Part 1: Visualizing forecasting algorithm performance using time series instance spaces

2. time series features

2.1 choosing features

- F_1 : Spectral entropy a relatively

  large value of F_1 indicates more uncertainty about the future, and is harder to forecast
- F_2 : Strength of trend

  $F_2 = 1 - \frac{var(R_t)}{var(X_t - S_t)}$
- F_3 : Strength of seasonality

  $F_3 = 1 - \frac{var(R_t)}{var(X_t - T_t)}$
- F_4 :Seasonal period

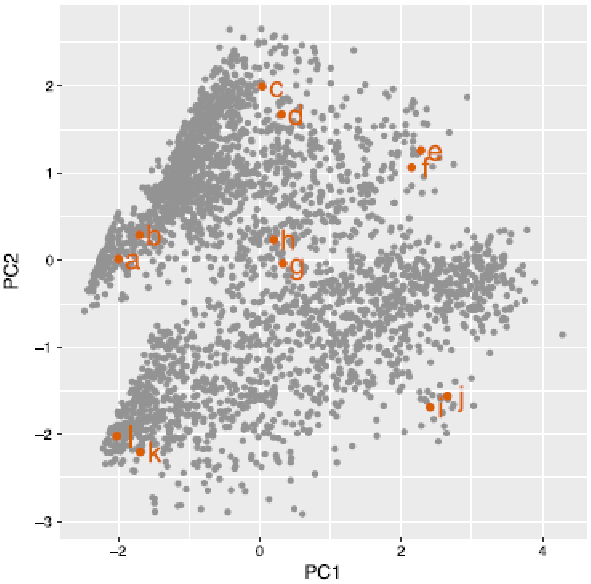
  $F_4 = 4$ for quarterly data; $F_4 = 12$ for monthly data.
- F_5 : First order autocorrelation

  $F_5 = Corr(R_t, R_{t-1})$
- F_6 : Optimal Box-Cox transformation parameter

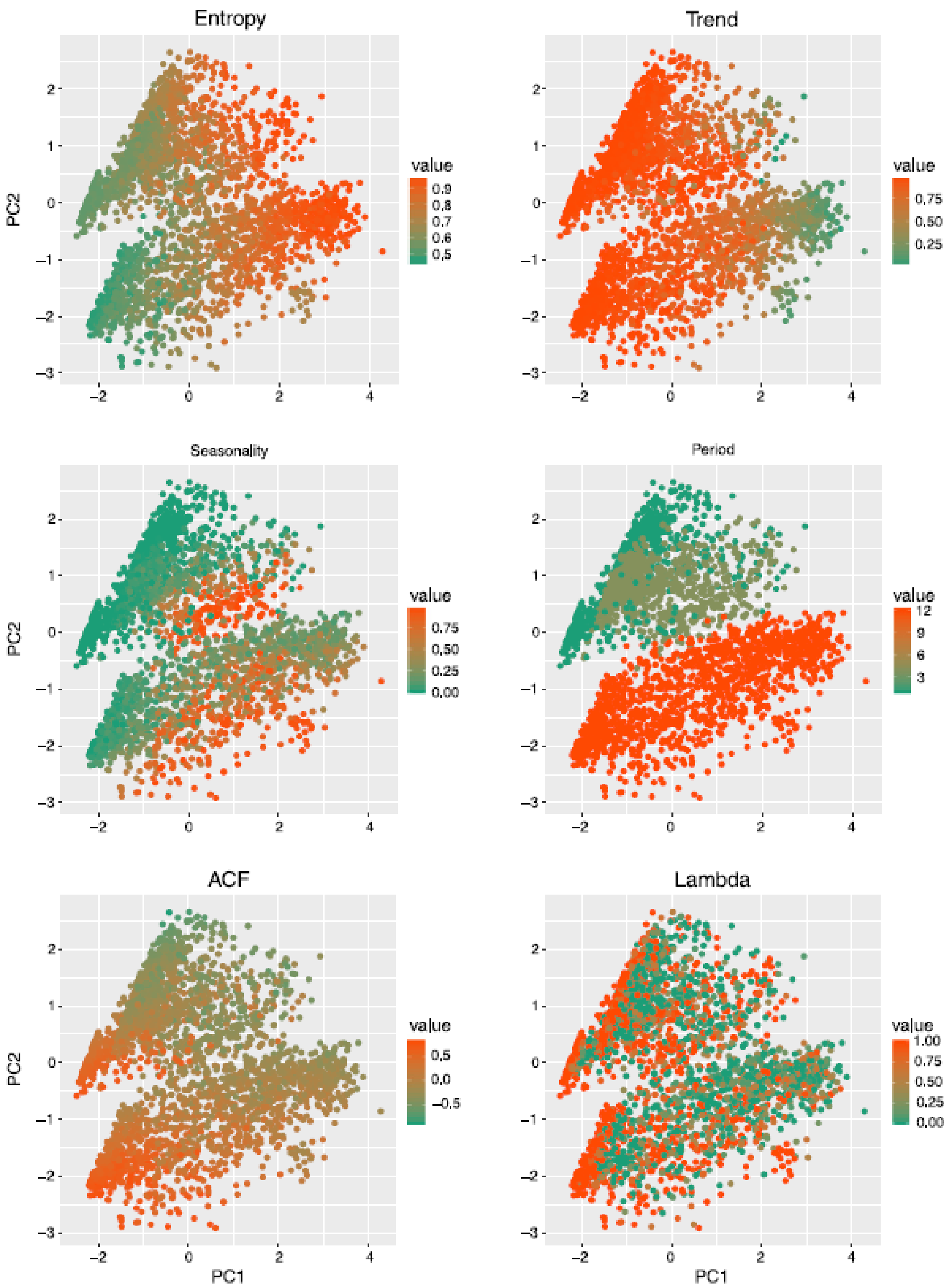
  measure the degree of change of variation in the data

2.2 Visualizing data

Use a dimension reduction method **PCA** to project the M3 dataset all onto a two-dimensional space in order to allow a easy visualization of the data.

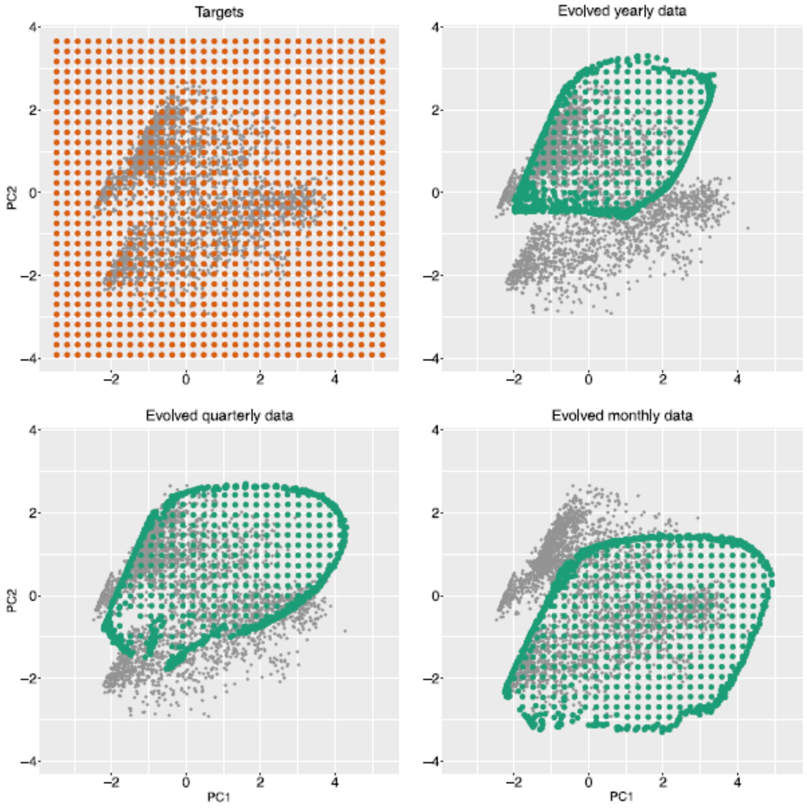


Interesting properties of instances are demonstrated in their feature distributions across the instance space. More interestingly, the current time series in the M3 dataset have failed to fill the entire instance space.



