Averaging Predictive Distributions Across Calibration Windows for Day-Ahead Electricity Price Forecasting

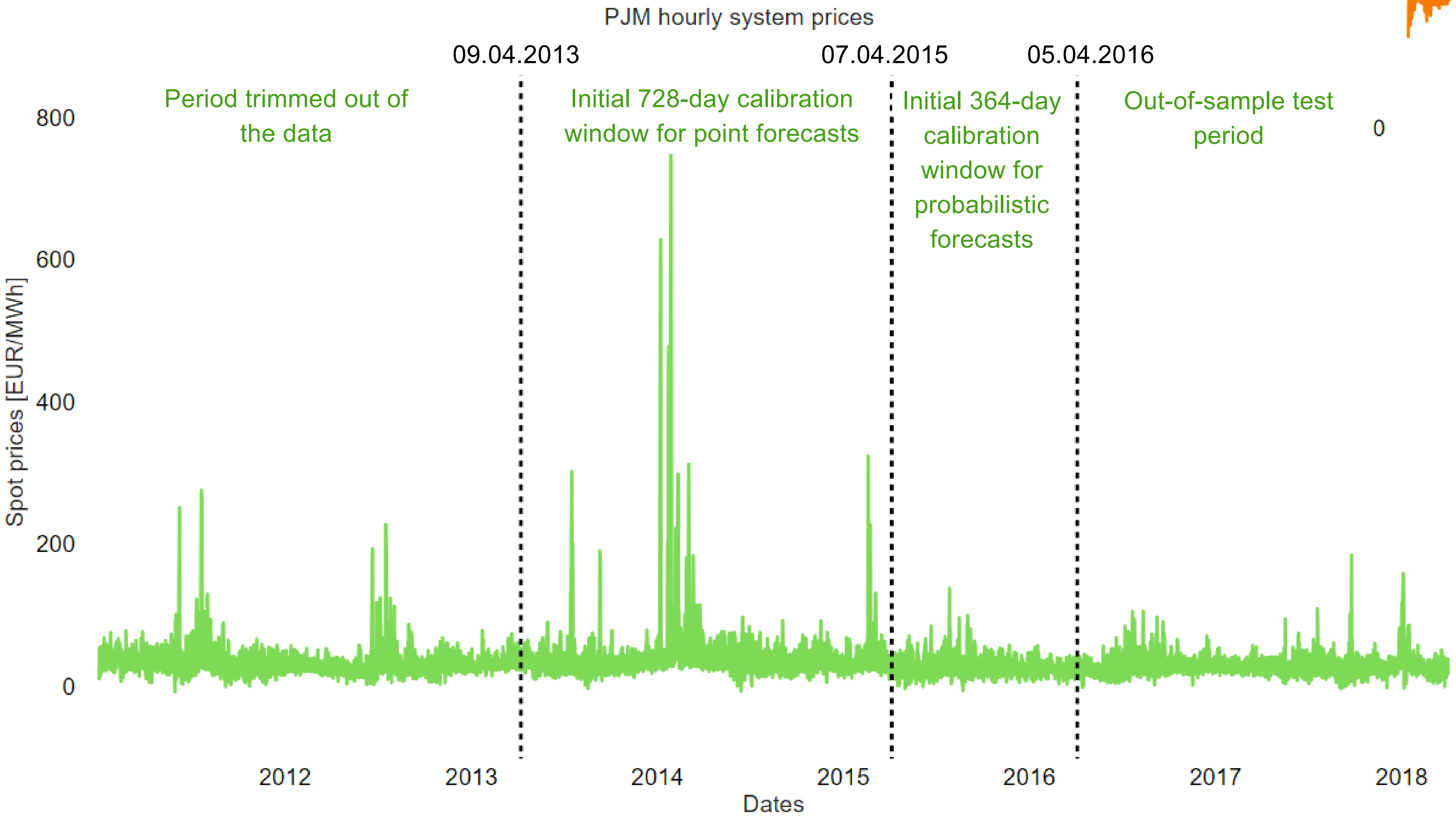
by Tomasz Serafin, Bartosz Uniejewski and Rafał Weron

