Machine Learning (2019) 108:913–944
https://doi.org/10.1007/s10994-018-05774-y

# Arbitrage of forecasting experts

Vitor Cerqueira[1,2] · Luís Torgo[1,2,3] · Fábio Pinto[2] · Carlos Soares[1,2]

Received: 11 March 2018 / Accepted: 20 November 2018 / Published online: 4 December 2018

## Abstract
Forecasting is an important task across several domains. Its generalised interest is related to the uncertainty and complex evolving structure of time series. Forecasting methods are typically designed to cope with temporal dependencies among observations, but it is widely accepted that none is universally applicable. Therefore, a common solution to these tasks is to combine the opinion of a diverse set of forecasts. In this paper we present an approach based on arbitrating, in which several forecasting models are dynamically combined to obtain predictions. Arbitrating is a metalearning approach that combines the output of experts according to predictions of the loss that they will incur. We present an approach for retrieving out-of-bag predictions that significantly improves its data efficiency. Finally, since diversity is a fundamental component in ensemble methods, we propose a method for explicitly handling the inter-dependence between experts when aggregating their predictions. Results from extensive empirical experiments provide evidence of the method's competitiveness relative to state of the art approaches. The proposed method is publicly available in a software package.

**Keywords** Dynamic ensembles · Metalearning · Time series · Combining expert advice · Forecasting · Dependency and diversity

Editor: Gavin Brown.

✉ Vitor Cerqueira
vitor.cerqueira@fe.up.pt

Luís Torgo
ltorgo@dcc.fc.up.pt

Fábio Pinto
fabiohscpinto@gmail.com

Carlos Soares
csoares@fe.up.pt

[1] INESC TEC, Porto, Portugal

[2] University of Porto, Porto, Portugal

[3] Dalhousie University, Halifax, Canada

## 1 Introduction

Time series is an important topic in several research communities. The generalised interest in time series arises from the dynamic characteristics of many real-world phenomena. Uncertainty is a major issue in these problems, which complicates the exact understanding of their future behaviour. This is the key motivation for the study of forecasting methods.

Organisations across a wide range of domains rely on forecasting as a decision support tool. For example, financial analysts forecast the behaviour of stock prices for economic profit. Intelligent transportation systems forecast the short-term traffic flow to enhance the operational efficiency in road networks.

In the last few decades the research community produced a considerable number of contributions on forecasting methods. These have been designed to cope with the time dependency of the data. Time series often comprise non-stationarities and time evolving complex structures, also known as concept drift (Gama et al. 2014), which hamper the forecasting process.

One of the most common approaches to forecasting is the dynamic combination of several experts, i.e., dynamic ensemble methods. Ensemble methods have been shown to provide a superior predictive performance relative to single learning algorithms (Brown et al. 2005). Notwithstanding, selecting the weights of each individual expert in the combination rule is known to be a difficult task.

The state of the art approaches for dynamically combining experts for forecasting are mostly based on estimates of predictive performance. The loss of each expert is tracked over time and used to combine them in an adaptive way. Some of these approaches have interesting theoretical loss upper bounds based on regret minimisation (Cesa-Bianchi and Lugosi 2006).

Metalearning approaches are also commonly used. For example stacking (Wolpert 1992), which directly models inter-dependencies between experts. This characteristic may be important to take into account the diversity among experts, which is a key component in ensemble learning (Brown et al. 2005).

In this paper we present a metalearning strategy to combine the available forecasting models in an adaptive way. However, contrary to stacking, we separately model the individual expertise of each forecasting model and specialise them across the time series. Consequently, the forecasting models are combined in such a way that they are only selected for predicting examples that they are expected to be good at. Moreover, as opposed to tracking the error on past instances, our combination approach is more proactive as it is based on predictions of future loss of models. This can result in a faster adaptation to changes in the environment.

The motivation for our approach is that different learning models have different areas of expertise across the input space. In time series forecasting there is evidence that forecasting models have a varying relative performance over time (Aiolfi and Timmermann 2006). Moreover, it is also common for the underlying process generating the time series to have recurrent structures due to factors such as seasonality (Gama and Kosina 2014). In this context, we hypothesise that the arbitrage metalearning strategy enables the ensemble to better detect changes in the relative performance of models or changes between different regimes and quickly adapt itself to the environment.

The proposed metalearning strategy, hereby denoted as Arbitrated Dynamic Ensemble (ADE), is based on arbitrating (Ortega et al. 2001), a method from the family of mixture of experts (Jacobs et al. 1991). A meta-learner is created for each base-learner that is part of the ensemble. Each meta-learner is specifically designed to model how apt its base counterpart is to make a prediction for a given test example. This is accomplished by analysing how the error incurred by a given learning model relates to the characteristics of the data. At test time,

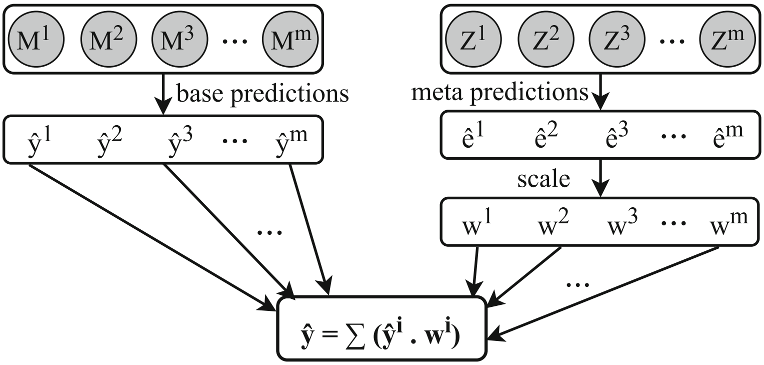

**Fig. 1** Workflow of ADE for a new prediction. The base-learners $M$ produce the predictions $\hat{y}^i, i \in \{1, \ldots, m\}$ for the next value of the time series. In parallel, the meta-learners $Z$ produce the weights $w^i$ of each base-learner according to the predictions of their error ($\hat{e}^i$). The final prediction $\hat{y}$ is computed using a weighted average of the predictions relative to the weights

the base-learners are weighted according to their expected degree of competence in the input observation, estimated by the predictions of the meta-learners. This is illustrated in Fig. 1.

While a given base-learner $M^i$ is trained to model the future values of the time series, its metalearning associate $Z^i$ is trained to model the error of $M^i$. The arbiter $Z^i$ then can make predictions regarding the error that $M^i$ will incur when predicting the future values of the time series. The larger the estimates produced by $Z^i$ (relative to the other models in the ensemble) the lower the weight of $M^i$ will be in the combination rule.

Diversity among the experts is a fundamental component in building ensemble methods (Brown et al. 2005). We start by addressing this issue implicitly, by using experts with different learning strategies, i.e. heterogeneous ensembles. Our assumption is that the ensemble heterogeneity is useful to cope with the different dynamic regimes of time series. Besides heterogeneity we encourage diversity explicitly during the aggregation of the output of experts. This is achieved by taking into account not only predictions of performance produced by the arbiters, but also the correlation among experts in a recent window of observations.

We validate the proposed method in 62 real-world time series. Empirical experiments suggest that our method is competitive with different adaptive methods for combining experts and other metalearning approaches such as stacking (Wolpert 1992). In the interest of reproducible research, ADE is publicly available as an R software package.[1] Moreover, all experiments reported in the paper are also reproducible.[2]

In summary, the contributions of this paper are:

- ADE, a method for the arbitrage of forecasting experts;
- The introduction of a blocked prequential procedure in the arbitrage approach to obtain out-of-bag predictions in the training set in order to increase the data used to train the metalearning models;
- A sequential re-weighting strategy for controlling the redundancy among the output of the experts using their correlation in a recent window of observations;
- An extensive empirical study encompassing: statistical comparisons with state of the art approaches; analysis on the different deployment strategies of the proposed method;

[1] **tsensembler**: on CRAN or at https://github.com/vcerqueira/tsensembler.

[2] Instructions at: https://github.com/vcerqueira/forecasting_experiments.

sensitivity analysis on the main parameters of the proposed method; relative scalability analysis in terms of execution time; and a study on the value of increasing the number of experts in the ensemble.

We start by reviewing the related work in Sect. 2. The methodology is addressed in Sect. 3, where we formalise ADE and our contributions. The experiments and respective results are presented in Sect. 4, which includes the comparisons to the state of the art. The results are discussed in Sect. 5. Section 6 concludes the paper.

## 2 Related work

In this section we review the literature related to our work. First we explain the position of the proposed method in the literature (Sect. 2.1). Then we briefly describe the state of the art methods for dynamically combining expert outputs, both using windowing and metalearning approaches (Sects. 2.2, 2.3). We list their characteristics and limitations as well as highlight our contributions. Particularly in the latter, we overview previous publications that led to this work. Finally, we briefly overview the typical approaches for encouraging diversity in ensemble methods (Sect. 2.4).

### 2.1 Dynamic combiners

Dynamic ensemble methods for forecasting is a well studied topic in the literature. For example, Clemen (1989) presented an annotated bibliography comprising over 200 approaches.

This work is focused on the application of dynamic combination approaches for numerical and univariate time series forecasting tasks. According to the taxonomy presented by Kuncheva (2004), our approach can be regarded as a *dynamic combiner* one. This type of strategies builds the experts in advance. The ensemble then adapts to concept drift by dynamically changing the combination rule.

### 2.2 Windowing strategies for expert combination

Combining different experts is a difficult task, and several methods have been proposed to accomplish this. Particularly in forecasting, the simple average of the available experts (equal weights) has been shown to be a robust combination method (Clemen and Winkler 1986). Its competitive performance relative to approaches using estimated weights is known in the forecasting literature as the "forecasting combination puzzle" (Genre et al. 2013). Nonetheless, more sophisticated approaches have been proposed.

Simple averages are sometimes complemented with model selection before aggregation, also known as trimmed means. For example, Jose and Winkler (2008) propose trimming a percentage of the worst forecasters in past data, and average the output of the remaining experts.

One of the most common and successful approaches to combine predictive models in time dependent data is to weight them according to their performance. Typically the performance is determined on a window of recent data, or by using some other forgetting mechanism that promotes the importance of recency. The idea is that recent observations are more similar to the one we intend to predict, and thus they are considered more relevant. For example, Newbold and Granger (1974) use this approach for combining forecasters models. More recently, van Rijn et al. (2018) proposed a method for data streams classification. As opposed to fusing experts, they select the best recent performing one to classify the next observation.

AEC is a method for adaptively combining forecasters (Sánchez 2008). It uses an exponential re-weighting strategy to combine forecasters according to their past performance, including a forgetting factor to give more importance to recent values. Timmermann argues that for the prediction of stock returns models have only short-lived periods of predictability (Timmermann 2008). He proposes an adaptive combination based on the recent $R^2$ of forecasters. If all models have poor explained variance (low $R^2$) in the recent observations then the forecast is set to the mean value of those observations. Otherwise, the experts are combined by averaging their predictions with the arithmetic mean.

In online learning, several strategies have been proposed for aggregating experts advice. These are typically based on regret minimisation, and have interesting theoretical properties. Regret is the average error suffered with respect to the best we could have obtained. In this paper we focus on three of the following approaches: the exponentially weighted average, the polynomially weighted average, and the fixed share aggregation. For a thorough review of these methods we refer to the seminal work by Cesa-Bianchi and Lugosi (2006). Zinkevich (2003) proposed an online convex programming approach based on gradient descent that also guarantees regret bounds.

