Meta-learning how to forecast time series

Xiaoqian Wang

June 15, 2018

Outline

- Introduction
- Literature review
- Methodology
- Application to the M competition data
- Discussion and conclusions

1. Introduction

1 Introduction

- It is common to have to regularly forecast many millions of time series. (**computational challenges**)
- We need to propose a new fast algorithm for **model selection** and time series forecasting.

- Strategies for generating a large number of forecasts:
  1. use a single forecasting method across all the time series.
  2. select an appropriate forecasting method for each time series individually.

1 Introduction

Previous studies

1. A class of models is selected in advance, and many models within that class are estimated for each time series. The model with the **smallest AICc value** is chosen and used for forecasting.

   Drawback: relies on the expert judgement to select the most appropriate class of models to use.

2. Cross-Validation (CV) : Models from different classes may be applied, and the model with the **lowest cross-validated MSE** selected.

   Drawback: increases the computation time (at least to order n^2 where n is the number of series to be forecast).

So, there is a need for a fast and scalable algorithm to automate the process of selecting models with the aim of forecasting. (**forecast-model selection**)

1 Introduction

Meta-learning framework (FFORMS)

(classification algorithm)

- Inputs: the time series features.
- Outputs: the "best" forecasting model.
- The "offline" process:

  The classifier is built using a large historical collection of time series, in advance of the forecasting task at hand.

- The "online" process:

  Generating forecasts (only involves calculating the features of a time series and using the pre-trained classifier to identify the best forecasting model).

- **Advantage:** Generating forecasts only involves the estimation of a single forecasting model, with no need to estimate large numbers of models within a class, or to carry out a computationally-intensive cross-validation procedure.

2. Literature review

2.1 Time series features

- Not level, But "**feature space**".
- The choice of the most appropriate set of features depends on both **the nature of the time series** being analysed, and **the purpose of the analysis**.
- As our main focus is forecasting, we select features which have discriminatory power in selecting a good model for forecasting.