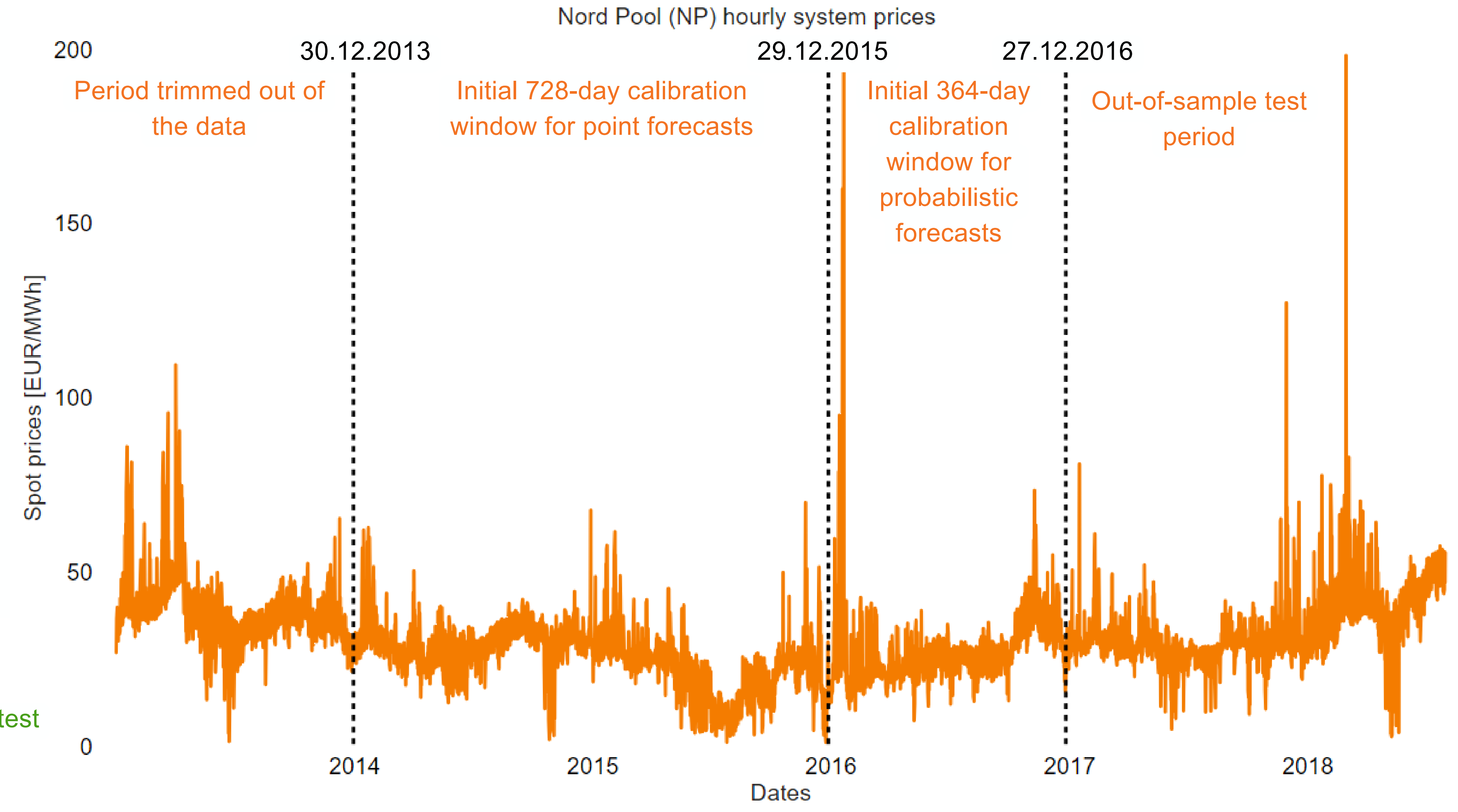
Project authors:
Natalia Czyżyk, Weronika Urbańczyk

Datasets

PJM

Hourly prices and day-ahead load
9 April 2013 to 2 April 2018 (1820 days)
Northeastern United States