The outlined models are related to our work in the sense that they employ adaptive heuristics to combine forecasters. However, these heuristics are incremental or sliding summary statistics on relative past performance. Our intuition is that these approaches have a short memory and may fail to capture long-range relationships between changes in the underlying time series and the performance of the experts efficiently. Conversely, we explore differences among experts to specialise them across the data space based on a regression analysis. Moreover, we use a more proactive heuristic that is based on the prediction of relative future performance of individual forecasters.

### 2.3 Metalearning strategies for expert combination

Metalearning provides a way for modelling the learning process of a learning algorithm (Brazdil et al. 2008). Several methods use this approach to improve the combination or selection of models (Pinto et al. 2016; Rossi et al. 2014; Todorovski and Džeroski 2003; Wolpert 1992).

A popular and successful approach for dynamically combining experts is to apply multiple regression on the output of the experts. For example, Gaillard and Goude (2015) describe a setup in which Ridge regression is used to aggregate experts by minimising the L2-regularised least-squares. The idea behind these approaches is similar to stacking (Wolpert 1992), a widely used approach to combine predictive models.

Our proposal follows a metalearning strategy called arbitrating. This approach was introduced before for dynamic selection of classifiers (Ortega et al. 2001). A prediction is made using a combination of different classifiers that are selected according to their expertise concerning the input data. The expertise of a model is learned using a meta-learner, one for each available base classifier, which models the confidence of its base counterpart. At runtime, the classifier with the highest confidence is selected to make a prediction.

The initial indication that arbitration produced interesting results in forecasting was evidenced in a case study regarding solar radiation forecasting (Cerqueira et al. 2017). In that work, the arbitration mechanism was adapted straightforwardly, showing an improvement over stacking (Wolpert 1992).

The proposed dynamic ensemble method ADE was first introduced in a previous work (Cerqueira et al. 2017a). The idea behind arbitration was reworked and applied to time series forecasting problems from several domains. Several of its drawbacks were addressed, such as the inefficient use of the available data, by using out-of-bag samples from the training set; a more robust combination rule by using a committee of recent well performing models; and the general translation to the time series forecasting tasks, which is fundamentally different than classification tasks. In this paper we extend and improve the approach. The main difference is a diversity inducing procedure during expert aggregation that explicitly models their inter-dependence. On top of this, we significantly enlarge the experiments used to validate the method. We also provide an in-depth analysis of ADE, to provide more insight about its characteristics.

### 2.3.1 Mixture of experts

The proposed dynamic ensemble is related to mixture of experts (Jacobs et al. 1991) (ME), in the sense that each expert is specialised in a certain region of the input space. The main difference to ME is the way the weights of the experts are computed. ME estimate the weights using a gating function. The gating function is typically a neural network with as many output units as experts and trained using Expectation–Maximisation. Our approach uses a set of arbiters that predict the loss of the experts. ADE also differs in the training procedure of the experts and how diversity is encouraged in the ensemble. ME are typically comprised by neural network experts built incrementally, and the gating function explicitly controls the patterns each neural network learns according to their relative performance. This results in relatively independent experts. Conversely, ADE works as a dynamic combiner approach (Kuncheva 2004). Diversity is introduced implicitly by employing a set of heterogeneous experts, which are trained with the whole set of available observations. During expert aggregation, diversity is also encouraged by considering the redundancy among the output of the experts.

## 2.4 Diversity creation methods

A wide range of contributions exist for encouraging diversity in ensemble methods. These are typically based on input manipulation [e.g. bagging (Breiman 1996)], output manipulation [e.g. Error-Correcting Output Coding (Dietterich and Bakiri 1991)], or manipulation of architectures used to build experts. For a comprehensive read on diversity creation approaches we refer to the survey by Brown et al. (2005).

We propose a method that encourages diversity during the aggregation of experts. This is accomplished by manipulating the experts' weights according to the redundancy of their output. To the best of our knowledge, there is no closely related approach in the machine learning literature. However, our approach is inspired on the notions of *diversity* in the context information retrieval. An example is the seminal approach Maximal Marginal Relevance (Carbonell and Goldstein 1998). This method is typically used to rank a list of documents to answer a given query by considering not only the relevance of each document individually, but also their redundancy to documents already ranked.

# 3 Arbitrated dynamic ensemble

In this section we formalise ADE. We start by describing the predictive task, and then explain the different steps of the methodology.

A time series $Y$ is a temporal sequence of values $Y = \{y_1, y_2, \ldots, y_t\}$, where $y_i$ is the value of $Y$ at time $i$. We focus on numeric time series, i.e., $y_i \in \mathbb{R}$, $\forall\ i \in \{1, \ldots, t\}$. We frame the problem of time series forecasting as a regression task. The temporal dependency is modelled by having the previous observations as attributes in the learning of the experts. In order to enhance the representation of the time series, this approach can be extended by using summary statistics on the embedding vectors, or other external domain-specific knowledge.

To be more precise, we use time delay embedding (Takens 1981) to represent $Y$ in an Euclidean space with embedding dimension $K$. Effectively, we construct a set of observations which are based on the past $K$ lags of the time series. Each observation is composed of a feature vector $x_i \in \mathbb{X} \subset \mathbb{R}^K$, which denotes the previous K values, and a target vector $y_i \in \mathbb{Y} \subset \mathbb{R}$, which represents the value we want to predict. The objective is to construct a model $f : \mathbb{X} \rightarrow \mathbb{Y}$, where $f$ denotes the regression function.

The proposed methodology in ADE for time series forecasting settles on the following three main steps:

- Training of the base-learners: the set of heterogeneous experts that are used to forecast future values of Y;
- Training the meta-learners: arbiters that model and predict the loss of the experts;
- Predicting $y_{t+1}$: Combining the output of the experts according to the output of the arbiters and the correlation among the output of the experts to forecast the next value of the time series.

### 3.1 Training the experts

The first step of ADE is to train $m$ individual forecasters. Each $M^j$, $\forall j \in \{1, \ldots, m\}$ is built using the available time series $Y$. The objective is to predict $y_{t+1}$, the next value of $Y$. This is accomplished by having experts build the model $f : \mathbb{X} \rightarrow \mathbb{Y}$.

$M$ is comprised by a set of heterogeneous models, for example decision trees and artificial neural networks. Heterogeneous models have different inductive biases and assumptions regarding the process generating the data. Effectively, we expect models to have different expertise across the time series. Later we will present an approach complementary to ensemble heterogeneity that encourages diversity during the aggregation of the experts (Sect. 3.3.3).

### 3.2 Training the arbiters

In the metalearning step of ADE the goal is to build models capable of modelling the expertise of each base-learner across the input space.

Our assumption is that not all models will perform equally well at any given prediction point. This idea is in accordance with findings reported in prior work (Aiolfi and Timmermann 2006). Systematic evidence was found that some models have varying relative performance over time and that other models are persistently good (or bad) throughout the time series. Furthermore, in many environments the dynamic concepts have a recurring nature, due to, for example, seasonality. These findings can be regarded as instances of the No Free Lunch theorem presented by Wolpert (2002). This theorem essentially states that no learning algorithm is the most appropriate for all tasks.

In effect, we use metalearning to dynamically weigh base-learners and adapt the combined model to changes in the relative performance of the base models, as well as for the presence of different regimes in the time series.

Our metalearning approach is based on an arbitrating architecture (Ortega et al. 2001) and mixture of experts (Jacobs et al. 1991). Specifically, a meta-learner $Z^j, \forall\ j \in \{1, \ldots, m\}$ is trained to build the following model:

$$e_i^j = f(x_i) \tag{1}$$

where $e_i^j$ is the absolute error incurred by $M^j$ in an observation $(x_i, y_i)$. We formalise the metalearning problem using the same feature set used by the experts to predict the future values of the time series.

We perform this regression analysis on a meta-level to understand how the error of a given model relates to the dynamics and the structure of the time series. Effectively, we can capitalise on this knowledge by dynamically combining base-learners according to the expectation of how they will perform.

### 3.2.1 Blocked prequential for out-of-bag predictions

Typical metalearning approaches for dynamic model selection or combination, only start the metalearning layer at run-time. This is the case of, for example, the original arbitrating formulation by Ortega et al. (2001) or the work of Gama and Kosina (2014). This is motivated by the need for unbiased samples to build reliable meta-learners. However, this means that at the beginning, few observations are available to train the meta-learners, which might result in under-fitting.

ADE uses the training set to produce out-of-bag predictions which are then used to compute an unbiased estimate of the loss of each base-learner. By retrieving out-of-bag samples from the training set we are able to significantly increase the amount of data available to the meta-learners. We hypothesise that this strategy improves the overall performance of the ensemble by improving the accuracy of each meta-learner.

We produce out-of-bag samples by running a blocked prequential procedure (Dawid 1984), a growing window approach. The available embedded time series used for training is split into $b$ equally-sized and sequential blocks of contiguous observations. In the first iteration, the first block is used to train the base-learners $M$ and the second is used to test them. Then, the second block is merged with the first one for training $M$ and the third block is used for testing. This procedure continues until all blocks are tested (except the first one). In summary, using out-of-bag samples allows using the available data to train both the experts (as described above) and the arbiters. This results in a more efficient use of the available time series, because it is used to fit both the experts and the arbiters. This data efficiency, and the preservation of the temporal order of observations was the main motivation for using the blocked prequential with a growing window. The meta-learning phase is described in Algorithm 1.

## 3.3 Predicting $y_{t+1}$

For predicting the next value of the time series, $y_{t+1}$, ADE combines the output of the experts $M$ according to the output of the arbiters and the recent correlation among the experts.

### 3.3.1 Committee of models for prediction

In the original arbitrating architecture the expert with the highest confidence (predicted by the arbiters) is selected to make a prediction. Our approach is to combine the output of the experts, as opposed to selecting a single one.

**Algorithm 1:** Training arbiters

```
input  : $Y$
input  : $M$
output : Set of arbiters Z
foreach $M^j$ in $M$ do
    $\hat{y}^j \leftarrow$ BlockedPrequential($M^j, Y, b$) // Retrieve out-of-bag predictions of the
        experts from the available $Y$
    $e_i^j = |y_i - \hat{y}_i^j|$ // Expert absolute loss in out-of-bag samples $y_i \in Y$
    $Z^j \leftarrow e_i^j = f(x_i)$ // training meta-model $Z^j$
end
Return Z
```

As described earlier, the predictive performance of forecasting models has been reported to vary over a given time series. We address this issue with a committee of models, where we trim recently poor performing models from the combination rule for an upcoming prediction [e.g. trimmed means (Jose and Winkler 2008)].

As we explain in Sect. 2, the state of the art approaches for dynamic combination in time series rely on past performance to quantify the weight of the experts. Specifically, this is typically used for dynamic selection (e.g. Jose and Winkler 2008) or dynamic combination (e.g. Newbold and Granger 1974). Here we use this information for dynamic selection. Formally, we select the $\Omega\%$ base-learners with lowest mean absolute error in the last $\lambda$ observations (${}^{\Omega}M$), suspending the remaining ones. The predictions of the meta-level models (${}^{\Omega}Z$) are used to weigh the selected forecasters.

