ScaDS.AI DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Training School “AI 4 Seismology”

TRAINING: Transformers for Time Series ML

SPEAKER: Matthias Täschner

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

SACHSEN Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

AGENDA

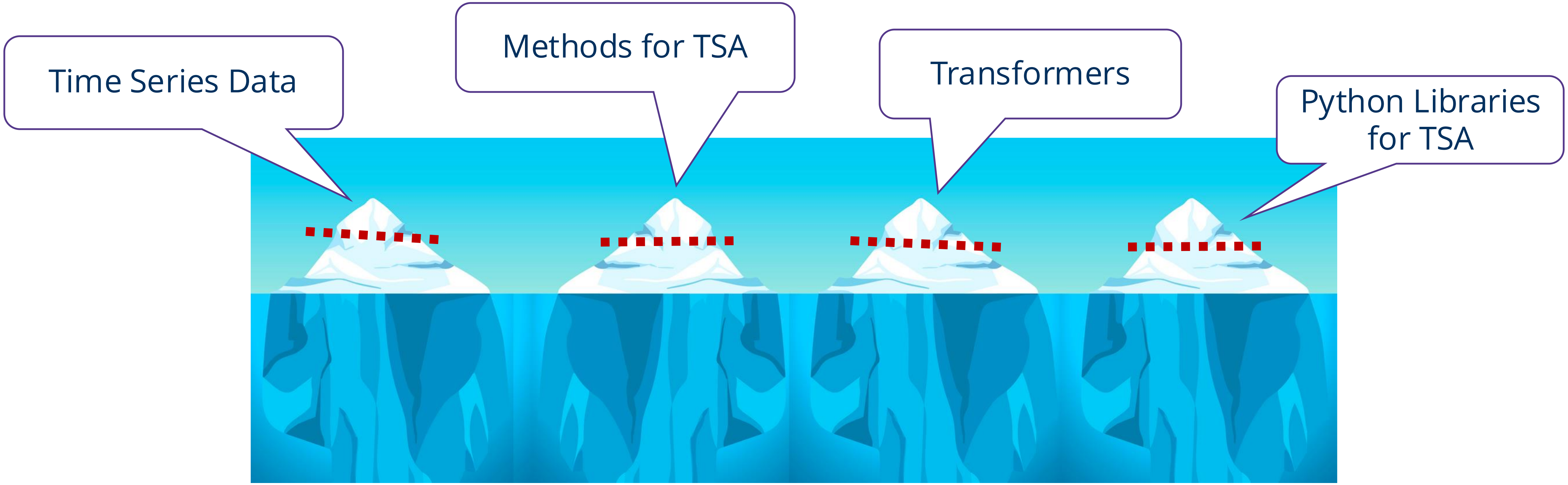
- Introduction to Time Series Analysis (TSA)
- Traditional Methods in TSA
  - Auto Regression and Moving Averages
  - Exponential Smoothing
  - State Space Models
- Machine Learning and Deep Learning for TSA
  - Tree-based models
  - Neural Networks (RNN, CNN)
  - AutoML
- Introduction to Transformers
  - Concepts, Architecture and Components
- Transformers for TSA
  - Challenges and Approaches
- Python Libraries

Expectation Management

Overview and Examples

Within 2 x 45 min:

Teaching the fundamentals as a basis for further learning and application

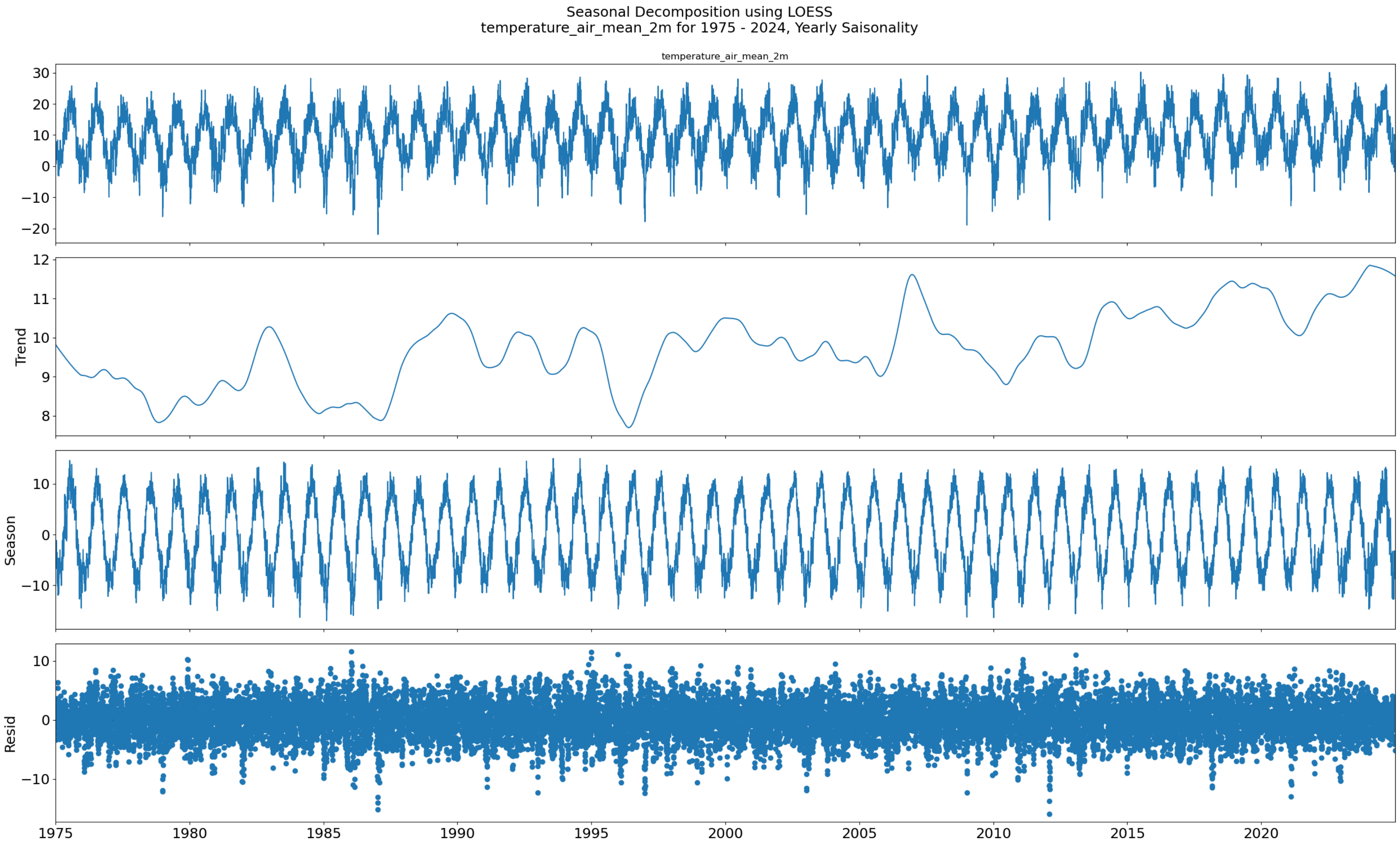


Introduction to Time Series Analysis

Time Series Data Components and Properties

- **Trend**: long-term direction (upward or downward) of data over time, mean is changing
- **Seasonality**: regular and predictable cycles in the data occurring at fixed intervals
- **Cyclical**: fluctuations at more irregular intervals, influenced by external factors, usually do not have fixed periods like seasonality
- **Residuals**: what's left when all known components like trend or seasonality are removed, differences between observed values and predicted values from a model
- **Noise**: inherent randomness in the data, also not explained by underlying patterns, typically assumed to be random with zero mean and constant variance and no autocorrelation ("white noise"), if a model is good the residuals should be only white noise
- **Stationarity**: statistical properties (mean, variance, covariance) are constant over time
- **Differencing**: subtract a data point by points at previous positions, e.g. for de-trending

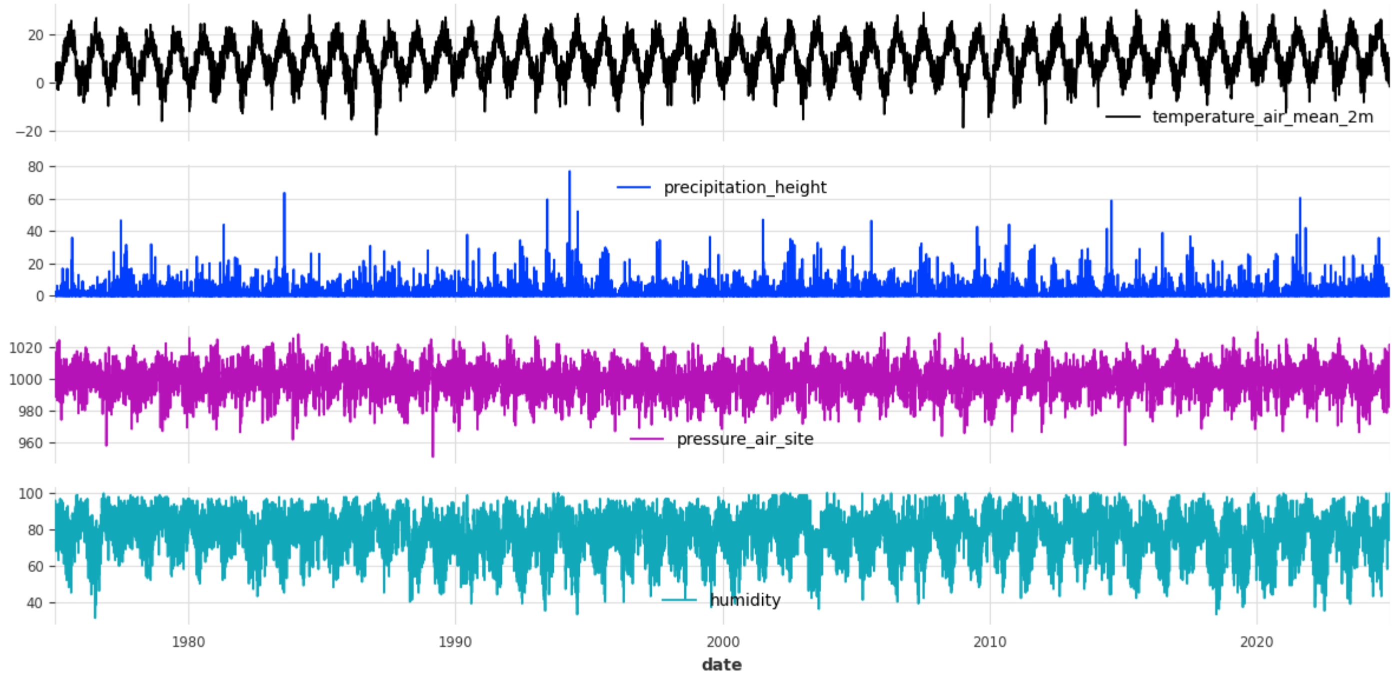
Introduction to Time Series Analysis



Introduction to Time Series Analysis

Time Series Data Components and Properties

- **Univariate**: one variable observed over time (e.g., temperature over years)
- **Multivariate**: multiple variables observed simultaneously (e.g., temperature, humidity, …)
- **Covariates**: additional variables that help to explain the target variable *(past, future, static)*



