

Meta-learning how to forecast time series

Thiyanga S. Talagala
Rob J Hyndman
George Athanasopoulos

Monash University, Australia

Large collections of time series



- Forecasting demand for thousands of products across multiple warehouses.

Objective

Develop a framework that automates the selection of the most appropriate forecasting model for a given time series by using an array of **features** computed from the time series.

- **Basic idea:**

Transform a given time series $y = \{y_1, y_2, \dots, y_n\}$ to a feature vector $F = (f_1(y), f_2(y), \dots, f_p(y))'$.

Objective

Develop a framework that automates the selection of the most appropriate forecasting model for a given time series by using an array of **features** computed from the time series.

- **Basic idea:**

Transform a given time series $y = \{y_1, y_2, \dots, y_n\}$ to a feature vector $F = (f_1(y), f_2(y), \dots, f_p(y))'$.

- Examples for time series features

Objective

Develop a framework that automates the selection of the most appropriate forecasting model for a given time series by using an array of **features** computed from the time series.

- **Basic idea:**

Transform a given time series $y = \{y_1, y_2, \dots, y_n\}$ to a feature vector $F = (f_1(y), f_2(y), \dots, f_p(y))'$.

- Examples for time series features
 - strength of trend

Objective

Develop a framework that automates the selection of the most appropriate forecasting model for a given time series by using an array of **features** computed from the time series.

- **Basic idea:**

Transform a given time series $y = \{y_1, y_2, \dots, y_n\}$ to a feature vector $F = (f_1(y), f_2(y), \dots, f_p(y))'$.

- Examples for time series features

- strength of trend
- strength of seasonality

Objective

Develop a framework that automates the selection of the most appropriate forecasting model for a given time series by using an array of **features** computed from the time series.

- **Basic idea:**

Transform a given time series $y = \{y_1, y_2, \dots, y_n\}$ to a feature vector $F = (f_1(y), f_2(y), \dots, f_p(y))'$.

- Examples for time series features
 - strength of trend
 - strength of seasonality
 - lag-1 autocorrelation

Objective

Develop a framework that automates the selection of the most appropriate forecasting model for a given time series by using an array of **features** computed from the time series.

- **Basic idea:**

Transform a given time series $y = \{y_1, y_2, \dots, y_n\}$ to a feature vector $F = (f_1(y), f_2(y), \dots, f_p(y))'$.

- Examples for time series features

- strength of trend
- strength of seasonality
- lag-1 autocorrelation
- spectral entropy

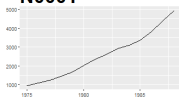
Feature-space of time series

STL-decomposition

$$Y_t = T_t + S_t + R_t$$

- strength of trend: $1 - \frac{\text{Var}(R_t)}{\text{Var}(Y_t - S_t)}$
- strength of seasonality: $1 - \frac{\text{Var}(R_t)}{\text{Var}(Y_t - T_t)}$

N0001



N0633



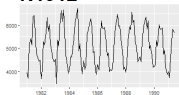
N0625



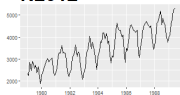
N0645



N1912



N2012

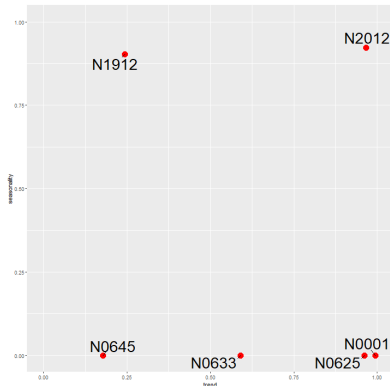
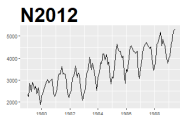
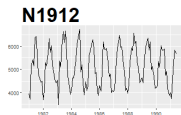
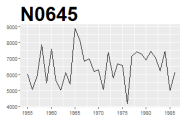
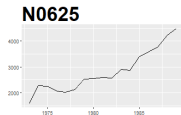
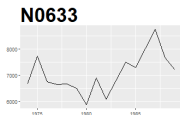
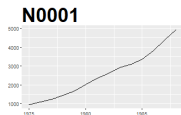


Feature-space of time series

STL-decomposition

$$Y_t = T_t + S_t + R_t$$

- strength of trend: $1 - \frac{\text{Var}(R_t)}{\text{Var}(Y_t - S_t)}$
- strength of seasonality: $1 - \frac{\text{Var}(R_t)}{\text{Var}(Y_t - T_t)}$