2.1 Time series features

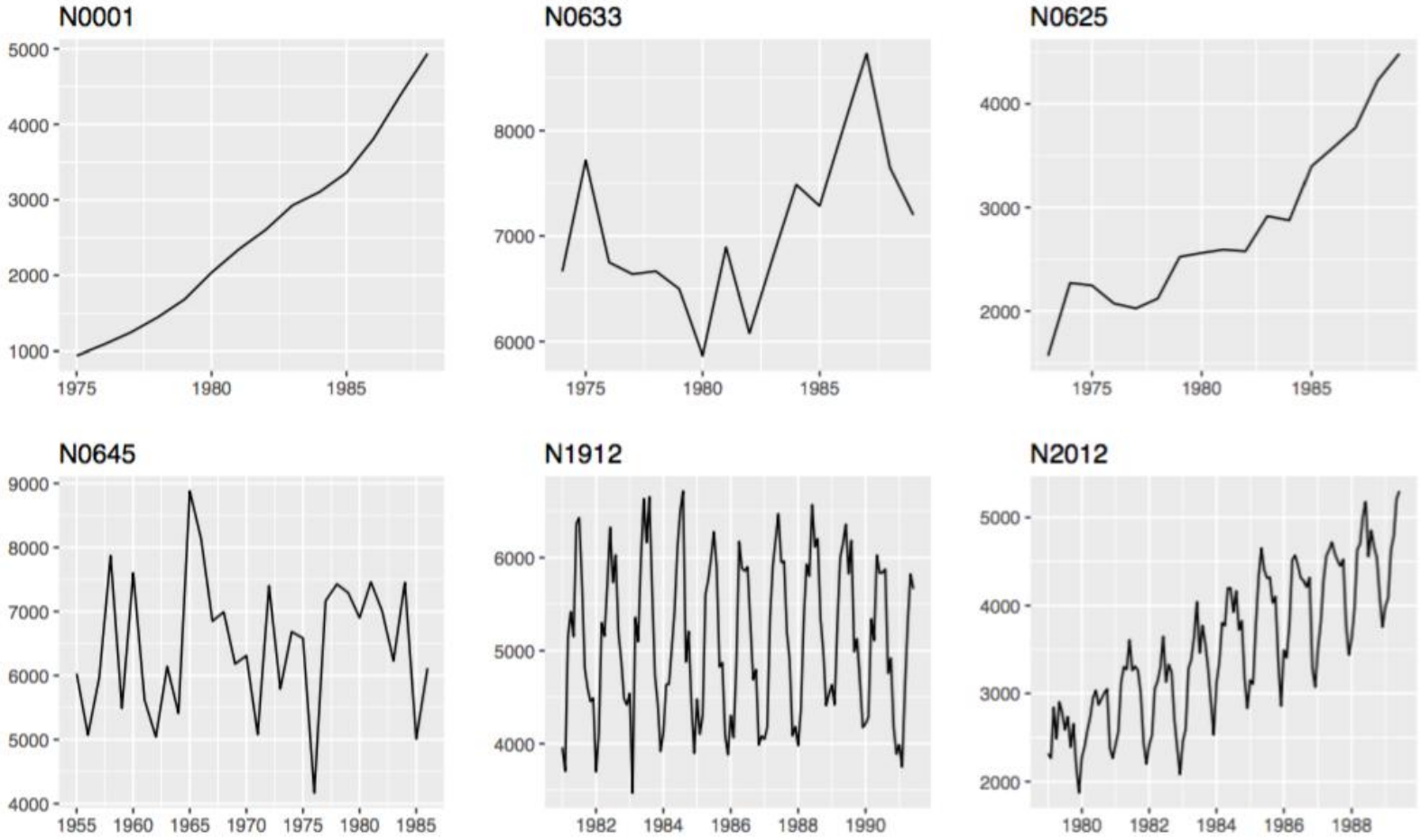


Figure 1: *Time-domain representation of time series*

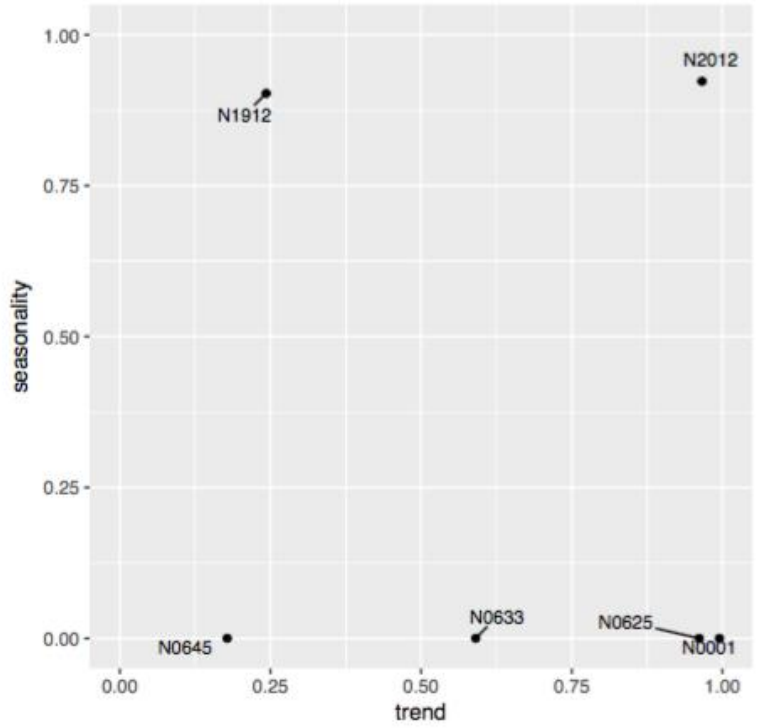


Figure 2: *Feature-based representation of time series*

2.2 What makes features useful for forecast-model selection?

- Reid (1972) points out that the performance of forecasting methods changes according to the nature of the data.
- Exploring the reasons for these variations may be useful in selecting the most appropriate model.

- Hyndman (2001), Lawrence (2001) and Armstrong (2001) argue that the characteristics of a time series may provide useful insights into which methods are most appropriate for forecasting.
