Agent-based modelling of market competition among flexibility options using Temporal Fusion Transformer networks

Felix Nitsch[1,*], Christoph Schimeczek[1], Valentin Bertsch[2]

[1] Department of Energy Systems Analysis
Institute of Networked Energy Systems
German Aerospace Center (DLR)
Curiestraße 4, 70563 Stuttgart, Germany
* corresponding author: Felix.Nitsch@dlr.de +49 711 6862-8865

[2] Chair of Energy Systems & Energy Economics
Ruhr-Universität Bochum
Universitätsstraße 150, 44801 Bochum, Germany

DLR

PARIS AGREEMENT – THE ENERGY TRANSITION IS SET

Credits: BMU/Sascha Hilgers

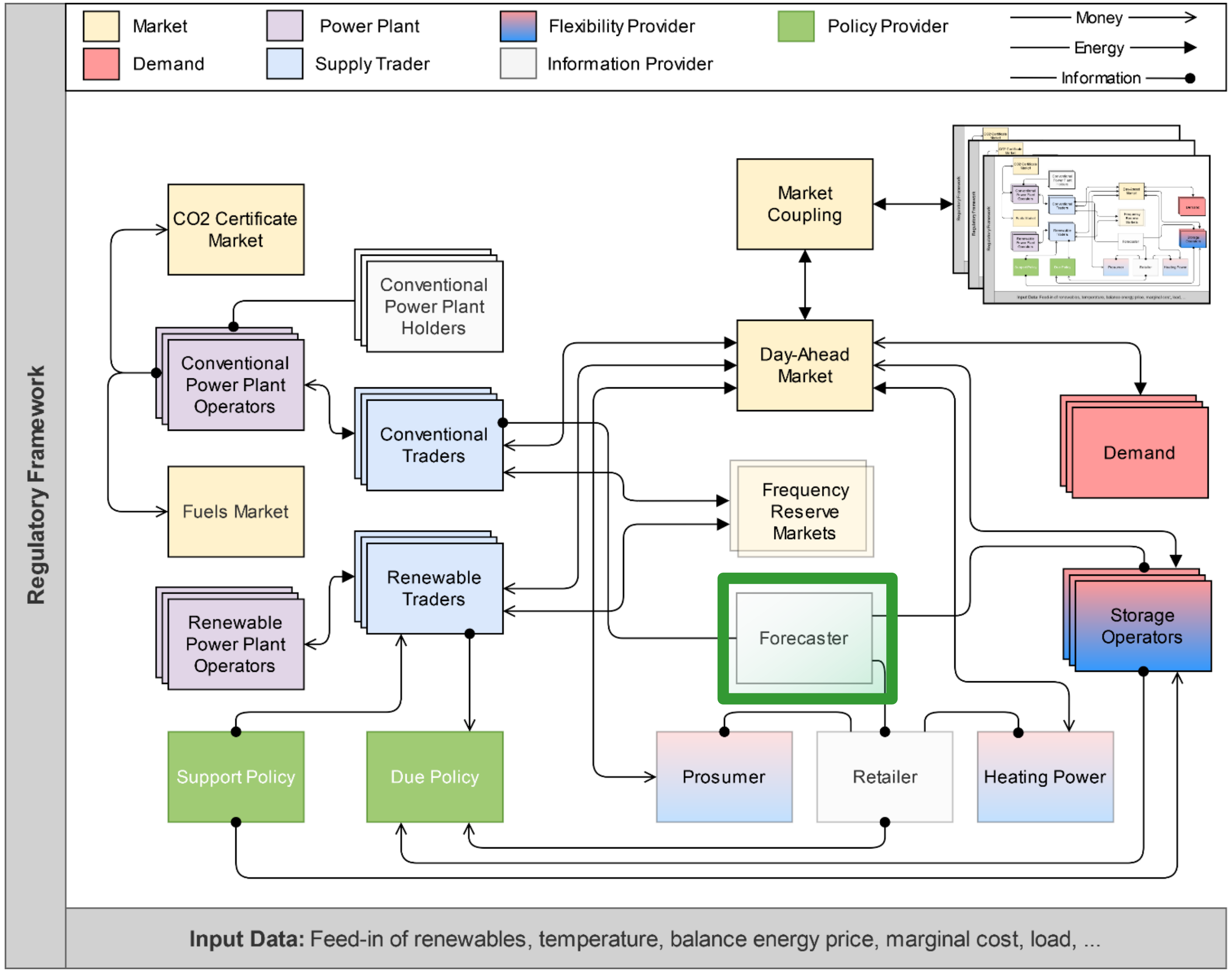
AMIRIS

Agent-based Market model for the Investigation of Renewable and Integrated energy Systems

