TECHNISCHE UNIVERSITÄT DRESDEN

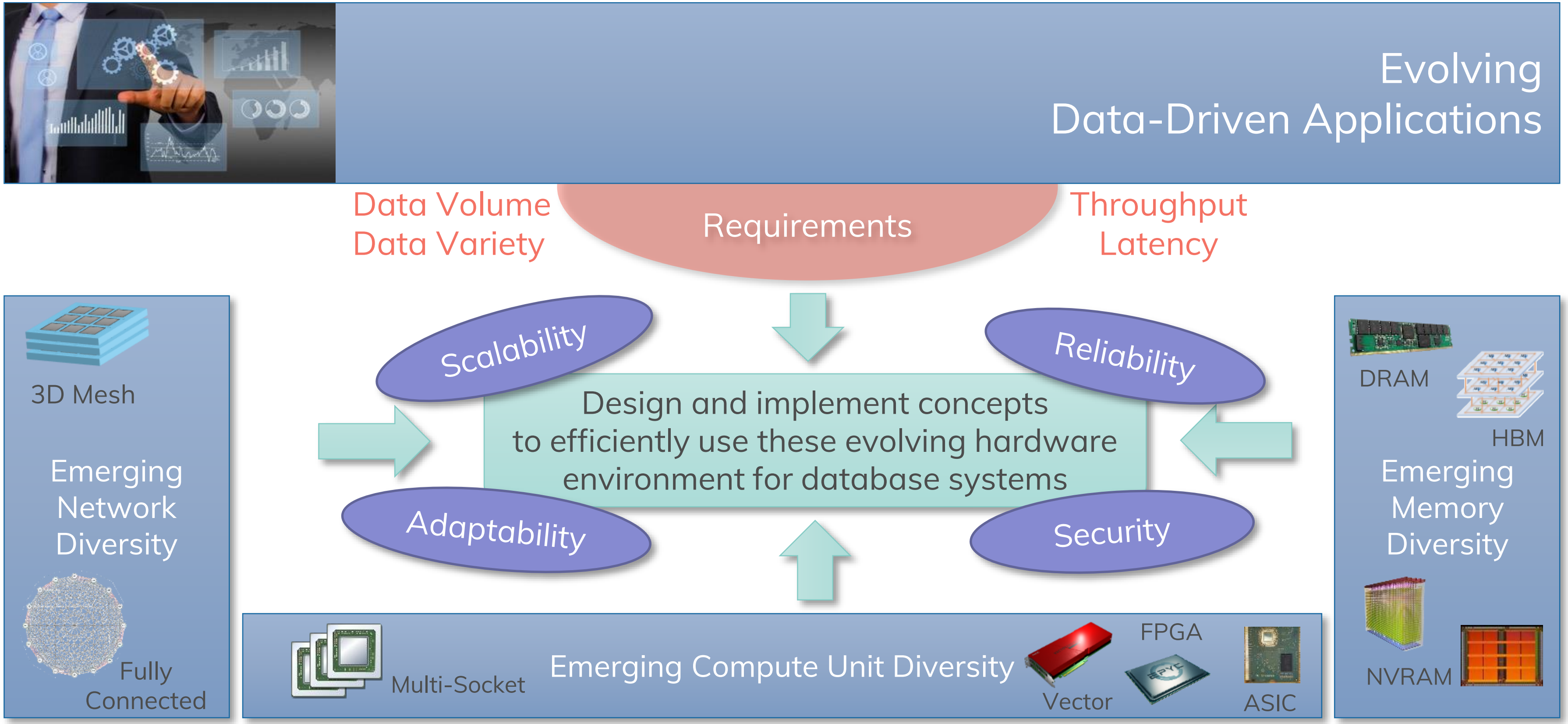
Dresden Database Systems Group

Beyond the Crystal Ball – Time Series Analysis and Forecasting

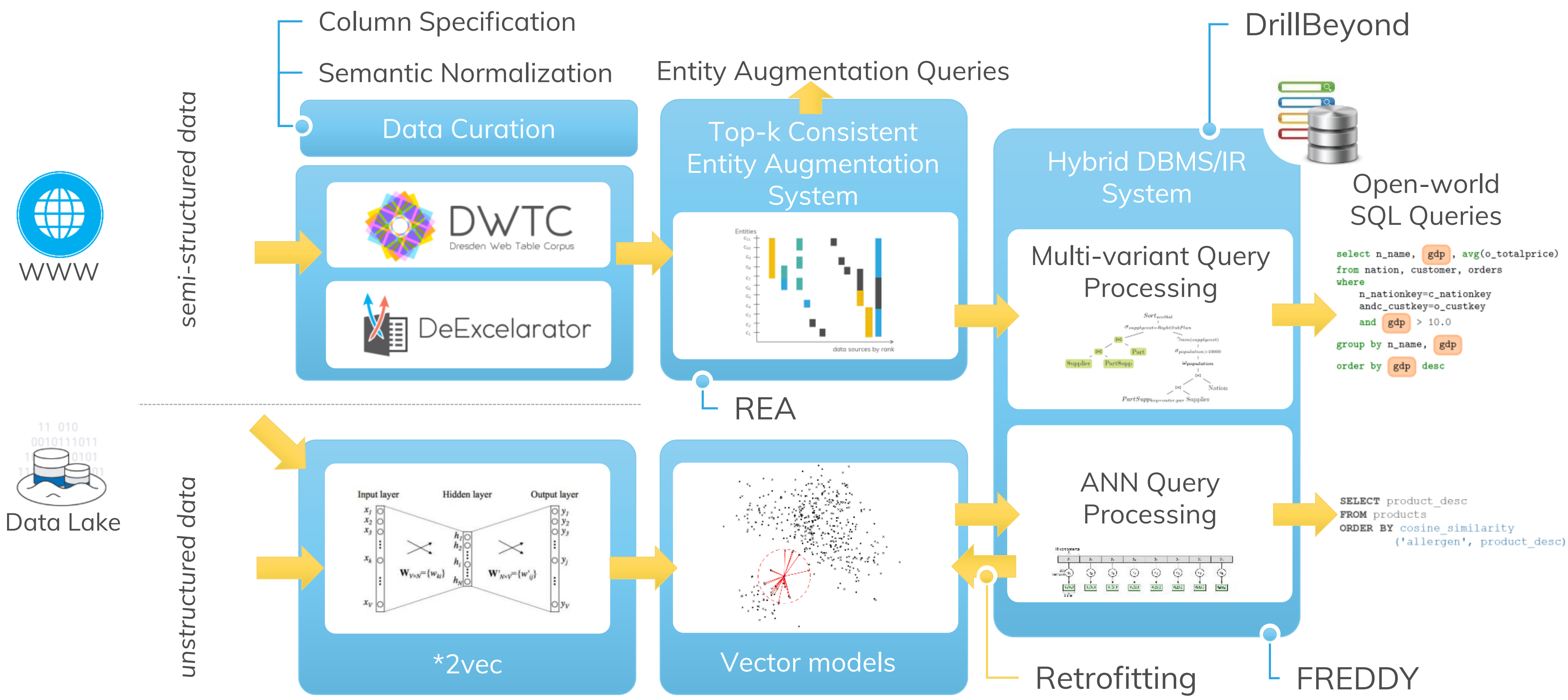
Claudio Hartmann – Lehrstuhl Datenbanken

System Architecture

Dresden Database Systems Group



Documents and Information Retrieval



Data Analytics

Dresden Database Systems Group

Support/Extend Database Technology

- Sampling Techniques
- Integration of Machine Learning into RDBMS
- Cardinality estimation using Artificial Neural Networks
- Monitoring database health status

Working with Data

- Time series Forecasting
- Time series properties and generation
- Clustering with human feedback

Flash Forward Query

ReEF
Renewable Energy Forecasting

GOFLEX

Agenda

Dresden Database Systems Group

What is a time series?

Health of a Database System

- Monitoring and Target
- Techniques

Feature-based Time Series Engineering

- Time Series Features
- What-if analysis

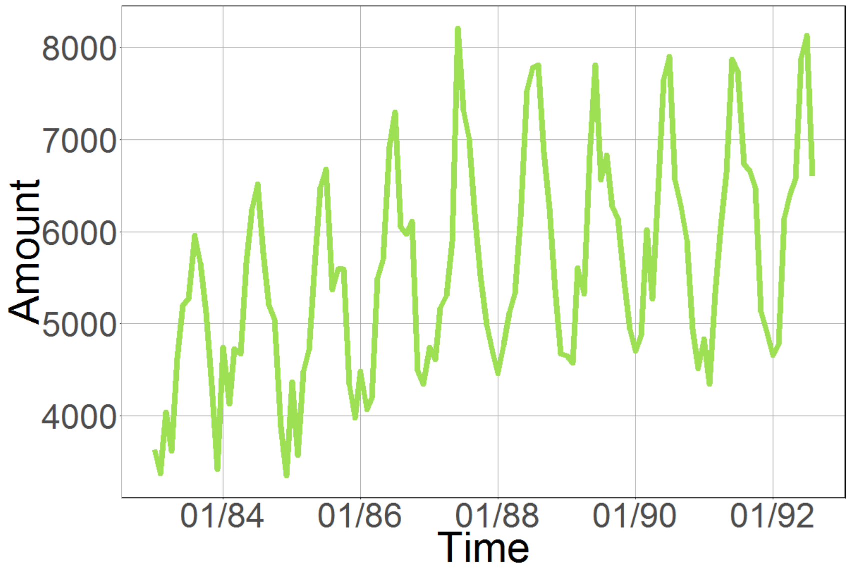
Forecasting Large-scale Time Series Data

