Uncertainty Estimation for High-dimensional Nonparametric Forecasting

Nuwani Palihawadana

Joint work with : Rob Hyndman, Xiaoqian Wang

1 July 2025

Nonparametric forecasting

$$y_t = f(\boldsymbol{a}_t, \boldsymbol{a}_{t-1}, \dots, \boldsymbol{a}_{t-s}, y_{t-1}, \dots, y_{t-k}) + \varepsilon_t,$$

- y_t – observed response at time t
- f – arbitrary function
- $\boldsymbol{a}_t$ – vector of exogenous variables at time t
- ε_t – stationary error with mean zero and constant variance σ^2

Nonparametric forecasting

$$y_t = f(\boldsymbol{a}_t, \boldsymbol{a}_{t-1}, \dots, \boldsymbol{a}_{t-s}, y_{t-1}, \dots, y_{t-k}) + \varepsilon_t,$$

- y_t – observed response at time t
- f – arbitrary function
- $\boldsymbol{a}_t$ – vector of exogenous variables at time t
- ε_t – stationary error with mean zero and constant variance σ^2

We also define:

- $\boldsymbol{z}_t = (y_t, \boldsymbol{x}_t)$ – observation at time t
- $\boldsymbol{x}_t = (\boldsymbol{a}_t, \boldsymbol{a}_{t-1}, \dots, \boldsymbol{a}_{t-s}, y_{t-1}, \dots, y_{t-k})$ – vector of all predictors at time t.
- $\hat{y}_t = \hat{f}(\boldsymbol{x}_t)$ – estimated model
- $e_t = y_t - \hat{y}_t$ – model residuals

Sparse Multiple Index (SMI) model

$$y_i = \beta_0 + \sum_{j=1}^{p} g_j(\boldsymbol{\alpha}_j^T \boldsymbol{x}_{ij}) + \sum_{k=1}^{d} f_k(w_{ik}) + \boldsymbol{\theta}^T \boldsymbol{u}_i + \varepsilon_i, \quad i = 1, \dots, n,$$

- y_i – univariate response
- $\boldsymbol{x}_{ij} \in \mathbb{R}^{\ell_j}$, $j = 1, \dots, p$ – p subsets of predictors entering indices
- $\boldsymbol{\alpha}_j$ – ℓ_j-dimensional vectors of index coefficients
- g_j, f_k – smooth nonlinear functions
- Additional predictors :
  - w_{ik} – nonlinear
  - $\boldsymbol{u}_i$ – linear

Sparse Multiple Index (SMI) model

$$y_i = \beta_0 + \sum_{j=1}^{p} g_j(\boldsymbol{\alpha}_j^T \boldsymbol{x}_{ij}) + \sum_{k=1}^{d} f_k(w_{ik}) + \boldsymbol{\theta}^T \boldsymbol{u}_i + \varepsilon_i, \quad i = 1, \dots, n,$$

- y_i – univariate response
- $\boldsymbol{x}_{ij} \in \mathbb{R}^{\ell_j}$, $j = 1, \dots, p$ – p subsets of predictors entering indices
- $\boldsymbol{\alpha}_j$ – ℓ_j-dimensional vectors of index coefficients
- g_j, f_k – smooth nonlinear functions
- Additional predictors :
  - w_{ik} – nonlinear
  - $\boldsymbol{u}_i$ – linear

Allow elements equal to zero in $\boldsymbol{\alpha}_j$ – "Sparse"

Sparse Multiple Index (SMI) model

$$y_i = \beta_0 + \sum_{j=1}^{p} g_j(\boldsymbol{\alpha}_j^T \boldsymbol{x}_{ij}) + \sum_{k=1}^{d} f_k(w_{ik}) + \boldsymbol{\theta}^T \boldsymbol{u}_i + \varepsilon_i, \quad i = 1, \dots, n,$$

- y_i – univariate response
- $\boldsymbol{x}_{ij} \in \mathbb{R}^{\ell_j}$, $j = 1, \dots, p$ – p subsets of predictors entering indices
- $\boldsymbol{\alpha}_j$ – ℓ_j-dimensional vectors of index coefficients
- g_j, f_k – smooth nonlinear functions
- Additional predictors :
  - w_{ik} – nonlinear
  - $\boldsymbol{u}_i$ – linear

Both "p" and the predictor grouping among indices are unknown.

