

# Renewable Energy Generation Prediction

Using Weather Data and Machine Learning

Christoph Emeis, Jessica Voß, Hossam Oraby

22.01.2026

# Introduction

## Problem:

Accurately predict daily solar PV power generation on privately owned solar plants using weather (forecast) data

→ Regression Task using time series

Using own data collections

Time period: Jun 2023- Jan 2026

Locations: Dobersdorf & Elmshorn

Potential use case:  
When should I charge the electric car this week?

# Literature Review

<b>Paper name</b>	<b>Forecasting renewable energy for microgrids using machine learning (2025)</b>	<b>Towards Accurate Forecasting of Renewable Energy: Building Datasets and Benchmarking Machine Learning Models for Solar and Wind Power in France (2025)</b>
Description	Local generation prediction	Nationwide generation prediction using weighted weather maps
Models	CNN, LSTM	Random forest, Neural Network
Parameters/Variables	7 (without sunlight data)	10 (with solar radiation)
Focus of the analysis	Evalution metrics (RMSE, MSE, MAE)	Cross-validation (Hold-out, K-Fold, Blocking )
Best model	CNN	CNN
Reference	<i>Discov Appl Sci</i> <b>7</b> , 449 (2025)	arXiv:2504.16100

# Dataset Characteristics

- The solar production of two privately owned roof solar panels
- Obtained weather data from DWD (German Meteorological Service)
- Chose the closed weather stations
- Various measured parameters



