ITMO UNIVERSITY

Saint Petersburg, Russia

Using the FEDOT framework functionality to fill in the gaps in time series

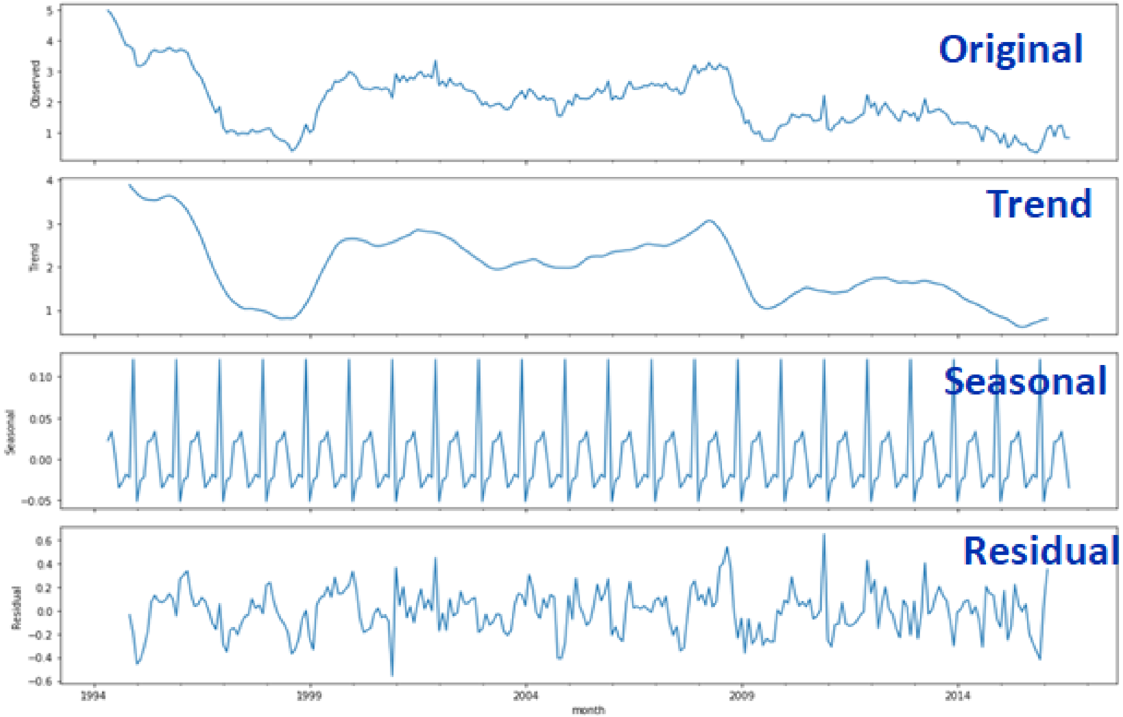
NSS-Lab ITMO

tutorial

Time series forecasting

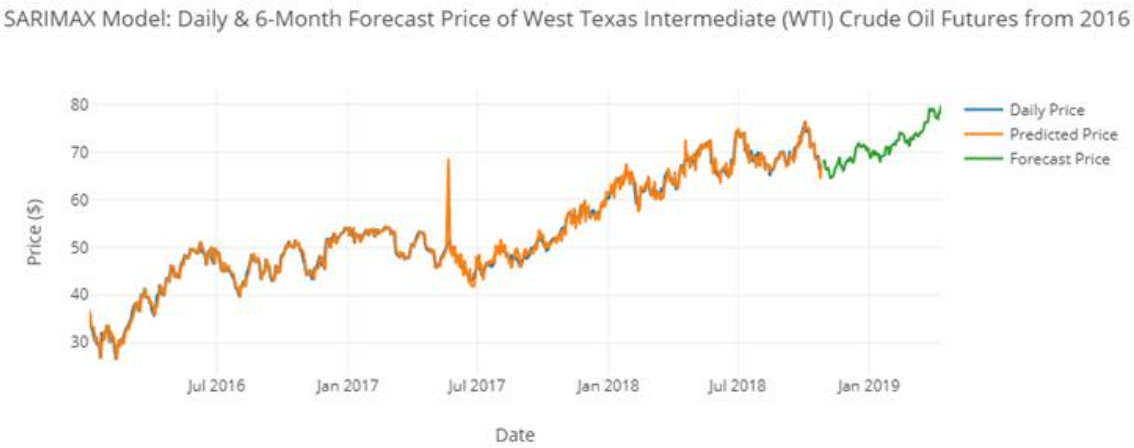
Components:

- **Trend** - long-term time series change;
- **Seasonality** – time series changes with constant period;
- **Cyclic** - time series changes with variable period;
- **Residuals —** a component that is left after other components have been calculated and removed from time series data.



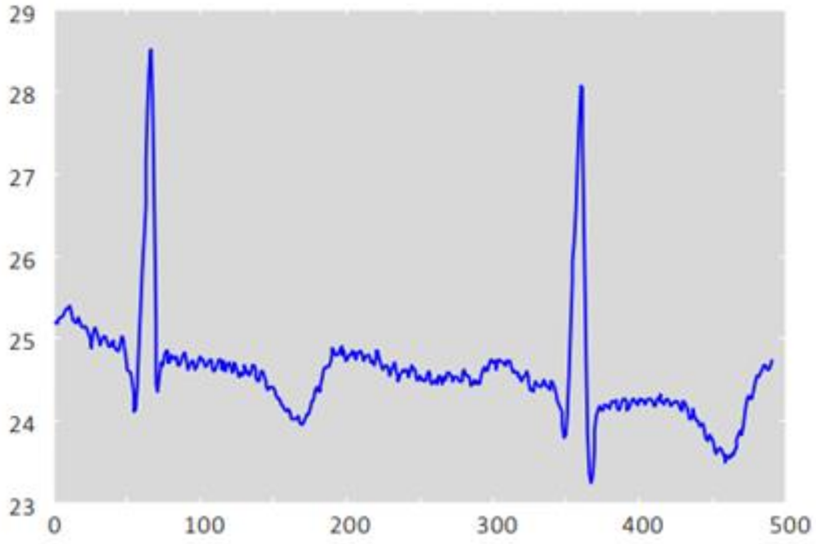
Additional factors

Example:

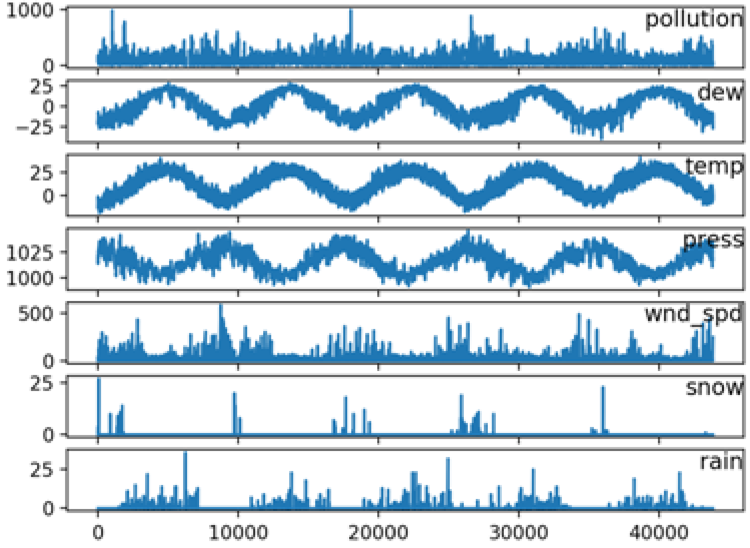


Univariate and multivariate time series

Univariate time series



Multivariate time series



Prediction quality metrics

ITMO UNIVERSITY

R^2 – explained variance

```
from sklearn.metrics import r2_score

print("Linear Regression R^2:",  round(r2_score(y, y_pred_lr), 3))
print("SMA R^2:",  round(r2_score(y , y_sma), 3))
```

```
Linear Regression R^2: 0.942
SMA R^2: 0.822
```