- Kang, Hyndman & Smith-Miles (2017) applied principal component analysis to project a large collection of time series into a two dimensional feature space in order to visualize what makes a particular forecasting method perform well or not.
- **An appropriate set of features should reveal the characteristics of the time series that are useful in determining the best forecasting method.**

2.3 Meta-learning for algorithm selection

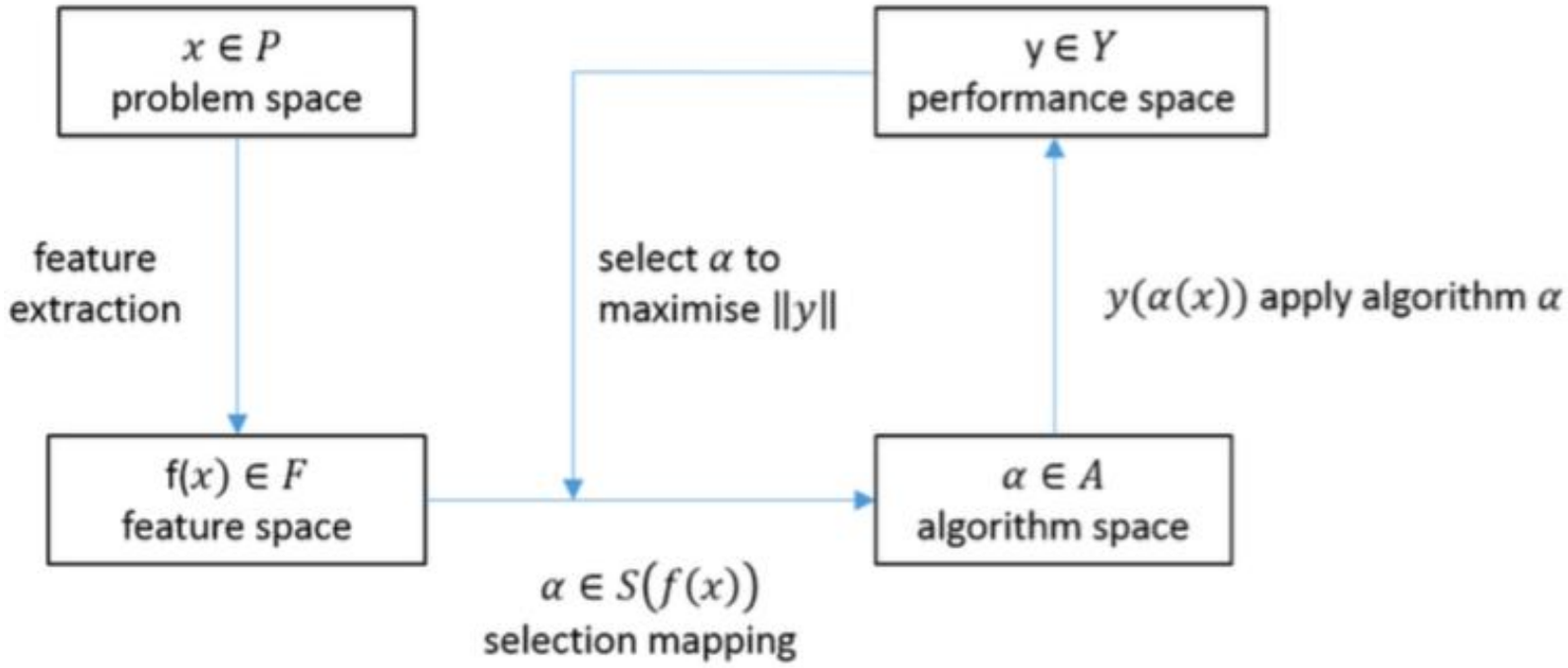
- John Rice was an early and strong proponent of the idea of meta-learning, which he called the algorithm selection problem (ASP).