Time series features

- length
- strength of seasonality
- strength of trend
- linearity
- curvature
- spikiness
- stability
- lumpiness
- first ACF value of remainder series
- parameter estimates of Holt's linear trend method
- spectral entropy
- Hurst exponent
- nonlinearity
- parameter estimates of Holt-Winters' additive method
- unit root test statistics
- first ACF value of residual series of linear trend model
- ACF and PACF based features - calculated on both the raw and differenced series

FFORMS: Feature-based **FOR**ecast **M**odel **S**election

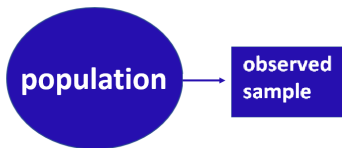
Offline: Classification algorithm is trained

Online: Use the classification algorithm to select appropriate forecast-models for new time series

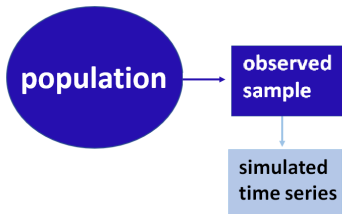


population

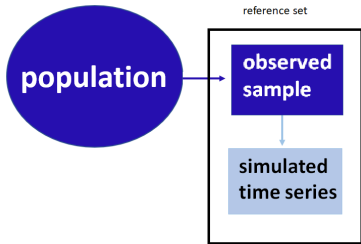
FFORMS: observed sample



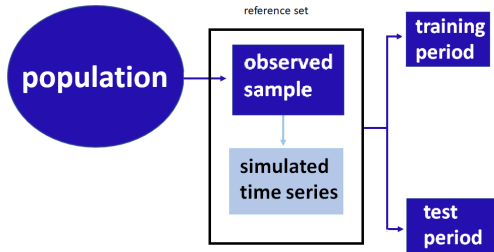
FFORMS: simulated time series



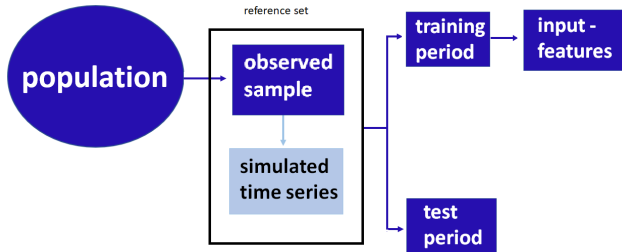
FFORMS: reference set



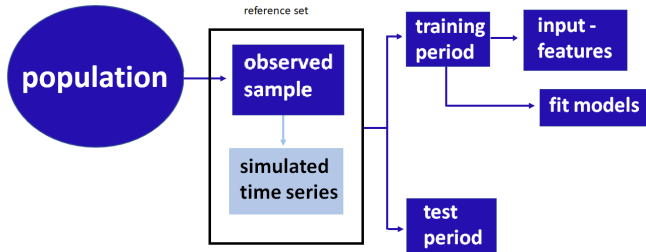
FFORMS: Meta-data