3. New time series generation in instance space

Given a target point , our goal is to evolve a new time series instance which is as close as possible to the target point when projected to the two dimensional instance space. The process relies on a genetic algorithm.



- There are large parts of the target space where we have not been able to generate series. This suggests that yearly, quarterly and monthly data have natural boundaries within this two-dimensional instance space, probably due to constraints on combinations of features.
- Evolved series are more evenly distributed away from the boundaries, while the M3 data have a higher density on the left side of the space.

  We selected six known time series at random from M3. we use the locations of these known time series as target points, then evolve new time series that lie near these target.

4. Comparison of time series forecasting methods in the instance space.

Benchmark methods: Naive, Seasonal naive, Theta, ETS, ARIMA, STL-AR;

For each series, we compute the **minimum** MASE value achieved from all six methods.

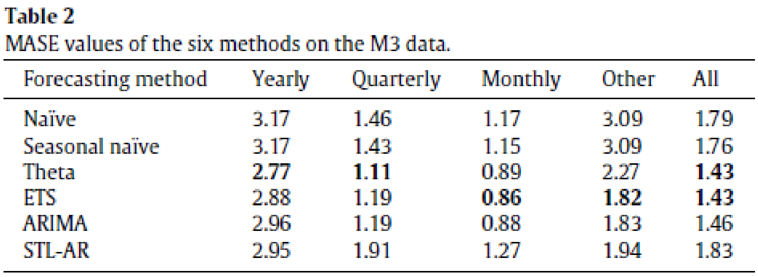
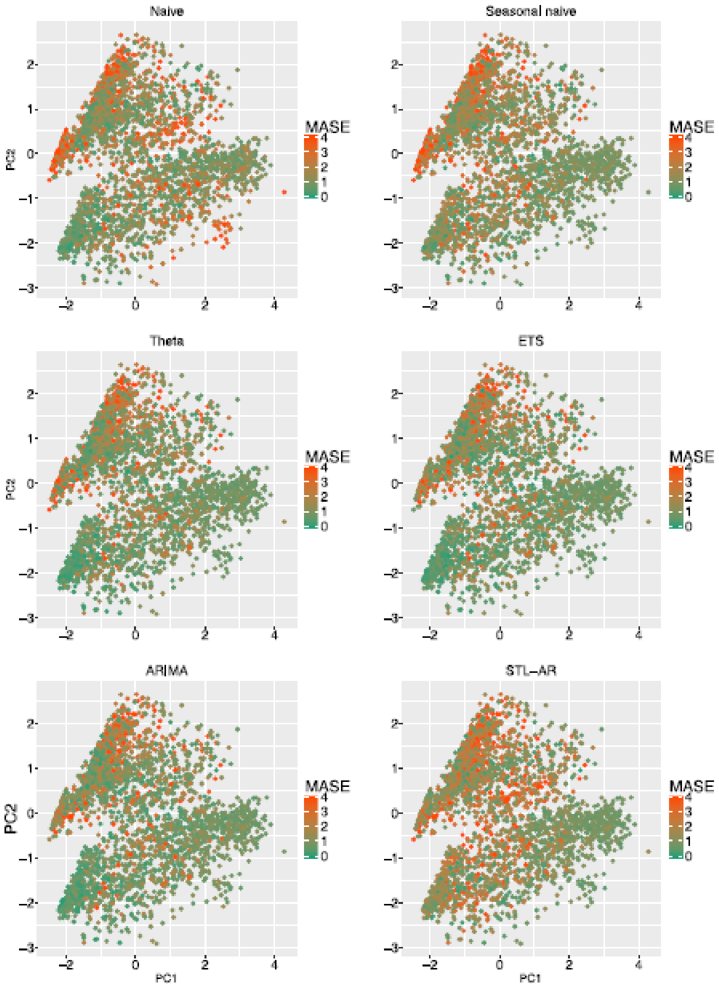


Table 2
MASE values of the six methods on the M3 data.

Forecasting method	Yearly	Quarterly	Monthly	Other	All
Naïve	3.17	1.46	1.17	3.09	1.79
Seasonal naïve	3.17	1.43	1.15	3.09	1.76
Theta	**2.77**	**1.11**	0.89	2.27	**1.43**
ETS	2.88	1.19	**0.86**	**1.82**	**1.43**
ARIMA	2.96	1.19	0.88	1.83	1.46
STL-AR	2.95	1.91	1.27	1.94	1.83

The plots show which particular regions of the instance space were forecast best by each method. The figure does suggest that there are regions of the instance space where some methods are best avoided. For example, the STL-AR performs worse on the quarterly series than using ARIMA directly.



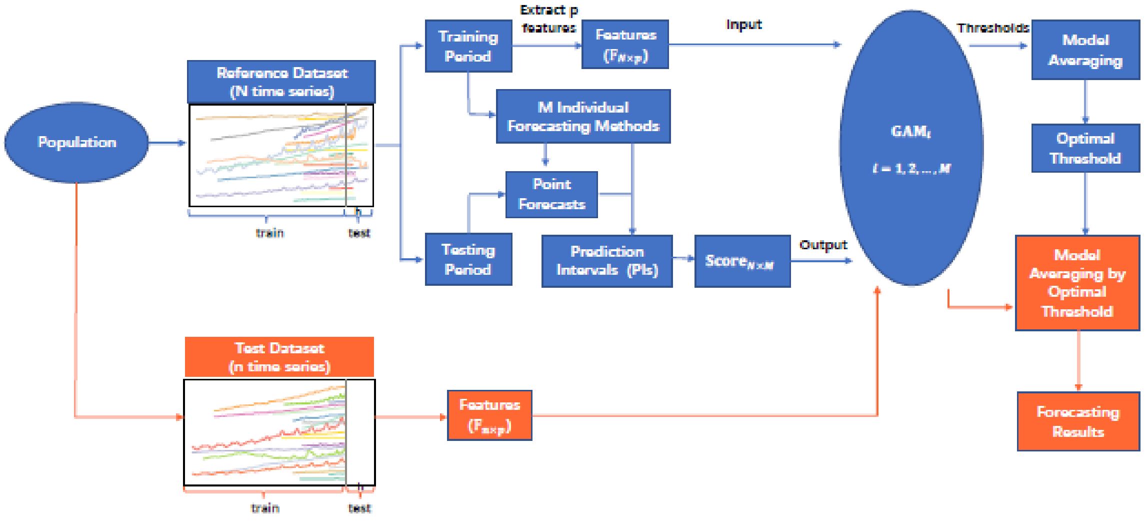
5. Conclusions

- Identify unusual time series which have very different combinations of features to other time series in the collection.
- Propose an algorithm for generating new time series with controllable characteristics.
- Conclusions based on the M3 competition data will not necessarily hold for other collections of time series with different distributions in the feature space.
- some forecasting methods perform better in some regions of the feature space than other methods.

  Further work: developing meta-forecasting algorithms that choose a specific forecasting method based on the location of a time series in the instance space.

