Automatic robust estimation for exponential smoothing: perspectives from statistics and machine learning

Devon Barrow[a,*], Nikolaos Kourentzes[b,c], Rickard Sandberg[d], Jacek Niklewski[e]

[a] *Birmingham Business School*
Department of Management, B15 2TT, United Kingdom
[b] *Skövde Artificial Intelligence Lab, School of Informatics, University of Skövde, Sweden.*
[c] *Lancaster University Management School*
Department of Management Science, Lancaster, LA1 4YX, United Kingdom
Email: nikolaos@kourentzes.com
[d] *Stockholm School of Economics*
Center for Data Analytics, Sveavägen 65, 113 83 Stockholm, Sweden
Email: Rickard.Sandberg@hhs.se
[e] *Coventry University*
Faculty of Business, Environment and Society, Coventry, CV1 5FB, United Kingdom
Email: aa6367@coventry.ac.uk

Abstract

A major challenge in automating the production of a large number of forecasts, as often required in many business applications, is the need for robust and reliable predictions. Increased noise, outliers and structural changes in the series, all too common in practice, can severely affect the quality of forecasting. We investigate ways to increase the reliability of exponential smoothing forecasts, the most widely used family of forecasting models in business forecasting. We consider two alternative sets of approaches, one stemming from statistics and one from machine learning. To this end, we adapt M-estimators, boosting and inverse boosting to parameter estimation for exponential smoothing. We propose appropriate modifications that are necessary for time series forecasting while aiming to obtain scalable algorithms. We evaluate the various estimation methods using multiple real datasets and find that several approaches outperform the widely used maximum likelihood es-

*Correspondance: D.K.Barrow, Birmingham Business School, Department of Management, B15 2TT, United Kingdom. Tel: +44 77962 67674

Email address: d.k.barrow@bham.ac.uk (Devon Barrow)

Preprint submitted to Expert Systems with Application

timation. The novelty of this work lies in (1) demonstrating the usefulness of M-estimators, (2) and of inverse boosting, which outperforms standard boosting approaches, and (3) a comparative look at statistics versus machine learning inspired approaches.

Keywords: Forecasting, Exponential smoothing, M-estimators, Boosting, Bagging

---

1. Introduction

In today's data-rich environment, the number of forecasts required to support operations decision making continues to increase to the point where automation becomes a prerequisite (Ord et al., 2017). For example, in retailing it is not uncommon that several thousands of forecasts are required on a daily (or shorter) frequency to support baseline forecasting (Fildes et al., 2019). Therefore, the reliability of the forecasting process becomes of paramount importance and can be judged by its ability to generate forecasts which provide consistent performance across time (Kourentzes et al., 2017). This requires approaches that guard against over-fitting and unstable forecast selection (Barrow and Kourentzes, 2016) to deliver forecasts which are consistent over time (Kourentzes et al., 2019a). We refer to this property as forecast consistency, that is, the tendency of forecasts across time to exhibit similar behaviour, and note that it is crucial for practice as it permits reliable planning and decision making.

In achieving this goal of robust automated forecasting, the exponential smoothing family of models is often the method of choice, widely used in practice due to its simplicity, reliability and established track record, both in research and application (Gardner Jr, 2006). It is also computationally very efficient, making it ideal for large scale forecasting (Ord et al., 2017). However, it is not immune to the aforementioned challenges, and even state of the art implementations (proposed by Hyndman et al., 2002) have their limitations, as discussed in Section 2.

This research focuses on investigating robust approaches to forecasting with exponential smoothing for automation. We conduct a large scale empirical evaluation of best in class approaches to robust parameter estimation for exponential smoothing. From the statistics standpoint, we consider a series of well researched M-estimators. From machine learning, we consider the well-known boosting approach and evaluate an adaptation based on the

application of an inverse weighting scheme, first evaluated by Kuncheva and Whitaker (2002). To our knowledge, this is the first application of boosting to exponential smoothing. In addition, we consider bagging, which has been already shown to help produce accurate exponential smoothing forecasts (Bergmeir et al., 2016). The contributions of this research can be summarised as follows:

1. Evaluate estimators for the widely used exponential smoothing family of forecasting models beyond the established maximum likelihood estimation, drawing from the statistically motivated M-estimators and the machine learning bagging and boosting. In doing so, we:
2. Propose a methodology for tuning M-estimators that is both appropriate for time series forecasting, but also scalable to address modern applications' needs;
3. Propose an adaptation of boosting and inverse boosting for forecasting with exponential smoothing;
4. Provide a contrast of statistics and machine learning approaches in terms of forecast accuracy, bias, and implementation efficiency implications, where the latter two are often overlooked in the literature even though they are of prime importance for practice.

We demonstrate the usefulness of M-estimators and the proposed inverse boosting approach. We find that both statistics and machine learning perspectives can result in superior estimators, in terms of forecast accuracy. However, when considering forecast bias, statistical M-estimators outperform their machine learning counterparts. Accuracy in terms of bias is of particular importance being a key determinant of the quality of the decisions supported by the forecasts (Sanders and Graman, 2009; Kourentzes et al., 2019b). Additionally, M-estimators are found to be more computationally efficient. Overall, we find that robust parameter estimation using the pseudo-Huber cost provides the best performance, with minimal computational cost implications. Nonetheless, the proposed inverse boosting variant remains competitive and demonstrates its usefulness for forecasting purposes over the conventional boosting logic. While statistics and machine learning inspired estimators continue being considered separately for forecasting applications, we highlight the need and the benefits of joint research attention in this area.

The rest of this paper is structured as follows: in Section 2 we introduce exponential smoothing and highlight the limitations of standard parameter

estimation approaches. In Section 3 we discuss the relevant literature for M-estimators, bagging and boosting. Section 4 outlines the various estimation methods considered in this research and the proposed modifications for working with exponential smoothing. Section 5 outlines the experimental design of our empirical evaluation and our findings, followed by a discussion in Section 6 and concluding remarks in Section 7.

2. Parameter estimation for the exponential smoothing models

In this section we briefly introduce the exponential smoothing family of forecasting models and the current state-of-the-art for estimating its parameters.

Exponential smoothing analyses a time series as the total of a local level, a local trend, and a local seasonality. These terms can interact additively (A) or multiplicatively (M), with the trend being linear or nonlinear (damped). Each component is updated by a common error process, which can also vary between additive or multiplicative resulting in 30 formulations. Table 1 depicts 15 of the 30 combinations excluding the error term which accounts for the remaining 15 formulations. The simplest of this family of models is the well-known simple exponential smoothing, suitable for forecasting data with no clear trend or seasonal pattern. This corresponds to the N, N combination in Table 1 that is typically modelled with additive errors, giving us the ANN model, where the first A standing for additive errors:

$$y_t = l_{t-1} + \varepsilon_t, \tag{1}$$

$$l_t = l_{t-1} + \alpha\varepsilon_t, \tag{2}$$

where y_t is the target time series observation at period $t = 1, \ldots, n$ and n being the sample size, l_t the local level at period t, $0 < \alpha < 1$ is a smoothing parameter, and $\varepsilon_t \sim N(0, \sigma)$ are i.i.d. errors. Therefore, the model prescribes that the observed time series is generated as a smooth local level with some additive error and the level itself is updated by $\alpha\varepsilon_t$. Lower values of α result in slow updating of the local level and vice versa. To produce a forecast for period $t + h$, where h is the forecast horizon, we take the conditional expectation of y_t to get the well-known simple exponential smoothing method:

$$\hat{y}_{t+h|t} = l_t = (1 - \alpha)l_{t-1} + \alpha y_t,$$

since $\varepsilon_t = y_t - l_{t-1}$. More complex models are structured similarly, where each component is described by an additional state equation. For more details the reader is referred to Hyndman et al. (2008) or Ord et al. (2017).

Table 1: Combinations of trend and seasonal components under a state space-based approach

Trend	Seasonal		
	None (N)	Additive (A)	Multiplicative (M)
None (N)	N, N	N, A	N, M
Additive (A)	A, N	A, A	A, M
Additive Damped (A_d)	A_d, N	A_d, A	A_d, M
Multiplicative (M)	M, N	M, A	M, M
Multiplicative Damped (M_d)	M_d, N	M_d, A	M_d, M

The estimation of the parameters of exponential smoothing has attracted considerable attention in the literature. Early work by Gardner Jr (1985) and later by Makridakis et al. (2008) recommend the minimisation of Mean Squared Error (MSE) to find good values for the smoothing parameters, and propose a series of heuristics for the estimation of the initial values for the components. Furthermore, to increase the reliability of the resulting forecasts, and their robustness to outliers, recommendations to keep the parameters below 0.3 or 0.5 have emerged (for examples see Makridakis et al., 1982; Johnston and Boylan, 1994). Following on the work by Hyndman et al. (2002) that embedded exponential smoothing within the single source of error state-space modelling framework, we use maximum likelihood estimation to obtain the optimal smoothing parameters, initial values, and σ. This further permits us to easily obtain prediction intervals and use information criteria to select between the alternative forms of exponential smoothing (Hyndman et al., 2008), substantially simplifying its use in practice. This has made the state-space formulation of exponential smoothing the current standard widely used in both research and practice (Gardner Jr, 2006; Ord et al., 2017). The maximum likelihood estimation is based on minimising the quadratic errors:

$$\mathcal{L}^*(\theta, \mathbf{x}_0) = n \log \left(\sum_{t=1}^{n} \varepsilon_t^2 \right) + 2 \sum_{t=1}^{n} \log |r(\mathbf{x}_{t-1})|,$$

where θ is a vector containing the model parameters, $\mathbf{x}_0$ is a vector containing the initial values, and $r(\mathbf{x}_{t-1})$ is equal to 1 for additive errors and μ_t, the

conditional mean of $y_{t|t-1}$ or simply the predicted value for period t, for multiplicative errors. It is equivalent to minimise the augmented sum of squared errors criterion (Hyndman et al., 2008):

$$\mathbf{S}(\theta, \mathbf{x}_0) = \left| \prod_{t=1}^{n} r(\mathbf{x}_{t-1}) \right|^{2/n} \sum_{t=1}^{n} \varepsilon_t^2.$$

Observe that for the case of additive errors, where $r(\mathbf{x}_{t-1}) = 1$, this becomes the well-known MSE.

For (1) we need to estimate three parameters, α, σ and the initial level l_0, which appears when we write (2) for l_1. As additional components (trend and seasonality) are introduced to the models then more parameters and initial values are needed. For example, for the damped-trend seasonal exponential smoothing, applied to monthly data, we need to estimate four smoothing parameters, fourteen initial values, twelve of those corresponding to the initial monthly seasonal profile, and σ. Therefore, optimising exponential smoothing can be fairly trivial or rather complex depending on the model form used. Poor parameterisation can result in erratic forecasts.