- **Algorithm selection problem.**

  For a given problem instance x $\in P$, with features $f(\mathrm{x}) \in F$, find the selection mapping $\mathrm{S}(f(\mathrm{x}))$ into algorithm space A, such that the selected algorithm $\alpha \in A$ maximizes the performance mapping $y(\alpha(x)) \in Y$.

- The main challenge: identify the selection mapping S from the feature space to the algorithm space.

Rice's framework for the Algorithm Selection Problem



P: The problem space F: The feature space

A: The algorithm space Y: The performance metric

2.4 Forecast-model selection using meta-learning

- **Forecast-model selection problem.**

  For a given time series x $\in P$, with features $f(\mathrm{x}) \in F$, find the selection mapping $\mathrm{S}(f(\mathrm{x}))$ into the algorithm space A, such that the selected algorithm $\alpha \in A$ **minimizes forecast accuracy error metric** $y(\alpha(x)) \in Y$ on the test set of the time series.

- Existing methods differ with respect to the way they define the problem space (A), the features (F), the forecasting accuracy measure (Y) and the selection mapping (S).

3. Methodology

3 Methodology

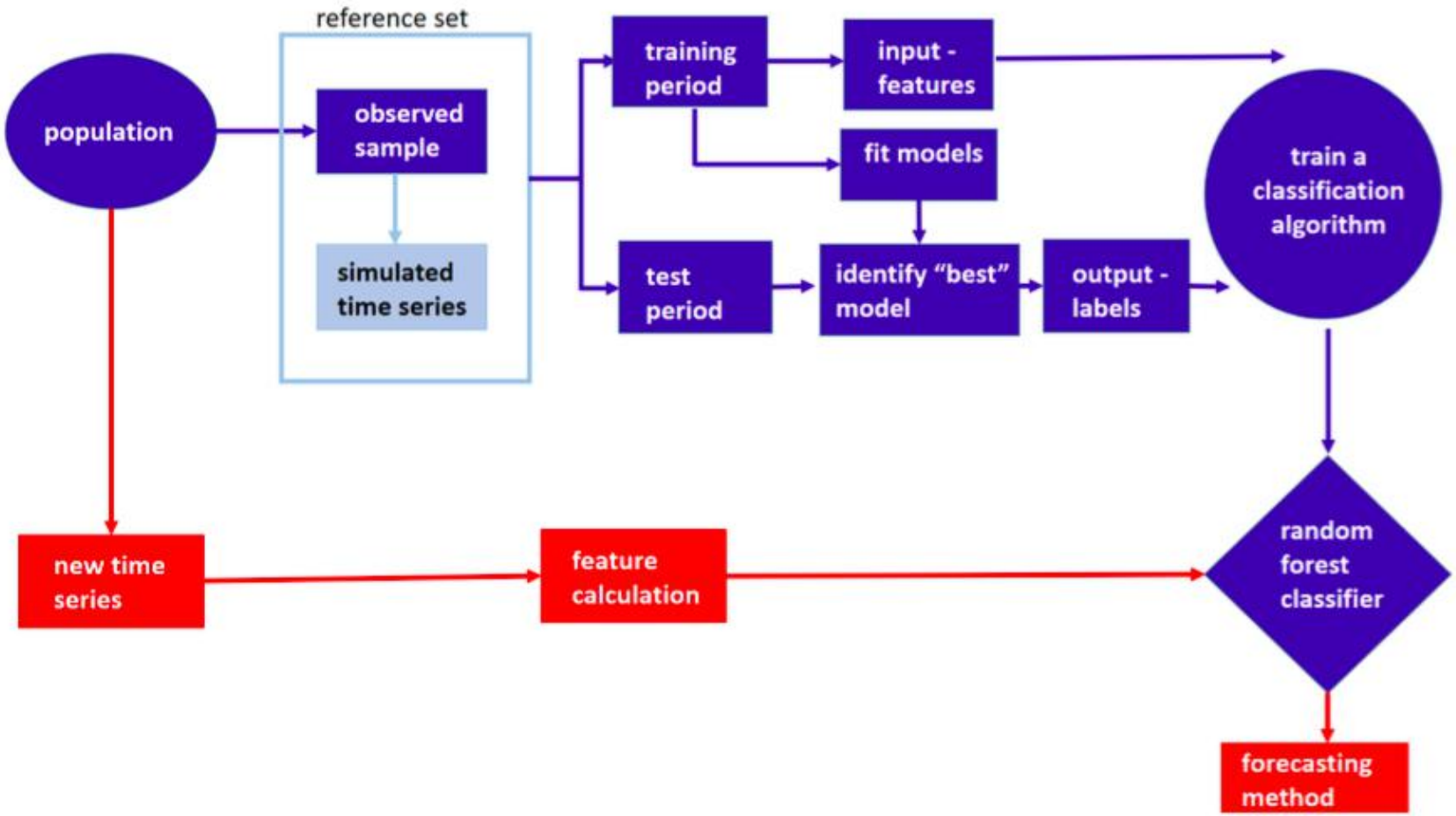


Figure 4: *FFORMS (Feature-based FORecast-Model Selection) framework. The offline phase is shown in blue and the online phase by red.*

3.1 Augmenting the observed sample with simulated time series

- In order to **train classification algorithm**, we need a large collection of time series which are similar to those we will be forecasting.
- Hence, any conclusions made from the classification framework refer only to the population from which the sample has been selected.
- **In practice, we may wish to augment the set of observed time series by simulating new time series similar to those in the assumed population**.

  1. exponential smoothing models
  2. ARIMA models

  Using the automated ets and auto.arima algorithms we identify models, based on model selection criteria (such as AICc) and simulate multiple time series from the selected models within each model class.

  Assuming the models produce data that are similar to the observed time series ensures that the simulated series are similar to those in the population.
- This is done in the **offline** phase.

3.2 Input: features

- Features should enable identification of a suitable forecast model for a given time series.
- The features used should capture the dynamic structure and consider interpretability, robustness to outliers, scale and length independence.
- We consider only features that can be computed rapidly (**online phase**).

3.3 Output: labels

- The task of the classification algorithm is to identify the “best” forecasting method for a given time series.
- The candidate models considered as labels will depend on the observed time series.
- The model with the lowest forecast error measure over the test period is deemed “best”.
- The more candidate models that are considered as labels, the more computational time is required; however the pay-off could be significant gains in forecast accuracy.

3.4 Random forest algorithm

- Let $(y_1, z_1), (y_2, z_2), \ldots, (y_N, z_N)$ represent the reference set, where input y_i is an m-vector of features, output z_i corresponds to the class label of the ith time series, and N is the number of series in the reference set.
- Each tree in the forest is grown based on a bootstrap sample of size N from the reference set.
- At each node of the tree, randomly select $f < m$ features from the full set of features. The best split is selected among those f features.

- The split which results in the most homogeneous subnodes is considered **the best split**.
- Evaluation indexes: classification error rate, **the Gini index,** cross entropy
- Each tree gives a prediction and the majority vote over all individual trees leads to the final decision.

4. Application to the M competition data

4 Application to the M competition data

	Experiment 1				Experiment 2			
	Source	Yearly	Quarterly	Monthly	Source	Yearly	Quarterly	Monthly
Observed series	M1	181	203	617	M3	645	756	1428
New series	M3	645	756	1428	M1	181	203	617

- Augment the observed sample with simulated time series.

  In each experiment, we fit ARIMA and ETS models to the full length of each series in the corresponding observed samples using the auto.arima and ets functions in the forecast package.

  For the **annual** and **quarterly** data, we simulate a further 1000 series.

  For the **monthly** time series, we simulate a further 100 series.

- The tasks:

  1. identification of the candidate forecast models as output labels.
  2. computation of features.

