Grid Load Peak Prediction with sonnen
Final Presentation

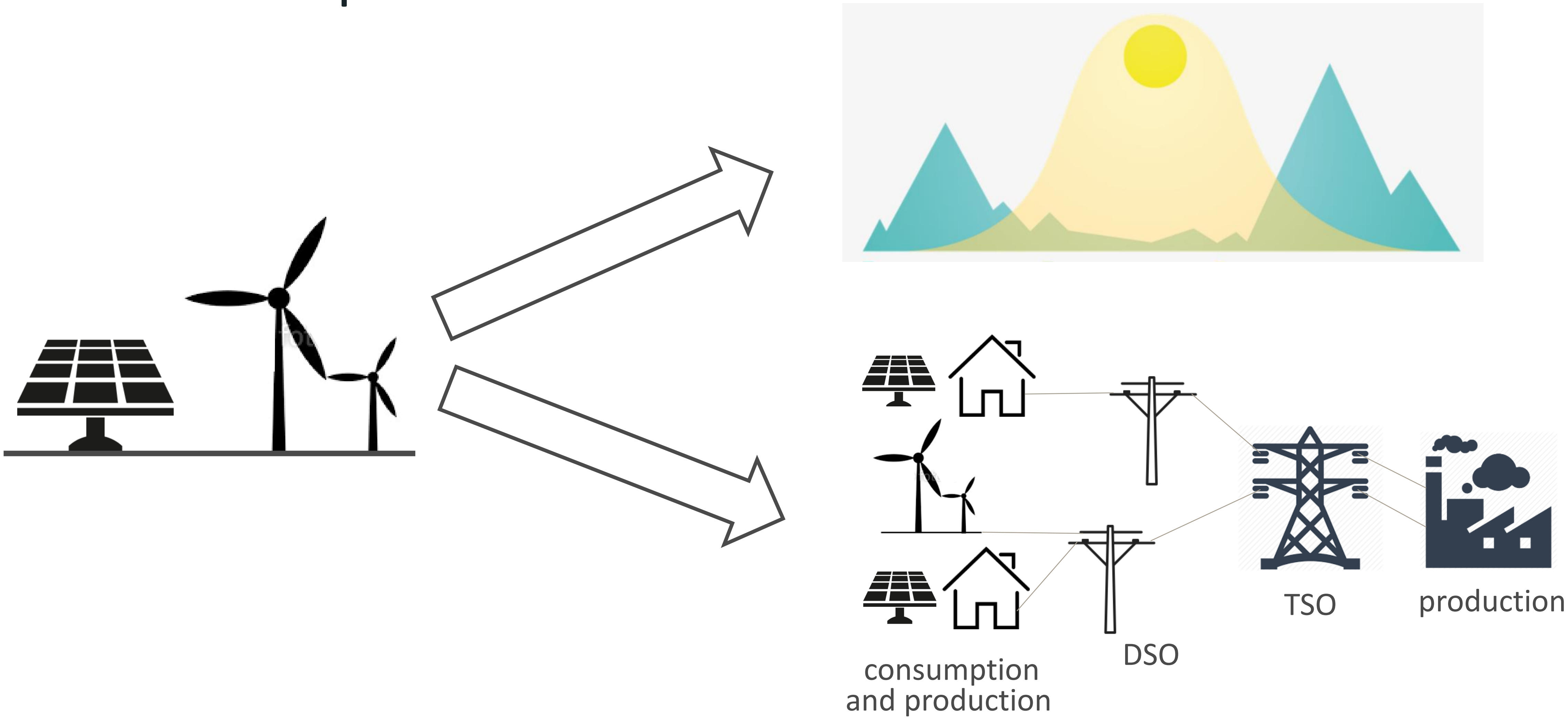
TUM Data Innovation Lab

Constantin Gahr, Ayishetu Haruna and Chris Löchel

Agenda

1) Problem description
2) Data exploration
   - Power data
   - Weather data
3) Models
   - Statistical model
   - SVM model and neural network
   - ARIMA model
4) Web application
5) Conclusion