Mean squared error /
Root Mean Square Error

```
from sklearn.metrics import mean_squared_error

print("Linear Regression MSE:", round(mean_squared_error(y, y_pred_lr), 3))
print("SMA MSE:",  round(mean_squared_error(y , y_sma), 3))
```

```
Linear Regression MSE: 1882343.713
SMA MSE: 5774211.042
```

Mean absolute percentage error

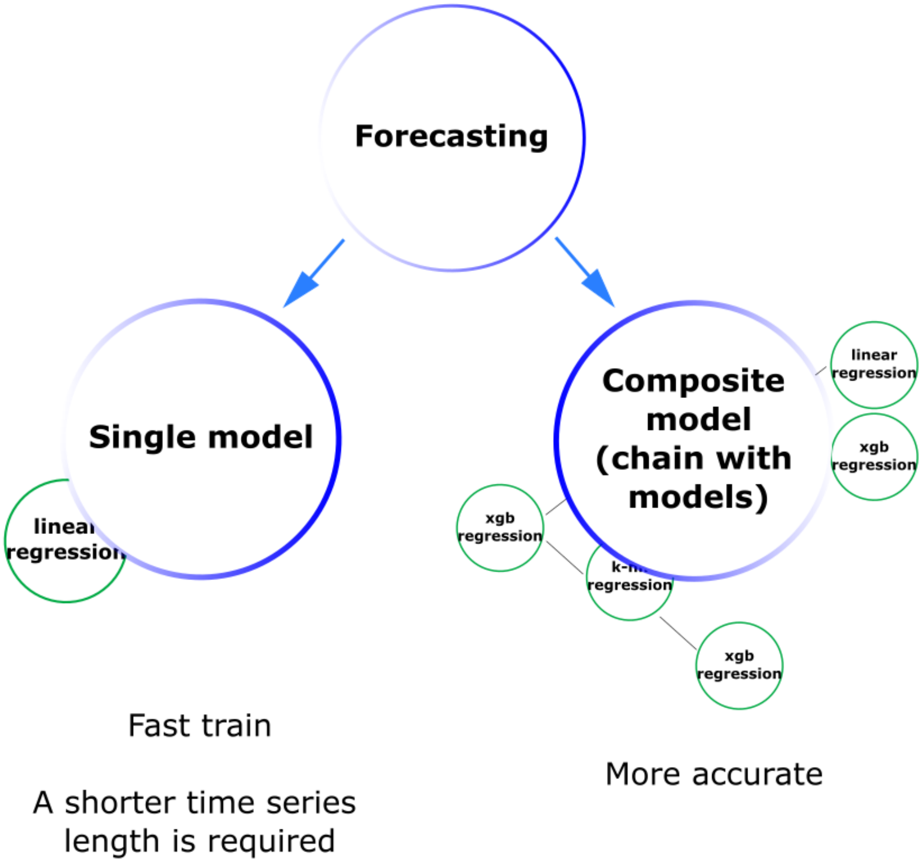
```
def mean_absolute_percentage_error(y_true, y_pred):
    return round(np.mean(np.abs((y_true - y_pred) / y_true)) * 100, 3)

print("Linear Regression MAPE:",  mean_absolute_percentage_error(y, y_pred_lr))
print("SMA MAPE:", mean_absolute_percentage_error(y , y_sma))