Model
- Electricity market simulation
- Focus: Germany & Europe
- To be **Open Source** in 2021

Agents
- Conventional Plants
- Renewable Plants
- Traders
- Flexibilities
- Markets
- Policy
- Forecasting

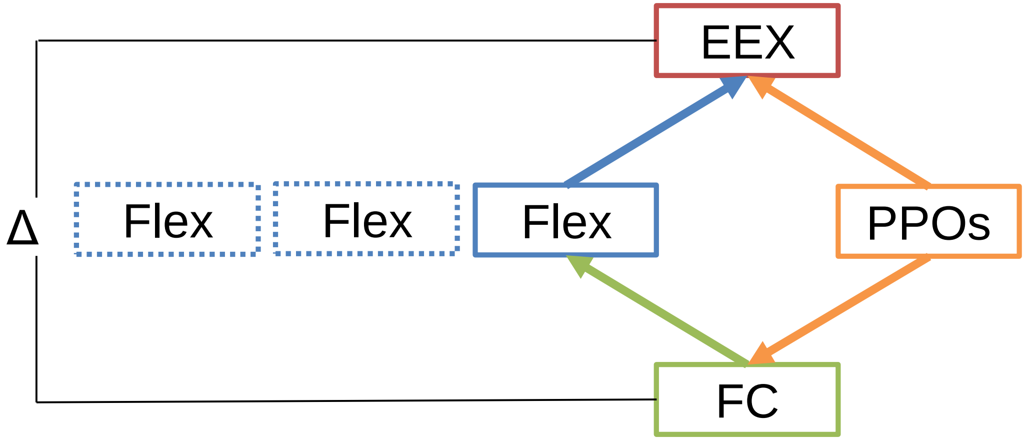
Calculates
- Electricity prices
- Plant dispatch
- Market values
- Emissions
- System costs



DLR

Status Quo: Price forecasting in AMIRIS

1. Power plant operators send future bids to forecast agent
2. Forecast agent calculates forecasted price
3. Forecasts are sent to one Flex-option agent
4. Flex-option agent optimizes its operational strategy
5. All traders send final bids to Energy Exchange
6. Energy Exchange calculates final electricity price



Final and forecasted price difference caused by flex-option agent actions
(i.e. charging → „higher price", discharging → „lower price")

Challenge:
Multiple flex-option agents mutually distort their forecasts due to their competitive actions
→ Significant impacts on the accuracy of the price forecast

