

Supported by:



Federal Ministry  
for Economic Affairs  
and Climate Action

on the basis of a decision  
by the German Bundestag

En4u, FKZ, 03EI1029A

# ENHANCING HOUSEHOLD-LEVEL OPERATIONAL DECISION MAKING WITH MACHINE LEARNING

EURO24, 1<sup>th</sup> of July 2024, Copenhagen

A. Achraf EL GHAZI, Ulrich FREY, Evelyn SPERBER

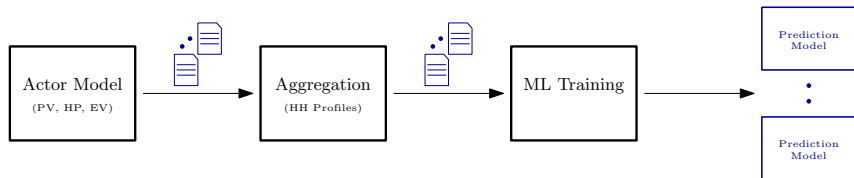
German Aerospace Center (DLR)



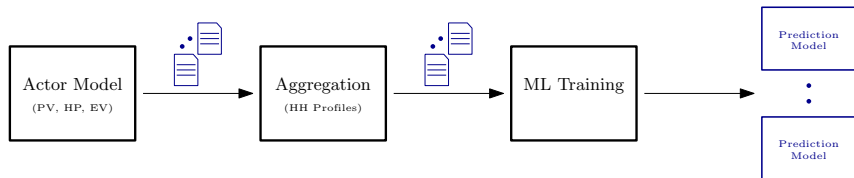
- Agent-based modeling is a comprehensive tool for analyzing the electricity market.
- Considering the behavior of individual household (HH) actors (PV, HP, EV) is important.
  - Traditional aggregation neglects diversity in individual decision-making processes.
  - Influence of varying attributes of individual actors is overlooked.
- Direct modeling of individual HH actors is challenging w.r.t. to resources.

- Agent-based modeling is a comprehensive tool for analyzing the electricity market.
- Considering the behavior of individual household (HH) actors (PV, HP, EV) is important.
  - Traditional aggregation neglects diversity in individual decision-making processes.
  - Influence of varying attributes of individual actors is overlooked.
- Direct modeling of individual HH actors is challenging w.r.t. to resources.

**Solution:**  
An accurate prediction of the aggregated behavior of these actors



- Actor Model: The actual optimization models for PV, HP, and EV.
- Aggregation over the population represented by each HH profile.
- HH profiles fine-grainity differs among technologies, e.g., Weather-Locations
  - are impotent in PV and HP profiles,
  - can be ignored in EV profiles.
- Result: One prediction model for each technology and HH profile.



- Actor Model: The actual optimization models for PV, HP, and EV.
- Aggregation over the population represented by each HH profile.
- HH profiles fine-grainity differs among technologies, e.g., Weather-Locations
  - are impotent in PV and HP profiles,
  - can be ignored in EV profiles.
- Result: One prediction model for each technology and HH profile.



Reduces the complexity of ABM models to one agent per technology and HH profile.

# Challenges of Training the Prediction Models



1. Multiple input variables and one target variable.
2. Input variables and their forward and backward windows are fixed by the corresponding actor model.
3. Each input variable has an individual forward and backward window.
4. Input and output are time series (TS).

# Challenges of Training the Prediction Models



1. Multiple input variables and one target variable.
2. Input variables and their forward and backward windows are fixed by the corresponding actor model.
3. Each input variable has an individual forward and backward window.
4. Input and output are time series (TS).

especially 3., makes the use of popular TS forecasting frame works challenging

# Our Solution: ModPredictor



- <https://gitlab.com/modpredictor>

The screenshot shows the GitLab web interface for the repository 'modpredictor'. The page includes a navigation sidebar on the left with options like 'Verwalten', 'Planen', 'Code', 'Build', 'Bereitstellung', 'Betreiben', 'Überwachen', and 'Analysieren'. The main content area displays the repository name 'modpredictor' and a commit history table. The commit table has columns for 'Name', 'Letzter Commit', and 'Letzte Aktualisierung'. Below the table, the 'README.md' file content is visible, including the project name, name, description, and usage instructions.

Name	Letzter Commit	Letzte Aktualisierung
src	ensured to return the best model wrt val_loss over all csv, epochs, and cross validation splits	vor 1 Monat
README.md	Write README.md	vor 2 Monaten
csv.yml	bugfix: adjust file names to *.yml for pr...	vor 1 Monat

**README.md**

**modpredictor**

**Name**

Modpredictor

**Description**

This project trains a LSTM-model for time-series prediction. The input data is in the `input/` folder. There is data for heat pumps (HP), electric vehicles (EV), and Photovoltaic-Storage systems (PV).

**Usage**

TRAINING: Train each model with the respective training file, i.e. `train_hp`, `train_ev`, `train_pv`. PREDICTION: Predict each model by using the respective prediction file, i.e. `pred_hp`, `pred_ev`, `pred_pv`. You may use the default file for prediction, which is the file



## Our Solution: ModPredictor



- Language: Python.
- ML Dependencies: scikit-learn, keras, tensorflow.
- Uses LSTM with variable sizes and deepness (not yet automatic).
- Two approaches for variable forward and backward windows.
- Ignore concrete time and focus on sequences.

## Our Solution: ModPredictor



- Language: Python.
- ML Dependencies: scikit-learn, keras, tensorflow.
- Uses LSTM with variable sizes and deepness (not yet automatic).
- Two approaches for variable forward and backward windows.
- Ignore concrete time and focus on sequences.