Overlapping of predictors among indices is not allowed.

Other models

- Nonparametric additive model with backward elimination (Backward):
  - No linear combinations (indices)
  - Fully additive

Other models

- Nonparametric additive model with backward elimination (Backward):
  - No linear combinations (indices)
  - Fully additive
- Groupwise Additive Index Model (GAIM):
  - Predefined predictor groups
  - No overlapping predictors among groups

Other models

- Nonparametric additive model with backward elimination (Backward):
  - No linear combinations (indices)
  - Fully additive
- Groupwise Additive Index Model (GAIM):
  - Predefined predictor groups
  - No overlapping predictors among groups
- Projection Pursuit Regression model (PPR):
  - All predictors enter all indices

Forecast uncertainty

- Uncertainty of a forecast $\rightarrow$ **Prediction Interval (PI)**

Forecast uncertainty

- Uncertainty of a forecast $\rightarrow$ **Prediction Interval (PI)**
- Theoretical $100(1-\alpha)\%$ prediction interval:

$$\hat{y}_{t+h|t} \pm z_{\alpha/2} \times \hat{\sigma}_h,$$

where

- y – time series $y_1, \ldots, y_T$
- $\hat{y}_{t+h|t}$ – h-step-ahead point forecast for y_{t+h} given observations up to t
- $z_{\alpha/2}$ – $\alpha/2$ quantile of standard normal distribution
- $\hat{\sigma}_h$ – estimate of std. deviation of h-step forecast distribution

Forecast uncertainty

- Uncertainty of a forecast $\rightarrow$ **Prediction Interval (PI)**
- Theoretical $100(1-\alpha)$% prediction interval:

$$\hat{y}_{t+h|t} \pm z_{\alpha/2} \times \hat{\sigma}_h,$$

where

  - y – time series $y_1, \ldots, y_T$
  - $\hat{y}_{t+h|t}$ – h-step-ahead point forecast for y_{t+h} given observations up to t
  - $z_{\alpha/2}$ – $\alpha/2$ quantile of standard normal distribution
  - $\hat{\sigma}_h$ – estimate of std. deviation of h-step forecast distribution
- Main issue:
  - Difficult to analytically calculate h-step forecast variances for $h > 1$

Block bootstrap (BB)

Block bootstrapping

- Randomly resample blocks from the historical model residuals, and join together
- Retains serial correlation in the data

BB with time series cross-validation Forecasting

Step 1 : Split the data into an initial training window, $\{\boldsymbol{z}_1, \dots, \boldsymbol{z}_{tr}\}$, and an initial test window, $\{\boldsymbol{z}_{tr+1}, \dots, \boldsymbol{z}_{tr+H}\}$.

Step 2 : Train forecasting model on training window.

Step 3 : Obtain the series of in-sample residuals, $e_1, \dots, e_{tr}$.

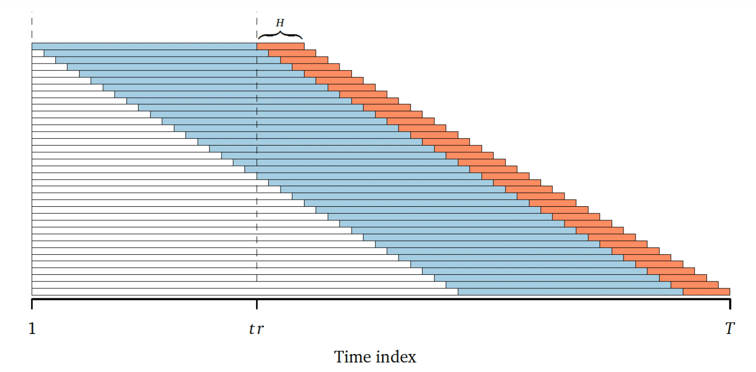
Step 4 : Perform BB by using residuals series; generate several bootstrapped series.

