Abstract

The well-developed ETS (Exponential Smoothing or Error, Trend, Seasonality) method incorporating a family of exponential smoothing models in state space representation has been widely used for automatic forecasting. The existing ETS method uses information criteria for model selection by choosing an optimal model with the smallest information criterion among all models fitted to a given time series. The ETS method under such a model selection scheme suffers from computational complexity when applied to large-scale time series data. To tackle this issue, we propose an efficient approach for ETS model selection by training classifiers on simulated data to predict appropriate model component forms for a given time series. We provide a simulation study to show the model selection ability of the proposed approach on simulated data. We evaluate our approach on the widely used forecasting competition data set M4, in terms of both point forecasts and prediction intervals. To demonstrate the practical value of our method, we showcase the performance improvements from our approach on a monthly hospital data set.

Keywords: ETS, model component selection, time series features, M4 dataset, hospital dataset

1. Introduction

In today’s data-rich environment, modern business decisions often involve forecasting large collections of time series data, and people often want to select optimal models for their data in an effective and automatic manner (Petropoulos et al., 2022). However, selecting one optimal model requires professional knowledge and relevant experience, making automatic and accurate forecasting not a straightforward task (Li et al., 2022).
Hyndman et al. (2002) incorporated a family of exponential smoothing models into a well-developed modeling framework namely ETS for automatic forecasting. The ETS method is able to identify three components of a given time series: error, trend, and seasonality, as well as their component forms (additive or multiplicative). The error component can be identified as additive (‘A’) or multiplicative (‘M’). The trend and seasonality components can be none (‘N’), ‘A’ or ‘M’. Also, the trend can be damped or not.

In the context of ETS modeling, model selection refers to identifying appropriate forms of three components (error, trend and seasonality) for a given time series. The ETS method uses information criteria such as AIC (Akaike Information Criterion), AICc (The corrected Akaike Information Criterion), and BIC (Bayesian Information Criterion) for the selection purpose, and it proceeds in two steps: (i) calculate information criteria of all candidate models fitted to data; and (ii) select a model with the smallest information criterion among all models. However, in modern business environment, large retailers often need to forecast large-scale data from thousands of products in a forecast cycle (Li et al., 2022). For example, Walmart has roughly 5,000 stores in the U.S., and it is required to generate approximately 1 billion unique forecasts for business decisions (Seaman, 2018). The existing ETS method under such a model selection scheme is computationally expensive when used to forecast large-scale time series data. There is an urgent need to propose an efficient approach for ETS model selection to forecast large collections of time series in a batch and effective manner.

The issue of the computation cost of a large family of models has attracted some attention in recent studies. Nikolopoulos & Petropoulos (2018) proposed suboptimal forecasting solutions for faster and robust systems where they compared the performance of a suboptimal search for smoothing parameters of exponential smoothing models with that of a more computationally expensive and optimal search. Petropoulos & Grushka-Cockayne (2021) also questioned the necessity of such a large family of models, and suggested a change from a focus on the search of suboptimal parameters to that of suboptimal models.

A data-driven approach that uses time series features for forecast model selection and combination over large-scale time series data has flourished in recent studies (e.g., Montero-Manso et al., 2020; Li et al., 2020; Talagala et al., 2021). However, the interest of existing feature-based model selection focused more on the selection among different types of models. Until now, the use of features for model selection among a family of models has not yet attracted enough attention. In order to improve the efficiency of the ETS method, we explore the possibility of the use of features for ETS model selection by training classifiers on simulated data to predict model component forms for a given time series.
We show performance improvements in model selection from our method on simulated data. We conduct extensive empirical evaluations of the proposed method by applying it to 95 thousand real time series from the Makridakis forecasting competitions (Makridakis et al., 2020). Significance tests results show that the proposed approach results in robust forecasts at long-term and all horizons for yearly, quarterly and monthly data. To demonstrate the practical value of the proposed approach, we show the performance improvements from our method on a hospital data set compared with the information criteria method.

The key innovations of our approach are as follows.

• The existing use of features for model selection focused more on the selection among different types of models while our work makes the first attempt to select an appropriate model among a family of models.
• The proposed ETS model selection approach can effectively reduce computational cost when applied to large-scale time series data by predicting model component forms using pre-trained classifiers.

The rest of the article is organized as follows: Section 2 offers a short literature review on relevant studies. Section 3 describes the methodology for the proposed approach. Section 4 presents the empirical results including a simulation study, real data applications and significance tests. Section 5 offers a case study. Section 7 offers our discussions and insights. Finally, Section 8 provides our conclusions.

2. Background research

2.1. The exponential smoothing family of models

The exponential smoothing methods have been around since the 1950s when Brown (1956) proposed a simple exponential smoothing method for demand prediction. The work of Brown (1959) and Gardner Jr (1985) led to the development of exponential smoothing models towards automated forecasting (Hyndman et al., 2002). Ord et al. (1997) and Hyndman et al. (2002) showed that all exponential smoothing models can be well written in the form of state space representation. Hyndman et al. (2002) incorporated a family of exponential smoothing models into a well-developed modeling framework namely ETS. The framework incorporates state space models, parameter estimation, point prediction and interval prediction.

2.2. Information criteria for model selection

Information criteria approaches are designed for parametric methods under the framework of maximum likelihood estimation. They are used to correct for the bias of maximum likelihood
by adding a penalty term to compensate for the over-fitting of more complex models (Bishop, 2006). These commonly used information criteria are employed to evaluate how well a given model fits the data, and how complicated the fitted model is.

Although information criteria could partially address the problem of over-fitting (Meira et al., 2021), the fact that the information criteria are based on one-step prediction error implies that the selected model does not necessarily produce accurate and reliable forecasts when applied to long forecasting horizons (Fildes & Petropoulos, 2015). To tackle this issue, cross-validation can be used to assess the performance of models over multiple time windows, but this is not necessarily guaranteed, and Billah et al. (2006) showed that the performance of information criteria approaches is superior to that of the commonly used validation approach. However, the use of cross-validation is suitable for longer series and is computationally expensive (Li et al., 2022). What’s more, the information criteria approaches do not take parameters uncertainty into consideration (Bishop, 2006), and in practice, they tend to select simple models.

