Time Series Prediction of the NAO Index

Janika Rhein, Dikshyant Acharya, Tetiana Prykhodko and Ludwig Meder

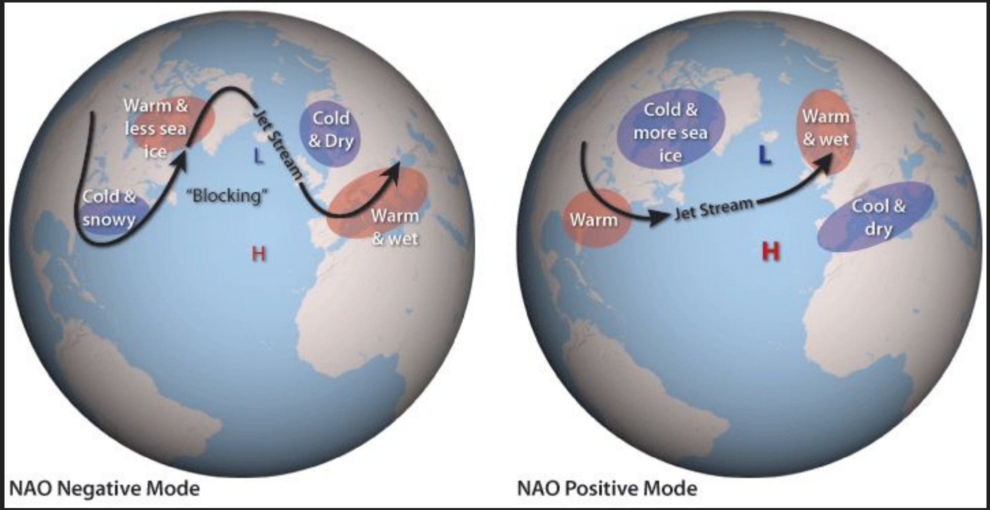
10.07.2024

Outline - delete later on

- What is the North Atlantic Oscillation (**NAO**)
  - Definition, both ways to calculate it
  - Importance (impact on society) -> motivation for prediction
  - Available forecasts, maybe machine learning approaches
- Data
  - time series characteristics: monthly/daily
  - showing them with example weather state
- Our approaches
  - monthly time series
    - our models
  - daily time series
    - our models
- Conclusion
  - Monthly data - not good (no former literature)
  - Daily data - better -> maybe best with rolling window approach

The North Atlantic Oscillation **(NAO)**

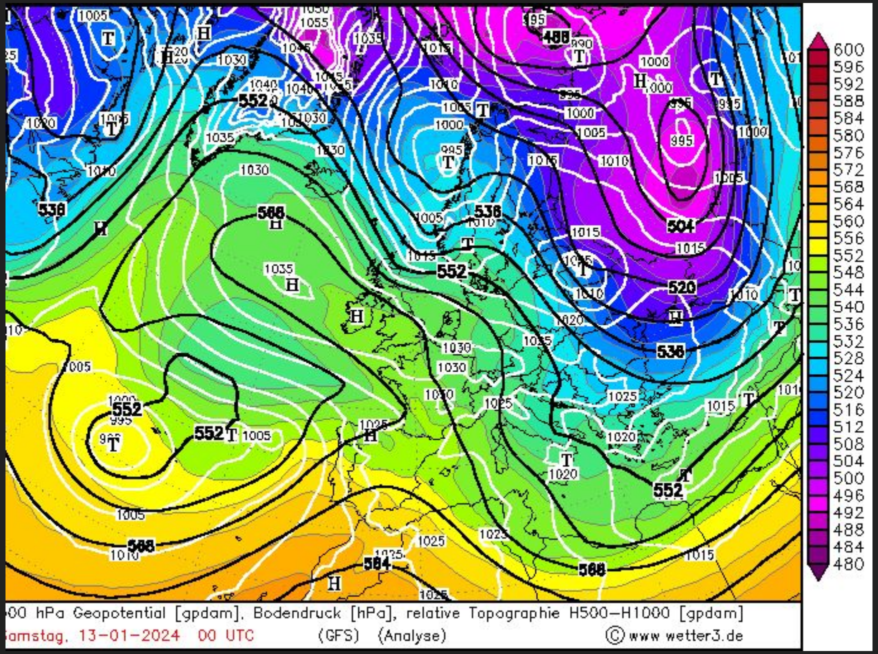
- Pressure difference between Azores High and Icelandic Low
- **3 Phases:**
    - Positive: Strongly developed pressure systems
    - Neutral: One pronounced pressure system
    - Negative: Low pressure difference or turn of signs