Step 5 : Obtain H-step-ahead simulated future values, $\hat{y}_{tr+1|tr}, \dots, \hat{y}_{tr+H|tr}$, for each bootstrapped series.

Step 6 : Calculate quantiles of the sets of simulated future values at each horizon.

Step 7 : Iteratively roll training window forward by one observation; repeat steps 2–6 until prediction intervals have been constructed for the entire test set.

BB with time series cross-validation Forecasting



- *tr* : length of training window
- *H* : forecast horizon

Conformal prediction (CP)

Conformal prediction

- A distribution-free approach (Vovk et al. 2005)
- Relies only on the assumption of **exchangeability of data**
- Provides theoretical coverage guarantees

Conformal prediction (CP)

Conformal prediction

- A distribution-free approach (Vovk et al. 2005)
- Relies only on the assumption of **exchangeability of data**
- Provides theoretical coverage guarantees

Split Conformal Prediction (SCP) :

- A holdout method for generating prediction intervals
  - **Training set** – train forecasting model
  - **Calibration set** – calculate forecast errors (*nonconformity scores*)
  - **Test set** – obtain prediction intervals

CP methods for non-exchangeable data

Weighted Split Conformal Prediction (WSCP) (Barber et al. 2023) :

- Weighting quantiles using fixed (data-independent) weights

CP methods for non-exchangeable data

Weighted Split Conformal Prediction (WSCP) (Barber et al. 2023) :

- Weighting quantiles using fixed (data-independent) weights

Adaptive Conformal Prediction (ACP) (Gibbs & Candès 2021) :

- Update nominal α based on achieved coverage

CP with time series cross-validation Forecasting

Step 1 : Split data into an initial training window, $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{tr}\}$, an initial calibration window, $\{\boldsymbol{z}_{tr+1}, \ldots, \boldsymbol{z}_{tr+tc}\}$, and an initial test window, $\{\boldsymbol{z}_{tr+tc+1}, \ldots, \boldsymbol{z}_{tr+tc+H}\}$.

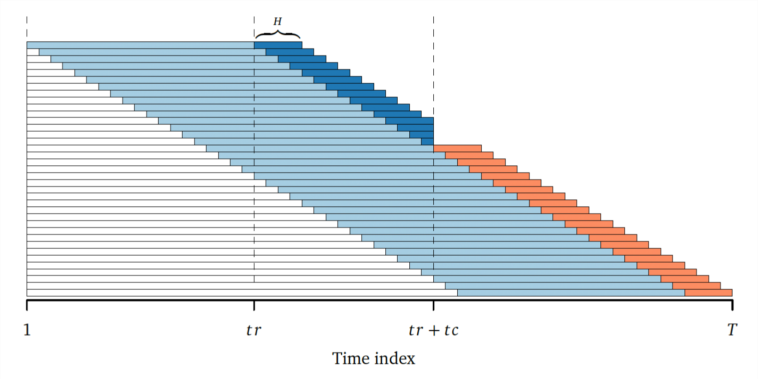
Step 2 : Estimate forecasting model on initial training window. Obtain H-step-ahead forecasts, $\hat{y}_{tr+h|tr}$, and forecast errors $s_{tr+h|tr} = y_{tr+h} - \hat{y}_{tr+h|tr}$, for $h = 1, \ldots, H$.

Step 3 : Perform cross-validation forecasting, while repeating step 2, until forecast errors are obtained for entire initial calibration window.

Step 4 : Apply the conformal prediction method of interest: SCP, WSCP, or ACP, to compute H-step prediction intervals.

Step 5 : Iteratively roll training window and calibration window forward by one observation; repeat steps 2–4 until prediction intervals are obtained for the entire test set.

