# Estimators of Prediction Intervals for Statistical and Machine Learning Forecasts

Theodosiou Filotas

June 2020

**Abstract**

The need for accurate forecasts, to support decisions in companies, has resulted in the wide research of statistical and machine learning forecasting methods. As point forecasts are not 100% accurate, dealing with the uncertainties is fundamental to decrease the risk of costly, inaccurate decisions. Prediction intervals express these uncertainties associated with each forecast. Most of the work regarding prediction interval estimation is focused on methods heavily relying on strong assumptions like independent, identical and normally distributed errors. However, these strong assumptions produce unrealistic intervals and increase the risk of stock-outs. As a result, such methods should be used with caution. This work offers an extensive review of the existing literature regarding prediction intervals estimation, along with a comparison of the widely used methods, applied on a large number of real-time series. Guidelines for the usage of each method, applied on both statistical and machine learning models, are given. Moreover, by exploring the correlation between point forecast errors and prediction intervals, the results suggest that selecting the best possible model in terms of point forecast accuracy, should be a priority, as is a key determinant of the quality of the prediction intervals.

## 1 Introduction

In recent years, the rapid increase of available historical data, along with the rise in computational resources, have created an explosion of interest for accurate time-series forecasts. This is particularly true for Demand Forecasting, where past sales data are used for estimations about future demand (Armstrong 2017). It helps decision-makers within enterprises in estimating how many of their products will be sold, so their inventory is planned accordingly. As a result and by considering consequences such as poor services and economic loss, that could immediately affect companies due to inaccurate forecasts, a variety of techniques, which include widely used statistical models, along with novel machine learning approaches, are constantly reviewed and adjusted.

Forecasting models are typically deployed to produce point forecasts. Point forecasts capture the expected demand, reflected in a single number. This gives

a false sense of certainty to users. Instead, one has to consider the uncertainty of a forecast, more aptly captured by the predictive distribution. Measuring the uncertainty, as a product of multiple sources of errors like the data or the model used, along with its parameters, or the actual context of the forecasting domain, around these forecasts is strongly correlated with the confidence decision-makers show on the model's estimations and thus, their decisions. Users can plan different strategies to deal with the different outcomes that might be produced due to future uncertainty resulting in no false certainty of point forecasts.(Chatfield 1998),

Prediction intervals that accompany point forecasts are used to capture these uncertainties, as they provide an upper and lower limit, where the unknown future value is expected to lie in between, with a specific probability (Chatfield 1996a).

As far as demand forecasting is concerned, in translating forecasts into actions, an unexpected rise or fall in future demand might lead to a stockout or additional expenses due to overstocked inventory (Trapero et al. 2018*a*). To account for these issues, companies use safety stocks, which stand for extra products to mitigate such risks, and their calculation is directly connected with measuring forecast uncertainty and thus, prediction intervals. For example, to stock the appropriate amount of inventory a retailer will need to satisfy 95 out of the 100 customers that enter his store, the 95% upper prediction interval is necessary.

Despite the wide extent of the literature regarding point forecasts, methods for estimating prediction intervals are not as widely explored. Most commonly used methods, such as theoretical and simulation-based ones (Hyndman et al. 2008, Hyndman & Athanasopoulos 2018) heavily rely on unrealistic assumptions, such as independent, identically (i.i.d), and normally distributed errors. However, in real world applications, these assumptions are not viable, and thus, using such methods can result in a higher risk of a stockout (Trapero et al. 2018*a*). Another alternative that overlaps the normality assumption, which is shown to be unrealistic (Trapero et al. 2018*b*), is bootstrapping-based methods.

A family of methods for measuring prediction intervals, where both the i.i.d. and normality assumptions are relaxed, are empirical methods (Kourentzes & Athanasopoulos 2020*a*, Isengildina et al. 2006, Trapero et al. 2018*a*). Despite their promising results and their development in areas such as prediction intervals theory and financial risk management, they are not as widely applied on safety stock calculations (Trapero et al. 2018*a*).

This work aims to present an extensive review of the existing methods on computing prediction intervals for statistical models (SM), along with a presentation of the advantages and limitations of each method. The performance of the different methods is assessed using mean interval score, an interval evaluation metric, on different time series of various horizons. We also discuss the implementation of the various methods described, along with the connection between errors of point forecasts and prediction intervals

In addition, due of the rapid development of the machine learning (ML) field in the past years, several ML models have been explored and developed

for forecasting purposes. As these models mainly focus on prediction tasks, they typically lack the analytical expressions used by the statistical model for estimating the variance of the forecast. As this is one of the major differences between SM and ML models. Thus, most methods used by SM for the estimation of prediction intervals are not applicable on ML models.

Besides exploring the various methods for the estimation of prediction intervals, this paper also aims to show if and how some methods used for statistical models could be modified to be transferred and used by machine learning models. Finally, a comparison of the performance of such methods on statistical and machine learning models, for prediction interval forecasting is presented.

The rest of the paper is organized as follows: Section 2 provides an introduction to the existing literature on estimating prediction intervals. Section 3 describes how the experiment is designed in terms of models, data, and evaluation methods used. Section 4 explains the different methods implemented on both the statistical and machine learning model. Final results are given in Section 5 with a discussion on the findings taking place in Section 6. Lastly, Section 7 presents the concluding remarks.

# 2 Related Work

In general, for most Statistical Models, prediction intervals(PIs) are estimated using,

$$[\hat{y}_{t+h|t} - c\hat{\sigma_h}, \hat{y}_{t+h|t} + c\hat{\sigma_h}], \tag{1}$$

where $\hat{y}_{t+h|t}$ is the point forecast for the forecasted horizon h, c is the desired coverage probability and $\hat{\sigma_h}$ is the conditional variance of the point forecast (Hyndman et al. 2008). The main problem with this approach is the estimation of $\hat{\sigma_h}$. For some models, theoretical formulas exist for either calculating or estimating $\hat{\sigma_h}$. For example Hyndman et al. (2008), shows that for an Exponential Smoothing ETS(A,N,N) model, forecast variance $\hat{\sigma_h}^2$ is given by,

$$\hat{\sigma_h}^2 = \sigma^2[1 + \alpha^2(h-1)], \tag{2}$$

where $\alpha$ is a parameter of the ETS(A,N,N) model and $\sigma$ is the is the residual variance

However, this is not true for every model. On top of that, because of the complexity of some formulas, especially for models with multiplicative errors, results are estimations and not actual values. This is mainly an issue for bigger values of h, where estimations start to greatly diverge from the actual value (Hyndman et al. 2008).

Another simple way to approximate $\sigma(h)$ is from,

$$\hat{\sigma_h} = \sqrt{h}\hat{\sigma_1}, \tag{3}$$

where $\hat{\sigma_1}$ is the one-step ahead standard deviation of the forecast's error and can be estimated as follows:

$$\hat{\sigma_1} = \sqrt{MSE_{t+1}} \tag{4}$$

For every new observation the MSE can be updated through

$$MSE_{t+1} = \alpha' {e_t}^2 + (1 - \alpha')MSE_t, \tag{5}$$

where $e_t$ is the difference between the forecasted and the actual value. However, this method has been heavily criticized Chatfield (2000).

When the approximation of sigma through a theoretical formula is not feasible, simulation-based methods (Hyndman et al. 2008, Hyndman & Athanasopoulos 2018) could be used. For the selected model, M future paths, with M being a large integer (in practice, M is set to 5000) could be simulated resulting in the distribution of the simulated point forecasts. For each simulated path, a random error $\epsilon_t$ is randomly picked. The error term can be sampled either from a normal distribution with 0 mean and $\sigma^2$ the variance of the residuals or with replacement, from the in-sample pool of error (ie bootstrap-based methods). Prediction intervals are then extracted from the distribution of forecasts.

Finally, another alternative, which is not as widely explored, is empirical-methods (Trapero et al. 2018*a*, Isengildina et al. 2006, Chatfield 2000). In simple terms, the desired intervals are extracted from the in-sample distribution of forecasting errors. A limitation of this approach is that in-sample errors are usually lower than the equivalent out-of-sample ones (Barrow & Kourentzes 2016). To overcome this limitation an extra unseen validation set is used to estimate the distribution of errors (Williams & Goodman 1971). From the distribution of in-sample or validation-set errors, the desired lower and upper quantile is extracted and is summed up and down on the point forecast. All the aforementioned methods are furtherly detailed in the following sections.

# 3 Designing the Experiment

## 3.1 Models Used

### 3.1.1 State Space Models

From the Statistical State Space Models, Exponential Smoothing(ETS) is selected, due to its popularity as a statistical method for demand forecasting. (Willemain et al. 2004). Its popularity is based on its simple implementation and its relatively good overall accuracy (Trapero et al. 2018*b*). Exponential smoothing methods use weighted averages of previous observations to produce forecasts (Hyndman & Athanasopoulos 2018). The weights of the observations decay as the observations become older. Consequently, the most recent observations have higher weights. In other words, in its simplest form, the forecast of an ETS at time $T + 1$ is the weighted average given by:

$$\hat{y}_{T+1|T} = \sum_{j=0}^{T-1} a(1-a)^j y_{T-j} + (1-a)^T l_0, \tag{6}$$

where $0 \leq a \leq 1$ is the smoothing parameter and $l_0$ is the first fitted value at time 1, which is initially estimated. As described before, for large T the last term $(1-a)^T l_0$ converges to zero and does not affect the forecast at time $T+1$.

Each ETS model consists of a measurement equation, which describes the observed data, and some state equations that describe how the unobserved components or states (level, trend, seasonality) change over time (Hyndman et al. 2008).