- Forecasting Process
- Big Data Implications
- CSAR

Terminology

Time Series

- Sequence of measure values
- Ordered by time
- Equidistant
  - Constant time distance between measure values
- Complete
  - No missing values



Health of a Database System

Health of a Database System

Dresden Database Systems Group

Monitoring

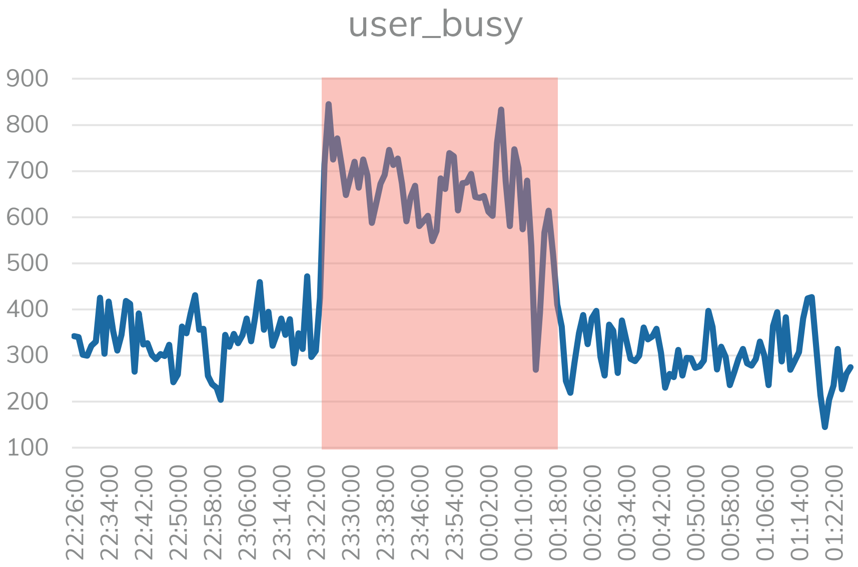
- Several measurements that serve as indicator for the health status of the database

Target

- Issue automatic warning for situations that would lead to customer complaints

Possible Health indicators

- CPU (user_busy, io_busy, system_busy, idle)
- user connections
- physical reads/writes, …



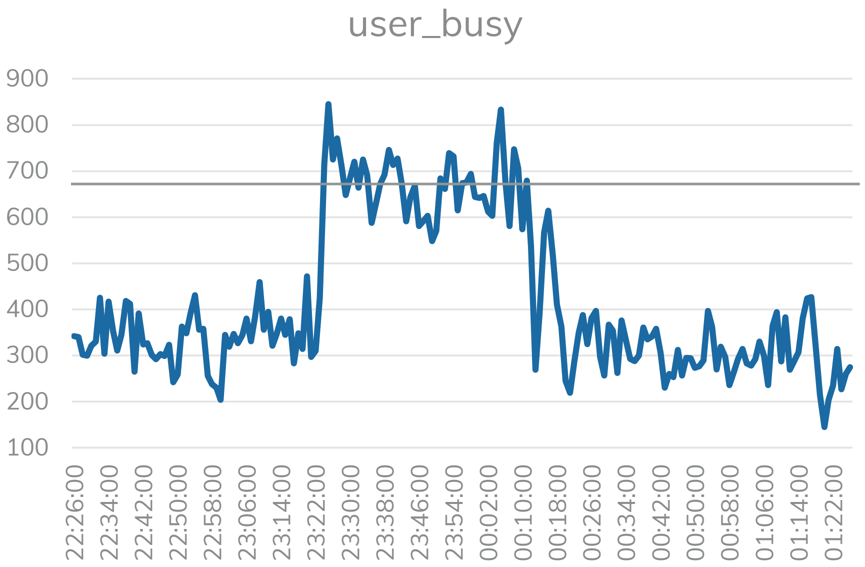
Health of a Database System

Threshold/Limits

- Define upper threshold
- Issue warning when threshold is exceeded

Problem

- Manual work
- Proper definition of threshold
  - Too high → Warning too late
  - Too low → Waring when there is no issue

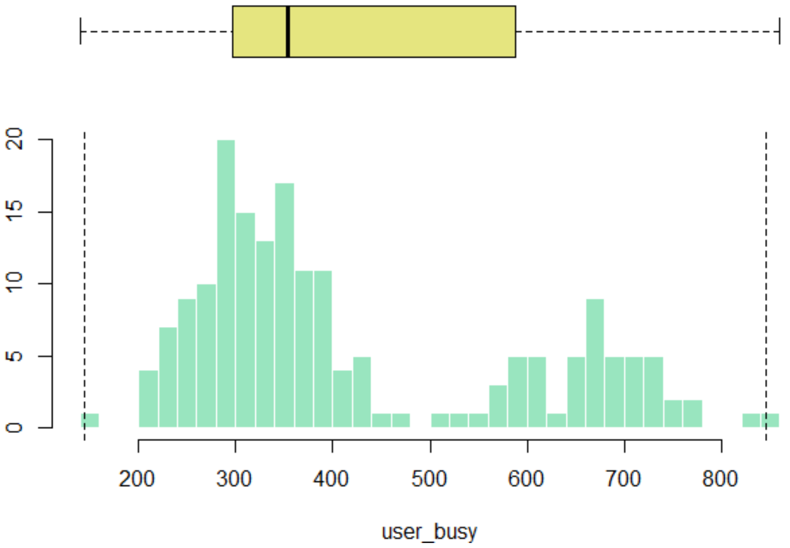


Health of a Database System

Dresden Database Systems Group

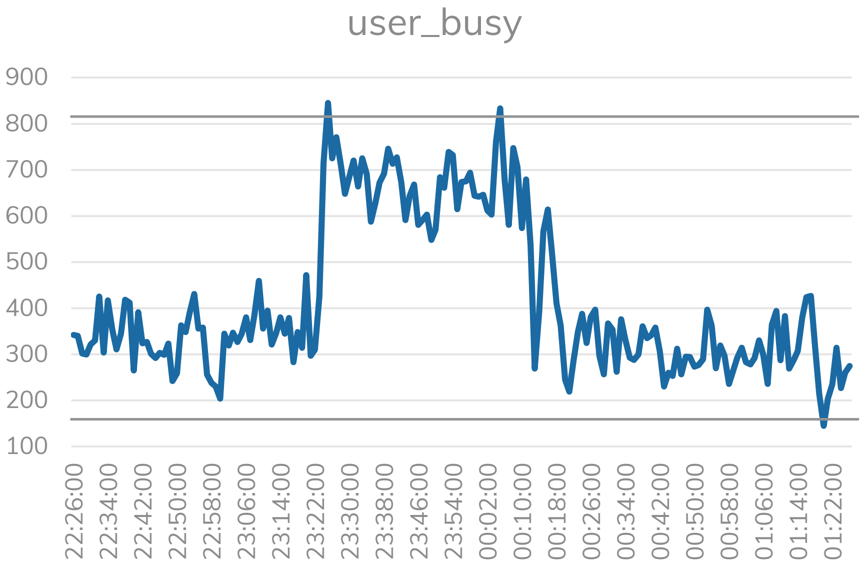
Automatic threshold using IQR

- Define threshold based on box plot statistics
- Issue warning when threshold is exceeded



Problem

- May detect normal states as outliers



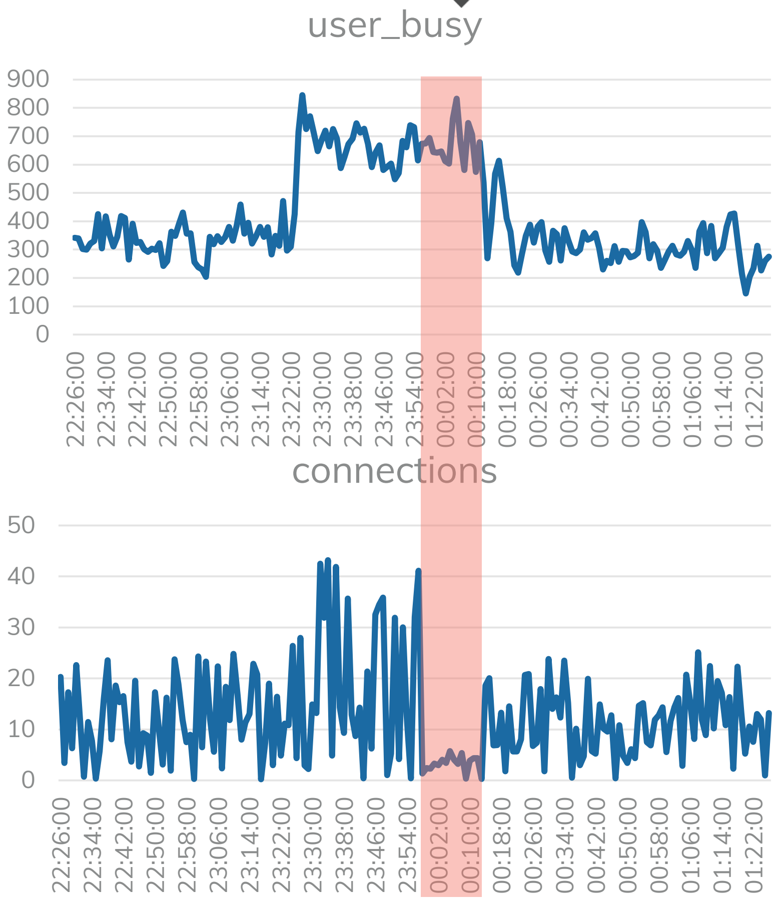
Health of a Database System

Co-development of measurements

- Examine proportions of several measurements
- Issue warnings for abnormal proportions