In summary, if we expect $M^j$ to make a large error $e^j$ in a given observation relative to the other experts, we assign it a small weight—or even suspending it—in the final prediction. Conversely, if we expect $M^j$ to incur a small loss relative to its peers, we increase its weight for the upcoming prediction.

### 3.3.2 Combining the experts

The weigh of an expert $M^j$ in ${}^{\Omega}M$ is determined by a simple transformation of the predicted loss by the arbiters ${}^{\Omega}Z$. This is formalised by the following equation:

$$w_{t+1}^j = \frac{scale\left(-\hat{e}_{t+1}^j\right)}{\sum_{j \in {}^{\Omega}M} scale\left(-\hat{e}_{t+1}^j\right)} \tag{2}$$

where $\hat{e}_{t+1}^j$ is the prediction made by ${}^{\Omega}Z^j$ for the absolute loss that ${}^{\Omega}M^j$ will incur in $y_{t+1}$; $w_{t+1}^j$ is the weigh of $M^j$ for observation $y_{t+1}$; and scale denotes the min–max scaling function used to transform the vector of predicted loss into a 0–1 scale. The normalisation with respect to the summation in Eq. 2 is performed so that the combination is convex, i.e., the weights sum to 1. The experts that are suspended (Sect. 3.3.1) are simply assigned a weight of 0.

### 3.3.3 Sequential re-weighting of experts

Most combination approaches, dynamic ones particularly, weigh experts by maximising estimates of predictive performance (c.f. Sect. 2). However, in cases where the experts are highly redundant it is important to model their inter-dependence.

Brown et al. (2005) stress that the diversity among experts is a critical component for increasing the ensemble's predictive performance. To address this problem, Jacobs (1995) points out that ensemble methods require:

a. "training procedures that result in relatively independent experts";
b. "aggregation methods that explicitly or implicitly model the dependence among the experts".

We address the first issue (a.) implicitly by focusing on heterogeneous ensembles. These are comprised by experts with different inductive biases. The second issue (b.) is addressed explicitly by re-weighting the experts at each prediction point according to their recent correlation.

For clarity, we have at this point and for a given time instance $y_{t+1}$:

- the output of the experts $\hat{y}^M_{t+1} = \{\hat{y}^1_{t+1}, \ldots, \hat{y}^m_{t+1}\}$;
- and their respective weights predicted by the arbiters and scaled accordingly: $w^M_{t+1} = \{w^1_{t+1}, \ldots, w^m_{t+1}\} : \sum_{i=1}^m w^i_{t+1} = 1$.

To model the inter-dependence among experts we frame their aggregation as a ranking task, in which experts are ranked sequentially by their decreasing weight (the one predicted to perform better is ranked first). The intuition for the ranking approach is borrowed from the information retrieval literature. For example, the algorithm Maximal Marginal Relevance (Carbonell and Goldstein 1998) ranks a list of documents to answer a given query by maximising a function that couples the relevance and redundancy of documents. As such, the value of the second most relevant document (with respect to a given query) also depends on its redundancy to the most relevant document. The point is to emphasise the novelty of information in the document set and enhance their complementarity.

Notwithstanding, time series comprise characteristics that this type of methods need to cope with, e.g. the variance in relative performance that forecasters show over a time series. We formalise our idea for the dynamic combination of forecasting experts in Algorithm 2. We use the correlation among the output of the experts to quantify their redundancy. This correlation is computed in a window of recent observations to cope with eventual non-stationarities of time series.

A given expert $i$ is penalised for its correlation to each expert $j$ already ranked. This penalty is determined by the multiplication of the correlation and the weights of expert $i$ and expert $j$ (line 8). The penalty formula takes a multiplication because its elements work on one another: if an expert $M^i$ is fully correlated with other experts already ranked ($M^j \in {}^{\Omega}M^j \setminus {}^{\Omega}M^i : w^j > w^i$), its weight is absorbed by the latter and $M^i$'s weight becomes zero. Conversely, if $M^i$ is completely uncorrelated with its ranked peers, $M^i$ is ranked with its original weight. In summary, this approach allows the control of redundant information in the output of the experts. A practical advantage of this method is that it requires no parameter tuning, except for the correlation function.

The final prediction is the weighted average of the predictions made by the experts $\hat{y}^j$ with respect to their re-weighted relevance $w'^j_{t+1}$ (Algorithm 3):

$$\hat{y}_{t+1} = \sum_{j \in {}^{\Omega}M} \hat{y}^j_{t+1} \cdot w'^j_{t+1} \tag{3}$$

**Algorithm 2:** Sequential re-weighting of experts

```
input  : predictions of experts in the last $\lambda$ observations: $\hat{y}^{M}_{(t-\lambda):(t+1)}$
input  : weight of experts for t+1: W
output: re-estimated weights $W'$
W ← Sort(W, decreasing) // sort weights in decreasing order
$W' \leftarrow \{\}$ // List with final weights
$W'_1 \leftarrow W_1$ // First element of $W'$ is the weight of the predicted to be
    the best expert
foreach remaining expert i in W do
    $W'_i \leftarrow W_i$
    foreach expert j in $W'$ do
        $cor_{ij} \leftarrow \mathrm{Cor}(\hat{y}^{i}_{(t-\lambda):(t+1)}, \hat{y}^{j}_{(t-\lambda):(t+1)})$ // Correlation between the
            predictions of expert i and expert j in the last $\lambda$
            observations
        $\eta_{ij} \leftarrow cor_{ij} \cdot W'_j \cdot W'_i$ // penalty that expert j applies to expert i
        $W'_j \leftarrow W'_j + \eta_{ij}$
        $W'_i \leftarrow W'_i - \eta_{ij}$
    end
end
return $W'$
```

**Algorithm 3:** Forecasting $\hat{y}_{t+1}$

```
input  : Time series $Y$ up to time $t$
input  : Experts M
input  : Arbiters Z
input  : Committee ratio $\Omega$
input  : Window size $\lambda$
output: $\hat{y}_{t+1}$
${}^{\Omega}M \leftarrow$ Subset(M, $\lambda$, $\Omega$)
${}^{\Omega}Z \leftarrow$ Subset(Z, $\lambda$, $\Omega$) // Form the committees ${}^{\Omega}M$ and ${}^{\Omega}Z$ according to
    performance on the last $\lambda$ observations
Get loss predictions $\hat{e}^{j}_{t+1}$ from $Z^j \in {}^{\Omega}Z$
Compute weights $w^{j}_{t+1} = scale(-\hat{e}^{j}) / \sum_{j \in {}^{\Omega}M} scale(-\hat{e}^{j})$
Get predictions $\hat{y}^{j}_{(t-\lambda):(t+1)}$ from $M^j \in {}^{\Omega}M$ // Predictions of selected experts in
    the last $\lambda$ observations
Apply Algorithm 2 to weights: $w'^{j}_{t+1} \leftarrow$ SequentialReweight($\hat{y}^{j}_{(t-\lambda):(t+1)}, w^{j}_{t+1}$) // Calibrate
    weights according to expert's correlation
Compute final prediction $\sum_{j:M^j \in {}^{\Omega}M} \hat{y}^{j}_{t+1} \cdot w'^{j}_{t+1}$
```

## 4 Experiments

In this section we present the experiments carried out to validate ADE. We start by describing the overall setup. We compare the proposed method to state of the art approaches for combining the output of experts. Specifically, we focus on approaches designed to cope with temporal dependencies. Afterwards, we perform sensitivity analyses to enhance our understanding of the components of ADE. To encourage reproducible research, we published the code used to perform these experiments (c.f. footnote 2).

The experiments were designed to answer the following research questions:

Q1: How does the performance of the proposed method compares to the performance of the state-of-the-art methods for time series forecasting tasks and state of the art methods for combining forecasting models?

Q2: Is it beneficial to use a weighing scheme in our arbitrating strategy instead of selecting the predicted best expert as originally proposed (Ortega et al. 2001)?

Q3: Is it beneficial to use out-of-bag predictions from the training set to increase the data used to train the meta-learners?

Q4: How does the performance of ADE vary by the introduction of a committee, where poor recent base-learners are discarded from the upcoming prediction, as opposed to weighing all the models?

Q5: What is the impact of the sequential re-weighting procedure in ADE's performance?

Q6: How does the performance of ADE vary by using different updating strategies for the base and meta models?

Q7: How sensitive is ADE to the parameters $\Omega$ and $\lambda$, and to the size of the ensemble in terms of the number of experts?

Q8: How does it scale in comparison to other state of the art approaches for combination of forecasters in terms of computational effort?

Q9: What is the impact of the sequential re-weighting procedure in state of the art approaches for combining experts? Moreover, how does this approach compare with methods that handle correlation in the feature space (e.g. principal components analysis)?

### 4.1 Experimental setup

To address the research questions we used 62 real world time series from several domains. These are briefly described in Table 1. We limited the time series portfolio by size: we use time series with size above 750 for having enough data to fit both the experts and the arbiters; and size below 3000 in the interest of computational efficiency.

To account for trend we applied a KPSS statistical test (Kwiatkowski et al. 1992) to the data. Time series that are not trend-stationary according to this test are differenced until the test is passed. This approach is commonly used for trend inclusion in forecasting models, for example ARIMA. Specifically, we follow the procedure adopted by the automatic forecasting model auto.arima from the forecast R package (Hyndman 2014). The number of differences applied to each time series is described in the last column of Table 1.

We estimate the optimal embedding dimension ($K$) using the method of False Nearest Neighbours (Kennel et al. 1992). This method analyses the behaviour of the nearest neighbours as we increase $K$. According to Kennel et al. (1992), with a low sub-optimal $K$ many of the nearest neighbours will be false. Then, as we increase $K$ and approach an optimal embedding dimension those false neighbours disappear. We set the tolerance false nearest neighbours to 1%. The embedding dimension estimated for each series is shown in Table 1.