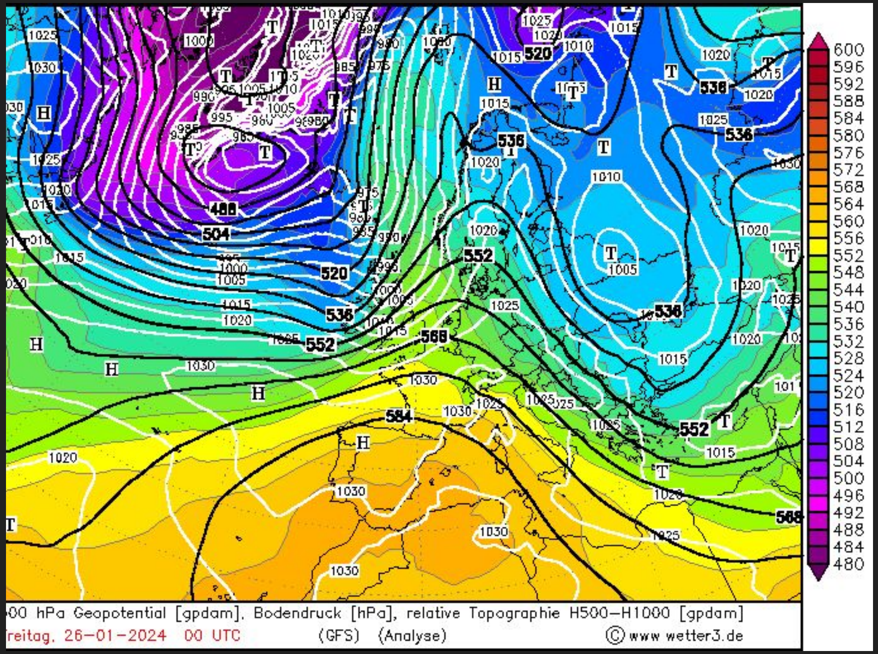
https://www.climate.gov/media/5522

NAO - Implications: An Example

NAO- 13.01.24: -3.93

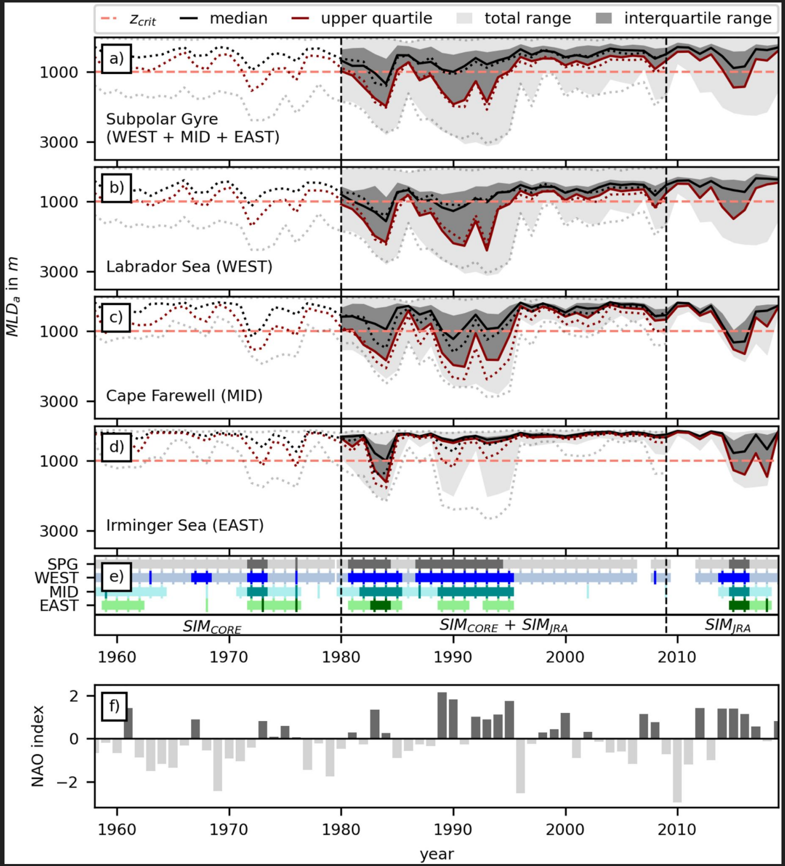


NAO+ 26.01.24: + 3.07



NAO - Implications

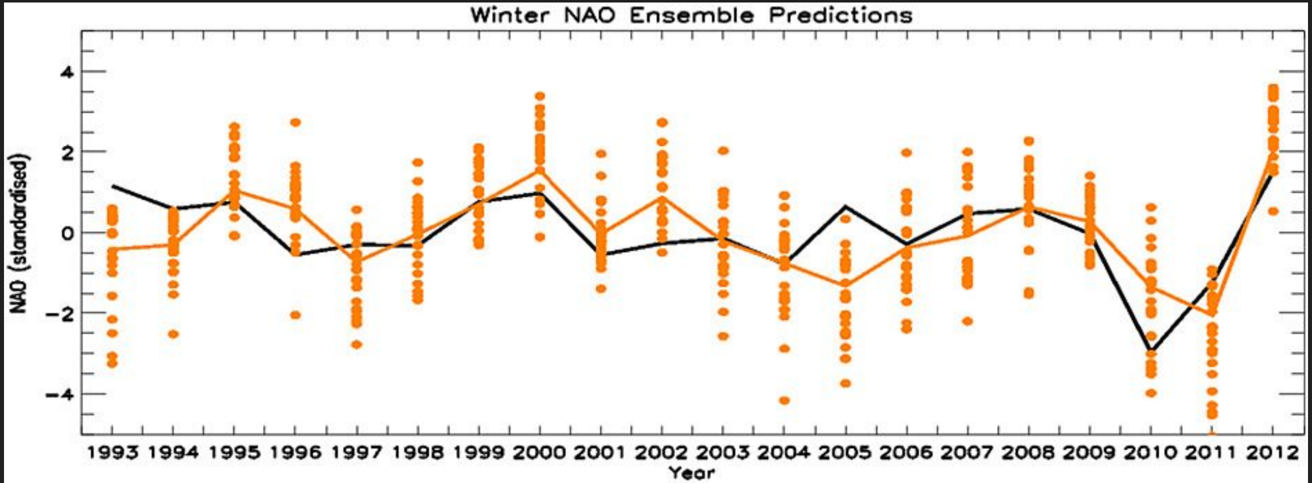
- European (North American) Weather
  - temperature and precipitation
  - heat waves/ cold spells
- Deep Convection
  - -> large scale ocean circulation



Rühs et al. (2019)

NAO - Numerical Forecast Example

- GloSea5 (based on HadGEM3)
- atm: 0.85°x0.55° (lon x lat), 85 vertical levels
- ocean: 0.25°, 75 vertical levels
- 24 ensemble member



Scaife et al. (2014)

Data

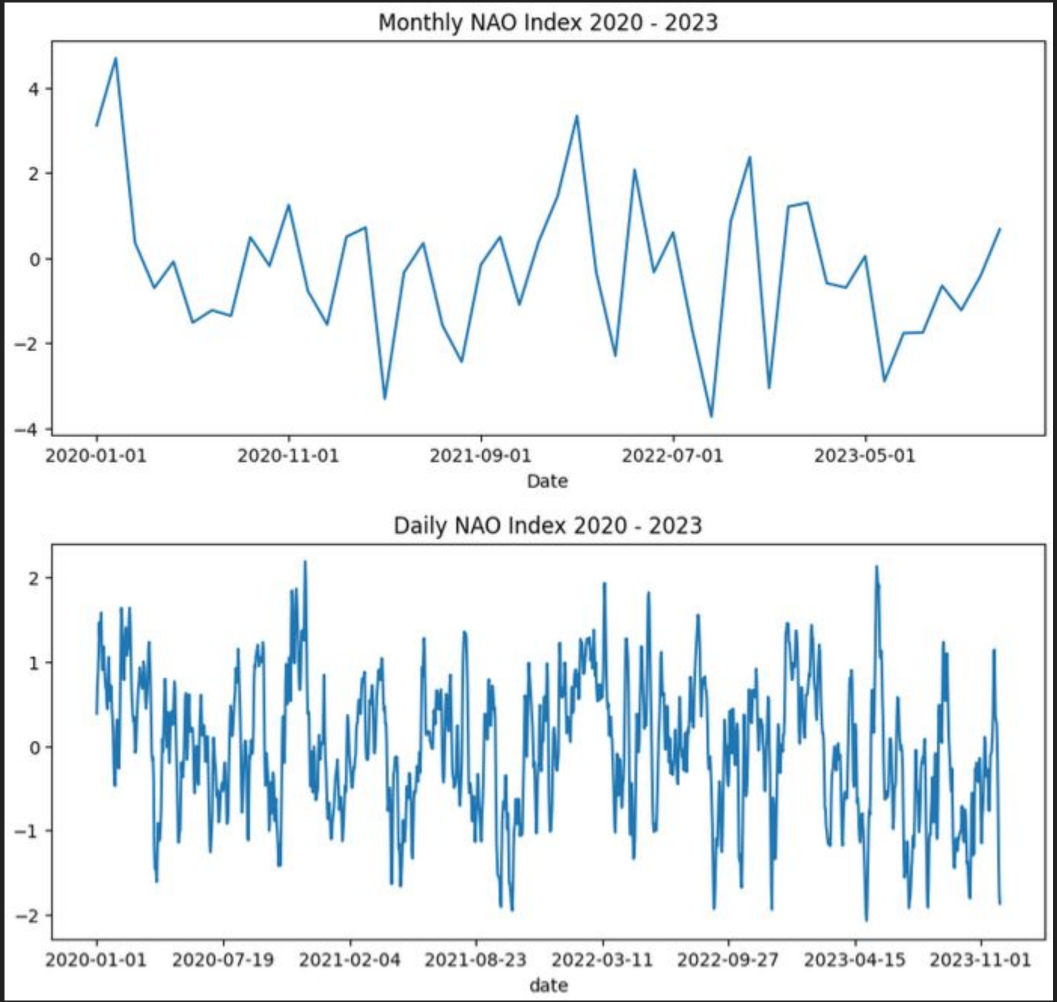
Monthly Dataset:

- 1825 - 2023
- SLP’ Azores - SLP’ Iceland

Daily Dataset:

- 01.01.1950 - 12.05.2024
- based on the first (rotated) Principal Component/EOF

SLP’: Normalised Pressure Anomaly



Evaluation Metrics

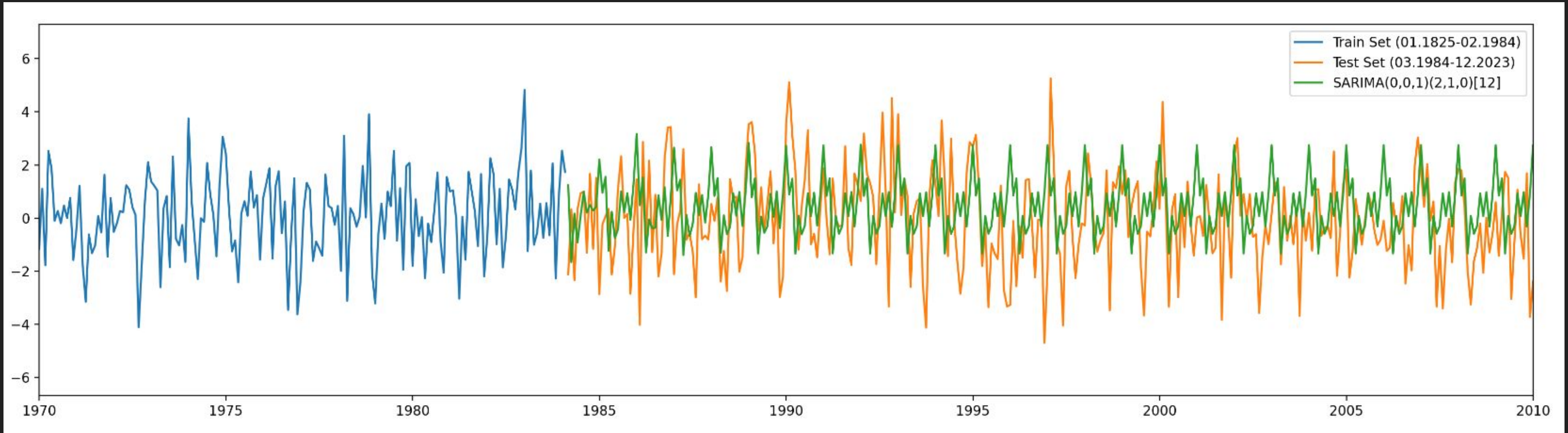
- Mean Squared Error
- Mean Absolute Error
- Median Absolute Error
- Phase Percentage

```
def phase_check(y_test,y_pred):
    """
    Args:
      y_test
      y_pred
    Returns: phase agreement of the two time series in percentage
    """
    y_test_phases = np.zeros(len(y_test))
    y_pred_phases = np.zeros(len(y_pred))
    negative_test = np.where(y_test <-1)[0]
    negative_pred = np.where(y_pred <-1)[0]
    positive_test = np.where(y_test >1)[0]
    positive_pred = np.where(y_pred >1)[0]
    y_test_phases[negative_test] = -1
    y_pred_phases[negative_pred] = -1
    y_test_phases[positive_test] = 1
    y_pred_phases[positive_pred] = 1
    counter = np.sum(y_test_phases==y_pred_phases)
    phase_percent = 100/len(y_test) * counter
    return phase_percent
```

Baseline: SARIMA

- monthly dataset
- 80% for training, 20% for testing/predicting
    - -> 478 months forecast
- SARIMA(0,0,1)(2,1,0)[12]

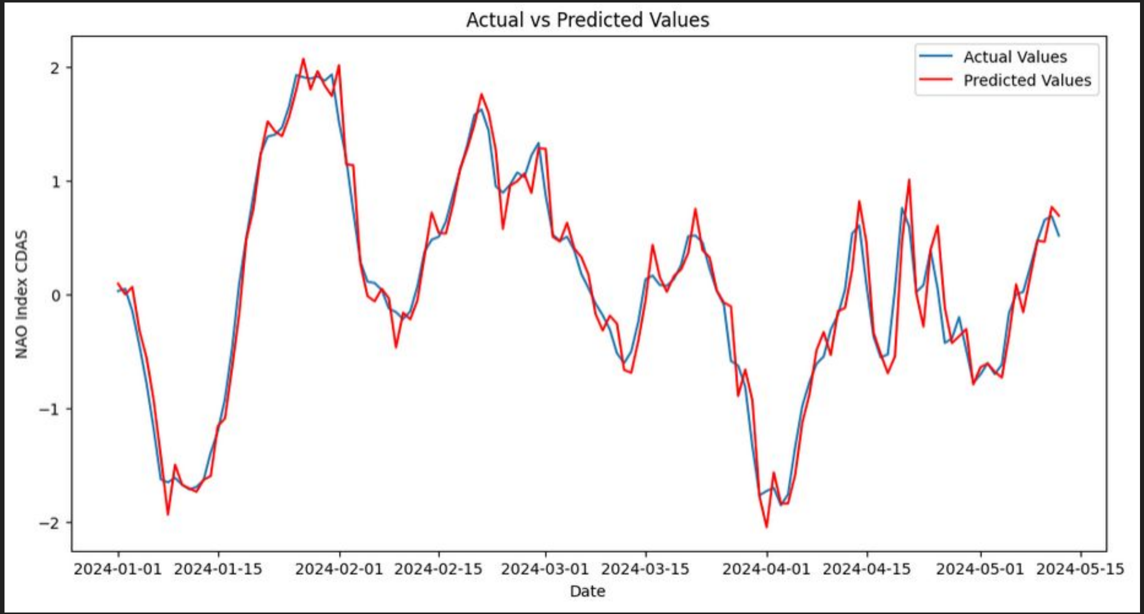
Metrics for Testset:
MSE: 4.01
Phase Percent: 38.08%