Problem

- Know all effects in advance
- Lots of manual work



Health of a Database System

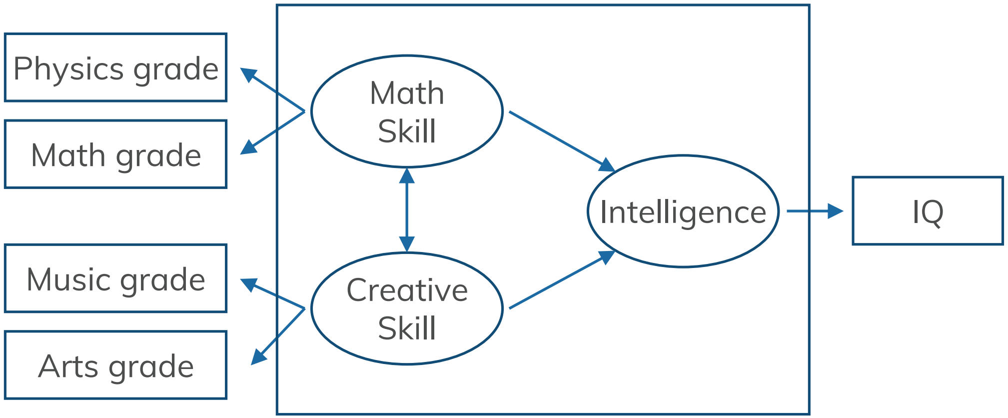
Dresden Database Systems Group

Structural Equation Modeling

- Model influences between system components
- Analysis of Latent Factors

Model Properties

- Item
  - Measurable variables
- Latent Factor
  - Not measurable variables
- Measurement model
  - Model connections between items and factors using covariance



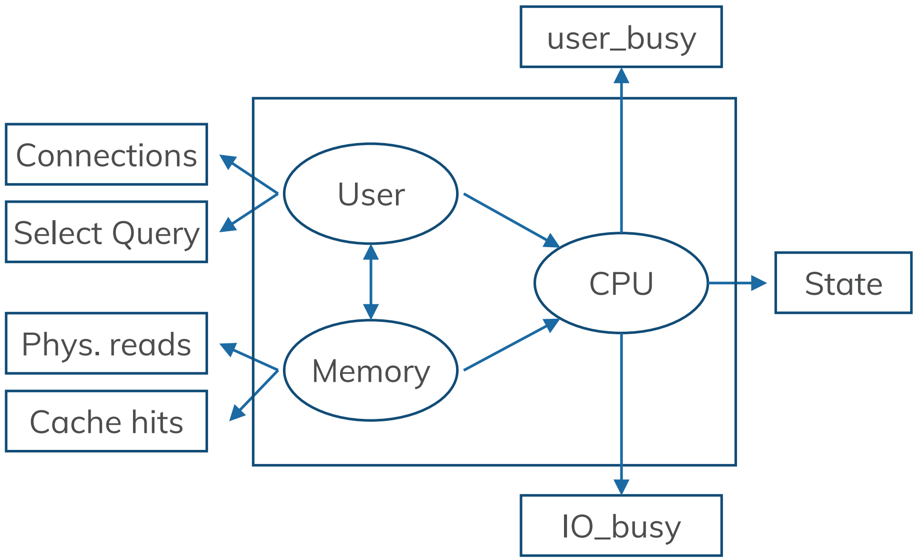
Health of a Database System

Structural Equation Modeling

- Model influences between system components
- Analysis of Latent Factors

Model Properties

- Item
  - Measurable variables
- Latent Factor
  - Not measurable variables
- Measurement model
  - Model connections between items and factors using covariance

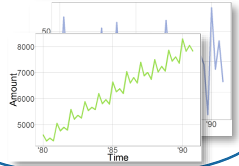


Feature-based Time Series Engineering

Process of Time Series Engineering

Dresden Database Systems Group

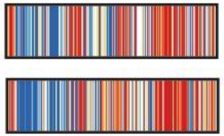
Time Series Dataset



Representation Engineering

Represent

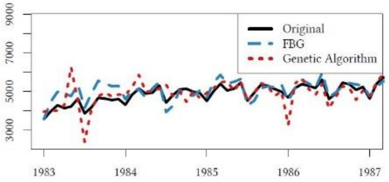
Representation Dataset



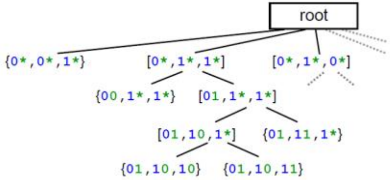
Distance

Data-mining Tasks

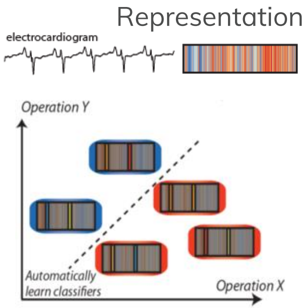
Generation



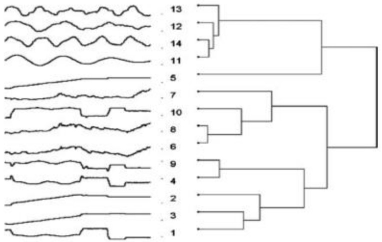
Indexing



Classification



Clustering

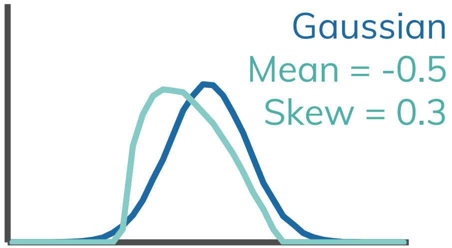


Figures: Kegel et al., 2018; Shieh and Keogh, 2013; Fulcher et al., 2013; Wang et al., 2006

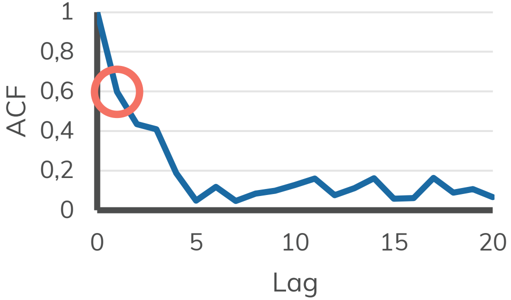
Example Time Series Features

Dresden Database Systems Group

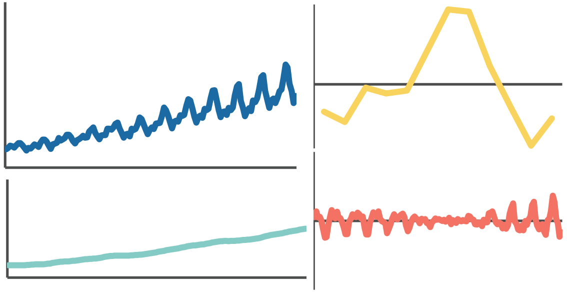
Distribution



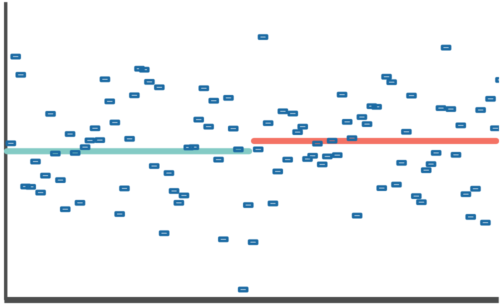
Correlation



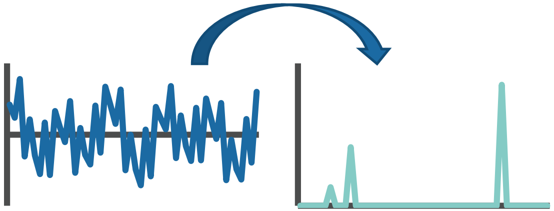
Components



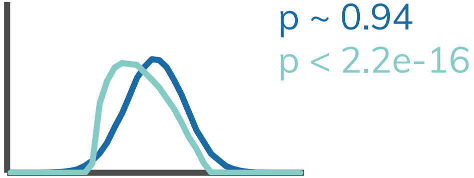
Stationarity



Frequency Domain



Tests



p-value of Normality test

Time Series Components

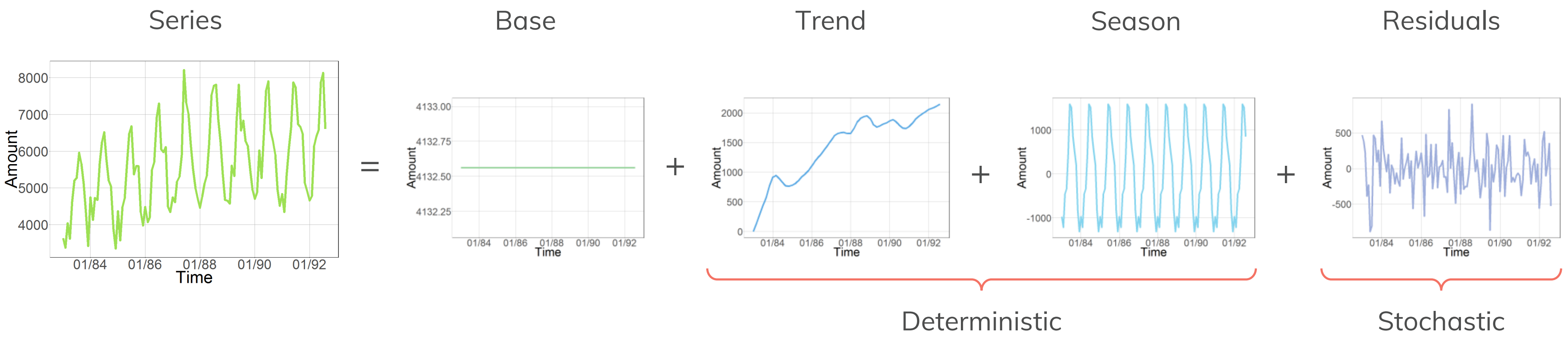
Dresden Database Systems Group

Decomposition

- Base: stationary part of the time series
- Trend: long-term change in the mean level
- Season: cyclical repeated behavior
- Residuals: unstructured information assumed to be random

Often, base is part of the trend component!