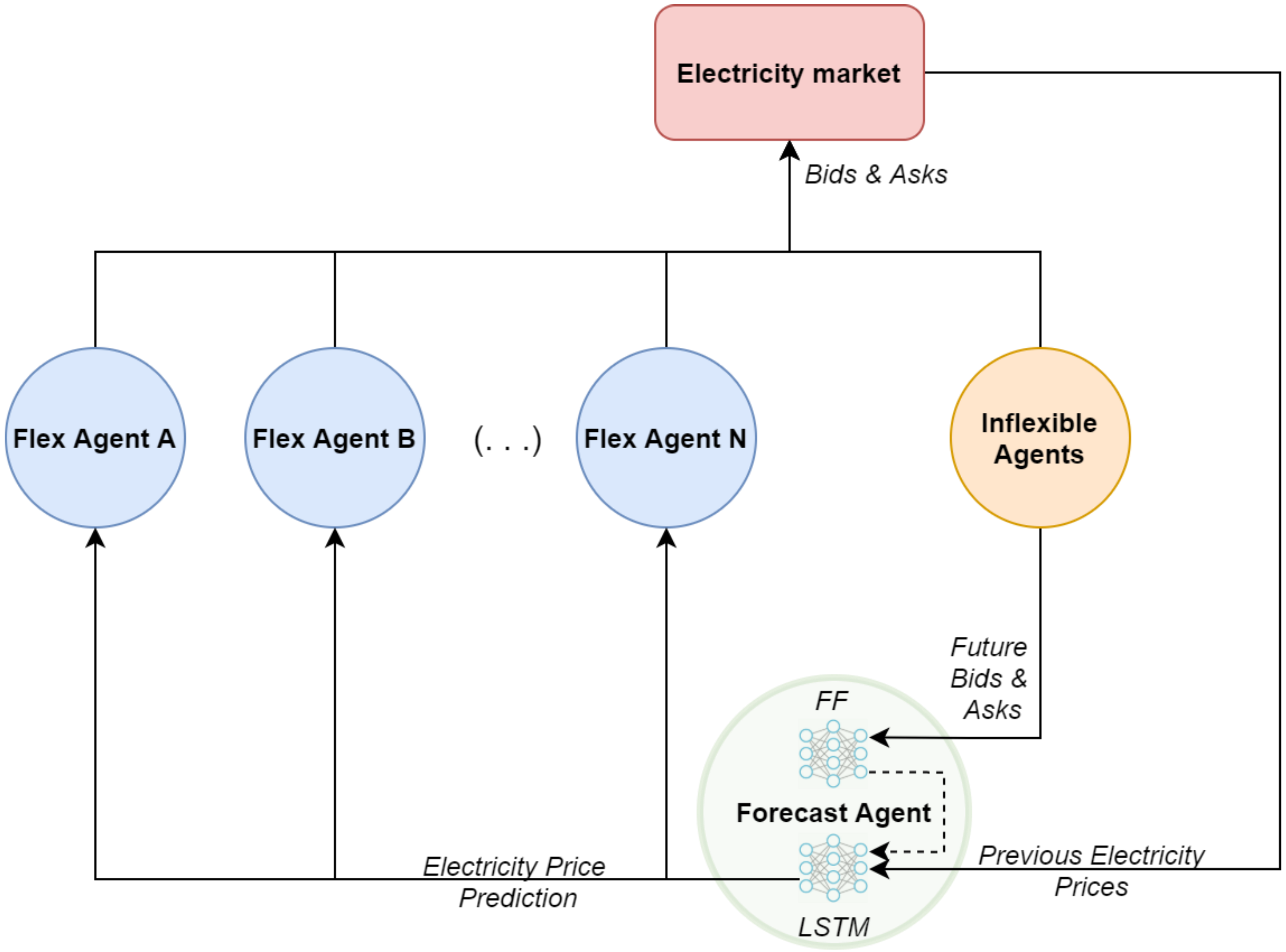
First attempt

Aim

Central forecast agent is learning bidding behaviour of flexibility options and their impacts on prices

Architecture

- 1. Feed-forward model (FF)
- 2. Long-short term memory model (LSTM)
- Inputs:
  - Previous prices
  - Previous residual load
- Output:
  - Price forecast for next 3 hours



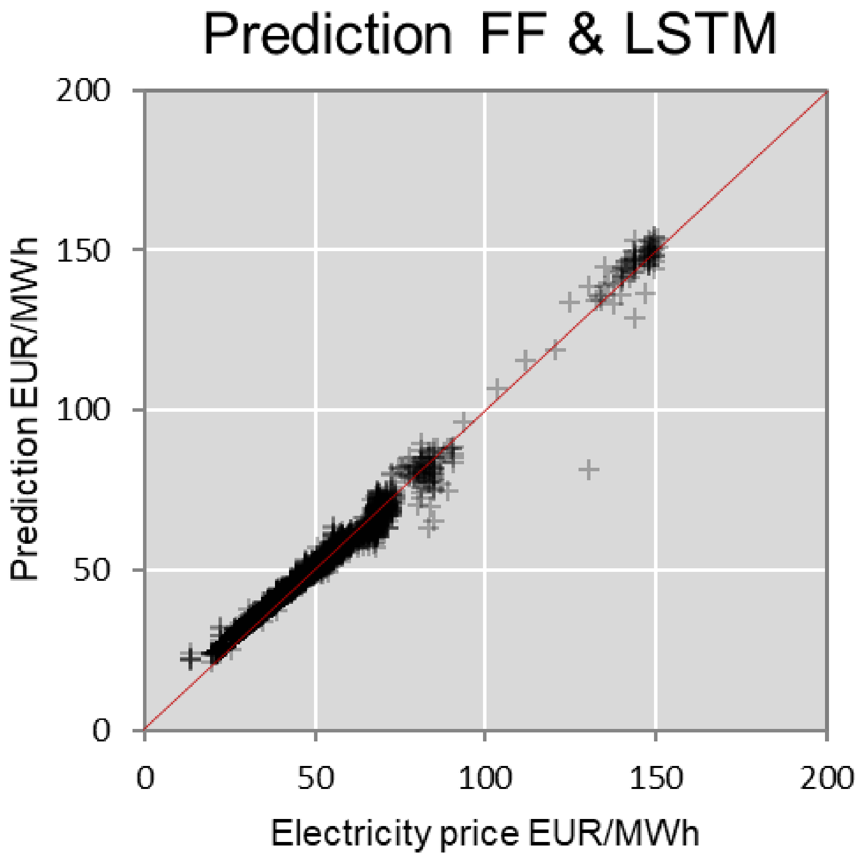
First Attempt’s Results

Result

- Successful proof of concept
- Accurate predictions only up to t+3 time steps

But…

1. t+3 time steps not enough to build operational strategy
2. Uncertainties of prediction values unknown
3. Inconvenient two-staged training process (1. FF → 2. LSTM)
4. “Black box" characteristic of ML prediction