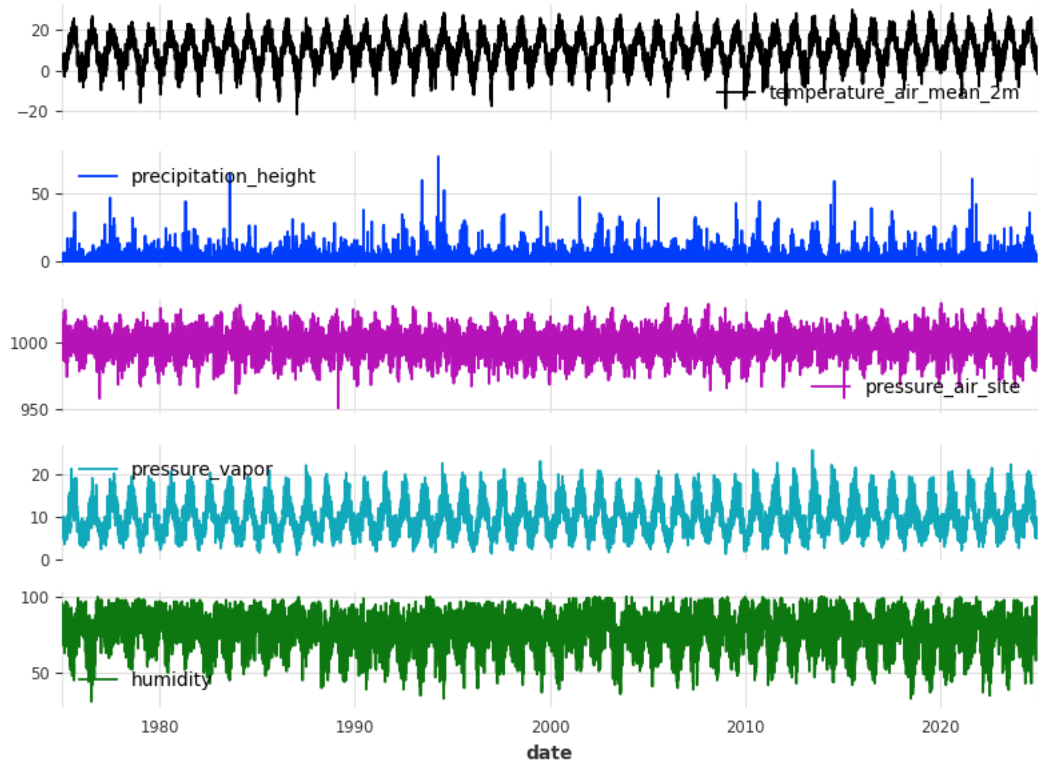
Introduction to Time Series Analysis

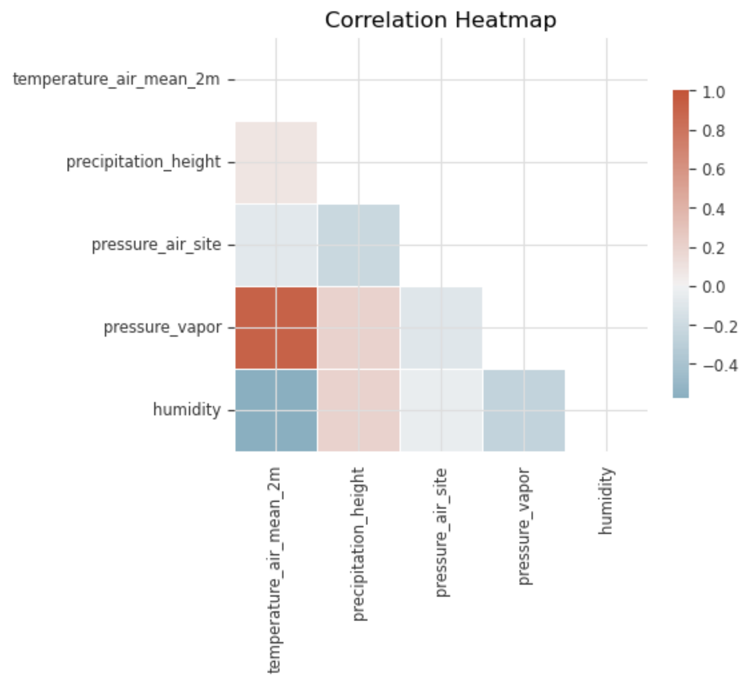
Time Series Data Components and Properties

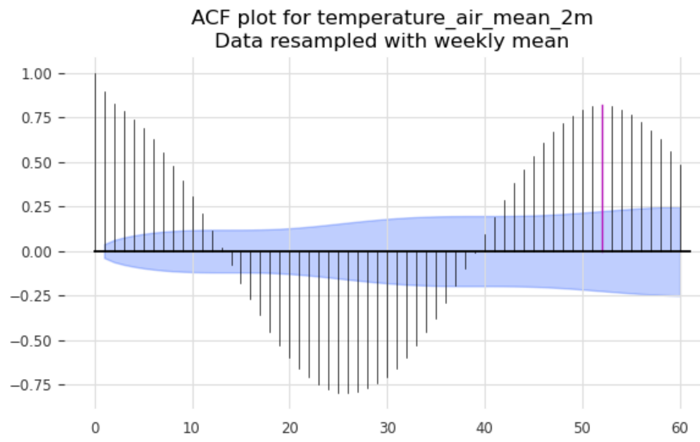
- **Covariance**: measures how two variables move together, unstandardized joint variability
- **Correlation**: Standardized relationship between two variables within [-1,1]
- **Autocorrelation (AC)**: correlation of a time series with a lagged version of itself, how do past values influence future values, helps identifying general temporal dependencies, e.g. seasonality
- **Partial Autocorrelation (PAC)**: correlation of a time series with itself at a specific lag, after removing the influence of the values at shorter lags

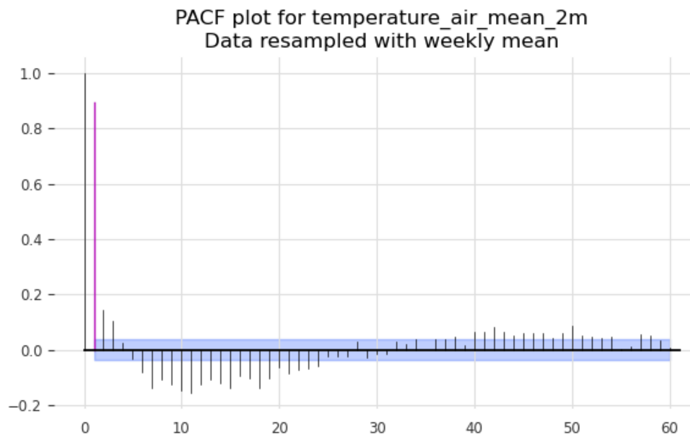
Introduction to Time Series Analysis

Time Series Data Components and Properties









Introduction to Time Series Analysis

Time Series Data Components and Properties

Specifics in Earth Science and Remote Sensing

- Long-term pattern
- Lot of noise
- Multivariate data with covariance

Introduction to Time Series Analysis

Time Series Analysis

Exploratory Analysis – Understand your data

- Identify patterns, temporal dependencies, and relationships

Classification – Assign labels to series

- Categorize time series data, e.g. what is avalanche, human activity, …

Clustering – Grouping similar series

- Group time series data based on similar patterns or behavior

Forecasting – Predict future values

- Estimate future data points based on the time series and its patterns

Anomaly Detection – Spot unusual behavior

- Identify outliers or unexpected observations

Goal:
Modelling the time series as good as possible

Traditional Methods in Time Series Analysis

Auto Regression and Moving Averages

Auto Regression (AR)

- Value at time t depends on its own previous values, today depends on yesterday / the day before

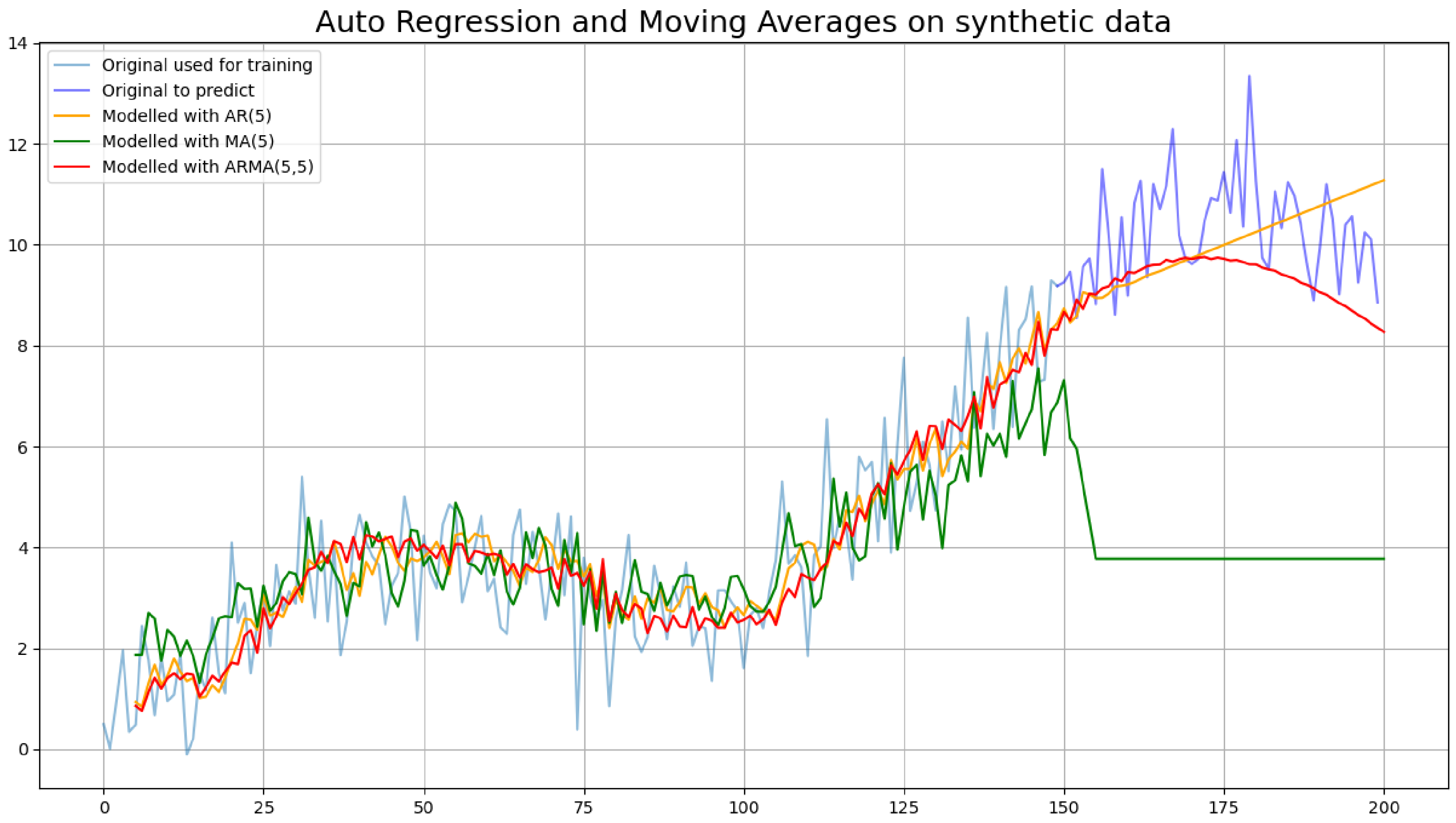
Moving Average (MA)