Additive Composition $x_t = base_t + \ trend_t + season_t + res_t$



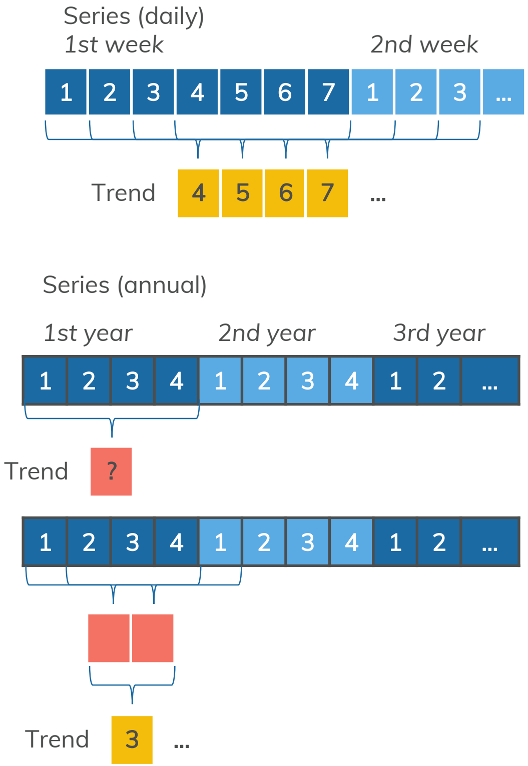
Time Series Decomposition

Basic Idea

- Moving-average filter of continuous windows
- $tr_t = \frac{x_t + x_{t-1} + x_{t-2} + \dots + x_{t-N+1}}{N}$
- Extraction of trend by windows that take into account the season length
- Extraction of season by averaging each time instance of the same seasonal position (all Mondays, all Tuesdays,…)
- Disadvantage: does not decompose the endpoints

Average Centering

- A technique needed if season length is even
- Take two moving averages and average their result

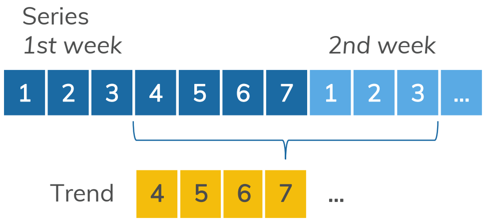


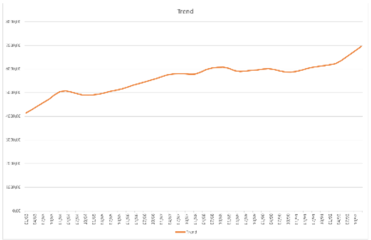
A season of length 7 such as
- Monday (1)
- Tuesday (2)
- …

Time Series Decomposition

Retrieval of Trend

- Moving-average filtering
- In case of even season length, centralize first



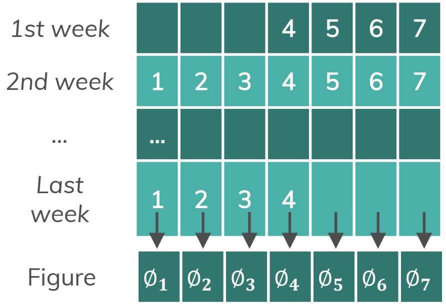


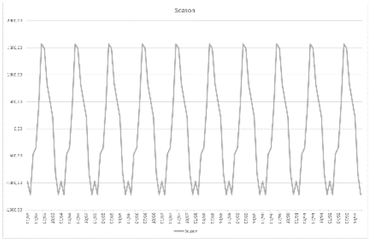

```
trend <- filter(series, rep_len(1,7)/7)
```

Season Retrieval

- Detrend series
- Figure represents the average of each time instance of a season
- De-mean figure

```
detrend <- series - trend
```



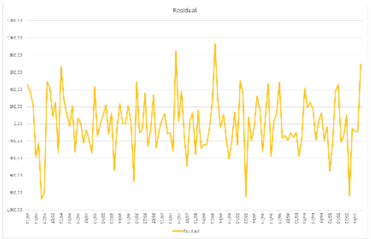



```
season <- figure - mean(figure)
```

Residuals

- Subtract components

```
residuals <- series - season - trend
```



Dresden Database Systems Group

TECHNISCHE UNIVERSITÄT DRESDEN

Feature-based Generation Method (FBG)

Dresden Database Systems Group

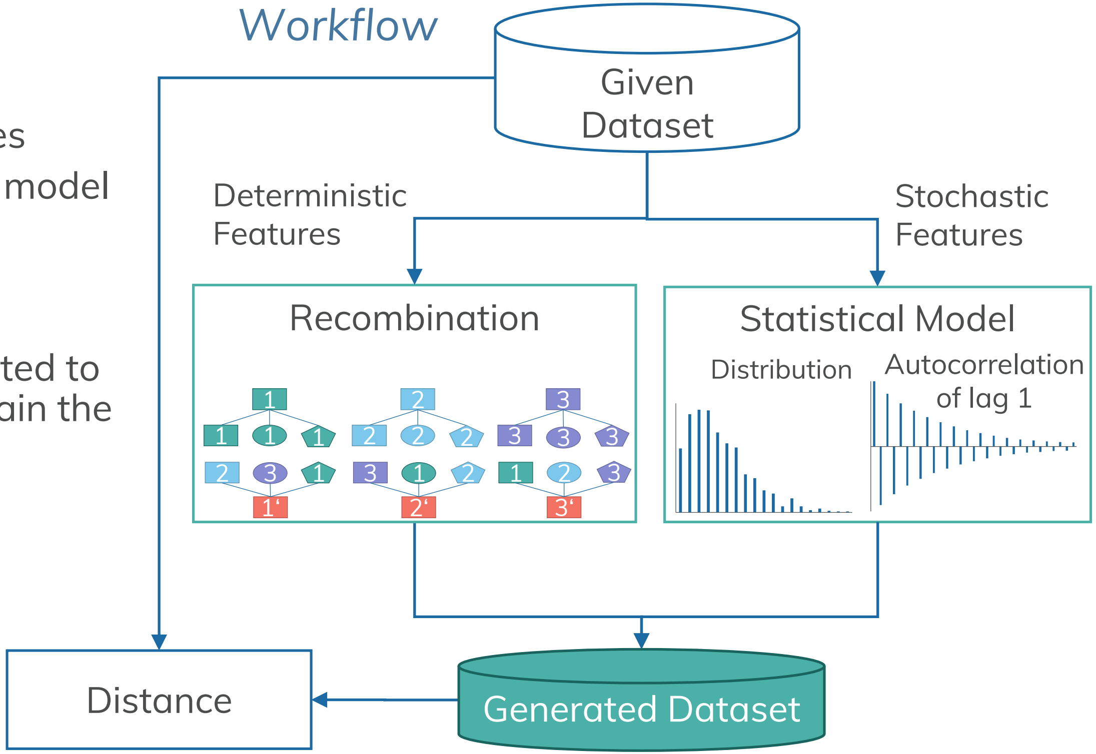
Idea

- Feature-based representation
- Recombination of deterministic features
- Simulation of residuals with statistical model

Use cases

- Anonymization of data sets
- Generate a data set that is closely related to the original data set but does not contain the actual data