The formulas which describe these state space equations, depend on the selection of the model's parameter combination. For example an ETS with Multiplicative Errors, Seasonality and an Additive trend, or in short, an ETS(M,A,M) model, is consisted by the following equations:

$$y_t = (l_{t-1} + b_{t-1})s_{t-m}(1 + \epsilon_t)$$

$$l_t = (l_{t-1} + b_{t-1})(1 + \alpha\epsilon_t)$$

$$b_t = b_{t-1} + \beta(l_{t-1} + b_{t-1}\epsilon_t)$$

$$s_t = s_{t-m}(1 + \gamma\epsilon_t),$$

where $y_t$ is the measurement equation and the rest are the state space ones. Smoothing parameters $\alpha$, $\beta$,$\gamma$, and $\sigma$ along with the initial value $l_0, b_0$ and $s_0$, are optimized during the training phase by minimizing the sum of squared errors or maximizing the likelihood. The error process $\epsilon_t$ stands for the unpredictable part of each state-space equation and its probability distribution needs to be initially defined. It is usually assumed that $\epsilon_t$ is i.i.d and normally distributed with zero mean and variance $\sigma^2$. It is added, depending on the model's parameters, through multiplication or addition.

For each time series used, the optimal structure(Trend, Seasonality, and type of Errors) is automatically picked by considering the Akaike's Information Criterion(AICc)(Hyndman et al. 2008, Hyndman & Athanasopoulos 2018). The AICc is defined as :

$$AICc = AIC + \frac{2k(k+1)}{T-k-1}, \tag{7}$$

where k the total number of parameters and initial states, and AIC is Akaike's Information Criterion, given by :

$$AIC = -2log(L) + 2k, \tag{8}$$

with L being the maximized likelihood of the model (Hyndman & Athanasopoulos 2018).

It is important to highlight that some combinations can lead to numerical problems on the estimations of the prediction intervals. In particular, for non-linear models with multiplicative errors, the theoretical formulas return an approximation of the PI, while for models with the multiplicative trend or a mixture of additive errors and multiplicative seasonality(or vice versa) no theoretical formulas do exist (Hyndman et al. 2008).

### 3.1.2 Machine Learning Models

From the Machine Learning Model's family, XGBoost (Extreme Gradient Boosting) is selected. Over the year it has been dominating applied machine learning competitions, by using far fewer computational resources than other models (Chen & Guestrin 2016). It is a decision-tree-based ensemble algorithm that uses a gradient-boosting framework. XGboost idea is based on combining many weaker models, to build a stronger one by iteratively adding trees on top of each other, with the errors of the previous tree being corrected by the next one. This concept is called boosting (Mishina et al. 2014).

The main difference between XGboost and other ensemble-tree-based models is that every new model is fitted to the residuals of the previous model and then minimizes the new loss (ie gradient boosting) (Chen & Guestrin 2016). What is more, XGBoost uses a depth-first approach, which greatly accelerates the computational performance, while it sequentially builds each tree by using a parallelized implementation. (Morde 2019)

Despite all the advantages stated above, successfully fitting and optimizing XGBoost for forecasting is not as simple compared to a state-space model. First, models from the two families require data in different formats to produce forecasts.

Table 1: Input Data for the Two Models

| | ETS | XGBboost | | | | |
|---|---|---|---|---|---|---|
| | y | y | $lag_1$ | $lag_2$ | $lag_3$ | $lag_4$ |
| 1 | 119 | 119 | - | - | - | - |
| 2 | 104 | 104 | 119 | - | - | - |
| 3 | 118 | 118 | 104 | 119 | - | - |
| 4 | 115 | 115 | 118 | 104 | 119 | - |
| 5 | 126 | 126 | 115 | 118 | 104 | 119 |
| 6 | 141 | 141 | 126 | 115 | 118 | 104 |
| 7 | 135 | 135 | 141 | 126 | 115 | 118 |
| 8 | 125 | 125 | 135 | 141 | 126 | 115 |

Table 1, First 8 values of a quarterly time series, in the appropriate format for fitting an ETS and a XGBoost model

While for ETS the whole time series is given as input, with y being the actual value, for XGBoost past values need to be converted to tabular features, as described in Table 1. In other words, for every value $y_t$, we construct lags $lag_1 = y_{t-1}, lag_2 = y_{t-2}, ..., lag_n = y_{t-n}$, where k is the frequency of the given series. It is also important to highlight that the initial n values are removed as they do not have a complete set of past observations. As with all Machine Learning Algorithms, an appropriate feature selection is necessary, as not all features are equally important and a wrong selection might affect the performance of the model.

For a given time series, Autocorrelation Coefficients, measure the relationship between lagged values (Hyndman & Athanasopoulos 2018, Arranz 2005) . For each lag, there are several autocorrelation coefficients $r_k$ which measure the relationship between $y_t$ and $y_{t-k}$. A value $r_k$ is given by,

$$r_k = \frac{\sum_{t=k+1}^{T}(y_t - \hat{y})(y_{t-k} - \hat{y})}{\sum_{t=1}^{T}(y_t - \hat{y})^2}, \tag{9}$$

where T is the length of the time series.

A limitation of the Autocorrelation Coefficients is that if $y_t$ and $y_{t-1}$ are correlated, in addition to $y_{t-1}$ and $y_{t-2}$, then naturally, $y_t$ and $y_{t-2}$ will be too. However, this might not be the case, as this correlation might be a result of the formulas used to get the calculation. For this reason, Partial Autocorrelations are picked. These measure the relationship between $y_t$ and $y_{t-k}$ while removing these effects from past lags $1, 2, ...k-1$ (Hyndman & Athanasopoulos 2018).

For every time series used, Partial Autocorrelations Coefficients are calculated and the Partial Autocorrelations Function(PACF) is estimated. PACF shows which lags have the most significant correlation with the actual values $y_t$ (Hyndman & Athanasopoulos 2018)

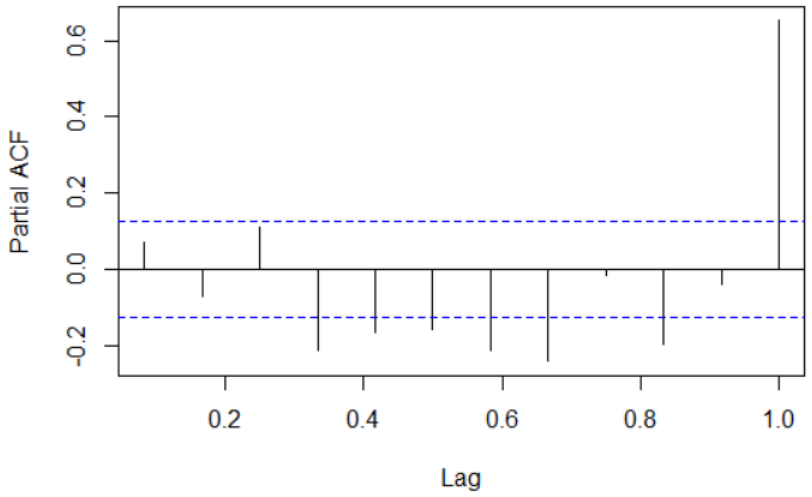


Figure 1: PACF, for the 12 lags of a monthly Time Series

Lags with Coefficients over 0.1 or lower than -0.1 are automatically selected as the features for each model fitted on a particular time series. A limitation of PACF and ACF, in general, is that they are both linear methods, while XGBoost is not. According to literature, in practice, this is not a major issue and both methods could also be applied to non-linear models (Crone & Kourentzes 2007).

Another challenge of fitting an XGBoost model for forecasting is "forcing" the model to "learn" both the trend and the seasonality of the series while understanding the variations of the seasonality per year. This is challenging as due to the relatively small amount of training observations, adding excessive complexity can lead to overfitting. To overcome this issue, time series is transformed into stationary and seasonally stationary by taking the appropriate number of differences and seasonal differences. In other words, as series are converted into stationary, trend and seasonality do not have a strong effect on observations at the different times (Hyndman & Athanasopoulos 2018). In particular, for every value $X_t$,the following transformations are applied:

$$X'_t = X_t - X_{t-1}(1)$$
$$X''_t = X'_t - X'_{t-h}(2),$$

where h is the frequency of the given time series.

It is important to highlight that these differences are not required for every time series. Additionally, on some occasions, the transformed data might not be stationary after one transformation. Thus, the same transformations might be applied more than one time. As proposed by Hyndman & Athanasopoulos (2018), the Augmented Dickey-Fuller (Mushtaq 2011) test is considered to determine the appropriate number of differences for each time series. After forecasting, the predictions are reversed back into their original form.

Finally to optimize the performance of XGBoost, successfully tuning its hyperparameters is fundamental. An automatic Random Search on the generated hyperparameter's grid, validated by a 5-fold cross-validation, is implemented for every fitted model on each given time series (Probst et al. 2018). The optimal hyperparameter combination is selected in terms of Mean Squared Error(MSE).

## 3.2 Evaluation Method

To assess the PIs produced by the different methods, Interval score is used as an evaluation metric (Kourentzes & Athanasopoulos 2020*b*, Gneiting & Raftery 2007). With interval score, a narrow interval is rewarded, while a penalty relative to parameter a is given, if the true value, is not included on the produced interval (Gneiting & Raftery 2007). Interval Score(IS) is given by,

$$(u-l) + \frac{2}{a}(l-x)ID(x,l) + \frac{2}{a}(x-u)ID(u,x), \tag{10}$$

where x is the true value, u and l stand for the upper and lower interval. ID(a,b) returns 1 if $a > b$, or in other words, when the true value lies outside the forecasted interval, and 0 otherwise.

Data is split into a training and a test set, with the test set including the final two periods of each time series. For example, for monthly time series, two full years are kept as a test set. Initially, each model is fitted into the training set and the intervals for the next period, $t+1, t+2, ...t+h$ are forecasted and evaluated against the first h values of the test-set.