4.1 Identifying output labels

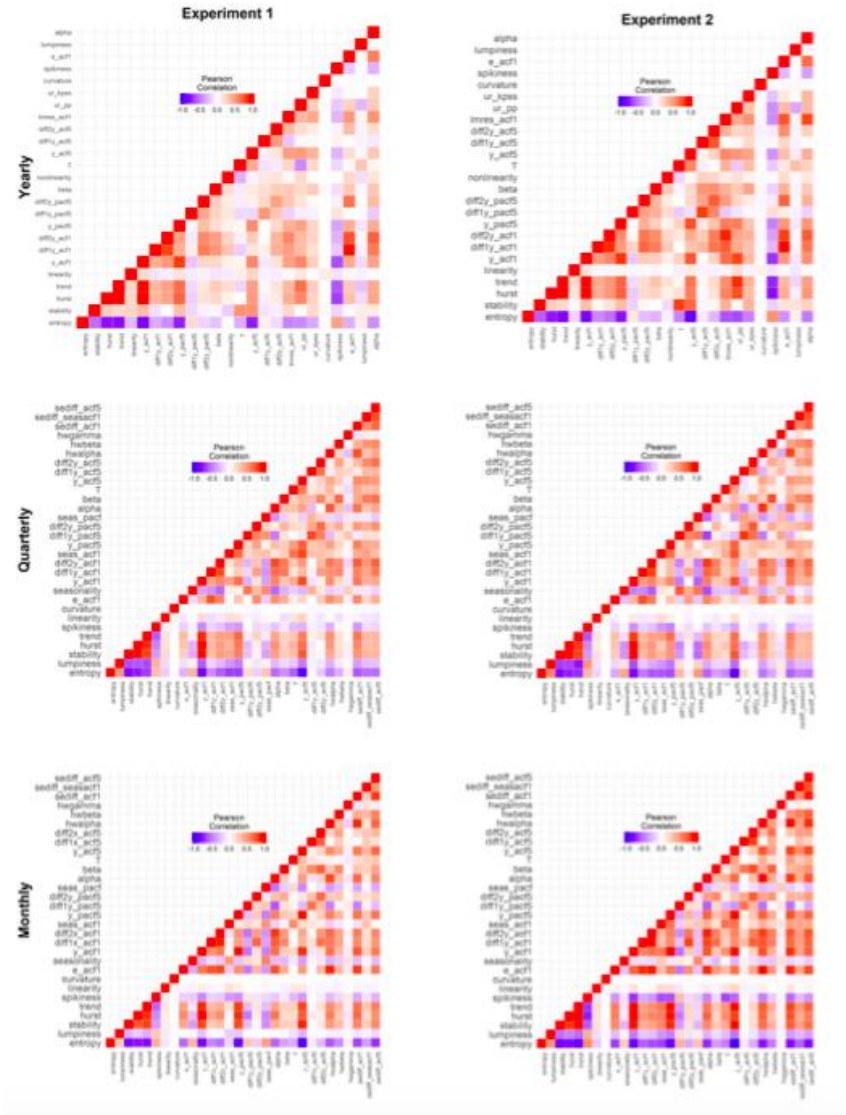
- For the annual series:
  a) White noise (WN)
  b) AR/MA/ARMA
  c) ARIMA;
  d) Random walk with drift (RWD);
  e) Random walk (RW);
  f) Theta;
  g) Exponential Smoothing Model (ETS) without trend and seasonal components;
  h) ETS with trend component and without seasonal component;
  i) ETS with damped trend component and without seasonal component;
- Further include the following labels for both quarterly and monthly data:
  j. STL-AR;
  k. ETS with trend and seasonal components;
  l. ETS with damped trend and seasonal components; smallest MASE
  m. ETS with seasonal components and without trend component;
  n. SARIMA;
  o. Seasonal naive method.

4.2 Feature computation process

- We use a set of 25 features for yearly data and a set of 30 features for seasonal data.

	Feature	Description	Non-seasonal	Seasonal
1	T	length of time series	✓	✓
2	trend	strength of trend	✓	✓
3	seasonality	strength of seasonality	-	✓
4	linearity	linearity	✓	✓
5	curvature	curvature	✓	✓
6	spikiness	spikiness	✓	✓
7	e_acf1	first ACF value of remainder series	✓	✓
8	stability	stability	✓	✓
9	lumpiness	lumpiness	✓	✓
10	entropy	spectral entropy	✓	✓
11	hurst	Hurst exponent	✓	✓
12	nonlinearity	nonlinearity	✓	✓
13	alpha	ETS(A,A,N) $\hat{\alpha}$	✓	✓
14	beta	ETS(A,A,N) $\hat{\beta}$	✓	✓
15	hwalpha	ETS(A,A,A) $\hat{\alpha}$	-	✓
16	hwbeta	ETS(A,A,A) $\hat{\beta}$	-	✓
17	hwgamma	ETS(A,A,A) $\hat{\gamma}$	-	✓
18	ur_pp	test statistic based on Phillips-Perron test	✓	-
19	ur_kpss	test statistic based on KPSS test	✓	-
20	y_acf1	first ACF value of the original series	✓	✓
21	diff1y_acf1	first ACF value of the differenced series	✓	✓
22	diff2y_acf1	first ACF value of the twice-differenced series	✓	✓
23	y_acf5	sum of squares of first 5 ACF values of original series	✓	✓
24	diff1y_acf5	sum of squares of first 5 ACF values of differenced series	✓	✓
25	diff2y_acf5	sum of squares of first 5 ACF values of twice-differenced series	✓	✓
26	seas_acf1	autocorrelation coefficient at first seasonal lag	-	✓
27	sediff_acf1	first ACF value of seasonally-differenced series	-	✓
28	sediff_seacf1	ACF value at the first seasonal lag of seasonally-differenced series	-	✓
29	sediff_acf5	sum of squares of first 5 autocorrelation coefficients of seasonally-differenced series	-	✓
30	lmres_acf1	first ACF value of residual series of linear trend model	✓	-
31	y_pacf5	sum of squares of first 5 PACF values of original series	✓	✓
32	diff1y_pacf5	sum of squares of first 5 PACF values of differenced series	✓	✓
33	diff2y_pacf5	sum of squares of first 5 PACF values of twice-differenced series	✓	✓

Correlation matrix plots for the reference sets



- The variability in the correlations reflects the diversity of the selected features. In other words, the features we have employed seem to capture different characteristics of the time series.
- The structure of the correlation matrices across the same frequencies of the two experiments seem to be fairly similar.
- This sends a strong signal that the M1 and M3 collections of time series may have similar feature spaces.

