

# Supplementary Material

Multiresolution Neural Networks for Electricity Market Forecasting:  
Addressing Complexity with  $q$ -Wavelet Methods

*Journal of the Royal Statistical Society: Series A (JRSS A)*

Foued Saâdaoui and Hana Rabbouch

## Overview

This supplementary material provides a detailed description of the implementation and replication workflow for the proposed *Multiresolution Neural Networks for Electricity Market Forecasting* method. The approach integrates  $q$ -MODWT-MRA wavelet decomposition with *Nonlinear Autoregressive (NAR) networks* to capture both long- and short-term dynamics in electricity market time series. The methodology involves decomposing the series into low- and high-frequency components, modeling each component independently with NAR or NARX networks, and recombining the forecasts after applying inverse fractional differencing. This procedure enhances prediction accuracy and accommodates the complex nonlinearities and long-memory behaviors commonly observed in energy markets. The overview outlines the key steps of the methodology, including preprocessing, fractional differencing, wavelet-based component separation, neural network training, and reconstruction of the final forecasts.

## 1 MATLAB Implementation

The proposed method can be implemented directly in MATLAB using both standard toolboxes and the accompanying custom scripts. The time series  $\{y_t\}$ —supplied either in its raw level form, its log-transformed version  $\{\log(y_t)\}$ , or its differenced form  $\{\Delta y_t\}$  when required for stationarity—is first decomposed into low- and high-frequency components through  $q$ -*MODWT-MRA wavelets*. This step combines functionalities from the MATLAB Wavelet Toolbox with additional routines for  $q$ -fractional differencing. Each resulting wavelet component is then modeled independently using *NAR* or *NARX neural networks* available in the Neural Network Toolbox. The individual forecasts are subsequently recombined, after which an inverse fractional differencing of order  $-q$  is applied to recover predictions on the original scale. The full workflow can be reproduced using the provided MATLAB scripts together with the standard toolboxes.

## 2 Wavelet Decomposition and Component Classification

1. Apply MODWT-MRA wavelets to the fractionally differenced time series to obtain  $J$  components.
2. Combine *low-frequency components*, consisting of the approximation level together with the lowest-frequency detail levels, to form a single *low-frequency component* representing slow-moving trends [2].
3. Combine *high-frequency components*, consisting of the remaining detail levels, to form a *high-frequency component* capturing short-term fluctuations.
4. Input each component separately into the NAR/NARX networks to model dynamics at multiple resolutions.

### 3 Code Access and Dependencies

All custom scripts developed for this study are provided as supplementary material. Dependencies include:

- *NAR Toolbox* (Neural Network Toolbox for MATLAB) for training NAR/NARX networks [5].
- *Shimotsu fractional differencing code* to estimate the fractional differencing parameter  $q$ : <https://shimotsu.web.fc2.com/styled-3/>, [4].
- *GPH code by Ludwig Kanzler* for estimating long-memory parameters: <http://fmwww.bc.edu/repec/bocode/g/gph.m>, [1].
- For a user-friendly alternative to implementing NARX networks, the GUI developed by the authors can be used : <https://github.com/FouedSaadaoui/narxgui>, [3].

### 4 Fractional Differencing

To capture long-memory behavior, the fractional differencing parameter  $q$  is estimated using *GPH* [1] and *Local Whittle* tests [4]. The data are then fractionally differenced prior to wavelet decomposition to ensure local stationarity. Numerous publicly available codes and implementations perform similar fractional differencing and long-memory estimation procedures, offering alternative options for replication or adaptation in different contexts.

### 5 Instructions for Replication

To replicate the study results:

1. Verify that the MATLAB Wavelet Toolbox and the Neural Network Toolbox are installed [5].

2. Estimate the fractional differencing parameter  $\tilde{q}$  using either Kanzler's GPH procedure [1] or the Local Whittle estimator [4].
3. Apply fractional differencing of order  $\tilde{q}$  to the time series using Shimotsu's implementation [4].
4. Apply MODWT-MRA decomposition to obtain low- and high-frequency components.
5. Classify components into *low-frequency components* (approximation + lowest-frequency details) and *high-frequency components* (remaining details).
6. Input each component separately into the NAR/NARX network. Include exogenous variables such as *DST dummy variables*, if applicable [3].
7. Train the networks and generate forecasts.
8. Recombine the forecasted components and apply inverse fractional differencing of order  $-q$  to obtain final forecasts in the original scale.

## 6 Figures, Descriptive Statistics, and Analyses

All figures, descriptive statistics, and computational tests reported in this paper were generated using MATLAB, relying solely on the standard toolboxes included in a regular MATLAB installation. The analyses are fully reproducible using built-in MATLAB functionalities, without requiring any additional or custom toolboxes, ensuring transparency, replicability, and ease of adoption by other researchers.

## 7 Data Description and Access

The dataset used in this study is publicly available and can be accessed via the permanent URL provided in the main manuscript<sup>1</sup>. A representative sample of the dataset is also

---

<sup>1</sup><https://aemo.com.au/>

available in the accompanying GitHub repository<sup>2</sup>.

## References

- [1] L. Kanzler. GPH: Matlab module to calculate Geweke-Porter-Hudak long memory statistic. Statistical Software Components T850805, Boston College Department of Economics, 1998. URL <http://fmwww.bc.edu/repec/bocode/g/gph.m>.
- [2] F. Saâdaoui and H. Rabbouch. A wavelet-based hybrid neural network for short-term electricity prices forecasting. *Artificial Intelligence Review*, 52:649–669, 2019. doi: 10.1007/s10462-018-9610-9.
- [3] F. Saâdaoui. NARX-GUI: A tool for neural network forecasting with external inputs. GitHub Repository, 2024. URL <https://github.com/FouedSaadaoui/narxgui>. License CC BY 4.0.
- [4] K. Shimotsu and P. C. B. Phillips. Exact local whittle estimation of fractional integration. *The Annals of Statistics*, 33(4):1890–1933, 2005. doi: 10.1214/009053605000000309.
- [5] The MathWorks, Inc. *MATLAB R2025a*. Natick, Massachusetts, United States, 2025. URL <https://www.mathworks.com/products/matlab.html>. Wavelet Toolbox and Neural Network Toolbox.

---

<sup>2</sup>[https://github.com/FouedSaadaoui/JRSSA\\_Supplementary\\_Material.git](https://github.com/FouedSaadaoui/JRSSA_Supplementary_Material.git)