Part 2: The uncertainty estimation of feature-based time series forecasts

2. Methodology



2.1 Interval forecast evaluation

In this paper, we adopt the central $(1-\alpha)\times 100\%$ **prediction intervals** for estimating the uncertainty in point forecasts. MSIS scores are applied to measure the performance of the generated PIs.

$$MSIS = \frac{1}{h}\frac{\sum_{t=n+1}^{n+h}(U_t - L_t)I\{Y_t < L_t\} + \frac{2}{\alpha}(Y_t - U_t)I\{Y_t > L_t\}}{\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t - Y_{t-m}|}$$

where Y_t are the true values of the future, $[L_t, U_t]$the generated prediction intervals, h the forecasting horizon, n the length of the historical data, and m the time interval symbolizing the length of the time series periodicity, for example, the yearly, quarterly, and monthly data are 1, 4, and 12, respectively. I is the indicator function, which returns 1 when the judgment function inside is true, otherwise returns 0. □ is the level of significance of the generated prediction intervals.

2.2 Linking time series features with interval forecasting accuracy

GAM (Generalized additive model)

$$g(E(Y)) = A\theta + s_1(X_1) + s_2(X_2) + \cdots + s_p(X_p)$$

where Y is the response variable, $X = (X_1, X_2, \cdots, X_p)$ is a matrix of p explanatory variables, g is the link function used to establish the relationship between $E(Y)$and the set of explanatory variables X, A is a matrix of strictly parametric model components, θ is a vector of corresponding parameters, and the terms $s_1(\cdot), s_2(\cdot), \cdots, s_p(\cdot)$are smooth, non-parametric functions,one for each covariate X_k.

we first establish GAMs to characterize the relationship between interval forecasting accuracy and time series features in the offline part. Considering p extracted features and M pre-prepared candidate forecasting methods, the GAM we trained for the i-th benchmark method can be written as:

$$g(E(log(MSIS_i))) = \beta_{i0} + \beta_{i1}F_1 + \cdots + \beta_{ik}F_k + s_{i1}(F_{k+1}) + \cdots + s_{i(p-k)}(F_p)$$

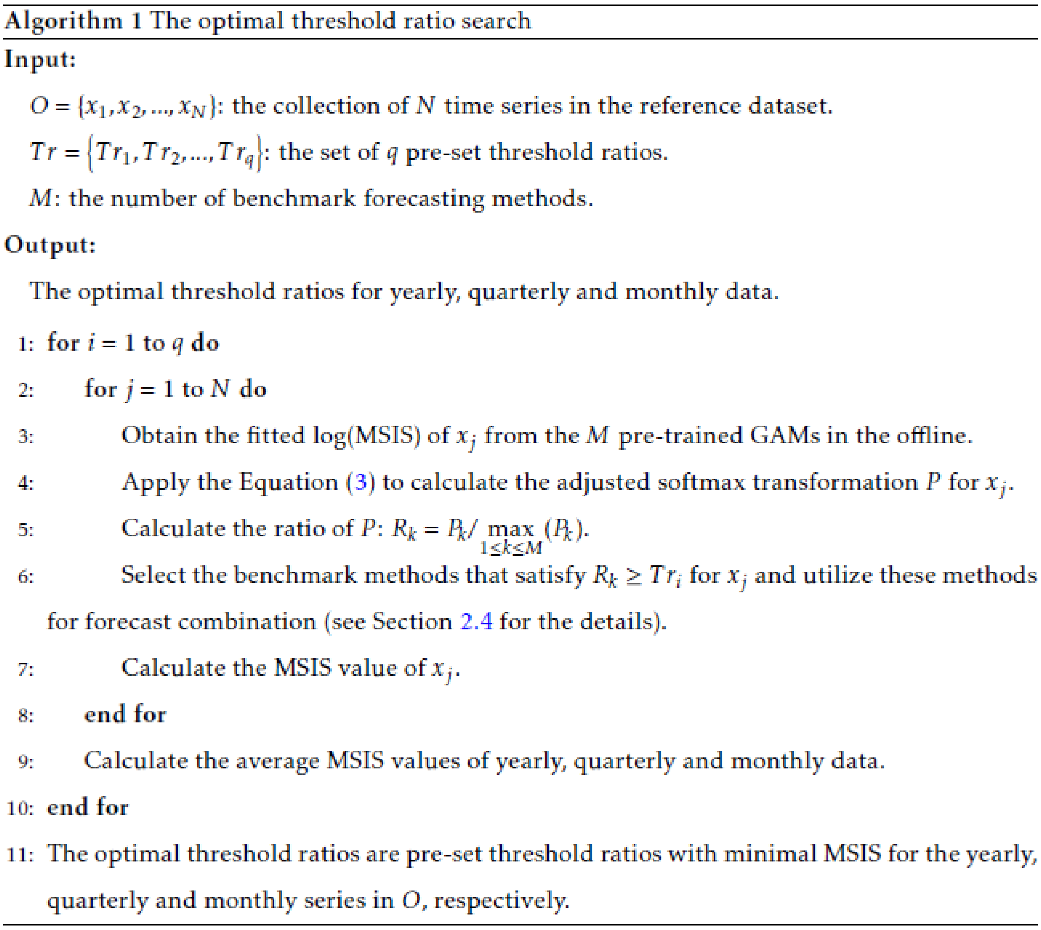
where$i = 1, 2, \cdots, M$, $MSIS_i$ is the score vector of i-th method, $F = \{F_1, \cdots, F_p\}$ denotes a predictor matrix consisting of extracted features, $F_1, \cdots, F_k$ are linear predictors with dummy features, $F_{k+1}, \cdots, F_p$ are predictors that can be modeled non-parametrically in addition to linear terms, g is the link function used to establish the relationship between mean of the response variable and the set of predictors,β_{i0} denotes the intercept of the regression, $\beta_{i1}, \cdots, \beta_{ik}$ are regression coefficients of linear terms, and the terms $s_{i1}(\cdot), \cdots, s_{i(p-k)}(\cdot)$are smooth, non-parametric functions.

Optimal threshold ratio search

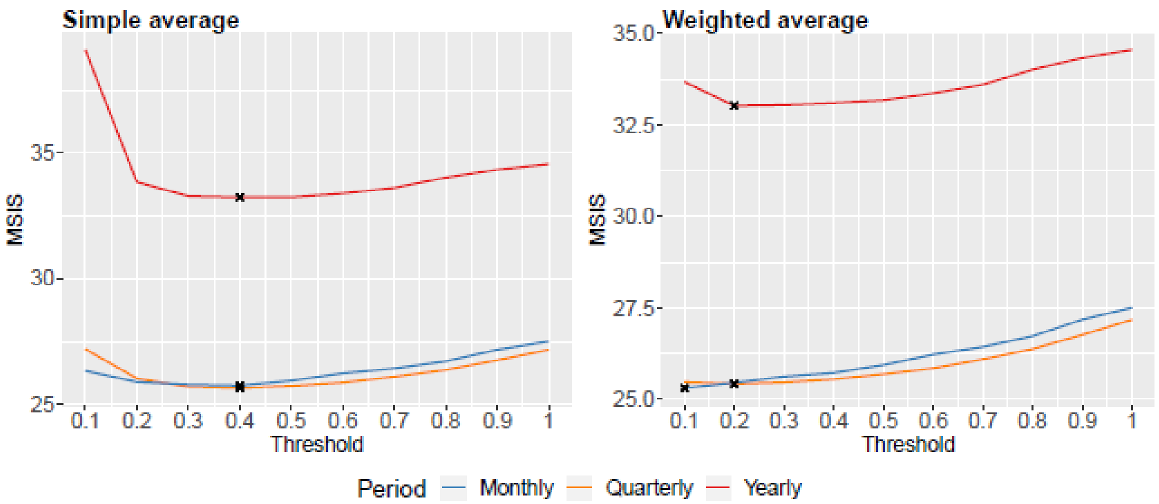
Transfer MSIS scores into probability by adjusted softmax function:

$$P_{ij} = \frac{exp\{\frac{\mu_i - log(\hat{MSIS_{ij}})}{\sigma_i}\}}{\sum_{k=1}^{M} exp\{\frac{\mu_i - log(\hat{MSIS_{ij}})}{\sigma_i}\}},$$

where $i = 1, \cdots, N$ and$j = 1, \cdots, M$,$\mu_i$and $\sigma_i$denote the mean and standard deviation of the fitted values obtained by the M pre-trained GAMs for i-th time series, respectively.



```
Algorithm 1 The optimal threshold ratio search
Input:
    $O = \{x_1, x_2, ..., x_N\}$: the collection of $N$ time series in the reference dataset.
    $Tr = \{Tr_1, Tr_2, ..., Tr_q\}$: the set of $q$ pre-set threshold ratios.
    $M$: the number of benchmark forecasting methods.
Output:
    The optimal threshold ratios for yearly, quarterly and monthly data.
for $i = 1$ to $q$ do
    for $j = 1$ to $N$ do
        Obtain the fitted log(MSIS) of $x_j$ from the $M$ pre-trained GAMs in the offline.
        Apply the Equation (3) to calculate the adjusted softmax transformation $P$ for $x_j$.
        Calculate the ratio of $P$: $R_k = P_k / \max_{1 \le k \le M} (P_k)$.
        Select the benchmark methods that satisfy $R_k \ge Tr_i$ for $x_j$ and utilize these methods
    for forecast combination (see Section 2.4 for the details).
        Calculate the MSIS value of $x_j$.
    end for
    Calculate the average MSIS values of yearly, quarterly and monthly data.
end for
The optimal threshold ratios are pre-set threshold ratios with minimal MSIS for the yearly,
    quarterly and monthly series in $O$, respectively.
```



2.3 Interval combination methods

Assuming T benchmark forecasting methods are selected for a time series according to a pre-defined threshold ratio, the h-step point forecast for the weighted average is defined as:

$$f_{weighted} = \frac{1}{T}\sum_{k=1}^{T} P_k f_k$$

Where P_k denotes the probability of the k-th method being selected and is calculated from the adjusted softmax function.

The lower bound and upper bound of the model averaged prediction interval are defined as:

$$f^l_{weighted} = f_{weighted} - \frac{1}{T}\sum_{k=1}^{T} P_k (f_k - f^l_k),$$

$$f^u_{weighted} = f_{weighted} - \frac{1}{T}\sum_{k=1}^{T} P_k (f^u_k - f_k),$$

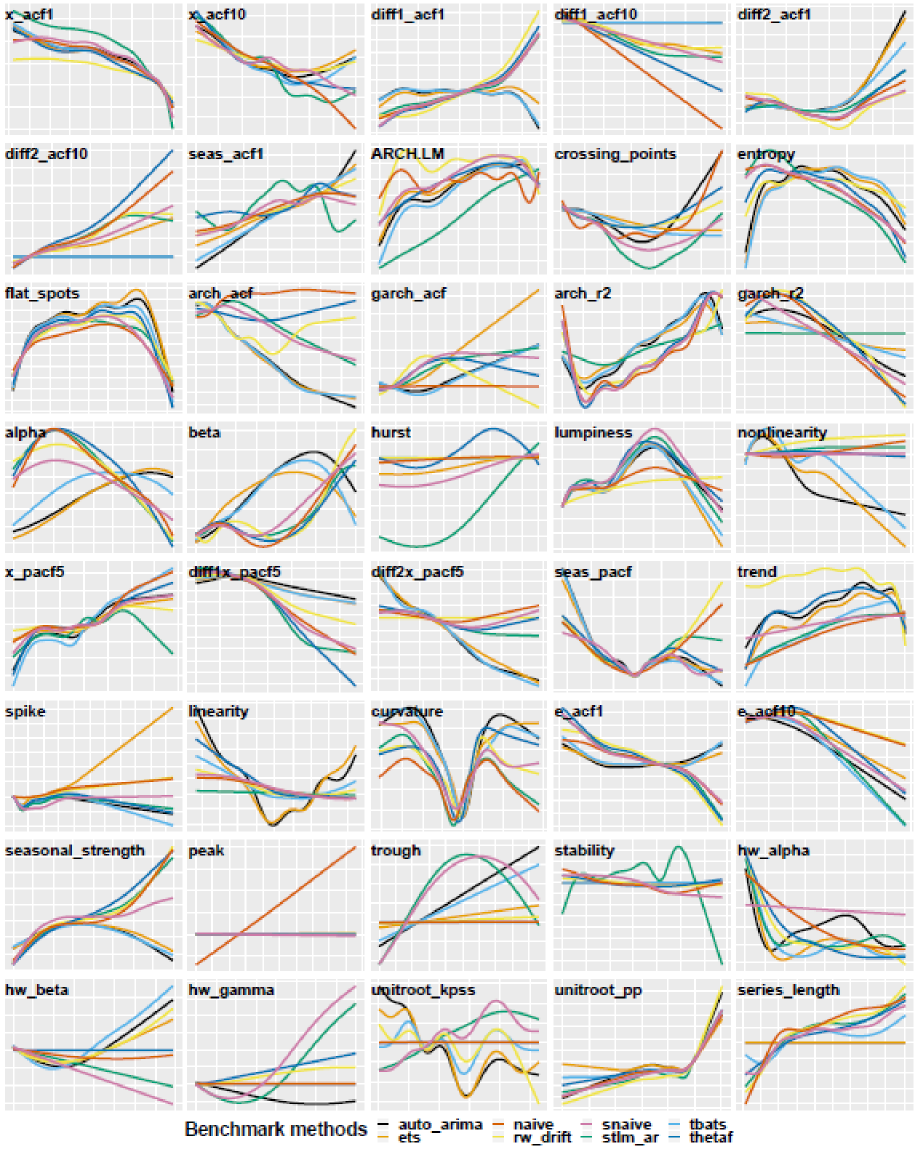
3. Application to the M3 competition data

- reference dataset: time series generated by GRATIS
- test dataset: M3

3.1 Experimental setup

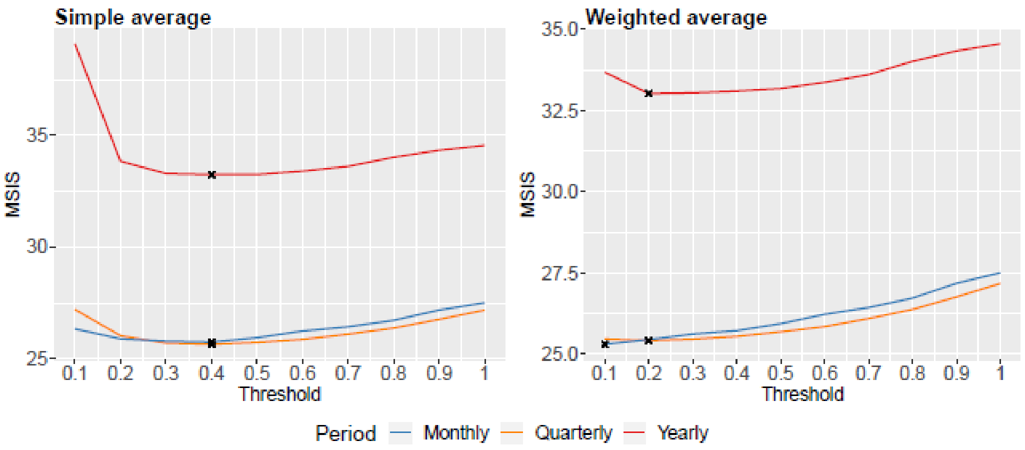
42 features are selected to capture the characteristic of time series from various aspects. We obtain the point forecasts, as well as the prediction intervals for 8 benchmark forecasting methods. By analyzing he partial effects of time series features on the interval forecasting accuracy, we found that:

- different features have different partial effects on the same benchmark method.
- the partial effects of the same feature on each benchmark method are different, while they behave similarly in some cases.



