University of Stuttgart
Institute for Statics and Dynamics

Machine Learning Methods in Mechanics

Predicting Renewable Energy Production using Machine Learning Methods

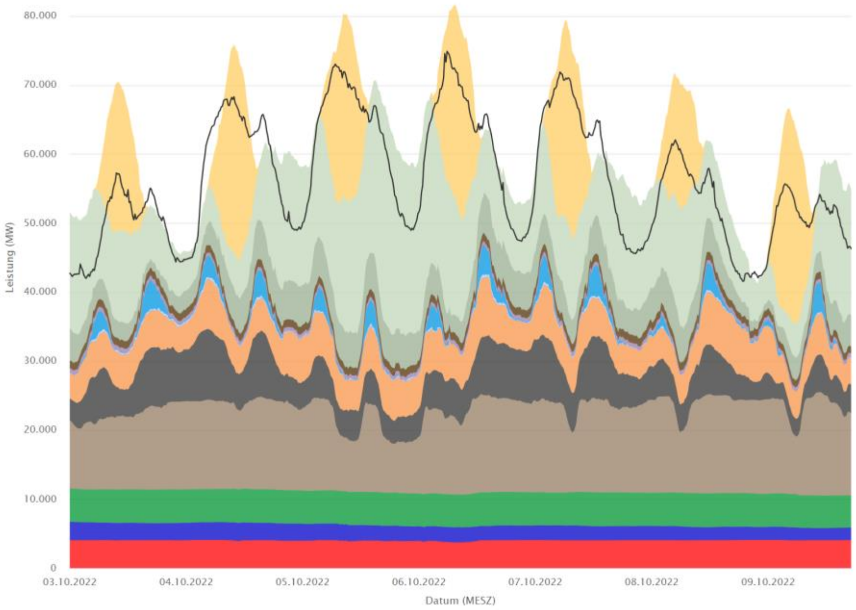
Dilara Yildiz, Luis Gentner, Leon Sengün

Outline

1. Motivation
2. Electricity Data
3. Feature Engineering
4. Benchmark Model Exploration
5. Iterative Model Development
6. Hyperparameter Optimization
7. Results
8. Outlook

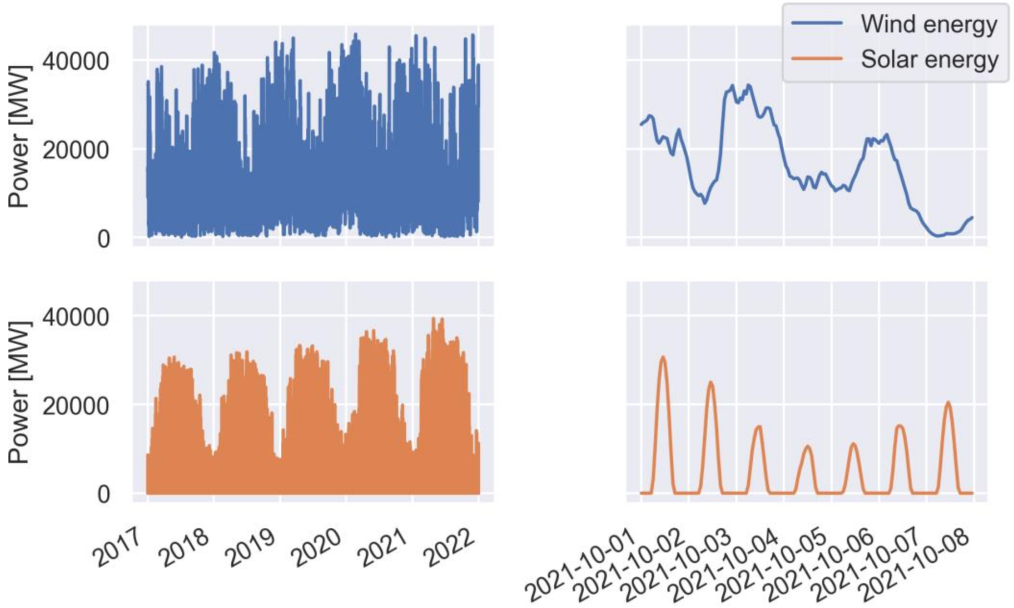
1. Motivation

- Renewable energies are fluctuating
- Useful for power plant operators
- **Goal:**
  - Predict renewable energy production 12 hours into the future



2. Electricity Data
Past inputs and labels

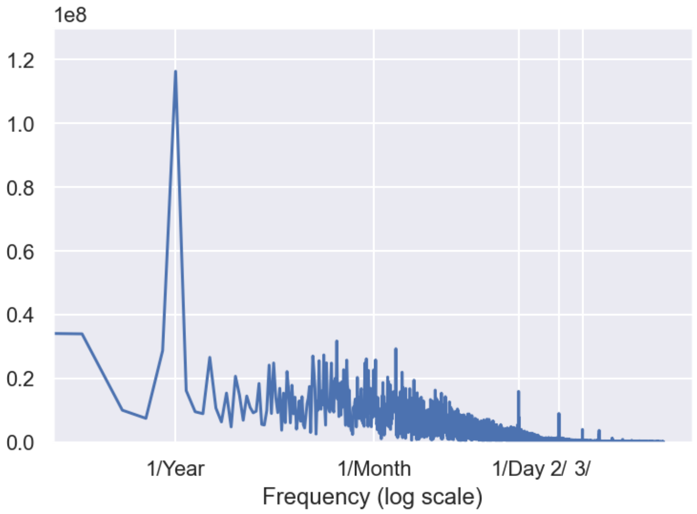
- Hourly production of wind and solar energy in Germany
- Timeframe: 2017 - 2021
- Source: Energy-Charts (from Fraunhofer ISE)



