, G., Mangili, F.: Should we really use post-hoc tests based on mean-ranks? The Journal of Machine Learning Research **17**(1), 152–161 (2016)
5. Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity creation methods: a survey and categorisation. Information Fusion **6**(1), 5–20 (2005)
6. Brown, G., Wyatt, J.L., Tiňo, P.: Managing diversity in regression ensembles. Journal of machine learning research **6**(Sep), 1621–1650 (2005)
7. Cerqueira, V., Torgo, L., Oliveira, M., Pfahringer, B.: Dynamic and heterogeneous ensembles for time series forecasting. In: 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA). pp. 242–251 (Oct 2017)
8. Cerqueira, V., Torgo, L., Mozetič, I.: Evaluating time series forecasting models: An empirical study on performance estimation methods. Machine Learning **109**(11), 1997–2028 (2020)
9. Cerqueira, V., Torgo, L., Pinto, F., Soares, C.: Arbitrage of forecasting experts. Machine Learning **108**(6), 913–944 (2019)
10. Cesa-Bianchi, N., Lugosi, G.: Prediction, Learning, and Games. Cambridge University Press, New York, NY, USA (2006)
11. Clemen, R.T., Winkler, R.L.: Combining economic forecasts. Journal of Business & Economic Statistics **4**(1), 39–46 (1986)
12. De Stefani, J., Le Borgne, Y.A., Caelen, O., Hattab, D., Bontempi, G.: Batch and incremental dynamic factor machine learning for multivariate and multi-step-ahead forecasting. International Journal of Data Science and Analytics **7**(4), 311–329 (2019)
13. Hibon, M., Evgeniou, T.: To combine or not to combine: selecting among forecasts and their combinations. International journal of forecasting **21**(1), 15–24 (2005)
14. Kuncheva, L.I.: Classifier ensembles for changing environments. In: International Workshop on Multiple Classifier Systems. pp. 1–15. Springer (2004)
15. Newbold, P., Granger, C.W.: Experience with forecasting univariate time series and the combination of forecasts. Journal of the Royal Statistical Society. Series A (General) pp. 131–165 (1974)
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)
17. Picard, R.R., Cook, R.D.: Cross-validation of regression models. Journal of the American Statistical Association **79**(387), 575–583 (1984)
18. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains: a review and perspectives. Journal of Artificial Intelligence Research **70**, 683–718 (2021)

19. van Rijn, J.N., Holmes, G., Pfahringer, B., Vanschoren, J.: Having a blast: Meta-learning and heterogeneous ensembles for data streams. In: Data Mining (ICDM), 2015 IEEE International Conference on. pp. 1003–1008 (Nov 2015)
20. Taieb, S.B., Bontempi, G., Atiya, A.F., Sorjamaa, A.: A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. Expert systems with applications **39**(8), 7067–7083 (2012)
21. Venkatraman, A., Hebert, M., Bagnell, J.A.: Improving multi-step prediction of learned time series models. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
22. Wolpert, D.H.: Stacked generalization. Neural networks **5**(2), 241–259 (1992)
23. Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural computation **8**(7), 1341–1390 (1996)