In the next step, the model will be re-fitted on an updated training set, which will also include the first observation from the previously defined test set. A new h-steps-ahead forecast will take place and will be evaluated against an updated test set. The new forecast is equivalent to values $t+2$ to $t+(h+1)$ of the test set. This procedure is repeated until a final forecast for $t+h$ to $t+2h$ is performed (Ord et al. 2017).

Figure 2 describes this exact procedure. Blue dots represent the training set for each iteration, and red dots representing the last value of each test-set.

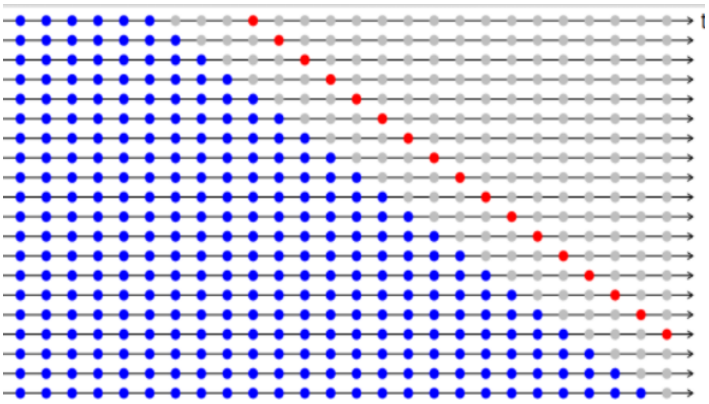

Figure 2: Evaluation on rolling 4-steps-ahead forecasting origin, (Hyndman & Athanasopoulos 2018)

This method is known as evaluation on a rolling forecasting origin with re-estimation. It provides a more reliable assessment of each method's performance, as the evaluation does not depend entirely on a single h-step-ahead forecast which might include outliers and irregularities (Ord et al. 2017).

It is important to highlight that naturally the different time series, have by definition different scales and units. This is reflected in the interval scores produced for each time series. Consequently, the results are not comparable. To overcome this issue, Geometric Mean Relative Absolute Error(GMRAE) (Davydenko & Fildes 2013) is used as it produces non-scale depending errors.

GMRAE takes the Geometric Mean of the absolute values of an evaluation metric produced for each method, divided by the score of a method picked as a benchmark. In this work, the modified Geometric Mean Relative Absolute Interval Score(GMRAIS) is used. The theoretical method applied to ETS is picked as a benchmark. Let IS be the interval score and N the total number of origins:

$$GMRAIS = \sqrt[N]{\prod^{N} \left| \frac{IS_A}{IS_{Bench}} \right|}, \tag{11}$$

where $IS_{Bench}$ is the score of the theoretical method applied on the ETS and $IS_A$ the score of every other method. A Relative error lower than 1 indicates

that method A outperformed the benchmark one, while scores bigger than 1 show that method A produced better results.

### 3.3 Data Used

Applying every method on a single set of time series, would not give trustworthy results, as some methods might be biased to the properties of the single-time series selected. As a result, different time series are selected and the results are aggregated. In total, 76 different monthly time series are picked. Each contains a total of 240 observations. Moreover, to test how each method performs on different horizons, 89 quarterly time series are also selected, in addition to 88 weekly ones. The total length of each quarterly time series is 36 observations, while for weekly ones, 173 total values are contained.

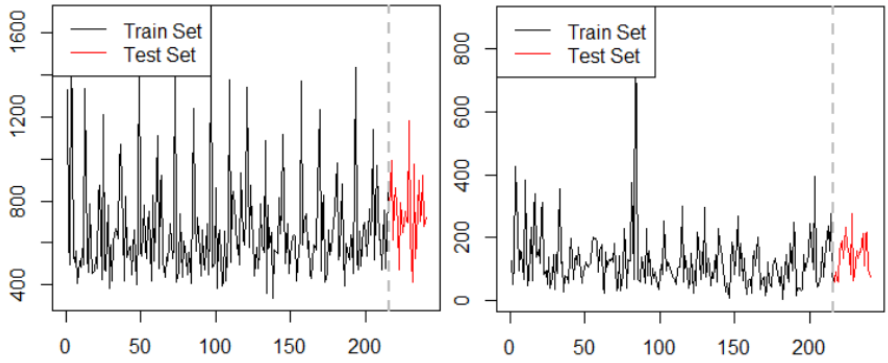


Figure 3: Example of two Monthly Time Series

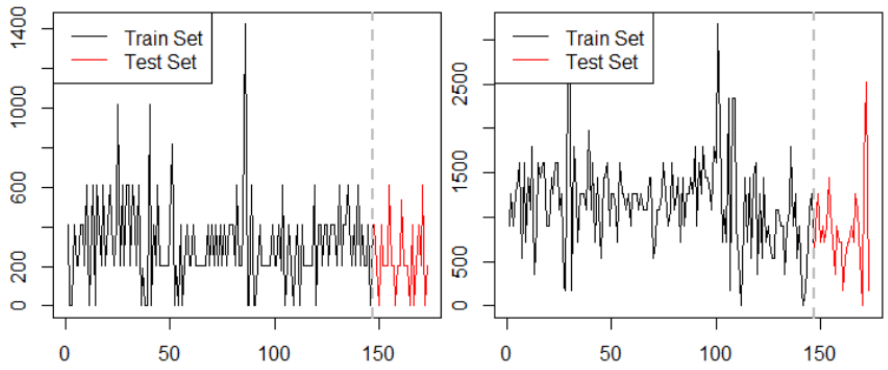


Figure 4: Example of two Monthly Weekly Series

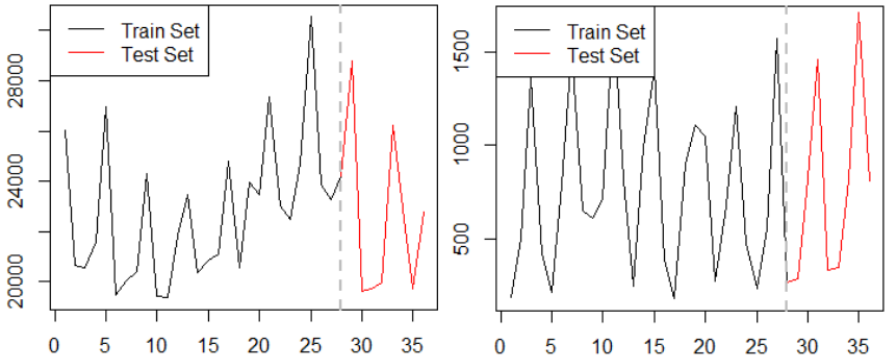


Figure 5: Example of two Monthly Quarterly Series

As presented in Figures 3, 4 and 5 above by considering the different frequencies of each set of time series, an even train-test split is not possible. To produce comparable results, for every different set of frequencies, the last two periods of each time series are kept as a test set. Results are compared per forecasting horizon on each set of time series. Finally, the Mean Interval Score (MIS) per horizon, for each method on each of the three different sets is also calculated.‘

# 4 Methods

## 4.1 Algebric-Theoretical

The first method used as a benchmark is the Algebraic. It has been briefly introduced in Section 2 and is based on Equation (1). As mentioned, the method is solely applicable for the ETS model and is based on several realistic (and unrealistic) assumptions.

First of all, one has to assume that there is no bias in the measurement process, and the true values of the process are reflected on the data used to produce the intervals. In other words, the expected value of each error term is zero(Ord et al. 2017). Then:

$$\mu_{t+h|t} = E(y_{t+h}|\mathbf{x_t}), \tag{12}$$

where $\mu_{t+h|t}$ is the mean of the distribution of the future values of the series on a given time. It is known as point forecast. In addition, the corresponding forecast variance is given by,

$$\upsilon_{t+h|t} = V(y_{t+h}|\mathbf{x_t}), \tag{13}$$

By making the aforementioned statistical assumption, the state vector of the last period of the observation $\mathbf{x_t}$, is known, and hence, the prediction distribution can be produced (Hyndman et al. 2008).

Secondly, it should be assumed that there is no correlation between errors, neither on the same time step nor across all horizons (Ord et al. 2017). If errors were related, then the model used for forecasting would not be optimal. The formula for estimating the variance of the model's forecast is given by:

$$Var\left(\sum_{i=1}^{h} \hat{y}_{t+i}\right) = \hat{\sigma}_{t+1}^2 + \hat{\sigma}_{t+2}^2 + cov(\hat{\sigma}_{t+1}, \hat{\sigma}_{t+2}), \tag{14}$$

Because of difficulties in estimating $\hat{\sigma}_{t+2}^2$ and $cov(\hat{\sigma}_{t+1}, \hat{\sigma}_{t+2})$, assuming the errors are uncorrelated means that the covariance between the errors along with $\hat{\sigma}_{t+2}^2$, have no effect and can be excluded from the variance's formula.

Furthermore, the variance of the errors has to be assumed to be constant and conditional to horizon h. Violating this assumption might have relatively little impact on the point forecast, but it would critically affect the construction of PIs. If errors were to increase or decrease over time, the generated PIs for these future times would become either too narrow or too wide (Ord et al. 2017).

By making these assumptions, the errors are identically and independently distributed, with zero means and a constant variance. The fourth fundamental assumption is that the errors are normally distributed. Despite this being an unrealistic assumption in most real-world applications, it is fundamental for applying the theoretical formulas to calculate $\sigma$.

These four assumptions could be summarized by the notation: $\epsilon_t \sim IIN(0, \sigma^2)$ or by stating that errors are identical, independently, and normally distributed with zero means and a constant variance. Lastly, assuming that there is no bias in picking the model's parameters and hence no extra uncertainty exists, is also necessary.