- Value at time t depends on past forecast errors, today depends on how wrong we were recently

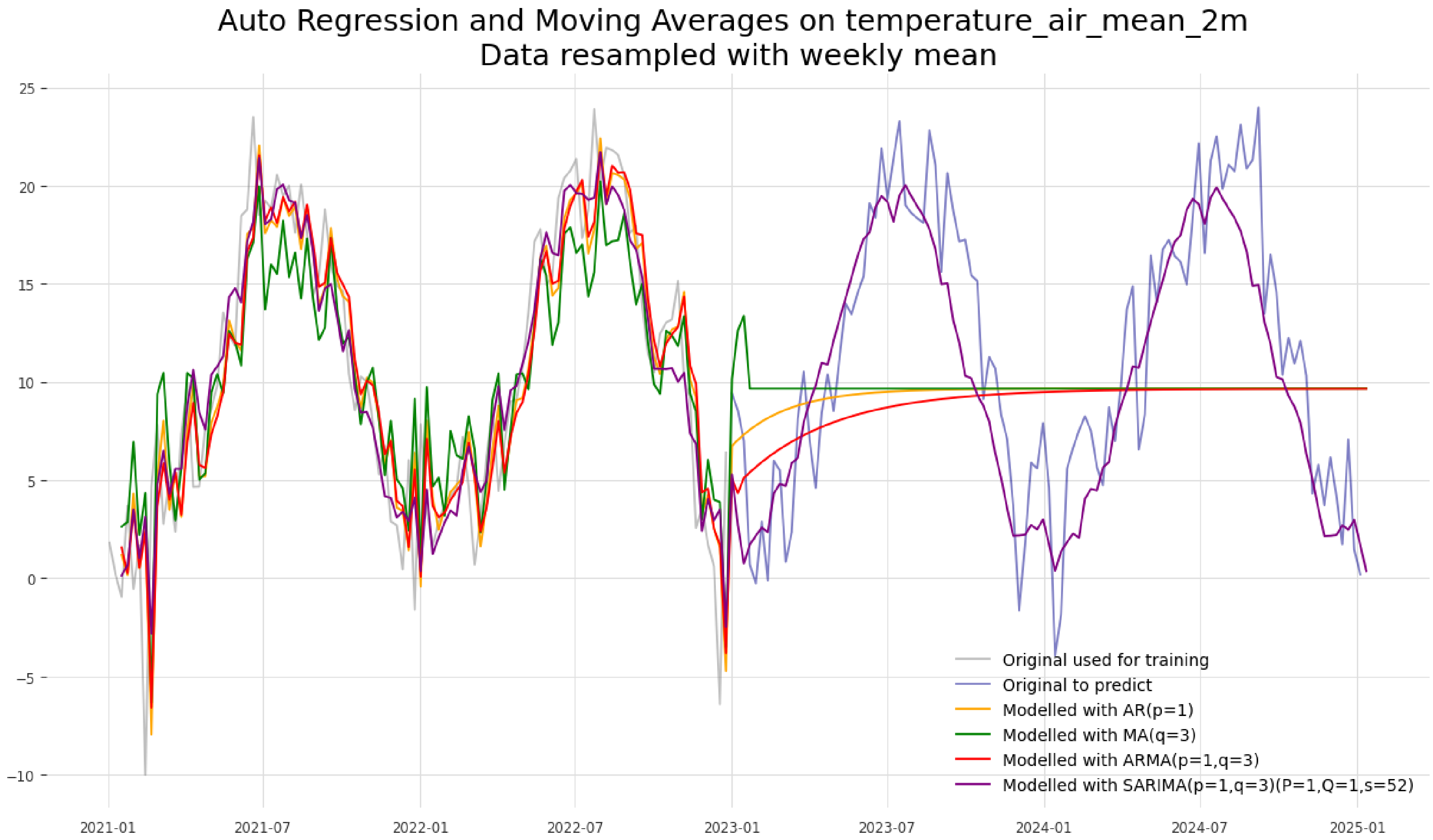
Combined as ARMA models and variants

- AR component with order p, can be identified via PACF
- MA component with order q, can be identified via ACF
- Additional components for Differencing (I), Seasonality modelling (S), exogeneous variables (X), …
- SARIMAX (Seasonal Autoregressive Integrated Moving Average with exogenous variables)
- VARMA (Vector Autoregressive Moving Average – for multivariate time series)

Traditional Methods in Time Series Analysis



Traditional Methods in Time Series Analysis



Traditional Methods in Time Series Analysis

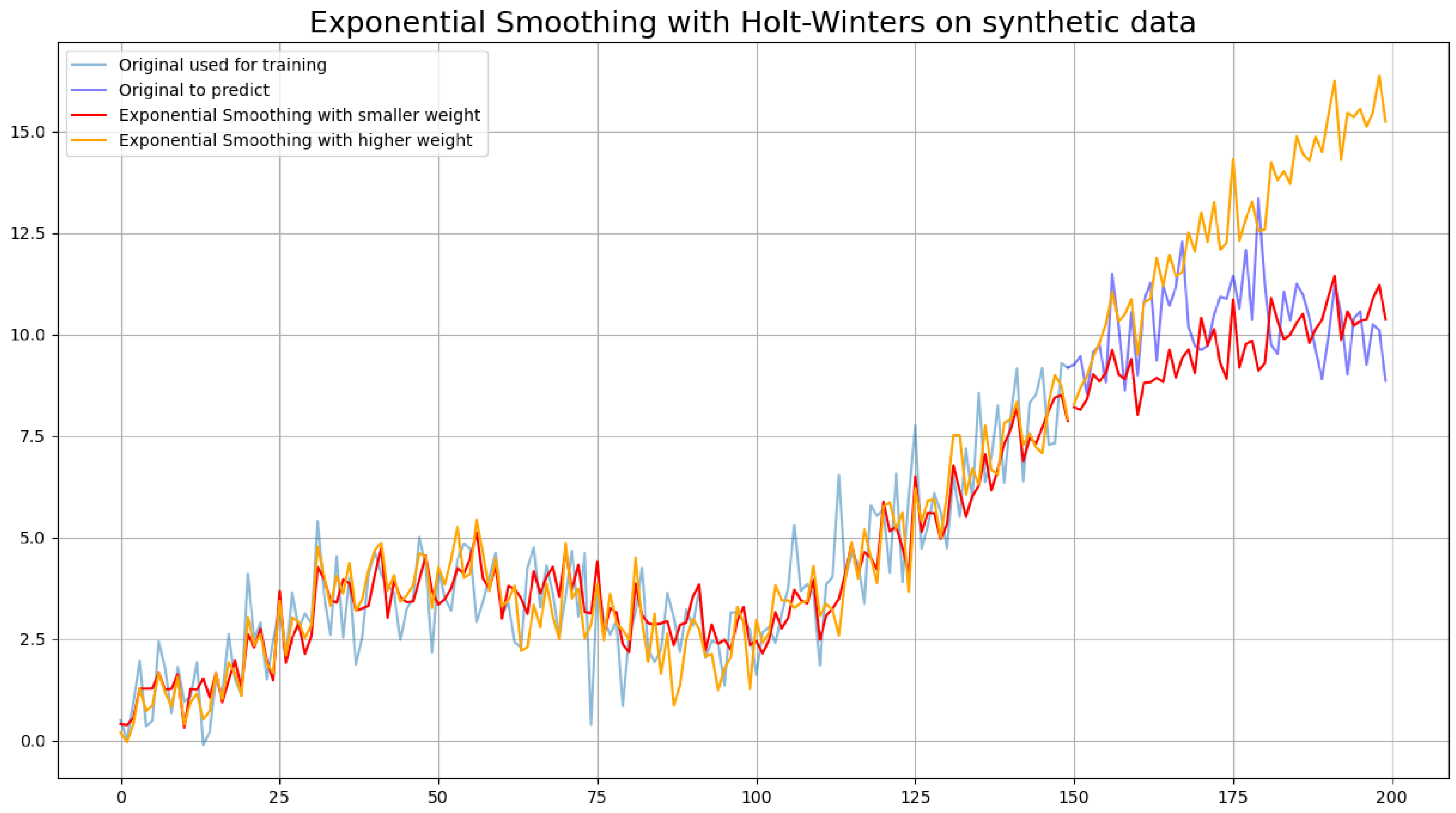
Exponential Smoothing

- Value at time t is the weighted average of past observations, with more weight on recent values
- Recent values matter more, older ones fade away
- Good for short-term modelling and forecasts, easy to interpret