CP with time series cross-validation Forecasting



- *tr* : length of training window
- *tc* : length of calibration window
- *H* : forecast horizon

Conformal bootstrap

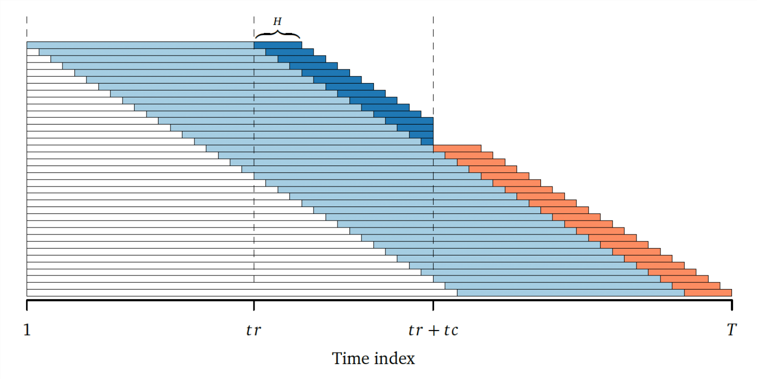
- **A natural integration of BB and SCP**

Conformal bootstrap (CB)

Conformal bootstrap

- **A natural integration of BB and SCP**
- **Exploits the strengths of both the methods**
  - Preserves temporal dependencies inherent in multi-step forecasts
  - Accounts for additional uncertainty brought into the process by ex-ante forecasting

CB with time series cross-validation Forecasting



- tr : length of training window
- tc : length of calibration window
- H : forecast horizon

CB with time series cross-validation Forecasting

Step 1 : Split data into an initial training window, $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{tr}\}$, an initial calibration window, $\{\boldsymbol{z}_{tr+1}, \ldots, \boldsymbol{z}_{tr+tc}\}$, and an initial test window, $\{\boldsymbol{z}_{tr+tc+1}, \ldots, \boldsymbol{z}_{tr+tc+H}\}$.

CB with time series cross-validation Forecasting

Step 1 : Split data into an initial training window, $\{\boldsymbol{z}_1, \dots, \boldsymbol{z}_{tr}\}$, an initial calibration window, $\{\boldsymbol{z}_{tr+1}, \dots, \boldsymbol{z}_{tr+tc}\}$, and an initial test window, $\{\boldsymbol{z}_{tr+tc+1}, \dots, \boldsymbol{z}_{tr+tc+H}\}$.

Step 2 : Estimate forecasting model on initial training window. Obtain H-step-ahead forecasts, $\hat{y}_{tr+h|tr}$, and forecast errors $s_{tr+h|tr} = y_{tr+h} - \hat{y}_{tr+h|tr}$, for $h = 1, \dots, H$.

CB with time series cross-validation Forecasting

Step 1 : Split data into an initial training window, $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{tr}\}$, an initial calibration window, $\{\boldsymbol{z}_{tr+1}, \ldots, \boldsymbol{z}_{tr+tc}\}$, and an initial test window, $\{\boldsymbol{z}_{tr+tc+1}, \ldots, \boldsymbol{z}_{tr+tc+H}\}$.

Step 2 : Estimate forecasting model on initial training window. Obtain H-step-ahead forecasts, $\hat{y}_{tr+h|tr}$, and forecast errors $s_{tr+h|tr} = y_{tr+h} - \hat{y}_{tr+h|tr}$, for $h = 1, \ldots, H$.

Step 3 : Perform cross-validation forecasting, while repeating step 2, until forecast errors are obtained for entire initial calibration window.

CB with time series cross-validation Forecasting

Step 1 : Split data into an initial training window, $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{tr}\}$, an initial calibration window, $\{\boldsymbol{z}_{tr+1}, \ldots, \boldsymbol{z}_{tr+tc}\}$, and an initial test window, $\{\boldsymbol{z}_{tr+tc+1}, \ldots, \boldsymbol{z}_{tr+tc+H}\}$.