# Dataset Characteristics

---

## Energy Production:

- Daily data (916 days)
- 0- 90 kWh daily generation



Temperature



Humidity



Precipitation



Cloud  
Coverage



Sunshine  
duration

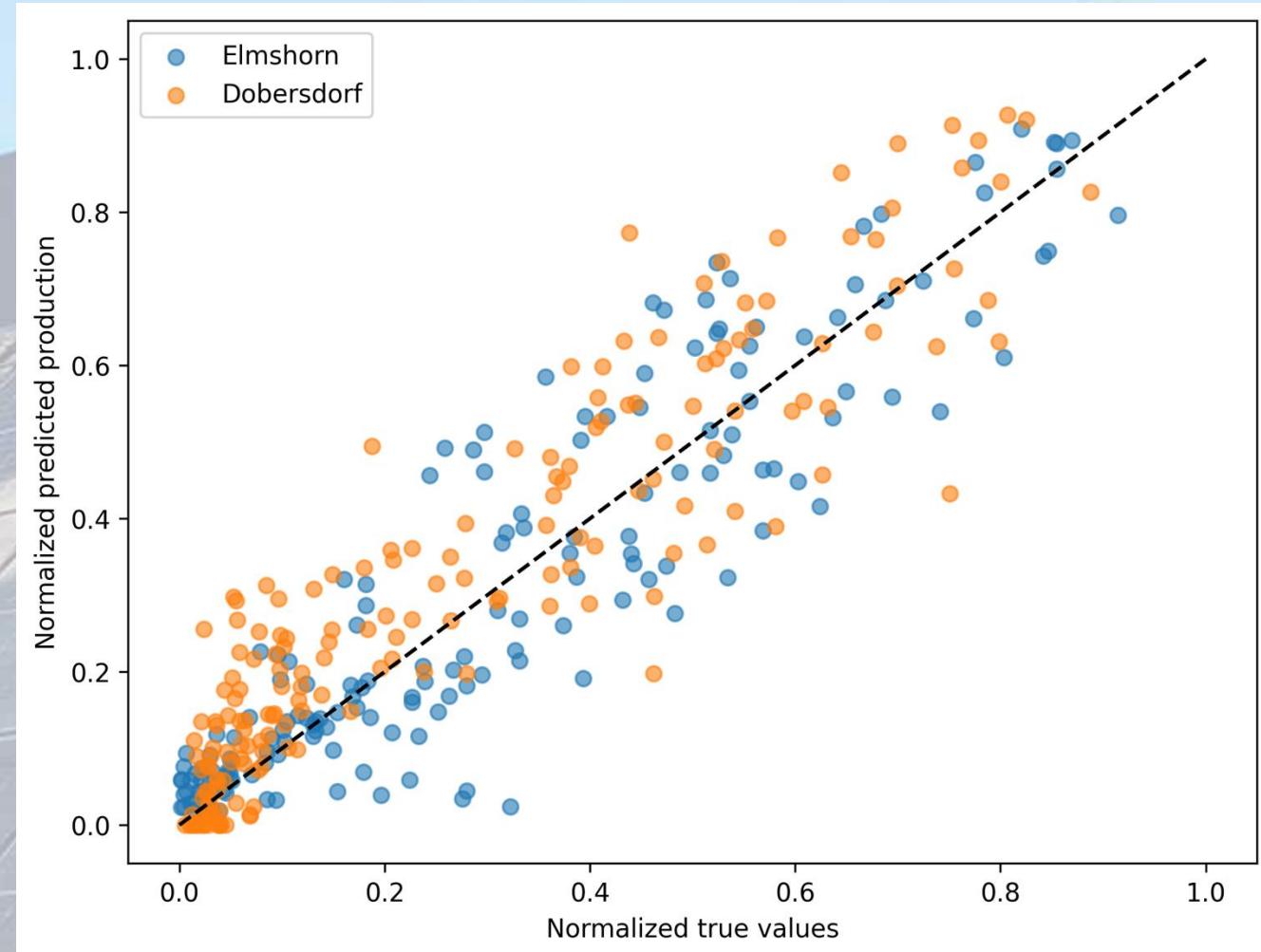


Pressure

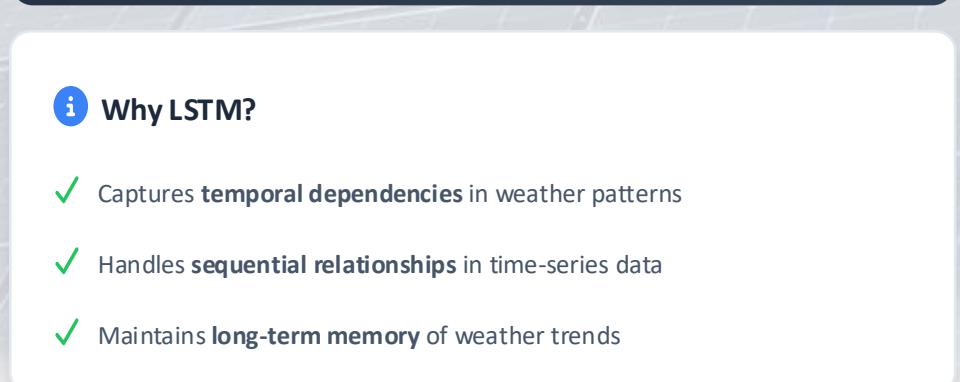
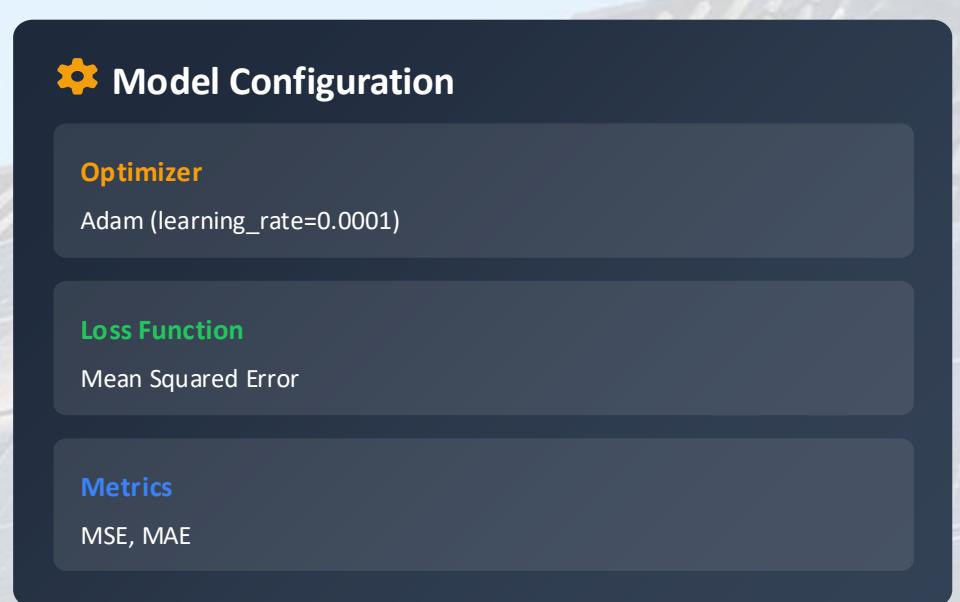
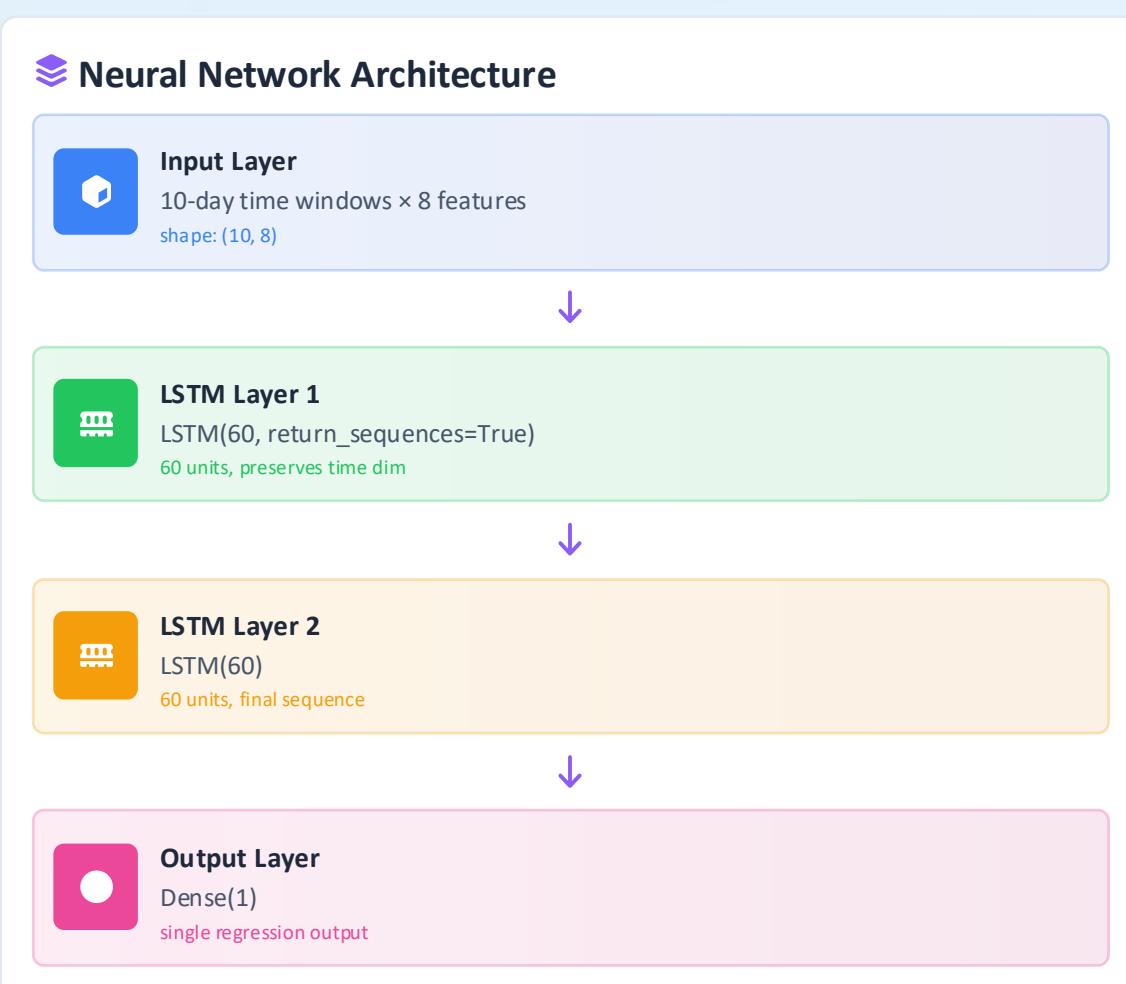
Missing values were set to 0