Under these assumptions, forecast mean and variance could be calculated for every ETS model, using formulas given by Hyndman et al. (2008). Then by using equation (1) the desired intervals can be forecasted. However, assuming reality away to overcome difficulties in estimations is not recommended for real-world applications.

## 4.2 Simulation-Based Methods

Another simple approach to get the desired intervals that has no restrictions on the model used for the estimations is simulation-based methods. One simply simulates M future paths and for every horizon h, extracts the desired intervals from the generated distribution of predictions (Hyndman et al. 2008).

Recalling from Section 3.1, the general ETS model has the form,

$$y_t = w_(\boldsymbol{x}_{t-1}) + r(\boldsymbol{x}_{t-1})\epsilon_t,$$
$$\boldsymbol{x_t} = \boldsymbol{f}(\boldsymbol{x}_{t-1}) + \boldsymbol{g}(\boldsymbol{x}_{t-1})\epsilon_t,$$

where $w(X)$ and $r(X)$ are scalar functions and $\boldsymbol{f}(X)$, $\boldsymbol{g}(X)$ are vector functions. $\epsilon_t$ is the uncertainty factor of the process with variance $\sigma^2$. For every $t = n+1, ..., n+h$, by generating the ${y_t}^{(1)}, {y_t}^{(2)}, ..., {y_t}^{(M)}$ set of predictions, the prediction distribution could be estimated (Hyndman et al. 2008, Ord et al. 1995).

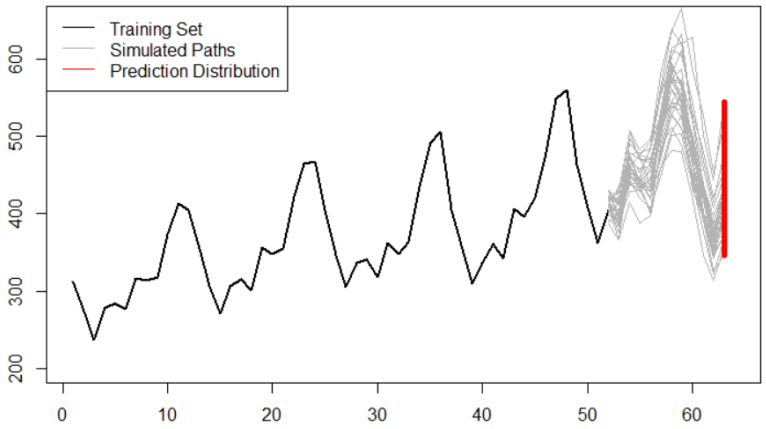


Figure 6: 30 Simulated future paths of the AirPassengers Time Series, generated using an ETS model. The solid red line represents the Prediction Interval for a 12-steps-ahead forecast

To implement this method, one has to assume that the future errors will follow a behavior similar to historical ones and that there is no bias in selecting the model's parameters. As with the theoretical approach, the errors are also assumed to be identically, independently and normally distributed (Hyndman et al. 2008).

For each $t = n+1, ..., n+h$ and for every $i = 1, 2, ...M$, an $\epsilon_t$ value is picked from the assumed Gaussian distribution, using a random number generator and is used to estimate the distribution of the ${y_t}^{(i)}$ predictions.

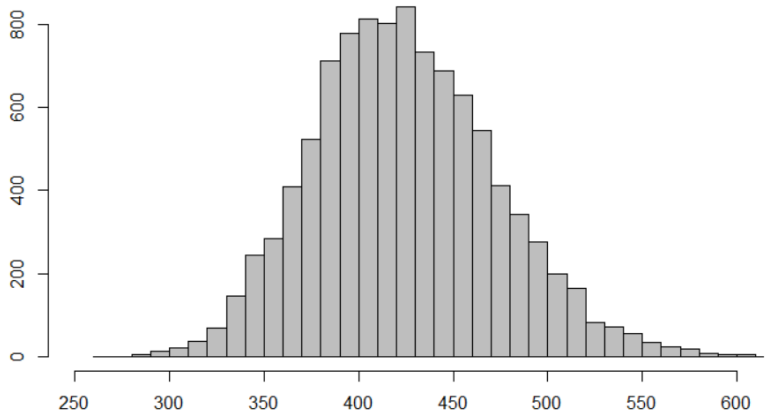


Figure 7: The distribution of Simulations for the last month of the monthly AirPassengers Time Series

#### 4.2.1 Direct Method

Hyndman et al. (2008) suggests that from the generated distribution, the $100(1-a)\%$ PI for the forecast horizon h, could be directly extracted by approximating the $\frac{a}{2}$ and the $1-\frac{a}{2}$ quantiles of the simulated $\boldsymbol{y}_{t+h|t} = \left\{{y_{t+h}}^{(1)}, {y_{t+h}}^{(2)}, ..., {y_{t+h}}^{(M)}\right\}$ set. Then we simply pick the values which lie between the 5% and the 95% observation. For example, for the hypothetical set of predictions, $\boldsymbol{y}_{t+h|t} = \{1, 2, 3, ...100\}$ the 95% PI is [5,95].

In this work, this method is refered as the "Direct Method". In general, it is one of the most commonly used methods. Nevertheless, more methods to estimate the desired prediction interval exist and they should be considered. In this work, the following methods are additionally considered.

#### 4.2.2 Mean-Sigma method

On the "Mean-Sigma" method, for every forecast horizon h, the mean forecast, standing for the mean value of the distribution, is estimated. Afterward, the spread of errors, in terms of their standard deviation around the mean, is multiplied by the conditional probability c. In other words the PI is estimated with the following formula:

$$PI(c)_2 = [\mu \pm \sigma c] = [\mu - \sigma c, \mu + \sigma z] \tag{15}$$

#### 4.2.3 Mean-Direct method

The third approach is similar to the first. Instead of extracting the quantiles directly for the distribution of predictions, they are extracted from the errors

approximated around the distribution's mean. Then they are summed with the mean forecast as follows:

$$PI(c)_3 = [\mu + l, \mu + u], \tag{16}$$

where l is the desired lower quantile and u the equivalent upper one.

#### 4.2.4 Mean-KDE method

Lastly, in the fourth method, the full prediction density is estimated with a kernel density estimator(KDE) applied on the set of errors around the mean forecast (Silverman 1986). The probability kernel density function is given by,

$$f(x) = \frac{1}{Nh} \sum_{j=1}^{N} K(\frac{x - X_j}{h}), \tag{17}$$

with N the sample size, K the kernel smoothing function, and h the bandwidth. From the estimated distribution the lower and upper quantile are approximated and are summed (the lower interval is subtracted) to the mean forecast as described in the methods above.

After applying the defined methods on the distribution of predictions shown in Figure 7 the following intervals are produced:

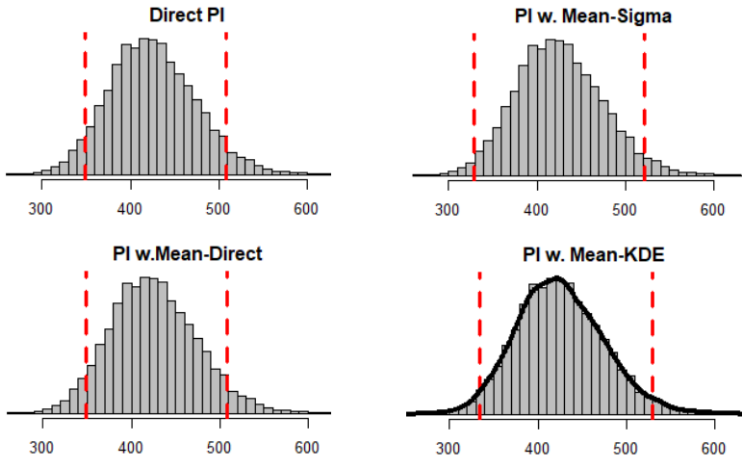


Figure 8: Intervals extracted from the distribution of predictions on Figure 4

To give an early indication regarding the accuracy of the PIs generated by each method, 12-step-ahead prediction intervals, are calculated using rolling origins for the monthly AirPassengers series. The results are given in Figure 9.

To conclude on the simulation-based methods, the usage of the different methods requires several underlying assumptions. This might lead to problems in some real-world applications. As shown, different methods applied to the

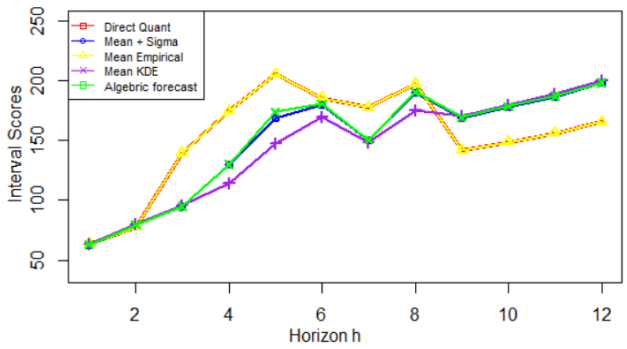


Figure 9: CrossValidated ISs of the 4 methods for 12-steps-ahead forecasted PIs

simulated distribution, produce different intervals and a standardized approach should not be followed. Depending on the sample size and the nature of the given series, more than one method should be considered.

### 4.3 Boostrap-Based Methods

Boostrap-Based Methods refer to a family of methods based on an approach equivalent to simulation-based methods. By simulating different future paths, the distribution of predictions is estimated (Hyndman et al. 2008). Then, from the generated distribution, the desired intervals are extracted using techniques described in the previous section.