To illustrate this, we forecast a time series corresponding to quarterly observations of admissions to the accident and emergency department of a major UK hospital between 2013 and 2018 (see Figure 1). Beginning from $Q1$ 2017 we increase the fitting sample to $Q4$ 2017, creating four forecast origins. Forecasting from each origin, we adopt a fixed model structure, estimating a linear trend exponential smoothing model (requiring 2 smoothing parameters and 2 initial values). We forecast one year ahead from each origin. Observe that despite adopting the same forecasting model we get substantially different forecasts, by only marginally changing the fitting sample. This is due to the difficulty in parameter estimation for exponential smoothing and the sensitivity of the quadratic loss that is the basis of the maximum likelihood estimation. Therefore, while exponential smoothing is widely used in practice, particularly for large scale automatic forecasting, it is not uncommon that forecasts can lack consistency. This can have substantial cost implications, but also reduce the trust of users in the forecasts (Dietvorst et al., 2015).

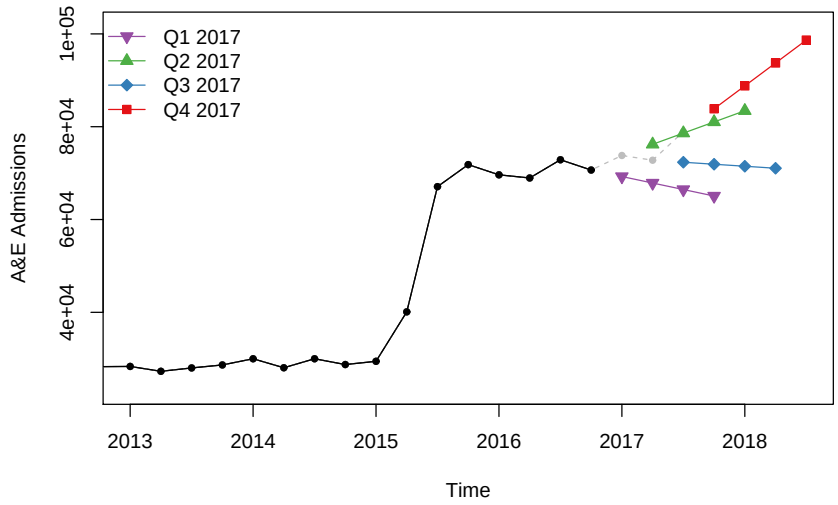


Figure 1: Rolling forecasts for an example time series using linear trend exponential smoothing, ETS(M,A,N).

3. Beyond maximum likelihood estimation for exponential smoothing

As the previous example illustrates, it is quite easy to find cases where the maximum likelihood estimation will either not perform well (Kourentzes et al., 2014), or will provide parameters that are on the bounds of the admissible range, suggesting that the resulting forecasts may be of poor quality (Gardner Jr, 1985). This has motivated the literature to investigate alternative estimation procedures, which we discuss below, drawing from statistics and machine learning.

3.1. Statistics: LAD-estimators and M-estimators for exponential smoothing

Maximum likelihood estimation is by far the most widely used approach to specify model parameters. However, in the literature, there are ample arguments suggesting limitations, especially when the observed values may be contaminated by outliers (Huber, 1981, 1992). In the context of robust estimation of time series models, both LAD-estimators (Least Absolute Deviation) and M-estimators (for 'maximum-likelihood like' estimators) are common choices, because they offer some protection against outliers.

There is very limited work in the use of LAD-estimators in the wider time series forecasting literature, although more has been done in a regression context, with Davis and Wu (2006) summarising the arguments in favour of LAD-estimators when handling heavy-tailed time series. McDonald and Xu (1994) find that LAD-estimators made ARIMA forecasts more accurate than least squares based ones for their sample of series. Gardner Jr (2006) makes the case for using absolute errors for optimising the smoothing parameters of exponential smoothing, given that it can capture only the local level, trend, and seasonal components and should not be influenced by large errors due to outlying observations. However, note that minimising the absolute errors provides optimal forecasts for the median of the distribution of the target variable (Gneiting, 2011), while exponential smoothing provides conditional mean forecasts.

M-estimators are used in the literature more frequently (for examples, see Maronna and Yohai, 2000; Lee et al., 2009; Baldauf and Silva, 2012; Maronna et al., 2018), due to their increased flexibility. For instance, one of the most commonly used functions is the Huber loss function (Huber, 1981). The Huber loss combines the absolute and squared error loss functions, yielding the sensitivity of the quadratic loss, and the robustness of the absolute loss. Nonetheless, there are not many examples of the use of M-estimators in time series models, beyond regression, with examples being the works by Cipra (1992) and Gelper et al. (2010). Cipra (1992) investigated expressions for simple forms of the exponential smoothing method using absolute loss and the Huber loss with fixed parameters (Huber, 1981) and provide limited, but encouraging empirical evidence. Gelper et al. (2010) look at the case of the trend exponential smoothing and recommend a pre-cleaning of the fitting sample, which was based on Huber loss. They provide evidence of the good performance of their pre-cleaning approach for the trend exponential smoothing and provide the formulation for extending this to the trend-seasonal case, but without extensive empirical evaluation for the latter. Crevits and Croux (2016) explore a direct implementation of the Huber loss on exponential smoothing models, and while reporting desirable behaviour in the presence of outliers on simulated data, it substantially under-performs when used on real data.

Lastly, it is desirable to work with loss functions that lend themselves to ease of implementation. For example, the commonly considered Huber loss M-estimator is not continuous and differentiable for all values. This has lead to the development of approximations such as the Pseudo-Huber loss,

also called Charbonnier loss (Charbonnier et al., 1994, 1997) or L1-L2 loss (Zhang, 1997), that overcomes these limitations.

The very limited use of M-estimators for extrapolative forecasting, and particularly for exponential smoothing, draws our attention to testing their performance. Their properties are attractive for large scale forecasting applications, where forecast consistency is important. In contrast to LAD-estimators, how to adapt M-estimators for exponential smoothing remains an open question, which we explore further in Section 4.1.

3.2. Machine learning: Bagging and Boosting for exponential smoothing

In dealing with uncertainty and improving the robustness of forecasts, the most widely employed machine learning methods have been bagging, short for bootstrap and aggregating (Breiman, 1996), and boosting (Schapire, 1990).

3.2.1. Bagging

The origins of bagging dates back to the seminal works of Breiman (1996, 1999), in which he proposed that a set of predictors could be estimated across multiple bootstrap samples of the original data, and subsequently averaged to generate a combined prediction. Since then, bagging has been extensively applied in machine learning for classification (Skurichina and Duin, 1998; Bauer and Kohavi, 1999; Hothorn and Lausen, 2005; Lemmens and Croux, 2006), regression (Breiman, 2001; Chen and Ren, 2009; Borra and Di Ciaccio, 2002), and more recently to time series forecasting (Inoue and Kilian, 2008; Bergmeir et al., 2016; Barrow and Crone, 2016b; Athanasopoulos et al., 2018; Dantas and Oliveira, 2018).

Bagging for exponential smoothing has received relatively little attention. Cordeiro and Neves (2009) proposed an approach for bootstrapping exponential smoothing obtaining promising results for long seasonal time series. However, they found that the forecasts from their bagged approaches were often not as good as conventionally generated forecasts, in particular for short time series.

Bergmeir et al. (2016) proposed a new methodology for bagging exponential smoothing, outperforming the approach of Cordeiro and Neves (2009). Petropoulos et al. (2018) show empirically that bagging for exponential smoothing performs well as it mitigates the data uncertainty related to the inherent randomness of the in-sample time series data, and the modelling uncertainty related to parameter estimation and model selection. Although the authors find relatively moderate improvements from bagging exponential

smoothing, in stark contrast to those obtained when bagging neural networks and decision trees (Barrow and Crone, 2016a), this work provides a useful cross-over between statistical forecasting and machine learning that demonstrates the need for further research in the area.

3.2.2. Boosting

Boosting, first introduced by Schapire (1990), is a general method for sequentially aggregating a set of predictive models, each trained on a re-weighted (or re-sampled, if applicable) training set. The main principle underlying the sequential re-weighting and/or re-sampling of observations in the training set is to allow the model to 'learn' the more difficult to predict observations. Poorly predicted observations receive higher weight in successive boosting rounds. The method has evolved several times since its introduction and can be broadly classified into three families. The first relates to the first boosting algorithms proposed by Schapire (1990) and Schapire et al. (1998), developed solely for classification problems. These were followed by the development of the AdaBoost algorithm by Freund and Schapire (1997), who extended boosting to both classification and regression. The most recent addition to the family of boosting methods is Gradient Boosting developed by Friedman (2001, 2002). This was inspired by the work of Friedman et al. (2000) who linked boosting to function estimation and additive function expansion. Boosting in all its forms is now a standard approach in machine learning, applied to regression (Avnimelech and Intrator, 1999; Drucker, 1997; Shrestha and Solomatine, 2006; Gey and Poggi, 2006) and classification problems (Bühlmann and Yu, 2003; Sun et al., 2007; Dettling and Bühlmann, 2003; Dietterich, 2000; Al-Shemarry et al., 2018; Owusu et al., 2014).

Kuncheva and Whitaker (2002) investigate a modification of boosting for classification tasks, the 'inverse boosting', which attempts to reduce the focus of the model on the outlying and difficult to learn observations. The reported results are not encouraging, which has led to relatively little attention to this modification. Gao et al. (2013) revisit this and find only occasional improvements in classification performance above normal boosting.

Boosting has also gained some prominence in time series forecasting. Since the 2012 survey of boosting algorithms in time series forecasting by Barrow (2012) who found 38 studies involving boosting, there have been numerous applications, involving artificial neural networks (Barrow and Crone, 2016a; Khwaja et al., 2017), decision trees (Krauss et al., 2017; Persson

et al., 2017) and regression (Taieb and Hyndman, 2014; Mittnik et al., 2015) approaches to name a few. However, while boosting has been applied to machine learning algorithms, such as neural networks and decision trees (Zheng, 2010; Israeli et al., 2019), to our knowledge, boosting has not been applied to exponential smoothing, even though this is one of the most widely used forecasting models in practice. This motivates our investigation into using boosting for forecasting with exponential smoothing.

Furthermore, we argue that there are parallels between inverse boosting and M-estimators, both attempting to moderate the effect of extreme values and instead focus on the underlying structure of a time series. This property of inverse boosting can be advantageous for exponential smoothing models. However, inverse boosting has not been investigated in a time series forecasting context and this is the first study that looks into boosting and inverse boosting for exponential smoothing, detailed further in Section 4.3.