# Baseline Models

- Simple neural network:  
2 Input variables  
(mean temperature  
& sunlight duration)
- 1 hidden layer with 2 nodes,  
ReLU activation,  
MAE, ADAM, lr=0.001
- Training time < 4s
- Accuracy ~ 91%



# Advanced Model - LSTM Architecture



# Advanced Model - Hyperparameter tuning

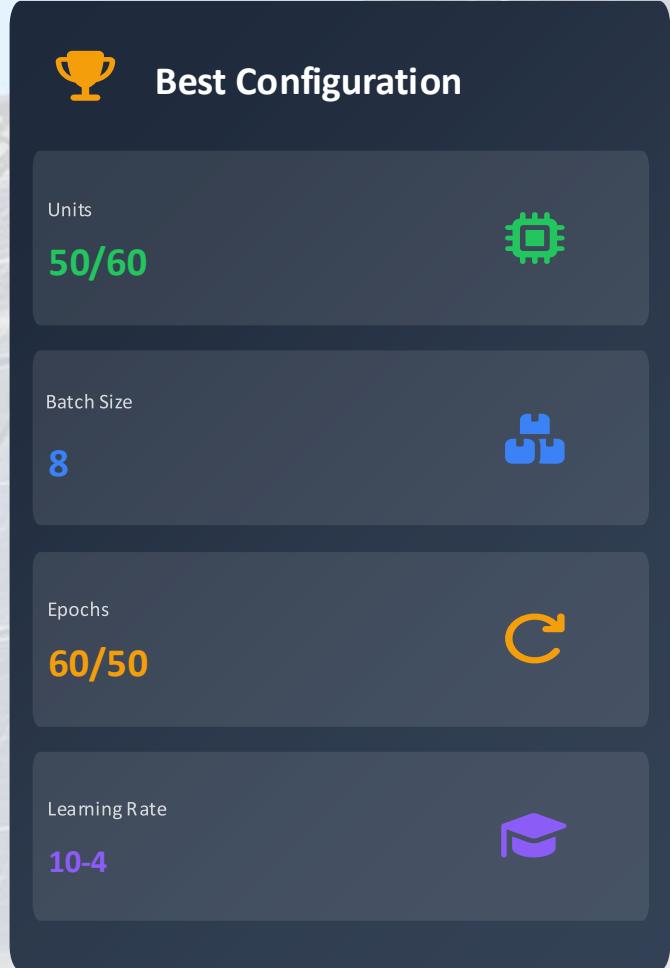
```
# Implement hyperparameter tuning

from scikeras.wrappers import KerasRegressor
from sklearn.model_selection import GridSearchCV
time_step = 10
def model1(units=40,learning_rate=0.1):
    model1 = Sequential()
    model1.add(Input(shape=(time_step, 8)))