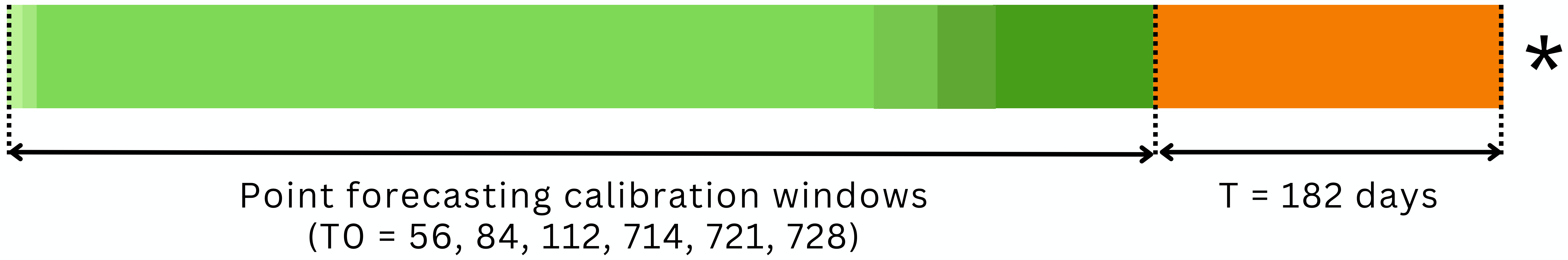


Nord Pool

Hourly system prices and day-ahead consumption
31 December 2013 to 31 July 2018 (1674 days)
Denmark, Finland, Norway and Sweden

Calibration windows

Probabilistic forecasting calibration windows range between T = 14 and 364 days



* - target day for which the predictive distributions are computed

1 2 3

Point forecasts

expertDoW,nl Ziel and Weron model

$$X_{d,h} = \beta_{h,1}X_{d-1,h} + \beta_{h,2}X_{d-2,h} + \beta_{h,3}X_{d-7,h} + \beta_{h,4}X_{d-1,min} + \beta_{h,5}X_{d-1,max} + \beta_{h,6}X_{d-1,24} + \beta_{h,7}C_{d,h} + \sum_{i=1}^{7}\beta_{h,7+i}D_i + \varepsilon_{d,h}$$

NP2018	Forecasted value	Real value
day 1 hour 1	21,359	19,612
day 1 hour 2	21,228	16,416
day 1 hour 3	20,178	16,522
…	…	…
day 728 hour 22	27,337	23,797
day 728 hour 23	22,758	21,369
day 728 hour 24	20,509	19,525

Rolling calibration windows of lengths	MAE (NP2018)	MAE (PJM)
56	2,44	3,80
84	2,34	3,79
112	2,31	3,73
714	2,06	4,16
721	2,06	4,17
728	2,05	4,18

1 2 3

Probabilistic forecasts

for quantiles in range (0.01, 0.99)

Calibration windows range between T = 14 and 364 days

QRA - Quantile Regression Averaging

$$Y_{d,h} = [\,1\ \hat{P}_{d,h}(56,T)\ \ \hat{P}_{d,h}(84,T)\ \ \hat{P}_{d,h}(112,T)\ \ \hat{P}_{d,h}(714,T)\ \hat{P}_{d,h}(721,T)\ \hat{P}_{d,h}(728,T)]$$

$$Q_p(P_{d,h}) = Y_{d,h}w_q$$

QRA involves applying quantile regression to a pool of point forecasts

QRM - Quantile Regression Machine

$$Y_{d,h} = [1\ \bar{P}_{d,h}(T)]$$

$$Q_p(P_{d,h}) = Y_{d,h}w_q$$

QRM first averages point predictions across the six calibration windows, then applies quantile regression to the combined forecast