```

```
Linear Regression MAPE: 4.0
SMA MAPE: 22.493
```

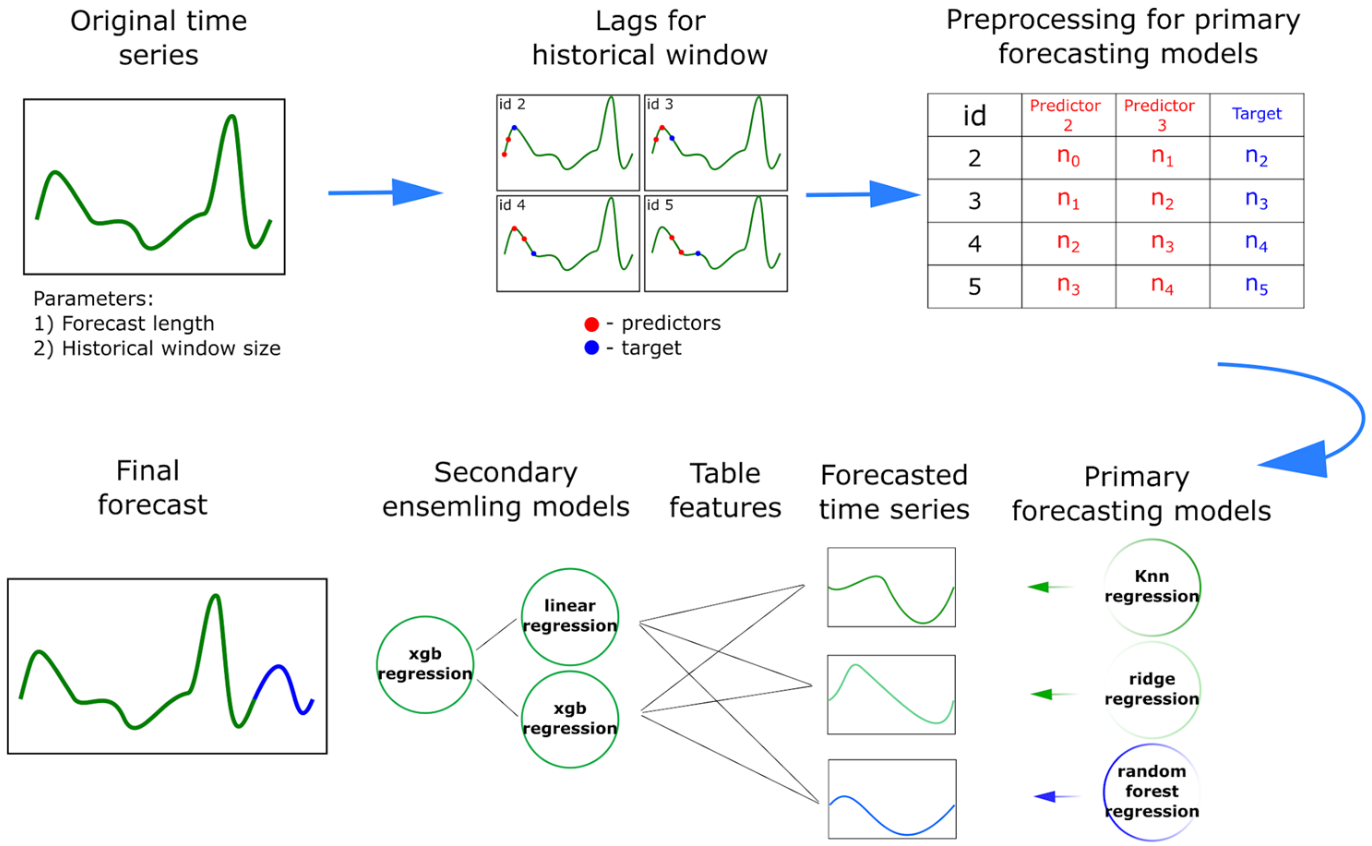
Two ways to build models using FEDOT

ITMO UNIVERSITY

- Based on the framework's functionality it is possible to build time series forecasting systems from a single model and then select the optimal hyperarameters for it;
- Or you can build chains of models that are harder to train, but will be more accurate.