Workflow

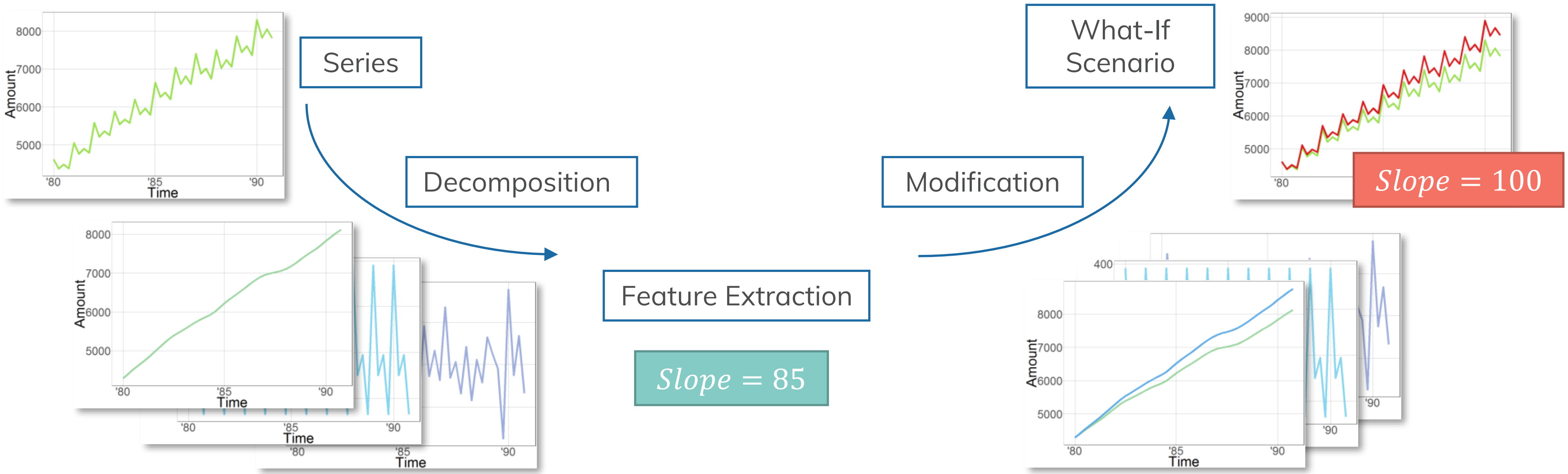


Kegel et al., 2018

What-if Analysis

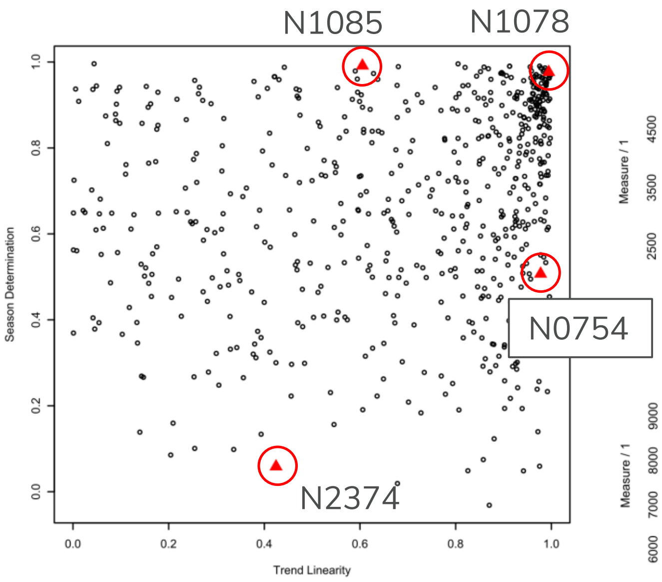
General Idea

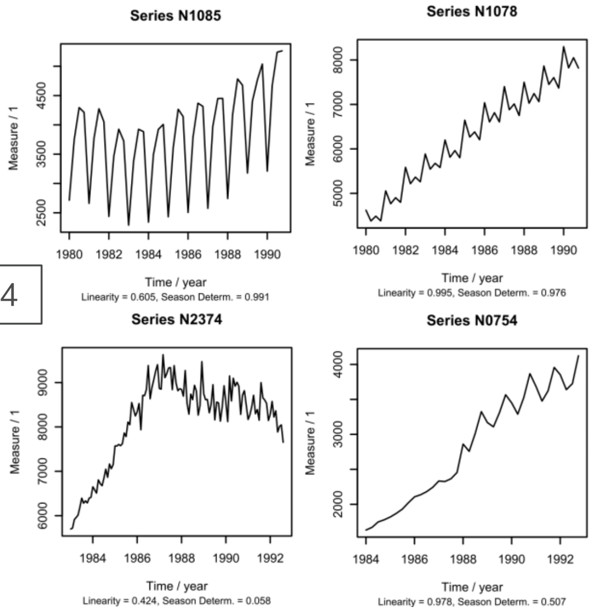
- Represent time series by their features
- Generate a what-if scenario by setting factors that modify features



Feature Extraction

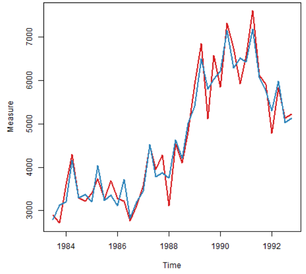
Feature Space and Selected Time Series



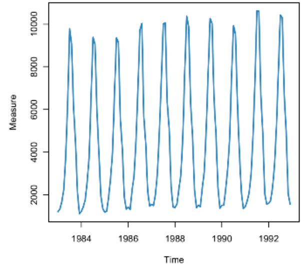


Season Determination R^2_{seas}

- $R^2_{seas} = 1 - \frac{var(res_t)}{var(res_t + seas_t)}$



Weak Season

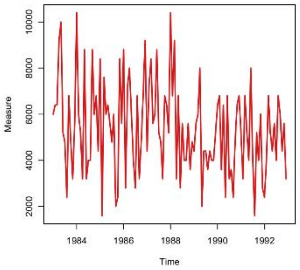


Strong Season

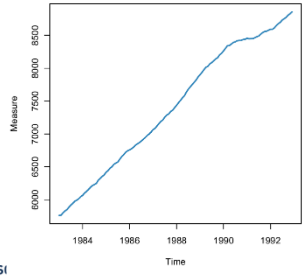
Feature Extraction

Trend Determination R^2_{tr}

- $R^2_{tr} = 1 - \frac{var(res_t)}{var(res_t + tr_t)}$



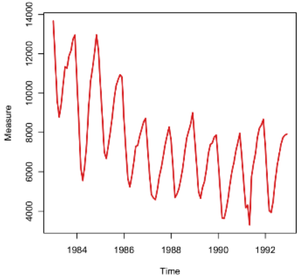
Weak Trend



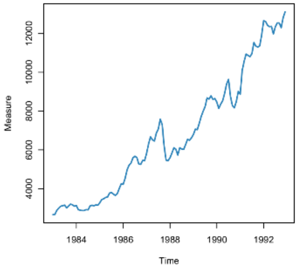
Strong Trend

Trend Slope θ_2

- Suppose a linear trend within STL trend: $tr_t = \theta_1 + \theta_2 \cdot l_t + \delta_t$



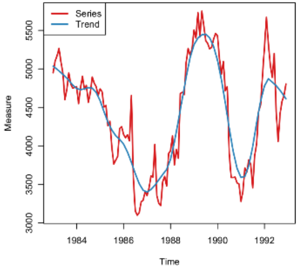
Negative Trend



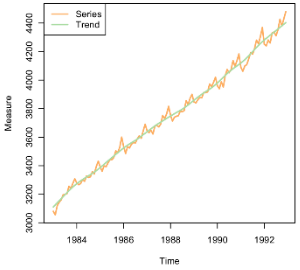
Positive Trend

Trend Linearity R^2_{lin}

- $R^2_{lin} = 1 - \frac{var(\delta_t)}{var(tr_t)}$



Non-linear Trend



Linear Trend

Component Modification

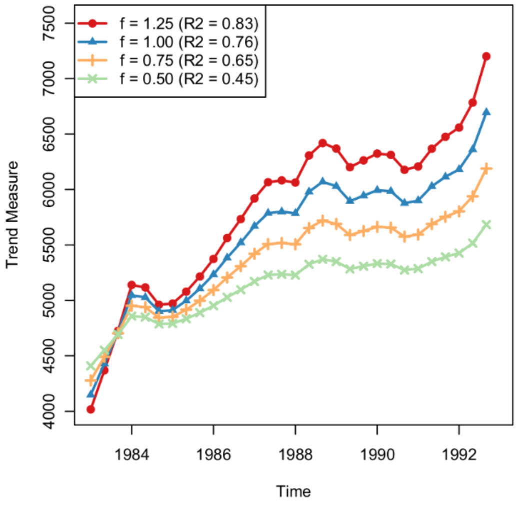
Dresden Database Systems Group

Trend Determination Factor

- Strengthen/weaken trend

$$tr_{t,f} = \theta_1 + f \cdot (\theta_2 \cdot l_t + \delta_t)$$

$$R^2_{tr} = 1 - \frac{var(res_t)}{var(res_t + tr_{t,f})}$$

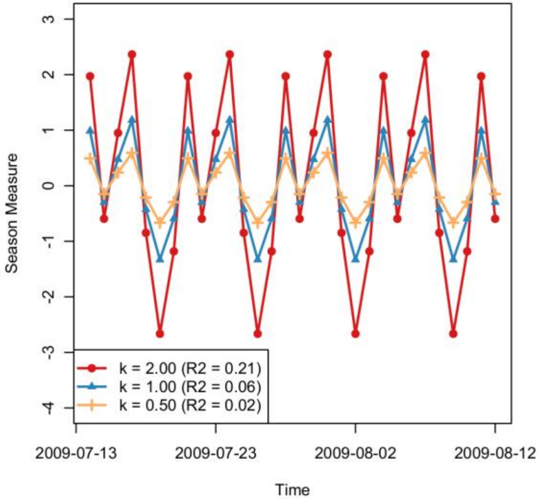


Season Determination Factor

- Strengthen/weaken season

$$seas_{t,k} = k \cdot seas_t$$

$$R^2_{seas} = 1 - \frac{var(res_t)}{var(res_t + seas_{t,k})}$$



Forecasting Large-scale Time Series Data

Forecasting Process

Dresden Database Systems Group

1. Model Identification

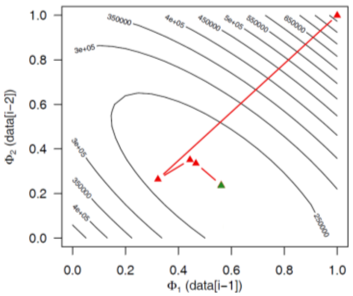
Method	Error	AIC
ARIMA	23.47	25.72
HoltWinters	18.38	20.43
VAR		
MARS		