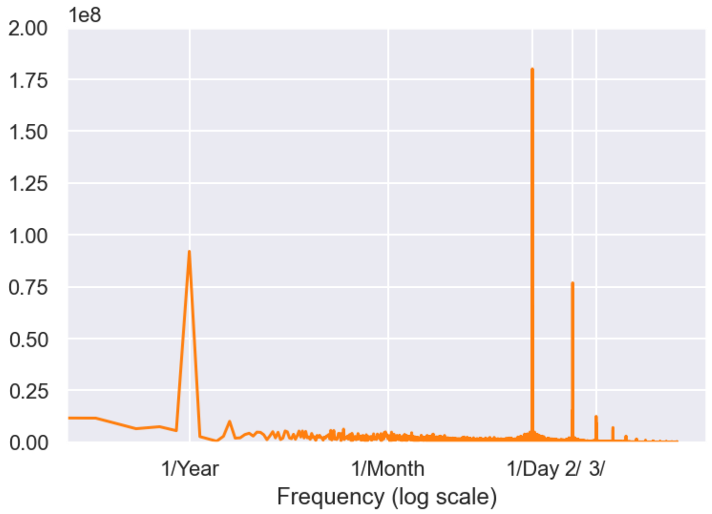
3. Feature Engineering
Frequency Analysis

- Solar and wind energy production show high yearly and daily periodicity

Wind energy production:



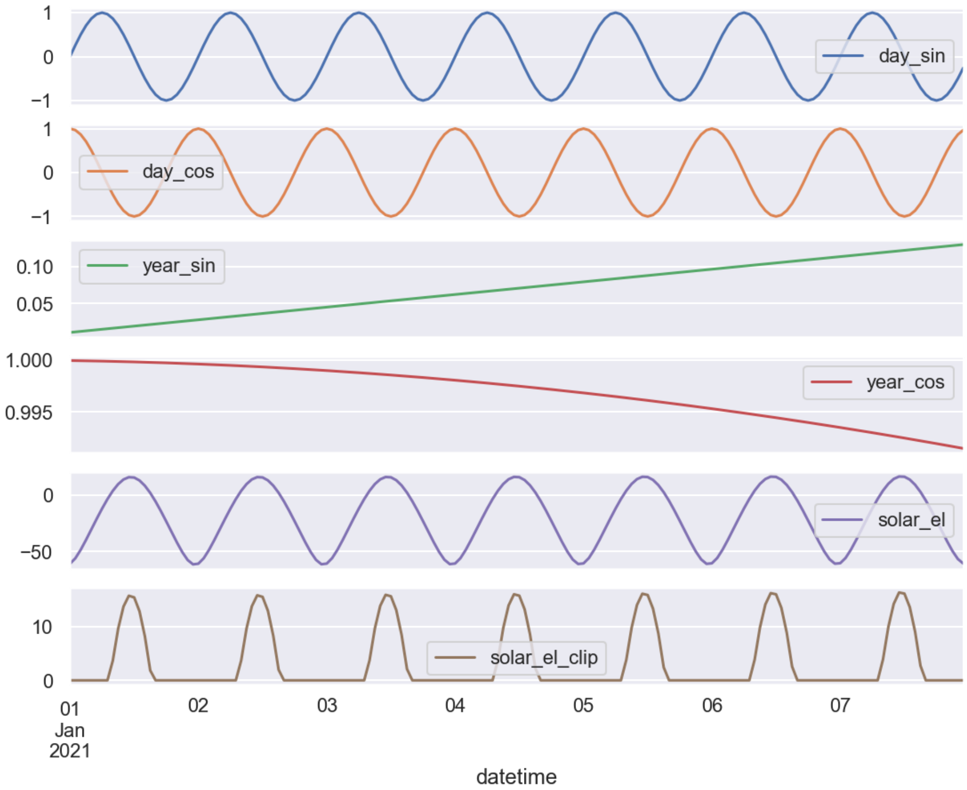
Solar energy production:



- → Create yearly and daily time signals

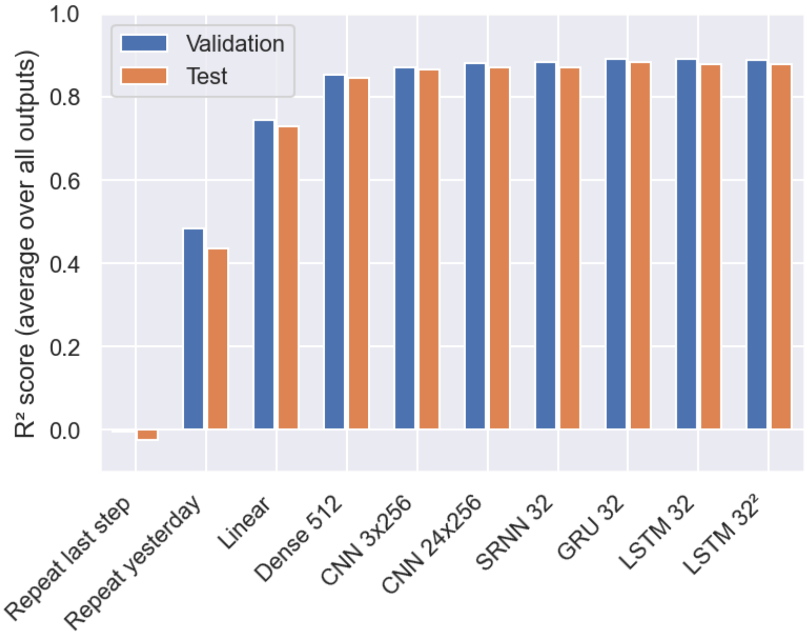
3. Feature Engineering
Time Signals and Solar Elevation