An important distinction with the simulation-based methods is that there is no need for any assumptions regarding the distribution of the errors. The only fundamental underlying assumptions needed are summarized by $\epsilon_t \sim IID$. In other words, errors are assumed to be identically and independently distributed. As with simulation-based methods, the set of predictions $\boldsymbol{y}_{t+h|t} = \left\{{y_{t+h}}^{(1)}, {y_{t+h}}^{(2)}, ..., {y_{t+h}}^{(M)}\right\}$ is generated for every $t = n+1, ..., n+h$.

For every ${y_{t+h}}^{(i)}$, instead of randomly sampling an error term from a Gaussian Distribution, errors are picked through re-sampling, from the in-sample distribution of historical errors (Hyndman et al. 2008). Consequently, assuming that future errors follow a behavior similar to past errors and that the process of picking sample $M_i$ is uncorrelated to picking sample $M_{i-1}$ are necessary. Compared to previous methods, these are more relaxed assumptions.

As with simulation-based methods, from the simulated distribution of predictions, more than one approach could be applied to extract the desired prediction intervals. In this work, the same 4 methods described in Section 4.2 are considered.

## 4.4 Empirical-Methods

Finally, the last set family of methods is Empirical methods (Trapero et al. 2018*a*, Isengildina et al. 2006, Chatfield 2000). These methods are not as popular as the aforementioned. They are based on the hypothesis that PIs can be produced by estimating historical errors. In contrast with the previous methods, empirical methods follow a different methodology for the estimation of predictions intervals.

To begin with, the main assumption necessary is that future errors follow approximately the same distribution as historical ones. Smith & Sincich (1988) showed that this is a reasonable and feasible assumption. In that regard, one major advantage of empirical methods over the rest is that they are not based on assumptions far away from reality as they can be applied to any type of distribution. However, in order for accurate PIs to be estimated an adequate sample size is necessary (Isengildina et al. 2006).

Initially, for every time period, the in-sample errors (ie the difference between the prediction of the fitted model on the training set and the actual values) are calculated and their distribution is generated. It should be pointed out that in-sample errors are usually lower than test-set errors (Barrow & Kourentzes 2016). This might have an effect on the performance of empirical methods. A possible workaround is using a validation set that has not been used for fitting the model. In this work, due to the relatively small size of the time series used, no validation set is picked. From the distribution of in-sample errors, two non-parametric methods are considered to extract the upper and lower quantile.

### 4.4.1 Direct Empirical method

The first method is the direct method as described in the previous sections. It directly extracts the values which lie between the 5% and the 95% observations.

### 4.4.2 KDE Empirical method

The second method suggests calculating the probability density function on the set of errors using KDE. The formula for KDE is given in Equation 17. As proposed by Trapero et al. (2018*a*) and Kourentzes & Athanasopoulos (2020*a*), the Silverman's bandwidth and Epanechnikov kernel are picked. When the quantiles are extracted, the fitted model produces a point forecast for the forecast horizon h. Finally, the intervals are given by:

$$PI = [MeanForecast + LowInterval, MeanForecast + UpperInterval] \quad (18)$$

## 4.5 Methods For XGBoost

Concerning XGBoost models, no theoretical formulas exist for the estimation of PIs. In addition, as presented in Figure 10, simulation-based methods produce extremely narrow intervals. This is not unexpected as machine learning models have no state-space factor. In other words, the forecasts variance is smaller.

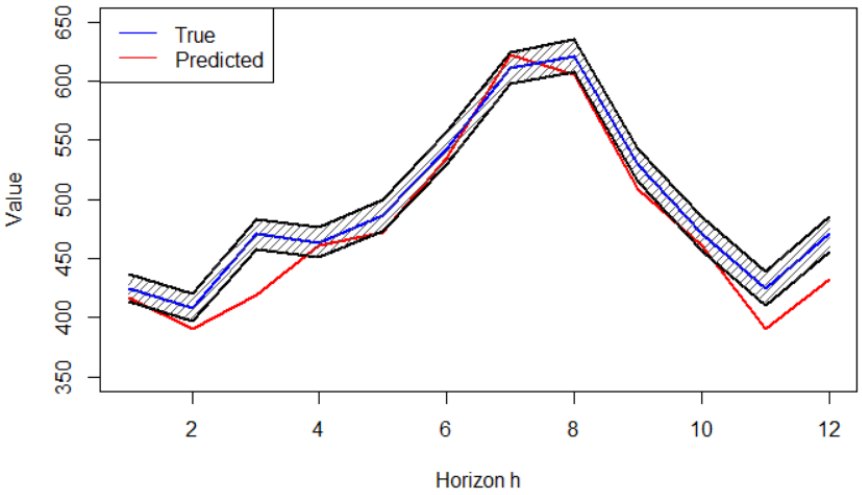


Figure 10: Prediction Interval for the last year of the AirPassenger's Time Series using XGBoost with Mean-KDE Simulations methods

Thus, the only family of methods applied for XGBoost is Empirical methods as defined in Section 4.4. From the in-sample distribution of errors, the upper and lower quantiles are extracted using the Direct and the KDE method. These are then added to the point forecast produced for the selected forecasting horizon

# 5 Results

In this section, the results from the various methods, applied to the ETS models, are presented. As the three sets contain time series of different frequencies, methods are initially compared for every set of monthly, quarterly, and weekly time series, so aggregated results per forecast horizon h, can be presented. Then the interval scores of the Empirical methods applied on the optimized XGBoost models are presented and discussed. Finally, the results of the two families of models are compared.

## 5.1 Results on Statistical Models

In the work, only the 95% PIs are estimated. In some future work, methods will be compared on more PIs. Figure 11 shows the results for the monthly time series. A total of 76 time series are explored and the GMRAIS for every method and every horizon h, is calculated. For every time series, the mean values per forecast horizon h is given.

As it can be observed bootstrap with the direct extraction of the PI produces the best scores across all methods. On the other hand, empirical methods

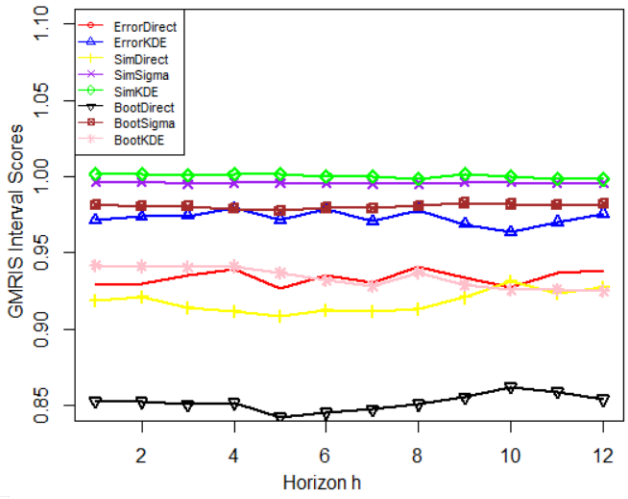


Figure 11: Mean GMRAIS per forecast horizon for 76 Monthly Time Series

outperformed the algebraic one and had similar scores with the simulation-direct method.

As shown in Table 2, by comparing the scores between BootstrapDirect and DirectEmpirical, it can be pointed out that there is no significant difference between the two. A possible explanation for their small difference is that on empirical methods the in-sample errors of some series are much smaller than the equivalent out-sample. This would not be an issue if more observations were present. Another workaround is the usage of a validation set to extract the quantiles.

| **Horizon** | **BootDirect** | **ErrorDirect** |
|---|---|---|
| 1 | 0.852 | 0.928 |
| 2 | 0.852 | 0.93 |
| 3 | 0.85 | 0.935 |
| 4 | 0.84 | 0.938 |
| 5 | 0.845 | 0.926 |
| 6 | 0.847 | 0.935 |
| 7 | 0.85 | 0.93 |
| 8 | 0.85 | 0.941 |
| 9 | 0.861 | 0.933 |
| 10 | 0.858 | 0.927 |
| 11 | 0.853 | 0.936 |
| 12 | 0.851 | 0.938 |

Table 2, GMRAE Comparisson between BootDirect and ErrorDirect

Figure 12, shows the equivalent results for the 88 weekly time series. The

performance of the methods is equivalent with the monthly set. Bootstrap and simulation methods with the direct extraction of the PI produces the best overall results. The Empirical Direct method does not fall far behind. What is important to highlight on both weekly and monthly data is that KDE methods are outperformed.

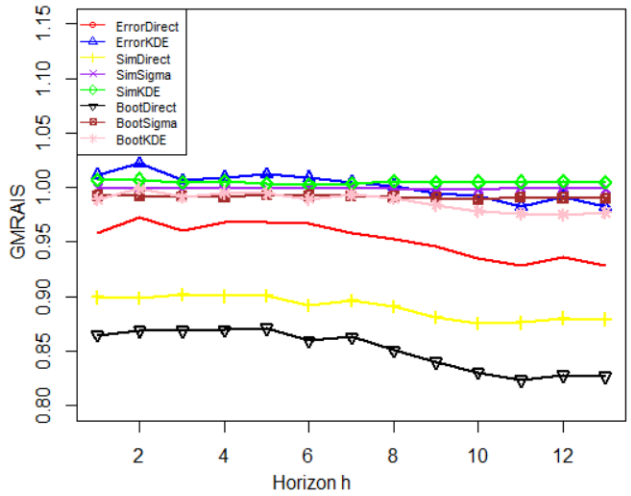


Figure 12: Mean GMRAE per forecast horizon for 88 Weekly Time Series

Finally, quarterly time series present a different pattern. The results are given in Figure 13. Theoretical methods outperform empirical ones in addition to BootKDE. Simulation methods with direct extraction outperform every other approach. An explanation for these differences is the smaller sample size of the quarterly series( ie 204 fewer observations than monthly time-series and 137 fewer than weekly ones). For such a smaller sample size, simulation-based methods seem to produce the best results, while the theoretical method outperforms some of the approaches.