Step 2 : Estimate forecasting model on initial training window. Obtain H-step-ahead forecasts, $\hat{y}_{tr+h|tr}$, and forecast errors $s_{tr+h|tr} = y_{tr+h} - \hat{y}_{tr+h|tr}$, for $h = 1, \ldots, H$.

Step 3 : Perform cross-validation forecasting, while repeating step 2, until forecast errors are obtained for entire initial calibration window.

Step 4 : Use the model estimated on the most recent training window to generate H-step-ahead simulated future values.

- Autoregressive: block bootstrap 1-step forecast errors; block size = H.
- Non-autoregressive: bootstrap multi-step forecast errors at each time point as a whole.

CB with time series cross-validation Forecasting

Step 1 : Split data into an initial training window, $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{tr}\}$, an initial calibration window, $\{\boldsymbol{z}_{tr+1}, \ldots, \boldsymbol{z}_{tr+tc}\}$, and an initial test window, $\{\boldsymbol{z}_{tr+tc+1}, \ldots, \boldsymbol{z}_{tr+tc+H}\}$.

Step 2 : Estimate forecasting model on initial training window. Obtain H-step-ahead forecasts, $\hat{y}_{tr+h|tr}$, and forecast errors $s_{tr+h|tr} = y_{tr+h} - \hat{y}_{tr+h|tr}$, for $h = 1, \ldots, H$.

Step 3 : Perform cross-validation forecasting, while repeating step 2, until forecast errors are obtained for entire initial calibration window.

Step 4 : Use the model estimated on the most recent training window to generate H-step-ahead simulated future values.

- Autoregressive: block bootstrap 1-step forecast errors; block size = H.
- Non-autoregressive: bootstrap multi-step forecast errors at each time point as a whole.

Step 5 : Calculate quantiles of the sets of simulated future values at each horizon.

CB with time series cross-validation Forecasting

Step 1 : Split data into an initial training window, $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{tr}\}$, an initial calibration window, $\{\boldsymbol{z}_{tr+1}, \ldots, \boldsymbol{z}_{tr+tc}\}$, and an initial test window, $\{\boldsymbol{z}_{tr+tc+1}, \ldots, \boldsymbol{z}_{tr+tc+H}\}$.

Step 2 : Estimate forecasting model on initial training window. Obtain H-step-ahead forecasts, $\hat{y}_{tr+h|tr}$, and forecast errors $s_{tr+h|tr} = y_{tr+h} - \hat{y}_{tr+h|tr}$, for $h = 1, \ldots, H$.

Step 3 : Perform cross-validation forecasting, while repeating step 2, until forecast errors are obtained for entire initial calibration window.

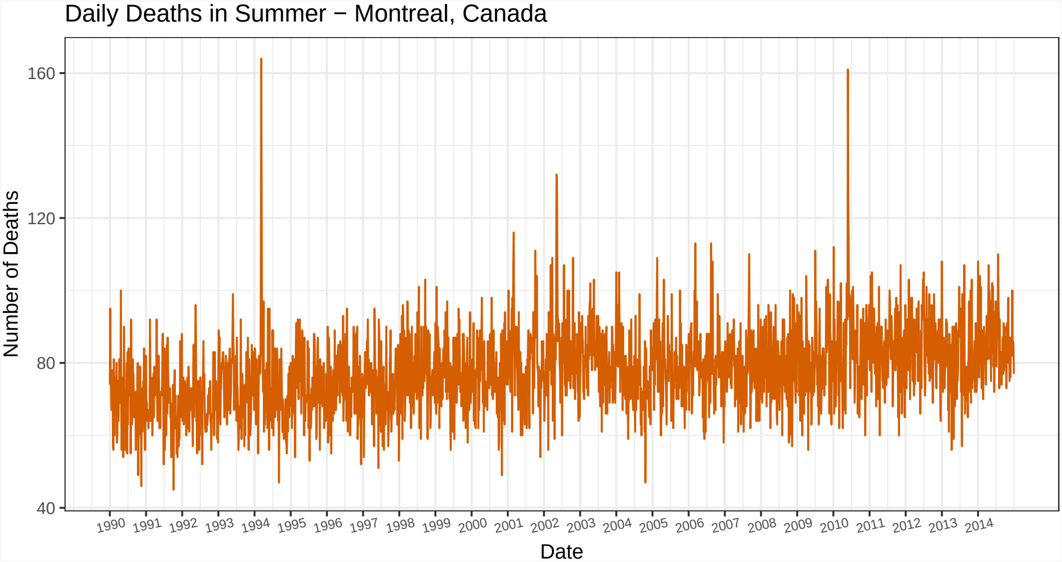
Step 4 : Use the model estimated on the most recent training window to generate H-step-ahead simulated future values.