1 2 3

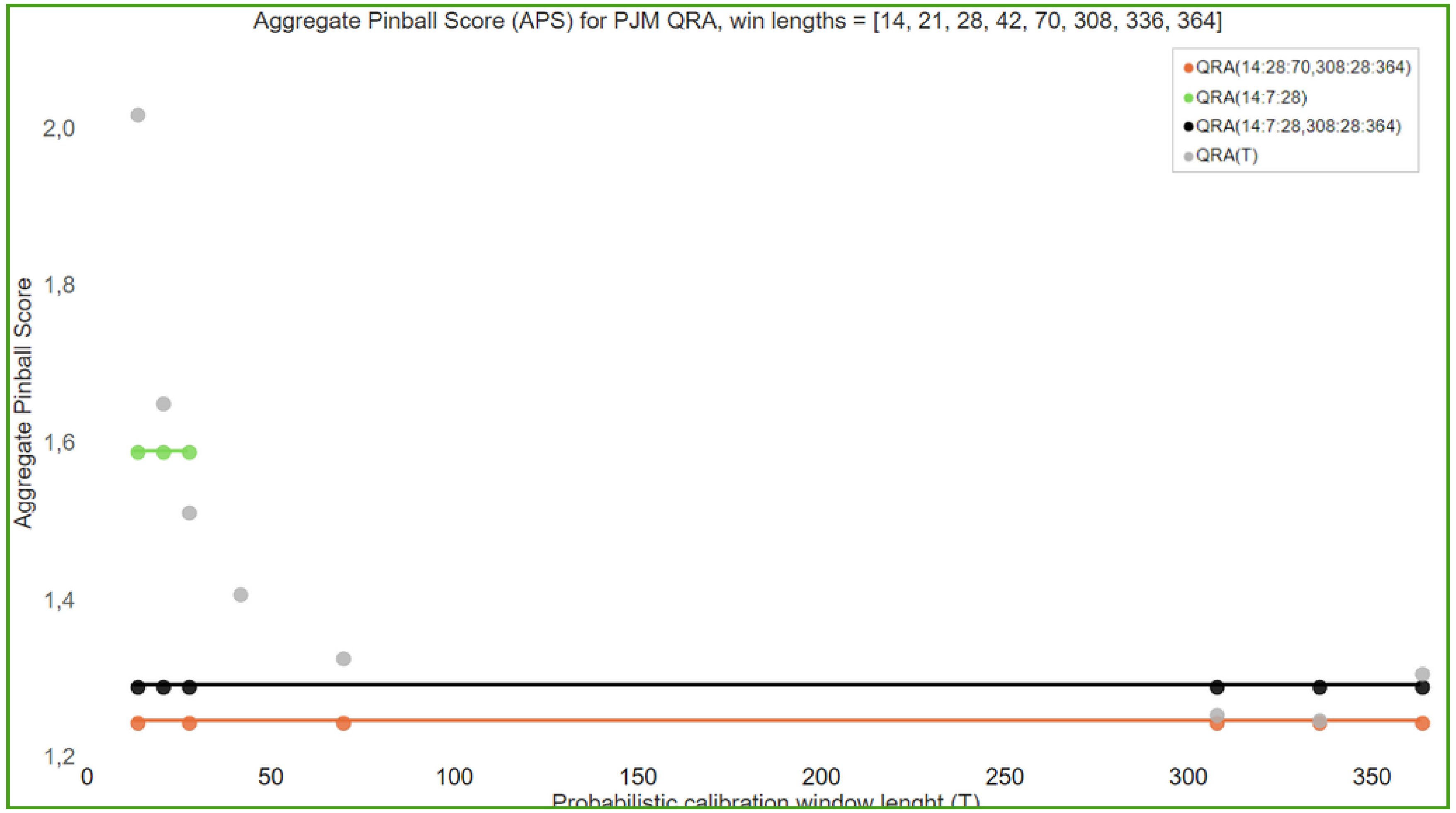
Pinball score

$$PS\left(\hat{Q}_q\left(P_{d,h}\right), P_{d,h}, q\right) = \begin{cases} (1-q)\left(\hat{Q}_q\left(P_{d,h}\right) - P_{d,h}\right) & \text{for } P_{d,h} < \hat{Q}_q\left(P_{d,h}\right) \\ q\left(P_{d,h} - \hat{Q}_q\left(P_{d,h}\right)\right) & \text{for } P_{d,h} \geq \hat{Q}_q\left(P_{d,h}\right) \end{cases}$$