Baseline: ARIMA

- daily dataset
  - 1950-2024 -> Training
  - 2024- 05.2024 -> Validation
- 60 Data to predict 1 data

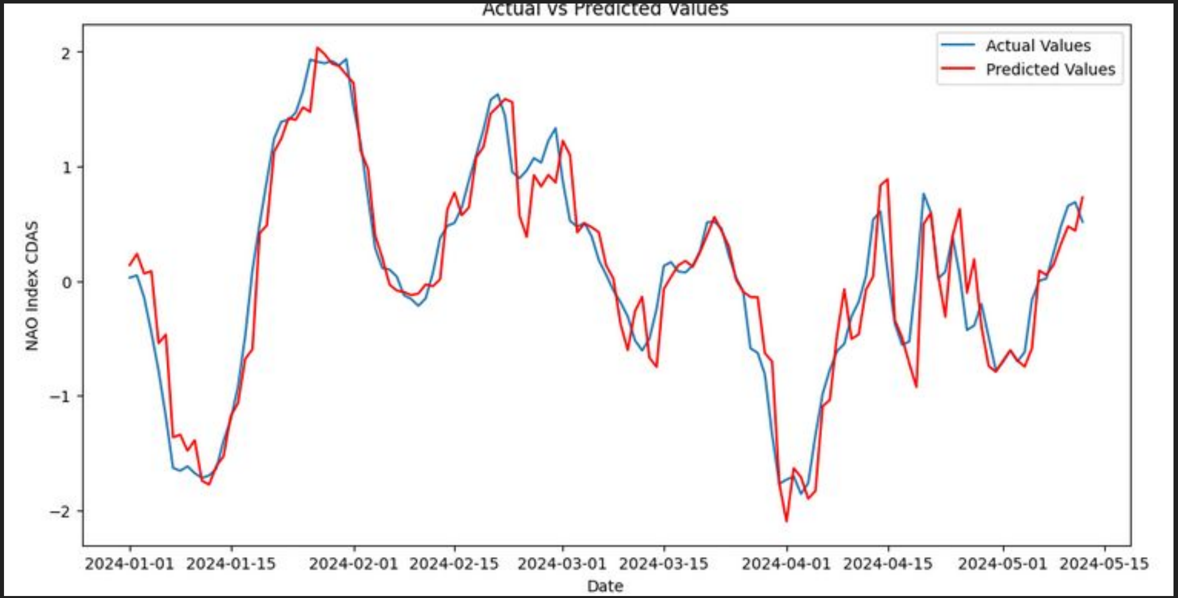
Metrics for Validation Set:
MAE: 0.14



ARIMA

- daily dataset
  - 1950-2024 -> Training
  - 2024- 05.2024 -> Validation
- 100 Data to predict 2 data

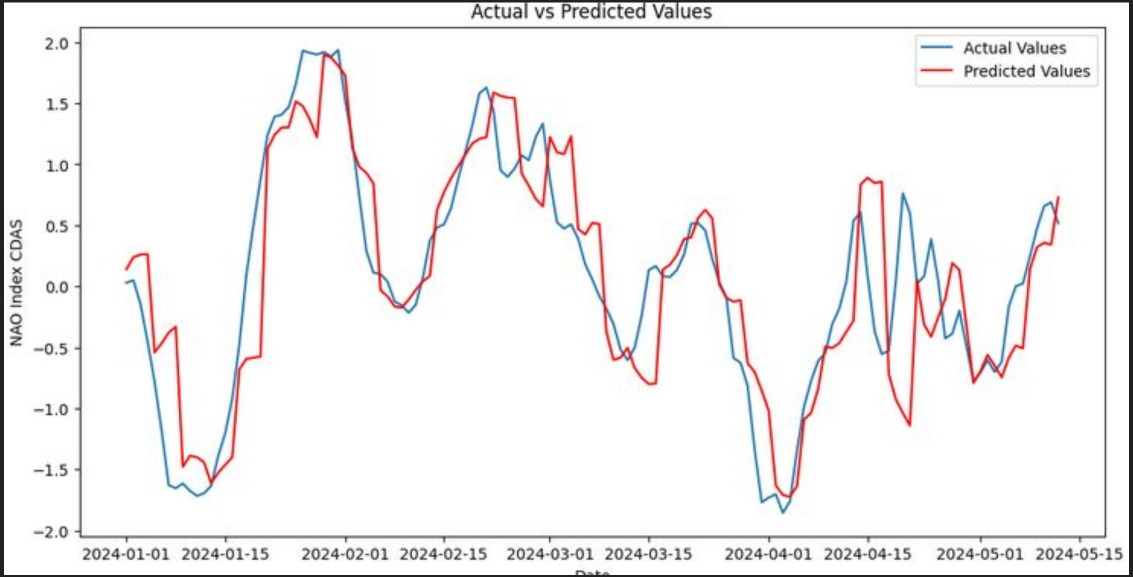
Metrics for Validation Set:
MAE: 0.2



ARIMA

- daily dataset
  - 1950-2024 -> Training
  - 2024- 05.2024 -> Validation
- 100 Data to predict 4 data

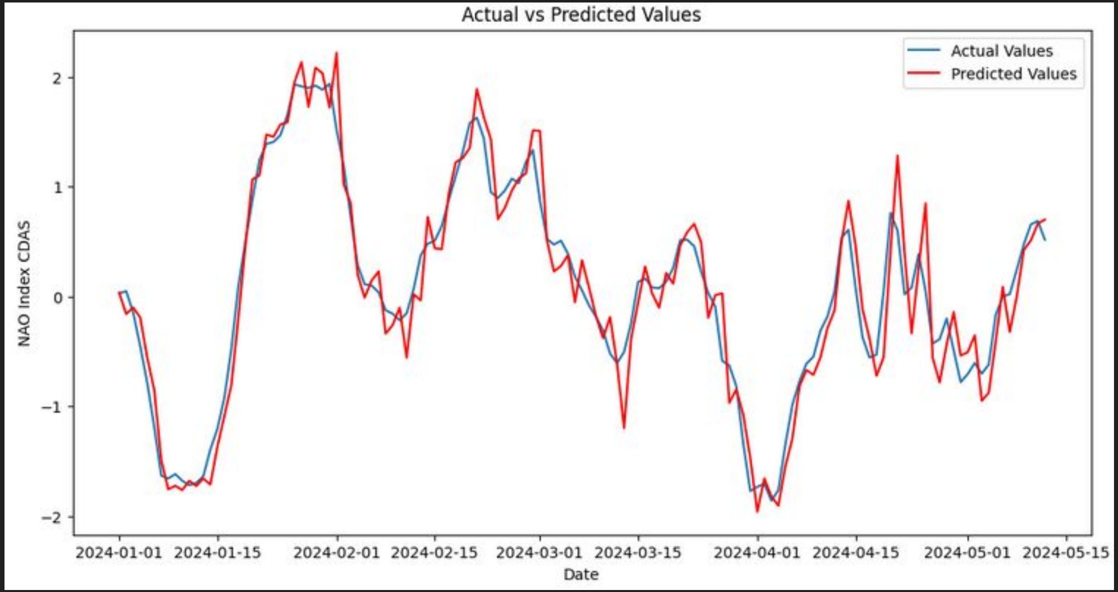
Metrics for Validation Set:
MAE: 0.36



SARIMA

- daily dataset
  - 1950-2024 -> Training
  - 2024- 05.2024 -> Validation
- 60 Data to predict 1 data