4. Robust methods for estimating exponential smoothing models

Drawing on the existing work on LAD and M-estimators, bagging, and boosting, in this section we detail how these approaches work and our proposed modifications to make them applicable for exponential smoothing models.

4.1. LAD- and M-estimators

4.1.1. The absolute error loss

The conventional quadratic loss is sensitive to extreme errors and therefore we consider loss functions that are less responsive to large errors. The first alternative is to use the absolute loss, as summarised by the Mean Absolute Error (MAE):

$$\text{MAE} = n^{-1} \sum_{t=1}^{n} |\varepsilon_t|,$$

where n is the sample size and ε_t are the one-step-ahead in-sample errors. The importance of an error increases linearly to its size, instead of proportionally, as is the case with squared errors. Minimising MAE results in forecasts that are optimal to the median of the distribution of the target variable.

4.1.2. The Huber loss

A second alternative is the Huber loss which combines the absolute and quadratic loss in the following way (Huber, 1992):

$$l(k)_{\text{Huber}} = \sum_{t=1}^{n} (u_t), \quad u_t = \begin{cases} \varepsilon_t^2, & \text{if } |\varepsilon_t| \leq q \\ |\varepsilon_t|, & \text{otherwise} \end{cases},$$

where q is a threshold. Therefore, the effect of very large errors scales linearly, while errors within the threshold are quadratic. Typically, when considering the Huber loss, residuals are scaled to unit variance and the threshold q is set to a reasonable value, such as 1.345, so as to achieve high efficiency (Dutter and Huber, 1981). Scaling the residuals requires a robust estimation of their variance, which in a time series context is not trivial (Gelper et al., 2010), given that the underlying data generation process of the target variable is unknown, and the fitted forecast function has to contend with several uncertainties. Instead, we propose setting q as the $p\%$ quantile of the error distribution. The advantage of this approach is that we avoid the need to scale the residuals, since the scale is part of the resulting quantile, but also bounded to $50\% < p \leq 100\%$, as the 50% quantile is the median, giving us a limited search space. This approach results in a data-driven setting of q for the target time series, which may help overcome the poor reported performance of Huber loss based exponential smoothing by Crevits and Croux (2016), where the standard prefixed threshold was considered. Herein also lies another contribution of this work.

4.1.3. The Psedo-Huber loss

The Huber loss introduces a discontinuity at points $-q$ and q that the Psedo-Huber loss overcomes, offering a smooth approximation of the former:

$$l(k)_{\text{Pseudo-Huber}} = \sum_{t=1}^{n} \left(q^2 \left(\sqrt{1 + \left(\frac{\varepsilon_t}{q} \right)^2} - 1 \right) \right).$$

The threshold q alters the steepness of the loss function, adjusting its sensitivity to higher errors. Similarly to the Huber loss, we do not scale the residuals and instead tie q to an appropriate quantile of the error distribution.

4.1.4. Setting the threshold q

Both the Huber and the Pseudo-Huber loss functions require the user to specify the threshold q. As we argued before, given the various uncertainties

in the specification of the forecasting equation, the nature of the residuals can vary substantially across time series. Our objective is to devise a methodology that requires minimal user intervention and therefore we propose setting the threshold in a data driven approach, for each time series. We split the fitting sample of a time series into training and validation sets. We use the training set to optimise model parameters for a given q and record the resulting errors in the validation set. The q that minimises the validation set errors is selected. This follows the ideas of cross-validation, but with the necessary restrictions for time series modelling. Purely regression models can be fully cross-validated, however models that have recursive forms, such as exponential smoothing and ARIMA cannot, and therefore one has to retain the time order and rely on a validation set at the end of the fitting sample (Ord et al., 2017). Note that given that we have tied q to the $p\%$ quantile of the error distribution, without losing too much search resolution, we can restrict the search into 49 possible percentiles, from the 51^{st} to the 100^{th}, substantially speeding the search. We trialled a more exhaustive search, but found no benefits, especially when coupled with the additional computational demands.

4.1.5. Summary of loss functions

The aforementioned loss functions are visualized in Figure 2 which provides examples of the effect on the shape of Huber and Psedo-Huber loss for two different values of q. A small value shapes both Huber and Pseudo-Huber closer to the absolute loss, while higher values shape the loss functions closer to quadratic loss. The Pseudo-Huber requires higher values of q to approximate the shape of Huber, for a given threshold. For instance, the Pseudo-Huber loss in panel (ii) of Figure 2 is similar to the Huber loss in panel (i).

4.2. Bagging for Exponential Smoothing

In this study, we adopt the bagging approach of Bergmeir et al. (2016) and Petropoulos et al. (2018) as this represents the best known benchmark for bagging exponential smoothing. The approach first applies a Box-Cox transformation (Box and Cox, 1964) to stabilize the variance of the time series and to ensure the time series can be modelled additively. The selection of the λ parameter for the Box-Cox transform is automated using the procedure of Guerrero (1993). The data is then decomposed based on the existence or non-existence of seasonality. Non-seasonal time series are decomposed using

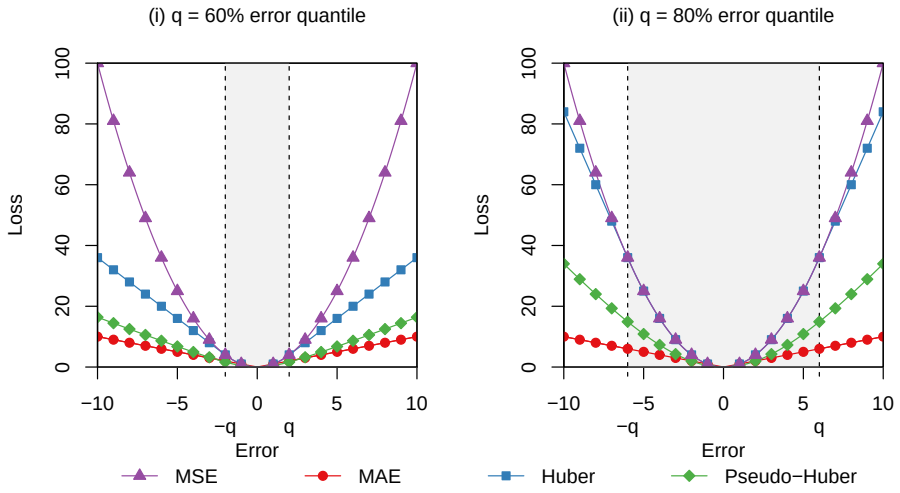


Figure 2: Loss functions for MSE, MAE, Huber and Pseudo-Huber for two different values of q, expressed as a quantile of the error distribution.

the Loess-based procedure of Shyu et al. (2017), while seasonal time series are decomposed using the Seasonal and Trend decomposition procedure of Cleveland et al. (1990) based on Loess, also referred to as STL decomposition. The error term or remainder obtained from the decomposition, after any trend or seasonal terms are removed, is subsequently bootstrapped. However, the errors while assumed stationary may be autocorrelated. While several bootstrap methods are possible, the authors suggest using the moving block bootstrap by Kunsch (1989). The bootstrap error series is then combined with the original trend and/or seasonal structural components to obtain a new bootstrapped series. Repeated bootstraps of the error (remainder) series, produce multiple new bootstrapped series. Each bootstrapped series, including the original time series, can then be used to fit a different exponential smoothing model whose forecasts are averaged to produce a combined forecast giving the final bagged forecast.

4.3. Boosting for Exponential Smoothing

We describe AdaBoost.R2 (Drucker, 1997) and AdaBoost.RT (Shrestha and Solomatine, 2006), two variants of the original AdaBoost algorithm used in time series forecasting and adapt these for exponential smoothing. Boosting is done over k iterations, with K the maximum number of iterations.

Initially, we set a constant vector of weights $w_{t,k=1} = 1$, for $t = 1, \ldots, n$. AdaBoost.R2 and AdaBoost.RT work as follows, for each iteration k:

1. Calculate a weight $p_{t,k} = w_{t,k} / \sum_{t=1}^{n} w_{t,k}$, where $p_{t,k} \in [0, 1]$ by construction.
2. Estimate model parameters, weighting each ε_t by $p_{t,k}$.
3. Given the estimated model, calculate the loss $L_{t,k}$ using equations (3) or (5) for AdaBoost.R2 or AdaBoost.RT respectively and evaluate the relevant stopping criterion using the average loss $\bar{L}_k = \sum_{t=1}^{n} L_{t,k} p_{t,k}$.
4. If $k \leq K$ and the stopping criterion is not met, update weights $w_{t,k+1} = w_{t,k} b_k^{(1-L_{t,k})}$, where b_k is defined below for the two boosting algorithms, and return to step 1. Otherwise, stop the algorithm at iteration K^* and proceed to the final step.
5. Construct the final forecast as a combination of the forecasts at each iteration with weights b_k:

$$\hat{y}_{t+h} = \sum_{k}^{K^*-1} \left(\frac{b_k \hat{y}_{t+h,k}}{\sum_{k}^{K^*-1} b_k} \right).$$

Observe that in step 2 we weight the in-sample one-step-ahead errors, so as to achieve the desired behaviour at each iteration k. Step 5, linearly combines all forecasts up to iteration $K^* - 1$ using as weights the normalised values of b_k. In principle, if these values were known a priori, then boosting could be seen as a weighted estimator, much like the aforementioned M-estimators.

4.3.1. The AdaBoost.R2 Algorithm

In the case of AdaBoost.R2, the function for calculating loss $L_{t,k}$ is:

$$L_{t,k} = \left| \frac{y_t - \hat{y}_{t,k}}{max(|y_t - \hat{y}_{t,k}|)} \right|, \tag{3}$$

where $\hat{y}_{t,k}$ is the fitted value at period t for the model trained on the k^{th} iteration and $max(|y_t - \hat{y}_{t,k}|)$ is the maximum prediction error over all observations, so that $L_{t,k} \in [0, 1]$. For each iteration we calculate the forecast combination weight b_k as:

$$b_k = \frac{\bar{L}_k}{1 - \bar{L}_k}. \tag{4}$$

We use as a a stopping criterion $\bar{L}_k > 0.5$ that ensures that $b_k > 0$.