- Time of day and time of year:
  - Periodic representations
  - Sine and cosine signals
- Solar elevation:
  - Clipped and unclipped signals
- Precisely known for the past and future



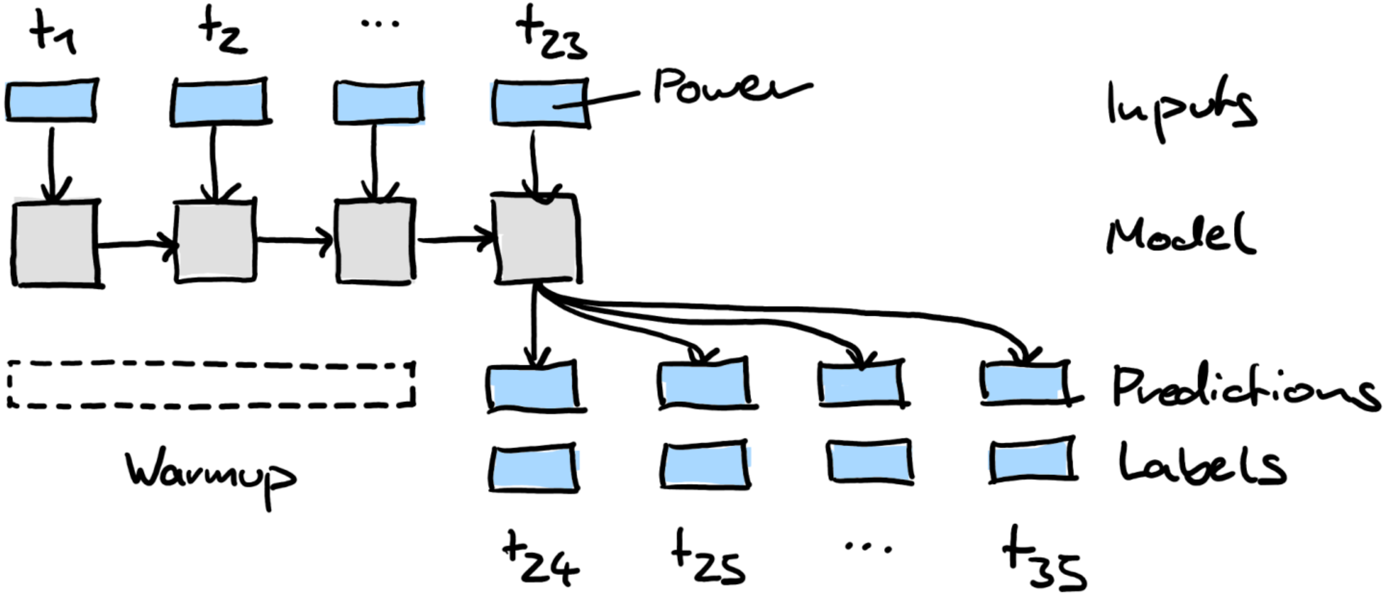
4. Model Architecture Exploration

- Creation of non-trainable benchmark models:
  - Repeat last step
  - Repeat yesterday
- Searching for feasible model architectures
- Results:
  - CNNs and RNNs show best performance: GRU 32 with best R^2 score: 0.882
  - Continuing with RNNs due to their better performance for increased input and prediction lengths



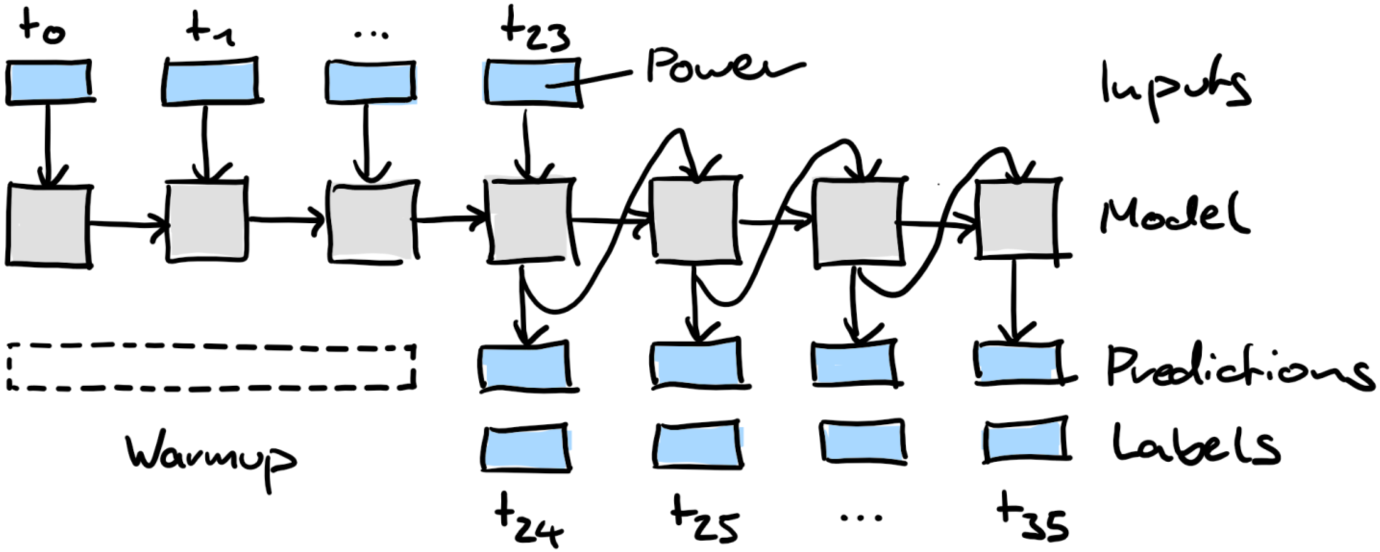
5. Iterative Model Development
Single-shot RNNs