Metrics for Validation Set:
MAE: 0.19



LSTM

- daily dataset
    - 1950-2024 -> Training
    - 2024- 05.2024 -> Validation
- 50 Data to predict 1 data

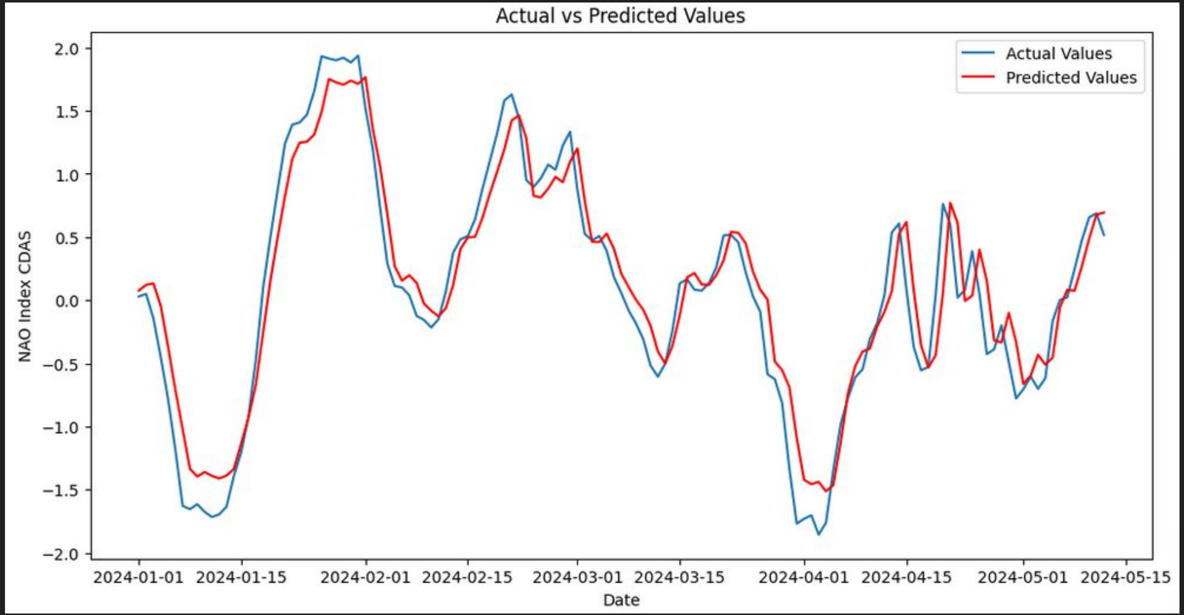
```
# Build the enhanced Bidirectional LSTM model
model = Sequential()
model.add(Bidirectional(LSTM(100, activation='relu', return_sequences=True),
                        input_shape=(X_train.shape[1], 1)))
model.add(Dropout(0.2))
model.add(Bidirectional(LSTM(100, activation='relu')))
model.add(Dropout(0.2))
model.add(Dense(50, activation='relu'))
model.add(Dense(future_points))

model.compile(optimizer='adam', loss='mse')
```

LSTM

- daily dataset
  - 1950-2024 -> Training
  - 2024- 05.2024 -> Validation
- 60 Data to predict 1 data

Metrics for Validation Set:
MAE: 0.18



GRU

- daily dataset
    - 1950-2024 -> Training
    - 2024- 05.2024 -> Validation
- 60 Data to predict 1 data

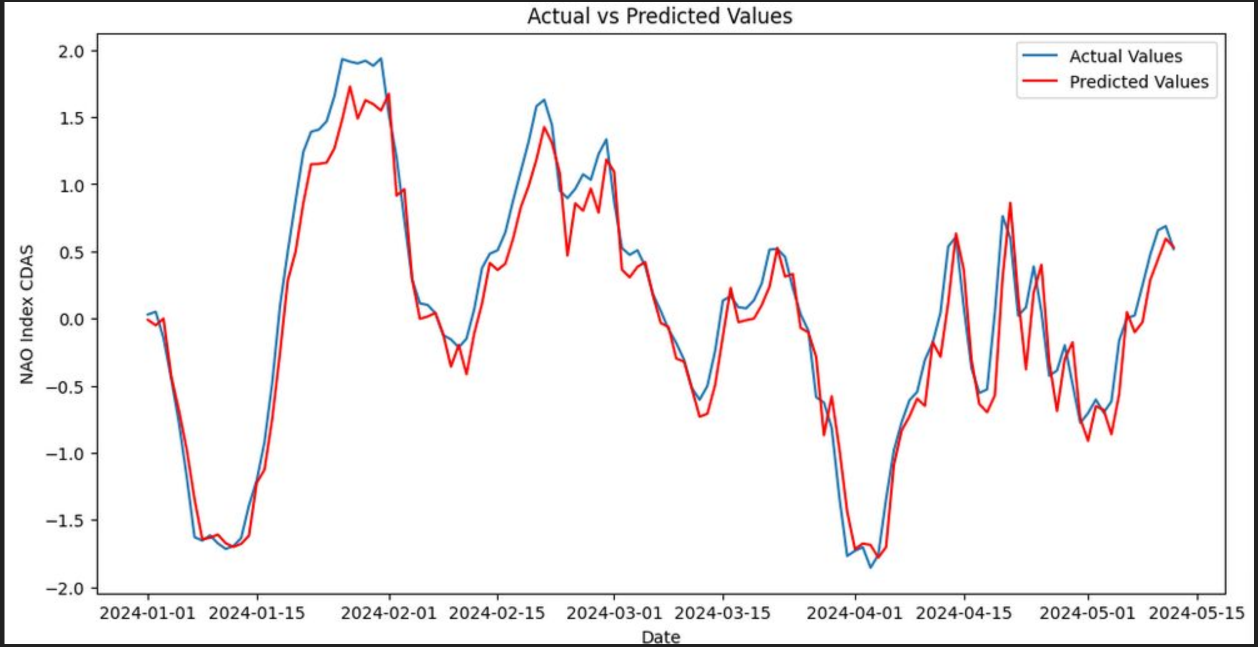
```python
# Build the GRU model
model = Sequential()
model.add(GRU(100, activation='relu', return_sequences=True,
              input_shape=(X_train.shape[1], 1)))
model.add(Dropout(0.2))
model.add(GRU(100, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(50, activation='relu'))
model.add(Dense(future_points))

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.001), loss='mse')
```