The feature set used by the forecasters M includes the previous K values (embedding vector), together with the following characteristics computed in each embedding vector:

- Local trend, estimated according to the ratio between the standard deviation of the embedding vector and the standard deviation of the differenced embedding vectors;
- Skewness, for measuring the symmetry of the distribution of the embedding vectors;
- Mean, as a measure of centrality of the embedding vectors;
- Standard deviation, as a dispersion metric;

**Table 1** Datasets and respective summary

| ID | Time series | Data source | Data characteristics | Size | K | I |
|---|---|---|---|---|---|---|
| 1 | Rotunda AEP | Porto water consumption from different locations in the city of Porto (Cerqueira et al. 2017a) | Half-hourly values from Nov. 11, 2015 to Jan. 11, 2016 | 3000 | 30 | 0 |
| 2 | Preciosa mar | | | 3000 | 9 | 1 |
| 3 | Amial | | | 3000 | 11 | 0 |
| 4 | Global horizontal radiation | Solar radiation monitoring (Cerqueira et al. 2017a) | Hourly values from Apr. 25, 2016 to Aug. 25, 2016 | 3000 | 23 | 1 |
| 5 | Direct normal radiation | | | 3000 | 19 | 1 |
| 6 | Diffuse horizontal radiation | | | 3000 | 18 | 1 |
| 7 | Average wind speed | | | 3000 | 10 | 1 |
| 8 | Humidity | Bike sharing (Cerqueira et al. 2017a) | Hourly values from Jan. 1, 2011 | 1338 | 11 | 0 |
| 9 | Windspeed | | Mar. 01, 2011 | 1338 | 12 | 0 |
| 10 | Total bike rentals | | | 1338 | 8 | 0 |
| 11 | AeroStock 1 | Stock price values from different aerospace companies (Cerqueira et al. 2017a) | Daily stock prices from January 1988 through October 1991 | 949 | 6 | 1 |
| 12 | AeroStock 2 | | | 949 | 13 | 1 |
| 13 | AeroStock 3 | | | 949 | 7 | 1 |
| 14 | AeroStock 4 | | | 949 | 8 | 1 |
| 15 | AeroStock 5 | | | 949 | 6 | 1 |
| 16 | AeroStock 6 | | | 949 | 10 | 1 |
| 17 | AeroStock 7 | | | 949 | 8 | 1 |
| 18 | AeroStock 8 | | | 949 | 8 | 1 |
| 19 | AeroStock 9 | | | 949 | 9 | 1 |
| 20 | AeroStock 10 | | | 949 | 8 | 1 |
| 21 | CO.GT | Air quality indicators in an Italian city (Lichman 2013) | Hourly values from Mar. 10, 2004 to Apr. 04 2005 | 3000 | 30 | 1 |
| 22 | PT08.S1.CO | | | 3000 | 8 | 1 |
| 23 | NMHC.GT | | | 3000 | 10 | 1 |
| 24 | C6H6.GT | | | 3000 | 13 | 0 |
| 25 | PT08.S2.NMHC | | | 3000 | 9 | 0 |

**Table 1** continued

| ID | Time series | Data source | Data characteristics | Size | K | I |
|---|---|---|---|---|---|---|
| 26 | NOx.GT | | | 3000 | 10 | 1 |
| 27 | PT08.S3.NOx | | | 3000 | 10 | 1 |
| 28 | NO2.GT | | | 3000 | 30 | 1 |
| 29 | PT08.S4.NO2 | | | 3000 | 8 | 0 |
| 30 | PT08.S5.O3 | | | 3000 | 8 | 0 |
| 31 | Temperature | | | 3000 | 8 | 1 |
| 32 | RH | | | 3000 | 23 | 1 |
| 33 | Humidity | | | 3000 | 10 | 1 |
| 34 | Electricity total load | Hospital energy loads (Cerqueira et al. 2017a) | Hourly values from Jan. 1, 2016 to Mar. 25, 2016 | 3000 | 19 | 0 |
| 35 | Equipment load | | | 3000 | 30 | 0 |
| 36 | Gas energy | | | 3000 | 10 | 1 |
| 37 | Gas heat energy | | | 3000 | 13 | 1 |
| 38 | Water heater Energy | | | 3000 | 30 | 0 |
| 39 | Total demand | Australian electricity (Koprinska et al. 2011) | Half-hourly values from Jan. 1, 1999 to Mar. 1, 1999 | 2833 | 6 | 0 |
| 40 | Recommended retail price | | | 2833 | 19 | 0 |
| 41 | Sea level pressure | Ozone level detection (Lichman 2013) | Daily values from Jan. 2, 1998 to Dec. 31, 2004 | 2534 | 9 | 0 |
| 42 | Geo-potential height | | | 2534 | 7 | 0 |
| 43 | K Index | | | 2534 | 7 | 0 |
| 44 | Flow of Vatnsdalsa river | Data market (Hyndman 2017) | Daily, from Jan. 1, 1972 to Dec. 31, 1974 | 1095 | 11 | 0 |
| 45 | Rainfall in Melbourne | | Daily, from from 1981 to 1990 | 3000 | 29 | 0 |
| 46 | Foreign exchange rates | | Daily, from Dec. 31, 1979 to Dec. 31, 1998 | 3000 | 6 | 1 |
| 47 | Max. temperatures in Melbourne | | Daily, from from 1981 to 1990 | 3000 | 7 | 0 |
| 48 | Min. temperatures in Melbourne | | Daily, from from 1981 to 1990 | 3000 | 6 | 0 |

**Table 1** continued

| ID | Time series | Data source | Data characteristics | Size | K | I |
|---|---|---|---|---|---|---|
| 49 | Precipitation in River Hirnant | | Half-hourly, from Nov. 1, 1972 to Dec. 31, 1972 | 2928 | 6 | 1 |
| 50 | IBM common stock closing prices | | Daily, from Jan. 2, 1962 to Dec. 31, 1965 | 1008 | 10 | 1 |
| 51 | Internet traffic data I | | Hourly, from Jun. 7, 2005 to Jul. 31, 2005 | 1231 | 10 | 0 |
| 52 | Internet traffic data II | | Hourly, from Nov. 19, 2004 to Jan. 27, 2005 | 1657 | 11 | 1 |
| 53 | Internet traffic data III | | from Nov. 19, 2004 to Jan. 27, 2005—data collected at 5 min intervals | 3000 | 6 | 1 |
| 54 | Flow of Jokulsa Eystri river | | Daily, from Jan. 1, 1972 to Dec. 31, 1974 | 1096 | 21 | 0 |
| 55 | Flow of O. Brocket | | Daily, from Jan. 1, 1988 to Dec. 31, 1991 | 1461 | 6 | 1 |
| 56 | Flow of Saugeen river I | | Daily, from Jan. 1, 1915 to Dec. 31, 1979 | 1400 | 6 | 0 |
| 57 | Flow of Saugeen river II | | Daily, from Jan. 1, 1988 to Dec. 31, 1991 | 3000 | 30 | 0 |
| 58 | Flow of Fisher River | | Daily, from Jan. 1, 1974 to Dec. 31, 1991 | 1461 | 6 | 0 |
| 59 | No. of Births in Quebec | | Daily, from Jan. 1, 1977 to Dec. 31, 1990 | 3000 | 6 | 1 |
| 60 | Precipitation in O. Brocket | | Daily, from Jan. 1, 1988 to Dec. 31, 1991 | 1461 | 29 | 0 |
| 61 | Min. temperature | Porto weather (Cerqueira et al. 2017a) | Daily values from Jan. 1, 2010 to Dec. 28, 2013 | 1456 | 8 | 0 |
| 62 | Max. temperature | | | 1456 | 10 | 0 |

- Serial correlation, estimated using a Box-Pierce test statistic;
- Long-range dependence, using a Hurst exponent estimation with wavelet transform;
- Chaos, using the maximum Lyapunov exponent to measure the level of chaos in the system.

These statistics are commonly used to summarise the overall structure of time series (Wang et al. 2009). The metalearning models use the same feature set used by the base forecasters.

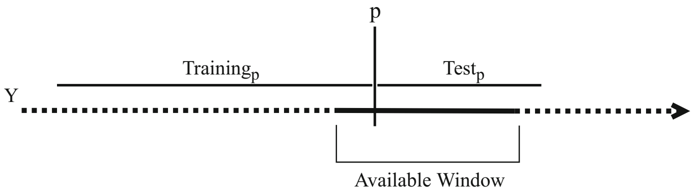

**Fig. 2** Example of one iteration of the repeated holdout procedure. A point $p$ is chosen from the available window. Then, the previous 50% of observations are used for training, while the subsequent 25% observations are used for testing

The final representation of the time series is exemplified in the following matrix:

$$Y_{[n,K]} = \left[\begin{array}{cccccccc|c} y_1 & y_2 & \dots & y_{K-1} & y_K & S_{trend_1} & \dots & S_{chaos_1} & y_{K+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{i-K+1} & y_{i-K+2} & \dots & y_{i-1} & y_i & S_{trend_i} & \dots & S_{chaos_i} & y_{i+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n-K+1} & y_{n-K+2} & \dots & y_{n-1} & y_n & S_{trend_n} & \dots & S_{chaos_n} & y_{n+1} \end{array}\right]$$

Taking the first row of the matrix as an example, the target value is $y_{K+1}$, while the attributes are the previous K values $\{y_1, \dots, y_K\}$ along with the above-mentioned statistics $\{S_{trend}, \dots, S_{chaos}\}$. In the meta-level, the target value is replaced by the absolute loss of a predictive model in that observation.

### 4.1.1 Evaluation procedure

The methods included in the experiments were evaluated using the root mean squared error (RMSE). A repeated holdout procedure in 15 testing periods was used as an estimation method. This consists in repeating a learning plus testing cycle 15 times using different but overlapping observations. This approach has been shown to provide robust performance estimates in time series forecasting tasks (Cerqueira et al. 2017b). In our setup, each repetition uses 50% of the time series size $t$ for training, while the subsequent 25% observations are used for testing. The window of used observations was chosen randomly following the idea of Monte Carlo approximation. This process is illustrated in Fig. 2. A point $p$ is randomly chosen from the the available window (constrained by the training and testing sizes). This point then marks the end of the training set, and the start of the testing set.

## 4.2 Ensemble setup and baselines

The set $M$ of experts forming the ensemble are summarised in Table 2. Different parameter settings are used for each of the individual learners, adding up to **50** base models. The parameters that are not specified were set with default values or are automatically tuned. This choice of number of experts will be analysed in Sect. 4.4.3.

As exploratory analysis, we show in Fig. 3 the distribution of the rank of each expert across the 62 problems. A rank of 1 means that the respective model was the best performing one in a given dataset. In the interest of readability, the legend describing the experts only shows the respective ID. Generally, the range of the distribution of rank is large, and even the experts with low median rank are among the best in some of the time series problems.

**Table 2** Summary of the experts

| ID | Algorithm | Parameter | Value |
|---|---|---|---|
| SVR | Support vector regr. (Karatzoglou et al. 2004) | Kernel | {Linear, RBF Polynomial, Laplace} |
| | | Cost | {1} |
| | | $\epsilon$ | {0.1} |
| MARS | Multivar. A. R. splines (Milborrow 2012) | Degree | {1, 3} |
| | | No. terms | {7, 15} |
| | | Forward thresh. | {0.001} |
| RF | Random forest (Wright 2015) | No. trees | {100, 250, 500} |
| PPR | Proj. pursuit regr. (R Core and Team 2013) | No. terms | {2, 5} |
| | | Method | {Super smoother, spline} |
| RBR | Rule-based regr. (Kuhn et al. 2014) | No. iterations | {10, 25, 50, 100} |
| GBR | Generalized boosted regr. (Ridgeway 2015) | Depth | {5, 10} |
| | | Distribution | {Gaussian, Laplace} |
| | | No. trees | {500, 1000} |
| | | Learning rate | {0.1} |
| MLP | Multi-layer perceptron (Venables and Ripley 2002) | Hidden units | {3, 5, 7, 10, 15, 25} |
| | | Decay | {0.01} |
| GLM | Generalised linear regr. (Friedman et al. 2010) | Penalty mixing | {0, 0.2, 0.4, 0.6, 0.8, 1} |
| GP | Gaussian processes (Karatzoglou et al. 2004) | Kernel | {Linear, RBF, Polynomial, Laplace} |
| | | Tolerance | {0.001, 0.01} |
| PCR | Principal comp. regr. (Mevik et al. 2016) | *Default* | – |
| PLS | Partial least regr. (Mevik et al. 2016) | Method | {Kernel, SIMPLS} |
| ARIMA | ARIMA (Hyndman 2014) | *Auto* | – |
| ETS | Exp. smoothing (Hyndman 2014; De Livera et al. 2011) | Method | {ETS, TBATS} |

The rule-based model RBR, a variant of Quinlan's model tree, presents a remarkable rank distribution.