Nitsch, F. and Schimeczek C. (2020). Model in model: Electricity price forecasts in agent-based energy system simulations. INREC Conference. https://elib.dlr.de/136017/

Temporal Fusion Transformers (TFT)

- Novel approach (Lim et al. (2020))
- Attention-based architecture
- Significant performance improvements over existing benchmarks (see Table)

Main features

- Gating mechanisms

  skip over unused components
- Variable selection networks

  select relevant input variables
- Static covariate encoders

  integrate static features
- Temporal processing

  learn long- & short-term relationships

Table 2: P50 and P90 quantile losses on a range of real-world datasets. Percentages in brackets reflect the increase in quantile loss versus TFT (lower q-Risk better), with TFT outperforming competing methods across all experiments, improving on the next best alternative method (underlined) between 3% and 26%.

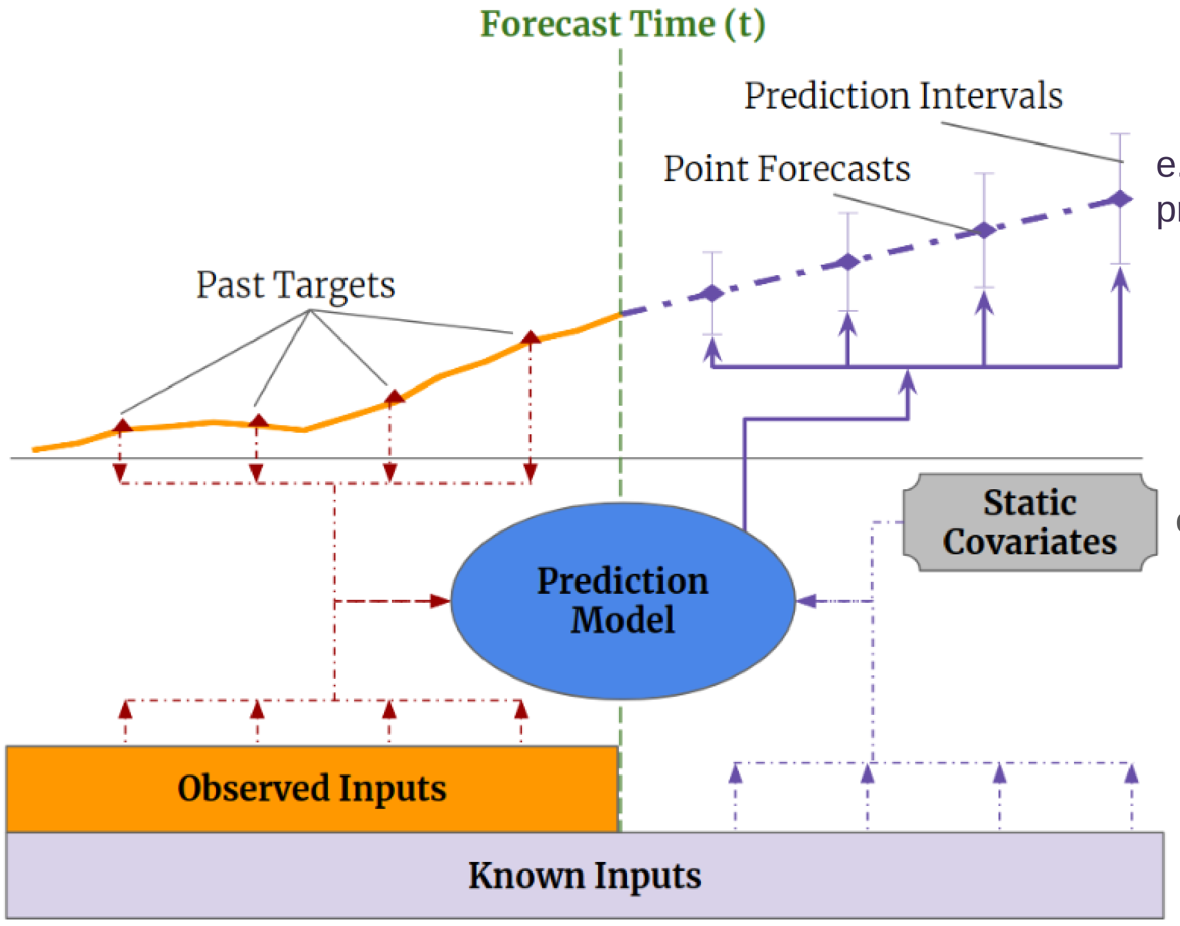
	ARIMA	ETS	TRMF	DeepAR	DSSM
Electricity	0.154 (*+180%*)	0.102 (*+85%*)	0.084 (*+53%*)	0.075 (*+36%*)	0.083 (*+51%*)
Traffic	0.223 (*+135%*)	0.236 (*+148%*)	0.186 (*+96%*)	0.161 (*+69%*)	0.167 (*+76%*)
	ConvTrans	**Seq2Seq**	**MQRNN**	**TFT**	
Electricity	<u>0.059 (*+7%*)</u>	0.067 (*+22%*)	0.077 (*+40%*)	**0.055***	
Traffic	0.122 (*+28%*)	<u>0.105 (*+11%*)</u>	0.117 (*+23%*)	**0.095***	

