Feature-based **FOR**ecast **M**odel **A**veraging

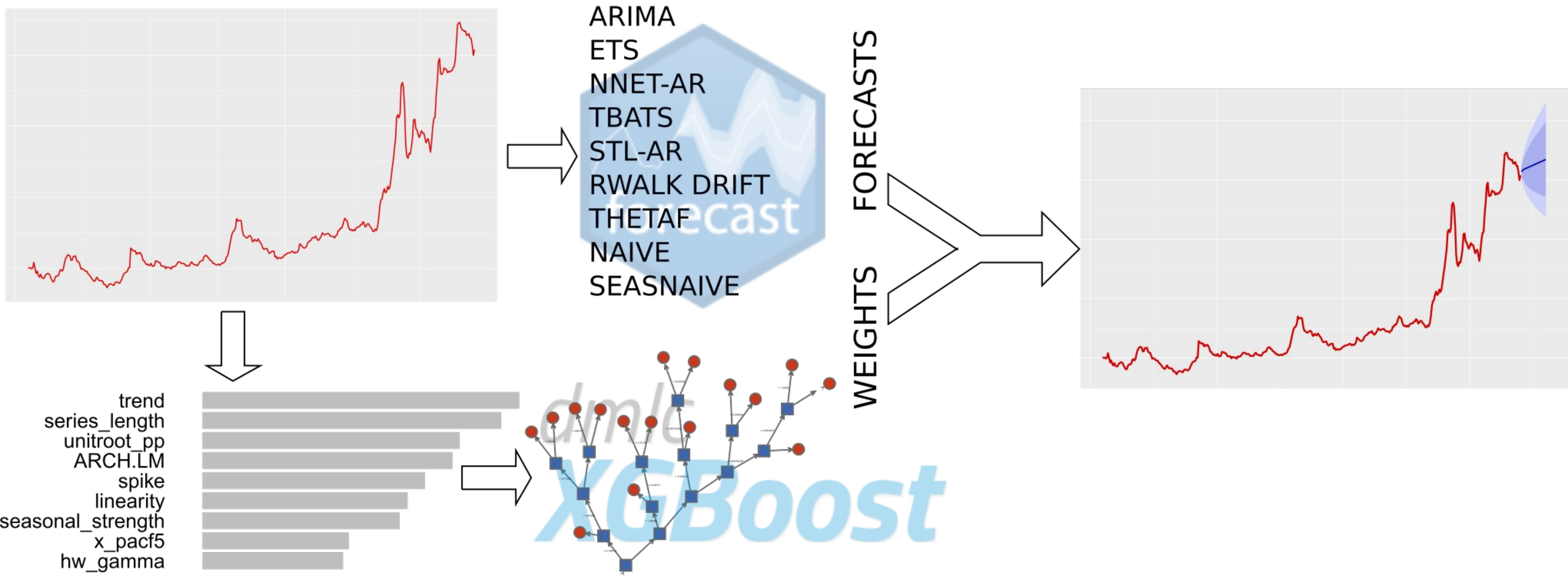
Combining Statistical and ML Methods

Pablo Montero-Manso @ Univ. A Coruña, Spain
Thiyanga Talagala @ Monash Univ., Australia
Rob J. Hyndman @ Monash Univ., Australia
George Athanasopoulos @ Monash Univ., Australia

A weighted average of individual forecast methods

Per-series weights are generated by a model, trained on a large set of time series

Forecasting with FFORMA



A fixed *pool* of 9 individual forecast methods

- ARIMA
- TBATS
- RANDOM WALK with DRIFT
- ETS
- THETA
- STL Decomposition with AR residuals
- NAIVE
- SEASONAL NAIVE
- NEURAL NETWORK AUTOREGRESSIVE

What guided the creation of this *pool* of methods?

- **Simplicity:** Methods already implemented in the **forecast** R package
  - Automatically fit models with **default** parameters
- Discarded some computationally expensive, such as bagged versions of ARIMA and ETS
- Added **THETA** due to past performance

Weights for the combination (1)

- A **decision tree model** generates the weights.

dmlc XGBoost

- The model is trained on a *Reference Dataset*, a **Temporal Holdout** version **of the M4 Dataset**

Weights for the combination (2)

- For each series in the *Reference Dataset:*
  - Extract *features* from the series.
    - Autocorrelations, length, frequency…
  - Calculate forecast error of each method in the *pool.*

- The **input** of the model are the *features.*

Weights for the combination (3)

- The model produces weights or “probabilities”.
- The average forecast error made by selecting the method in the *pool* following this probabilities is the function to be minimized.
  - This is not the same as minimizing the weighted average of the forecasts in the *pool.*

Why these *features*?