    model1.add(LSTM(units, return_sequences=True))
    model1.add(LSTM(units))
    model1.add(Dense(1))
    model1.compile(optimizer = keras.optimizers.Adam( learning_rate=learning_rate), loss='mean_squared_error')
    return model1

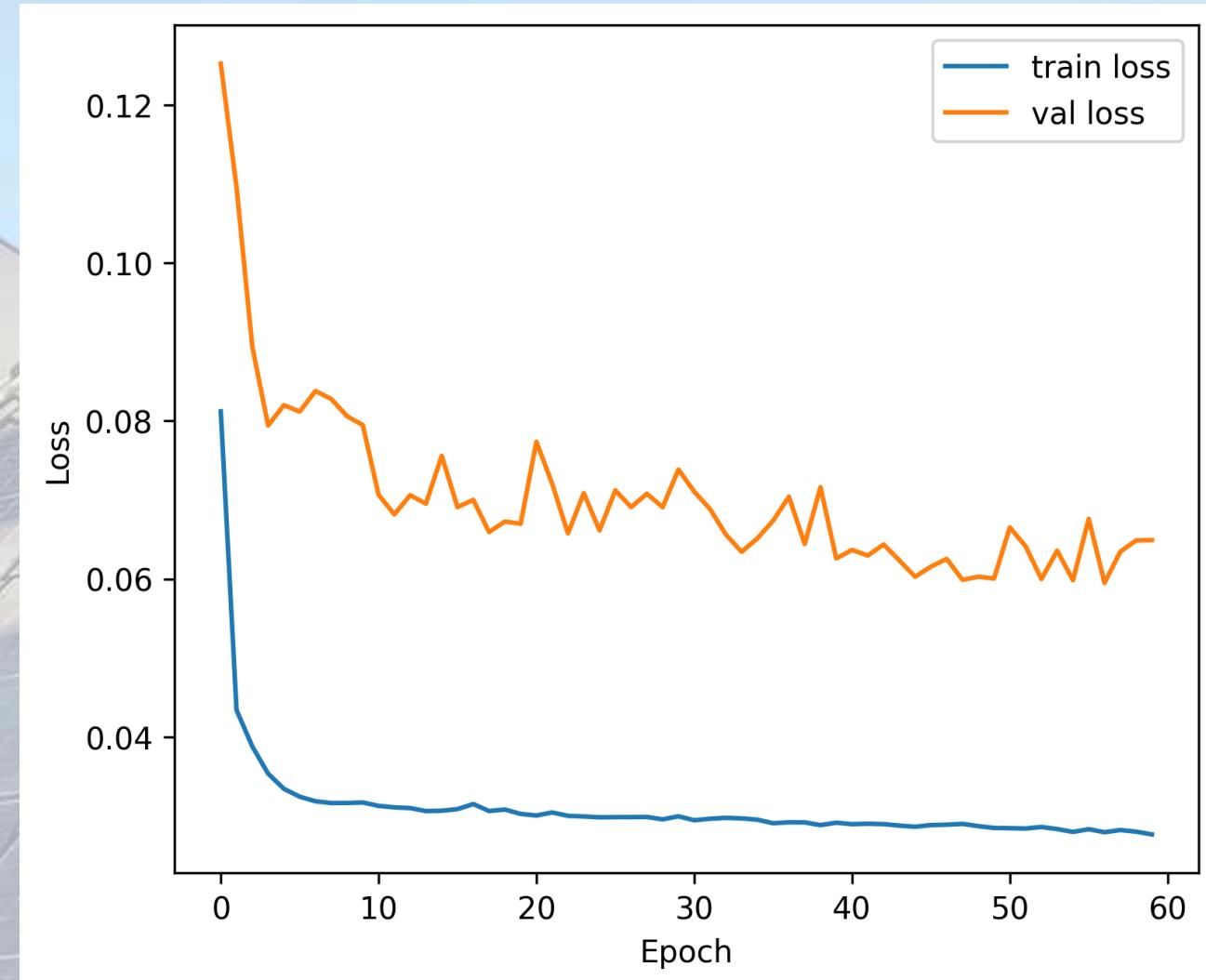
model = KerasRegressor(
    model=model1,
    verbose=0
)

param_grid = {
    "model__units": [50,60],
    "batch_size": [8,16],
    "epochs": [50,60],
    "model__learning_rate": [0.0001, 0.00001]
}

grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=5)
grid_search.fit(X, y)
```