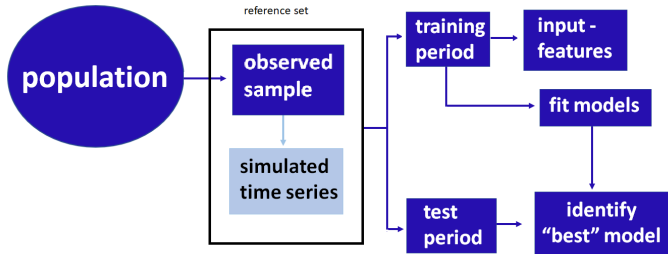
FFORMS: Meta-data



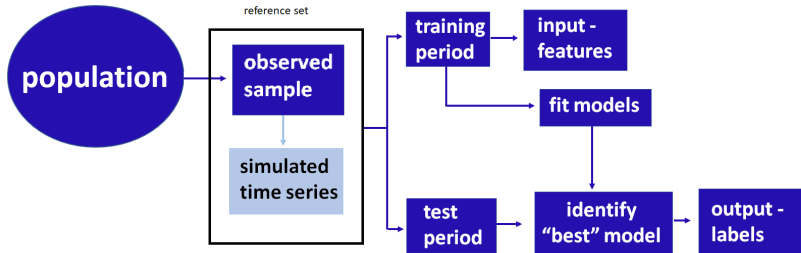
FFORMS: Meta-data



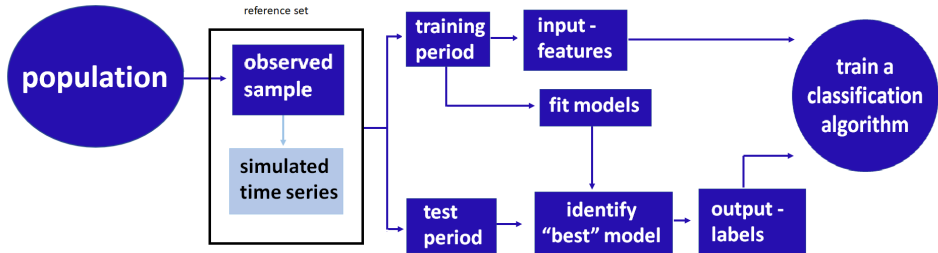
FFORMS: Meta-data



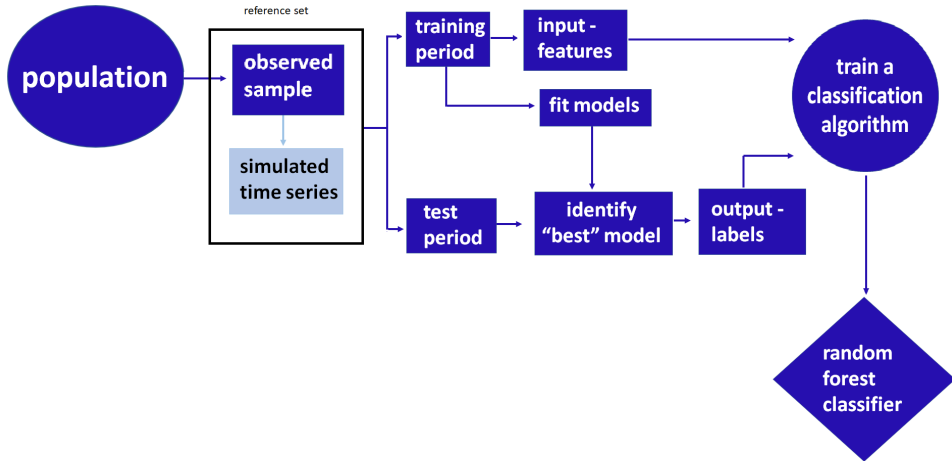
FFORMS: Meta-data



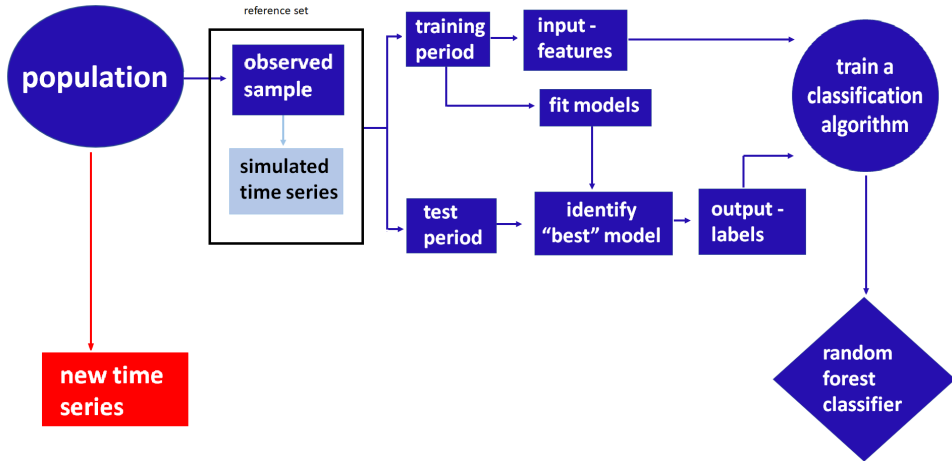
FFORMS: Meta-data



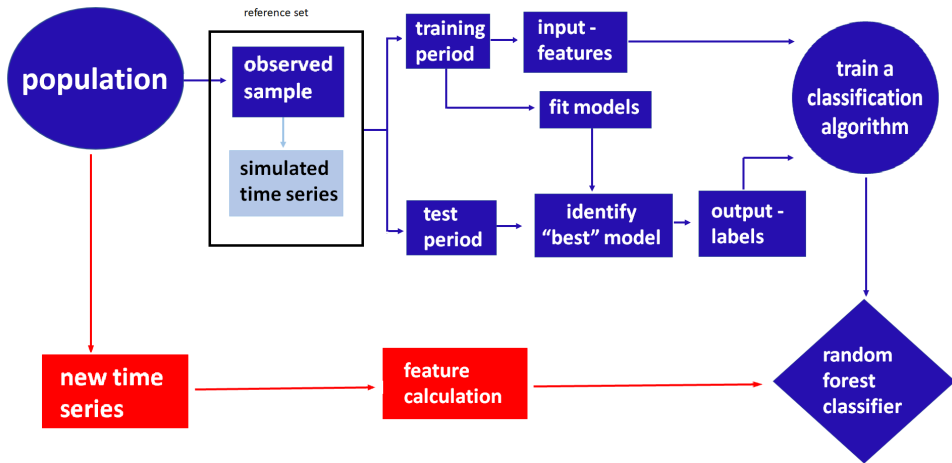
FFOMS: Random-forest classifier



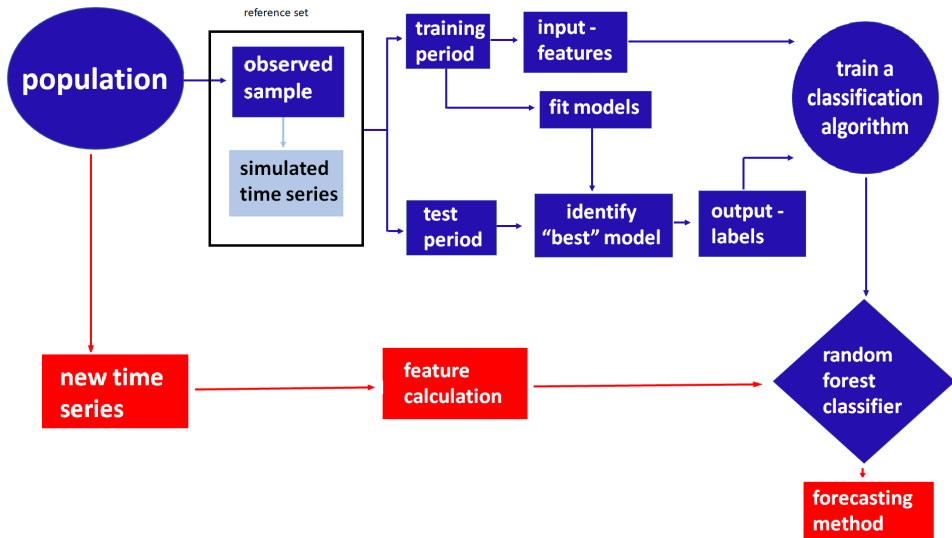
FFORMS: “online” part of the algorithm



FFORMS: “online” part of the algorithm



FFORMS: “online” part of the algorithm



Application to M competition data

- Proposed algorithm is applied to yearly, quarterly and monthly series separately
- We run two experiments for each case.

	Experiment 1				Experiment 2			
	Source	Y	Q	M	Source	Y	Q	M
Observed series	M1	181	203	617	M3	645	756	1428
New series	M3	645	756	1428	M1	181	203	617

- Proposed a novel framework for forecast-model selection using meta-learning based on time series features.

- Proposed a novel framework for forecast-model selection using meta-learning based on time series features.
- Our method almost always performs better than common benchmark methods, and better than the best-performing methods from the M3 competition.

- Proposed a novel framework for forecast-model selection using meta-learning based on time series features.
- Our method almost always performs better than common benchmark methods, and better than the best-performing methods from the M3 competition.
- The framework is general and can be applied to any large collection of time series.

- Proposed a novel framework for forecast-model selection using meta-learning based on time series features.
- Our method almost always performs better than common benchmark methods, and better than the best-performing methods from the M3 competition.
- The framework is general and can be applied to any large collection of time series.
- Advantage: Not necessary to estimate several different models for the data and undertake an empirical evaluation of their forecast accuracy on a given time series.



available at:

<https://github.com/thiyanagt/seer>