GRU

- daily dataset
  - 1950-2024 -> Training
  - 2024- 05.2024 -> Validation
- 60 Data to predict 1 data

Metrics for Validationset:
MAE: 0.18



NBeatsNet with monthly / daily data

- 24 data to predict one entry
- epochs: 50
- learning rate: 0.001
- training / val / test = 60 / 20 / 20

Monthly

```
Metrics for Test Set:
Mean squared error: 0.1583
Phase Percent: 99.7802
```

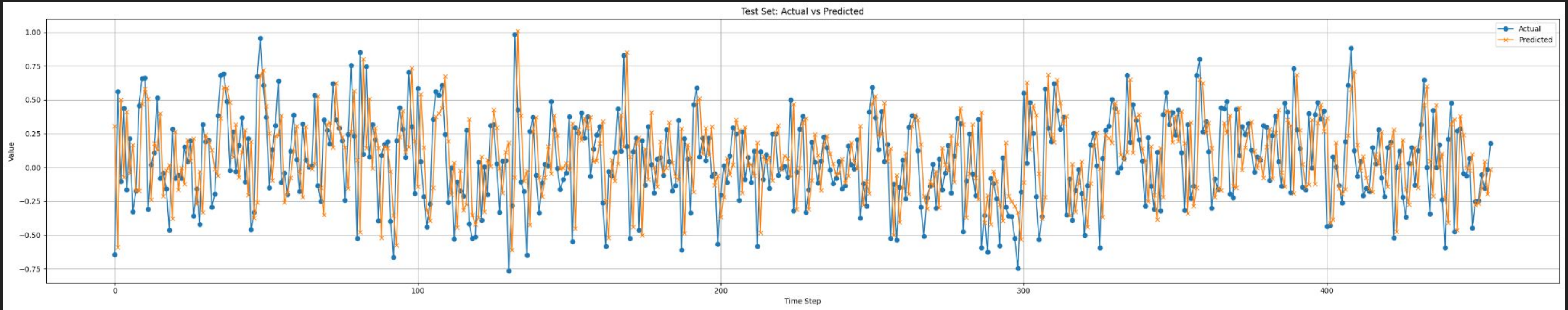
Daily

```
Metrics for Test Set:
Mean squared error: 0.0135
Phase Percent: 99.9076
```

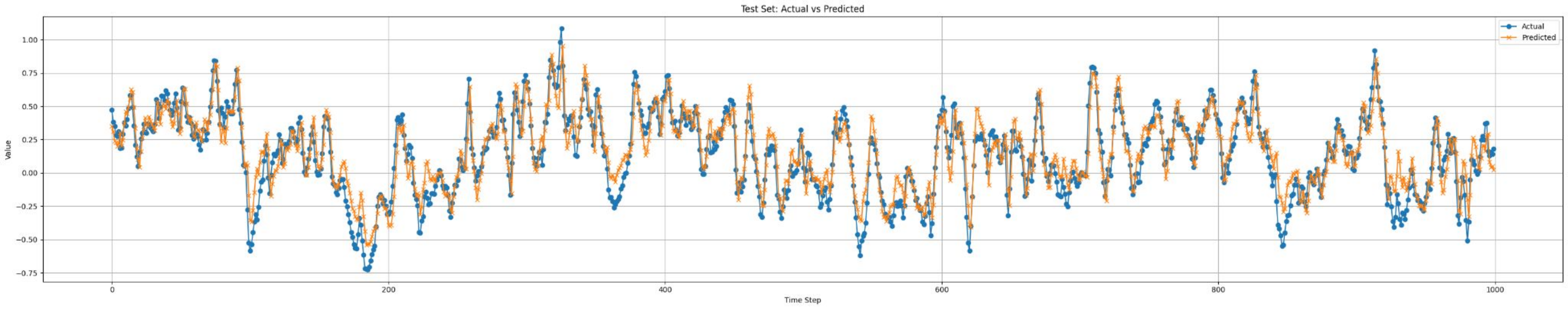
```
model = NBeatsNet(stack_types=(NBeatsNet.GENERIC_BLOCK, NBeatsNet.GENERIC_BLOCK),
                  forecast_length=1,
                  backcast_length=input_length,
                  hidden_layer_units=64,
                  nb_blocks_per_stack=3,
                  device=device)

model.to(device)
criterion = nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)
```

NBeatsNet with monthly data

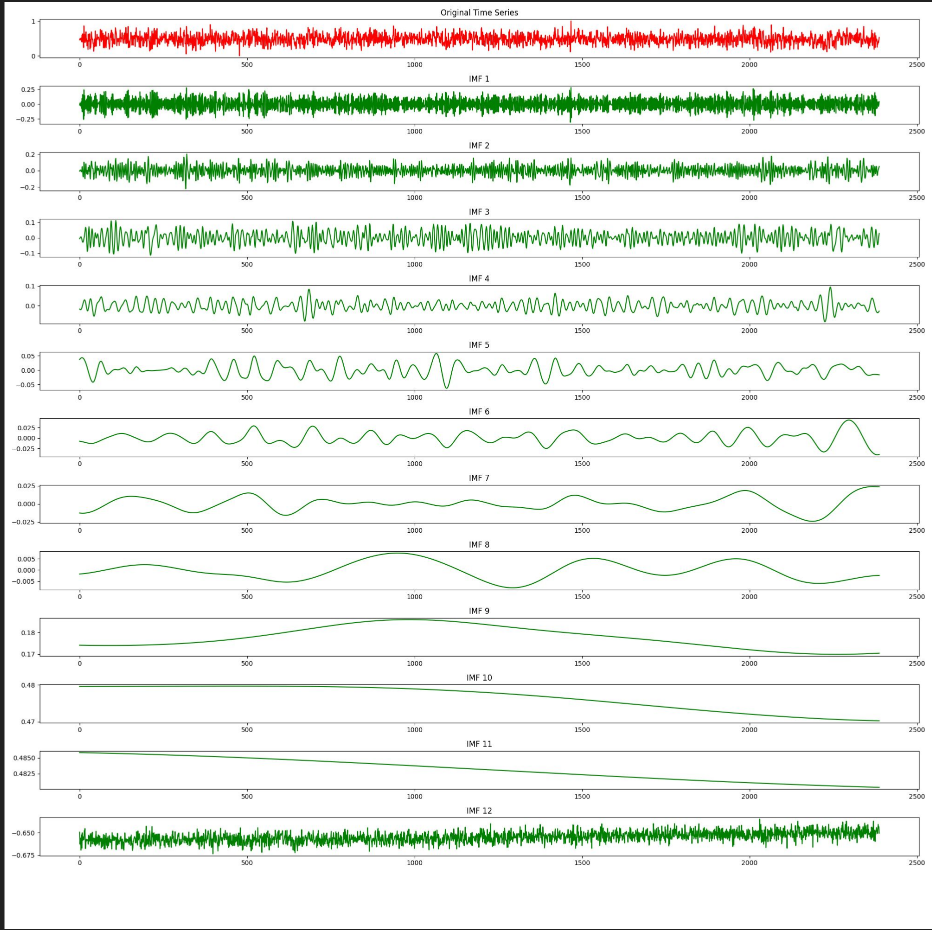


NBeatsNet with daily data (section of 1000 days)



EEMD and XGBoost

- ensemble empirical mode decomposition
- decomposes TS in intrinsic mode functions (IMFs) and residuals
- increases model performance!