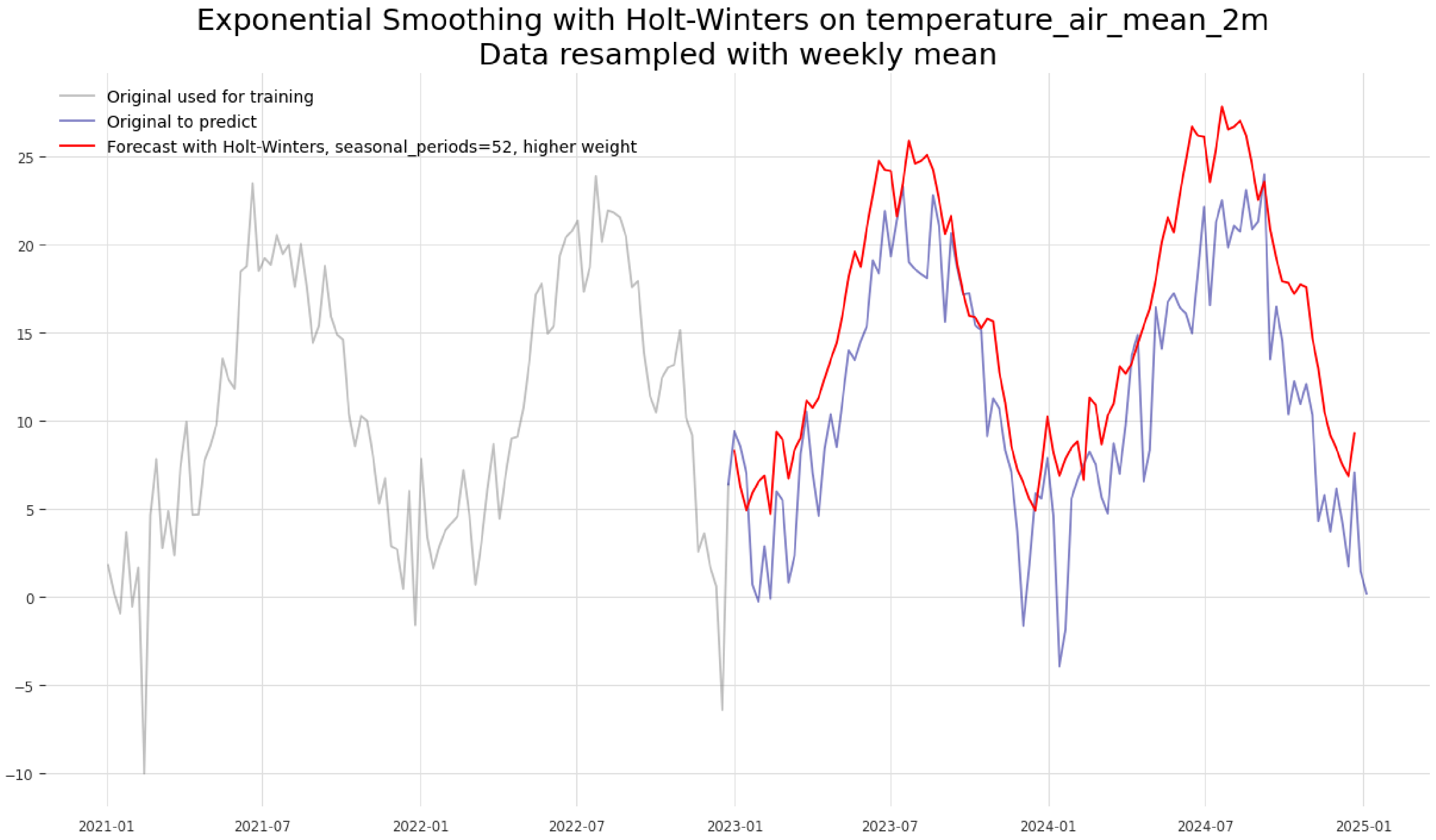
Variants

- Holt’s Method: Adds trend smoothing
- Holt-Winters: Adds seasonality smoothing

Traditional Methods in Time Series Analysis



Traditional Methods in Time Series Analysis

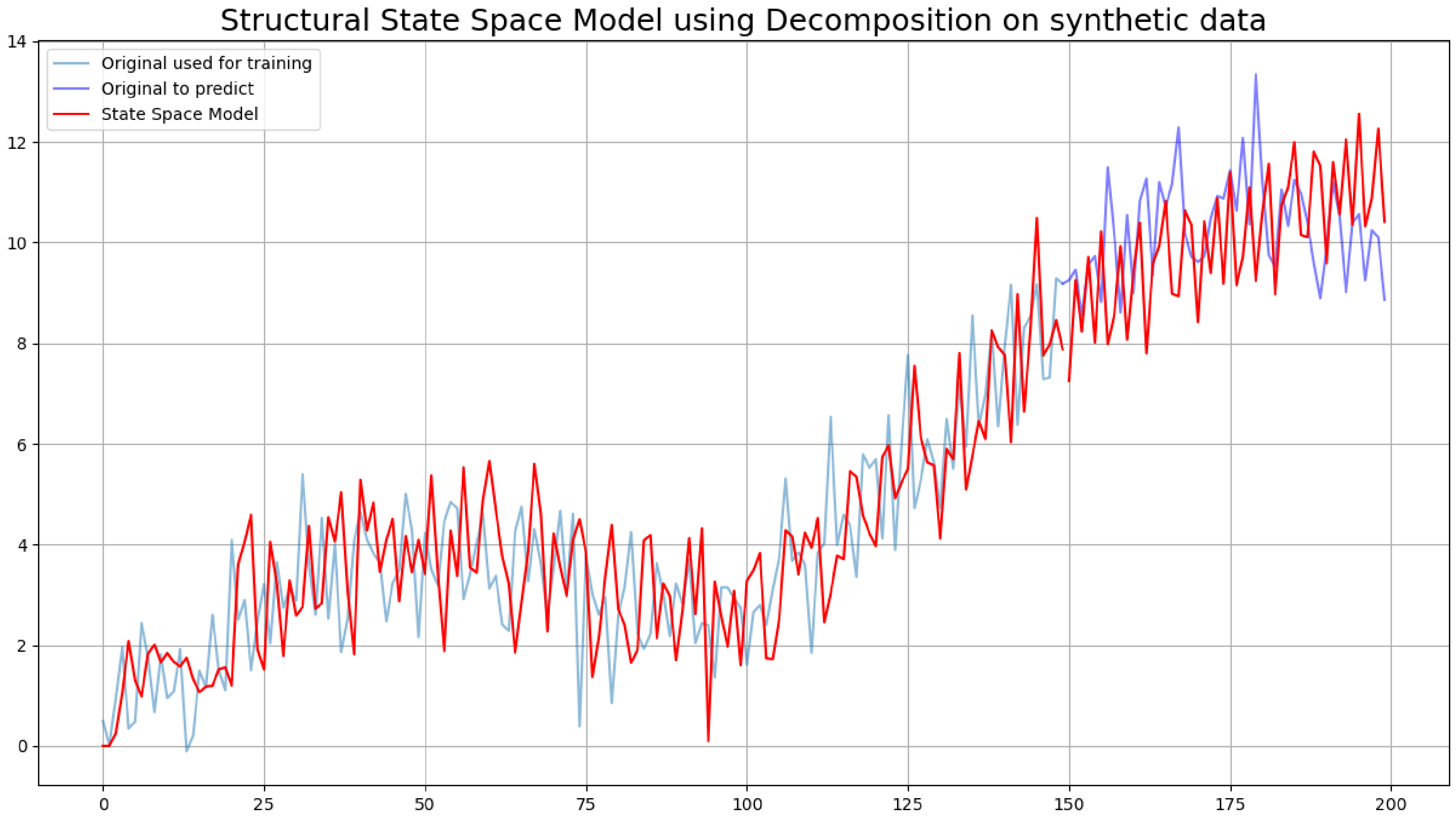


Traditional Methods in Time Series Analysis

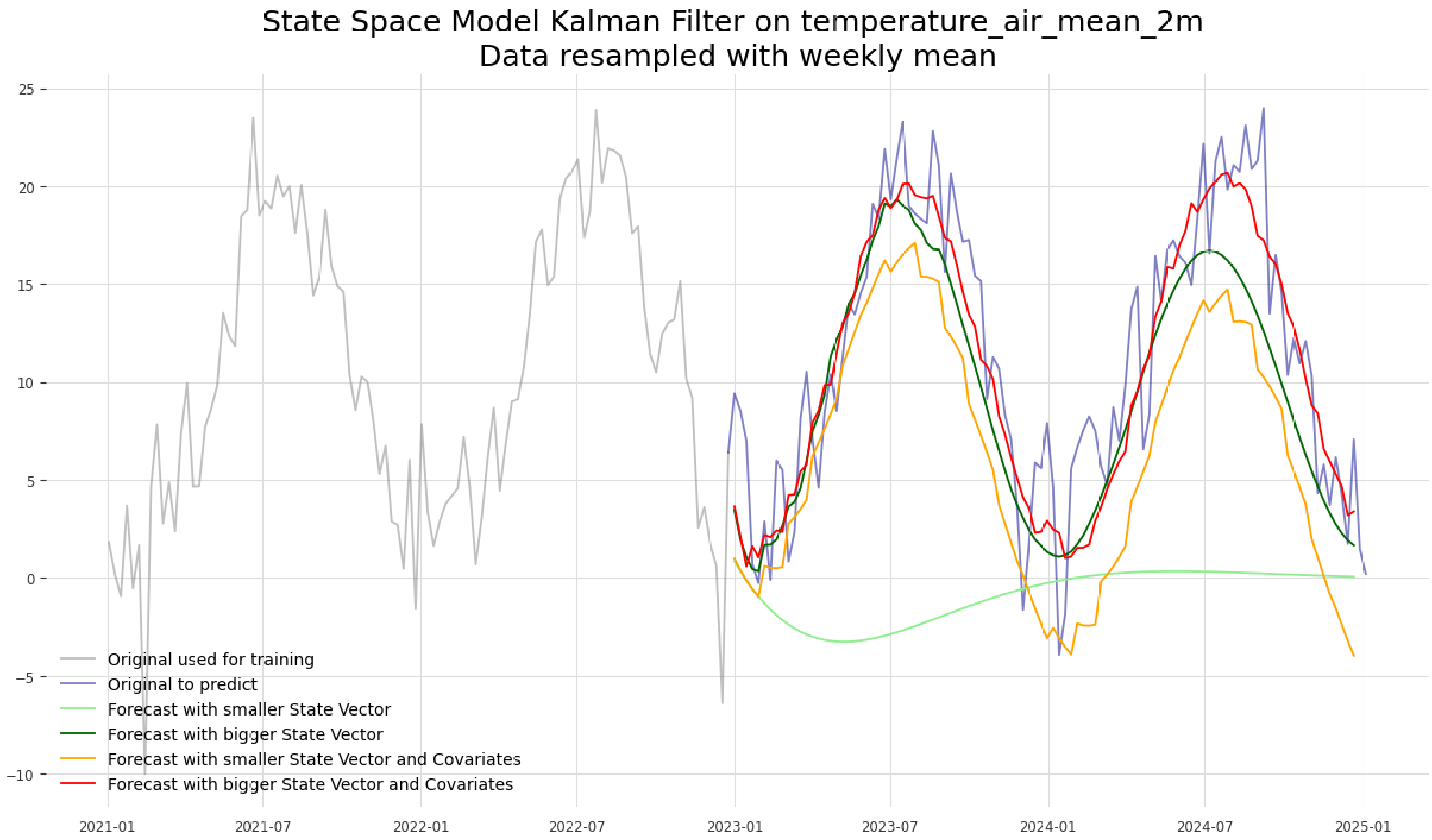
State Space Models

- Mathematical framework where the system is assumed to evolve over hidden internal states that generate the observed data
- There's a hidden process evolving over time, we only observe a noisy or incomplete version of it
- State equation: how does the hidden state evolves over time, with adjustments
- Observation equation: describes how observed data is related to hidden state
- State Vector Size: how many internal variables ("concepts") the model tracks, e.g. trend, season, noise, …
- Handles noisy and incomplete data well