Optimize

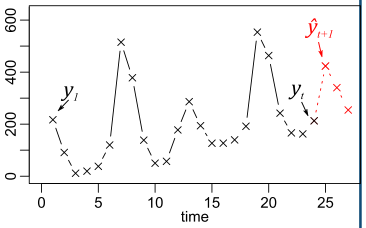
Predict

Evaluate

2. Model Estimation



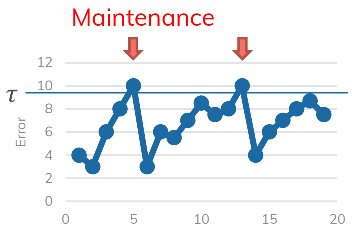
3. Forecast Calculation



4. Model Evaluation

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|x_t - \hat{x}_t|}{(x_t + \hat{x}_t)/2}$$

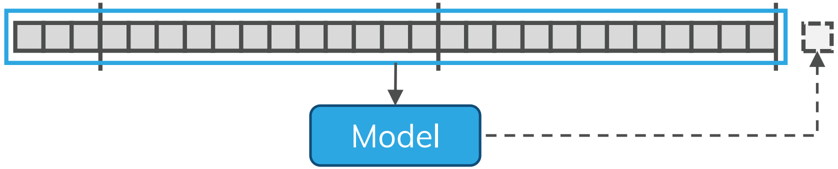
5. Model Adaptation



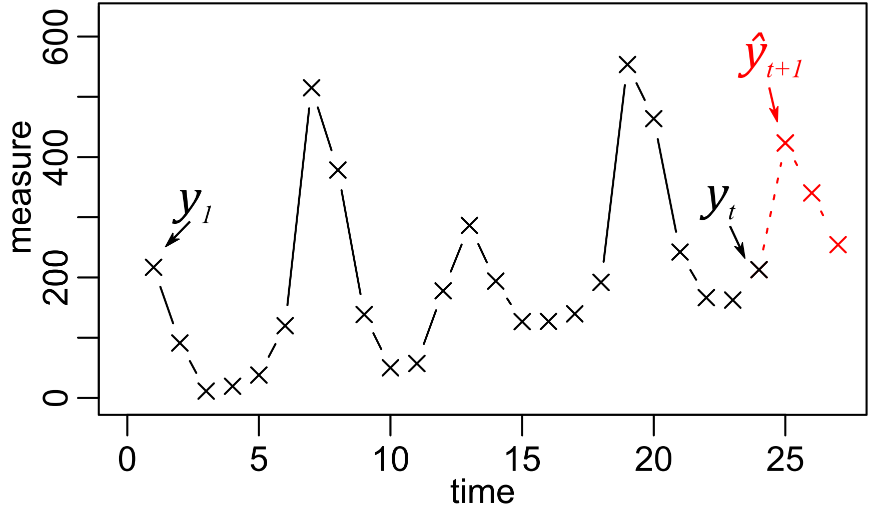
Traditional Time Series Forecasting

Univariate Forecast Models

- ARIMA/Exponential Smoothing
- Focus on only one time series at a time
- Widely applied in many domains
- auto.ARIMA/ETS to properly configure the model for a given time series



$$\hat{y}_{t+1} = c + \epsilon_t + \sum_{i=1}^{p} \phi_i y_{t-i+1} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j+1}$$



Large-scale Time Series Data

Dresden Database Systems Group

Many and Long Series

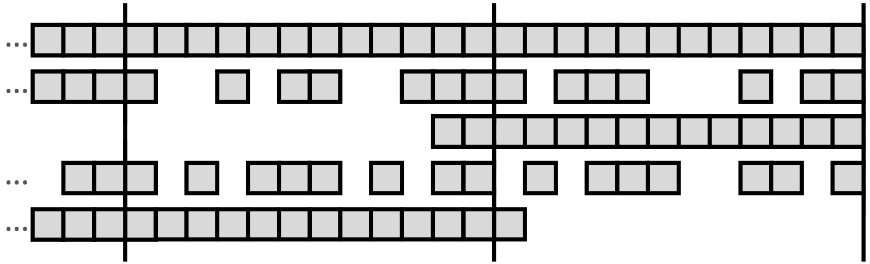
- High number of monitored objects (Smart Meter in every household, sales of individual products)
- Fine monitoring granularity leads to very long time series histories

High Levels of Noise

- Time series on fine granularity tend to be very noisy

Missing values

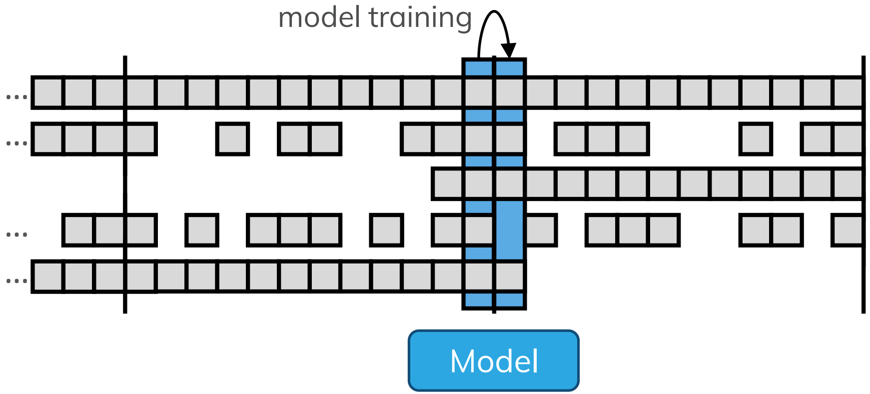
- Missing values lead to inapplicability of all most conventional models



Cross-sectional Forecasting

Core Approach

- Represent a whole data set with one model
- Focus on cross-sections instead of the entire time series
- Model transition from one cross-section to the next one
- All time series with values in the blue cross-sections contribute to the model training
- The model represents the average transition of the entire data set



$$\hat{\vec{y}}_{t+1} = c + \phi_1 \cdot \vec{y}_t$$

Cross-sectional Forecasting

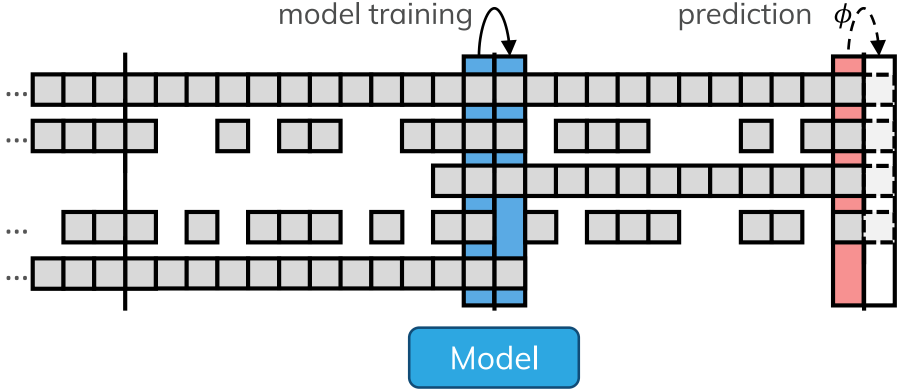
Dresden Database Systems Group

Core Approach

- Represent a whole data set with one model
- Focus on cross-sections instead of the entire time series
- Model transition from one cross-section to the next one
- All time series with values in the blue cross-sections contribute to the model training
- The model represents the average transition of the entire data set

Model Application

- Assume the transition remains constant over seasons
- Apply model on the most current data
- Train a specific model for every transition in a season



$$\vec{\hat{y}}_{t+1} = c + \phi_1 \cdot \vec{y}_t$$

Still misses adaptability!