EEMD and XGBoost

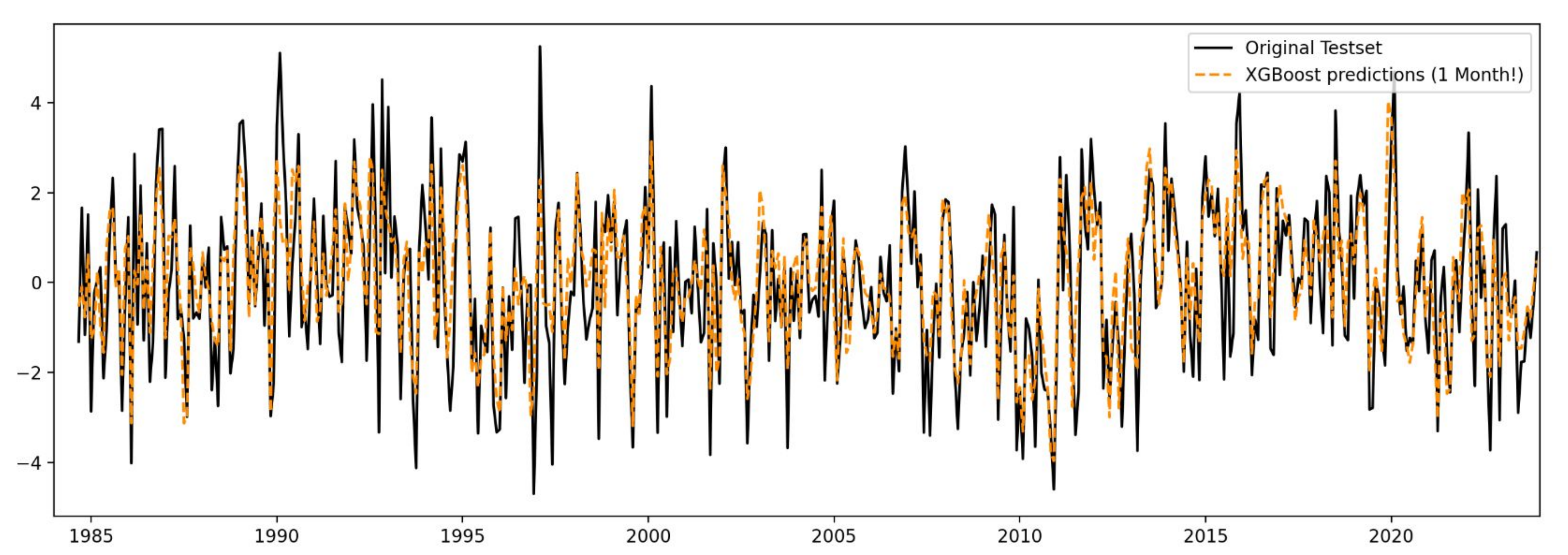
- 24 months to predict 1
- fine tuning did not improve the model…

Metrics for Testset:
MSE: 1.22
Phase Percent: 70.55%

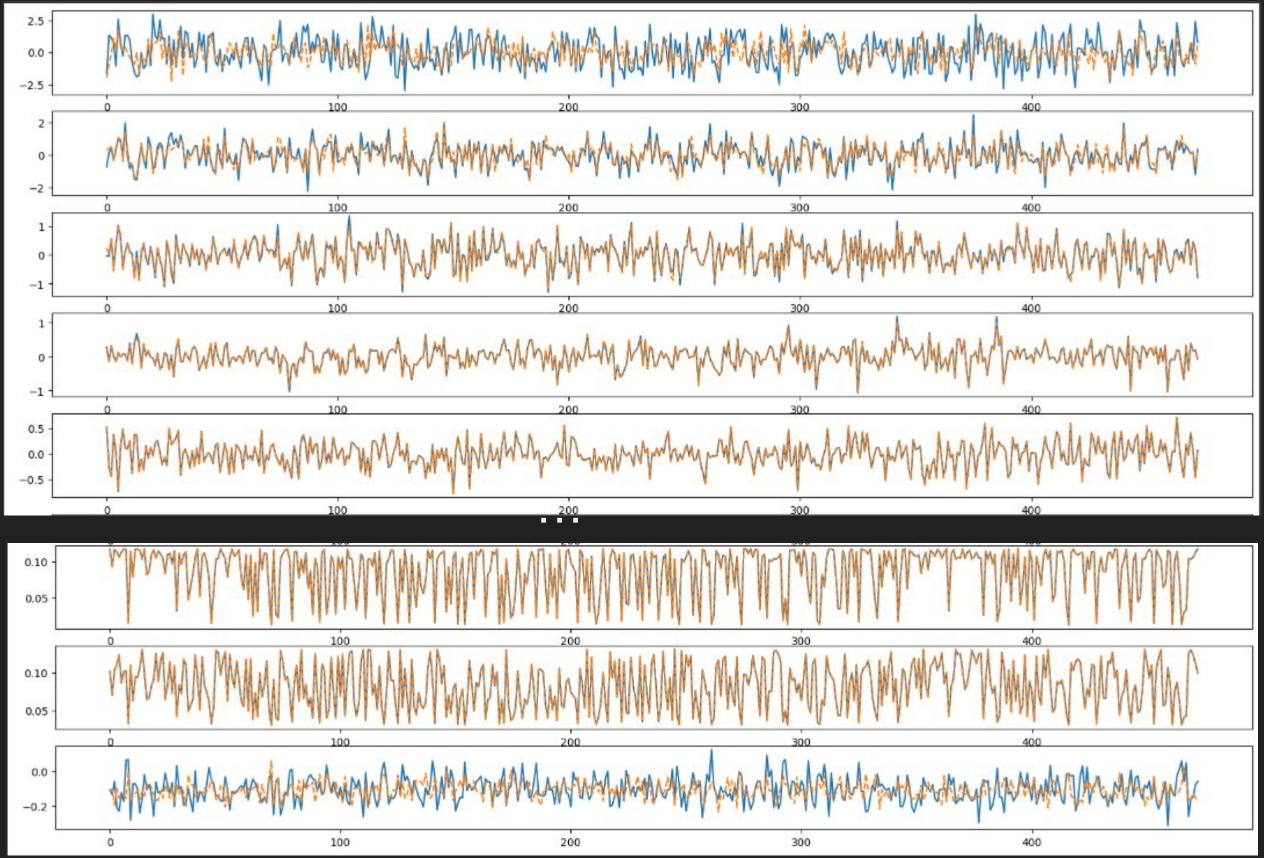
```
XGBRegressor

XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)
```