Traditional Methods in Time Series Analysis



Traditional Methods in Time Series Analysis

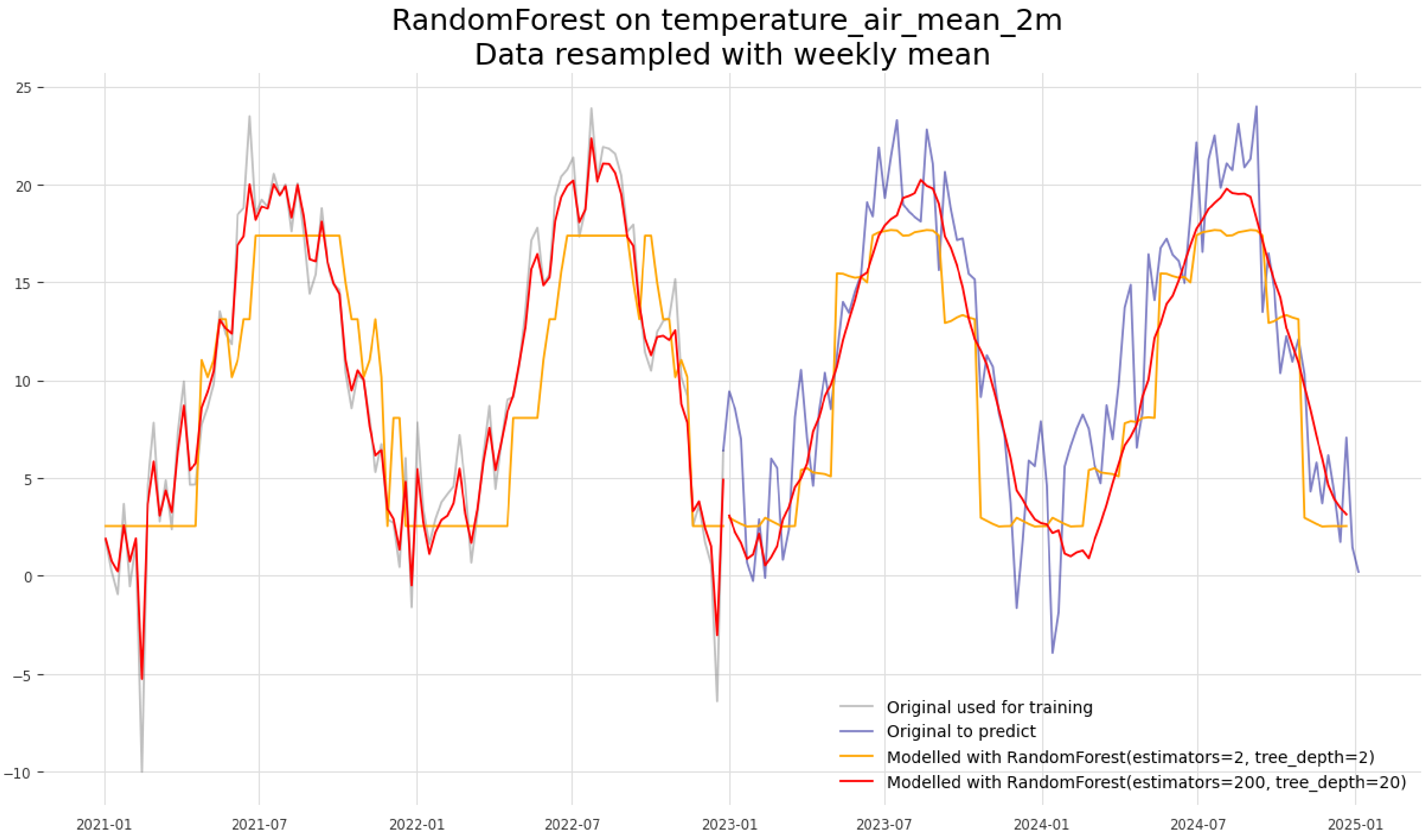


Machine Learning and Deep Learning in Time Series Analysis

Tree-based models

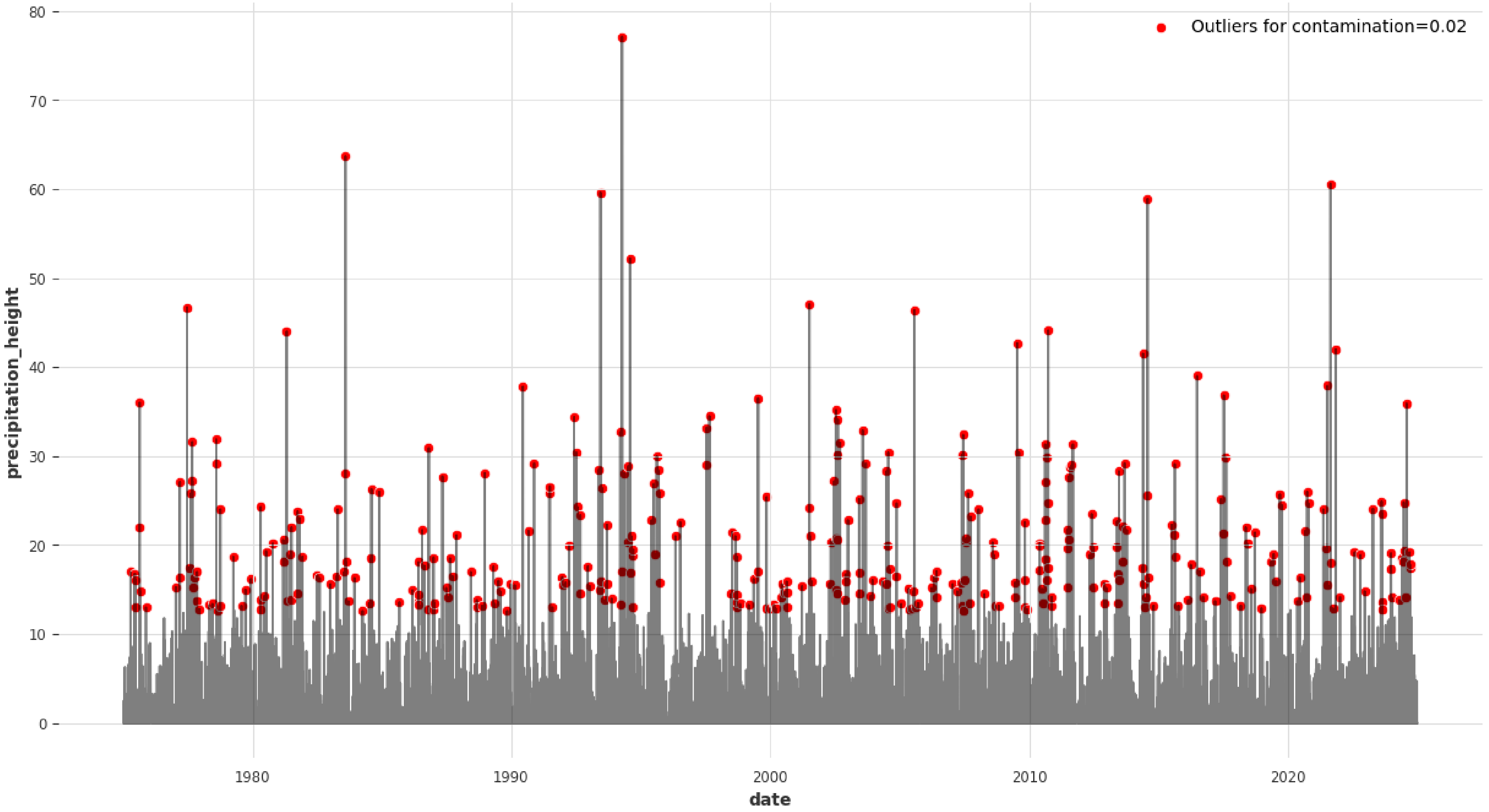
- Split the data into smaller pieces (branches) based on rules, use the splits to make predictions
- Works better with additional features used for the split rules (lags, moving averages, external variables)
- Tree parameters have a huge impact on modelling and prediction quality (e.g. number and depth of trees)
- Decision Trees, Random Forests, XGBoost
- Special forms of tree-based models for anomaly detection: Isolation Forest