2.3. Feature-based model selection and combination

The use of machine learning for model selection is not novel, many attempts have been made at feature-based forecasting for univariate time series. Reid (1972) pointed out that the forecasting performance is influenced by time series features. Kang et al. (2017) projected time series into an instance space, and provided some insights into the forecasting performance of different models. Recently, time series features have been used for forecast model selection and combination, aiming at selecting an appropriate model or predicting combination weights of all candidate models for a given time series. The performance of feature-based forecasting is highly dependent on the selection of an appropriate set of time series features. Collopy & Armstrong (1992) developed a rule-based system to select optimal models for the series based on 99 features. Talagala et al. (2018) trained a decision tree for the model selection purpose using 42 features. Montero-Manso et al. (2020) trained XGBoost to predict an optimal combination weight of each candidate model. Their method ranked 2nd in M4 competition (Makridakis et al., 2020). Talagala et al. (2021) predicted forecasting error using time series features with a Bayesian multivariate surface regression model. Instead of using manual features, Li et al. (2020) proposed an automated approach to extracting time series imaging features, and the extracted features are also used for forecast model combination.
3. Feature-based ETS model component selection

3.1. Taxonomy of exponential smoothing models in the ETS method

Table 1 presents the taxonomy of exponential smoothing models in the ETS method. Theoretically, the number of exponential smoothing models from the combinations of three component forms can reach 30, as shown in Table 1. However, in practice, the 11 models illustrated in Table 1 with gray background would lead to infinite forecast variance problems when applied to long forecasting horizons. For more details about this issue, see Chapter 15 of Hyndman et al. (2008). Considering that models with multiplicative trend would lead to unrealistically explosive forecasts (Petropoulos & Grushka-Cockayne, 2021), the R package forecast (Hyndman et al., 2020) has, by default, excluded these four models ('MMN', 'MMdN', 'MMM', and 'MMdM') by setting the argument `allow.multiplicative.trend` as `False` in the function `ets()`. Hence, the number of applicable models shaded with purple in Table 1 in the ETS method is 15. For non-seasonal data (e.g., yearly data), the number of applicable models is six.

| Trend       | Additive Error | Multiplicative Error |
|-------------|----------------|----------------------|
|             | Seasonality    |                      |
|             | N   | A   | M   | N   | A   | M   |
| N (None)    | ANN | ANA | ANM | MNN | MNA | MNM |
| A (Additive)| AAN | AAA | AAM | MAN | MAA | MAM |
| Ad (Additive damped)| AAdN | AAda | AAdM | MAdN | MAdA | MAdM |
| M (Multiplicative) | AMN | AMA | AMM | MMN | MMA | MMM |
| Md (Multiplicative damped) | AMdN | AMdA | AMdM | MMdN | MMdA | MMdM |

3.2. The proposed framework

The aim of this work is to employ machine learning methodology to predict an optimal ETS model for a given time series. A simple approach to this task would be to view it as a multi-label classification problem, that is, we train a classifier to link time series features with the 15 applicable ETS models in Table 1. However, training a classifier with 15 classes is not an easy task. To tackle this issue, an alternative way is to adopt a divide and conquer strategy (Zhou, 2021) that trains a classifier to predict each component form separately.

Fig. 1 shows the proposed framework of feature-based ETS model selection consisting of the following two parts:
1. **Offline (Training phase).** The training phase consists of the process of data simulation, feature extraction as well as classifier building and evaluation. In the simulation process, we will simulate some time series from the 15 applicable ETS models. The extracted features over the simulated series with corresponding model component forms will be used for the classifier building and evaluation. Three classifiers will be trained on simulated data separately for the prediction of error, trend and seasonality component forms.

2. **Online (Testing phase).** The testing part requires feature extraction over any new time series. These extracted features are used as input to the pre-trained classifiers from the training phase to obtain error, trend and seasonality component forms. We will check the applicability of a selected model from the combination of predicted component forms, and model adjustment will be performed depending on whether the selected model is applicable. Finally, we fit the selected model to data, and then make predictions from the fitted model using the function `forecast::ets()` (Hyndman et al., 2020) with the argument `model` being the selected model form.

![Diagram](image)

Fig. 1. The proposed framework of feature-based ETS model component selection.

3.3. **Feature extraction**

A collection of 42 time series features developed by Montero-Manso et al. (2020) has been used for forecast model selection and combination over large collections of time series with good results in recent studies (e.g., Montero-Manso et al., 2020; Wang et al., 2021; Talagala et al., 2021). The combination method of Montero-Manso et al. (2020) ranked 2nd in one of largest time series competitions namely M4 competition (Makridakis et al., 2020). The collection of features is available in the R package `M4metalearning` (Montero-Manso et al., 2020).
Apart from the above features used for forecasting, a collection of 22 high-performing features carefully selected from 7,000 features in the **htcsa** toolbox (Fulcher & Jones, 2017) has exhibited strong classification performance across different time series problems (Lubba et al., 2019). The collection of features is available in the R package **Rcatch22** (Henderson, 2021). In this work, in order to characterize an observed time series more comprehensively, and thus improve the classification accuracy of model selection, we choose 59 features from these two classes of features for feature extraction. A set of five features from Montero-Manso et al. (2020) using estimated parameters of fitted ETS models to characterize a time series is discarded. The specific features as well as their corresponding descriptions can be found in Appendix A.

### 3.4. Classifier building for the prediction of ETS model component forms using LightGBM

XGBoost and pGBRT are effective implementations of a popular machine learning algorithm GBDT (Gradient Boosting Decision Tree) with many engineering optimizations. Nonetheless, the efficiency and scalability of both are still unsatisfactory when applied to data with high feature dimension and large size (Ke et al., 2017). To tackle this issue, Ke et al. (2017) proposed a new GBDT implementation namely LightGBM with two novel techniques GOSS (Gradient-based One-Side Sampling) and EFB (Exclusive Feature Bundling). With GOSS, only small data with large gradients are used for the estimation of information gain. With EFB, the number of features is reduced by bundling mutually exclusive features.