4.3 Model calibration

- The random forest (RF) algorithm is highly sensitive to **class imbalance**.
- The degree of class imbalance is reduced to some extent by augmenting the observed sample with the simulated time series.
- Three approaches to address the class imbalance:
  1. incorporating class **priors** into the RF classifier;
  2. using **the balanced RF algorithm** introduced by Chen, Liaw & Breiman (2004);
  3. re-balancing the reference set with **down-sampling**.
- We only report the results obtained by the RF built on unbalanced data (**RF-unbalanced**) and the RF with class priors (**RF-class priors; classwt**).
- ntree=1000, f=m/3
- we are not interested in accurately predicting the class, but in finding the best possible forecast model. Therefore we **report the forecast accuracy obtained from the FFORMS framework,** rather than the classification accuracy.

4.4 Summary of the main results

- We build separate RF classifiers for yearly data, quarterly data and monthly data.
- For Experiment 2, in the case of yearly and quarterly data we take a subset of the simulated time series when training the RF-unbalanced and RF-class priors, to reduce the size of the reference set.

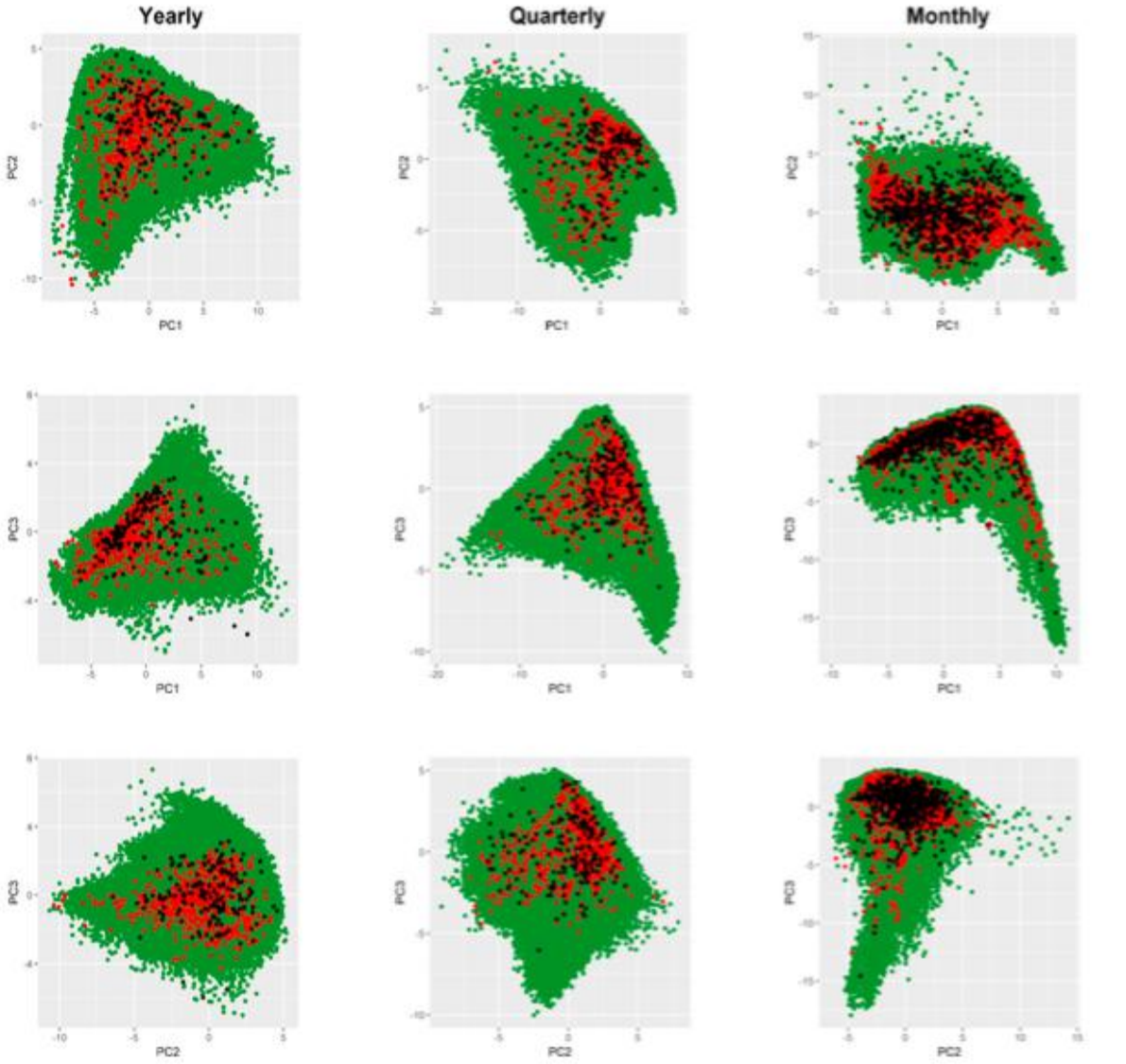


Figure 6: *Experiment 1: Distribution of time series in the PCA space. Distribution of yearly series are shown in the first column, distribution of quarterly series are shown in the second column, and the distribution of monthly series are shown in the third column. On each graph, green indicates simulated time series, black indicates observed time series, while orange denotes new time series.*

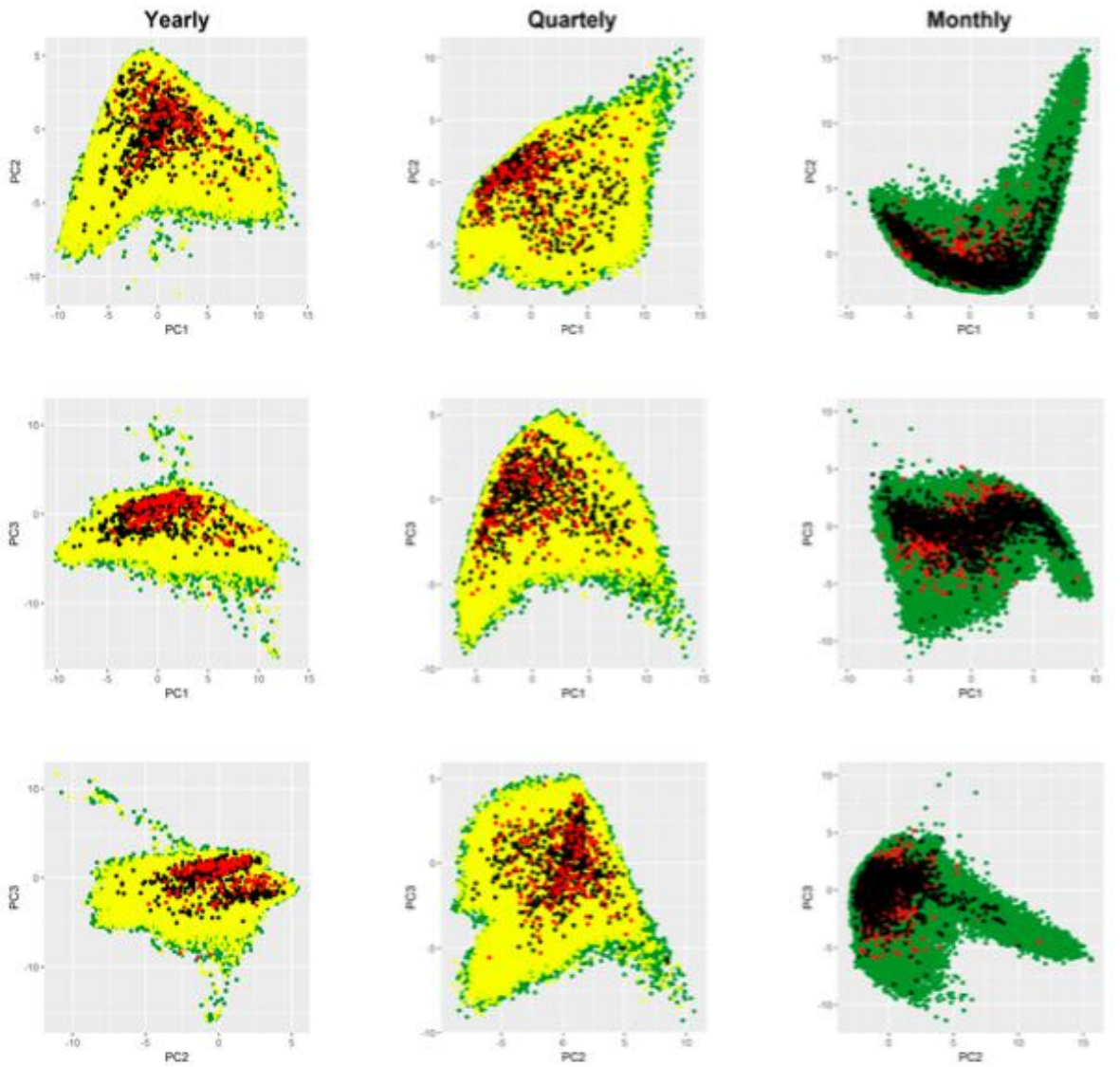


Figure 7: *Experiment 2: Distribution of time series in the PCA space. Distribution of yearly series are shown in the first column, distribution of quarterly series are shown in the second column, and the distribution of monthly series are shown in the third column. On each graph, green indicates simulated time series, yellow shows a subset of simulated time series, black indicates observed time series, while orange denotes new time series.*

Figures 6-7 show that:

1. the distribution of the observed time is very similar to that of the new time series.
2. the distribution of the simulated time clearly nests and fills in the space of the new time series.
3. all the observed time series fall within the space of all simulated data.