- Hyndman, Wang and Laptev. **“Large scale unusual time series detection”** (2015).
- Kang, Hyndman & Smith-Miles. **“Visualising forecasting algorithm performance using time series instance spaces”** (2017).
- Talagala, Hyndman and Athanasopoulos. “**Meta-learning how to forecast time series”** (2018).

- Implemented in the **tsfeatures** R package

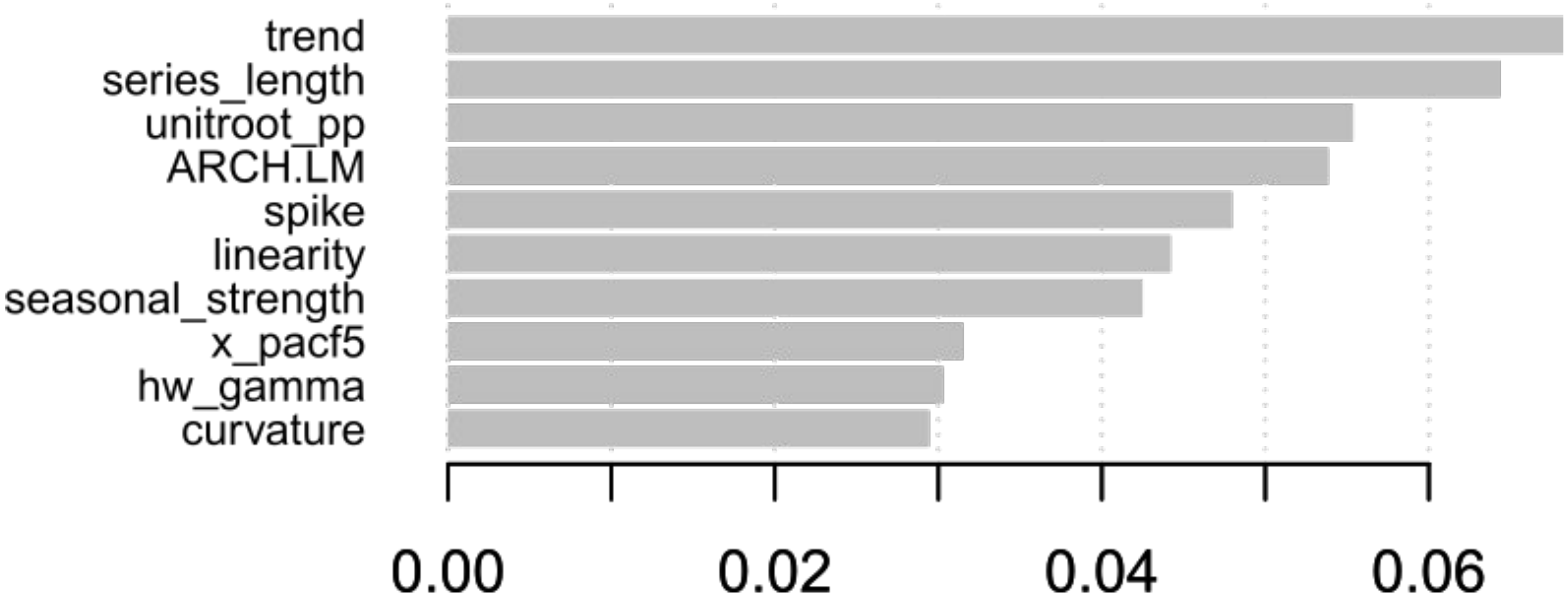
Results

2nd according to average OWA: 0.838

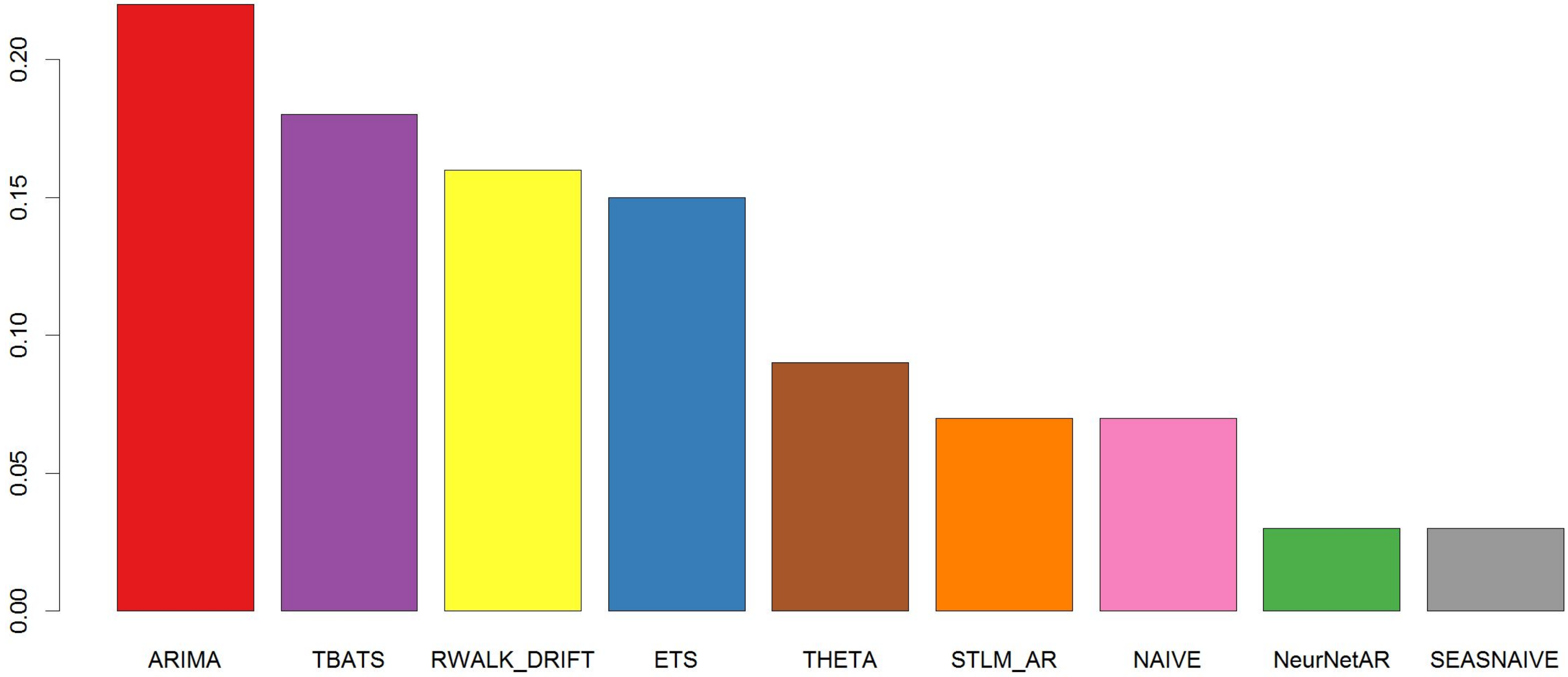
1st: 0.821

3rd: 0.841

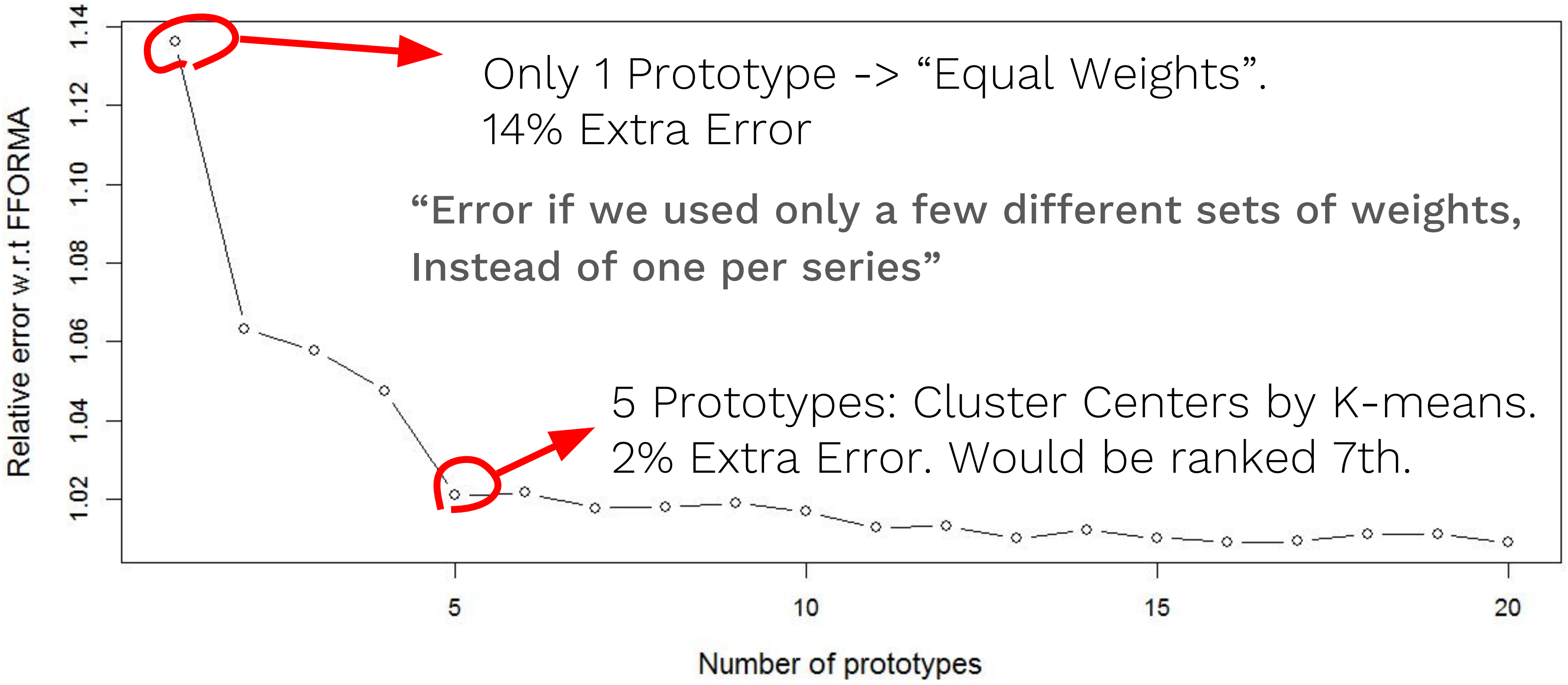
Most important features