(a) P50 losses on simpler univariate datasets.

Lim, B., Arik, S. O., Loeff, N., & Pfister, T. (2020). Temporal fusion transformers for interpretable multi-horizon time series forecasting. arXiv preprint arXiv:1912.09363.

TFT Forecasting Concepts

i.e. present time



e.g. next 24 hours including pre-defined quantiles

e.g. previous 24 hours which where observed

e.g. region, market

e.g. prices, load

e.g. load forecast, calendar information

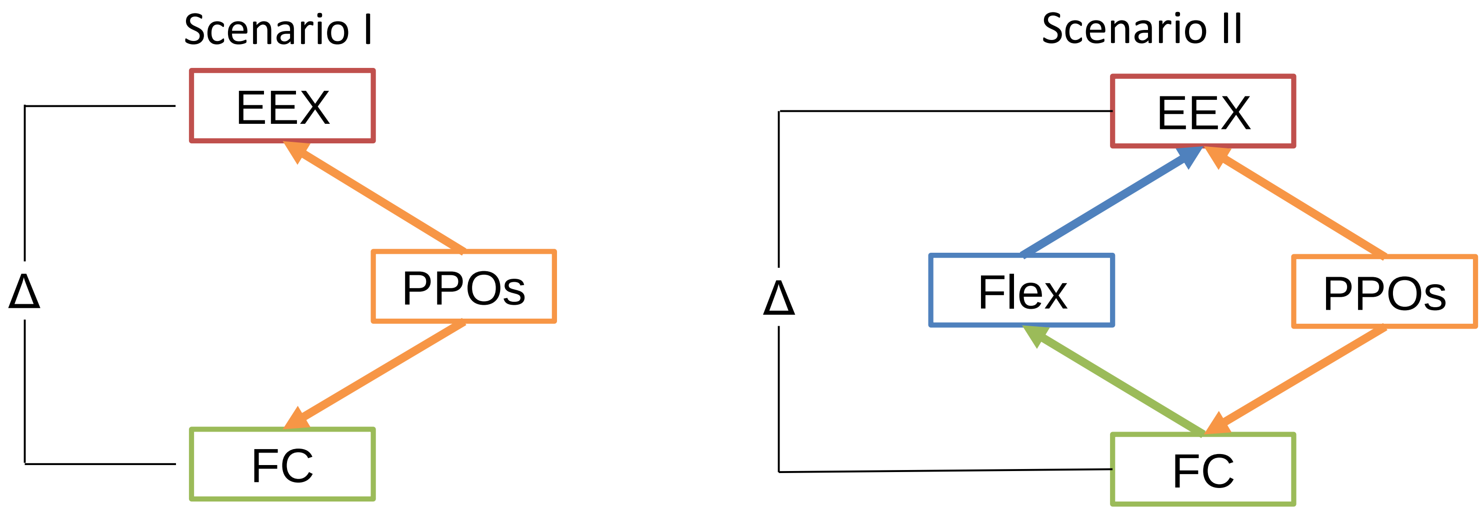
Encoder

Decoder

adapted from Lim et al. (2020)

Workflow

- **1. Hyperparameter scan**
  - Number of hidden layers
  - Dropout
  - Learning rate
  - etc.