In this study, instead of training a classifier with 15 classes using LightGBM, we separately train three classifiers \( f_e, f_t, \) and \( f_s \) to link time series features with each model component form. The three classifiers \( f_e, f_t, \) and \( f_s \) for the prediction of error, trend and seasonality component forms are shown in Table 2.

Table 2. Three classifiers used for the prediction of three component forms of an ETS model for a given time series.

| Classifier | Description | Number of classes | Class labels |
|------------|-------------|-------------------|--------------|
| \( f_e \) | A classifier for the prediction of error component form | 2 | ‘A’, ‘M’ |
| \( f_t \) | A classifier for the prediction of trend component form | 3 | ‘A’, ‘Ad’, ‘N’ |
| \( f_s \) | A classifier for the prediction of seasonality component form | 3 | ‘A’, ‘M’, ‘N’ |

### 3.5. Model check and adjustment

Each pre-trained classifier will provide a certain probability that each component form is selected. Before fitting a selected model to data, it is required to confirm if the model from
the combination of three predicted component forms is applicable. Fig. 2 presents the specific procedure of model check as well as the corresponding adjustment.

- **Check 1.** Due to that we train each classifier on all simulated data with different frequencies, it is unavoidable to mistakenly predict a seasonal ETS model for yearly data. If so, we set the seasonality component of the selected model as ‘N’.

- **Check 2.** If a selected model is unapplicable (like ‘ANM’, ‘AAM’, and ‘AAdM’), we replace its error and seasonality component forms with those from a model with the largest probability of being selected among all alternative models. The selected probability of an alternative model is calculated by multiplying the probabilities of each component.

- **Check 3.** Applying a selected model with multiplicative errors to a time series with zero or negative values would lead to instability of the model (Hyndman & Athanasopoulos, 2018). To tackle this issue, we change its error component form to ‘A’. Subsequently, we need to check if the updated model is applicable. If not, we replace its seasonality component form with one with the largest probability among all alternative forms.

- **Check 4.** To ensure a valid estimation of parameters in a model with a damped trend, the function `forecast::ets()` (Hyndman et al., 2020) requires that the number of a time series should be larger than the number of parameters plus four when fitting it to data. If this condition is not met, we change a damped trend form to another one with a selected probability being ranked second among all trend component forms.

---

Fig. 2. The procedure of model check and adjustment.
4. Empirical evaluation

4.1. Simulation study

4.1.1. Data simulation

Considering that M4 competition data (Makridakis et al., 2020) has good representativeness of reality (Spiliotis et al., 2020), to simulate some series that are realistic, we obtain some prior information from M4 data for this simulation study, as shown in Table 3. The length sets for simulating time series with different frequencies are obtained from M4 data set. Particularly, for an ETS model with ‘Ad’ trend, it is required to set \( \phi \) parameter to control the degree of damping, and we let \( \phi \sim U(0.80, 0.98) \) with reference to the function `forecast::ets()` (Hyndman et al., 2020). A model set with 15 applicable ETS models is used for quarterly and monthly data simulation, while another one with six models is for yearly data. In simulating a time series with a specific frequency, we randomly chose an ETS model form as well as a length value from the corresponding model set and length set.

| Parameter | Description | Value |
|-----------|-------------|-------|
| frequency | Period of time series | 1: yearly; 4: quarterly; 12: monthly |
| model     | The 15 applicable models, six for yearly data | Randomly chosen from the model sets |
| obs       | Length of time series | Randomly chosen from the length sets obtained from M4 data set |
| phi       | Damping parameter for a model with ‘Ad’ trend | \( U(0.80, 0.98) \) |

Finally, we simulated 60,000 yearly, 60,000 quarterly and 120,000 monthly time series. The numbers for each seasonal and non-seasonal ETS model are approximately 12,000 and 22,000. Fig. 3 shows some typical simulated time series from the 15 applicable models.

To visualize if the simulated data set has a good representation of reality, we project the simulated data as well as M4 data into a two-dimensional feature space with t-SNE (Van der Maaten & Hinton, 2008), as shown in Fig. 4. Given the two-dimensional feature spaces of M4 and simulated data sets, we quantify the miscoverage of the simulated data set over M4 data set in the following steps (Kang et al., 2020).

- Cut the x and y dimensions of the instance space into \( N_b = 30 \) bins, and obtain \( N_b^2 = 900 \) subgrids.
• In the constructed subgrids, let

\[ I_{i,S} = \begin{cases} 0 & \text{if no time series in the simulated data set fall into the } i\text{-th subgrid;} \\ 1 & \text{otherwise.} \end{cases} \]  (1)

An analogous definition of \( I_{i,M} \) for M4 data set.

• The relative miscoverage of the simulated data set over M4 data set is

\[ \text{miscoverage}_{S/M} = N_b^{-2} \sum_{i=1}^{N_i^2} [(1 - I_{i,S}) \times I_{i,M}]. \]  (2)

The miscoverage values of simulated data over M4 data are 0.026, 0.003 and 0.002 for yearly, quarterly and monthly data respectively. Together with Fig. 4, we conclude that the simulated data set has good coverage over M4 data, and thus has good representativeness of reality.

4.1.2. Performance of model selection on the simulated series

We split the simulated data into two parts: 80% of the series is used for training data and the rest for testing data. Three sets of optimal hyper-parameters of LightGBM from a search in subspaces of the hyper-parameter spaces are obtained by using a 5-fold cross-validation procedure on the training data. More details about these optimal hyper-parameters are available in Appendix B.

In this simulation study, to demonstrate the model selection ability of the proposed method, we compare its performance on the testing data with that of the information criteria method. Two widely used metrics Accuracy and Macro F\(_1\)-score are used for the evaluation of classification. The Accuracy is calculated as

\[ \text{Accuracy} = \frac{TS}{N} \times 100\%, \]  (3)

where \( TS \) is the number of series with true selected component forms or models, and \( N = 48,000 \) is the number of testing series.