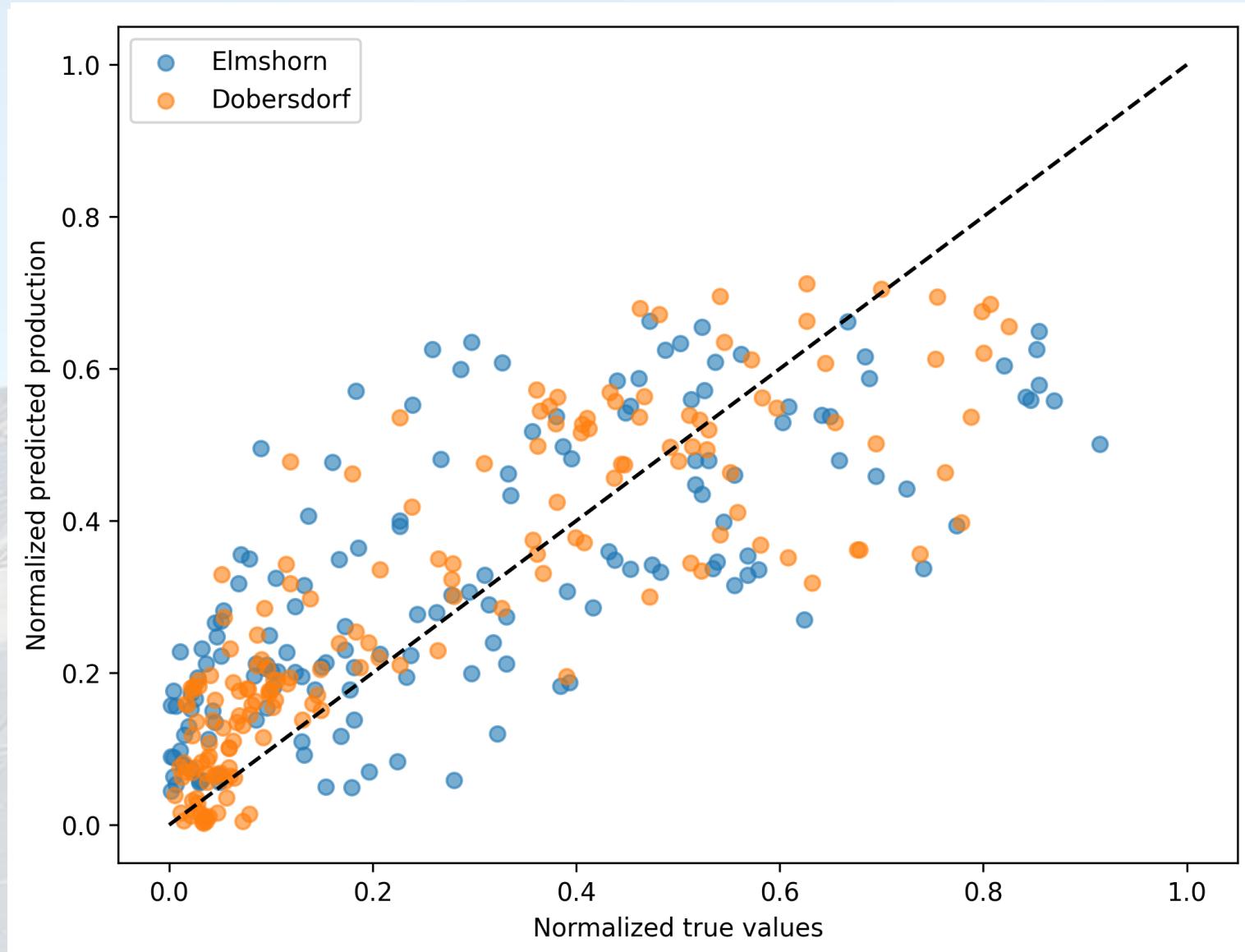


# Advanced Model – Training process



# Advanced Model Performance

- Training time < 20s
- Accuracy ~ 85-88 %
- Non-linear trend/ curved shape  
→ bias
- The models avoid predicting large value (>0.8)



# Challenges and Errors

Obtaining reliable data:

- Only few weather station in SH that measure sunlight duration
- We don't know how reliable the weather data is