Average weights received by methods in the *pool*

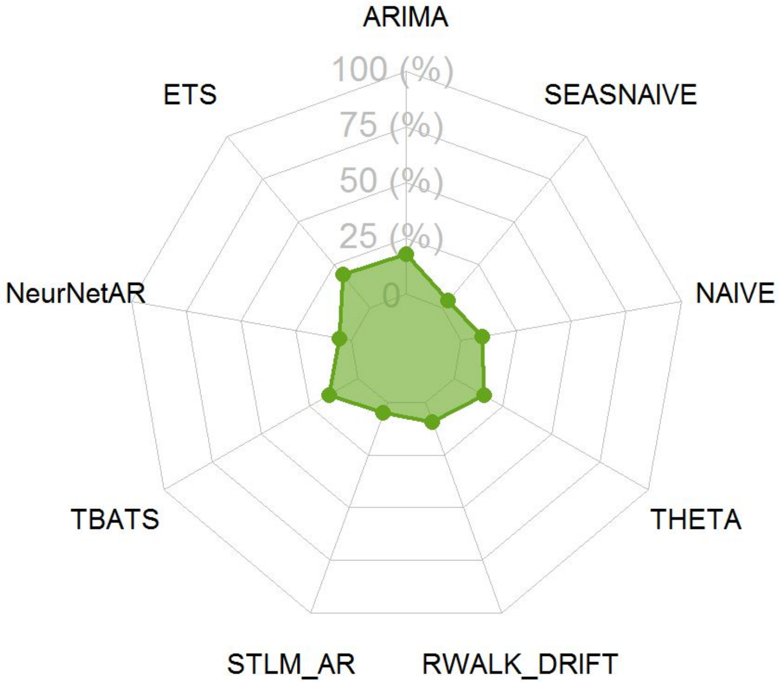


Looking for Prototypes in the weights

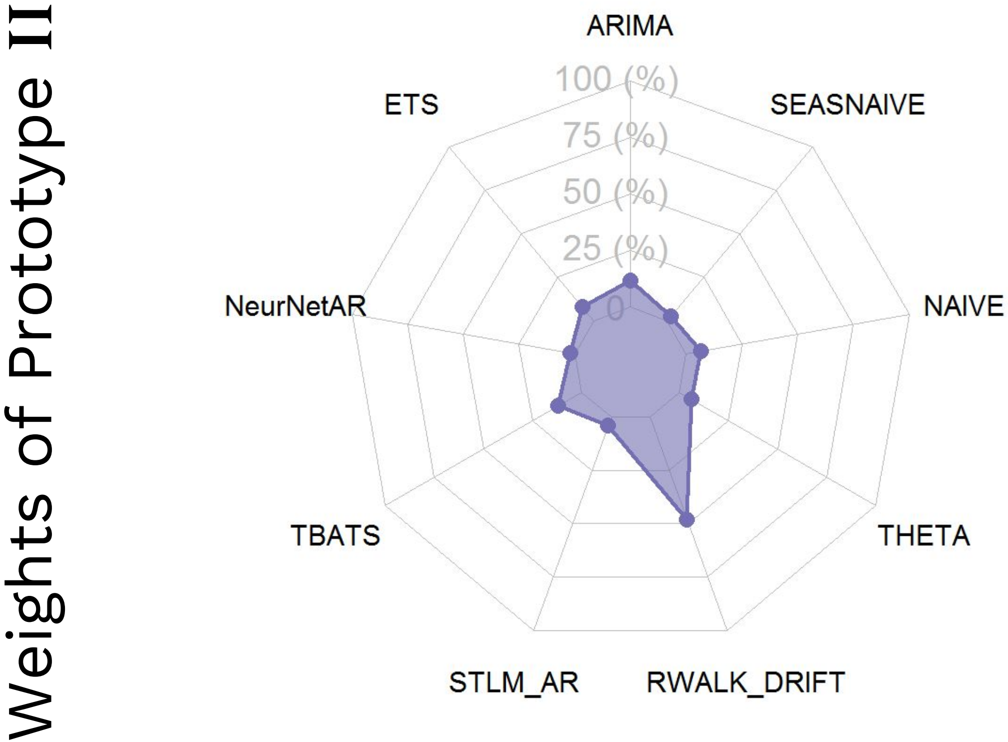


“Roughly Equal Weights”. 40000 Series in M4

Weights of Prototype I

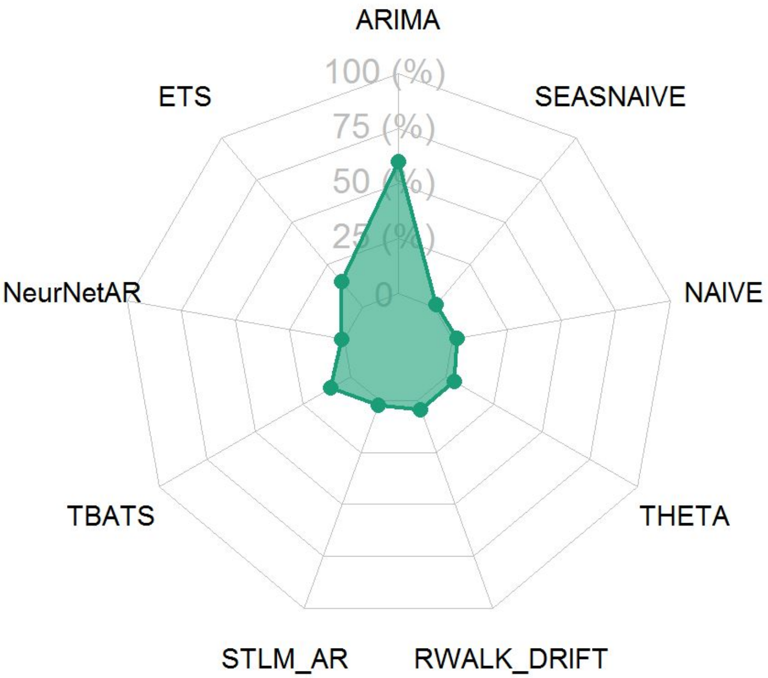


“Mostly RandomWalk Drift”. 20000 Series in M4



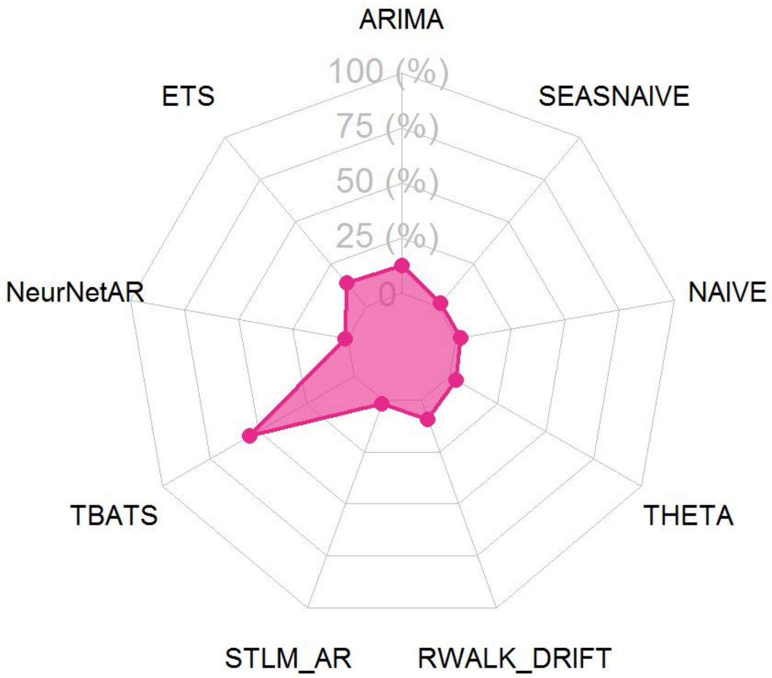
“Mostly ARIMA”. 16000 Series in M4

Weights of Prototype III



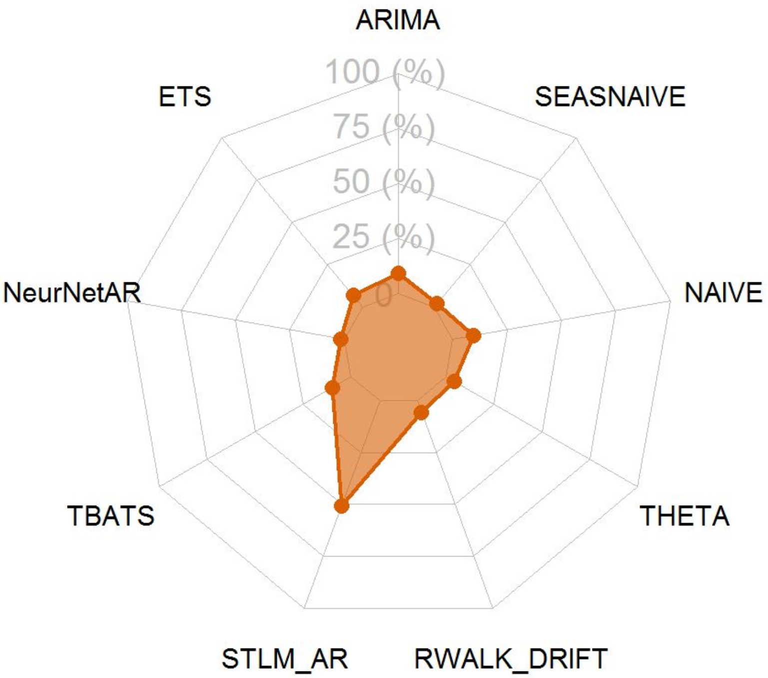
“Mostly TBATS”. 13000 Series in M4

Weights of Prototype IV



“Mostly STLM-AR”. 8000 Series in M4