Find the optimal thresholds :

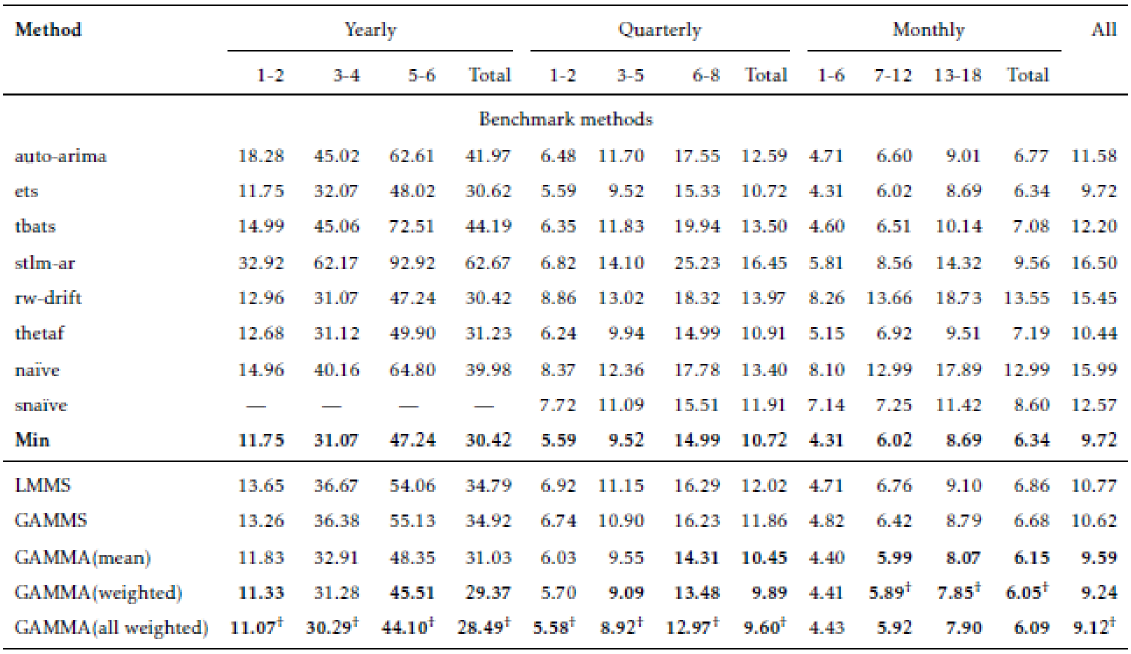
the optimal thresholds for yearly, quarterly and monthly series are all set to 0.4 for the simple average method; For the weighted average combination, we set the optimal thresholds for yearly, quarterly, and monthly series as 0.2, 0.2 and 0.1, respectively.



3.2 Forecasting results

- **MSIS**

For monthly series, GAMMA with the weighted average combination even performs better than GAMMA with all benchmark methods weighted combined. For yearly and quarterly series, GAMMA with all benchmark methods weighted combined ranks best.

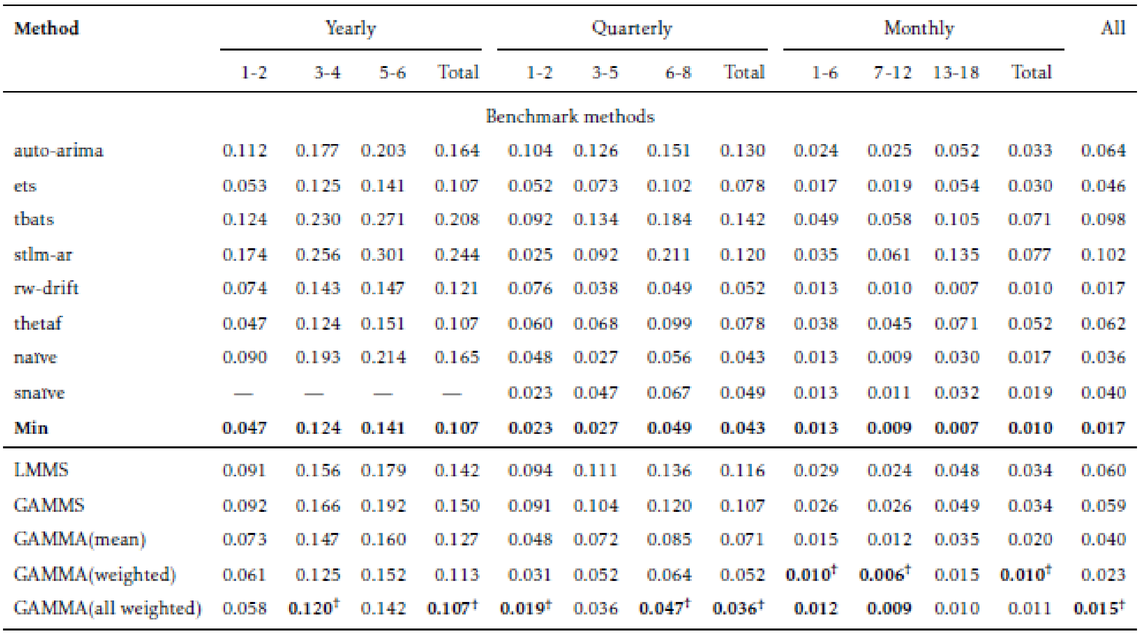


Method	Yearly				Quarterly				Monthly				All
	1-2	3-4	5-6	Total	1-2	3-5	6-8	Total	1-6	7-12	13-18	Total	
Benchmark methods													
auto-arima	18.28	45.02	62.61	41.97	6.48	11.70	17.55	12.59	4.71	6.60	9.01	6.77	11.58
ets	11.75	32.07	48.02	30.62	5.59	9.52	15.33	10.72	4.31	6.02	8.69	6.34	9.72
tbats	14.99	45.06	72.51	44.19	6.35	11.83	19.94	13.50	4.60	6.51	10.14	7.08	12.20
stlm-ar	32.92	62.17	92.92	62.67	6.82	14.10	25.23	16.45	5.81	8.56	14.32	9.56	16.50
rw-drift	12.96	31.07	47.24	30.42	8.86	13.02	18.32	13.97	8.26	13.66	18.73	13.55	15.45
thetaf	12.68	31.12	49.90	31.23	6.24	9.94	14.99	10.91	5.15	6.92	9.51	7.19	10.44
naïve	14.96	40.16	64.80	39.98	8.37	12.36	17.78	13.40	8.10	12.99	17.89	12.99	15.99
snaïve	—	—	—	—	7.72	11.09	15.51	11.91	7.14	7.25	11.42	8.60	12.57
Min	**11.75**	**31.07**	**47.24**	**30.42**	**5.59**	**9.52**	**14.99**	**10.72**	**4.31**	**6.02**	**8.69**	**6.34**	**9.72**
LMMS	13.65	36.67	54.06	34.79	6.92	11.15	16.29	12.02	4.71	6.76	9.10	6.86	10.77
GAMMS	13.26	36.38	55.13	34.92	6.74	10.90	16.23	11.86	4.82	6.42	8.79	6.68	10.62
GAMMA(mean)	11.83	32.91	48.35	31.03	6.03	9.55	**14.31**	**10.45**	4.40	**5.99**	**8.07**	**6.15**	**9.59**
GAMMA(weighted)	**11.33**	31.28	**45.51**	**29.37**	5.70	**9.09**	**13.48**	**9.89**	4.41	**5.89**†	**7.85**†	**6.05**†	**9.24**
GAMMA(all weighted)	**11.07**†	**30.29**†	**44.10**†	**28.49**†	**5.58**†	**8.92**†	**12.97**†	**9.60**†	4.43	**5.92**	**7.90**	**6.09**	**9.12**†

- **MASE**

GAMMA with the weighted average combination performs best on monthly series. GAMMA with all benchmark methods weighted combined generally performs quite well, especially for yearly and quarterly series.