4.3.2. The AdaBoost.RT Algorithm

In this case the loss for each iteration is calculated as:

$$L_{t,k} = \begin{cases} 1, & \text{if } |y_t - \mu_{t,k}| > \phi \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

where ϕ is a threshold controlling which observations are classed as 'difficult to predict'. As with the M-estimators, we set ϕ to a percentile of the distribution of the residuals. This overcomes any scaling issues that are part of the original AdaBoost.RT algorithm that uses percentage errors. Note that the loss function of AdaBoost.RT can be viewed as an outlier detector. Higher values of threshold ϕ focus on larger errors. As before, at each iteration k we calculate the average loss $\bar{L}_k$ and the forecast combination weight $b_k = \bar{L}_k$. The calculation of b_k implies a stopping criterion of $\bar{L}_k = 0$, implying that no observations regarded as extreme remain.

4.4. Inverse boosting

Boosting, with its focus on 'difficult to predict' observations, has the potential to strongly bias the resulting forecasts when the time series has high noise and outliers. Consider the extreme case wherein parameterising an exponential smoothing model an outlier is appropriately weighted minimally. That observation would generate a large error and consequently boosting would weight it heavily, worsening a possibly good estimate. To overcome this we adapt the inverse boosting (Kuncheva and Whitaker, 2002; Gao et al., 2013) for forecasting with exponential smoothing. The weight update procedure is modified as follows:

$$\mathrm{w}_{t,k+1} = w_{t,k} b_k^{(L_{t,k}-1)} \tag{6}$$

The result is that observations with the smallest error will have the largest weight, while those with the largest errors get the smallest weight.

We argue that inverse boosting is particularly relevant for the case of time series forecasting where data can be very noisy, contain outliers, or simply have a limited fitting sample. Exponential smoothing models do not capture extreme values explicitly and instead rely on the smoothing parameters to mitigate their effect on the estimation of the local level, trend and season. When these remain untreated in the time series they can heavily bias the resulting model parameters. Kourentzes and Petropoulos (2016) demonstrate

this in a promotional forecasting setting, providing evidence of substantial parameter bias and loss of forecast accuracy when maximum likelihood estimation is used. This is unsurprising, given that for real data all forecasting models are to some degree misspecified, which is bound to influence model parameter estimates (Chatfield, 1995). Inverse boosting is expected to weight such extreme observations less and approximate the structural components of the underlying data generating process more accurately. This has parallels with M-estimators, as well as research that has looked at ways to transform the raw data to aid model estimation (Gelper et al., 2010; Koehler et al., 2012).

It should be noted that boosting (normal or inverse) has no explicit outlier detection mechanism. In implementing boosting, we weight directly the fitting error of exponential smoothing. This provides a data-driven approach to weighting observations, and to separating observations into well or poorly predicted. Note that AdaBoost.RT adopts a threshold loss function, requiring a threshold ϕ. The setting of this threshold in AdaBoost.RT is equivalent to establishing a hard boundary to determine which observations are well or poorly predicted. This can be regarded as an implicit outlier classification.

4.4.1. An Illustrative Example

To demonstrate our argument of the applicability of inverse boosting for exponential smoothing, we generate a time series with 100 periods following:

$$y_t = 100 + 250I_t + \varepsilon_t, \tag{7}$$

where I_t is an indicator variable that is equal to 1 for period 25 and zero otherwise, generating an outlier in that period, and $\varepsilon_t \sim N(0, 40)$. This data generating process is chosen due to its simplicity, having known a mean of 100, and a single outlier. The objective is to understand the impact of an outlier on parameter estimation of exponential smoothing with the two boosting variants, given a very simple data generating process that is in principle easy to model. The resulting time series is illustrated in Figure 3 panel (i). For this example, we implement AdaBoost.RT to facilitate the estimation of the mean of the time series, in the normal and the inverse way, as presented in panel (ii) of the figure. We set the ϕ threshold parameter of AdaBoost.RT to the 75^{th} quantile of the error distribution. Further details on the implementation of boosting can be found in Section 5.3. The inverse converges in 10 iterations, while the alternative converges in 22 iterations

with minimal change from the 10^{th} iteration in Figure 3 panel (ii). Finally, panels (iii) and (iv) present the weights $p_{t,k}$ for each period of the time series, across the first 10 iterations, for normal and inverse boosting respectively. Light colouring relates to small weights and the period with the outlier is indicated by a small arrow.

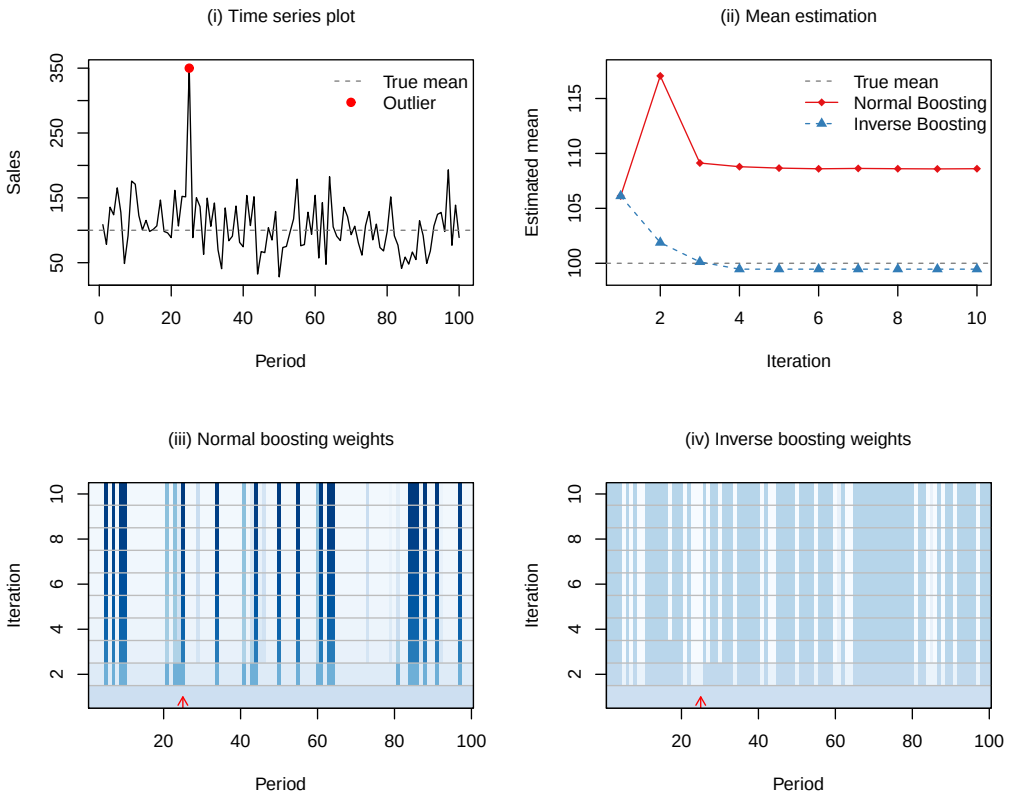


Figure 3: Illustrative example of normal and inverse boosting for AdaBoost.RT algorithm. Panel (i) plots a simulated time series with a true mean of 100 units and an outlier at period 25. Panel (ii) presents the evolution of estimate of the mean with the two alternative boosting methods, against the true mean. Panels (iii) and (iv) provide the evolution of weights $p_{t,k}$ across iterations, where colours from lighter to darker signify smaller to larger weights. Period 25 is indicated by a small arrow.

We observe, in panel (ii), that inverse boosting iteratively lessens the impact of the outlying and noisy periods to converge almost to the true value of 100 units. This is not the case for normal boosting, where it converges to a

much higher value, clearly biased from the noisy and outlying observations. This is indicated in panel (iii), where the dark coloured periods indicate large weights, where boosting is focused. Gradually, across iterations these periods get increasingly bigger weights, while other periods get increasingly smaller weights, starting from equal weights in the first iteration. For inverse boosting, in panel (iv), the behaviour is the opposite. These observations receive smaller weights, but also the difference of the weighting between periods is smaller, as indicated by the smaller differences in colour.

This example briefly illustrates our motivation for using inverse boosting for time series forecasting with exponential smoothing. In the next section, we contrast the aforementioned estimators using empirical data, to demonstrate the strengths and weaknesses of each.

5. Empirical evaluation

In this section we describe the main components of the experimental design. Using multiple datasets, forecast measures, and considering various characteristics of the time series, we investigate the forecast performance of the various approaches to estimation for exponential smoothing.

5.1. Datasets

For the empirical evaluation we use three datasets. The first is the well known M3-competition dataset (Makridakis and Hibon, 2000), which contains yearly, quarterly, monthly sampled data, along with a small set of series of unknown sampling frequency. The second dataset describes the demand of a fast-moving consumer goods manufacturer (FMCG), and allows exploration of weekly time series Barrow and Kourentzes (2016) and Kourentzes et al. (2019a). The last dataset records tourism flows in different regions of Australia. A detailed description is given by Athanasopoulos et al. (2009). We use a variety of datasets, sampling frequencies and forecast horizons, so as to evaluate the performance of the competing estimation methods in a wide set of conditions.

The M3-competition dataset has the advantage that it contains a large diversity of time series and has been widely studied in the forecasting literature, where exponential smoothing has been found to provide very competitive forecasts (Makridakis and Hibon, 2000; Hyndman et al., 2002, 2008). However, a large selection of time series originate from macro-economic, financial or otherwise aggregate indicators, which arguably do not reflect the typical

business forecasting applications. We address this limitation by including the FMCG and the Tourism dataset, which are tied to specific applications, have diverse characteristics from the M3-competition dataset and exponential smoothing, using maximum likelihood estimation, has been shown to perform well, providing a competitive benchmark for our evaluation. Furthermore, both these applications require the generation of a large number of forecasts and therefore requiring reliable large-scale automatic forecasting approaches, which is one of our motivations in investigating alternative estimators to maximum likelihood.

We consider different forecast horizons, and use longer test sets than the forecast horizons to facilitate a rolling origin evaluation (Ord et al., 2017) and collect a distribution of forecast errors. This enables us to increase the validity of our findings and facilitate statistical testing. Table 2 provides details about the sampling frequency, number of time series, size of the test set and forecast horizon for each dataset.