The \( F_1 \)-score

\[ F_1\text{-score}_i = 2 \times \frac{\text{Recall}_i \times \text{Precision}_i}{\text{Recall}_i + \text{Precision}_i} \times 100\% \]  (4)

is used to measure the classification performance of the \( i\)-th class by combining its Recall and Precision into a single metric.

The Macro \( F_1 \)-score

\[ \text{MacroF}_1\text{-score} = \left( \frac{1}{m} \sum_{i=1}^{m} F_1\text{-score}_i \right) \times 100\% \]  (5)

gives the same importance to each class by equally averaging \( F_1\text{-score}_i, i = 1, \ldots, m \) of all classes where \( m \) is the number of classes.
Table 4 presents the classification performance of our method against the information criteria method in the tasks of separate ETS model component selection as well as whole model selection. The ETS method using information criteria for model selection is implemented using the function `forecast::ets()` (Hyndman et al., 2020) with AICc as a default model selection criterion.

Fig. 3. Typical simulated series from the 15 models in the ETS method.
From Table 4, we can observe that our proposed method is superior to the information criteria method in all tasks on the simulated data.

Table 4. Classification performance of our proposed method against the information criteria method with regard to Accuracy and Macro $F_1$-score. Entries in bold highlight that our approach outperforms the information criteria approach.

| Component          | Information criteria | Feature-based |
|--------------------|-----------------------|---------------|
| Error component    | 85.05%                | 88.35%        |
| Trend component    | 69.73%                | 74.41%        |
| Seasonality component | 92.27%                | 95.48%        |
| ETS model          | 55.50%                | 63.51%        |

4.1.3. Feature importance analysis for each component selection

To have a better understanding of which features are important to each component form prediction, we identify the ten most important features among 59 features by comparing their information gain (Ke et al., 2017). The more the gain is, the more important the feature is. Fig. 5
presents the feature importance results for each component form prediction. We can observe that

- **Error component.** The error related feature `spike` that calculates the variance of the leave-one-out variances of the remainder from an STL decomposition plays the most important role in the identification of the error component form. The feature `flat_spots` that measures the longest flat spot in a series also has a significant effect on error component form identification. What’s more, the seasonal and trend strength of a series has an impact on error component selection in the sense that the error component is obtained after detrending and seasonal adjustment.

- **Trend component.** Some trend related features such as `trend` and `unitroot_pp` are identified as the two most important features. The feature `localSimple_mean3_stdErr` reflecting mean error from a rolling forecasting is also important to the trend component selection.

- **Seasonality component.** Some seasonality related features such as `seasonal_strength`, `seas_pacf` and `seas_acf1` are identified as important features. The feature `lumpiness` has a significant impact on the seasonality component selection due to that it calculates the variance of variances on different blocks with length of a seasonal period.

- **Summary.** The most important features are either from **M4metalearning** (Montero-Manso et al., 2020) or **Rcatch22** (Henderson, 2021), which justifies the suitability of using two sets of features for the tasks of component selection.
4.2. Forecasting with M4 competition data

To confirm the effectiveness of our proposed approach in real data, we compare our forecasts with those from the information criteria method over M4 data set. Considering that ETS models are not feasible for data with multiple seasonalities (e.g., daily and hourly data) or frequency larger than 24 (e.g., weekly data), we consider yearly, quarterly, and monthly subsets of M4 data set for this forecasting evaluation. The series in M4 competition come from different fields such as demographic, industry, finance, economics and others. The M4 dataset is in the M4comp2018 R package (Montero-Manso et al., 2018). The forecasting horizons for yearly, quarterly and monthly data are 6, 8 and 18 respectively. The total number of series used in this forecasting evaluation is 23,000, 24,000 and 48,000 for yearly, quarterly and monthly data respectively.

We employ two widely used metrics namely MASE (the Mean Absolute Scaled Error) (Hyndman & Koehler, 2006) and sMAPE (the Symmetric Mean Absolute Percentage Error) for point forecast evaluation. These two metrics are calculated as

\[
MASE = \frac{1}{h} \sum_{t=1}^{h} \frac{\sum_{s=s+1}^{s} |y_{T+t} - \hat{y}_{T+t}|}{\sum_{t=1}^{T} \sum_{s=s+1}^{s} |y_t - y_{t-s}|},
\]

\[
sMAPE = \frac{1}{h} \sum_{t=1}^{h} \frac{2 |y_{T+t} - \hat{y}_{T+t}|}{|y_{T+t}| + |\hat{y}_{T+t}|},
\]
where $\hat{y}_t$ is the point forecast of the real value $y_t$ of the time series at point $t$, $h$ is the forecasting horizon, and $s$ is the seasonal period of data.

The MSIS (Mean Scaled Interval Score) (Gneiting & Raftery, 2007)

$$
MSIS = \frac{1}{h} \sum_{t=1}^{h} \left( \frac{1}{2} \left( y_{T+t}^l - y_{T+t}^u \right) \right)^2 \mathbb{1}_{y_T^l < y_T^u} + \frac{1}{2} \left( y_{T+h}^l - y_{T+h}^u \right) \mathbb{1}_{y_{T+h}^l < y_{T+h}^u} + \frac{1}{2} \left( y_{T+h}^l - y_{T+h}^u \right) \mathbb{1}_{y_{T+h}^l > y_{T+h}^u}
$$

is used to measure the accuracy of a $(1 - \alpha) \times 100\%$ prediction interval $(\hat{y}_{T+t}^l, \hat{y}_{T+t}^u)$ where $\mathbb{1}_A$ is the indicator function for the condition $A$, and we set $\alpha = 0.05$.