Machine Learning and Deep Learning in Time Series Analysis



Machine Learning and Deep Learning in Time Series Analysis

IsolationForest for Anomaly Detection on precipitation_height

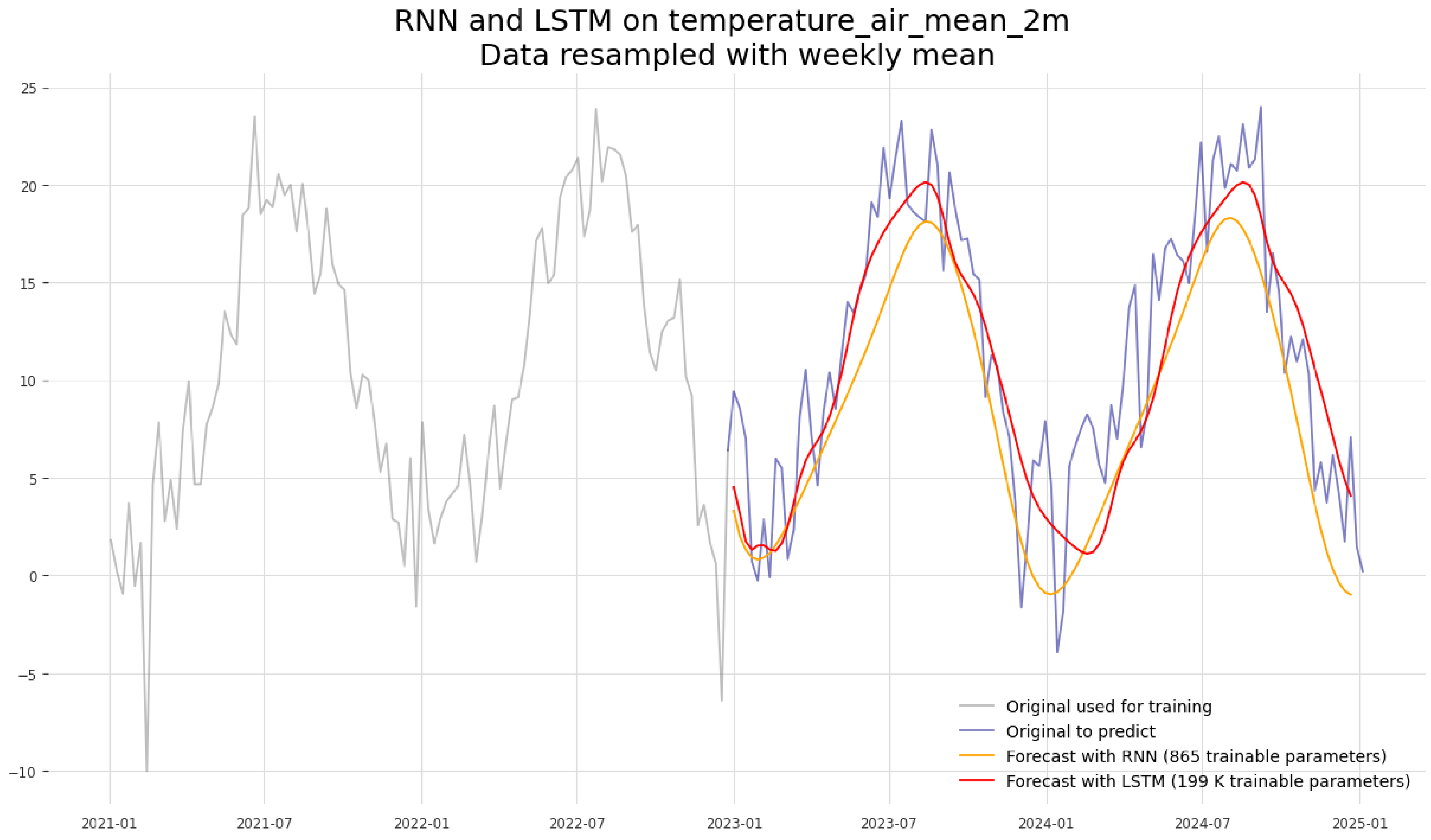


Machine Learning and Deep Learning in Time Series Analysis

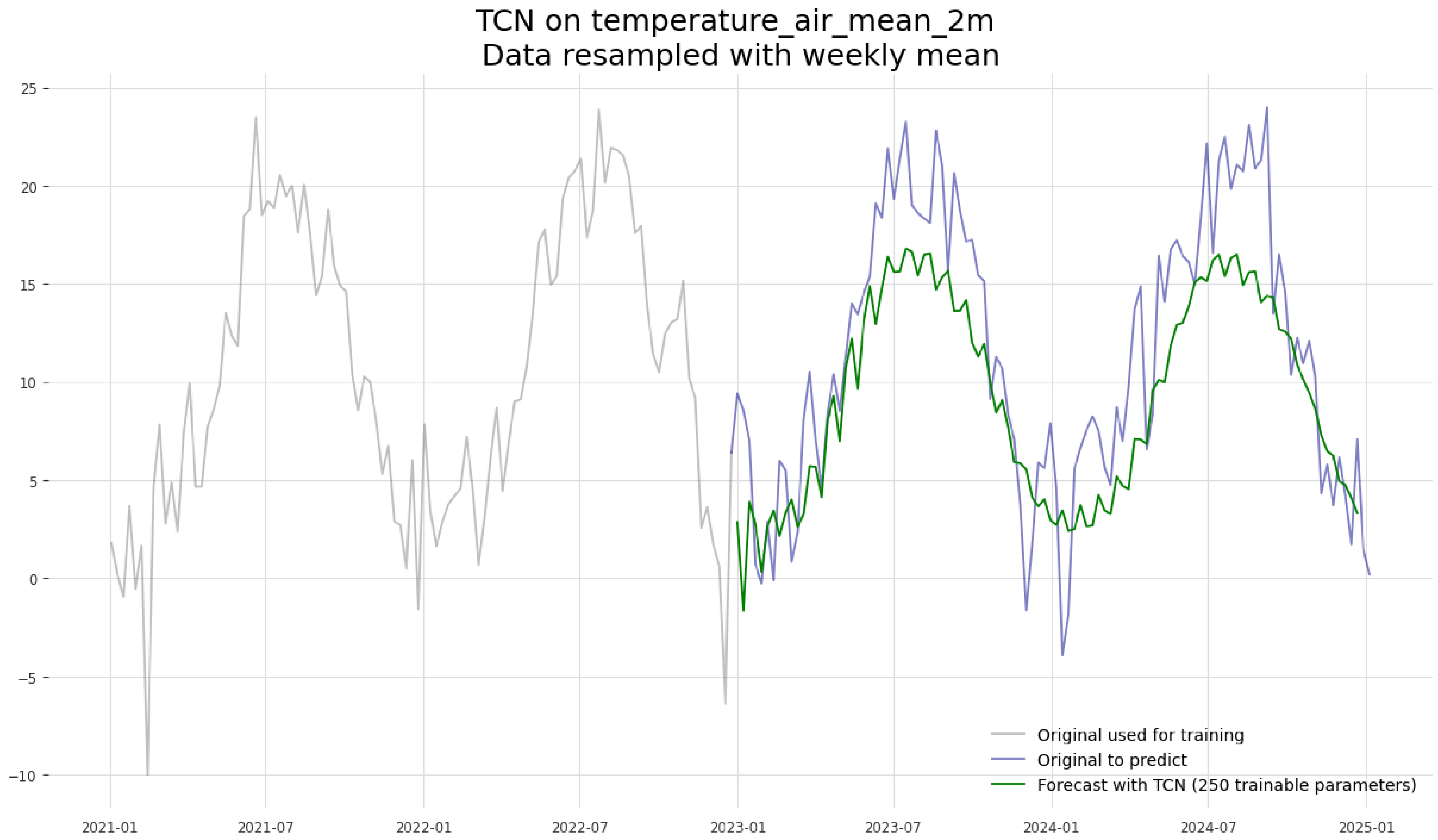
Neural Networks

- Make use of deep neural networks to learn pattern automatically, let the model figure out what matters: short-term spikes, long-term trends, etc.
- Especially useful when patterns are complex and nonlinear
- Recurrent Neural Networks (RNN) for sequences in general
- Long Short Term Memory (LSTM) to also remember long-term patterns
- (Temporal) Convolutional Neural Networks (CNN / TCN) to learn local patterns