- Prediction of multiple time steps at once (usually with an added Dense layer)



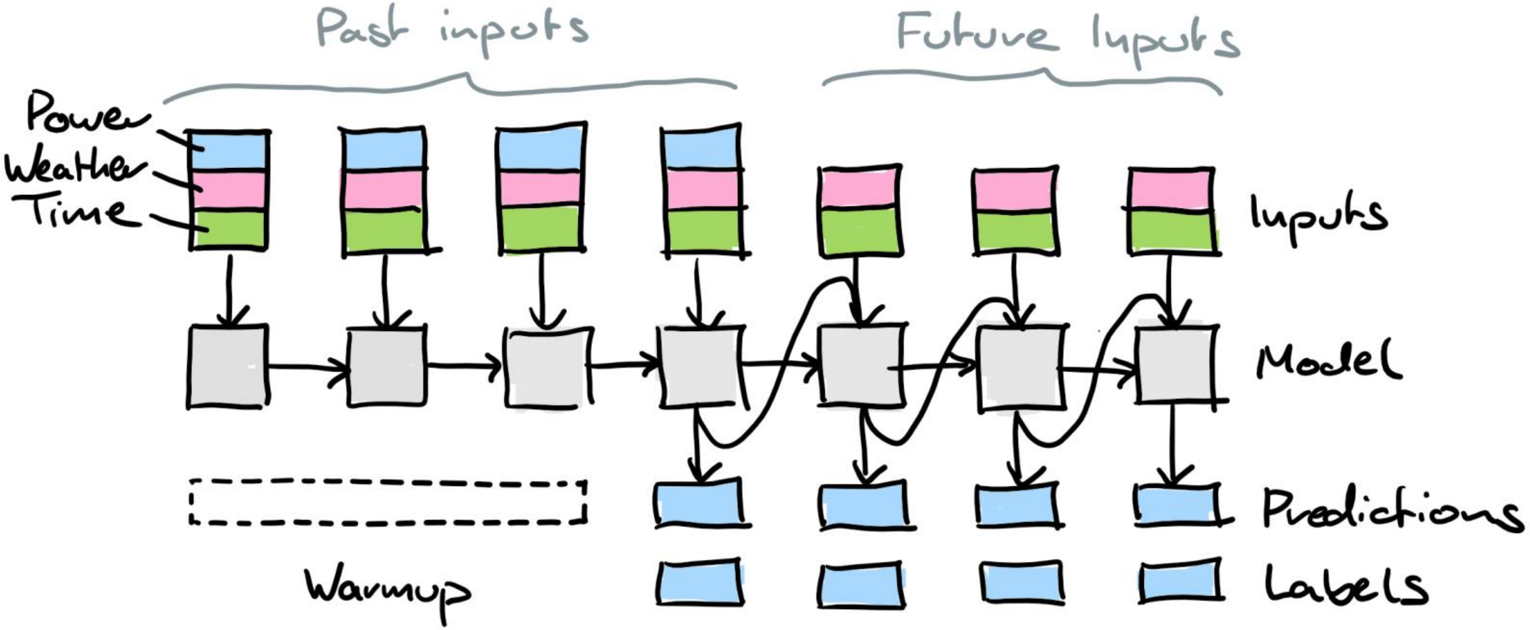
5. Iterative Model Development
Autoregressive RNNs

- The prediction of the last time step is used as input for the next time step
- Prediction of varying output length without retraining



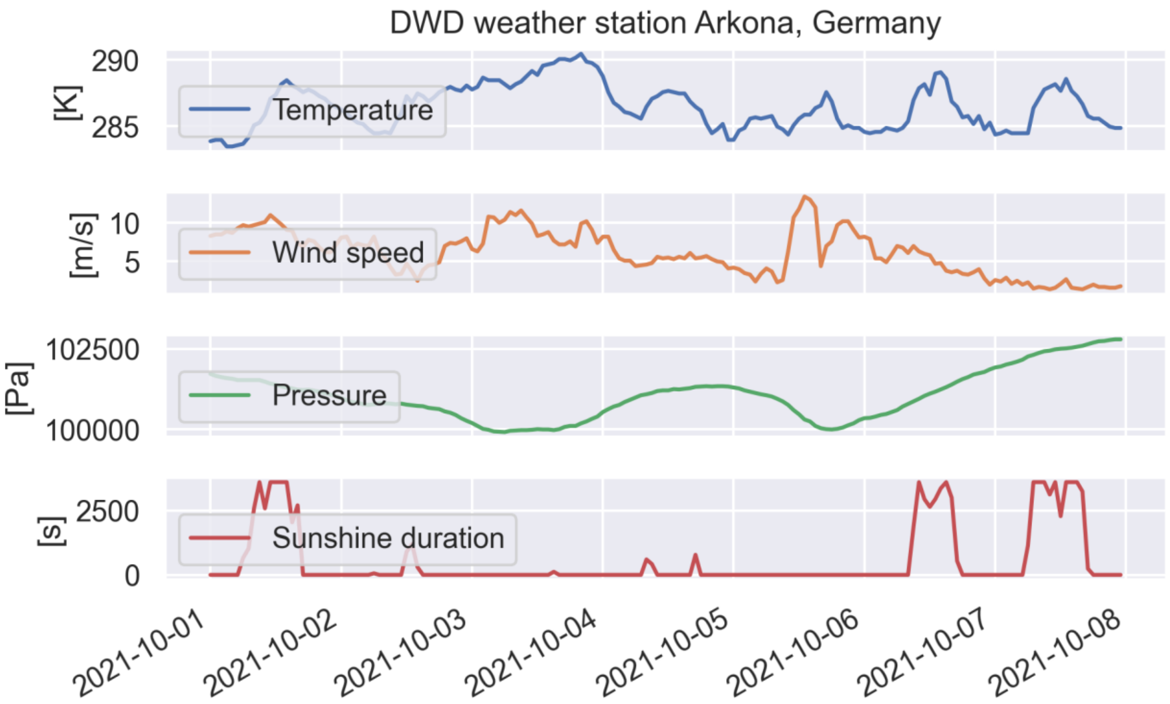
5. Iterative Model Development
Autoregressive RNNs with continous input