We use a Random Forest as meta-learner. The blocked prequential procedure used to obtain out-of-bag samples was run with 10 folds ($b = \mathbf{10}$). The committee for each prediction (Sect. 3.3.1) contains 50% of the forecasters with best performance in the last 50 observations ($\Omega$ and $\lambda$ values are set to **50**). We suspend only half the models in the interest of keeping the combined model readily adaptable to changes in the environment. An average performing model may rapidly become important and the combined model should be able to capture these situations. By setting $\lambda$ to 50 we strive for estimates of recent performance that renders a robust committee. The sensitivity of ADE to different values of $\Omega$ and $\lambda$ is analysed in Sect. 4.4.2. We used Pearson's method as the correlation function for the sequential re-weighting of experts (Sect. 3.3.3).

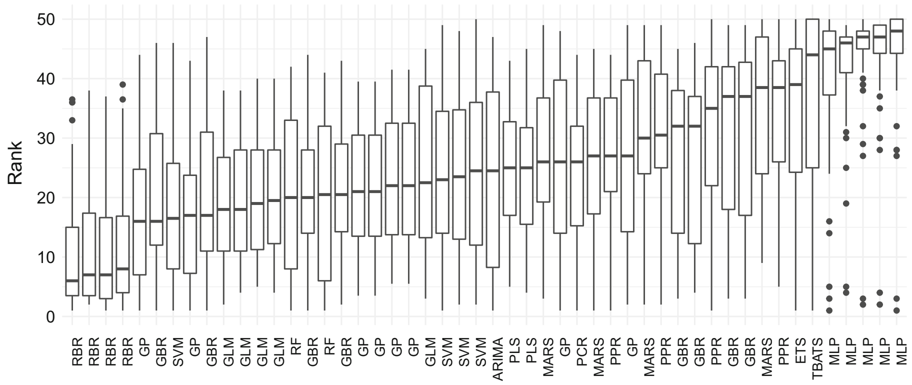

**Fig. 3** Distribution of rank of the base models across the 62 problems

We compare the performance of ADE with the following approaches:

Stacking: An adaptation of stacking (Wolpert 1992) for times series, where a meta-model is learned using the base-level predictions as attributes. To preserve the temporal order of observations, the out-of-bag predictions used to train the meta-learner (a random forest) are obtained using a blocked prequential procedure (c.f. Sect. 3.2.1). Different strategies for training the meta-learner (e.g. holdout) were tested and blocked prequential presented the best results;

Arbitrating: An approach following the original arbitrating method presented by Ortega et al. (2001), c.f. Sect. 2.3;

Simple: The approach in which the available experts are simply averaged using an arithmetic mean (Timmermann 2006);

SimpleTrim: Simple average with model selection: $\Omega$% of the best past performing models are selected and aggregated with a simple average;

LossTrain: Weighted static combination of experts, in which the weights are set according to the performance of experts in the training set;

BestTrain: An approach that selects the model with best performance in the training data to predict all the test set;

WindowLoss: Weighted adaptive combination of experts. The weights are computed according to the performance of the experts in the last $\lambda$ observations (Newbold and Granger 1974);

Blast: Similar to WindowLoss, but selects the best expert in the last $\lambda$ observations for prediction. van Rijn et al. (2018) showed its competitiveness using streaming data;

AEC: The adaptive combination procedure AEC (Sánchez 2008), c.f. Sect. 2.2;

ERP: The adaptive combination procedure proposed by Timmermann (2008), c.f. Sect. 2.2;

EWA: A forecast combination approach based on an exponentially weighted average—we refer to the seminal work by Cesa-Bianchi and Lugosi for a comprehensive description and theoretical properties (Cesa-Bianchi and Lugosi 2006, Section 2.1);

FixedShare: The fixed share approach due to Herbster and Warmuth (1998), which is designed for tracking the best expert across a time series (Cesa-Bianchi and Lugosi 2006, Section 5.2);

MLpol: The polinomially weighted average forecast combination (Cesa-Bianchi and Lugosi 2003). See Cesa-Bianchi and Lugosi for a comprehensive description and theoretical properties (Cesa-Bianchi and Lugosi 2006, Section 2.1);

OGD: An approach based on online gradient descent that provides theoretical loss bound guarantees (Zinkevich 2003);

ARIMA: A state-of-the-art method for time series forecasting. We use the implementation provided in the forecast R package (Hyndman 2014), which automatically tunes ARIMA to an optimal parameter setting.

Naive Baseline that uses the value of the previous observation ($y_t$) for predicting $y_{t+1}$;

SeasonalNaive: Baseline that uses the value of the observation from the previous seasonal period for predicting $y_{t+1}$ (Hyndman 2014). Particularly, for daily time series we use the value from the previous week, and for hourly time series we use the value from the day before;

ExpSmoothing: The exponential smoothing state space model typically used for forecasting (Hyndman 2014).

For the approaches EWA, MLpol, FixedShare, and OGD, we used the software package *opera* (Gaillard and Goude 2016).

The following variants of ADE were tested:

ADE-selectbest: A variant of ADE in which at each time point the best model is selected to make a prediction. Here best is the one with lowest predicted loss. This is in accordance with the original arbitrating architecture (Ortega et al. 2001);

ADE-allmodels: A variant of ADE, but without the formation of a committee. In this case, all forecasting models are weighed according to their expertise in the input data;

ADE-noreweight: A variant of ADE in which there is no reweight of the experts according to the correlation of their predictions (Sect. 3.3.3);

ADE-v0: The preliminary version of ADE (Cerqueira et al. 2017a). Besides the re-weighting of experts, this approach uses a linear transformation of the output of the arbiters, instead of the softmax function previously proposed (Cerqueira et al. 2017a);

ADE-vanilla: A baseline variant of ADE with a simpler weighting approach: the error ($\hat{y} - y$) predicted by arbiters is simply added to the output of the respective expert. The final prediction is computed according to the average of the shifted output of experts.

### 4.3 Results

We evaluate the results of the experiments from multiple perspectives. This includes a formal evaluation according to the Bayesian analysis described by Benavoli et al. (2017). Particularly, we employed the Bayesian correlated *t*-test to compare pairs of models in a single problem, and the Bayes sign test to compare pairs of methods across multiple problems. We define the *region of practical equivalence* (Benavoli et al. 2017) (ROPE) to be the interval [− 0.01,

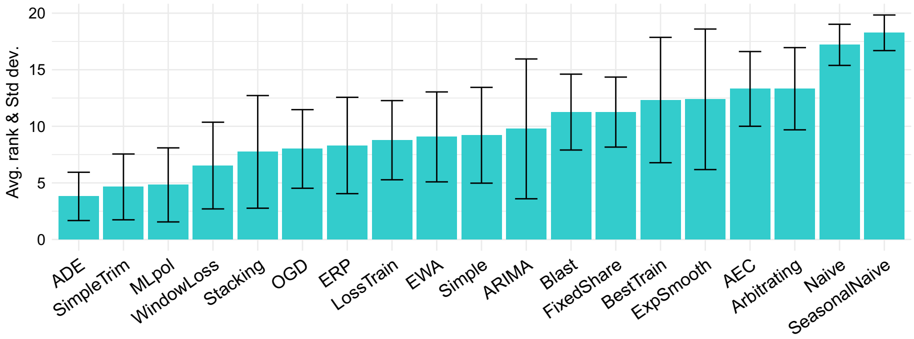

**Fig. 4** Average rank and respective standard deviation of ADE and state of the art methods

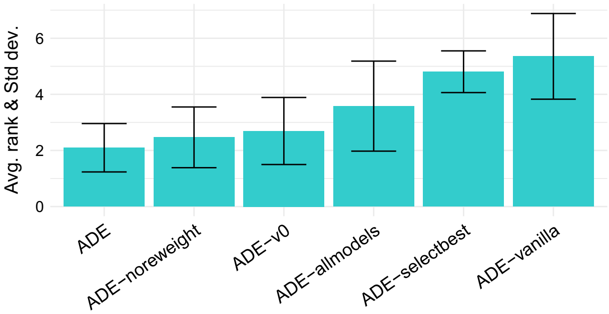

**Fig. 5** Average rank and respective standard deviation of ADE and its variants

0.01]. Essentially, this means that two methods show indistinguishable performance if the difference in performance between them falls within this interval. For a thorough read on Bayesian analysis for comparing predictive models we refer to the work by Benavoli et al. (2017). In this Bayesian analysis of the results we make a small change to the performance metric. Since RMSE varies according to scale, we normalise this value relative to the RMSE of the Simple aggregation approach, which is a standard forecast combination baseline. To be more precise, for each aggregation method Agg we compute the following value:

$$\text{nRMSE}(\texttt{Agg}) = \text{RMSE}(\texttt{Agg})/\text{RMSE}(\texttt{Simple})$$

Figures 4 and 5 represent the average rank, and respective standard deviation, of ADE and its variants, state of the art approaches for forecast combination, and other typical forecasting baselines. Figure 6 shows the log percentual difference in RMSE of ADE relative to other forecasting approaches. For this specific analysis the initial outliers in the results were removed for a better visualisation of the difference in performance. Figure 7 show the results of the Bayes sign test. This illustrates the proportion of probability that ADE wins, draws (result within the ROPE), or loses with each respective method. Table 3 presents the paired comparisons between the proposed method and all other approaches using the Bayesian correlated $t$-test. The numbers represent wins, draws, and losses of the proposed method. The numbers in parenthesis represent wins/draws/losses with probability above 95%.