4.3. Forecasting results

The ETS method using information criteria for model selection is implemented using the function `forecast::ets()` (Hyndman et al., 2020) with AICc as a default model selection criterion. For our method, we use the pre-trained classifiers from the simulation study for ETS model selection, and then make predictions using these selected models. Table 5 presents the mean forecasting results, and we can observe that:

- **Point forecasts.** Our method resulted in better point forecasts at almost all the forecasting horizons for yearly (except for $h = 1-4$ regarding sMAPE), quarterly (except for $h = 7-8$ regarding MASE) and monthly data.

- **Prediction intervals.** The performance of our approach is not better than that of the information criteria method regarding the mean of MSIS.

Table 5. Performance of our proposed method against the information criteria method with regard to the mean of MASE, sMAPE and MSIS values over M4 yearly, quarterly and monthly data. Entries in bold highlight that our approach outperforms the information criteria approach.

|                  | MASE   | sMAPE  | MSIS   |
|------------------|--------|--------|--------|
|                  | 1-2    | 3-4    | 5-6    | 1-6    | 1-2    | 3-4    | 5-6    | 1-6    | 1-2    | 3-4    | 5-6    | 1-6    |
| Yearly           |        |        |        |        |        |        |        |        |        |        |        |        |        |
| Information criteria | 1.934 | 3.456 | 4.943  | 3.444  | 9.809  | 15.743 | 20.516 | 15.356 | 18.983 | 34.772 | 50.934 | 34.897 |        |
| Feature-based    | 1.931 | 3.392 | 4.820  | 3.381  | 9.933  | 15.746 | 20.368 | 15.349 | 21.956 | 36.878 | 53.386 | 37.407 |        |
| Quarterly        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| Information criteria | 0.774 | 1.256 | 1.598  | 1.161  | 7.454  | 11.023 | 13.448 | 10.291 | 6.113  | 10.188 | 13.356 | 9.452  |        |
| Feature-based    | 0.772 | 1.256 | 1.599  | 1.160  | 7.402  | 10.943 | 13.410 | 10.232 | 6.390  | 10.694 | 14.045 | 9.918  |        |
| Monthly          |        |        |        |        |        |        |        |        |        |        |        |        |        |
| Information criteria | 0.650 | 0.967 | 1.227  | 0.948  | 9.933  | 13.637 | 17.005 | 13.525 | 5.152  | 8.552  | 11.188 | 8.297  |        |
| Feature-based    | 0.639 | 0.959 | 1.207  | 0.935  | 9.687  | 13.385 | 16.466 | 13.179 | 5.352  | 8.773  | 11.491 | 8.539  |        |

We further provide the corresponding median values in Table 6. We can observe that our proposed method is superior to the information criteria method at almost all the forecasting
horizons for yearly (except for \( h = 1-2 \) regarding MSIS), quarterly (except for \( h = 1-3 \) regarding sMAPE and \( h = 7-8 \) regarding point forecasts) and monthly (except for \( h = 1-6 \) regarding sMAPE) data.

Table 6. Performance of our proposed method against the information criteria method with regard to the median of MASE, sMAPE and MSIS values over M4 yearly, quarterly and monthly data.

|        | MASE       | sMAPE     | MSIS       |
|--------|------------|-----------|------------|
|        | 1-2 3-4 5-6 1-6 | 1-2 3-4 5-6 1-6 | 1-2 3-4 5-6 1-6 |
| Yearly | Information criteria | 1.339 2.255 3.162 2.329 | 5.342 8.669 11.629 8.966 | 7.777 13.905 20.623 15.487 |
|        | Feature-based | 1.318 2.214 3.102 2.289 | 5.263 8.426 11.320 8.721 | 8.238 13.162 18.278 15.280 |
|        | 1-3 4-6 7-8 1-8 | 1-3 4-6 7-8 1-8 | 1-3 4-6 7-8 1-8 |
| Quarterly | Information criteria | 0.573 0.910 1.123 0.886 | 3.513 5.598 6.835 5.608 | 3.993 6.199 7.658 5.977 |
|        | Feature-based | 0.570 0.904 1.124 0.886 | 3.530 5.597 6.882 5.600 | 3.980 6.115 7.575 5.939 |
|        | 1-6 7-12 13-18 1-18 | 1-6 7-12 13-18 1-18 | 1-6 7-12 13-18 1-18 |
| Monthly | Information criteria | 0.497 0.708 0.877 0.736 | 4.223 6.654 8.329 6.995 | 3.225 4.933 6.339 5.040 |
|        | Feature-based | 0.491 0.705 0.867 0.728 | 4.247 6.641 8.258 6.964 | 3.219 4.917 6.260 5.030 |

4.4. Forecasting accuracy analysis

We use violin plots to visually show the distribution of MASE, sMAPE and MSIS values at all horizons, as shown in Fig. 6. We can observe that for yearly and quarterly data, the ranges of MASE of our method are smaller than that of the information criteria method. For monthly data, the distributions of forecasting accuracy of both methods are similar to each other.

Together with Tables 5 and 6, we can conclude that the proposed method resulted in some unusual forecasts that have an effect on the mean of MSIS values, as shown in the first and second plots in the third column of Fig. 6. We find that these outliers come from some unusual series that let our method select some models with multiplicative errors for them, leading to large variances of prediction errors, and thus poor prediction intervals. Some of these unusual series can be available at https://github.com/Richard759/fETSms/blob/master/bad_case.pdf.

4.5. Significance tests

We perform Diebold-Mariano (DM) tests (Harvey et al., 1997) to verify if forecasts from the proposed approach are significantly better or worse than those of the information criteria method. The null hypothesis of this test is that the two approaches have the same forecast accuracy. The alternative hypothesis is that our method is less or more accurate than the benchmark method. In these tests, if the DM tests statistic falls in the lower or upper 2.5% tail of
the standard normal distribution, then we reject the null hypothesis and accept the alternative hypothesis. The DM tests are implemented using `forecast::dm.test()` (Hyndman et al., 2020) in R.