Time series forecasting with FEDOT

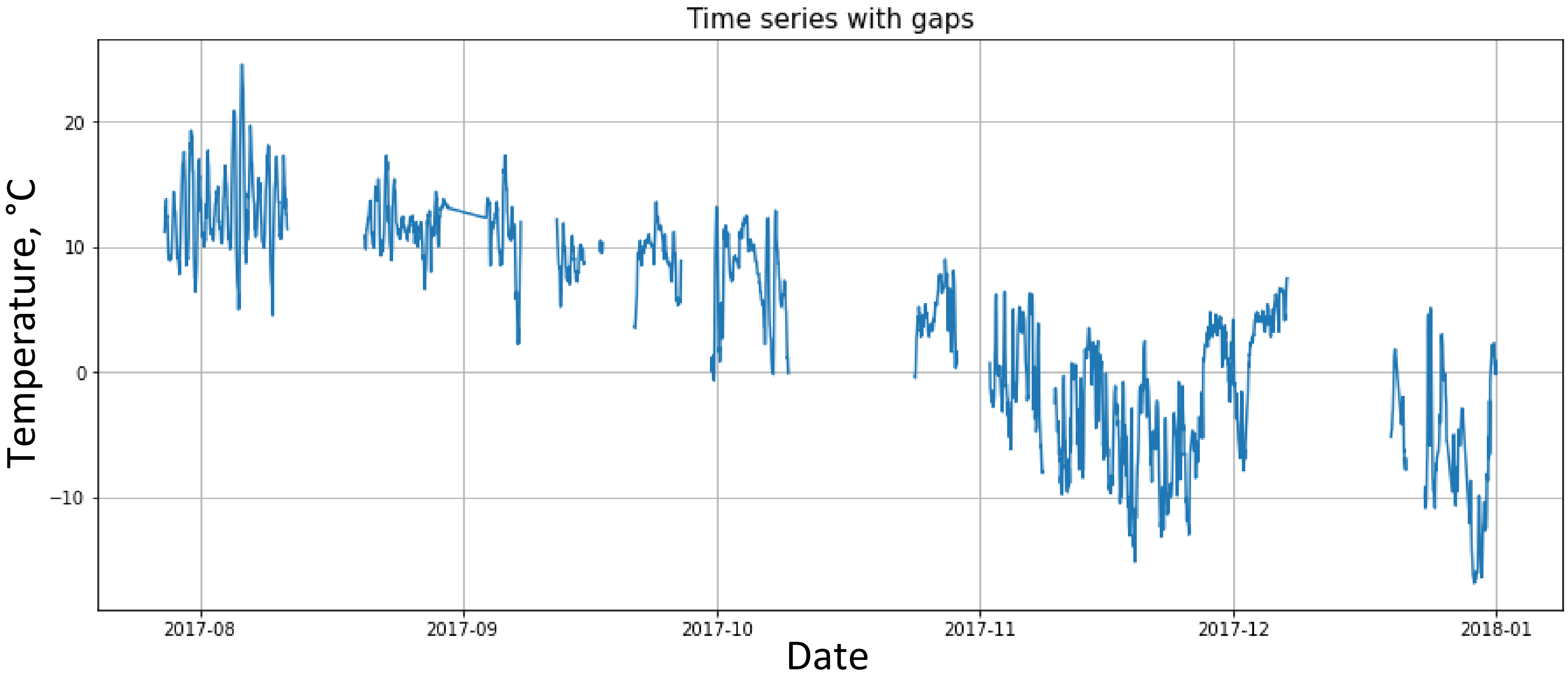


id	Predictor 2	Predictor 3	Target
2	n_0	n_1	n_2
3	n_1	n_2	n_3
4	n_2	n_3	n_4
5	n_3	n_4	n_5