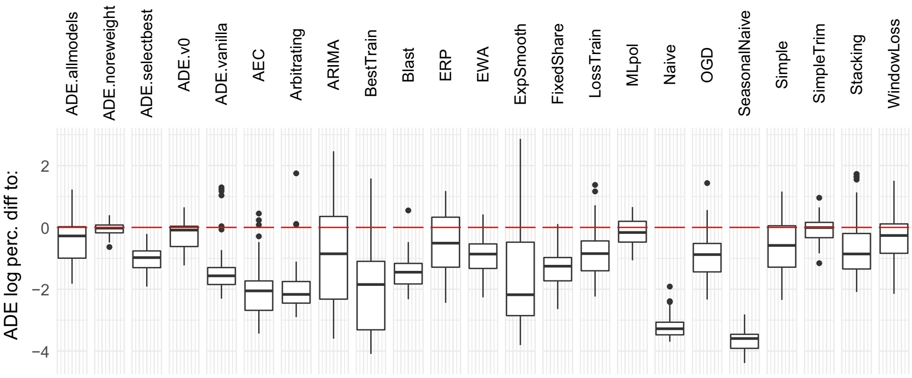

**Fig. 6** Distribution of the log percentual difference in performance of ADE relative to other forecasting methods. Negative values denotes better performance by ADE

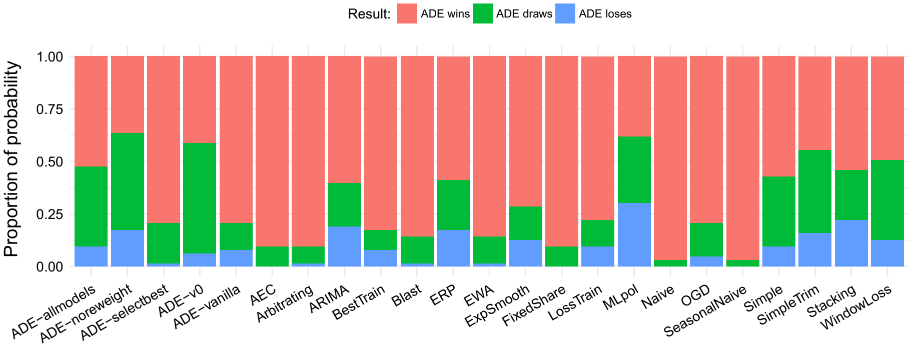

**Fig. 7** Proportion of probability of ADE winning/drawing/losing according to the Bayes sign test

### 4.3.1 Comparing ADE to the state of the art approaches

ADE presents the best average rank relative to state of the art aggregation methods. This value is considerably better compared to widely used approaches, including Stacking, Simple, or WindowLoss. From the numbers of Table 3, ADE wins in most of the problems against other approaches, most of the times in a considerable way (i.e., with probability above 95%). Among the combination approaches, BestTrain presents one of the lowest average ranks, which suggests that the combination of different experts is worthwhile in terms of predictive performance. The simple average aggregation coupled with model selection leads to an interesting average rank, which is only topped by that of ADE. These results are corroborated by the outcome of the Bayes sign test, which suggests that ADE has an higher probability of winning compared to each other approach.

Figure 6 is useful for visualising the magnitude in the difference in predictive performance, something which average ranks are blind to. The distribution of the percentual difference varies according to the model under comparison. In general, ADE shows a reasonable difference when compared with most of the other approaches.

These results answer the research question **Q1** regarding the performance of ADE relative to the state of the art approaches for combining forecasting experts.

**Table 3** Paired comparisons between ADE and the baselines in the 62 time series

| Method | ADE loses | ADE draws | ADE wins |
|---|---|---|---|
| Stacking | 12 (3) | 16 (2) | 34 (13) |
| Arbitrating | 1 (0) | 2 (0) | 59 (41) |
| Simple | 3 (1) | 24 (17) | 35 (24) |
| SimpleTrim | 4 (1) | 45 (32) | 13 (10) |
| LossTrain | 3 (1) | 21 (8) | 38 (25) |
| WindowLoss | 3 (2) | 35 (26) | 24 (19) |
| Blast | 1 (0) | 2 (0) | 59 (42) |
| AEC | 0 (0) | 6 (4) | 56 (47) |
| ERP | 8 (1) | 21 (8) | 33 (23) |
| BestTrain | 3 (0) | 6 (0) | 53 (42) |
| EWA | 0 (0) | 23 (8) | 39 (9) |
| FixedShare | 0 (0) | 6 (2) | 56 (27) |
| MLpol | 5 (2) | 43 (26) | 14 (2) |
| OGD | 1 (0) | 23 (8) | 38 (19) |
| ARIMA | 8 (5) | 17 (7) | 37 (33) |
| Naive | 0 (0) | 0 (0) | 62 (61) |
| SeasonalNaive | 0 (0) | 0 (0) | 62 (62) |
| ExpSmoothing | 6 (5) | 10 (4) | 46 (45) |
| ADE-selectbest | 1 (1) | 11 (2) | 50 (24) |
| ADE-allmodels | 3 (1) | 34 (22) | 25 (16) |
| ADE-noreweight | 1 (0) | 53 (47) | 8 (6) |
| ADE-v0 | 1 (1) | 46 (31) | 15 (9) |
| ADE-vanilla | 5 (1) | 2 (0) | 55 (34) |

Number in parenthesis represent a probability of win/draw/loss above 95% according to the Bayesian correlated *t*-test

Relative to the original arbitrating architecture, denoted as Arbitrating, the proposed method shows a considerable improvement, which results in a much better average rank. This proves that the introduced components are fundamental for the achieved performance, which answers question **Q2**.

### 4.3.2 Comparing ADE to its variants

ADE shows a consistent advantage over the performance of ADE-allmodels (**Q4**). This suggests that indeed it is worthwhile to prune the ensemble for each prediction (as opposed to combining all the forecasters). ADE's performance is also considerably better relative to ADE-selectbest, which gives evidence for the hypothesis that the combination of experts (as opposed to selection) provides better results (**Q3**). ADE is also superior to ADE-vanilla, which bypasses the weighting scheme, directly adjusting the output of the experts according to the predictions of the arbiters.

ADE shows a consistent improvement over the variant that does not perform a sequential re-weighting of the experts according to recent correlation (Sect. 3.3.3) (**Q5**). The magnitude of the difference in performance is small (Fig. 6), which is corroborated by the high number of draws shown in Table 3. However, it is important to note that the sequential re-weighting method does not generally compromise performance (only one loss in 62 problems), and improves it several times. Finally, ADE also shows a systematic improvement over its preliminary version (Cerqueira et al. 2017a). Besides not using the sequential re-weighting approach, ADE_v0 aggregates the output of the experts using a softmax function. We tested this approach in the experimental setup of this work and found that it does not improve the results over a linear transformation.

### 4.4 Further analyses of ADE

Following the comparison of ADE with the state of the art, in this section we provide a more detailed analysis of its workflow. The goal is to enhance our understanding of how the method works. This analysis encompasses: (1) an analysis on the different possible deployment strategies; (2) a sensitivity analysis on the parameters $\Omega$ and $\lambda$; (3) a scalability analysis in terms of relative computational time; (4) a study on the impact of adding additional experts; and (5) additional analysis of the sequential re-weighting method. If not stated otherwise the experimental setup is the same as described previously.

#### 4.4.1 Analyzing training strategies

In this section we address the research questions **Q4** and **Q6**. In a dynamic environment it is common to update the model over time, either online or in chunks of observations. Time-dependent data is prone to changes in the underlying distribution and continuous training of models ensures that one has an up-to-date model. Since ADE settles on two layers of models we analysed different approaches for updating these and study their implications in terms of predictive performance.

In the main experiments, ADE is trained using only the training data. To understand if and how should ADE be updated over time we tested the following strategies:

M0_Z0: both experts (M) and arbiters (Z) are trained in the training set and not updated during test time (ADE as reported in the main experiments);

M0_Z1: M is trained only in the training data but $Z$ is re-trained every $\Delta$ observations.

M1_Z0: M is re-trained every $\Delta$ observations but $Z$ is trained only in the training data.

M1_Z1: Both $M$ and $Z$ are re-trained every $\Delta$ observations, which is particularly interesting if the models in M are typical online methods (e.g. ARIMA);

ADE-runtime: A variant of ADE in which there is no blocked prequential procedure to obtain out-of-bag samples to increase the data provided to the meta-learners. In this scenario, the arbiters are trained in data obtained only at run-time every $\Delta$ observations, which is also in accordance with the original arbitrating strategy and other metalearning approaches used in time-dependent scenarios (Gama and Kosina 2014). M is fit only in the training data.

We set $\Delta$ to 100. In the interest of robustness, this analysis was carried out using the time series of size 3000 (33 datasets—see Table 1). Since the predictive models are updated frequently, in this particular analysis we settled for a simple holdout estimation procedure, where the training consists in the initial 70% of the data. The test set is comprised by the remaining 30% observations.

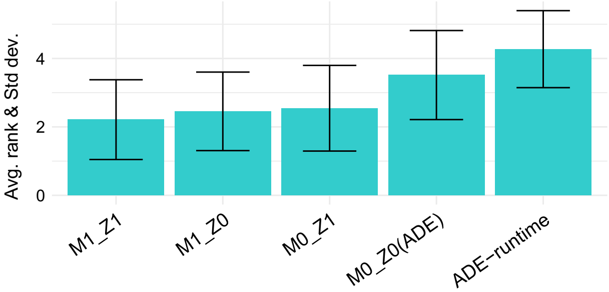

**Fig. 8** Average rank and respective standard deviation of ADE's deployment strategies

The results are presented in Fig. 8, with a barplot representing the average rank and respective standard deviation of each deployment strategy.

ADE (also denoted as M0_Z0 in this particular analysis) shows a better average rank relative to ADE-runtime, which suggests that it is better to get out-of-bag predictions from the available data to improve the fit of the meta-learners.

The results also suggest that updating the experts and the arbiters at run-time is better than not updating them. This outcome is expected due to the eventual presence of concept drift (Gama et al. 2014). Particularly, the M1_Z1 approach presents the best average rank. Although the difference in average rank is negligible, the results also suggest that updating the experts and not updating the arbiters (M1_Z0) renders a better average rank than the inverted strategy (M0_Z1).

### 4.4.2 Sensitivity analysis on $\Omega$ and $\lambda$

In this and the next subsection we answer the research question **Q7** regarding the sensitivity analysis of ADE. Besides the setup of experts and arbiters, ADE has two main parameters: $\Omega$, which represents the ratio of experts selected at each time step for forecasting; and $\lambda$, which denotes the window size used to compute the performance of the experts (for selecting which ones to arbitrate).

To some extent, these parameters are dependent not only on the ensemble composition, but also on the data itself. In this section we briefly analyse how the performance of ADE varies as the values of the parameters $\Omega$ and $\lambda$ change. We considered ADE with $\lambda = \{3, 5, 10, 15, 25, 50, 60, 75, 100, 150, 300, 450, 600\}$ and $\Omega = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ (values chosen arbitrarily). This renders a total of 117 variants of ADE. This analysis was carried out using the 33 time series of size 3000.

The results are shown in Fig. 9. The graphic illustrates two heatmaps. These relate the average rank (left heatmap) and respective standard deviation (right heatmap) of each ($\Omega$, $\lambda$) combination across the 33 datasets. Higher average rank (i.e. worse performance) and higher rank standard deviation are denoted by darker tiles.

Regarding $\Omega$, the best performing values are the ones in the middle of the searched distribution. In principle, this parameter depends to a great extent on the number of experts and their predictive ability. The results also suggest that, unless for extremely low $\lambda$ values, fixing $\Omega$ and varying $\lambda$ renders a relatively stable average rank.

The heatmap in the right side suggests that the ($\Omega$, $\lambda$) combinations with lowest rank standard deviation are in the middle of the searched distributions.