Weights of Prototype V



“Conclusion”

60% of the series are assigned roughly equal weights

40% of the series assign a large weight to only one method

Robustness against changes in the pool

- What happens if we change the individual forecast methods in the *pool*?

- **Ablation Study:** Remove one method from the *pool*, repeat the whole training + forecasting process.

Increase in error when removing methods from the *pool*

- STLM_AR: 0.06% increase in error
- NAIVE: 0.16%
- ETS: 0.16%
- SNAIVE: 0.2%
- Neural Net AR: 0.4%
- TBATS: 0.4%
- THETA: 0.5%
- ARIMA: 0.76%
- Random Walk Drift: 1.07%
- ARIMA **and** Rand Walk Drift: 2%

3rd Method in the M4: 0.36% increase

5th Method in the M4: 0.6% increase

7th Method in the M4: 2.6% increase

“Conclusion”

FFORMA “adapts” against changes in the *pool* of individual forecasting methods

The performance is affected by 2% when removing the 2 most important methods,
Comprising 40% of the weights in the original version

Future Work

- Performance can be improved
  - Features
  - Pool of forecast methods
  - Weight-generating Model
    - Deep Learning / Time Series Classification
    - Loss functions
- Explore transfer capabilities
  - Model trained in one dataset, actual forecast in others / Extra Data for training

Summary of FFORMA

- A weighted average of individual forecast models, weights are generated per-series using a decision tree-based model. The model is trained on a dataset of TS.
- Forecast performance 2nd in the M4 Competition
- 5 days comp. time for M4, mostly the individual methods
- 60% of the series are assigned roughly equal weights, 40% get a large weight on only one method.
- FFORMA adapts against changes in the pool of methods