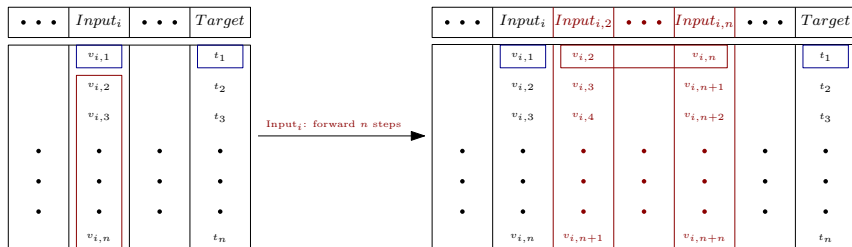
Will be soon available @GitLab as open source

# Variable Forward and Backward Windows – PV Example



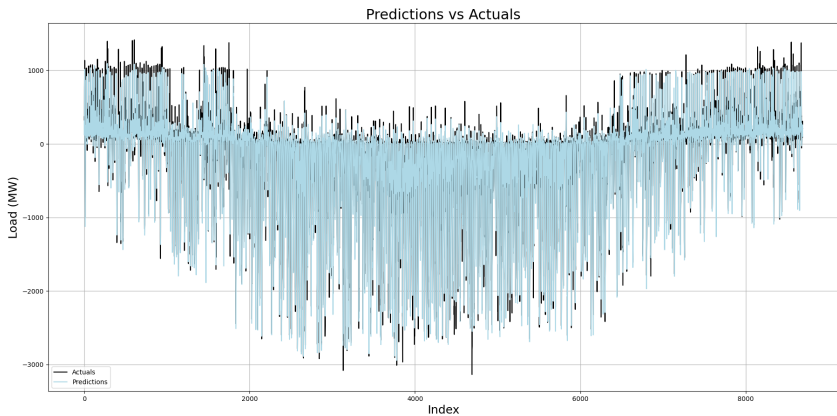
```
1 file_paths_train:          1 file_paths_train:          1 file_paths_train:
  - "../load_pv_MH2_Berlin.csv"    - "../load_hp_BAU_Kiel.csv"    - "../load_ev_profile0.csv"
3 test_ratio: 0              3 test_ratio: 0              3 test_ratio: 0
time_var: "TimeStep"         time_var: "TimeStep"         time_var: "timestamp"
5 input_vars:                5 input_vars:                5 input_vars:
  EnergyGenerationPerMW:         ambient_temperature:          price_EURkWh:
  Forward: 1                    Forward: 48                   Forward: 96
  StoredMWh:                    solar_radiation:              available_charging_kWh:
  Forward: 1                    Forward: 48                   Forward: 96
  ProsumersLoadInMW:           price:                         elec_consumption_kWh:
  Forward: 1                    Forward: 48                   Forward: 96
  SalesPriceInEURperMWh:       tapping_profile:              battery_level_kWh:
  Forward: 24                   Forward: 48                   Forward: 96
target_vars:                 target_vars:                  target_vars:
15 GridInteraction:            15 aggregated_consumption:    15 optimised_load_kWh:
  Backward: 24                 Backward: 48                  Backward: 24
17 model_dir: "output/pv_model" 17 model_dir: "output/hp_model" 17 model_dir: "output/ev_model"
scaling:                      scaling:                       scaling:
19 - "input"                   19 - "input"                  19 - input
  - "target"                   - "target"                    - target
21 epochs: 100                21 epochs: 100               21 epochs: 100
```

# Variable Forward and Backward Windows – Streamlining

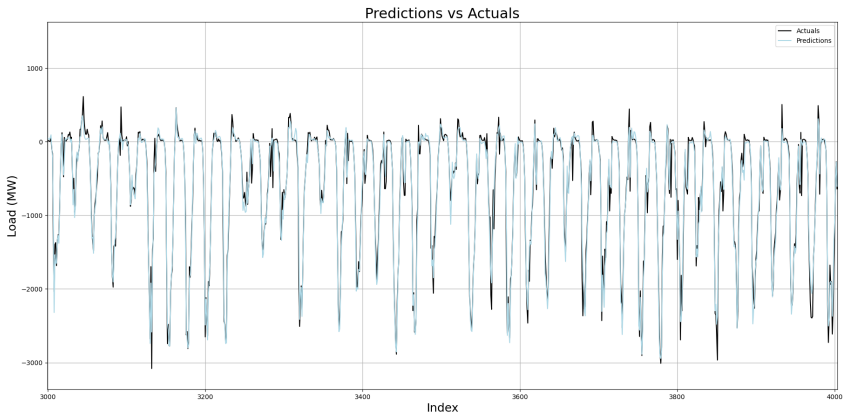


- General – usable with any NN
- Exact – no information change
- However, not feasible for big training data and moderate hardware
- Alternative: Padded tensors, Runtime improvement: > 1day ⇒ < 1Hour.

ENTIRE TEST SET:



ZOOM-IN: entry index 3000 – entry index 4000



## ModPredictor – All Results



MODEL	TRAINING	VALIDATION	TEST	MAE (MW)
PV	8760	CrossVal 1/4	8760	87.80
HP	8760	CrossVal 1/4	8760	241.99
EV	35040	CrossVal 1/4	35040	423.56

- Acceptable predictions for PS, HP, and EV.
- Enhanced understanding of the impact of individual household decision-making.
- ML techniques provide a scalable solution for modeling diverse actor decisions.



Supported by:



Federal Ministry  
for Economic Affairs  
and Climate Action

on the basis of a decision  
by the German Bundestag

# THANK YOU!

[aboubakr.elghazi@dlr.de](mailto:aboubakr.elghazi@dlr.de)

[ulrich.frey@dlr.de](mailto:ulrich.frey@dlr.de)