Machine Learning and Deep Learning in Time Series Analysis



Machine Learning and Deep Learning in Time Series Analysis

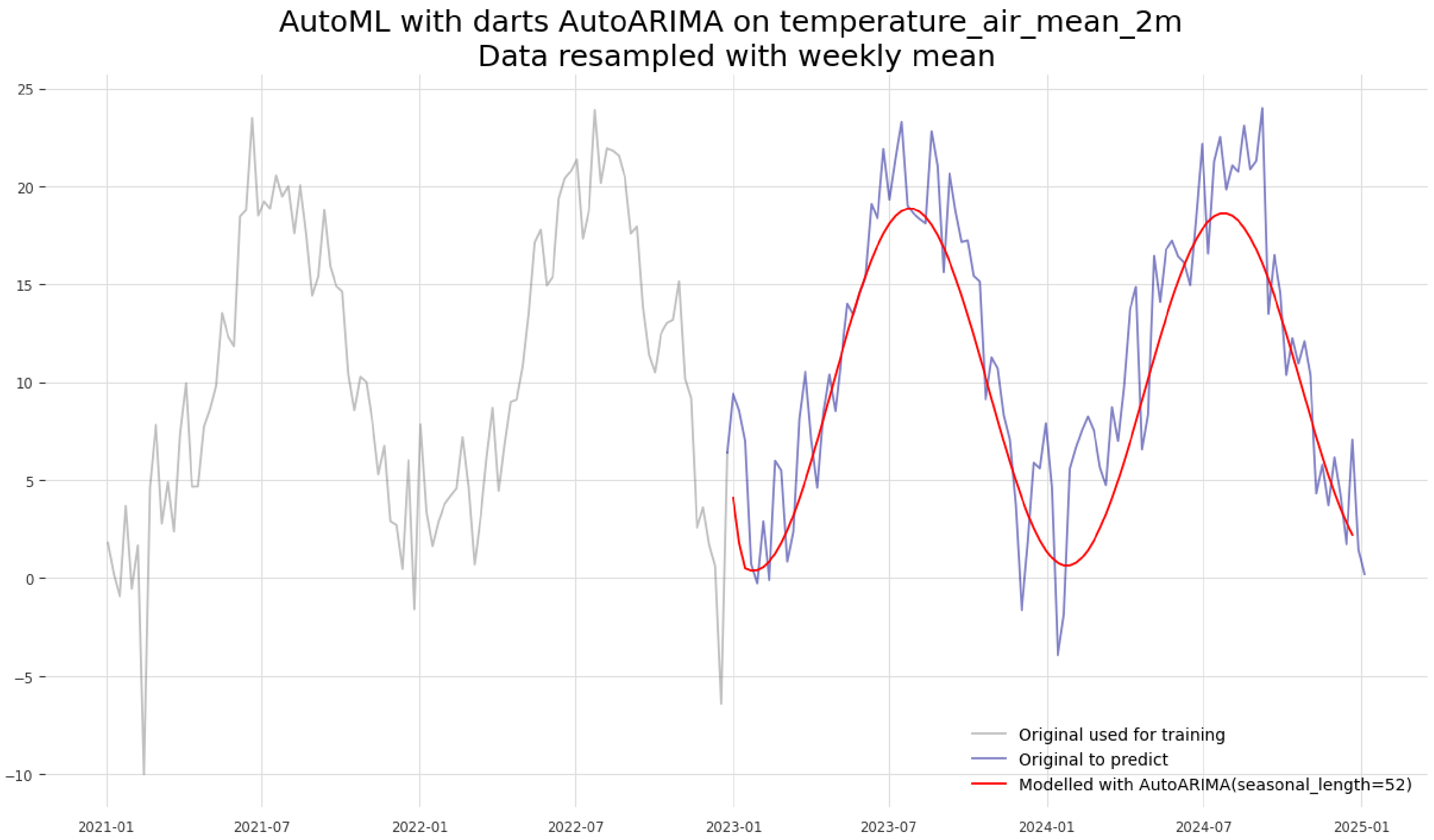


Machine Learning and Deep Learning in Time Series Analysis

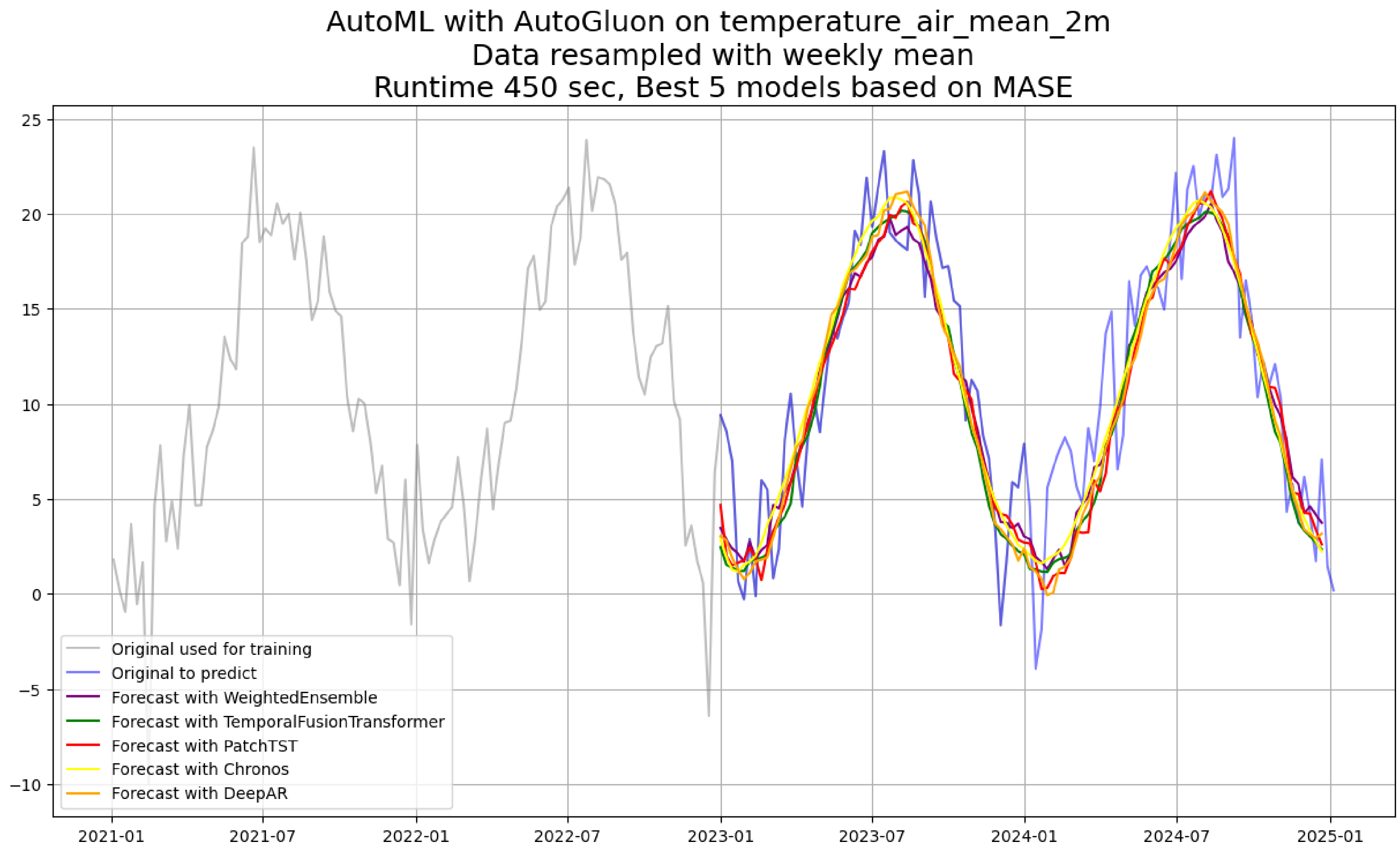
AutoML

- *"Why pick models yourself when a robot can test and pick them for you?"*
- AutoML automates the tedious parts of machine learning: feature engineering, model selection, hyperparameter tuning, ensembling, …
- AutoML tools for TSA: AutoTS, statsforecast, AutoGluon, …

Machine Learning and Deep Learning in Time Series Analysis



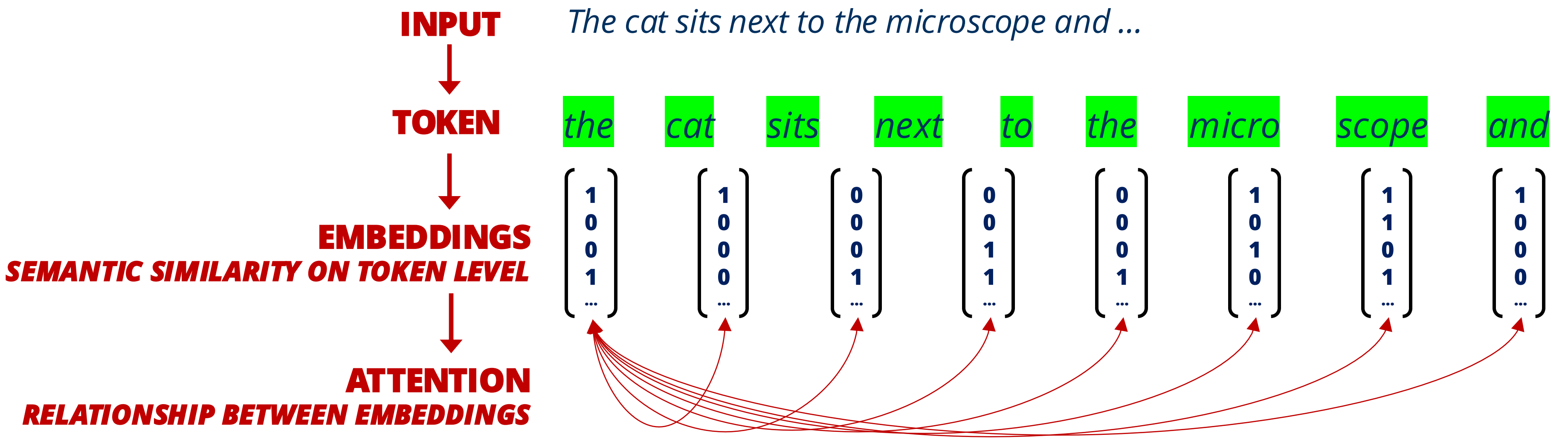
Machine Learning and Deep Learning in Time Series Analysis



Introduction / Recap to Transformers

Concepts

- Tokenization: breaking input data into smaller, meaningful units (tokens)
- Embeddings: mapping discrete tokens to continuous vectors that capture semantic similarity
- Self-Attention: pairwise relationship and relevance of token / embeddings
- Vocabulary: full set of tokens the model knows about, tokens are mapped to a vocabulary index



Introduction / Recap to Transformers

Architecture and Components

Add & Norm
Stabilize training process

Feed Forward
Transform information nonlinearly

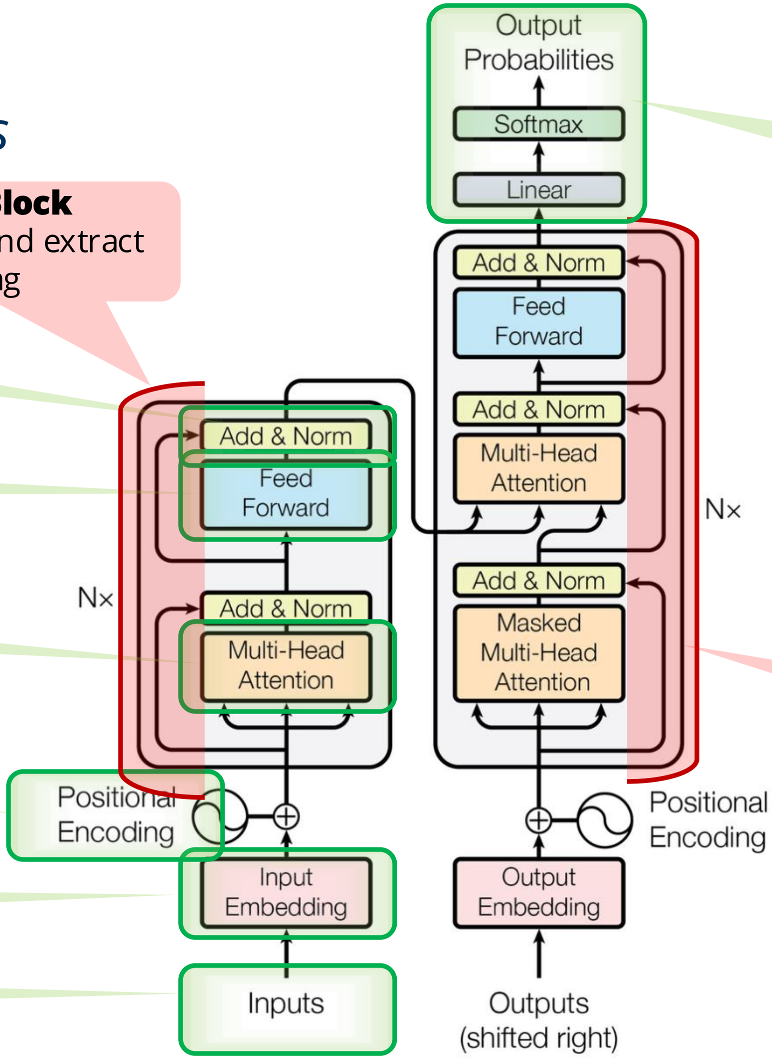
Multi-Head Attention
Find relationship between embeddings / tokens

Positional Encoding
Enrich embeddings with positional information

Embeddings
Create embeddings from tokens

Tokenizer
Build Tokens from input data

Encoder Block
Process input and extract meaning



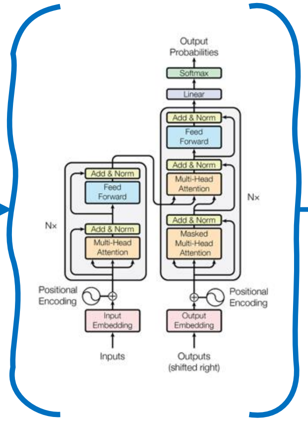
Output
Compute softmax probabilities for next tokens