Table 4. Comparison of the ACD values of the feature-based time series forecasting framework and other eight benchmark methods on M3.



Method	Yearly				Quarterly				Monthly				All
	1-2	3-4	5-6	Total	1-2	3-5	6-8	Total	1-6	7-12	13-18	Total	
Benchmark methods													
auto-arima	0.112	0.177	0.203	0.164	0.104	0.126	0.151	0.130	0.024	0.025	0.052	0.033	0.064
ets	0.053	0.125	0.141	0.107	0.052	0.073	0.102	0.078	0.017	0.019	0.054	0.030	0.046
tbats	0.124	0.230	0.271	0.208	0.092	0.134	0.184	0.142	0.049	0.058	0.105	0.071	0.098
stlm-ar	0.174	0.256	0.301	0.244	0.025	0.092	0.211	0.120	0.035	0.061	0.135	0.077	0.102
rw-drift	0.074	0.143	0.147	0.121	0.076	0.038	0.049	0.052	0.013	0.010	0.007	0.010	0.017
thetaf	0.047	0.124	0.151	0.107	0.060	0.068	0.099	0.078	0.038	0.045	0.071	0.052	0.062
naïve	0.090	0.193	0.214	0.165	0.048	0.027	0.056	0.043	0.013	0.009	0.030	0.017	0.036
snaïve	—	—	—	—	0.023	0.047	0.067	0.049	0.013	0.011	0.032	0.019	0.040
Min	**0.047**	**0.124**	**0.141**	**0.107**	**0.023**	**0.027**	**0.049**	**0.043**	**0.013**	**0.009**	**0.007**	**0.010**	**0.017**
LMMS	0.091	0.156	0.179	0.142	0.094	0.111	0.136	0.116	0.029	0.024	0.048	0.034	0.060
GAMMS	0.092	0.166	0.192	0.150	0.091	0.104	0.120	0.107	0.026	0.026	0.049	0.034	0.059
GAMMA(mean)	0.073	0.147	0.160	0.127	0.048	0.072	0.085	0.071	0.015	0.012	0.035	0.020	0.040
GAMMA(weighted)	0.061	0.125	0.152	0.113	0.031	0.052	0.064	0.052	**0.010**†	**0.006**†	0.015	**0.010**†	0.023
GAMMA(all weighted)	0.058	**0.120**†	0.142	**0.107**†	**0.019**†	0.036	**0.047**†	**0.036**†	**0.012**	**0.009**	0.010	0.011	**0.015**†

4. Conclusion

- We have proposed a general framework for interval forecasting based on time series features, which provides the uncertainty estimation of point forecasts.
- The interpretability of the effects of features on the interval forecasting accuracy in the proposed framework.

- We define a threshold ratio in the offline part and find the optimal threshold to select a plurality of appropriate forecasting methods for each time series for model averaging.

###

Part 3: Tree-based methods for clustering time series using domain-relevant attributes.

1. Introduction

- **Types of temporal-only clustering methods**

  The literature on time-series clustering includes several approaches, where studies differ in how the time series characteristics are measured, and on how these measures are clustered.The three main approaches for clustering time series reviewed in Liao (2005) include **the raw-data-based approach, the feature based approach, and the model-based approach**.

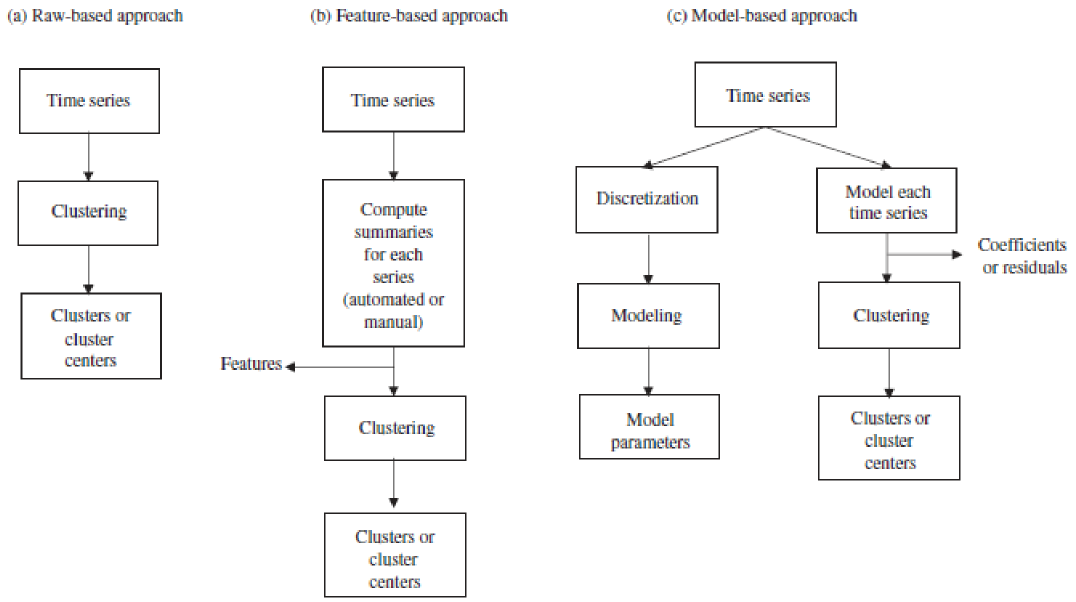

(a) Raw-based approach
Time series
Clustering
Clusters or cluster centers
(b) Feature-based approach
Time series
Compute summaries for each series (automated or manual)
Features
Clustering
Clusters or cluster centers
(c) Model-based approach
Time series
Discretization
Modeling
Model parameters
Model each time series
Coefficients or residuals
Clustering
Clusters or cluster centers

Figure 1. Different approaches for time-series clustering (reproduced and modified from Liao (2005)).

- **Incorporating domain-relevant attributes**

  When a large set of time series arises in the same context, there are often attributes of the series that conceptually link them into subgroups. For example, in a set of time series measuring the sales volume for each item in a supermarket, domain relevant attributes might include the category of the item (e.g. dairy, cosmetics, fruit), as well as whether they are perishable, seasonal, etc.

  In our literature search, domain relevant attributes are used where after the temporal-based clustering is performed, the domain relevant attributes are used to interpret the resulting.

2. Proposed tree-based approach

2.1 Model-based partitioning (MOB) tree

The MOB algorithm was proposed by Zeileis et al. (2008) and includes four main steps:

- Assuming the outcome Y and predictors $X = [X_1, \cdots, X_p]$, fit a parametric model to the data by estimating the parameters based on an objective function.
- Conduct a parameter instability test over the set of splitting variables $Z = [Z_1, \cdots, Z_q]$ and if there is instability, select Z_j with the highest instability; otherwise stop.

- Compute the split point that locally optimises an objective function, and split the model with respect to the variable with the highest instability (when there are some parameters with instability).
- In each resulting sub-sample (node), repeat the above steps.

These steps can be visualized and interpreted as a type of a classification and regression tree. The advantage of using a MOB tree for clustering is the ability to **specify domain-relevant attributes as the splitting variables (Z)**.

2.2 Single-step MOB method

In the single-step method we capture trend, seasonality and autocorrelation of the time series by running the MOB once, using the following linear regression model:

$y_t = \alpha_0 + \alpha_1 t + \beta_1 Season_{1t} + \beta_2 Season_{2t} + \cdots + \beta_m Season_{2m} + \gamma_1 y_{t-1} + \gamma_2 y_{t-2} + \cdots + \gamma_m y_{t-m} + \epsilon_t$ where $y_t (t = 1, 2, \cdots, T)$ is the value of the series at time t, $Season_{jt}$ is a dummy variable taking value 1 if time t is in season $j (j = 1, \cdots, m)$, and y_{t-k} is the *kth* lagged value.