In general, theoretical and simulation-based methods, relying on strong underlying assumptions do not produce the best results and their wide use is questioned. A Bootstrap method that overlaps the normality assumption produces the best PIs on the two sets of time series. However, it should be noted that bootstrap methods require the i.i.d assumption and their implementation might lead to an increase in the stock-out risk. As a result, and despite their promising results, bootstrap methods should be used with caution.

For time series with fewer observations, methods relying on the normality assumption outperformed empirical ones. Despite these findings, such methods should be avoided, as the normality assumption is not close to reality. Instead, Bootstrap methods, that overlap the are a promising alternative.

On monthly and weekly series, empirical methods based on realistic assumptions do not produce significantly worse results than the rest of the methods.

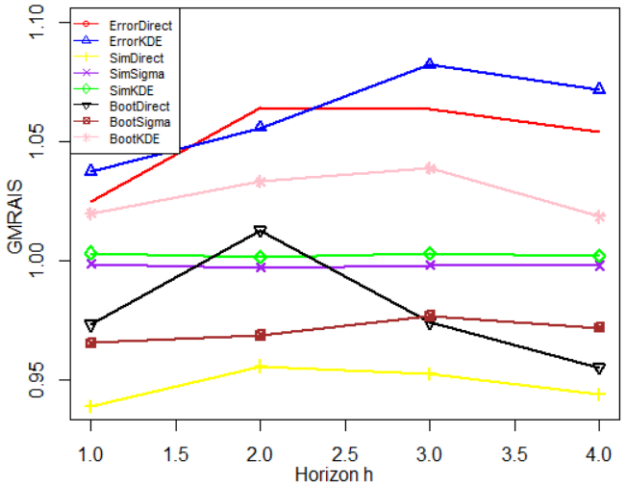


Figure 13: Mean GMRAIS per forecast horizon for 89 quarterly Time Series

In general, by considering the nature of the interval score as an evaluation function, slightly bigger scores are results of slightly wider intervals. When it comes to real-world application, where issues such as stock-out are important, having marginally bigger PIs could be preferred over unrealistically tighter intervals produced by rest of the methods.

Another advantage of empirical methods over simulation and bootstrap ones is the computational expense. To generate the prediction distribution, simulations/bootstrap methods require M (which for purposes of this work, is set to 10.000) forecasts for every rolling-origin on cross-validation. Thus, they require twice the time of empirical methods. This might be an issue on relatively bigger time series.

A limitation of Empirical methods, as pointed out by Isengildina et al. (2006) and as validated by the results on the quarterly set of series, is that empirical methods require a bigger sample size in comparison with the rest of the methods. To further optimize the performance of such methods, especially on time series with more observations, exploring errors from a validation set would produce tighter intervals and the methods would achieve better overall scores.

## 5.2 XGBoost Results

As mentioned in Section 4, when applying simulation-based methods with machine learning models, the intervals produced are narrow and hence these methods are not recommended. Machine learning models do not have the state-space component of ETS. In addition, since theoretical formulas for the estimation of the desired PI do not exist, the family of empirical methods is the only one that can be "transferred". The results of the methods on the different sets are given in Figure 14.

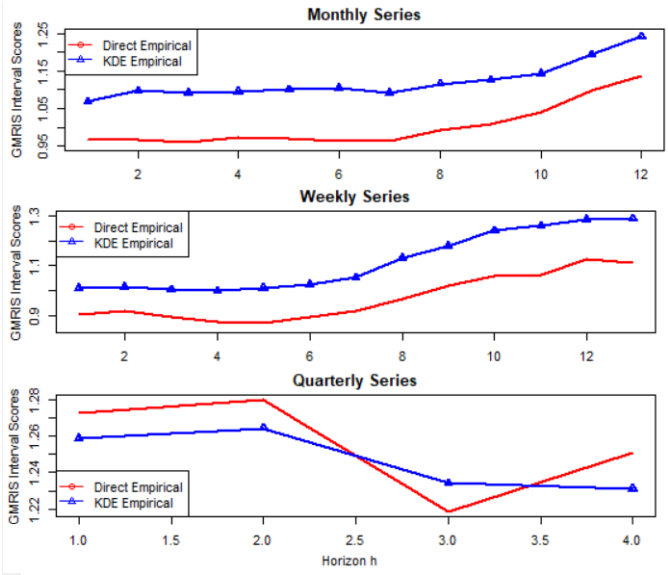


Figure 14: Mean GMRAIS per forecast horizon for the different set of TimeSeries

As far as monthly and weekly series are concerned, despite XGBoost producing good results with the Direct extraction on the earlier horizons, for the later months, it gets heavily outperformed. This is expected as the further ahead one is trying to forecast, the more uncertain it gets Hyndman & Athanasopoulos (2018).

On the other hand, on quarterly time series, both methods are heavily outperformed by the benchmark approach. Their poor performance can be explained by the relatively small size of the training set. A total of 28 observations is not an adequate sample size for a machine learning model and thus, inaccurate intervals are generated.

A possible explanation for XGboost's poor performance is that the automatic procedure used for feature selection and hyperparameters tuning might have failed for some time series. A manual optimization might have produced better results, but due to time limitations, this was not feasible. An example of the performance of a manually fitted XGBoost model on a single time series is presented in Figure 15. XGBoost outperforms the optimal ETS model on most horizons.

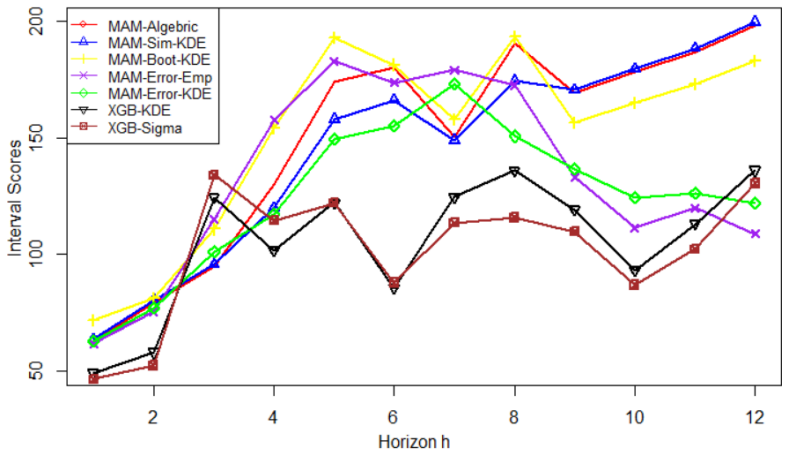


Figure 15: Interval scores comparison of a manually optimized XGBoost model and an ETS on the AirPassengers time series

## 5.3 Models Comparisson

In general, empirical methods produce promising results. In Table 3 the results of the methods applied on both models are presented. In addition, the point forecast MASE metric (Hyndman & Koehler 2006) is also given. Different explanations could be derived for the various set of time series.

On quarterly data, despite XGBoost having a smaller MASE than ETS the relatively smaller data size is the reason empirical methods do not properly work. As it stands out, the effect of data size is more clear on XGBoost than ETS as the model has more difficulties in fitting on the given series

For monthly and weekly series, and as discussed in the following section, a higher point forecast error contributes to the higher interval score of XGBoost over ETS. For the weekly time series, both the smaller data size and a bigger MASE are responsible for XGBoost not producing the best results. As mentioned, another possible explanation might be the automatic procedure used to fit the XGBoost model.

All in all, for the monthly time series, empirical methods produce the best results given the bigger sample size. Generally, empirical methods do not try to assume reality away, to produce good-enough results. By considering their performance, their relative limited research and usage are questioned. For an accurate estimation of PIs on real-world applications, when an adequate sample size is present, empirical methods are recommended.

Table 2: Mean Results

| | Quarterly | | Monthly | | Weekly | |
|---|---|---|---|---|---|---|
| Method | ETS | XGBoost | ETS | XGBoost | ETS | XGBoost |
| SimDirect | 0.947 | NA | 0.91 | NA | 0.89 | NA |
| SimMeanSigma | 0.998 | NA | 0.995 | NA | 0.998 | NA |
| SimMeanDirect | 0.947 | NA | 0.91 | NA | 0.89 | NA |
| SimMeanKDE | 1 | NA | 1 | NA | 1.004 | NA |
| BootDirect | 0.978 | NA | 0.85 | NA | 0.85 | NA |
| BootMeanSigma | 0.97 | NA | 0.98 | NA | 0.991 | NA |
| BootMeanDirect | 0.978 | NA | 0.85 | NA | 0.85 | NA |
| BootMeanKDE | 1.027 | NA | 0.933 | NA | 0.987 | NA |
| EmpDirect | 1.05 | 1.25 | 0.933 | 1 | 0.952 | 0.97 |
| EmpMeanKDE | 1.06 | 1.24 | 0.972 | 1.122 | 1.001 | 1.15 |
| **MASE** | **0.908** | **0.83** | **0.614** | **0.695** | **0.995** | **1.013** |

Table 3, Mean Intervals Scores per horizon for each method along with the point forecast MASE error. MASE was picked as suggested by Kourentzes & Athanasopoulos (2020*b*), for being a non-scale dependand evaluation metric

# 6 Discussion

## 6.1 Direct Methods and KDE estimation

For every family of methods and every model, the direct extraction of the interval produced better results than exploring the estimated density distribution function with a density estimator. As given in Figure 16, the Direct method, gets rid of the extreme observations on the two tails of the distribution, while KDE tries to smoothly include all the distributed values.