- Our approach: Combine the autoregressive predictions with forecasted inputs



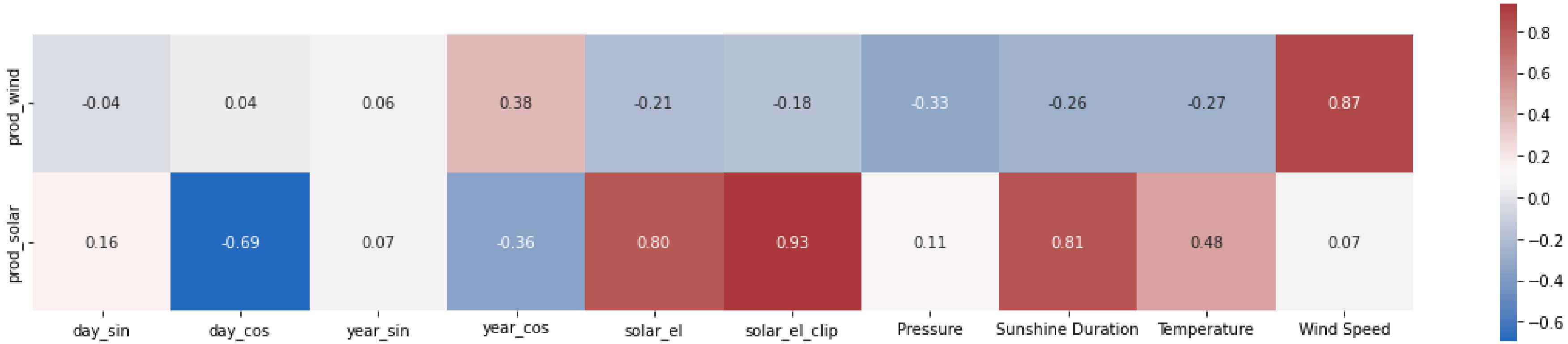
5. Iterative model development
Additional inputs

- Hourly resolution
- Timeframe: 2017 - 2021
- Source: DWD
- Available for up to 117 DWD stations across Germany



5. Iterative Model Development
Correlation analysis

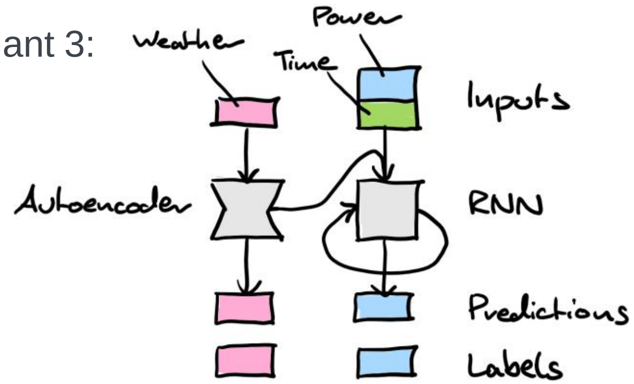
- Decision on which features are interesting/important
- Determine which features to include in model



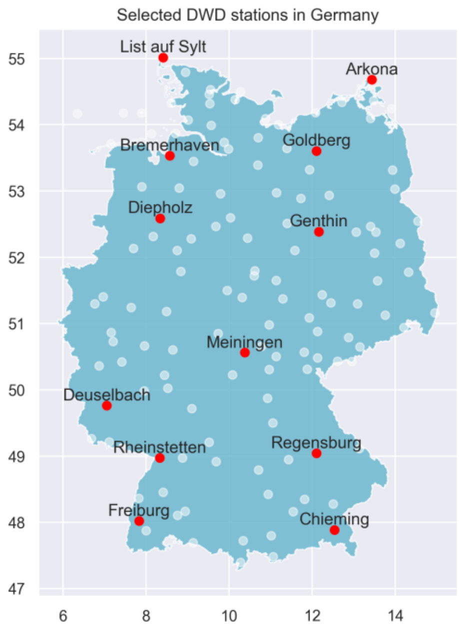
5. Iterative Model Development
Weather feature reduction