CSAR – Autoregression

Dresden Database Systems Group

Non-seasonal Autoregression

- Model the dependency of future values of their direct predecessors

$$\hat{\vec{y}}_{t+1} = c + \phi_1 \cdot \vec{y}_t + \cdots + \phi_p \cdot \vec{y}_{t-p+1}$$

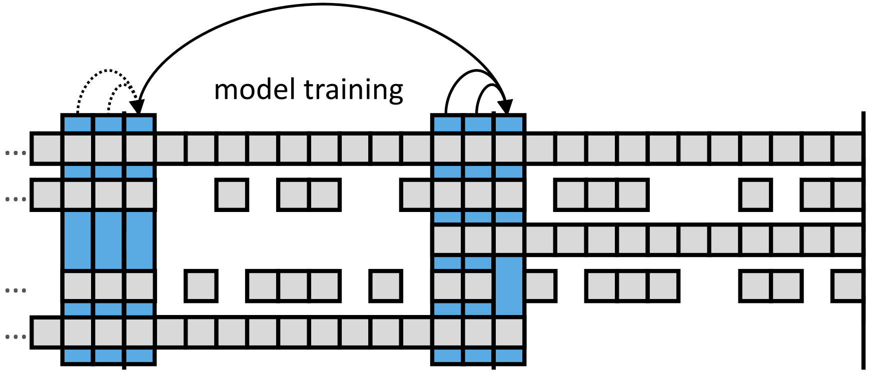
Seasonal Autoregression

- Model the dependency of future values of their seasonal predecessors

$$\hat{\vec{y}}_{t+1} = c + \Phi_1 \cdot \vec{y}_{t-s+1} + \cdots + \Phi_P \cdot \vec{y}_{t-P \cdot s+1}$$

Correction Terms

- Necessary if non-seasonal and seasonal components are combined



$$\hat{\vec{y}}_{t+1} = c + \phi_1 \cdot \vec{y}_t$$

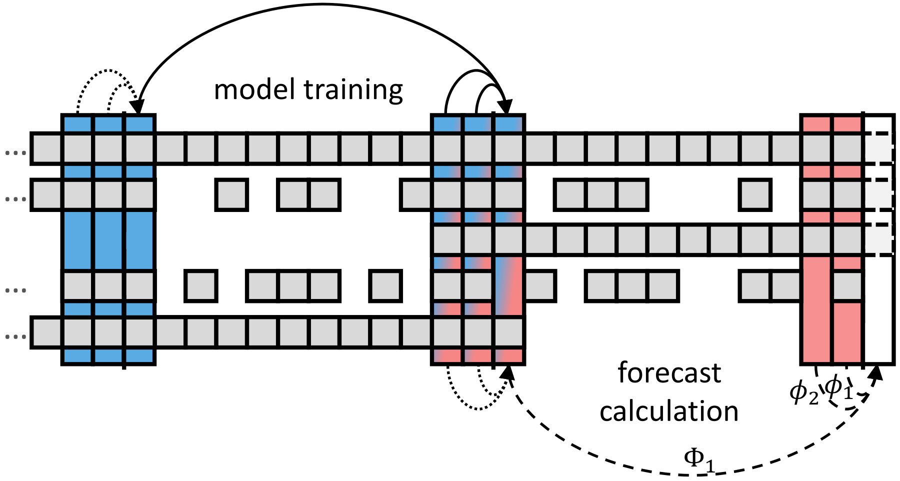
CSAR – Autoregression

Model Application

- On most recent data
- On all time series which have all necessary values

Behavioral Deviation

- Some time series do not follow the common behavior of the majority



$$\hat{\vec{y}}_{t+1} = c + \phi_1 \cdot \vec{y}_t + \phi_2 \cdot \vec{y}_{t-1} + \Phi_1 \cdot \vec{y}_{t-s+1} \\ +(-\Phi_1\phi_1) \cdot \vec{y}_{t-s} +(-\Phi_1\phi_2) \cdot \vec{y}_{t-s-1}$$

CSAR – Error Terms

Dresden Database Systems Group

Non-seasonal Error Terms

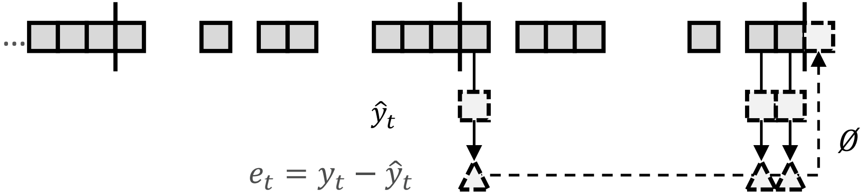
- Adjust forecasts according to systematic non-seasonal misprediction
- Deviation of individual time series from the general model for the data set

Seasonal Error Terms

- Adjust forecasts according to systematic seasonal misprediction
- Systematic deviation in the seasonal pattern

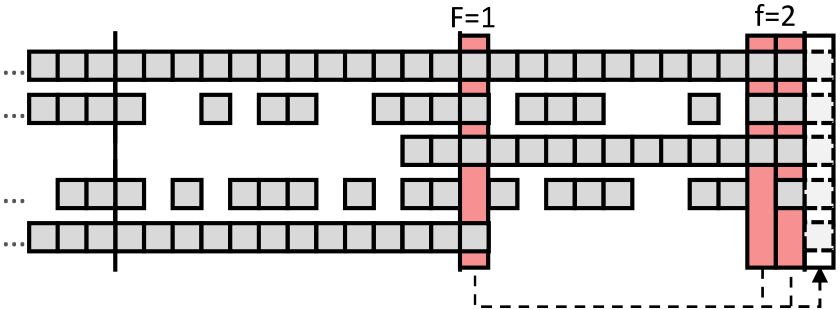
Missing Data

- If real value or forecast is missing, the value is ignored



$$\overline{e^n_{t+1}} = \frac{1}{f+F}\left(\sum_{i=0}^{f-1} e^n_{t-i} + \sum_{j=1}^{F} e^n_{t+1-j\cdot s}\right)$$

$$\hat{y}^n_{t+1} = c + \overline{e^n_{t+1}}$$



CSAR – Model Components

Autoregression

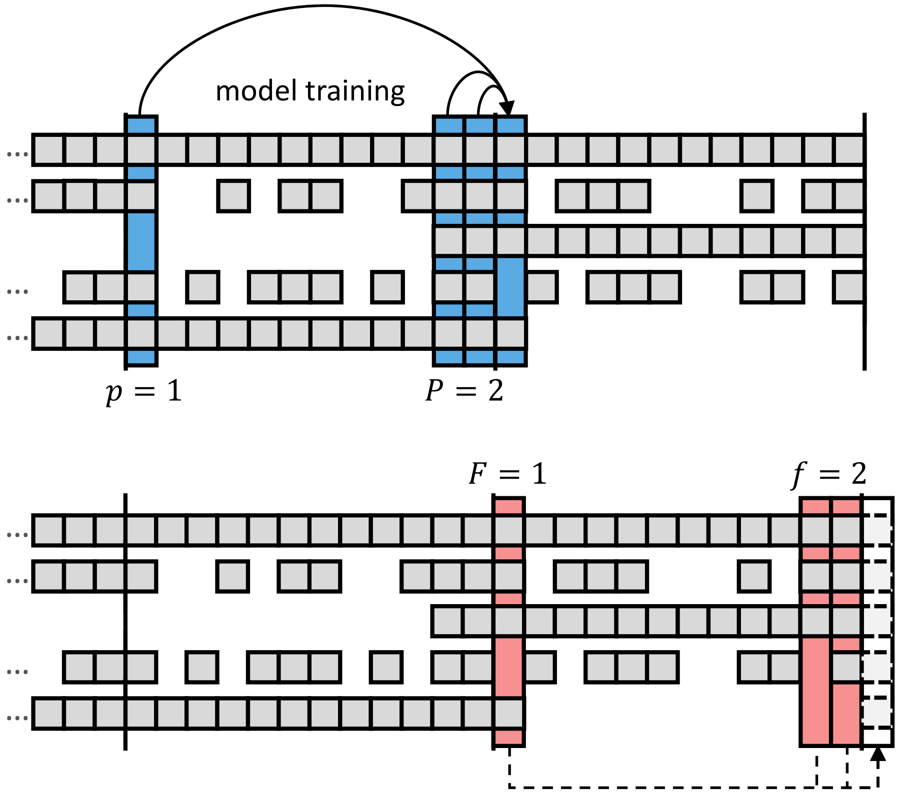
- p specifies number of non-seasonal AR terms
- P specifies number of seasonal AR terms

Error Terms

- f specifies number of non-seasonal error terms
- F specifies number of seasonal error terms

Model Configuration

- Adaptation to data set specific characteristics
- Influence on forecast accuracy
- Influence on execution time



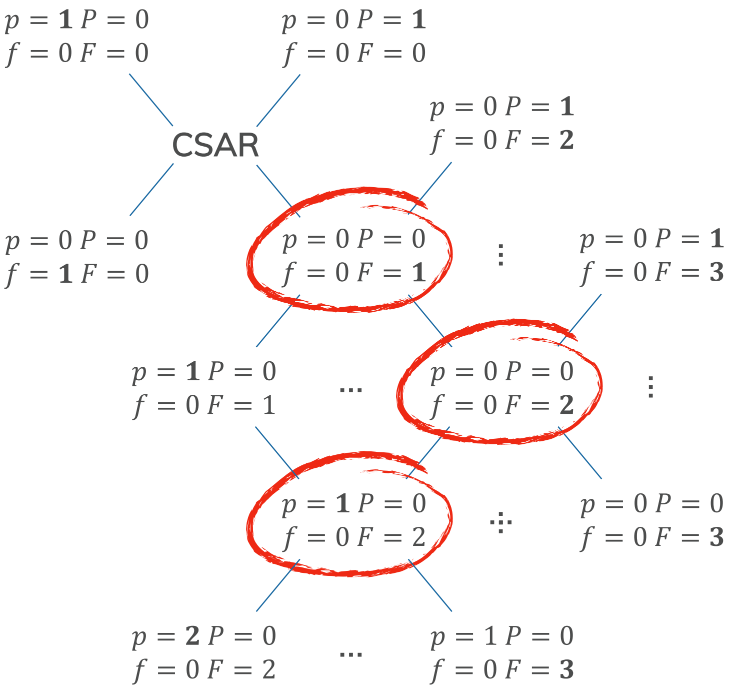
auto.CSAR – Structured Greedy Search

Step 1 – Base Models

- Compare basic model components
- Choose model with lowest error to identify most important model component