Table 2: Datasets used for the empirical evaluation

Dataset	Sampling	No. of series	Horizon	Sample size	Test set
M3	Annualy	645	4	20-47	6
M3	Quarterly	756	4	24-72	8
M3	Monthly	1428	12	66-144	18
M3	Other	174	12	71-104	18
FMCG	Weekly	229	13	173	52
Tourism	Quarterly	89	4	36	12

5.2. Evaluation design

In assessing the forecasting performance we track both the forecast accuracy and the forecast bias. We use the Average Relative Mean Absolute Error (AvgRelMAE), proposed by Davydenko and Fildes (2013). First we calculate the Mean Absolute Error of the forecasts of interest and subsequently we scale them according to a benchmark error. In this case, as benchmark

we use the result of the standard maximum likelihood estimation.

$$\text{MAE}_m = \frac{1}{n}\frac{1}{h}\sum_{j=1}^{n}\sum_{i=1}^{h}|y_{m,j+i-1} - \hat{y}_{m,j+i-1}|,$$

$$\text{AvgRelMAE} = \left(\prod_{l=1}^{m}\frac{\text{MAE}_{A,m}}{\text{MAE}_{B,m}}\right)^{1/m},$$

where $y_{m,j+i-1}$ is the observed value of the mth time series, over $i = 1, \ldots, h$ step-ahead forecasts, and $j = 1, \ldots, n$ forecast origins. The forecast $\hat{y}_{m,j+i-1}$ follows similar indices. The AvgRelMAE is the geometric mean of the ratio of the method of interest $\text{MAE}_{A,m}$ over the benchmark $\text{MAE}_{B,m}$ over all m series of the dataset. AvgRelMAE avoids computational issues of other scale-independent metrics, such as the Mean Absolute Percentage Error, which requires non-zero actuals and is not symmetric on reporting positive and negative errors. Furthermore, it is simple to interpret, where any value below 1 signifies an improvement over the benchmark of $(1 - \text{AvgRelMAE})100\%$.

In similar fashion, we define the bias metrics, where we first calculate the Mean Error (ME) of each forecast and then calculate the geometric mean of their absolutes (AvgRelAME):

$$\text{ME}_m = \frac{1}{n}\frac{1}{h}\sum_{j=1}^{n}\sum_{i=1}^{h}(y_{m,j+i-1} - \hat{y}_{m,j+i-1}),$$

$$\text{AvgRelAME} = \left(\prod_{l=1}^{m}\left|\frac{\text{ME}_{A,m}}{\text{ME}_{B,m}}\right|\right)^{1/m}.$$

The AvgRelAME loses the sign information, so it does not track whether we over- or under-forecast, but on the other hand retains the size of the bias. We are interested in this calculation as the size of the bias is strongly connected with the economic impact of the decisions supported by the forecasts (Sanders and Graman, 2009; Kourentzes et al., 2019b).

Finally, given the rolling origin design, we conduct statistical testing of the error distributions, to evaluate whether any reported differences are significant. To this end, we adopt the non-parametric Friedman test and the post-hoc Nemenyi tests (Hollander et al., 2013). We do this to avoid multiple pairwise testing and any distributional assumptions. First, we apply the Friedman test to identify whether there is evidence that at least one of the

competing methods performs significantly different from the rest. We then apply the weaker Nemenyi test to group the various methods when there is no evidence of significant differences. The tests are used as implemented in the `tsutils` (Kourentzes, 2019) package for R (R Core Team, 2018), using the function `nemenyi()`.

5.3. Estimating methods

We produce all forecasts using the exponential smoothing family of models (Hyndman et al., 2002, 2008). The appropriate model form or structure for exponential smoothing is selected using the Akaike Information Criterion corrected for small sample sizes (AICc, Burnham and Anderson, 2004). As AICc requires the likelihood of the alternative models to be maximised, we do the selection of the model form using parameters optimised by maximum likelihood estimation. We retain the selected model form for all competing parameter estimation methods. This restriction can be easily lifted by using cross-validated errors for model selection (Fildes and Petropoulos, 2015; Kourentzes et al., 2019a), however, this would add a further degree of variability in our comparisons, and therefore we opt to keep the selected model form fixed for each time series.

We use the selected model form together with parameters estimated via maximum likelihood as the benchmark approach, hereafter referred to as *Base*. From the statistical approaches, the first evaluated is absolute errors, instead of quadratic, referred to as *MAE*. We also use the Huber and Pseudo-Huber loss functions, hereafter referred to as *Huber* and *PHuber* respectively. Both require setting a cut-off to switch between the different regimes of the loss function. This cut-off is specified using cross-validation, with 20% of the training sample as a validation set, calculating one-step ahead errors over this period. We directly calculate the cut-off point as a percentile of the error distribution and therefore have no need to calculate a robust scaling variance for the errors. We trialed Huber and Pseudo-Huber loss functions with a fixed threshold, as recommended in the literature (Huber, 1981, 1992; Kelly, 1992), but these resulted in inferior results that are not reported here.

From the set of machine learning approaches, we use bagging for exponential smoothing, as proposed by Bergmeir et al. (2016) and refined by Petropoulos et al. (2018). This is referred to as *Bagging*. We also implement boosting and the proposed inverse boosting, hereafter called *Boost* and BoostInv. Results are presented for the AdaBoost.RT algorithm only, as the performance of AdaBoost.R2 was substantially worse. For both Boost and

BoostInv we set the AdaBoost.RT threshold to the 75^{th} quantile of the error distribution.

Although it is possible to cross-validate this threshold, we found that this was computationally prohibitive.

All forecasts are generated using the `forecast` package (Hyndman et al., 2018) for R (R Core Team, 2018), using the function `ets()` for generating the exponential smoothing forecasts and the function `baggedETS()` for generating the bagged exponential smoothing forecasts, as implemented by the authors of the aforementioned papers.

5.4. Results

First, we present the summary statistics for accuracy and bias size. These are followed by the results of the aforementioned statistical tests to highlight where the reported differences in performance are significant. Subsequently, we analyse the performance of the competing approaches by the characteristics of the time series.

5.4.1. Summary statistics

Table 3 summarises the AvgRelMAE and AvgRelAME, reporting accuracy and bias size respectively, across methods, and overall across datasets. In each column, the best performing method is highlighted in boldface. The Base case is used as the denominator in the calculation of the error metrics, and therefore has always the value of 1.

First, we focus on AvgRelMAE. Overall, across all datasets, we see that of the M-estimators', both MAE and P-Huber appear to be advantageous over Base, the conventional maximum likelihood estimation. P-Huber always improves, albeit in some cases only marginally, upon Base, while MAE has equal or better accuracy compared to Base, apart from the M3 yearly series, where it is marginally less accurate. Huber behaves more erratically, with very poor performance for the M3 yearly and Tourism series, while offering gains for the M3 quarterly and FMCG series. Overall, it performs poorly. Of the machine learning inspired approaches, Bagging and Boost do not provide overall more accurate results than Base. More specifically, Boost performs always worse, while Bagging is the most accurate on the M3 monthly series out of all competing methods and performs well for the FMCG and Tourism sets. On the other hand, it is outperformed by Base in all remaining sets. The proposed BoostInv performs much better, being the overall best machine learning inspired approach, with its accuracy closely matching that of MAE.

Table 3: Forecasting performance summary

Method	Dataset						
	M3				FMCG	Tourism	Overall
	Year	Quarter	Month	Other			
AvgRelMAE (accuracy)							
Base	1	1	1	1	1	1	1
MAE	1.003	0.987	1	0.996	**0.978**	**0.971**	0.989
Huber	1.578	0.997	1.034	1.214	0.984	2.200	1.274
P-Huber	**0.999**	**0.985**	0.996	**0.980**	0.985	**0.971**	**0.986**
Bagging	1.035	1.035	**0.975**	1.049	0.993	0.938	1.003
Boost	1.074	1.052	1.159	1.095	1.133	1.128	1.106
BoostInv	1.005	0.997	1.002	0.991	0.982	0.972	0.991
AvgRelAME (bias)							
Base	1	1	1	1	1	1	1
MAE	**0.968**	**0.967**	**0.990**	1.003	0.938	0.943	**0.968**
Huber	1.559	0.981	1.074	1.247	0.950	2.013	1.255
P-Huber	0.973	0.981	1.036	0.996	0.947	**0.922**	0.975
Bagging	1.039	1.125	1.019	**0.958**	1.028	0.979	1.023
Boost	1.065	1.113	1.174	1.003	1.591	1.142	1.168
BoostInv	1.024	0.991	1.024	1.019	**0.884**	0.937	0.978

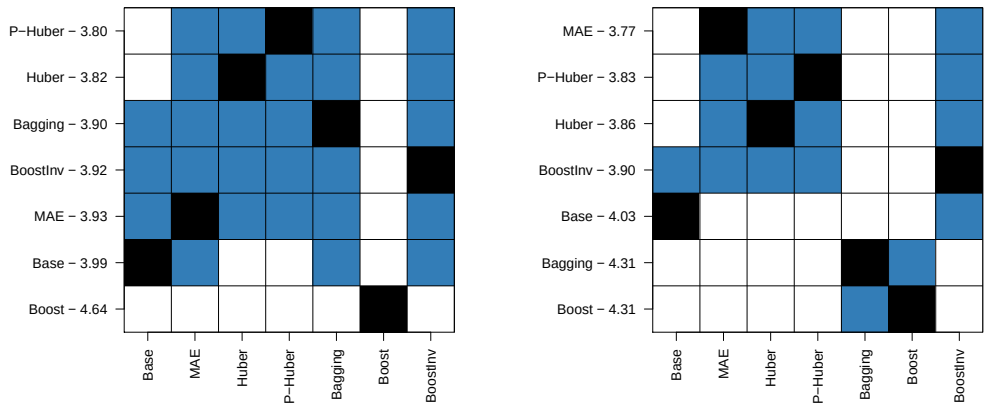
The general overall message here is the consistent performance of P-Huber which always improves over the Base.

The bias size results (AvgRelAME) are qualitatively similar. MAE is the best performer, closely followed by P-Huber and BoostInv. Huber's results remain erratic, while the bias size statistics for both Bagging and Boost demonstrate poor performance compared to the benchmark Base. This aspect of the evaluation has often been overlooked, even though there is clear evidence of the importance of bias for the economic value of forecasts, as discussed before. Crucially, it shows consistent benefit from using MAE or P-Huber, over the benchmark Base.

5.4.2. Friedman and Nemenyi test results

Both AvgRelMAE and AvgRelAME, due to the double aggregation in calculating first the MAE and ME respectively and subsequently their geometric means, can give the misleading impression of small differences. We test the results using the Friedman and Nemenyi tests, to evaluate whether the reported differences are due to real effects, or to randomness. Figures 4a

and 4b visualise the results for the accuracy and bias size measurements respectively. For both cases, the Friedman test indicates significant differences (p-value: 0.000). The plots provide the Nemenyi resulting groups. The vertical axis in the plots ranks the methods according to their mean rank, which is provided as well, across all time series. As the mean rank does not consider the size of errors, but merely the ranking (both Friedman and Nemenyi tests are non-parametric), the ordering of methods is different from the overall performance reported in Table 3. The horizontal axis orders the methods in the same way as listed in Table 3. The plots can be read either horizontally or vertically, where all methods with shaded cells, in a row/column, belong to the same group. Methods belonging to the same group have statistically insignificant differences at 5% level. For instance, for the AvgRelMAE (Figure 4a), the first column tests from the Base's accuracy (highlighted with a black cell) and indicates that there is no evidence of significant differences with the accuracy of Bagging, BoostInv, and MAE, while P-Huber and Huber are significantly better and Boost is significantly worse.