- Input variants:
  1. No weather input
  2. 4 Parameters of 12 DWD stations: 48 features
  3. Autoencoded input of 117 stations: 468 → 48 features
  4. PCA reduced input of 117 stations: 468 → 8 features

Variant 3:

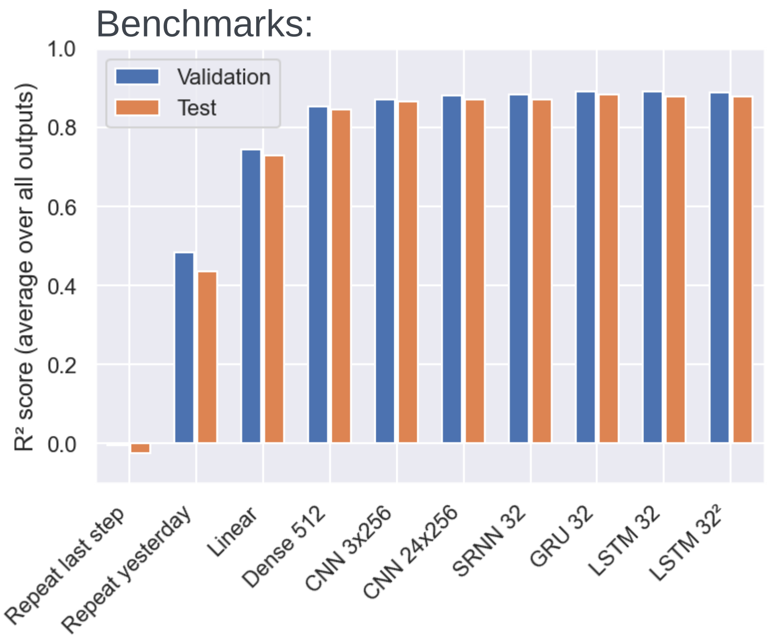


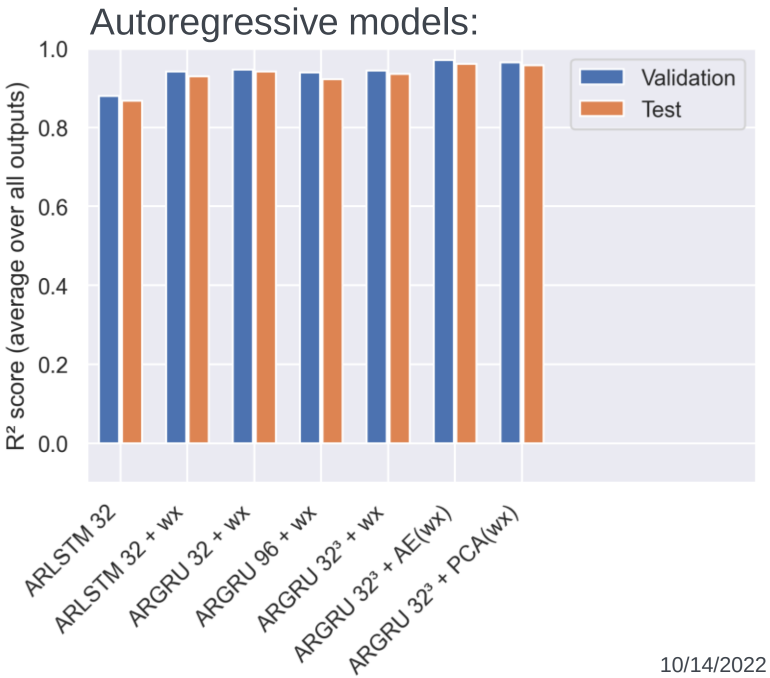
Selected stations:



5. Iterative Model Development
Performance Comparison

- Best performance: ARGRU 32^3 + AE(wx): R^2 0.962
  ARGRU 32^2 + PCA(wx): R^2 0.957

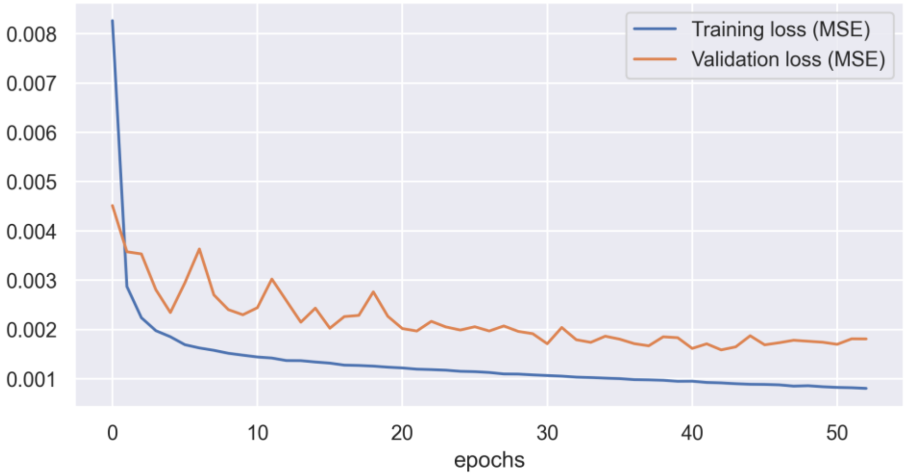




5. Iterative Model Development
Final Model Architecture

- Deep autoregressive GRU with continous input
- 3 layers [32, 32, 32]
- Inputs (24 time steps):
  - Solar and wind energy production (during past/warmup phase)
  - 8 time and sun elevation features
  - 8 PCA reduced weather features
- Prediction of 12 time steps