- This indicates that we have not reduced the feature diversity from the observed sample. By augmenting the observed series with simulated time series, we have been able to increase the diversity and evenness of the feature space in the reference sets.

Table 3: *MASE values calculated over the new series for Experiments 1 and 2.*

	Experiment 1: new series M3					Experiment 2: new series M1				
	Yearly					Yearly				
	$h=1$	$1-2$	$1-4$	$1-6$	Rank	$h=1$	$1-2$	$1-4$	$1-6$	Rank
RF-unbalanced	1.06	1.40	2.17	2.82	3.50	**0.98**	**1.40**	**2.43**	3.39	**1.50**
RF-class priors	1.04	1.38	2.15	2.79	2.50	1.01	**1.40**	**2.43**	**3.38**	**1.50**
auto.arima	1.11	1.48	2.28	2.96	5.83	1.06	1.47	2.51	3.47	3.33
ets	1.09	1.44	2.20	2.86	4.67	1.12	1.59	2.72	3.77	5.00
WN	6.54	6.91	7.48	8.07	9.00	6.38	7.08	8.59	10.01	8.00
RW	1.24	1.68	2.48	3.17	8.00	1.35	2.00	3.50	4.89	7.00
RWD	**1.03**	**1.36**	**2.05**	**2.63**	**1.00**	1.04	1.44	2.51	3.49	3.67
Theta	1.12	1.47	2.18	2.77	3.50	1.15	1.70	3.00	4.19	6.00
	Quarterly					Quarterly				
	$h=1$	$1-4$	$1-6$	$1-8$	Rank	$h=1$	$1-4$	$1-6$	$1-8$	Rank
RF-unbalanced	0.59	**0.81**	**0.97**	1.12	**2.25**	**0.74**	**1.08**	**1.35**	**1.57**	**1.00**
RF-class priors	0.59	0.82	**0.97**	1.13	3.13	0.76	1.12	1.40	1.62	2.63
auto.arima	0.59	0.85	1.02	1.19	4.75	0.78	1.17	1.50	1.74	5.25
ets	**0.56**	0.82	0.99	1.17	3.75	0.78	1.11	1.42	1.66	3.00
WN	3.25	3.59	3.70	3.87	10.00	3.97	4.27	4.45	4.64	10.00
RW	1.14	1.16	1.32	1.46	7.00	0.97	1.35	1.67	1.95	7.50
RWD	1.20	1.17	1.36	1.47	6.50	0.95	1.26	1.56	1.81	5.38
STL-AR	0.70	1.27	1.60	1.91	8.34	0.96	1.63	2.05	2.43	8.63
Theta	0.62	0.83	**0.97**	**1.11**	2.50	0.79	1.13	1.42	1.67	3.88
Snaive	1.11	1.09	1.30	1.43	6.75	1.52	1.56	1.87	2.08	7.75
	Monthly					Monthly				
	$h=1$	$1-6$	$1-12$	$1-18$	Rank	$h=1$	$1-6$	$1-12$	$1-18$	Rank
RF-unbalanced	0.60	0.68	0.76	0.87	3.22	0.61	0.76	**0.90**	**1.03**	**1.77**
RF-class priors	0.60	0.67	0.75	**0.86**	**2.00**	0.60	**0.75**	0.92	1.06	2.83
auto.arima	**0.55**	**0.64**	**0.74**	0.87	2.83	0.60	0.76	0.96	1.12	4.94
ets	**0.55**	**0.64**	**0.74**	**0.86**	2.72	**0.59**	0.76	0.93	1.07	3.44
WN	2.01	2.08	2.15	2.27	10.00	1.93	2.09	2.18	2.28	10.00
RW	0.84	0.97	1.04	1.17	8.03	1.05	1.24	1.33	1.47	7.25
RWD	0.84	0.96	1.02	1.14	6.89	1.06	1.27	1.39	1.55	8.61
STL-AR	0.64	0.81	1.04	1.27	7.89	0.63	0.91	1.17	1.39	7.38
Theta	0.58	0.67	0.77	0.89	4.22	0.61	**0.75**	0.92	1.04	2.27
Snaive	0.95	0.97	0.99	1.15	7.19	1.06	1.11	1.14	1.31	6.47

Table 3 show that:

- FFORMS meta-learning algorithm performs quite well in both experimental settings.
- This indicates that the meta-learning algorithm benefits from being trained on the larger observed sample of time series (M3 series) while forecasting a smaller new set (M1 series).

5. Discussion and conclusions

Conclusions:

- We have proposed **FFORMS algorithm** for forecast-model selection using meta-learning based on time series features.
- The method almost always performs better than common benchmark methods.
- The classifier is trained offline.
- We have also introduced a simple set of time series features that are useful in identifying the “best” forecast method for a given time series, and can be computed rapidly.

future work

- For future work, we will explore the use of other classification algorithms within the FFORMS algorithm, and test our approach on several other large collections of time series.

Thank you!