As it can be pointed out, the density function tries to smoothly include all the observations of the distribution. Due to some asymmetries, some gaps in the distribution, it fails to do so and as a result, wider intervals are produced. As it can be concluded and in accordance with Isengildina et al. (2006) a bigger error sample would result in a better-fitted density function, as no such "gaps" would be present. KDE would then fit the distribution more accurately and according to the literature's suggestions, it would outperform the Direct Empirical method.

When this is not the case, the Direct Extraction of the PI is a very promising alternative for smaller samples. To sum up, using different techniques to extract the PI from either the prediction or the error distribution is highly recommended. More than one method should be considered as no method works universally better.

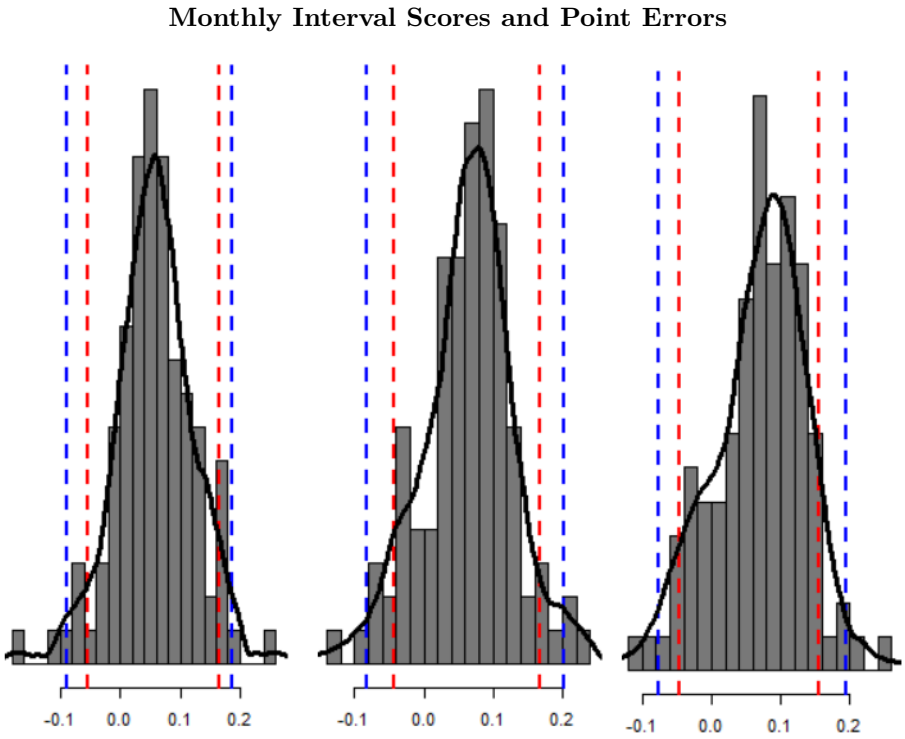


Figure 16: Distribution of Errors for three random horizons. Blue lines are intervals extracted from Kernel Density Estimator while red are from the direc t method

## 6.2 Correlation Between Intervals and Point Forecasts

Most of the PI estimation methods used in this work have a tight bond with the point forecast accuracy of each model. Despite using different approaches to extract the desired quantiles, summing them with the point forecast is the final step of each method. With this in mind, the correlation between the accuracy of the point forecast, in terms of absolute error, and interval score has been explored.

For each set of monthly, quarterly, and weekly time series, the correlation between the cross-validated interval scores and the equivalent absolute error has been explored. All methods produce a similar pattern and thus, there was no need to individually explore the behavior of each method. In addition, results produced by both XGBoost and ETS also follow a similar behavior and thus, are not presented individually.

There is a direct correlation between interval score and absolute error. The best intervals with the lowest IS are the ones that have the lowest point forecast error. This is natural as if the predicted interval does not include the true value

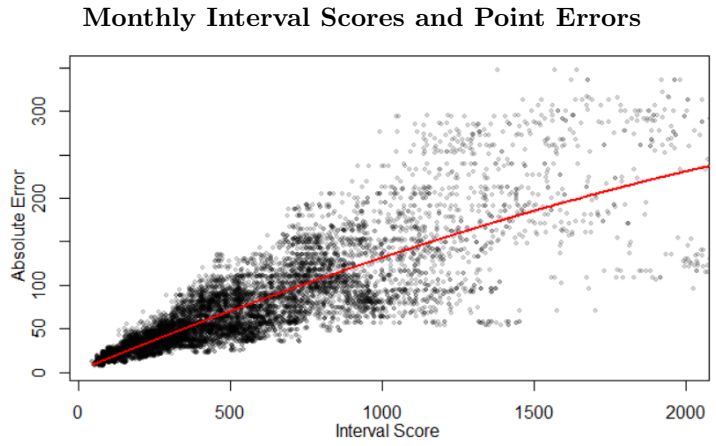


Figure 17: Correlation Between Interval Scores and Absolute Error.l

the IS is in turn higher. However, as shown in Figure 17, forecasts with a high absolute error, produce bad intervals. As a result, having a good point forecast is fundamental in producing accurate intervals.

For quarterly time series with a fewer number of observations, the results are given in Figure 18. We observe bigger interval scores and absolute errors compared to monthly data. The best intervals, with an IS less than 250 have low absolute error. This confirms our initial hypothesis. For an optimal interval to be produced a good point forecast is necessary. Furthermore, as IS rises, so does the absolute error. The rate is even faster than on monthly data. Lastly, for weekly time series, no big differences are present. However, the correlation seems to be more clear than the other two sets.

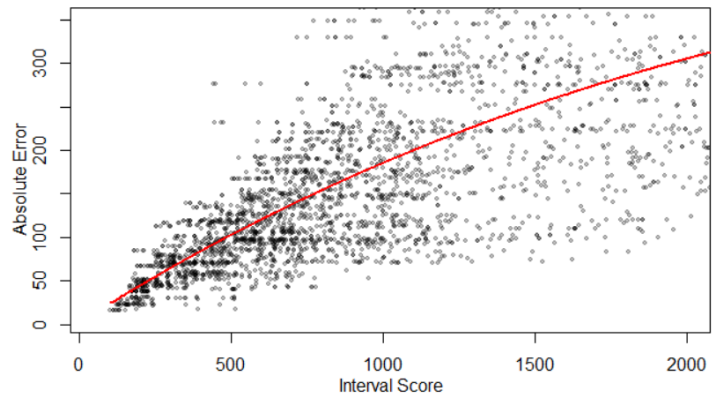


Figure 18: Correlation Between IS and Absolute Error on Quarterly Time Series

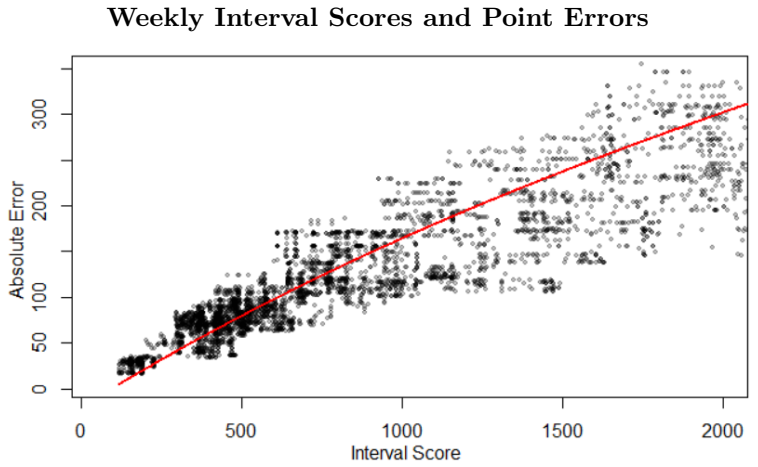


Figure 19: Correlation Between IS and Absolute Error on Weekly Series

In summary, there is some correlation between interval scores and absolute errors. The results suggest that for an accurate interval to be produced, a model producing an adequate point forecast should be used. Applying the recommended methods with a misspecified model would result in extra bias that translated into extra future uncertainty.

In some future work, the correlation between point and interval forecasting would be furtherly explored. In this work, absolute error and interval score are the only forecasting metrics considered. As both of these methods are scale-dependent, using different combinations of metrics, including non-scale dependent, would result in a better interpretation of the results. From my point of view, using different metrics would result in minor changes in the correlations.

Figures 20 and 21 give an example for added uncertainty due to using misspecified models. The intervals produced by two non-optimal (in terms of AICc) ETS models are compared with the intervals forecasted with the best (again,in terms of AICc) model. For this comparison, GMRAIS is used as defined in Section 3.2. The errors of the best candidate model are used as the benchmark for the GMRAIS formula. In other words, a GMRAIS lower than 1, indicates that a method applied on a non-optimal model outperforms the equivalent method on the benchmark-optimal one.

As the results indicate, no method on any model, for any forecast horizon, outperforms its equivalent applied on the optimal model. In addition, as given in Table 4, the three models produce significantly different mean interval scores.

It is important to highlight that on, the ETS(M,A,A) (ie the second best model), the empirical methods seem to perform evenly well as with the ETS(M,A,M). They also outperform every other method. This is not true for the ETS(A,A,A) as every method and mostly the empirical ones are underachieving. A possible explanation is the usage of additive errors instead of multiplicative.