EEMD and XGBoost



EEMD and XGBoost



IMF1

IMF2

IMF3

IMF4

IMF5

IMF10

IMF11

Residuals

Extreme Learning Machine (ELM) with daily data

ELM:

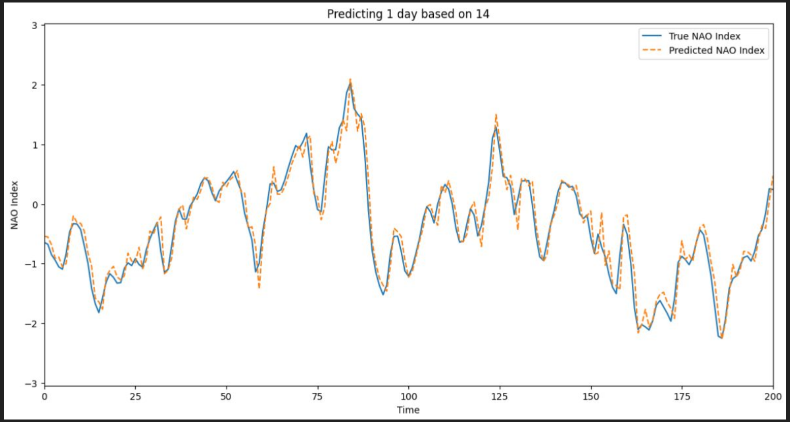
- one hidden layer
- weights and biases: randomly assigned
- extreme fast training process due to least-square solutions of output weights

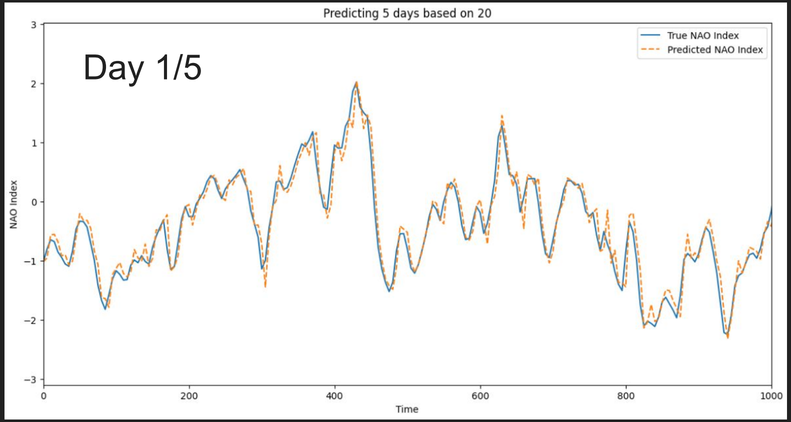
- 20 days to predict one

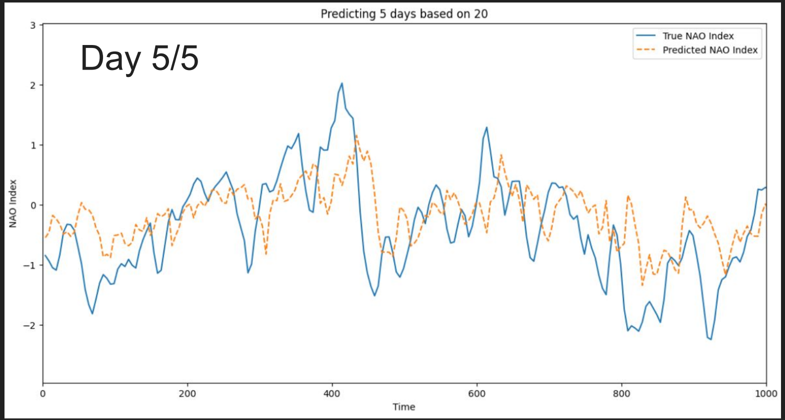
```
# Initialize and train the ELM model
elm = ELM(X_train_flat.shape[1], y_train_flat.shape[1],batch=400)
elm.add_neurons(128, 'sigm') # , 'tanh'
elm.train(X_train_flat, y_train_flat, 'MSE')#, 'V', Xv=X_val_flat, Tv=y_val_flat)
```

MSE: 0.04, MAE: 0.16, Phase Ratio: 91.6%

Extreme Learning Machine (ELM) with daily data







Model	Type	Finetuning	Normalization	Preprocessing	External TS	Frequency	Input length	Output length	MSE	Phase
XGBoost	ens_learning	None	True	None	None	Monthly	24	1	3.85	38.35
XGBoost	ens_learning	None	True	EEMD	None	Monthly	24	1	1.22	70.55
ELM	NN	10 neurons	True	None	None	Monthly	12	1	3.28	38.2
ELM	NN	73 neurons	True	None	El Nino Index	Monthly	36	1	3.4	39.1
ELM	NN	64 neurons	True	None	None	Daily	14	1	0.045	91.7
ELM	NN	128 neurons	True	None	None	Daily	14	5	0.275	80.7
ELM	NN	128 neurons	True	None	None	Daily	62	14	0.51	76.4
ELM	NN	128 neurons	True	None	None	Daily	123	31	0.63	75.1
LSTM	RNN	64 neurons	True	None	None	Daily	48	1	0.48	91.3
LSTM	RNN	64 neurons	True	None	None	Daily	48	5	0.27	73.0
NBeats	NN	None	True	True	None	Daily	24	1	0.013	99.90
NBeats	NN	None	True	True	None	Monthly	24	1	0.158	99.78

Outlook

- Verification of results
- Comparing models more systematically (daily/monthly data, EEMD/no EEMD)
- Combining the best models & methods
- Using rolling forecast

References

- Weather maps: https://www.wetter3.de/Archiv/
- NAO index values: https://verstat.no/nao-index-daily-monthly-and-yearly-average/
- Rühs, S., Oliver, E. C., Biastoch, A., Böning, C. W., Dowd, M., Getzlaff, K., ... & Myers, P. G. (2021). Changing spatial patterns of deep convection in the subpolar North Atlantic. *Journal of Geophysical Research: Oceans*, *126*(7), e2021JC017245.
- Scaife, A. A.; Arribas, A.; Blockley, E.; Brookshaw, A.; Clark, R. T.; Dunstone, N.; Eade, R.; Fereday, D.; Folland, C. K.; Gordon, M.; Hermanson, L.; Knight, J. R.; Lea, D. J.; MacLachlan, C.; Maidens, A.; Martin, M.; Peterson, A. K.; Smith, D.; Vellinga, M.; Wallace, E.; Waters, J.; Williams, A. (March 2014). "Skilful Long Range Prediction of European and North American Winters". *Geophysical Research Letters*. **41** (7): 2514–2519. Bibcode:2014GeoRL..41.2514S. doi:10.1002/2014GL059637. hdl:10871/34601. S2CID 127165980.
- Data: https://www.cpc.ncep.noaa.gov/products/precip/CWlink/pna/nao.shtml
-