2.3 Two-step MOB method

- step 1: we model the time series by fitting trend and seasonality through a linear regression model and split on potentially useful cross sectional domain-relevant attributes.

$$y_t = \alpha_0 + \alpha_1 t + \beta_1 Season_{1t} + \beta_2 Season_{2t} + \cdots + \beta_m Season_{2m} + \epsilon_t$$

- step 2: we capture autocorrelation by modeling **the forecast errors** from step 1, using a linear regression model with m lags, which is approximately similar to an m-order Autoregressive (AR) model.

$$e^c_{i,t} = y_{i,t} - \hat{y}^c_{i,t},$$

where $\hat{y}^c_{i,t}$ is the fitted value of series i at time t using the step 1 model in cluster c.

$$e^c_{i,t} = \gamma_0 + \gamma_1 e^c_{i,t-1} + \gamma_2 e^c_{i,t-2} + \cdots + \gamma_m e^c_{i,t-m} + \epsilon$$

2.4 Tuning and evaluation methods

- Pruning by reducing splits

  The MSE within a terminal node measures the similarity between time series within the same cluster (within-cluster similarity). We can therefore use this to evaluate performance for different numbers of splits. We thus search for the smallest number of splits that improves the MSE the most. In other words, we search for the smallest tree with acceptable MSE.

- Combining terminal nodes by testing for similarity

  our goal is grouping all similar series into a single cluster. For this purpose, we again use the fitted models in each terminal node and test the differences between the coefficients in each pair of linear regression models using the Chow test (Chow,1960).

3. Application: clustering restaurant sales series

We illustrate the tree-based clustering methods using a set of 611 daily time series from a restaurant chain in Taiwan, where each series measures the daily sales.

domain-relevant information on each series:

(1) Restaurant branch: {A, B, C, D, E, F, G}

(2) Meal category (menu item): {Baked pasta, Baked rice, Hot drink, Iced drink, Pizza thick, Pizza thin, Salad, Snack, Soup, Sweet, crustar, Tapas}

(3) Meal time: {Lunch, Afternoon, Dinner}

(4) Indoor: {Yes, No}

(5) Takeout: {Yes, No}

3.1 Choosing tree depth

- For the single-step method, depth =2 (split = 1) showed the biggest drop in MSE.
- For the two-step method, depth = 2 (split = 1) was best in step 1, and depth = 3 (split = 2) showed the best MSE improvement in step 2.

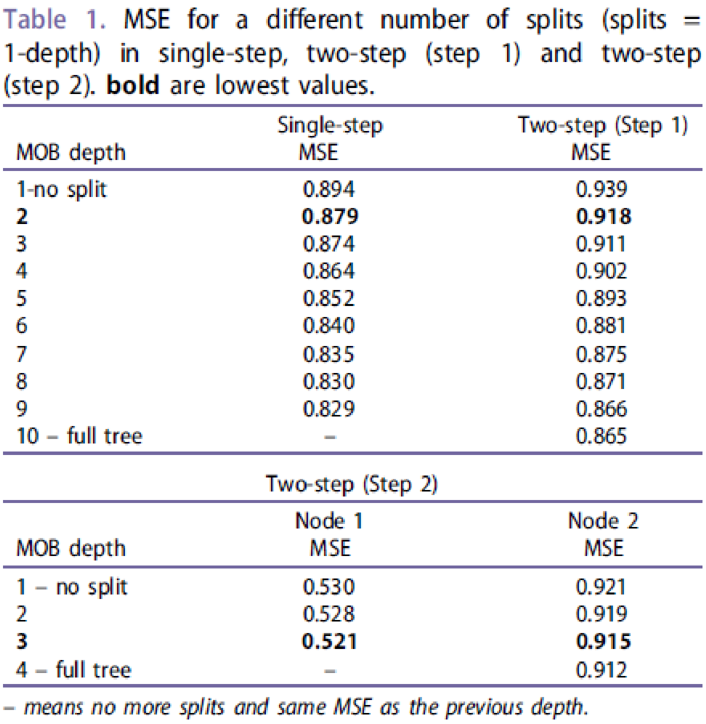


Table 1. MSE for a different number of splits (splits = 1-depth) in single-step, two-step (step 1) and two-step (step 2). **bold** are lowest values.

MOB depth	Single-step MSE	Two-step (Step 1) MSE
1-no split	0.894	0.939
2	**0.879**	**0.918**
3	0.874	0.911
4	0.864	0.902
5	0.852	0.893
6	0.840	0.881
7	0.835	0.875
8	0.830	0.871
9	0.829	0.866
10 – full tree	–	0.865

Two-step (Step 2)

MOB depth	Node 1 MSE	Node 2 MSE
1 – no split	0.530	0.921
2	0.528	0.919
3	**0.521**	**0.915**
4 – full tree	–	0.912

– means no more splits and same MSE as the previous depth.

3.2 Pruning the tree

The Chow test results in table 2 indicates that all the models are different and hence we keep all the nodes separate.

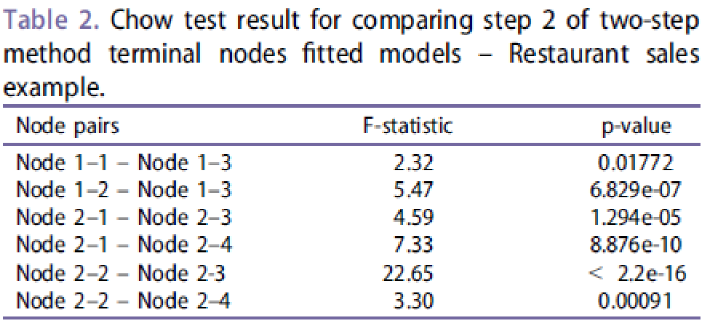
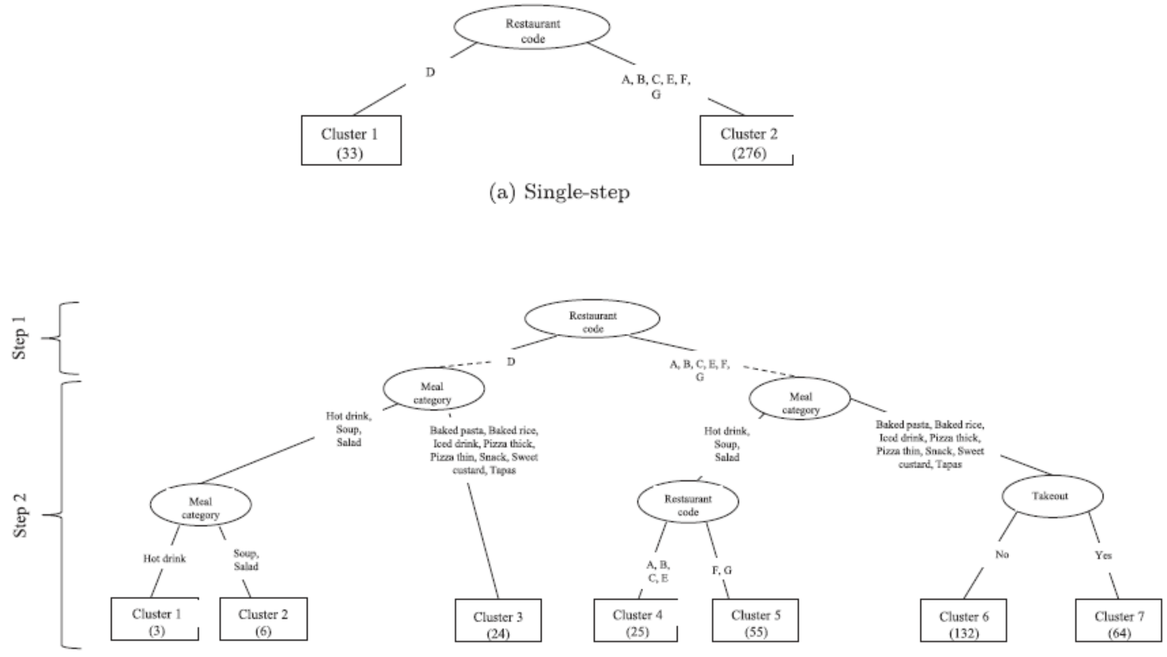


Table 2. Chow test result for comparing step 2 of two-step method terminal nodes fitted models – Restaurant sales example.