- **2. Training**
  - Based on AMIRIS simulation results
    - Scenario I: no flexibility options (easy to forecast, electric load ⇔ price)
    - Scenario II: extensive flexibility options (challenging to forecast, electric load ⇎ price)

Price Prediction – Scenario I – without flexibility option

Encoder 24h – Prediction 6h and 24h

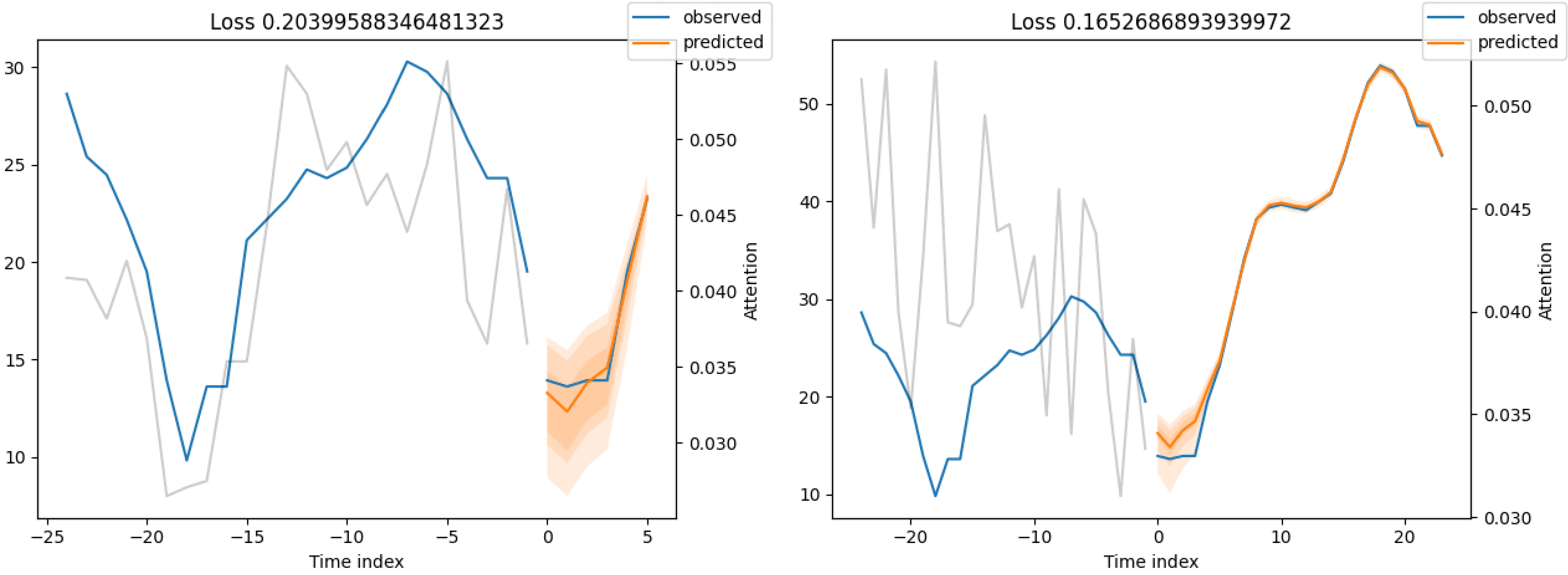


Fig.: Plotting of observed prices (blue) and predictions (orange) including predicted quartiles in EUR/MWh in Scenario I