- Autoregressive: block bootstrap 1-step forecast errors; block size = H.
- Non-autoregressive: bootstrap multi-step forecast errors at each time point as a whole.

Step 5 : Calculate quantiles of the sets of simulated future values at each horizon.

Step 6 : Iteratively roll training window forward by one observation; repeat steps 2–5 until prediction intervals have been constructed for the entire test set.

Forecasting heat exposure-related daily mortality



Forecasting heat exposure-related daily mortality

Data

- **Response: Daily deaths in Summer** – 1990 to 2014 – Montreal, Canada
- **Index Variables:**
  - Death lags
  - Max temperature lags
  - Min temperature lags
  - Vapor pressure lags
- **Nonlinear:** DOS (day of the season), Year

Forecasting heat exposure-related daily mortality

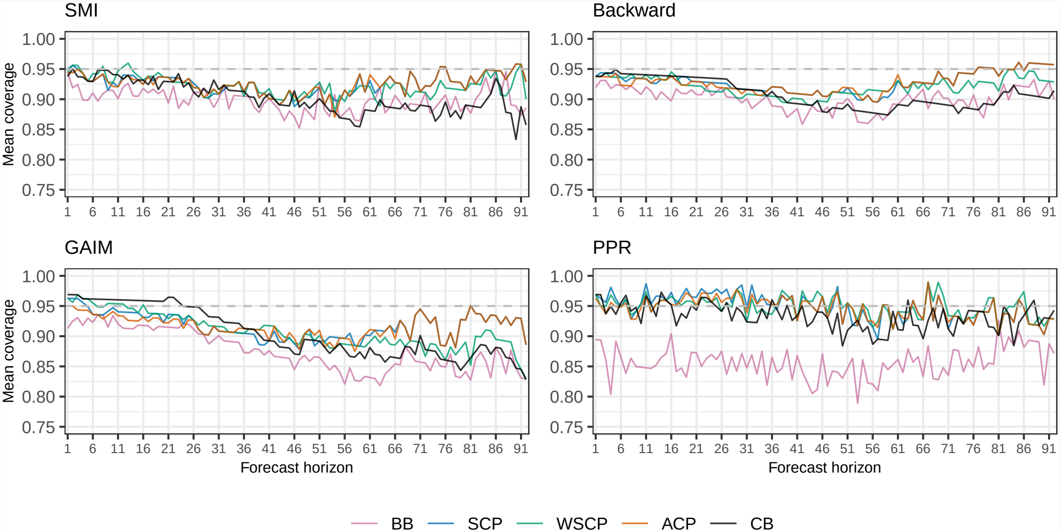
Data

- **Response: Daily deaths in Summer** – 1990 to 2014 – Montreal, Canada
- **Index Variables:**
  - Death lags
  - Max temperature lags
  - Min temperature lags
  - Vapor pressure lags
- **Nonlinear:** DOS (day of the season), Year