(a) AvgRelMAE (accuracy).

(b) AvgRelAME (bias).

Figure 4: Nemenyi test results across datasets

The tests on the accuracy statistics are illuminating in showing the strength of the various estimators evaluated here. Huber and P-Huber are grouped together as the most accurate. They do not exhibit significant differences from Bagging, BoostInv or MAE, but do so from the benchmark Base and the poor performing Boost. On the other hand, for Bagging, BoostInv and

MAE that belong to the same group there is not enough evidence of significant differences from Base, even though they exhibit better mean rank. The bias size test, groups together MAE, P-Huber, Huber, which do not exhibit significant differences from BoostInv, but are significantly less biased than Base, Bagging and Boost. Testing from BoostInv, there are no significant differences with Base while strikingly Bagging and Boost are grouped together as having significantly higher bias than Base or all other alternatives.

5.4.3. Time series structure and noise level

Contrasting the results of statistical testing with Table 3, we observe that Huber demonstrates two very distinct behaviours. Where its AvgRelMAE and AvgRelAME statistics are poor, it ranks well in the resulting Nemenyi groups. This is caused by the presence of a limited number of extreme errors. This leads us to explore the error distributions across time series type, and noise level, an analysis of particular relevance given that we evaluate robust approaches to exponential smoothing. The results are shown in Figure 5, where we provide snapshots of the distributions per method, split by type of time series. The snapshots of the distributions are boxplot inspired, providing the 5%, 25%, 50%, 75% and 95% quantiles of the distribution and the geometric mean. We use these instead of boxplots to summarise economically the distributions, without providing all outliers beyond one and half times the inter-quartile range, to avoid visual clutter. Any distributions that have extreme errors beyond the range of the vertical axis are indicated with a triangle on the top of the plot. Large differences between the median and the geometric mean point to asymmetries in the distributions, also captured by the reported quantiles, and the effect of the few extreme errors, above the 95% quantile. The time series types are separated according to the selected ETS model, which was done using AIC corrected for sample size. The methods are ordered according to their mean ranks, reported in Figure 4a.

Looking at the snapshot of the distribution of RelMAE for Huber, for the level time series, the reason for the discrepancy between Table 3 and Figure 4a becomes evident. The geometric mean is substantially distorted by extreme errors. The results of Boost indicate similar influences, while the other methods do not have such extreme effects. Looking at the results for the level series, it is interesting to observe that both Bagging and Boost have wide distributions, indicating erratic performance over the Base. P-Huber, Huber, and BoostInv demonstrate relatively tight distributions, while MAE is slightly wider. The same is true for the trend time series. The

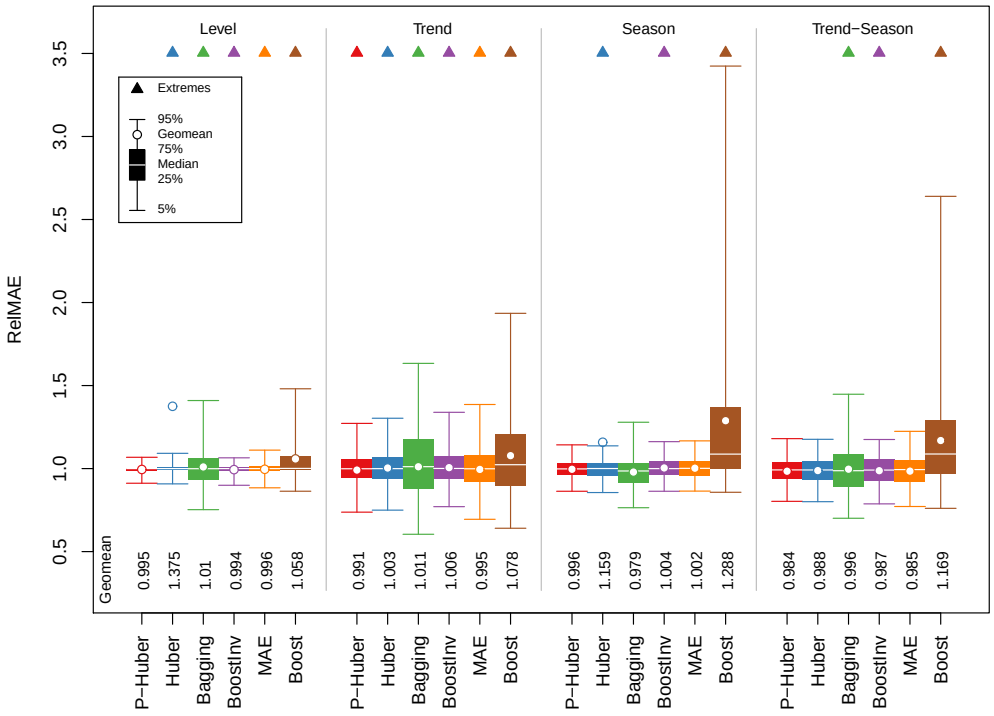


Figure 5: Snapshots of the distributions of RelMAE by time series category. The 5%, 25%, 50%, 75% and 95% quantiles are provided, along with the geometric mean (geomean) for each method. Cases where the distribution has extreme errors beyond the scale of the vertical axis (higher than the 95% quantile) are indicated by a triangle.

seasonal time series drive the poor performance of Boost. Observe that Bagging retains a relatively wide distribution, even though its geometric mean is the best reported. The rest of the methods have tight distributions. The results for the Trend-Season time series are consistent with the combined performance on the Trend or Season time series. The distributional view of the errors helps explain the weaker ranking of Bagging compared to P-Huber and Huber. The two latter provide improvements over Base more reliably, with tighter RelMAE distributions, irrespective of the type of time series. BoostInv results are interesting in that although they are not significantly better than Base, its distributions of errors are tight for all types of time series, attesting to the reliability of the proposed method, particularly when

contrasted with normal boosting.

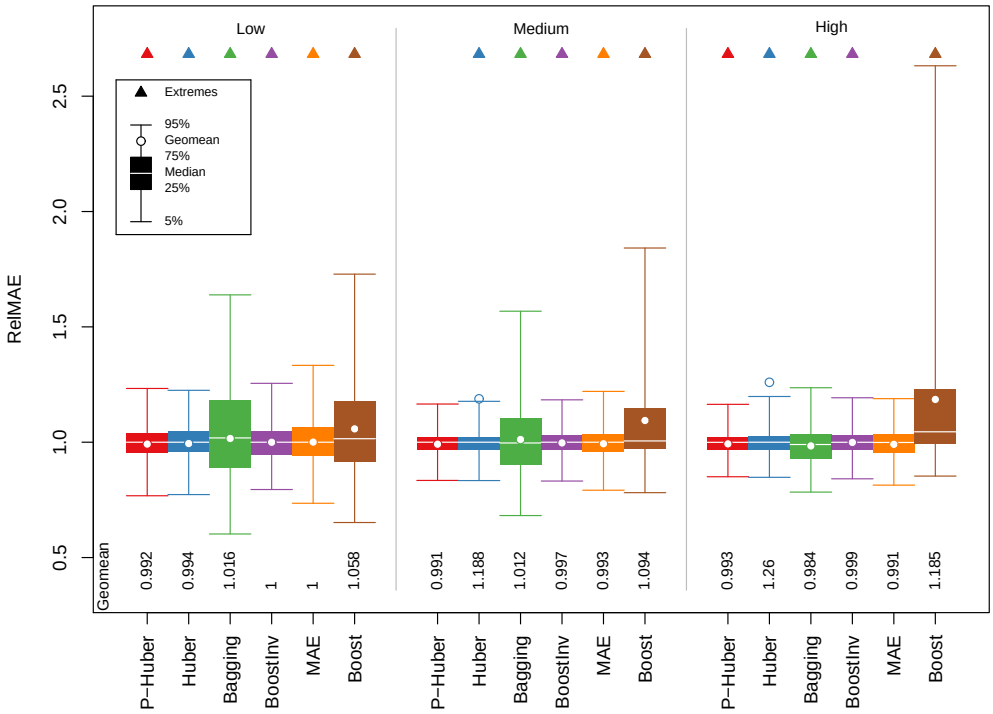


Figure 6: Snapshots of the distributions of RelMAE by time series noise. The 5%, 25%, 50%, 75% and 95% quantiles are provided, along with the geometric mean (geomean) for each method. Cases where the distribution has extreme errors beyond the scale of the vertical axis (higher than the 95% quantile) are indicated by a triangle.

Figure 6 provides similar snapshots of the RelMAE distributions, for the different methods, for different noise levels. To segment the time series we perform classical decomposition (Ord et al., 2017), and calculate the standard deviation of the irregular part (noise) of the time series, divided by the average of the trend component. This results in a robust coefficient of variation statistic. We rank all time series and classify the lowest third of all series as low noise, the second third as medium noise, and the remaining as high noise. Ignoring the distribution of Boost that performs poorly, we can see that distributions become tighter as the level of noise increases. This is expected as RelMAE is relative to the performance of Base, and the

maximum likelihood estimator performs very well for the relatively easy time series, with worsening performance for the harder ones. It is interesting to highlight that the distribution of Bagging improves substantially for the high noise time series, but still remains wider than the distributions of P-Huber, Huber, BoostInv and MAE. When exploring the results against the sample size of the time series, or the sampling frequency, we did not identify any clear patterns.

6. Discussion

Exponential smoothing has been one of the most widely used forecasting models, with evidence of good performance in numerous studies, forecasting competitions, and applications in the industry (Gardner Jr, 2006). Its ease of implementation, reliability, and transparency has made it ideal for large scale automatic forecasting. As such, it has attracted a lot of attention in research. One of the major innovations over the last years has been its reformulation within the single source of error state-space modelling framework (Hyndman et al., 2002), which provided the underlying statistical rationale for the model and therefore allowed the use of maximum likelihood estimation for its parameters, with various benefits such as resolving the selection of initial values, the generation of prediction intervals and automatic model selection using information criteria (Hyndman et al., 2008). We motivated this work by demonstrating that this state-of-the-art approach can lead to inconsistent forecasts, that can be a critical issue for large-scale automatic forecasting, but also weakening users' trust in the models (Dietvorst et al., 2015). To overcome this we investigated several robust estimators from the statistics and machine learning literature, many of which had not been previously tested. In terms of forecasting performance, we relied on accuracy and bias metrics (as captured by AvgRelMAE and AvgRelAME), however, given the ultimate objective of automatic large-scale forecasting, we also considered the stability of the solutions (as captured by the error distributions) and the implied computational cost. Here, we attempt to bring all these dimensions together to provide recommendations based on our results.