Gaps in time series

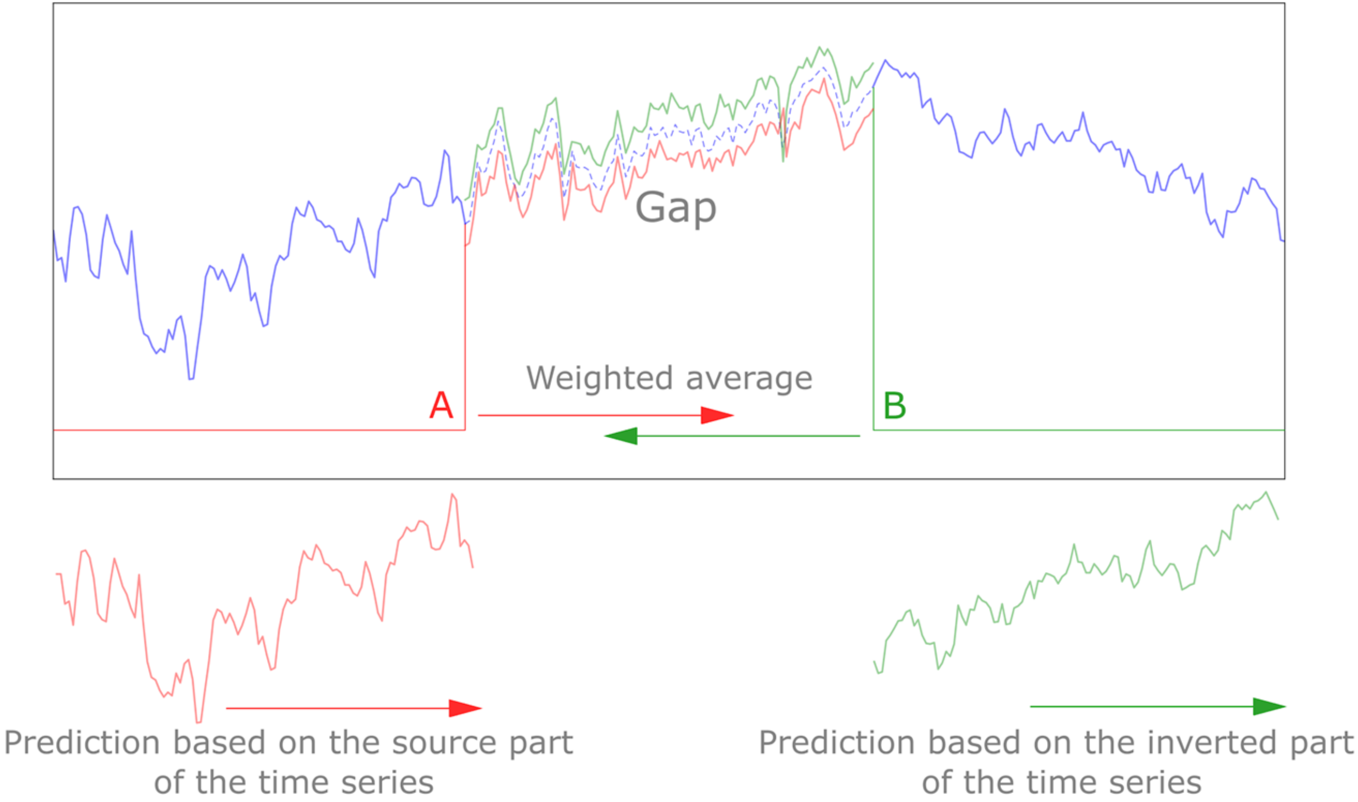
ITMO UNIVERSITY

- Gap filling task >= time series forecasting;
- For the gap filling task, to predict values, information can be used not only before the gap, but also the section of the time series after the omitted values.