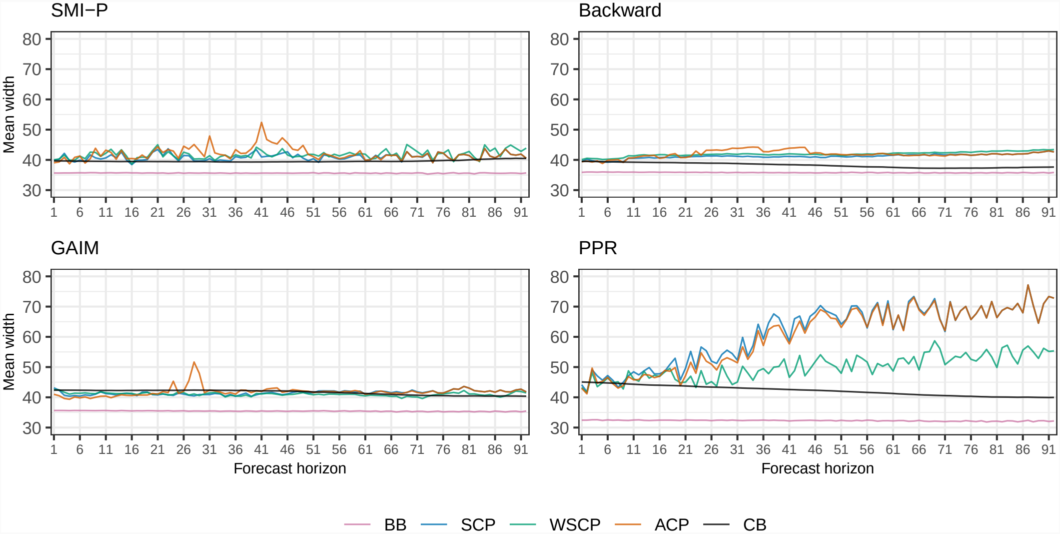
Data split

- ***tr*** - training window : 1748
- ***tc*** - calibration window : 300
- ***H*** - forecast horizon : 92

Mean coverage of 95% prediction intervals



Mean width of 95% prediction intervals



Conclusion

Summary of results:

- **Block bootstrap** prediction intervals were **narrow** and resulted in **under-coverage**

Conclusion

Summary of results:

- **Block bootstrap** prediction intervals were **narrow** and resulted in **under-coverage**
- **Conformal prediction** methods **generally performed well** across models and forecast horizons

Conclusion

Summary of results:

- **Block bootstrap** prediction intervals were **narrow** and resulted in **under-coverage**
- **Conformal prediction** methods **generally performed well** across models and forecast horizons
- **Conformal bootstrap** method **performed well in shorter horizons**; performance decreased in longer horizons

Conclusion

Summary of results:

- **Block bootstrap** prediction intervals were **narrow** and resulted in **under-coverage**
- **Conformal prediction** methods **generally performed well** across models and forecast horizons
- **Conformal bootstrap** method **performed well in shorter horizons**; performance decreased in longer horizons
  - Using only 1-step errors

Conclusion

Summary of results:

- **Block bootstrap** prediction intervals were **narrow** and resulted in **under-coverage**
- **Conformal prediction** methods **generally performed well** across models and forecast horizons
- **Conformal bootstrap** method **performed well in shorter horizons**; performance decreased in longer horizons
  - Using only 1-step errors
  - Predictor truncation when simulating sample paths

Conclusion

Summary of results:

- **Block bootstrap** prediction intervals were **narrow** and resulted in **under-coverage**
- **Conformal prediction** methods **generally performed well** across models and forecast horizons
- **Conformal bootstrap** method **performed well in shorter horizons**; performance decreased in longer horizons
  - Using only 1-step errors
  - Predictor truncation when simulating sample paths
- **Coverage performance** is influenced by: **data characteristics, forecasting model**, and **forecast horizon**

R Package - smimodel

- Block bootstrap
  - **bb_cvforecast()**
- Conformal bootstrap
  - **cb_cvforecast()**

github.com/nuwani-palihawadana/smimodel

Find me :

nuwanipalihawadana.netlify.app

in/nuwani-palihawadana

@nuwani-palihawadana

nuwani.kodikarapalihawadana@monash.edu