From Table 7, we can observe that:

- For yearly data, our approach significantly outperforms the information criteria method at long-term ($h = 5-6$) and all ($h = 1-6$) horizons.
- For quarterly data, our approach is significantly superior to the information criteria method at mid ($h = 4-6$), long-term ($h = 7-8$) and all ($h = 1-8$) horizons.
- For monthly data, our approach is significantly better than the information criteria method at long-term ($h = 13-18$) and all ($h = 1-18$) horizons.
• The proposed method resulted in robust forecasts at long-term and all horizons for all data.

Table 7. Diebold-Mariano (DM) tests for comparing the predictive accuracy of the proposed method with the information criteria method. The entries show the percentage of times the forecasts of our method are significantly better or worse than the forecasts of the information criteria method at different horizons. Entries in bold highlight that our method is significantly better than the information criteria method.

| Yearly | Quarterly | Monthly |
|--------|-----------|---------|
| 1-2    | 1-3       | 1-6     |
| 3-4    | 4-6       | 7-12    |
| 5-6    | 7-8       | 13-18   |
| 1-6    | 1-3       | 1-18    |

better: 1.600 3.574 5.770 16.839 2.350 9.163 6.329 17.879 11.135 16.683 20.410 22.846
worse: 1.630 3.613 5.448 15.539 2.646 8.921 6.096 17.242 11.279 17.031 19.779 22.238

5. Case study: forecasting with a monthly hospital data set

To further demonstrate the practical value of the proposed method in even specific forecasting domains, we apply our method to a monthly hospital data set. The data set consisting of 767 monthly series has been analyzed by Hyndman et al. (2008). The data set is in expsmoothing R package (Hyndman, 2015). We save the last 18 observations for forecasting evaluation. We employ the pre-trained classifiers from the simulation study to select an appropriate ETS model for any series in this set, and then make predictions using these selected models. From Table 8, we can find that our proposed approach resulted in better forecasting results than the information criteria method at almost all horizons (except for $h = 1-6$ regarding MSIS) in terms of point forecasts and prediction intervals.

Table 8. Performance of our proposed method against the information criteria method with regard to the mean of MASE, sMAPE and MSIS values over the hospital data. Entries in bold highlight that our approach outperforms the information criteria approach.

| MASE | sMAPE | MSIS |
|------|-------|------|
| 1-6  | 1-6   | 1-6  |
| 7-12 | 7-12  | 7-12 |
| 13-18| 13-18 | 13-18|
| 1-18 | 1-18  | 1-18 |

Information criteria
- MASE: 0.726 0.804 0.895 0.808
- sMAPE: 16.724 18.138 19.855 18.239
- MSIS: 4.536 5.519 6.486 5.514

Feature-based
- MASE: 0.712 0.780 0.865 0.786
- sMAPE: 16.364 17.778 19.353 17.832
- MSIS: 4.556 5.433 6.422 5.470
6. Computational time for producing forecasts

We save and compare the computational time for producing forecasts from our method as well as the information criteria method over the above M4 and hospital data sets. To ensure a fair comparison, all the experiments were carried out on an AMD Ryzen 7 4800H 2.90 GHz server with 8 cores. The computation time of the proposed method in the training phase for time series simulation, feature extraction and classifier building is 1,162, 1,683 and 7,386 seconds respectively.

From Table 9, we can observe that our proposed method is significantly faster than the information criteria method for producing forecasts in the online phase. It cannot be denied that the most computational process of our method is time series simulation, feature extraction, and classifier training in the training phase. However, this is acceptable in real application scenarios because classifiers only need to be trained once, and can be applied to different real data without training each time.

Table 9. Computational time in seconds for producing forecasts from our method and information criteria method in the online phase.

|                      | M4            | Hospital      |
|----------------------|---------------|---------------|
|                      | Information criteria | Feature-based | Information criteria | Feature-based |
| Feature extraction   | -             | 588           | -             | 5             |
| Model selection and forecasting | 5,782 | 367           | 67            | 4.1           |
| Total                | 5,782         | 955           | 67            | 9.1           |

7. Discussions

Considering that the existing ETS method using information criteria for model selection suffers from computational complexity when applied to large-scale time series data, we propose an efficient approach for ETS model selection that trains three classifiers on simulated data to predict model component forms. A simulation study has shown the model selection ability of the proposed method. DM tests results confirm that our approach produces robust forecasts at long-term and all horizons for yearly, monthly and quarterly data.

The most computational process of the proposed method is time series simulation, feature extraction, and classifier building in the training phase. However, this is acceptable in practice because classifiers for component form prediction only need to be trained once, and can be
applied to different real data without retraining. The good forecasting results on a hospital data set from a specific domain have demonstrated the robustness of the proposed method. We also have shown the proposed method is more efficient to produce forecasts over the M4 and hospital data sets in the online phase.

However, the good forecasting performance of the proposed approach depends on the selection of an appropriate set of time series features and careful parameter tuning. Compared with the information criteria method, our method achieved impressive performance, but sacrifices interpretability, that is, we have no idea about how well a selected model from our classifiers fits the data and how complicated the model is.

Our approach feature-based ETS model component selection (fETSmcs) is implemented as an R package. Our code is open-source and publicly available at https://github.com/Richard759/fETSmcs.

8. Concluding remarks

To tackle the inefficiency issue in the current big data settings, we explore the use of features for ETS model selection by training classifiers on simulated data to predict component forms. We have provided a simulation study as well as two real data applications to demonstrate of the effectiveness the proposed method.

However, the proposed approach depends on the manual choice of an appropriate set of features. Some research (e.g., Li et al., 2020) introduces automatic features for forecast model combination, and one potential direction for this research would be to adopt automatic features for ETS model selection. A potential avenue for future research would be to design a model selection criterion that incorporates information criteria as well as feature-based selection, aiming at achieving impressive performance while still maintaining good interpretability.

9. Acknowledgements

We are grateful to the Editor and two anonymous reviewers for their helpful comments that improved the contents of this paper.

We are also grateful to Professor Yanfei Kang from Beihang University for her feedback in the early stages of the development of the proposed method in this paper.
References

Billah, B., King, M. L., Snyder, R. D., & Koehler, A. B. (2006). Exponential smoothing model selection for forecasting. *International Journal of Forecasting, 22*, 239–247.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Brown, R. G. (1956). Exponential smoothing for predicting demand. Little, Cambridge, Massachusetts, .