Decoder Block
Take previous prediction and encoder output to predict next tokens

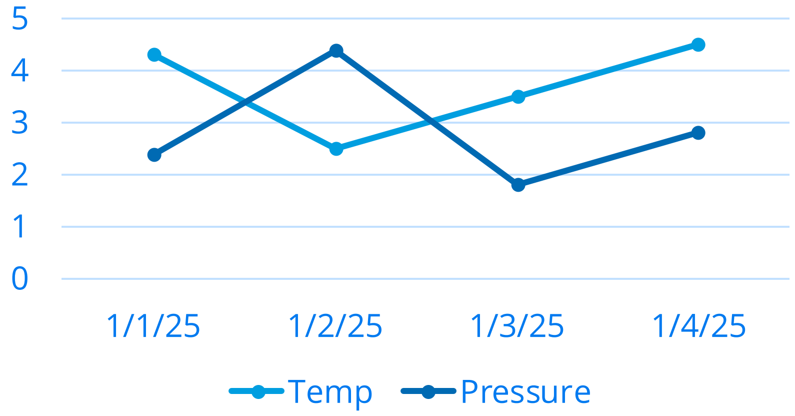
Source: Vaswani et al (2017), https://arxiv.org/abs/1706.03762

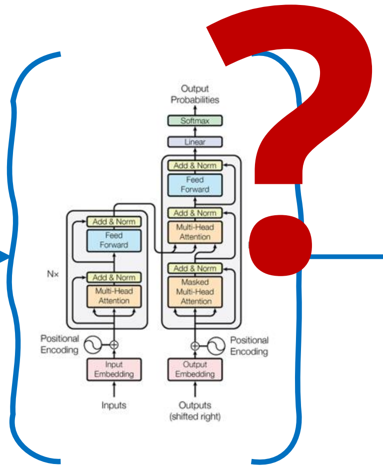
Transformers for Time Series Analysis

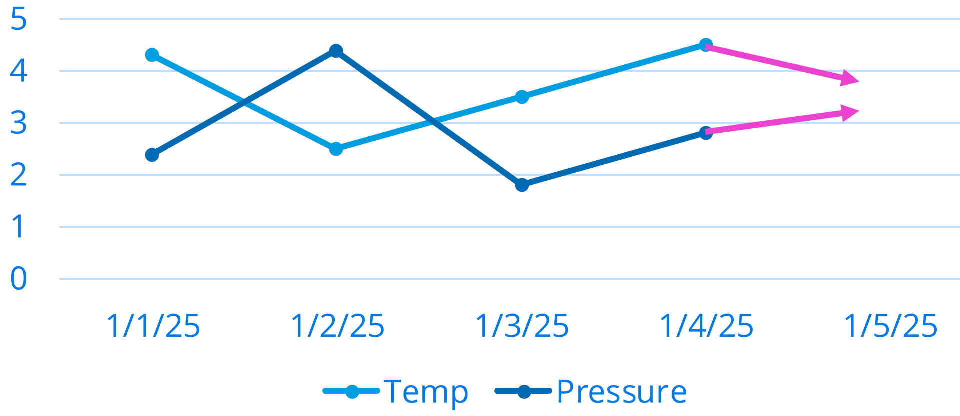
The cat sits next to the microscope and …



The cat sits next to the microscope and watches …







Source: Vaswani et al (2017), https://arxiv.org/abs/1706.03762

Transformers for Time Series Analysis

Challenges

No natural tokenization for time series data

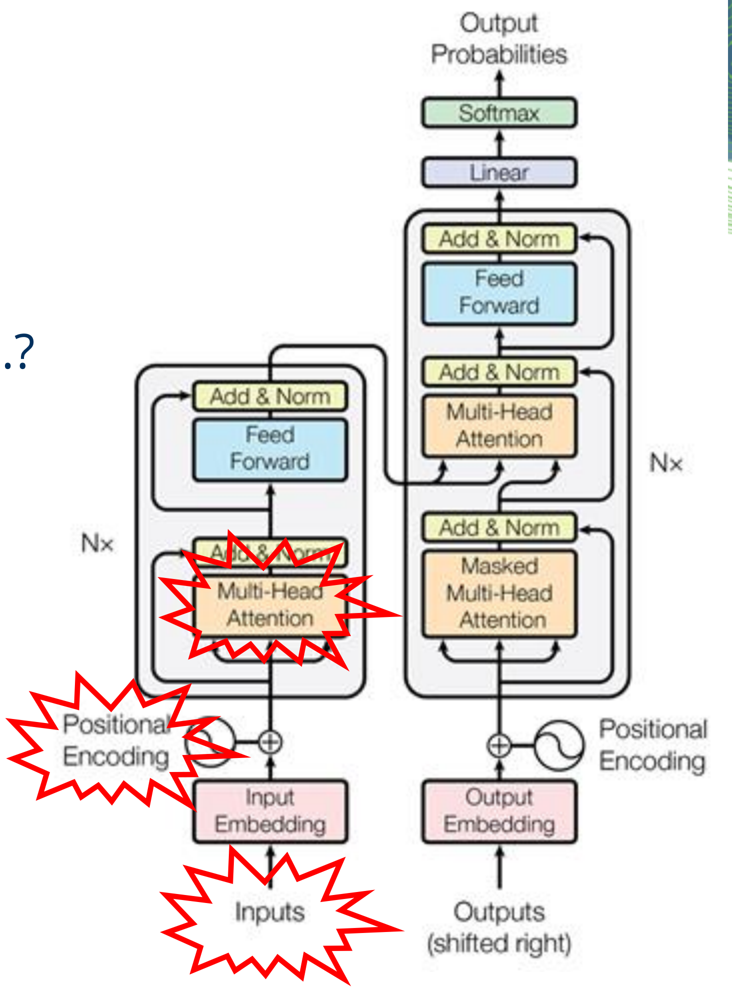
- Time series are continuous, not discrete words
- How to define a token: timestamp, window, feature vector of a series, ...?

Position and order matter even more

- Text: word order matters, but missing or switching a few words is fine
- Time series: missing or misordering timestamps can break forecasting
- Precise temporal relationship must be preserved

Sequence length and scalability

- Transformers self-attention mechanism has O(n^2)
- Long historical time series may cause memory explosion



Source: Vaswani et al (2017), https://arxiv.org/abs/1706.03762

Transformers for Time Series Analysis

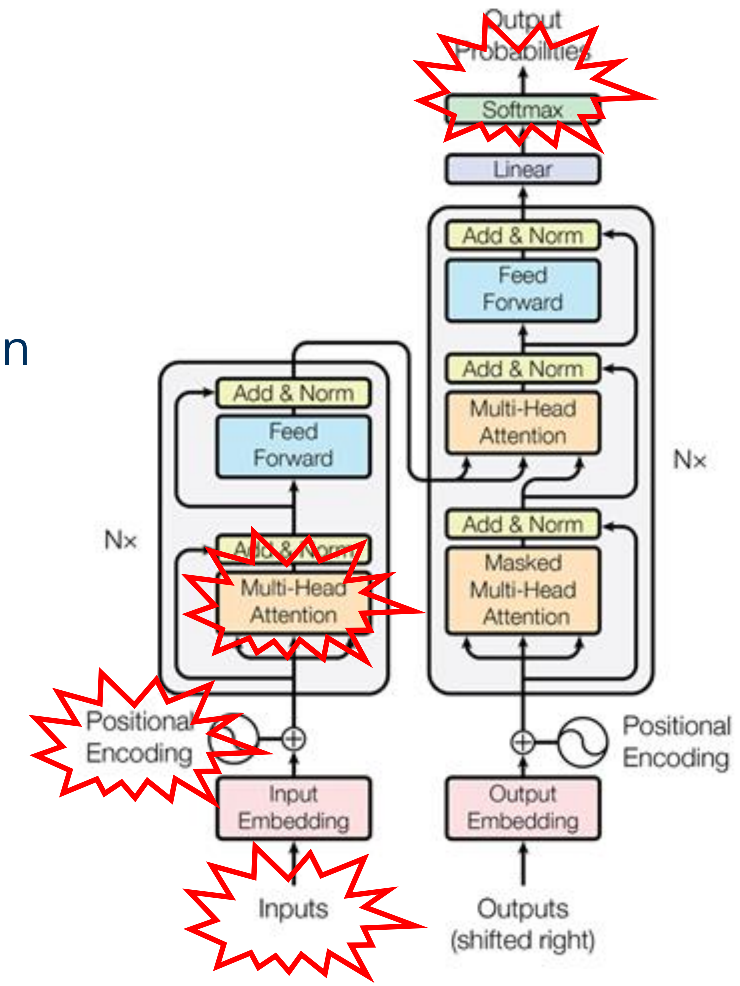
Challenges

Stationarity and distribution shifts

- Time series may change behavior over time (non-stationary)
- Transformers assume training and test data from the same distribution

No fixed vocabulary for output

- Transformers output a probability distribution over a fixed vocabulary
- In time series there is no vocabulary but we predict continuous values
- Output is a real number, not a class or token



Source: Vaswani et al (2017), https://arxiv.org/abs/1706.03762

Transformers for Time Series Analysis

Approaches

Tokenization

- Each timestep is a token, embeddings for additional features (e.g. Time2Vec)
- Scaling and quantization of continuous values into fixed vocabulary

Positions

- Timestamp encoding as additional positional information
- Time-aware or learnable embeddings to learn positions during training

Attention

- Sparse attention mechanism (local neighbors only, like CNN)
- Low-rank approximations

Output

- Linear output layer, regression losses for training
- Auto-correlation layers, probabilistic forecast layers

Transformers for Time Series Analysis

Approaches – Frameworks and Architecture Adaptions

Time Series Forecasting

- Informer
- PatchTST
- Temporal Fusion Transformer
- CHRONOS
- …

Spatio-Temporal Forecasting

- Earthformer
- …

Overviews and Surveys

- LLM for Time Series and Spatio-Temporal Data: https://arxiv.org/pdf/2310.10196
- Time-Series Transformer Review: https://github.com/qingsongedu/time-series-transformers-review

Python Libraries for Time Series Analysis

Python Libraries

- statsmodels: statistical models and time series analysis
- darts (u8darts): models and methods for time series forecasting and anomaly detection
- autogluon: AutoML predictors for tabular, multimodal and time series data
- optuna: hyperparameter optimization framework for machine learning

There are more: statsforecast, AutoTS, pytorch-forecasting, tsai, raytune, …