Node pairs	F-statistic	p-value
Node 1–1 – Node 1–3	2.32	0.01772
Node 1–2 – Node 1–3	5.47	6.829e-07
Node 2–1 – Node 2–3	4.59	1.294e-05
Node 2–1 – Node 2–4	7.33	8.876e-10
Node 2–2 – Node 2-3	22.65	< 2.2e-16
Node 2–2 – Node 2–4	3.30	0.00091

3.3 Displaying and interpreting the tree's clusters



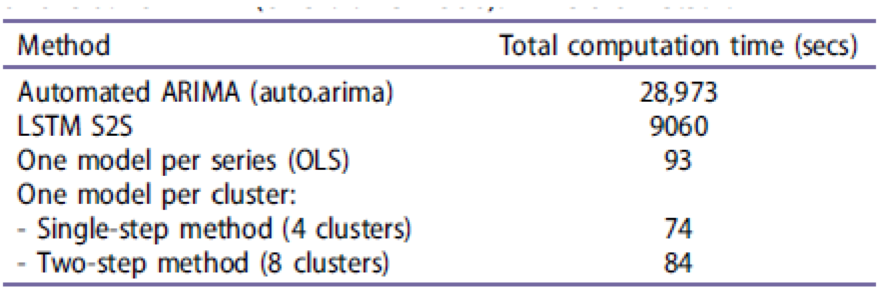
(a) Single-step

(b) Two-step

4. Using mob-based clustering for forecasting a large collection of time series.

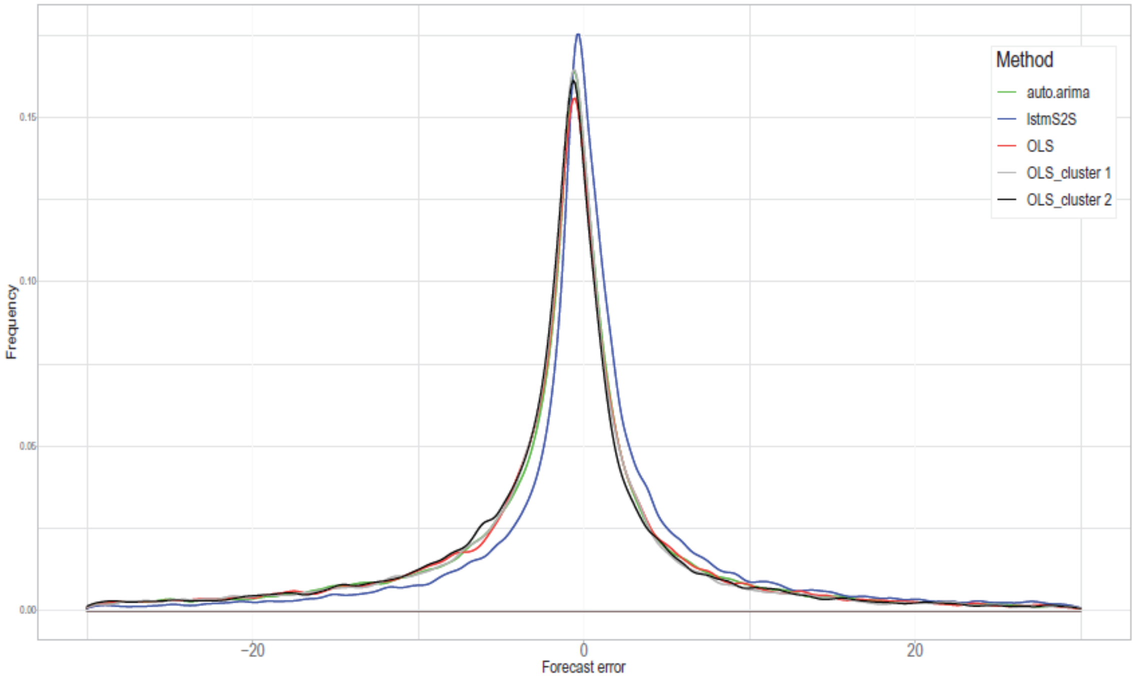
Forecasting daily Wikipedia pageviews

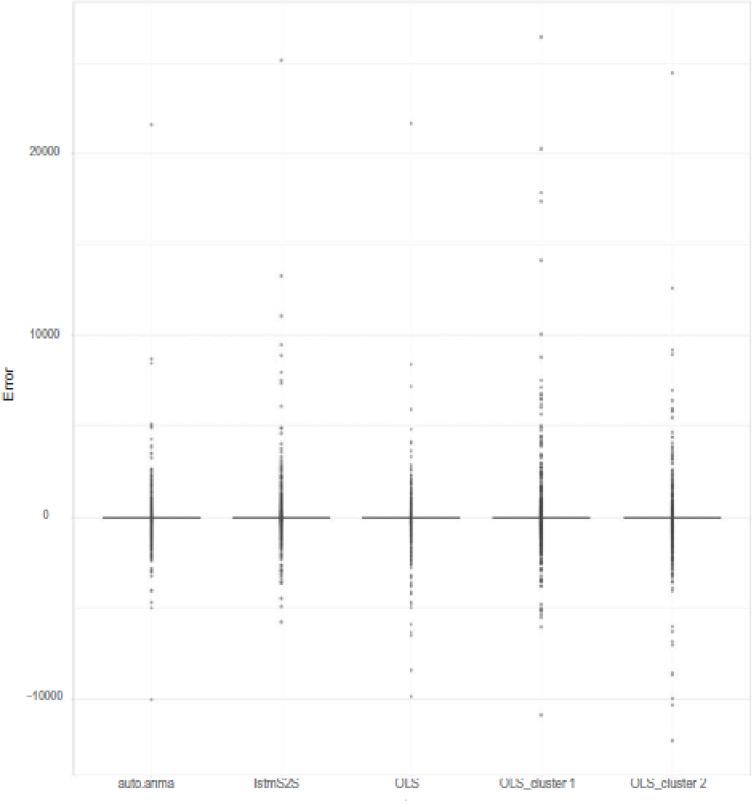
selected four domain-related attributes: Agent, Access, Purpose, Language.



Method	Total computation time (secs)
Automated ARIMA (auto.arima)	28,973
LSTM S2S	9060
One model per series (OLS)	93
One model per cluster:	
- Single-step method (4 clusters)	74
- Two-step method (8 clusters)	84

The performance for all methods appears very similar, with error density concentrated around zero. The boxplots highlight outliers. Here too the methods appear quite similar, although the OLS-Cluster methods have a slightly more outliers with large magnitude.





Methods of adding covariate variables into models

ARIMAX:

$$y_t = \beta x_t + \phi_1 y_{t-1} + \cdots + \phi_p y_{y-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \cdots - \theta_q \epsilon_{t-q}$$

Question

How to add covariate variables X into $MASE = f(F_1, F_2, \cdots, F_p)$