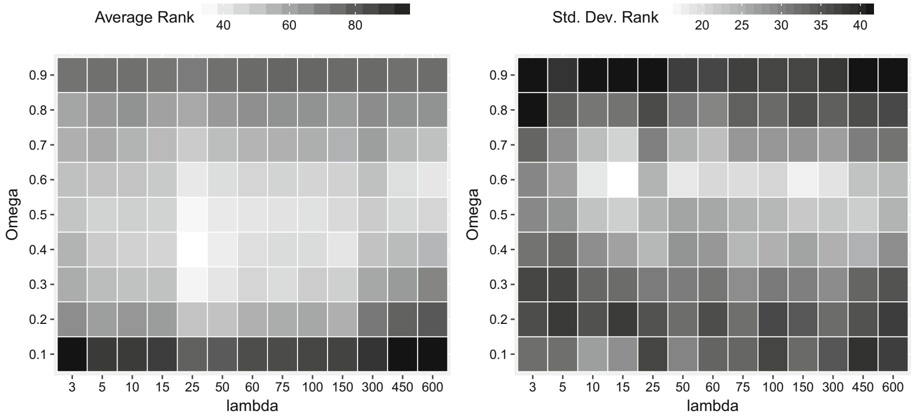

**Fig. 9** Heatmaps illustrating the average rank (left) and respective standard deviation (right) of ADE for varying $\Omega$ and $\lambda$ parameters. Darker tiles mean higher values

In principle and in practice, varying the value of $\lambda$ follows the stability-plasticity dilemma (Carpenter et al. 1991): small values of $\lambda$ (i.e. small window of recent observations) lead to greater reactiveness, but also makes the model susceptible to outliers. Conversely, higher values lead to greater stability, while losing some responsiveness and possibly containing outdated information.

### 4.4.3 Value of additional experts

In the experiments presented in the previous sections ADE was employed with 50 experts (Table 2). In this section we analyse the sensitivity of ADE to different ensemble compositions. Particularly, we tested ensembles with sizes from 5 to 100 by multiples of 5: $Q = \{5, 10, 15, \ldots, 95, 100\}$ , rendering a total of 20 different possible ensemble sizes for analysis.

We estimate the predictive performance of each composition using a Monte Carlo approximation. Specifically, for each Monte Carlo repetition and for each considered size $q \in Q$, we sampled without replacement $q$ experts from a pool of 100, and compute the performance of ADE with this configuration. Afterwards, we measure the relative performance of each size (averaged across 30 Monte Carlo simulations) with respect to the performance obtained when using the complete pool of 100 experts. We tested in 30 Monte Carlo repetition to obtain robust estimates of performance. The pool of 100 experts was created by adding different values to the parameters described in Table 2.

The result of this analysis is presented in Fig. 10. Generally, including more experts in the ensemble leads to a better performance, and closer to that of the ensemble with 100 models. However, the difference becomes negligible for values above 50. The uncertainty in performance is represented by the vertical bars and is computed according to the standard deviation across the Monte Carlo repetitions. This value also becomes increasingly small as more base models are included.

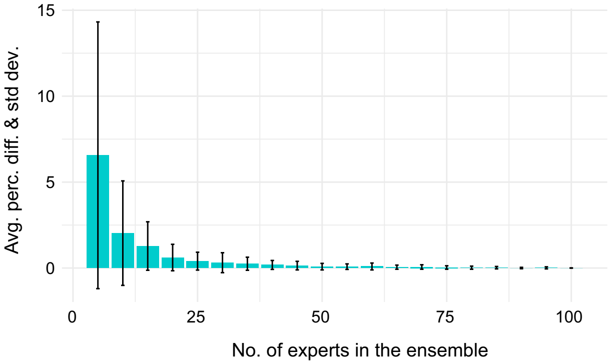

**Fig. 10** Average percentual difference in RMSE, and respective standard deviation, of ADE with different ensemble size compositions up to 100 models relative to ADE with 100 models

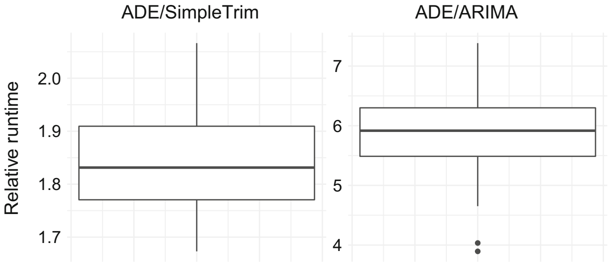

**Fig. 11** Log computational time spent by ADE relative to ARIMA and SimpleTrim approaches across the 62 problems

### 4.4.4 Scalability analysis

In the previous sections we have analysed ADE in terms of predictive performance. In this section we analyse ADE in terms of computation time. To accomplish this we measure the time spent in fitting ADE and using it to predict the test set. We use the time spent by ARIMA and SimpleTrim as references. The first is a state of the art approach to forecasting, while the second is the aggregation approach with closest average rank to ADE. We computed the time spent by ADE relative to the other two approaches across the 62 time series.

The results are presented in Fig. 11 as boxplots. On all problems, ADE takes more time to run than SimpleTrim. The difference of this method to ADE is mostly driven by the fitting and predictions of the arbiters. As expected, ADE also takes more time than ARIMA. Being a single model (as opposed to an ensemble), ARIMA has considerable less storage requirements when compared to ADE.

In summary, ADE scales worse than both approaches. Although omitted, it also takes more time than the remaining state of the art approaches used earlier (**Q8**).

### 4.4.5 Further analyses of the sequential re-weighting procedure

In Sect. 3.3.3 we presented an approach for handling the inter-dependencies among experts during their aggregation. The core arbitrage approach does not explicitly model the inter-dependencies among experts and this approach was designed to overcome this limitation.

**Table 4** Paired comparisons showing the impact of the sequential re-weighting approach in state of the art methods

| Method | Without re-weight wins | Draw | With re-weight wins |
|---|---|---|---|
| WindowLoss | 4 (2) | 31 (24) | 27 (24) |
| AEC | 32 (22) | 25 (18) | 5 (1) |
| EWA | 33 (14) | 25 (21) | 4 (0) |
| FixedShare | 41 (24) | 20 (18) | 1 (0) |
| MLpol | 27 (9) | 27 (20) | 8 (2) |
| OGD | 21 (4) | 26 (16) | 14 (5) |

Particularly, in the previous section we provided evidence of the benefits of the sequential re-weighting approach when applying it to ADE. Particularly, the results suggest that the magnitude of the impact is not substantial. Notwithstanding, applying this method does not generally decrease performance and improves it several times.

In this section we analyse the sequential re-weighting method from two more different perspectives, according to research question **Q9**. First, we study the impact of applying this procedure to other state of the art approaches for dynamic combination of forecasting experts. In the interest of fairness we focused this analysis only on approaches which perform dynamic expert combination using estimated weights. Second, and focusing on ADE, we compare sequential re-weighting of experts with approaches that handle correlation in the feature space. Specifically and before training the experts, two different approaches are tested: (1) attributes with correlation above 95% with other features are removed (ADE-corr-noreweight); (2) principal components analysis is applied to the data, keeping 95% of the variance (ADE-pca-noreweight). The value 95% was chosen arbitrarily. In this analysis we also study ADE-corr and ADE-pca, where ADE is applied with sequential re-weighting and the methods (1) and (2) described above.

The results of the first analysis are reported in Table 4, where each approach in the first column is compared with itself when using the sequential re-weighting approach. Similarly to Table 3, this table shows paired comparisons of the respective method with and without the application of the sequential re-weighting method. In parenthesis are denoted the results that happen with at least 95% probability according to the Bayesian correlated *t*-test.

Besides ADE, the results suggest that the approach is also beneficial to WindowLoss. However, when applied to the other tested approaches its impact vanishes and is often decreases the predictive performance.

Figure 12 shows the results of the second analysis. ADE shows the best average rank across the tested approaches. The average ranks suggests that, applying the sequential re-weighting procedure improves the predictive performance in the three variants of ADE. Even when accounting for correlation in feature space, the sequential re-weighting approach still improves the average rank during expert aggregation.

## 5 Discussion and future work

In this section we discuss the presented results. We start by addressing the limitations of ADE, where we also outline possible solutions to those shortcomings. Then we overview future work regarding ADE.

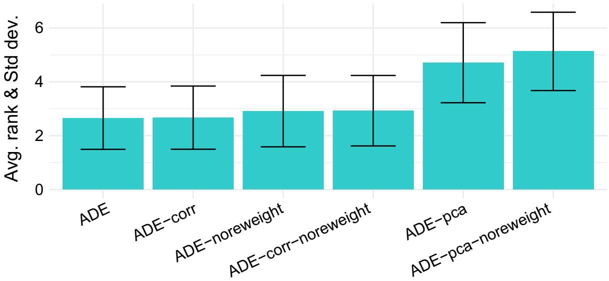

**Fig. 12** Average rank and respective standard deviation of ADE and its variants

### 5.1 On concept drift

Some of the design decisions behind ADE are based on prior work regarding the variance in relative performance of forecasting models over a time series (Aiolfi and Timmermann 2006) and with potential recurring structures present in the time series. However, there are cases in which time series change into new concepts and both the experts and arbiters may get outdated. Although we do not explicitly cover these scenarios, a possible strategy to address this issue is to track the loss of the ensemble. If its performance decreases beyond some tolerance new base-learners could be introduced (e.g. Gama and Kosina 2014) or existing ones re-trained. Since an arbitration approach provides a modular architecture, models can be added (or removed) as needed. Gama et al. (2014) survey several approaches for concept drift adaptation that also could be adopted.

### 5.2 On the sequential re-weighting procedure

In its preliminary version (Cerqueira et al. 2017a) we argued that one of ADE's limitation was that it did not directly modelled the inter-dependencies among experts. We address this issue in this work using a sequential re-weighting procedure that controls the redundancy among the output of the experts by considering their recent correlation. This approach is independent from ADE. However, its application with ADE is particularly interesting because the re-weighting occurs during aggregation and does not withhold ADE's modularity.

Despite the evidence of its benefits, the sequential re-weighting approach has space for improvement. Consider the following (rather extreme) example: one expert producing forecasts with a determined magnitude systematically below the true value, and another expert with similar behaviour but with forecasts above the true value. These two experts are highly correlated but in fact complement themselves greatly. Effectively, using the Pearson's correlation as a measure of similarity can be a sub-optimal solution in this case. Future work includes the exploration of better similarity functions. A possibly interesting line of enquiry is to follow Brown's work on the study of diversity in classifiers from an information theoretic perspective (Brown 2009). Particularly, instead of measuring the redundancy among experts only according to their outputs, we can also take into consideration the target value, i.e. conditional redundancy.

Finally, the application of the sequential re-weighting approach to other dynamic aggregation methods does not render the same positive effects as seen when it is applied to ADE. We plan to study this issue further in future work.

### 5.3 On scalability

In the previous section we identified the computational effort required by ADE relative to other approaches as its main limitation. In the future we plan to address this issue, by eventually adapting the method to a streaming scenario. One possibility is to use a single arbiter (instead of one arbiter for each expert) designed for multi-target regression, i.e. having a single regression model that forecasts the errors of all base models, though for ensembles with a large number of base models this can be cumbersome given the number of target variables we would have.

### 5.4 Scope of the experimental setup

From a broad perspective, forecasting can be split into different *varieties*. In this work we focus on uni-variate time series, assuming that only the variable of interest is available.

We also center our goal on predicting the next value of the time series, and assume immediate feedback from the environment. However, in many application domains one is often interested in predicting multiple steps in the future. Although we do not evaluate the proposed method in this setting, it can be extended to multi-step forecasting using state of the art approaches to this effect. We intend to study the application of ADE in these settings in future work.

Finally, as we describe in Sect. 4.1, we focus on time series with an high sampling frequency, specifically, half-hourly, hourly, and daily data. The main reason for this is because high sampling frequency is typically associated with more data, which is important for fitting the predictive models. Standard forecasting benchmark data are typically more centered around low sampling frequency time series, for example the M competition data (Makridakis et al. 1982).