Implemented approach

ITMO UNIVERSITY



FEDOT single model example

ITMO UNIVERSITY

- Necessary imports

```
import numpy as np
from core.composer.node import PrimaryNode, SecondaryNode
from core.composer.ts_chain import TsForecastingChain
from core.models.data import InputData
from core.repository.dataset_types import DataTypesEnum
from core.repository.tasks import Task, TaskTypesEnum, TsForecastingParams
```

- Declaring a chain of one model

```
chain = TsForecastingChain(PrimaryNode('ridge'))
```

- Declaring a time series forecasting task and preparing input data

```
task = Task(TaskTypesEnum.ts_forecasting,
            TsForecastingParams(forecast_length=forecast_length,
                                max_window_size=max_window_size,
                                return_all_steps=True,
                                make_future_prediction=True))

input_data = InputData(idx=np.arange(0, len(timeseries_train_part)),
                       features=None,
                       target=timeseries_train_part,
                       task=task,
                       data_type=DataTypesEnum.ts)
```

FEDOT single model example

- Train model

```
chain.fit_from_scratch(input_data)
```

- Preparing data for the forecast

```
test_data = InputData(idx=np.arange(0, len_gap),
                      features=None,
                      target=None,
                      task=task,
                      data_type=DataTypesEnum.ts)
```

- Make prediction

```
predicted_values = chain.forecast(initial_data=input_data, supplementary_data=test_data).predict
```

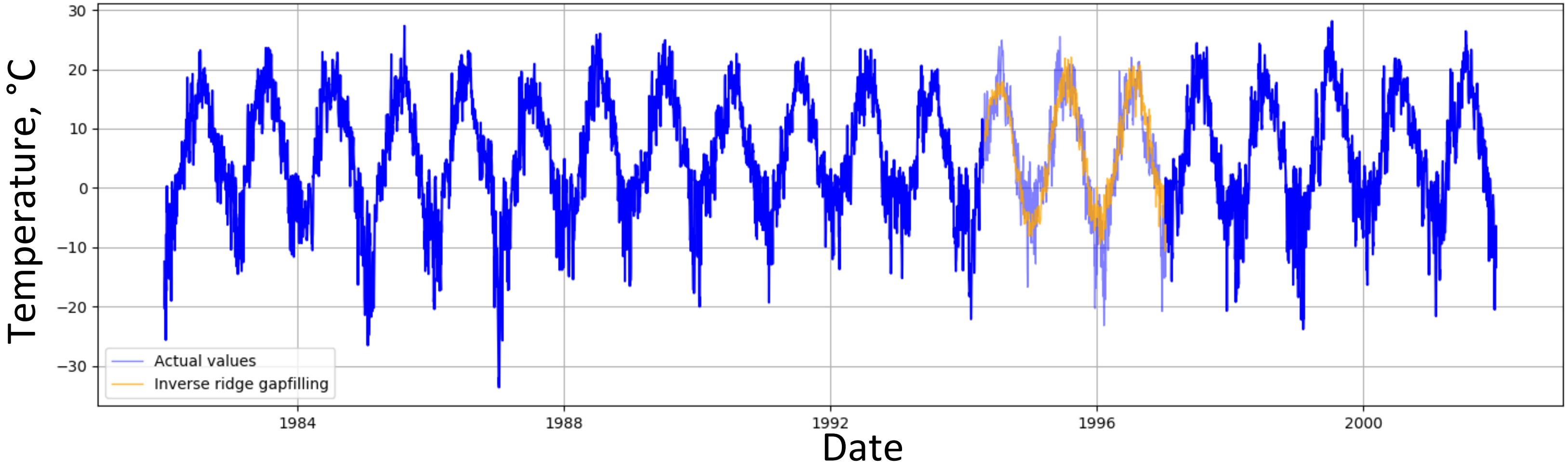
- Repeat the procedure for the inverted “right” part of the time series, weigh the forecasts and combine.

FEDOT single model example

- **Or you can use the implemented functionality of the framework - class ModelGapFiller;**
- To do this, declare an instance of the ModelGapFiller class and give it an input to the inverse_ridge function, where the ridge regression is used as the main model, your array with skips:

```
gapfiller = ModelGapFiller(gap_value=-100.0)
without_gap_arr_ridge = gapfiller.inverse_ridge(gap_array, max_window_size=250)
```