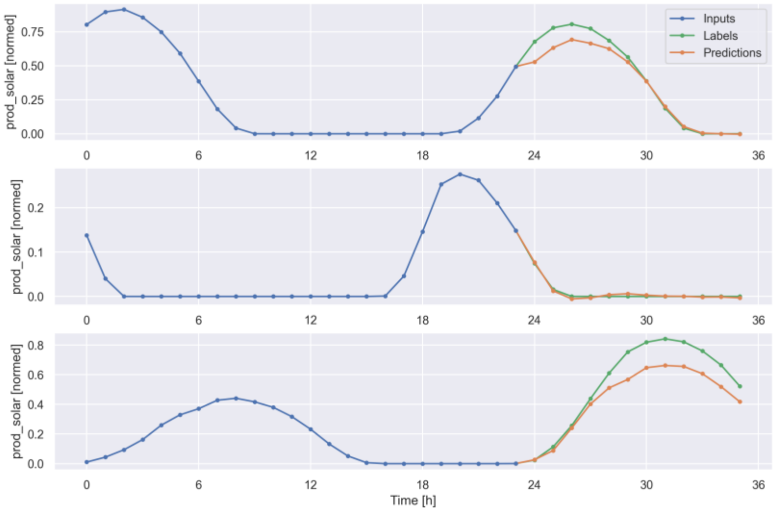
Loss plot:



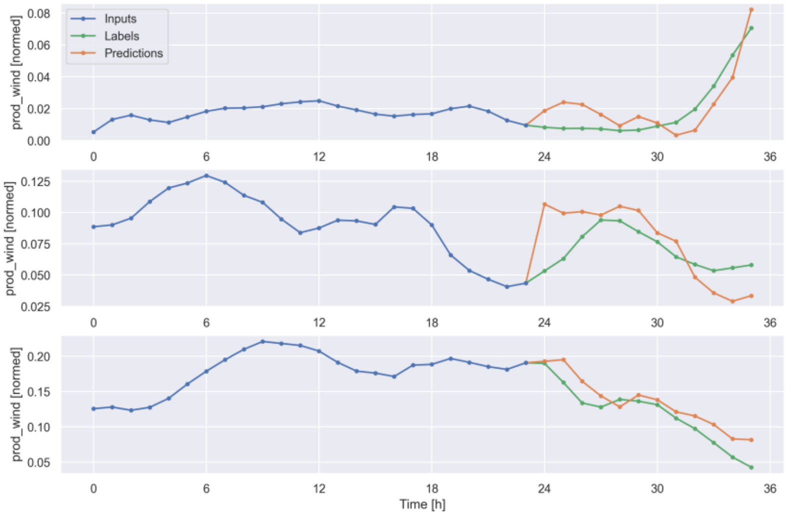
5. Iterative Model Development
Final Model Prediction Examples

- Better performance for solar energy predictions

Solar energy production:

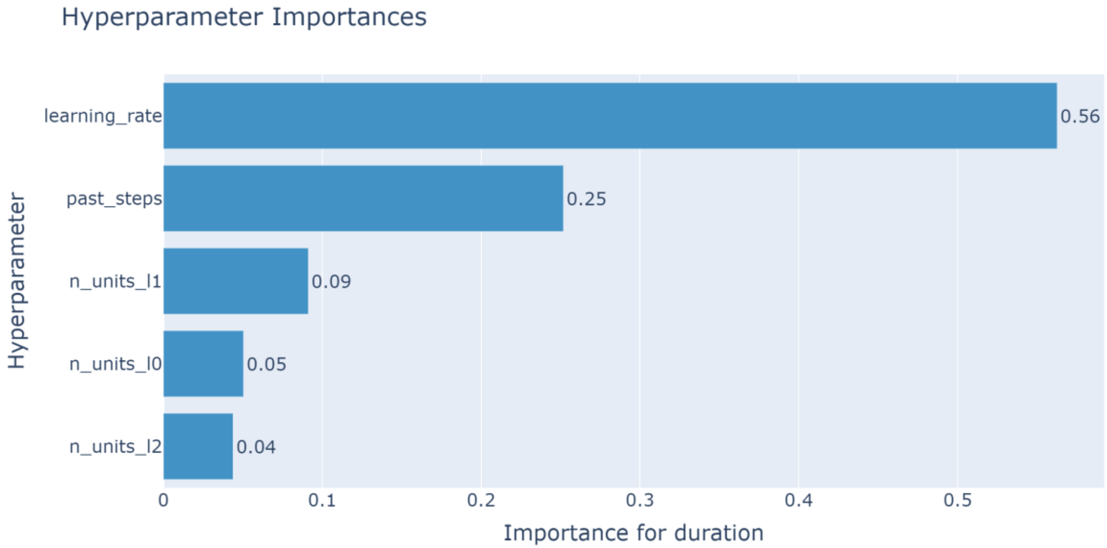


Wind energy production:



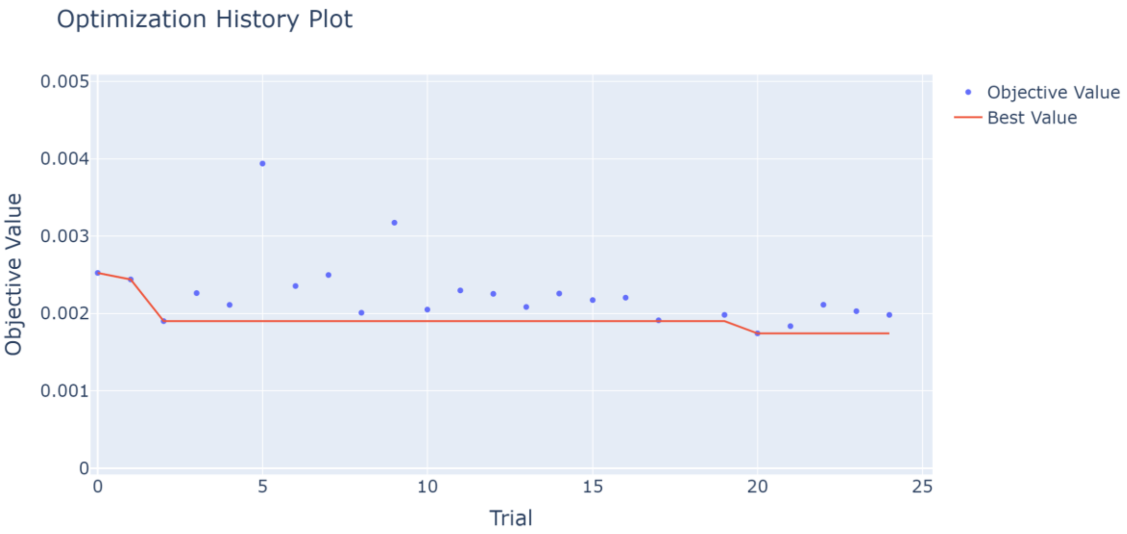
6. Hyperparameter Optimization

- Optuna: open source hyperparameter optimization framework
- Optimized parameters:
  - Learning rate
  - Units per layer
  - Past input time steps
- Optimization trials: 25



6. Hyperparameter Optimization
Optimization history and final parameters

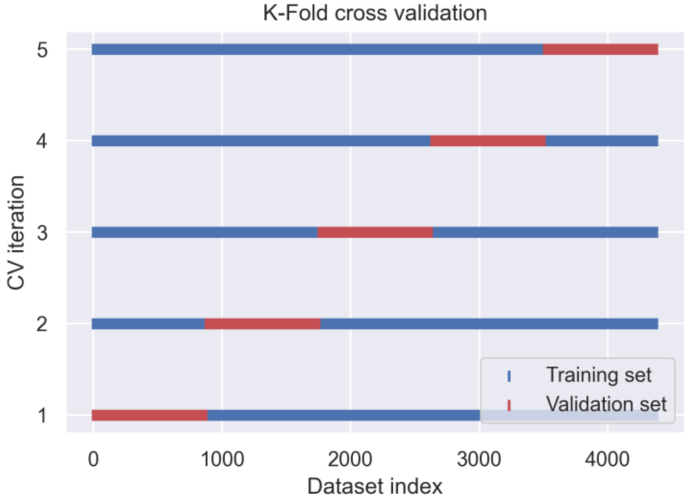
Parameter	Optimized Model	Previous Model
Past steps	92	24
Learning rate	1.7 e-3	5 e-4
Units 1st layer	15	32
Units 2nd layer	10	32
Units 3rd layer	11	32
Test R^2 score	0.9616	0.9567



7. Results
K-Fold cross validation

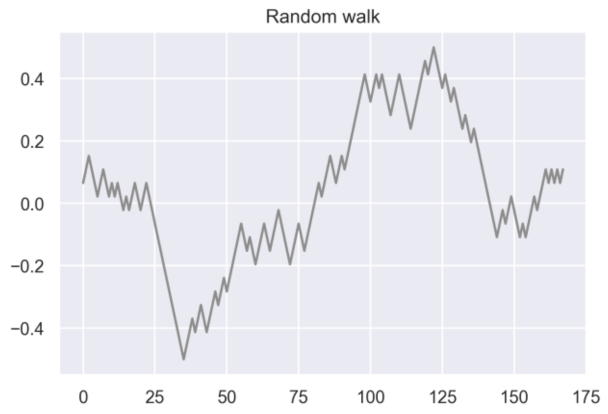
- The dataset was split into 5 subsets and the model was trained 5 times
- Each time a different subset was the validation set
- Results:

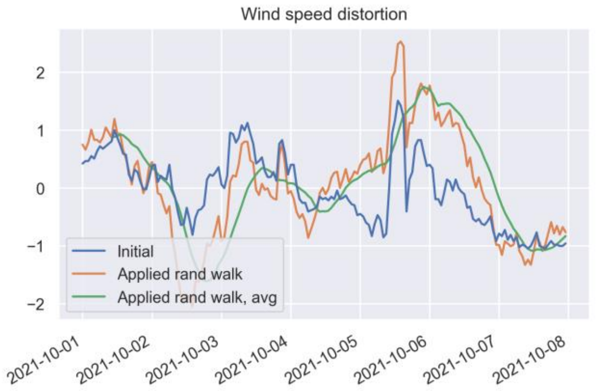
Fold	Val R^2	Test R^2
1	0.9617	0.9315
2	0.9690	0.9574
3	0.9686	0.9607
4	0.9659	0.9558
5	0.9516	0.9444
Average	**0.9633**	**0.9500**



8. Outlook

- Account for weather forecast inaccuracy:
  - Train with real forecast data
  - Distort weather data: Random walk?
- CNN Auto-encoder → Detect spatial relationships
- Employ statistic methods
  - e.g structural time series





University of Stuttgart
Institute for Statics and Dynamics

Thank you!

Dilara Yildiz, Luis Gentner, Leon Sengün

https://github.tik.uni-stuttgart.de/Leon-Senguen/Electricity-Prediction