In general, picking a misspecified model for a given time series would not

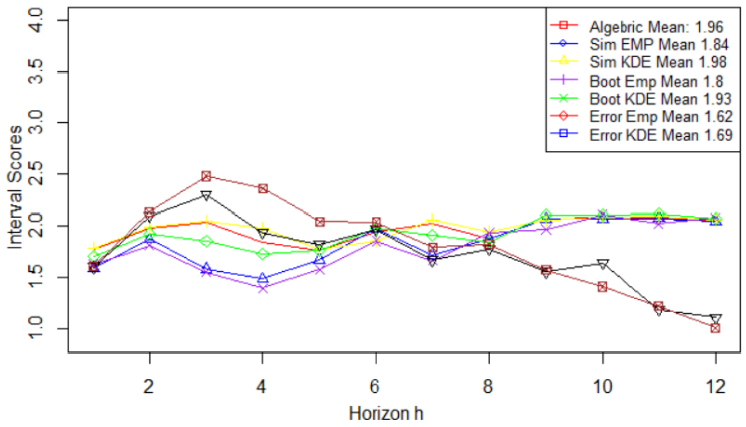


Figure 20: GMRAE Comparisson between optimal ETS(M,A,M) and non-optimal ETS(M,A,A) model

Table 3: Mean Results Per Horizon

| Method | ETS(M,A,A) | ETS(A,A,A) | ETS(M,A,M) |
|---|---|---|---|
| Algebric | 281.54 | **357.23** | 149.7 |
| SimDirect | 246.81 | 445.01 | 153.85 |
| SimMeanKDE | 280.87 | 359.66 | 146.86 |
| BootDirect | 246.73 | 492.46 | 168.1 |
| BootMeanKDE | 267.93 | 399.09 | 153.13 |
| EmpDirect | 201.83 | 494.38 | 132.67 |
| EmpMeanKDE | **200.75** | 452.83 | **124.63** |
| **Mean** | **246.63** | **428.66** | **146.99** |

Table 4, Mean Interval Scores Per Horizon, of the Two Non-Optimal Models and Optimal ETS(M,A,M)

result in the most accurate point forecast. In turn, as point forecast accuracy is shown to be correlated with the estimation of the PI , the forecasted prediction interval will not be optimal. Possibly, even regardless of the method used for its prediction. To sum things up, a proper model selection in terms of optimal point forecasting is highly recommended and it should be a priority before selecting the appropriate method.

# 7 Conclusion

Despite the wide usage of theoretical and simulation-based methods, they both require a number of underlying assumptions. Some of these are unrealistic and their misuse in real-world applications might increase the risk of a stock-out. This work offers a wide review of the methods available for the estimation of

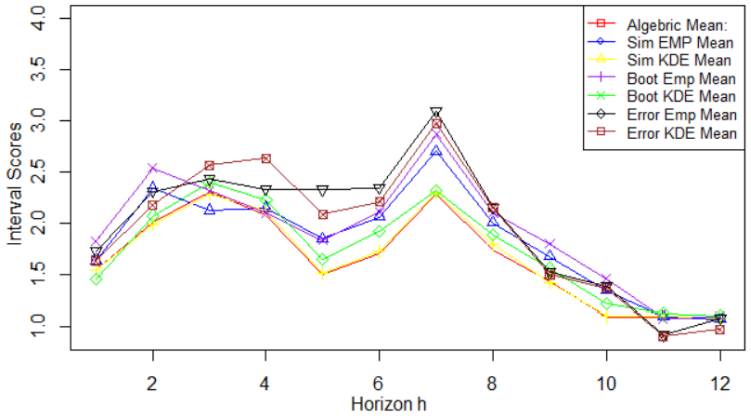


Figure 21: GMRAE Comparisson between optimal ETS(M,A,M) and non-optimal ETS(A,A,A) model

prediction intervals and offers guidelines for their implementation. PIs produced by each method on a significant amount of time series with different properties are evaluated and their performance is discussed.

This work shows that empirical methods based on more relaxed assumptions than the approaches preferred in literature do not produce vastly different results. For this reason, in specific scenarios, they are highly recommended. Producing slightly worse interval scores than bootstrap methods can be translated as forecasting wider (and in some cases more realistic) intervals. Bootstrap methods produced better and hence, tighter intervals, However, they require assuming i.i.d errors, which on real-world demand forecasting, might not be reasonable. Making decisions based on intervals produced by methods requiring unrealistic assumptions may result in an over or under-stocking a company's inventory

Results on the weekly time series show that empirical methods underperform when smaller data samples are present. This is a limitation of such approaches. Under these circumstances, bootstrap methods that overlap the highly unrealistic assumption of normally distributed errors, are a promising alternative. However, as mentioned, they should be used with caution.

Another important insight is that a standardized approach to extract the desired quantiles from an estimated distribution should be avoided. As it has been discussed, the direct extraction of the quantiles gave the best results. Nonetheless, for bigger sample sizes, using a kernel density estimator would be a better choice. In short, more than one method should be considered before producing the final forecast.

Regarding the performance of machine learning models, empirical methods produced promising results. Despite being outperformed by the same methods applied on an ETS model, they can easily be transferred and used for the estimation of prediction intervals for machine learning models. Explanations for their

less accurate performance include the relatively small size on quarterly series and the extra bias added due to the automatic procedure used for optimizing and fitting the models.

The rest of the methods are not applicable either because there are no formulas for the estimation of forecast's variance or because of the small prediction variance on simulation and bootstrap-based methods originated from the lack of the state-space component. In summary, empirical methods are recommended for the estimation of prediction intervals when it comes to machine learning models. However, their performance and their limitations should be furtherly researched. On some further work, empirical methods applied on models of the Neural Networks family will be explored and their performance will be compared with the rest of the models.

Another aspect of estimating prediction intervals that have been investigated is their correlation with the goodness of point forecasts. The results suggest that for an accurate PI to be produced, picking a model with high point forecast accuracy should be a priority.

This work focused on the practical implementation of the various methods on non-domain specific time series. Another feature that is tightly connected with decision-making within companies and should be considered when estimating prediction intervals, is the cost of errors bonded with forecasting PIs. Backordering as a result of errors in inventory management can have costly consequences for businesses. Measuring the correlation between the economic loss of companies and inaccurate prediction intervals should be a priority

# References


Armstrong, J. (2017), 'Demand forecasting ii: Evidence-based methods and checklists', p. 36.

Arranz, M. (2005), 'Tol-project portmanteau test statistics in time series'.

Barrow, G. & Kourentzes, N. (2016), 'Distributions of forecasting errors of forecast combinations: Implications for inventory management', *International Journal of Production Economics* **177**, 24–33.

Chatfield, C. (1996a), *The Analysis of Time Series*, 5th edn, Chapman and Hall/CRC.

Chatfield, C. (1998), 'Prediction intervals, department of mathematical sciences'.

Chatfield, C. (2000), *Time-Series Forecasting*, 1st edn, Chapman and Hall/CRC.

Chen, T. & Guestrin, C. (2016), Xgboost: A scalable tree boosting system, ACM, pp. 785,794.

Crone, S. & Kourentzes, N. (2007), Input variable selection for time series prediction with neural networks- an evaluation of visual, autocorrelation and spectral analysis for varying seasonality.

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

Gneiting, T. & Raftery, A. E. (2007), 'Strictly proper scoring rules, prediction, and estimation', *Journal of the American Statistical Association* **102**, 359–378.

Hyndman, R. J. & Athanasopoulos, G. (2018), *Forecasting: principles and practice*, 2nd edn, OTexts.

Hyndman, R. J. & Koehler, A. B. (2006), 'Another look at measures of forecast accuracy', *International Journal of Forecasting* **22**, 679–688.

Hyndman, R. J., Koehler, A. B., Ord, J. & Snyder, R. D. (2008), *Forecasting with Exponential Smoothing: The State Space Approach*, 1st edn, Springer-Verlag.

Isengildina, O., Irwin, S. H. & Good, D. L. (2006), Empirical confidence intervals for wasde forecasts of corn, soybean and wheat prices, 2006 Conference, April 17-18, 2006, St. Louis, Missouri 18995, NCR-134 Conference on Applied Commodity Price Analysis, Forecasting, and Market Risk Management.

Kourentzes, N. & Athanasopoulos, G. (2020*a*), 'Elucidate structure in intermittent demand series', **288**, 141–152.

Kourentzes, N. & Athanasopoulos, G. (2020*b*), 'Elucidate structure in intermittent demand series', **288**, 141–152.

Mishina, Y., Tsuchiya, M. & Fujiyoshi, H. (2014), Boosted random forest, *in* '2014 International Conference on Computer Vision Theory and Applications (VISAPP)', Vol. 2, pp. 594–598.

Morde, V. (2019), 'Xgboost algorithm: Long may she reign!'.
**URL:** *https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d*

Mushtaq, R. (2011), 'Augmented dickey fuller test', *SSRN Electronic Journal* .

Ord, K., Fildes, R. & Kourentzes, N. (2017), *Principles of Business Forecasting–2nd ed*, wessex, inc.

Ord, K., Koehler, A. & Snyder, R. (1995), 'Estimation and prediction for a class of dynamic nonlinear statistical models', *Journal of the American Statistical Association* **92**.

Probst, P., Bischl, B. & Boulesteix, A.-L. (2018), 'Tunability: Importance of hyperparameters of machine learning algorithms'.

Silverman, B. W. (1986), CRC Press.

Smith, S. & Sincich, T. (1988), 'Stability over time in the distribution of population forecast errors', *Demography* **25**, 461–74.

Trapero, J., Cardós, M. & Kourentzes, N. (2018*a*), 'Empirical safety stock estimation based on kernel and garch models', *Omega* .

Trapero, J., Cardós, M. & Kourentzes, N. (2018*b*), 'Quantile forecast optimal combination to enhance safety stock estimation', *International Journal of Forecasting* **35**, 239–250.

Willemain, T., Smart, C. & Schwarz, H. (2004), 'A new approach to forecasting intermittent demand for service parts inventories', *International Journal of Forecasting* **20**, 375–387.

Williams, W. & Goodman, M. (1971), 'A simple method for the construction of empirical confidence limits for economic forecasts', *Journal of the American Statistical Association* **66**, 752–754.