### 5.5 Other research lines

We plan to address the previous limitations of ADE by exploring the described potential solutions. Besides these, there are other interesting open research questions. Specifically, we will study ways of quantifying and leveraging the uncertainty of the arbiters regarding the loss that the experts will incur. For example, one could develop an approach in which, when the uncertainty of the output of the arbiters is high, the weights are smoothed. This could be accomplished efficiently using, for example, an infinitesimal jackknife (Wager et al. 2014) (provided random forests are used as arbiters).

We also plan to study the ability of the method, and how it can be adapted, to the timely detection of anomalies, i.e., activity monitoring (Fawcett and Provost 1999). Another interesting analysis could be using ADE in a continual learning setup, where instances for a sequence of tasks are observed over time.

## 6 Conclusions

In this paper we presented ADE, a dynamic ensemble method. We focused on time series forecasting problems, where the objective is to predict future values of a sequence of observations.

ADE is comprised of a set of forecasting experts pre-trained on the available data. A metalearning approach is used to dynamically estimate the weight factors of these experts at run-time. This is accomplished by having a set of arbiters that model the error of each expert and predict how well they will perform in future observations. The resulting weights are used for obtaining the aggregated prediction of the ensemble. This aggregation may temporarily assign zero weight to some experts if their current performance is estimated to be too bad. This suspension decision may be revised in future time steps thus contributing to the robustness of the approach to regime changes.

We argued that this metalearning approach is useful to better capture recurring changes in the environment. Particularly, long-range temporal dependencies (e.g. seasonal factors) that short-memory windowing approaches may fail to grasp efficiently.

Our proposal also includes a sequential re-weighting approach for modelling the interdependencies among experts. Specifically, this approach is designed to control and reduce the redundancy in the output of the experts during their aggregation. Within the proposed arbitrage approach we also include a procedure for retrieving out-of-bag observations from the training set. These are used to fit the arbiters, significantly improving the data efficiency of the method.

We carried out an extensive empirical study to better characterise the performance of our proposal. This study has provided clear evidence on the competitiveness of our method in terms of predictive performance when compared to the state of the art. We also discussed its limitations and provided guidelines for solving them in future work. The main point for improvement is the scalability of the method. We plan to address this issue and potentially adapt ADE to streaming or incremental scenarios.

In the interest of reproducible science all methods are publicly available as an *R* software package.

**Acknowledgements** The authors acknowledge the insightful comments by anonymous reviewers, G. Brown, L. Todorovsky, N. Moniz, and M. Almeida. This work is financed by the ERDF-European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation—COMPETE 2020 Programme within project “POCI-01-0145-FEDER-006961”, and by National Funds through the FCT-Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of Project UID/EEA/50014/2013; Project “NORTE-01-0145-FEDER-000036” is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). This work was partly funded by the ECSEL Joint Undertaking, the framework programme for research and innovation horizon 2020 (20142020) under Grant Agreement No. 662189-MANTIS-2014-1.

## References

Aiolfi, M., & Timmermann, A. (2006). Persistence in forecasting performance and conditional combination strategies. *Journal of Econometrics*, *135*(1), 31–53.

Benavoli, A., Corani, G., Demšar, J., & Zaffalon, M. (2017). Time for a change: A tutorial for comparing multiple classifiers through bayesian analysis. *The Journal of Machine Learning Research*, *18*(1), 2653–2688.

Brazdil, P., Carrier, C. G., Soares, C., & Vilalta, R. (2008). *Metalearning: Applications to data mining*. Berlin: Springer.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Brown, G. (2009). An information theoretic perspective on multiple classifier systems. *International Workshop on Multiple Classifier Systems* (pp. 344–353). Berlin: Springer.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: A survey and categorisation. *Information Fusion*, *6*(1), 5–20.

Brown, G., Wyatt, J. L., & Tiňo, P. (2005). Managing diversity in regression ensembles. *Journal of Machine Learning Research*, *6*(Sep), 1621–1650.

Carbonell, J., & Goldstein, J. (1998). *The use of mmr, diversity-based reranking for reordering documents and producing summaries* (pp. 335–336). ACM.

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1991). Artmap: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, *4*(5), 565–588. https://doi.org/10.1016/0893-6080(91)90012-T.

Cerqueira, V., Torgo, L., Pinto, F., & Soares, C. (2017). Arbitrated ensemble for time series forecasting. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 478–494). Springer.

Cerqueira, V., Torgo, L., Smailović, J., Mozetič, I. (2017). A comparative study of performance estimation methods for time series forecasting. In proceedings of the 4th international conference on on data science and advanced analytics (pp. 529–538). IEEE. https://doi.org/10.1109/DSAA.2017.7.

Cerqueira, V., Torgo, L., & Soares, C. (2017). Arbitrated ensemble for solar radiation forecasting. *International work-conference on artificial neural networks* (pp. 720–732). Cham: Springer.

Cesa-Bianchi, N., & Lugosi, G. (2003). Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, *51*(3), 239–261.

Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning, and games*. New York: Cambridge University Press.

Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting*, *5*(4), 559–583.

Clemen, R. T., & Winkler, R. L. (1986). Combining economic forecasts. *Journal of Business and Economic Statistics*, *4*(1), 39–46.

Dawid, A. P. (1984). Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society. Series A (General), 147*(2), 278–292.

De Livera, A. M., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, *106*(496), 1513–1527.

Dietterich, T. G., & Bakiri, G. (1991). Error-correcting output codes: A general method for improving multiclass inductive learning programs. In *AAAI* (pp. 572–577).

Fawcett, T., & Provost, F. (1999). Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 53–62). ACM.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1–22.

Gaillard, P., & Goude, Y. (2015). Forecasting electricity consumption by aggregating experts; how to design a good set of experts. In *Modeling and stochastic learning for forecasting in high dimensions* (pp. 95–115). Springer.

Gaillard, P., & Goude, Y. (2016) opera: Online prediction by expert aggregation. R package version 1.0. https://CRAN.R-project.org/package=opera.

Gama, J., & Kosina, P. (2014). Recurrent concepts in data streams classification. *Knowledge and Information Systems*, *40*(3), 489–507.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, *46*(4), 44.

Genre, V., Kenny, G., Meyler, A., & Timmermann, A. (2013). Combining expert forecasts: Can anything beat the simple average? *International Journal of Forecasting*, *29*(1), 108–121.

Herbster, M., & Warmuth, M. K. (1998). Tracking the best expert. *Machine Learning*, *32*(2), 151–178.

Hyndman, R. (2017). Time series data library. http://data.is/TSDLdemo. Accessed 11 December 2017.

Hyndman, R. J. (2014). With contributions from George Athanasopoulos, Razbash, S., Schmidt, D., Zhou, Z., Khan, Y., Bergmeir, C., Wang, E.: forecast: Forecasting functions for time series and linear models. R package version 5.6.

Jacobs, R. (1995). Methods for combining experts' probability assessments. *Neural Computation*, *7*(5), 867–888.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, *3*(1), 79–87.

Jose, V. R. R., & Winkler, R. L. (2008). Simple robust averages of forecasts: Some empirical results. *International Journal of Forecasting*, *24*(1), 163–169.

Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). kernlab—An S4 package for kernel methods in R. *Journal of Statistical Software*, *11*(9), 1–20.

Kennel, M. B., Brown, R., & Abarbanel, H. D. (1992). Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, *45*(6), 3403.

Koprinska, I., Rana, M., & Agelidis, V. G. (2011). Yearly and seasonal models for electricity load forecasting. *The 2011 international joint conference on neural networks (IJCNN)* (pp. 1474–1481). IEEE.

Kuhn, M., Weston, S., & Keefer, C. (2014). Code for Cubist by Ross Quinlan, N.C.C.: Cubist: Rule- and Instance-Based Regression Modeling. R package version 0.0.18.

Kuncheva, L. I. (2004). *Multiple classifier systems: 5th International workshop, MCS 2004, Cagliari, Italy, June 9–11, 2004. Proceedings, chap. Classifier ensembles for changing environments* (pp. 1–15). Berlin: Springer. https://doi.org/10.1007/978-3-540-25966-4_1.

Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, *54*(1–3), 159–178.

Lichman, M. (2013). UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed 30 Aug 2017.

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., et al. (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, *1*(2), 111–153.

Mevik, B. H., Wehrens, R., & Liland, K. H. (2016). pls: Partial least squares and principal component regression. R package version 2.6-0. https://CRAN.R-project.org/package=pls.

Milborrow, S. (2012). Earth: Multivariate adaptive regression spline models. Derived from mda:mars by Trevor Hastie and Rob Tibshirani.

Newbold, P., & Granger, C. W. (1974). Experience with forecasting univariate time series and the combination of forecasts. *Journal of the Royal Statistical Society. Series A (General), 137*(2), 131–165.

Ortega, J., Koppel, M., & Argamon, S. (2001). Arbitrating among competing classifiers using learned referees. *Knowledge and Information Systems*, *3*(4), 470–490.

Pinto, F., Soares, C., & Mendes-Moreira, J. (2016). Chade: Metalearning with classifier chains for dynamic combination of classifiers. In *Joint european conference on machine learning and knowledge discovery in databases*. Springer.

R Core Team. (2013). *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing.

Ridgeway, G. (2015) gbm: Generalized Boosted Regression Models. R package version 2.1.1.

Rossi, A. L. D., de Leon Ferreira, A. C. P., Soares, C., De Souza, B. F., et al. (2014). Metastream: A meta-learning based method for periodic algorithm selection in time-changing data. *Neurocomputing*, *127*, 52–64.

Sánchez, I. (2008). Adaptive combination of forecasts with application to wind energy. *International Journal of Forecasting*, *24*(4), 679–693.

Takens, F. (1981). *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80, chap. Detecting strange attractors in turbulence* (pp. 366–381). Berlin: Springer. https://doi.org/10.1007/BFb0091924.

Timmermann, A. (2006). Forecast combinations. *Handbook of Economic Forecasting*, *1*, 135–196.

Timmermann, A. (2008). Elusive return predictability. *International Journal of Forecasting*, *24*(1), 1–18.

Todorovski, L., & Džeroski, S. (2003). Combining classifiers with meta decision trees. *Machine Learning*, *50*(3), 223–249.

van Rijn, J. N., Holmes, G., Pfahringer, B., & Vanschoren, J. (2018). The online performance estimation framework: Heterogeneous ensemble learning for data streams. *Machine Learning*, *107*(1), 149–176.

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S* (4th ed.). New York: Springer. ISBN 0-387-95457-0.

Wager, S., Hastie, T., & Efron, B. (2014). Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research*, *15*(1), 1625–1651.

Wang, X., Smith-Miles, K., & Hyndman, R. (2009). Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series. *Neurocomputing*, *72*(10), 2581–2594.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, *5*(2), 241–259.

Wolpert, D. H. (2002). The supervised learning no-free-lunch theorems. In R. Roy, M. Köppen, S. Ovaska, T. Furuhashi, & F. Hoffmann (Eds.), *Soft computing and industry* (pp. 25–42). London: Springer. https://doi.org/10.1007/978-1-4471-0123-9_3.

Wright, M. N. (2015). Ranger: A fast implementation of random forests. R package

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (ICML-03)* (pp. 928–936).