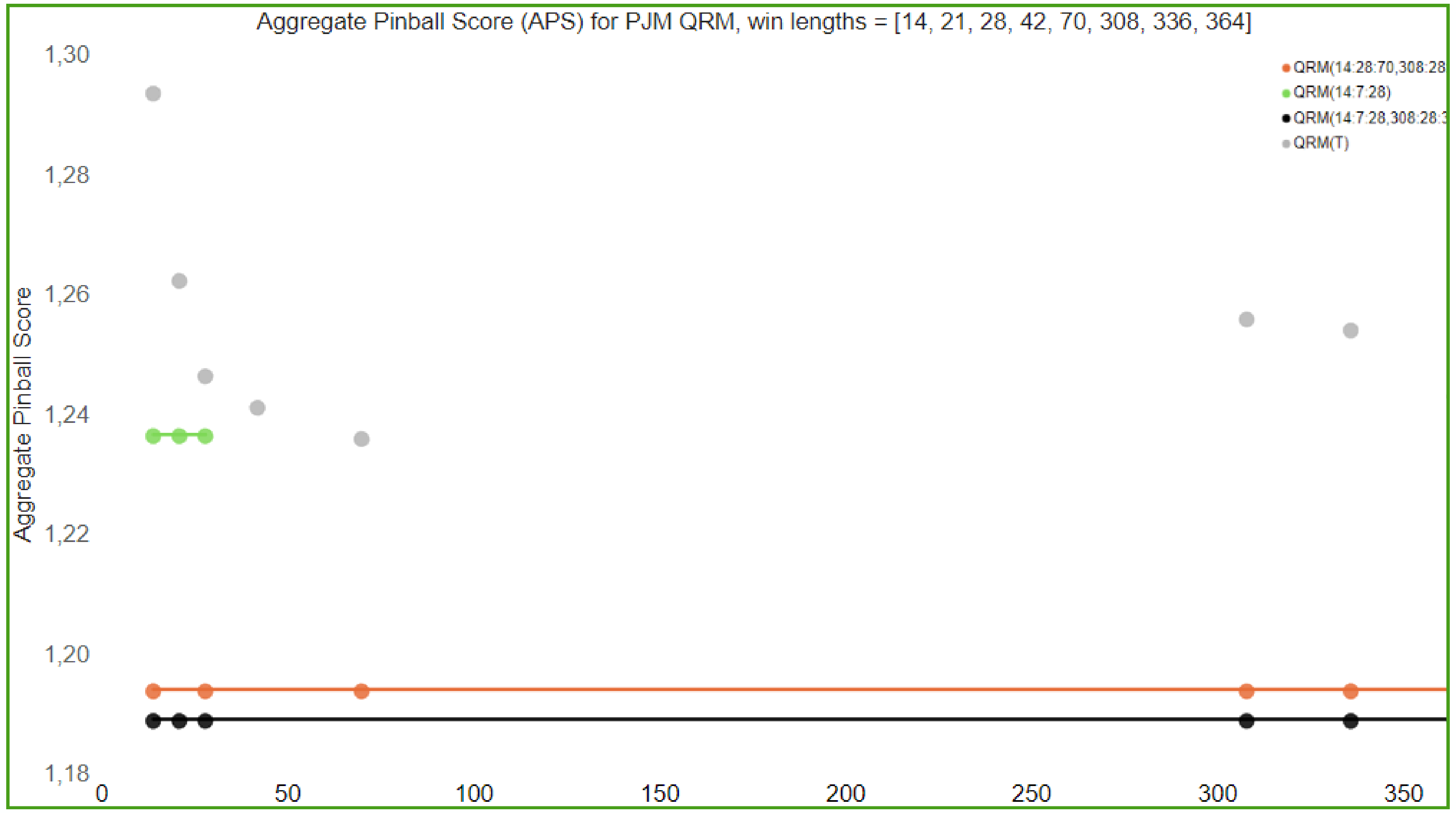
	14	21	28	42	70	308	336	364
NP2018 QRM	0,79	0,76	0,76	0,76	0,76	0,79	0,79	0,79
NP2018 QRA	1,31	1,01	0,92	0,86	0,83	0,79	0,79	0,80
PJM QRM	1,29	1,26	1,25	1,24	1,24	1,26	1,25	1,24
PJM QRA	2,02	1,65	1,51	1,41	1,32	1,25	1,25	1,31

APS results

PJM QRA



PJM QRM



Conclusions

Model for point forecasts proved to be well-performing, yielding an avarege MAE of 2.21 (NP2018) and 3.97 (PJM)

QRM is more time-efficient and yielding better results than QRA

For APS, the best results came from combinations of shortest and longest windows

We obtained mostly similar results to the original article, but some differences could be observed - mainly due to used languages and functions

Thank you!

Do you have any questions?