Overall, the M-estimators outperformed the LAD, bagging and boosting estimators. More specifically, The P-Huber provided the best combination of accuracy and bias (ranking first and second respectively) with well-behaved error distributions and minimal computational needs. Crucially, the improvements in both accuracy and bias are significantly different from the

benchmark maximum likelihood estimation. Although the average improvements are on the scale of a few percentage points over the benchmark, this can have substantial implications for practice, due to the application domain. For example, for a retailer, a few percentage points improvements in accuracy over the whole assortment of products can have very substantial inventory, financial and sustainability implications (Fildes et al., 2019; Ord et al., 2017). Beyond that, there are implications for the forecasting process. The increased consistency of forecasts can increase the trust of users in the model predictions and reduce interventions, that in many business forecasting context can reach up to 90% of the generated forecasts, adding the need for substantial corporate resources (Fildes et al., 2009; Ord et al., 2017). Considering the Huber estimator, it is generally outperformed by P-Huber and therefore further consideration is not needed. The results for the MAE estimator are noteworthy, as this estimator has been explored before in the literature (Gardner Jr, 2006). In terms of accuracy, it did not consistently outperform the benchmark, matching the results reported in the literature, but also explaining the statistically insignificant differences. However, in terms of bias it offered substantial improvements, ranking best across all methods considered. We argue that this is due to the effect of using absolute errors, where the estimator becomes optimal for the median of the target distribution (Gneiting, 2011). Although there are cases that this might be desirable, generally it is not expected to be better than the maximum likelihood estimator. Our results match this understanding and we argue that the slightly more complex M-estimators are more beneficial than MAE.

Bagging for exponential smoothing provided a machine learning based benchmark for our study. Although the results in the literature have demonstrated accuracy improvements (Bergmeir et al., 2016; Petropoulos et al., 2018), the effects on bias had not been considered before. We find that it performs significantly worse than the maximum likelihood estimator benchmark, which paired with its substantial computational requirements raises questions for its usefulness for the application setting we are considering. The results for boosting were also poor. On the other hand, BoostInv exhibited promising performance. Although it did not significantly outperform the benchmark and falls behind the leading P-Huber, it provides evidence that inverse boosting is not only applicable to time series forecasting but potentially more useful than boosting that has been successfully applied to other forecasting models (Barrow and Crone, 2016a). We also note that BoostInv is less computationally intensive than Bagging and it provided substantially

narrower error distributions. We argue that this is a useful finding, given the limited attention of inverse boosting in the literature and no prior evaluation for time series models.

Overall, we recommend that P-Huber, as adapted in this work for exponential smoothing, can have substantial benefits for practice, particularly for automatic large-scale forecasting applications, where reliability and consistency of forecasts are paramount.

Finally, the proposed methodology for tuning the M-estimators makes the approach scalable as an alternative to automatic forecasting with exponential smoothing. We use cross-validation to set the hyper-parameter q, so as to account for the characteristics of the time series at hand. Although this does add some computational cost, there is no need for an iterative algorithm, as is the case for Bagging, Boost or BoostInv, and therefore we argue that the gains they offer can out-weight the associated relatively small computational costs. This heuristic can potentially be beneficial in other areas that M-estimators are used, where the scaling of the errors and the setting of the threshold values remains a challenge. For our application, although we rely on cross-validation, due to the limited search space, we obtain results fast and therefore the M-estimators remain competitive to the maximum likelihood approach in terms of speed.

7. Conclusions

This paper investigated alternatives to the conventional maximum likelihood estimation to achieve robust parameter estimation for business forecasting. We explored this in the context of the exponential smoothing family of models, one of the most widely used forecasting approaches in practice. We looked at estimation methods inspired by research in statistics and machine learning. We found strong evidence that there are gains to be had in going beyond conventional estimation approaches.

Both M-estimators and machine learning approaches demonstrated gains. More specifically, using an absolute loss, which has been explored in the literature in the past for exponential smoothing, was shown to be a strong contender, but the somewhat more flexible Pseudo-Huber provided overall the best gains, in terms of accuracy and bias reduction. These came at a small increase in computation cost, providing a still usable improvement for practice, where the scale of the forecasting process, in terms of numbers of items to be forecasted, is a limitation that is often overlooked in research.

From the machine learning perspective, inverse boosting demonstrated better performance than conventional boosting and bagging. Although it did not match the overall performance of Pseudo-Huber, we argue that our findings suggest that additional work on inverse boosting for time series forecasting is needed, given that it significantly outperformed both boosting and bagging. This is in contrast to its moderate performance for classification tasks in the literature. Although all machine learning inspired approaches implied substantial computational cost, due to the iterative nature of the underlying algorithms, we do not advocate sidelining them. Instead, we call for more research on blended statistics and machine learning approaches, drawing on the benefits of both.

We argue that one of the contributions of this research is to bring together perspectives from both statistics and machine learning. Although in terms of the forecasting models our focus was narrow on exponential smoothing, a non-machine learning model, our viewpoint is that future research should be encouraged to look at these disciplines in conjunction. This has substantial implications for benchmarking of newly proposed approaches, some of which we discussed above, but also is of principal importance for the practicing analyst, who needs to solve a business challenge and is typically agnostic of whence the solution algorithm or model originates from. To best inform the user, we as researchers need to demonstrate the merits of the different approaches against the standard techniques in the arsenal of business forecasters, which nowadays includes both statistical and machine learning solutions.

Finally, considering the adoption of this research from practice, one has to consider the forecasting process that many organisations use, involving the generation of a baseline forecast supplemented by expert judgemental adjustments (Fildes et al., 2009; Ord et al., 2017). In such cases, robust predictions are desirable, as by definition any forecasting model will not capture all the elements of the underlying data generating process, a task that falls on experts using soft information available to the organisation. Therefore, as the forecast equation will have omitted terms, robust estimation and forecasting are critical. This makes the approaches investigated in this paper desirable. Given existing software infrastructure, switching to absolute loss is trivial. Similarly, using Pseudo-Huber remains relatively simple, in particular as the resulting model parameters can be seamlessly integrated into the existing process. This offers the advantage of being able to use prediction intervals that come naturally out of the forecasting models, which are useful

for translating forecasts into decisions. Machine learning inspired approaches are more involved in their implementation and since the generation of the final forecasts involves the combination of multiple forecasts, the generation of prediction intervals also becomes more complicated. Nonetheless, this comes at a great advantage: the user is forced to face the non-normality of forecast errors that is typically the reality (Trapero et al., 2019).

References

Al-Shemarry, M. S., Li, Y., Abdulla, S., 2018. Ensemble of adaboost cascades of 3L–LBPs classifiers for license plates detection with low quality images. Expert Systems with Applications 92, 216–235.

Athanasopoulos, G., Ahmed, R. A., Hyndman, R. J., 2009. Hierarchical forecasts for australian domestic tourism. International Journal of Forecasting 25 (1), 146–166.

Athanasopoulos, G., Song, H., Sun, J. A., 2018. Bagging in tourism demand modeling and forecasting. Journal of Travel Research 57 (1), 52–68.

Avnimelech, R., Intrator, N., 1999. Boosting regression estimators. Neural computation 11 (2), 499–520.

Baldauf, M., Silva, J. S., 2012. On the use of robust regression in econometrics. Economics Letters 114 (1), 124–127.

Barrow, D. K., 2012. Active model combination: an evaluation and extension of bagging and boosting for time series forecasting. Ph.D. thesis, Lancaster University.

Barrow, D. K., Crone, S. F., 2016a. A comparison of AdaBoost algorithms for time series forecast combination. International Journal of Forecasting 32 (4), 1103–1119.

Barrow, D. K., Crone, S. F., 2016b. Cross-validation aggregation for combining autoregressive neural network forecasts. International Journal of Forecasting 32 (4), 1120–1137.

Barrow, D. K., Kourentzes, N., 2016. Distributions of forecasting errors of forecast combinations: implications for inventory management. International Journal of Production Economics 177, 24–33.

Bauer, E., Kohavi, R., 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine learning 36 (1-2), 105–139.

Bergmeir, C., Hyndman, R. J., Benítez, J. M., 2016. Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation. International journal of forecasting 32 (2), 303–312.

Borra, S., Di Ciaccio, A., 2002. Improving nonparametric regression methods by bagging and boosting. Computational Statistics & Data Analysis 38 (4), 407–420.

Box, G. E., Cox, D. R., 1964. An analysis of transformations. Journal of the Royal Statistical Society. Series B (Methodological), 211–252.

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

Breiman, L., 1999. Prediction games and arcing algorithms. Neural computation 11 (7), 1493–1517.

Breiman, L., 2001. Using iterated bagging to debias regressions. Machine Learning 45 (3), 261–277.

Bühlmann, P., Yu, B., 2003. Boosting with the L2 loss: regression and classification. Journal of the American Statistical Association 98 (462), 324–339.

Burnham, K. P., Anderson, D. R., 2004. Multimodel inference: understanding AIC and BIC in model selection. Sociological methods & research 33 (2), 261–304.

Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M., 1994. Two deterministic half-quadratic regularization algorithms for computed imaging. In: Proceedings of 1st International Conference on Image Processing. Vol. 2. IEEE, pp. 168–172.

Charbonnier, P., Blanc-Féraud, L., Aubert, G., Barlaud, M., 1997. Deterministic edge-preserving regularization in computed imaging. IEEE Transactions on image processing 6 (2), 298–311.

Chatfield, C., 1995. Model uncertainty, data mining and statistical inference. Journal of the Royal Statistical Society. Series A (Statistics in Society), 419–466.

Chen, T., Ren, J., 2009. Bagging for gaussian process regression. Neurocomputing 72 (7-9), 1605–1610.

Cipra, T., 1992. Robust exponential smoothing. Journal of Forecasting 11 (1), 57–69.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., Terpenning, I., 1990. STL: A seasonal-trend decomposition procedure based on Loess. Journal of Official Statistics 6 (1), 3–33.

Cordeiro, C., Neves, M., 2009. Forecasting time series with boot. expos procedure. REVSTAT-Statistical Journal 7 (2), 135–149.