Brown, R. G. (1959). *Statistical forecasting for inventory control*. McGraw/Hill.

Collopy, F., & Armstrong, J. S. (1992). Rule-based forecasting: Development and validation of an expert systems approach to combining time series extrapolations. *Management Science, 38*, 1394–1414.

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

Fulcher, B. D., & Jones, N. S. (2017). hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. *Cell Systems, 5*, 527–531.

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

Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association, 102*, 359–378.

Harvey, D., Leybourne, S., & Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting, 13*, 281–291.

Henderson, T. (2021). *Rcatch22: Calculation of 22 CAnonical Time-Series CHaracteristics*. R package version 0.1.12.

Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O’Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., & Yasmeen, F. (2020). *forecast: Forecasting functions for time series and linear models*. URL: [http://pkg.robjhyndman.com/forecast](http://pkg.robjhyndman.com/forecast) R package version 8.14.

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. (2015). *expsmooth: Data Sets from “Forecasting with Exponential Smoothing”*. R package version: 2.3.
Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. 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., Snyder, R. D., & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18, 439–454.

Kang, Y., Hyndman, R. J., & Li, F. (2020). Gratis: Generating time series with diverse and controllable characteristics. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 13, 354–376.

Kang, Y., Hyndman, R. J., & Smith-Miles, K. (2017). Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting*, 33, 345–358.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 3146–3154.

Li, X., Kang, Y., & Li, F. (2020). Forecasting with time series imaging. *Expert Systems with Applications*, 160, 113680.

Li, X., Petropoulos, F., & Kang, Y. (2022). Improving forecasting by subsampling seasonal time series. *International Journal of Production Research*, (pp. 1–17).

Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D., & Jones, N. S. (2019). catch22: Canonical time-series characteristics. *Data Mining and Knowledge Discovery*, 33, 1821–1852.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36, 54–74.

Meira, E., Oliveira, F. L. C., & Jeon, J. (2021). Treating and pruning: New approaches to forecasting model selection and combination using prediction intervals. *International Journal of Forecasting*, 37, 547–568.

Mietus, J., Peng, C., Henry, I., Goldsmith, R., & Goldberger, A. (2002). The pnnx files: re-examining a widely used heart rate variability measure. *Heart*, 88, 378–380.
Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., & Talagala, T. S. (2020). FFORMA: Feature-based forecast model averaging. *International Journal of Forecasting, 36*, 86–92.

Montero-Manso, P., Netto, C., & Talagala, T. S. (2018). *M4comp2018: Data from the M4-Competition*. R package version: 0.1.0.

Nikolopoulos, K., & Petropoulos, F. (2018). Forecasting for big data: Does suboptimality matter? *Computers & Operations Research, 98*, 322–329.

Ord, J. K., Koehler, A. B., & Snyder, R. D. (1997). Estimation and prediction for a class of dynamic nonlinear statistical models. *Journal of the American Statistical Association, 92*, 1621–1629.

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E. et al. (2022). Forecasting: theory and practice. *International Journal of Forecasting (in press)*.

Petropoulos, F., & Grushka-Cockayne, Y. (2021). Fast and frugal time series forecasting. *Available at SSRN 3792565*.

Reid, D. (1972). A comparison of forecasting techniques on economic time series. *Forecasting in Action. Operational Research Society and the Society for Long Range Planning*.

Seaman, B. (2018). Considerations of a retail forecasting practitioner. *International Journal of Forecasting, 34*, 822–829.

Spiliotis, E., Kouloumos, A., Assimakopoulos, V., & Makridakis, S. (2020). Are forecasting competitions data representative of the reality? *International Journal of Forecasting, 36*, 37–53.

Svetunkov, I. (2021). *smooth: Forecasting Using State Space Models*. URL: [https://github.com/config-i1/smooth](https://github.com/config-i1/smooth) R package version 3.1.1.

Talagala, T. S., Hyndman, R. J., Athanasopoulos, G. et al. (2018). Meta-learning how to forecast time series. *Monash Econometrics and Business Statistics Working Papers, 6*, 18.

Talagala, T. S., Li, F., & Kang, Y. (2021). FFORMPP: Feature-based forecast model performance prediction. *International Journal of Forecasting (in press)*.

Wang, X., Kang, Y., Petropoulos, F., & Li, F. (2021). The uncertainty estimation of feature-based forecast combinations. *Journal of the Operational Research Society*, (pp. 1–15).

Wang, X., Wirth, A., & Wang, L. (2007). Structure-based statistical features and multivariate time series clustering. In *Seventh IEEE international conference on data mining (ICDM 2007)* (pp. 351–360). IEEE.
Zhou, Z.-H. (2021). *Machine Learning*. Springer.
Appendices

A. Feature descriptions

Tables 10 and 11 introduces the 59 used features as well as their corresponding descriptions.

B. Experimental setup for LightGBM

We performed hyper-parameter optimization with grid search by measuring the AUC (Area Under the ROC (Receiver Operating Characteristic) Curve) on a 20% holdout version of the training data in a 5-fold cross-validation procedure. The searching ranges of four hyper-parameters are as follows.

- **num_leaves**: The number of leaves ranges from 8 to 128.
- **min_delta_in_leaf**: The minimum number of the records a leaf may have ranges from 10 to 110.
- **max_bin**: The maximum number of bins that feature values will be bucketed in ranges from 5 to 255.
- **num_boost_round**: The number of boosting stages ranges from 100 to 1100.