Step 2 – Search Around the best Model

- Based on result from Step 1
- Vary optimal model components by +/- 1
- Vary seasonal and non-seasonal components by +/- 1
- Invert the constant
- Repeat until an iteration returns no new best model

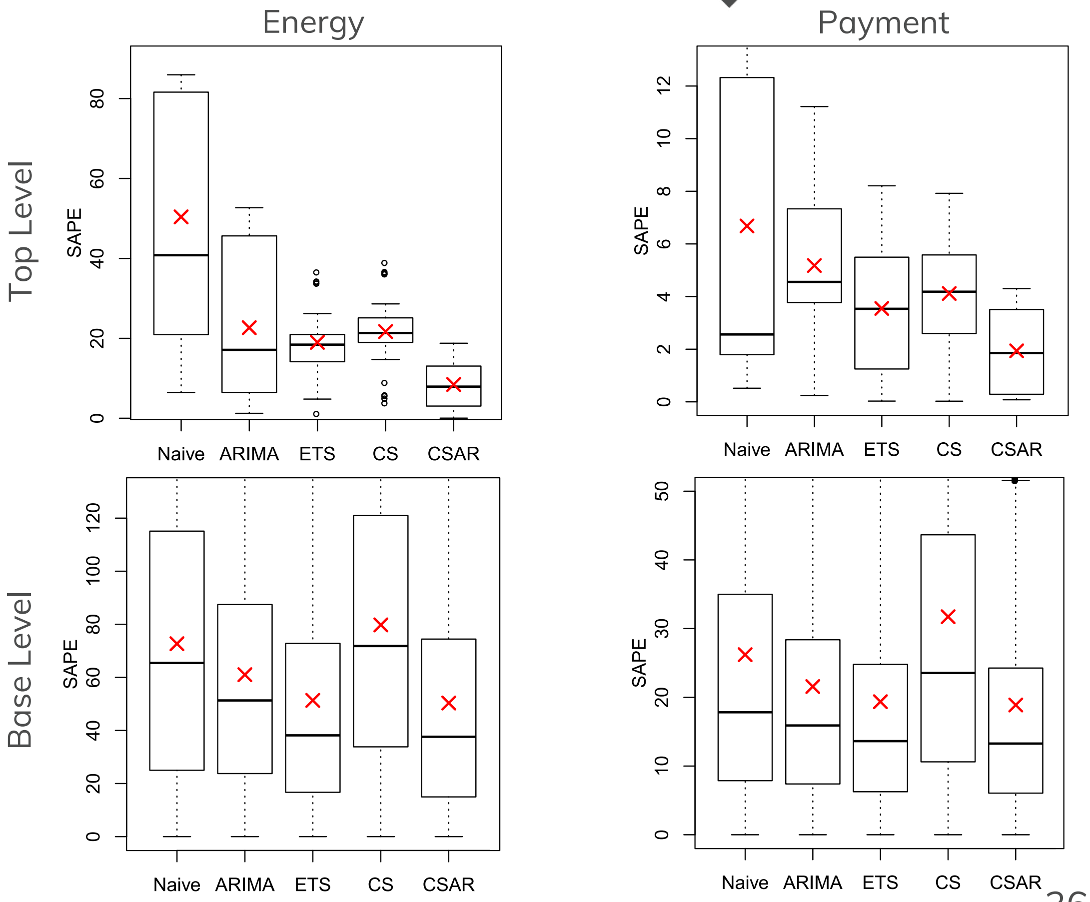


Forecast Accuracy

Experimental Set-Up

- Calculate long-range forecasts
  - Energy – 1 week (h=28)
  - Payment – 2 weeks (h=14)
- Calculate SAPE error between forecast and real time series values (Symmetric Absolute Percentage Error)

$$SAPE = \frac{|y - \hat{y}|}{(|y| + |\hat{y}|)/2} \cdot 100$$



Data Set Partitioning

Univariate vs. CSAR

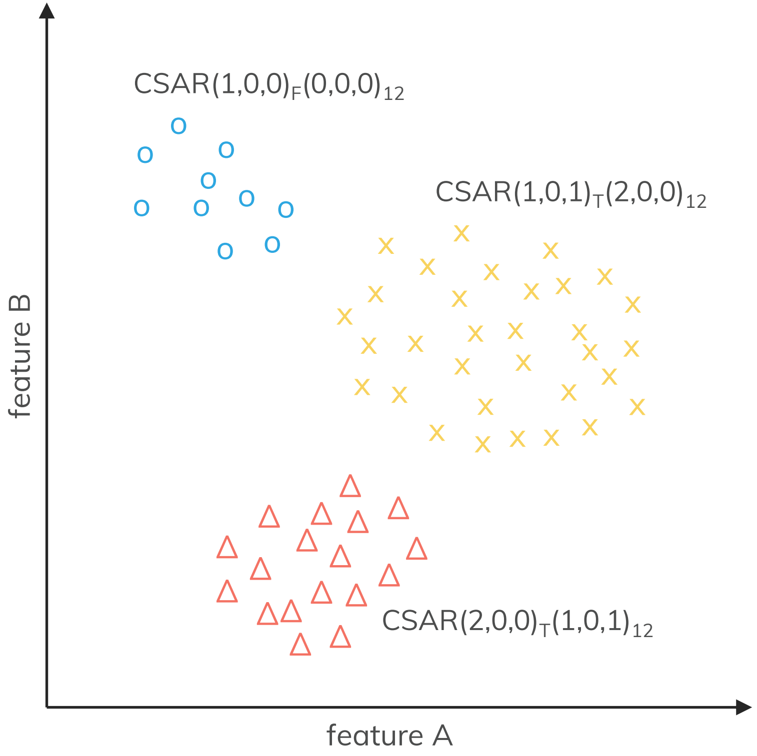
- Opposite extremes
- Univariate one model per time series
- CSAR one model for all time series

Partitioning

- Split data set into several Partitions
- Create one CSAR model for each partition

Expectation

- Better representation of time series
- Higher forecast accuracy



Model Adviser

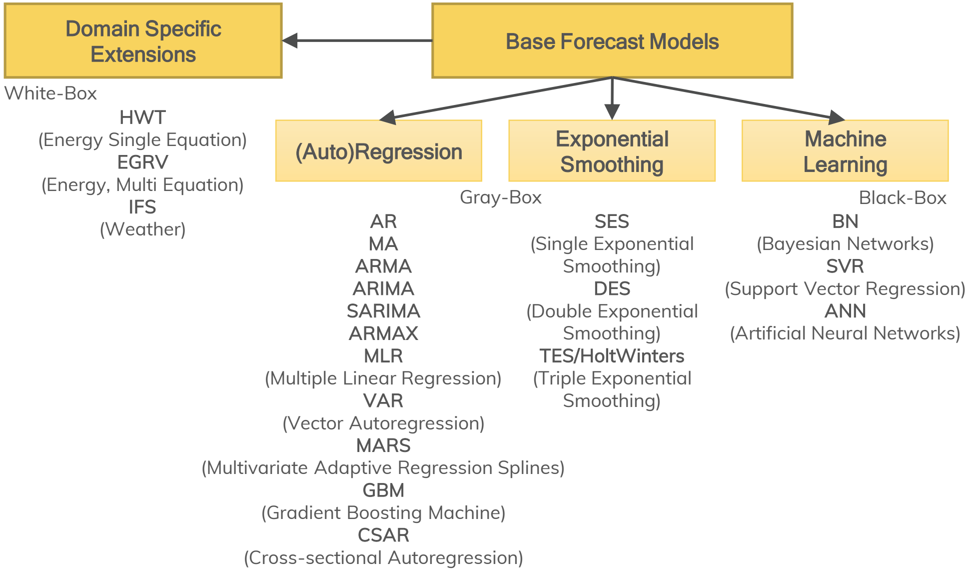
Dresden Database
Systems Group

Model Suitability

- Different models work well with different time
- Some are designed for very specific use cases

Which one to use?

- Not an easy decision for non-experts



Student Thesis Topics

Dresden Database Systems Group

Topics for Student Theses

https://wwwdb.inf.tu-dresden.de/study/theses/themen-fuer-arbeiten/

Student Theses at the Chair of Databases

Themenschwerpunkte

Aktuell werden studentische Arbeiten in folgenden Themenschwerpunkten vergeben:

Bereiche und aktuelle Angebote	Ansprechpartner
DB-Systemarchitektur, skalierbare und sichere Datenverarbeitung, DB-Systeme auf moderner Hardware Eine Auswahl aktueller Themen. ▪ Benchmark Design für adaptive Datenbanksysteme ▪ Integration des ERIS Storage Systems in Apache Spark ▪ Implementierung und Optimierung eines Codesgenerators für Kompressionsalgorithmen ▪ Evaluierung der Intel SGX Erweiterung für eine sichere Datenverarbeitung ▪ Implementierung und Evalution eines Storage Moduls für Graphdaten in ERIS	Dirk Habich
Informationsextraktion, Information Retrieval, Machine Learning Eine Auswahl aktueller Themen. ▪ Word2Vec-Modell über Webtabellen ▪ Column-specification mit System T ▪ Zeitreihenanalyse und -prognose	Maik Thiele
Für weitere individuelle Angebote kontaktieren Sie bitte den zuständigen Ansprechpartner.	