Price Prediction – Scenario II – with extensive flexibility options

Encoder 24h – Prediction 6h and 24h

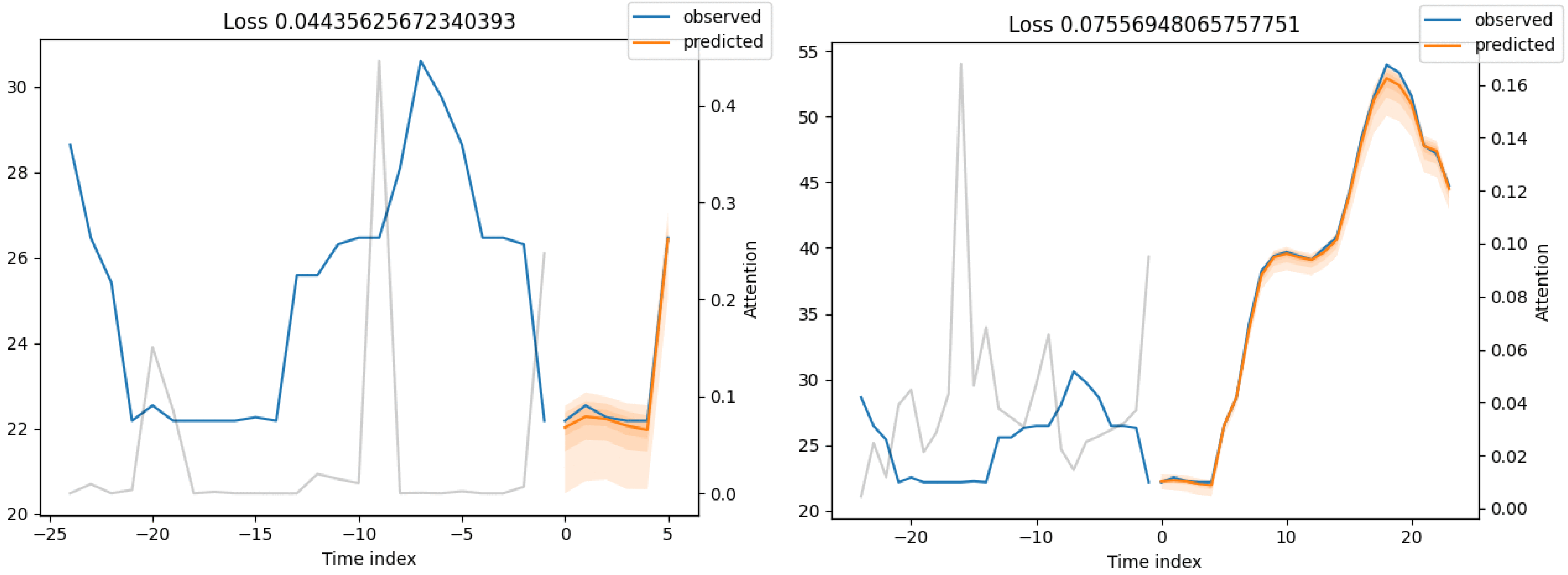
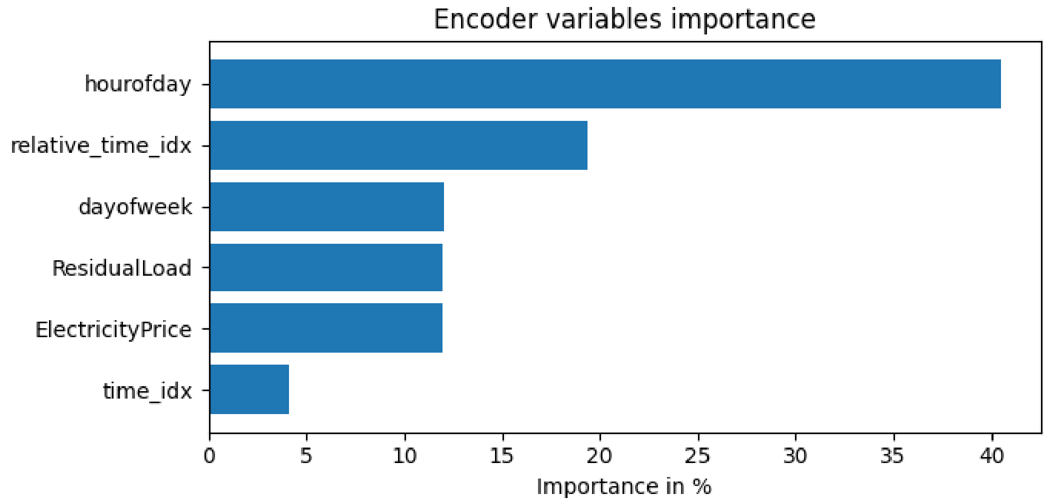


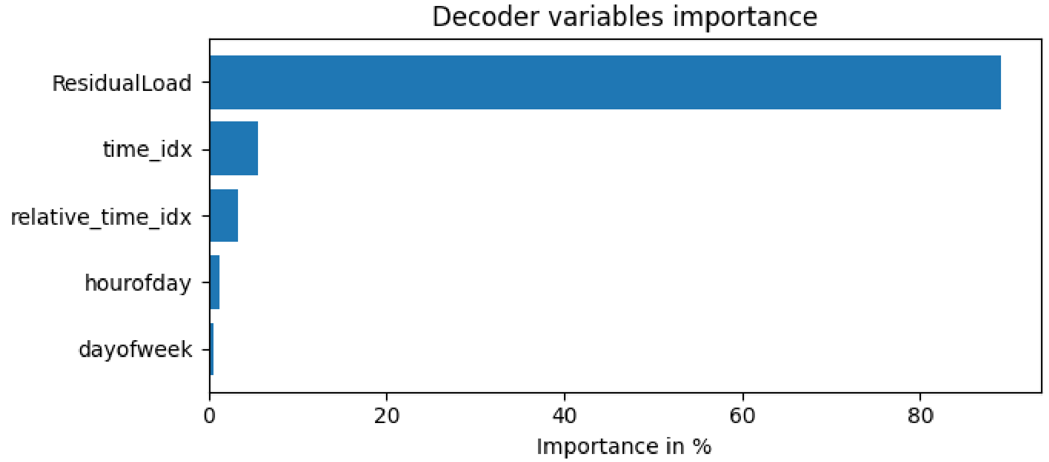
Fig.: Plotting of observed prices (blue) and predictions (orange) including predicted quartiles in EUR/MWh in Scenario II