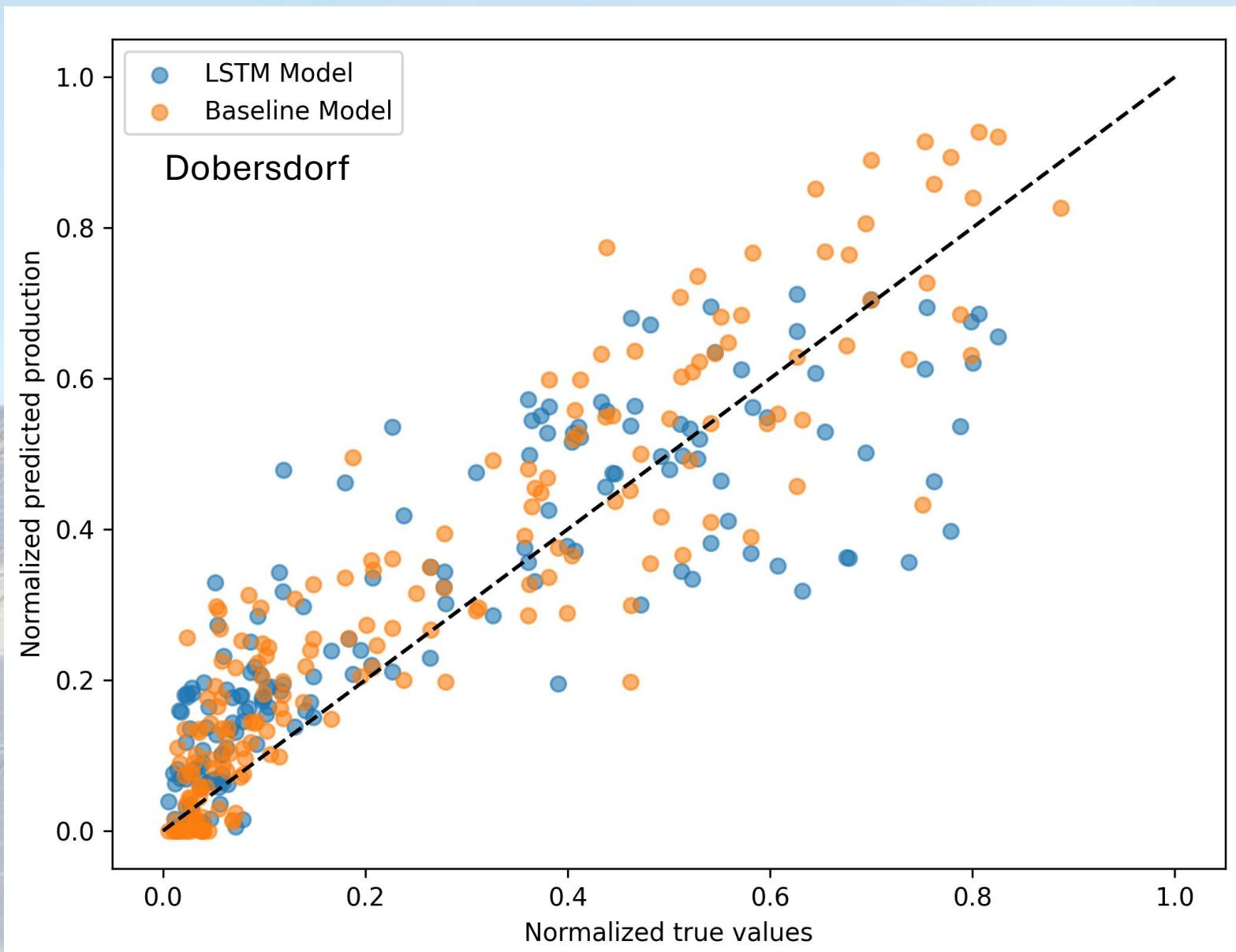
- Hyperparameter tuning:
- Unsuccessfully tried different ways for hyperparametertuning
- Data had wrong shape
- Took a long time

No comparability between our two locations

- We decided to treat Elmshorn and Dobersdorf independently

# Discussion

- Simple vs. Complex model
- Weak prediction of high and low solar production  
→ Not a big enough dataset?
- How will the model be affected when using actual forecast data ?



# Conclusion and Future Work

- Both LSTM and Baseline NN performed quite well  
→ Accuracy around 90%
- Baseline even outperformed the LSTM  
→ strongest features sunlight duration and temperature
- Improve data situation  
→ Use real forecast data for the actual locations
- Increase time-resolution  
→ Hourly or even shorter 10 min
- View whole regions and countries → SH or Germany



# Q&A

Thank you for your attention!

We are happy to answer questions 😊