- Get output



FEDOT composite model example

ITMO UNIVERSITY

- Instead of declaring a chain with a single model, you can put multiple models in the chain using code as example:

```
node_first = PrimaryNode('trend_data_model')
node_second = PrimaryNode('residual_data_model')
node_trend_model = SecondaryNode('linear', nodes_from=[node_first])
node_residual_model = SecondaryNode('linear', nodes_from=[node_second])

node_final = SecondaryNode('additive_data_model', nodes_from=[node_trend_model, node_residual_model])
chain = TsForecastingChain(node_final)
```

- In this case, a chain of 5 models is declared, where there are two input nodes (`'trend data model'`,`'residual_data_model'`), two intermediate nodes (`'linear', 'linear'`), and one final node (`'additive_data_model'`);

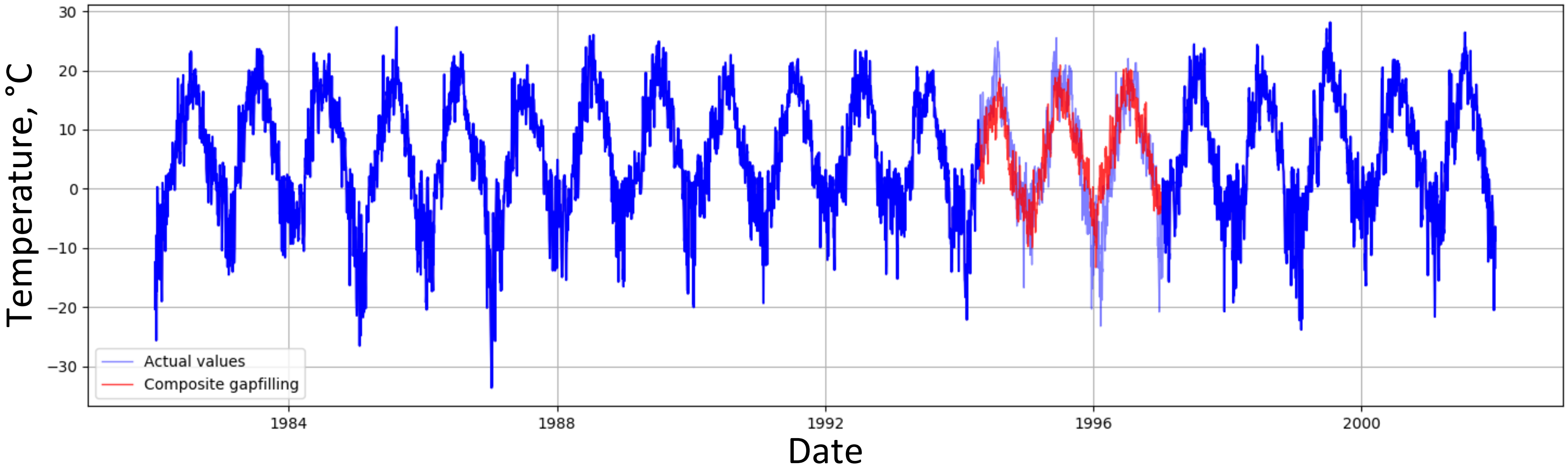
- Repeat all the same steps that we did from the chain with the same model.

FEDOT composite model example

- Or you can use a function from the class ModelGapFiller - composite_fill_gaps

```
gapfiller = ModelGapFiller(gap_value=-100.0)
without_gap_arr_composite = gapfiller.composite_fill_gaps(gap_array, max_window_size=1000)
```

- Get output



Conclusion

ITMO UNIVERSITY

- As part of the development of the FEDOT automatic machine learning framework, the time series prediction functionality was implemented;

- Based on time series forecasting methods, algorithms for efficient gap recovery in time series have been developed;

- Two functions for restoring omissions in one-dimensional arrays based on a single model (inverse_ridge) and a chain with multiple models (composite_fill_gaps) were implemented;

- Based on high-level commands, it is now possible to restore gaps in one-dimensional arrays with only 2 lines of code using FEDOT.

Thank you for attention!

www.ifmo.ru

ITsMOre than a UNIVERSITY