Considering computation time, we set fixed values for some other hyper-parameters. Table 12 reports the optimal values of four hyper-parameters of LightGBM as well as other hyper-parameters with fixed values for the three classifiers $f_e$, $f_t$ and $f_s$. 
Table 10. The **M4metalearning** features (Montero-Manso et al., 2020) used for ETS model component selection.

| Index | Feature            | Description                                                                 |
|-------|--------------------|------------------------------------------------------------------------------|
| 1     | x_acf              | First ACF of a series                                                        |
| 2     | x_acf10            | Sum of the squared first ten ACF of a series                                 |
| 3     | diff1_acf1         | First ACF of a differenced series                                            |
| 4     | diff1_acf10        | Sum of the squared first ten ACF of a differenced series                     |
| 5     | diff2_acf1         | First ACF of a twice-differenced series                                       |
| 6     | diff2_acf10        | Sum of squared first ten ACF of a series                                      |
| 7     | seas_acf1          | ACF of a seasonally differenced series, 0 for non seasonal series            |
| 8     | ARCH_LM            | A statistic for AR conditional heteroscedasticity                           |
| 9     | crossing_point     | The number of times a series crosses its median                              |
| 10    | entropy            | The spectral entropy of a series                                             |
| 11    | flat_spots         | The number of flat spots in a series                                         |
| 12    | arch_acf           | Sum of squares of first 12 ACF of a pre-whitened series                      |
| 13    | garch_acf          | Sum of squares of first 12 ACF of the squared residuals from a fitted GARCH(1,1) model |
| 14    | arch_r2            | $R^2$ value of an AR model applied to a pre-whitened series                  |
| 15    | garch_r2           | $R^2$ value of a GARCH(1,1) model applied to a pre-whitened series           |
| 16    | hurst              | Hurst                                                                        |
| 17    | lumpiness          | Lumpiness                                                                    |
| 18    | nonlinearity       | Nonlinearity                                                                 |
| 19    | x_pacf5            | Sum of squared first 5 PACF of a series                                       |
| 20    | diff1x_pacf5       | Sum of squared first 5 PACF of a differenced series                          |
| 21    | diff2x_pacf5       | Sum of squared first 5 PACF of a twice differenced series                    |
| 22    | seas_pacf          | PACF of at first seasonal lag, 0 for non seasonal series                     |
| 23    | nperiods           | The number of seasonal periods in a series                                   |
| 24    | seasonal_period    | The length of the seasonal period                                            |
| 25    | trend              | Trend                                                                        |
| 26    | spike              | Spike                                                                        |
| 27    | linearity          | Linearity                                                                    |
| 28    | curvature          | Curvature                                                                    |
| 29    | e_acf1             | First ACF of the remainder series from an STL decomposition                  |
| 30    | e_acf10            | Sum of first 10 squared ACF of the remainder series from an STL decomposition |
| 31    | seasonal_strength  | Strength of seasonality                                                      |
| 32    | peak               | Peak                                                                          |
| 33    | trough             | Trough                                                                       |
| 34    | stability          | Stability                                                                     |
| 35    | unitRoot_kpss      | A statistic for unit root test                                                |
| 36    | unitRoot_pp        | Another statistic for unit root test                                         |
| 37    | series_length      | Series length                                                                 |
Table 11. The \texttt{Rcatch22} features (Lubba et al., 2019) used for ETS model component selection.

| Index | Feature | Description |
|-------|---------|-------------|
| 38    | histogram\_mode5 | Mode of z-scored distribution using 5-bin histogram |
| 39    | histogram\_mode10 | Mode of z-scored distribution using 10-bin histogram |
| 40    | binaryStats\_mean\_longStretch1 | Longest period of consecutive values above mean |
| 41    | outlierInclude\_p\_001\_mdrmd | Time intervals between extreme events above mean |
| 42    | outlierInclude\_n\_001\_mdrmd | Time intervals between extreme events below mean |
| 43    | f1ecac | First 1/e crossing of ACF |
| 44    | firstmin\_ac | First minimum of ACF |
| 45    | summaries\_welch\_rect\_area\_5\_1 | Total power of lowest fifth of frequencies in a power spectrum |
| 46    | summaries\_welch\_rect\_centroid | Centroid of the Fourier power spectrum |
| 47    | localSimple\_mean\_3\_stdErr | Prediction error from a rolling forecasting |
| 48    | trev\_1\_num | A statistic for time-reversibility |
| 49    | histogramAMI\_even\_2\_5 | Automutual information |
| 50    | autoMutualInfoStats\_40\_gaussian\_fmmi | First minimum of an information function |
| 51    | hrv\_classic\_pnn\_40 | Proportion of continuous differences above 0.04\(\sigma\) (Mietus et al., 2002) |
| 52    | binaryStats\_diff\_longStretch0 | Longest length of incremental decreases |
| 53    | motifThree\_quantile\_hh | Entropy of two continuous letters in a symbolization |
| 54    | localSimple\_mean\_1\_tauresrat | Correlation changes after iterative differencing |
| 55    | embed2\_dist\_tau\_d\_expFit\_meanDiff | An exponential fit to continuous distances in 2-d space |
| 56    | fluctAnal\_2\_dfa\_50\_1\_2\_logi\_prop\_r1 | Proportion of slower timescale fluctuations with DFA scaling |
| 57    | fluctAnal\_2\_rsrangefit\_50\_1\_logi\_prop\_r1 | Proportion of slower timescale fluctuations with rescaled range fits |
| 58    | transitionMatrix\_3ac\_sumDiagCov | Trace of covariance of a transition matrix |
| 59    | periodicityWang\_th\_0\_01 | A periodicity measure (Wang et al., 2007) |
Table 12. Values of hyper-parameters of LightGBM on the training data.

| Hyper-parameters         | $f_e$ | $f_t$ | $f_s$ |
|--------------------------|-------|-------|-------|
| eta                      | 0.05  | 0.05  | 0.05  |
| num_leaves               | 92    | 80    | 64    |
| min_data_in_leaf         | 90    | 100   | 60    |
| max_bin                  | 175   | 225   | 175   |
| num_boost_round          | 600   | 800   | 1000  |
| bagging_fraction         | 0.8   | 0.8   | 0.8   |
| bagging_freq             | 4     | 4     | 4     |
| feature_fraction         | 0.7   | 0.7   | 0.7   |
| force_col wise           | True  | True  | True  |
| bagging_seed             | 123   | 123   | 123   |