Variable importance – Scenario II

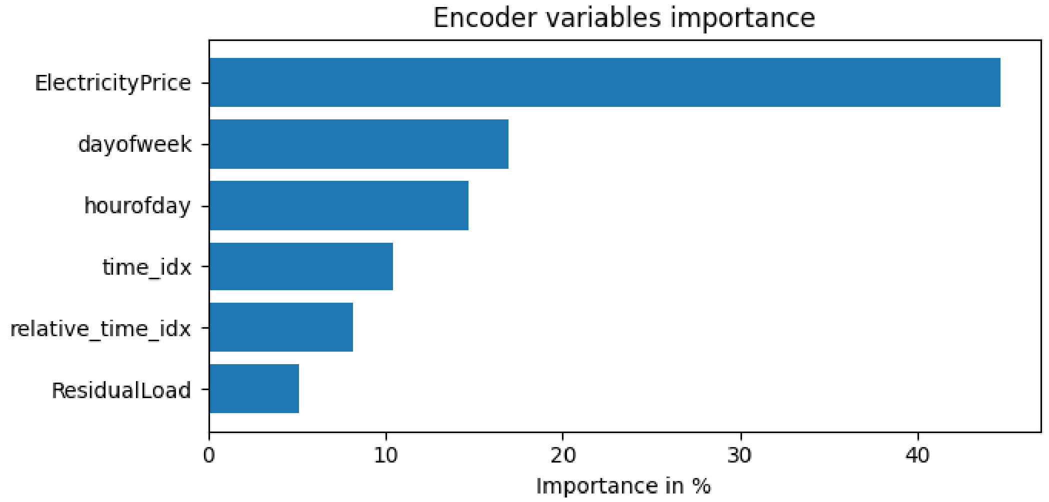
Encoder 24h

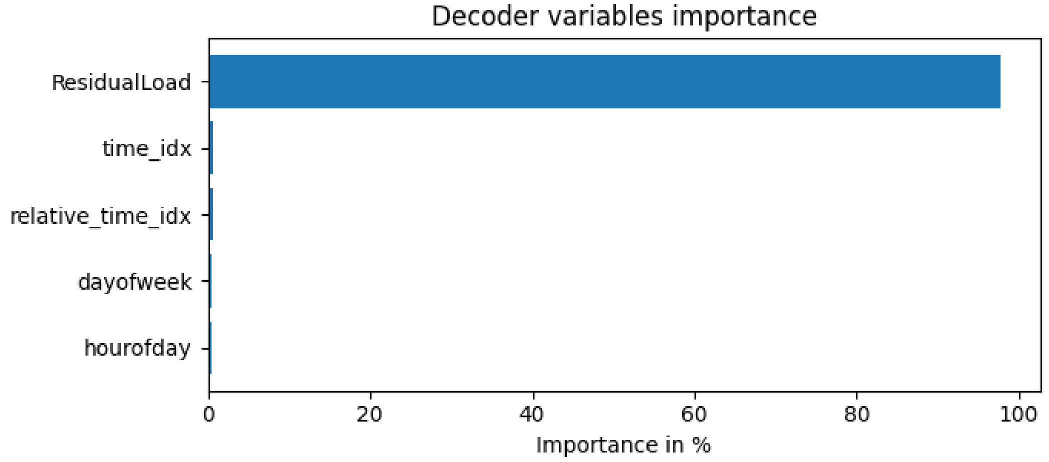
6h Prediction





24h Prediction





Conclusion

1. ~~Accurate predictions only up to t+3 time steps~~
   ► High quality forecasts for 24 time steps

2. ~~No way to consider uncertainties regarding the prediction values~~
   ► Prediction uncertainty estimates

3. ~~Inconvenient two-staged training process (1. FF → 2. LSTM)~~
   ► Very convenient training using pytorch, optuna and TFT implementation by Jan Beitner

4. ~~“Black box" characteristic of ML prediction~~
   ► “Attention" feature: identify important input variables

Outlook

- Generalize training data for scenarios
- Transfer ML models from Python to Java (AMIRIS)
- Develop strategies for flex-option planning to utilize uncertainty information

Contact: Felix Nitsch felix.nitsch@dlr.de German Aerospace Center | Institute of Networked Energy Systems | Energy Systems Analysis