Crevits, R., Croux, C., 2016. Forecasting with robust exponential smoothing with damped trend and seasonal components. KU Leuven.

Dantas, T. M., Oliveira, F. L. C., 2018. Improving time series forecasting: An approach combining bootstrap aggregation, clusters and exponential smoothing. International Journal of Forecasting 34 (4), 748–761.

Davis, R. A., Wu, R., 2006. LAD estimation with applications in time series analysis. Encyclopedia of Environmetrics 3.

Davydenko, A., Fildes, R., 2013. Measuring forecasting accuracy: The case of judgmental adjustments to SKU-level demand forecasts. International Journal of Forecasting 29 (3), 510–522.

Dettling, M., Bühlmann, P., 2003. Boosting for tumor classification with gene expression data. Bioinformatics 19 (9), 1061–1069.

Dietterich, T. G., 2000. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine learning 40 (2), 139–157.

Dietvorst, B. J., Simmons, J. P., Massey, C., 2015. Algorithm aversion: People erroneously avoid algorithms after seeing them err. Journal of Experimental Psychology: General 144 (1), 114.

Drucker, H., 1997. Improving regressors using boosting techniques. In: ICML. Vol. 97. pp. 107–115.

Dutter, R., Huber, P. J., 1981. Numerical methods for the nonlinear robust regression problem. Journal of Statistical Computation and Simulation 13 (2), 79–113.

Fildes, R., Goodwin, P., Lawrence, M., Nikolopoulos, K., 2009. Effective forecasting and judgmental adjustments: an empirical evaluation and strategies for improvement in supply-chain planning. International journal of forecasting 25 (1), 3–23.

Fildes, R., Ma, S., Kolassa, S., 2019. Retail forecasting: Research and practice. International Journal of Forecasting.

Fildes, R., Petropoulos, F., 2015. Simple versus complex selection rules for forecasting many time series. Journal of Business Research 68 (8), 1692–1701.

Freund, Y., Schapire, R. E., 1997. A decision-theoretic generalization of online learning and an application to boosting. Journal of computer and system sciences 55 (1), 119–139.

Friedman, J., Hastie, T., Tibshirani, R., 2000. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics 28 (2), 337–407.

Friedman, J. H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189–1232.

Friedman, J. H., 2002. Stochastic gradient boosting. Computational statistics & data analysis 38 (4), 367–378.

Gao, J., Chen, C., Zhen, D., Zhu, Q., 2013. An efficient version of inverse boosting for classification. Transactions of the Institute of Measurement and Control 35 (2), 188–199.

Gardner Jr, E. S., 1985. Exponential smoothing: The state of the art. Journal of forecasting 4 (1), 1–28.

Gardner Jr, E. S., 2006. Exponential smoothing: The state of the art – Part II. International journal of forecasting 22 (4), 637–666.

Gelper, S., Fried, R., Croux, C., 2010. Robust forecasting with exponential and Holt–Winters smoothing. Journal of forecasting 29 (3), 285–300.

Gey, S., Poggi, J.-M., 2006. Boosting and instability for regression trees. Computational statistics & data analysis 50 (2), 533–550.

Gneiting, T., 2011. Making and evaluating point forecasts. Journal of the American Statistical Association 106 (494), 746–762.

Guerrero, V. M., 1993. Time-series analysis supported by power transformations. Journal of Forecasting 12 (1), 37–48.

Hollander, M., Wolfe, D. A., Chicken, E., 2013. Nonparametric statistical methods. Vol. 751. John Wiley & Sons.

Hothorn, T., Lausen, B., 2005. Bundling classifiers by bagging trees. Computational Statistics & Data Analysis 49 (4), 1068–1078.

Huber, P. J., 1981. Robust statistics. John Willey & Sons.

Huber, P. J., 1992. Robust estimation of a location parameter. In: Breakthroughs in statistics. Springer, pp. 492–518.

Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., Yasmeen, F., 2018. forecast: Forecasting functions for time series and linear models. R package version 8.4.
URL `http://pkg.robjhyndman.com/forecast`

Hyndman, R., Koehler, A. B., Ord, J. K., Snyder, R. D., 2008. Forecasting with exponential smoothing: the state space approach. Springer Science & Business Media.

Hyndman, R. J., Koehler, A. B., Snyder, R. D., Grose, S., 2002. A state space framework for automatic forecasting using exponential smoothing methods. International Journal of forecasting 18 (3), 439–454.

Inoue, A., Kilian, L., 2008. How useful is bagging in forecasting economic time series? a case study of US consumer price inflation. Journal of the American Statistical Association 103 (482), 511–522.

Israeli, A., Rokach, L., Shabtai, A., 2019. Constraint learning based gradient boosting trees. Expert Systems with Applications.

Johnston, F., Boylan, J., 1994. How far ahead can an EWMA model be extrapolated? Journal of the Operational Research Society 45 (6), 710–713.

Kelly, G. E., 1992. Robust regression estimators—the choice of tuning constants. Journal of the Royal Statistical Society: Series D (The Statistician) 41 (3), 303–314.

Khwaja, A., Zhang, X., Anpalagan, A., Venkatesh, B., 2017. Boosted neural networks for improved short-term electric load forecasting. Electric Power Systems Research 143, 431–437.

Koehler, A. B., Snyder, R. D., Ord, J. K., Beaumont, A., 2012. A study of outliers in the exponential smoothing approach to forecasting. International Journal of Forecasting 28 (2), 477–484.

Kourentzes, N., 2019. tsutils: Time Series Exploration, Modelling and Forecasting. R package version 0.9.0.
URL `https://CRAN.R-project.org/package=tsutils`

Kourentzes, N., Barrow, D., Petropoulos, F., 2019a. Another look at forecast selection and combination: evidence from forecast pooling. International Journal of Production Economics 209, 226–235.

Kourentzes, N., Petropoulos, F., 2016. Forecasting with multivariate temporal aggregation: The case of promotional modelling. International Journal of Production Economics 181, 145–153.

Kourentzes, N., Petropoulos, F., Trapero, J. R., 2014. Improving forecasting by estimating time series structural components across multiple frequencies. International Journal of Forecasting 30 (2), 291–302.

Kourentzes, N., Rostami-Tabar, B., Barrow, D. K., 2017. Demand forecasting by temporal aggregation: using optimal or multiple aggregation levels? Journal of Business Research 78, 1–9.

Kourentzes, N., Trapero, J. R., Barrow, D. K., 2019b. Optimising forecasting models for inventory planning. International Journal of Production Economics, 107597.

Krauss, C., Do, X. A., Huck, N., 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. European Journal of Operational Research 259 (2), 689–702.

Kuncheva, L. I., Whitaker, C. J., 2002. Using diversity with three variants of boosting: Aggressive, conservative, and inverse. In: International Workshop on Multiple Classifier Systems. Springer, pp. 81–90.

Kunsch, H. R., 1989. The jackknife and the bootstrap for general stationary observations. The annals of Statistics, 1217–1241.

Lee, C.-C., Chiang, Y.-C., Shih, C.-Y., Tsai, C.-L., 2009. Noisy time series prediction using M-estimator based robust radial basis function neural networks with growing and pruning techniques. Expert Systems with Applications 36 (3), 4717–4724.

Lemmens, A., Croux, C., 2006. Bagging and boosting classification trees to predict churn. Journal of Marketing Research 43 (2), 276–286.

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., Winkler, R., 1982. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. Journal of forecasting 1 (2), 111–153.

Makridakis, S., Hibon, M., 2000. The M3-competition: results, conclusions and implications. International journal of forecasting 16 (4), 451–476.

Makridakis, S., Wheelwright, S. C., Hyndman, R. J., 2008. Forecasting methods and applications. John wiley & sons.

Maronna, R. A., Martin, R. D., Yohai, V. J., Salibián-Barrera, M., 2018. Robust statistics: theory and methods (with R). Wiley.

Maronna, R. A., Yohai, V. J., 2000. Robust regression with both continuous and categorical predictors. Journal of Statistical Planning and Inference 89 (1-2), 197–214.

McDonald, J. B., Xu, Y., 1994. Some forecasting applications of partially adaptive estimators of arima models. Economics Letters 45 (2), 155–160.

Mittnik, S., Robinzonov, N., Spindler, M., 2015. Stock market volatility: Identifying major drivers and the nature of their impact. Journal of Banking & Finance 58, 1–14.

Ord, J. K., Fildes, R., Kourentzes, N., 2017. Principles of Business Forecasting, 2nd Edition. Wessex Press Publishing Co.

Owusu, E., Zhan, Y., Mao, Q. R., 2014. A neural-AdaBoost based facial expression recognition system. Expert Systems with Applications 41 (7), 3383–3390.

Persson, C., Bacher, P., Shiga, T., Madsen, H., 2017. Multi-site solar power forecasting using gradient boosted regression trees. Solar Energy 150, 423–436.

Petropoulos, F., Hyndman, R. J., Bergmeir, C., 2018. Exploring the sources of uncertainty: Why does bagging for time series forecasting work? European Journal of Operational Research 268 (2), 545–554.

R Core Team, 2018. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
URL `https://www.R-project.org/`

Sanders, N. R., Graman, G. A., 2009. Quantifying costs of forecast errors: A case study of the warehouse environment. Omega 37 (1), 116–125.

Schapire, R. E., 1990. The strength of weak learnability. Machine learning 5 (2), 197–227.

Schapire, R. E., Freund, Y., Bartlett, P., Lee, W. S., 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. Annals of statistics, 1651–1686.

Shrestha, D. L., Solomatine, D. P., 2006. Experiments with AdaBoost. RT, an improved boosting scheme for regression. Neural computation 18 (7), 1678–1710.

Shyu, W. M., Grosse, E., Cleveland, W. S., 2017. Local regression models. In: Statistical models in S. Routledge, pp. 309–376.

Skurichina, M., Duin, R. P., 1998. Bagging for linear classifiers. Pattern Recognition 31 (7), 909–930.

Sun, Y., Kamel, M. S., Wong, A. K., Wang, Y., 2007. Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition 40 (12), 3358–3378.

Taieb, S. B., Hyndman, R. J., 2014. A gradient boosting approach to the kaggle load forecasting competition. International journal of forecasting 30 (2), 382–394.

Trapero, J. R., Cardos, M., Kourentzes, N., 2019. Empirical safety stock estimation based on kernel and garch models. Omega 84, 199–211.

Zhang, Z., 1997. Parameter estimation techniques: A tutorial with application to conic fitting. Image and vision Computing 15 (1), 59–76.

Zheng, J., 2010. Cost-sensitive boosting neural networks for software defect prediction. Expert Systems with Applications 37 (6), 4537–4543.