Problem description

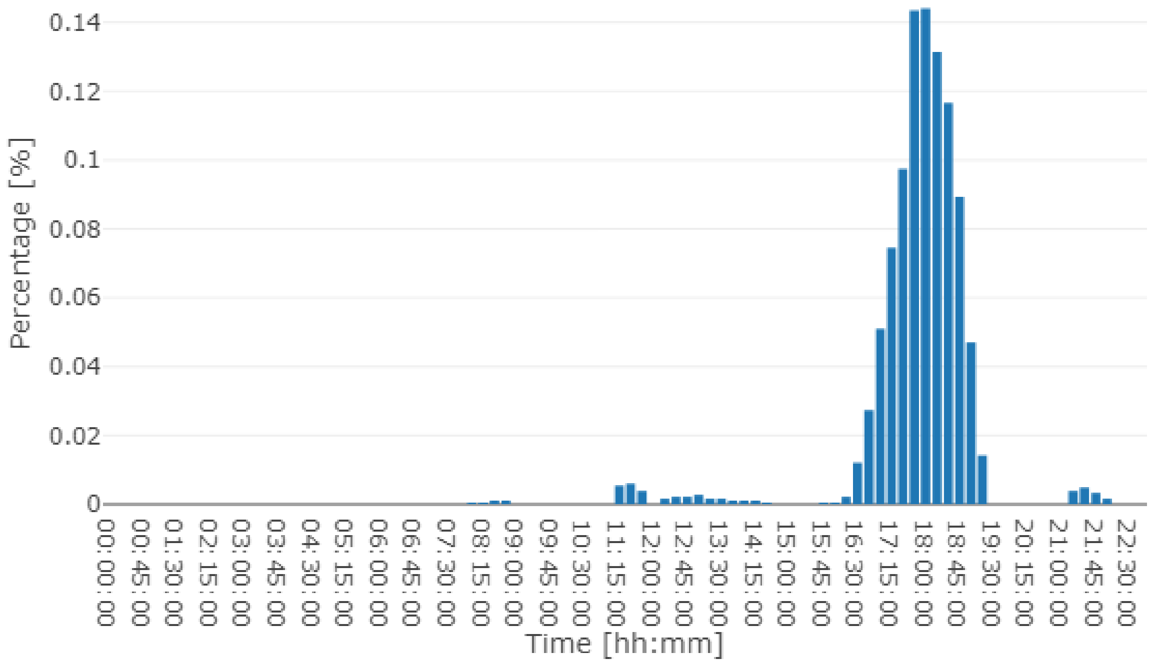


Data collection process

→ DSOs must publish grid load of the past year

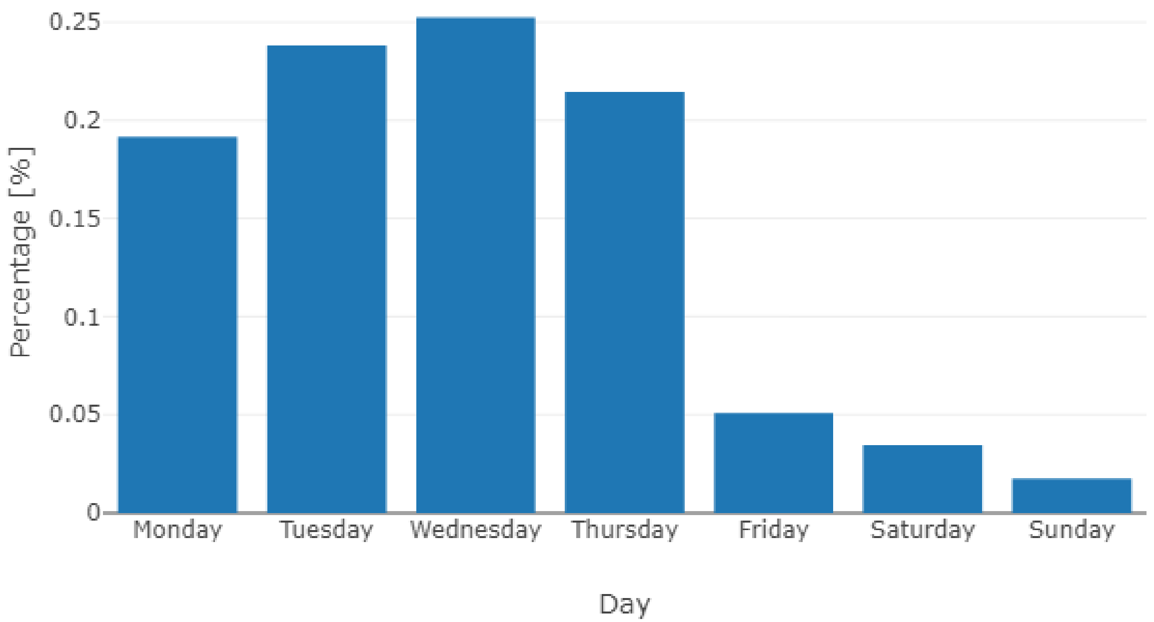
Statistical data exploration

Distribution of power usage within a day, values higher than 0.95 * peak value



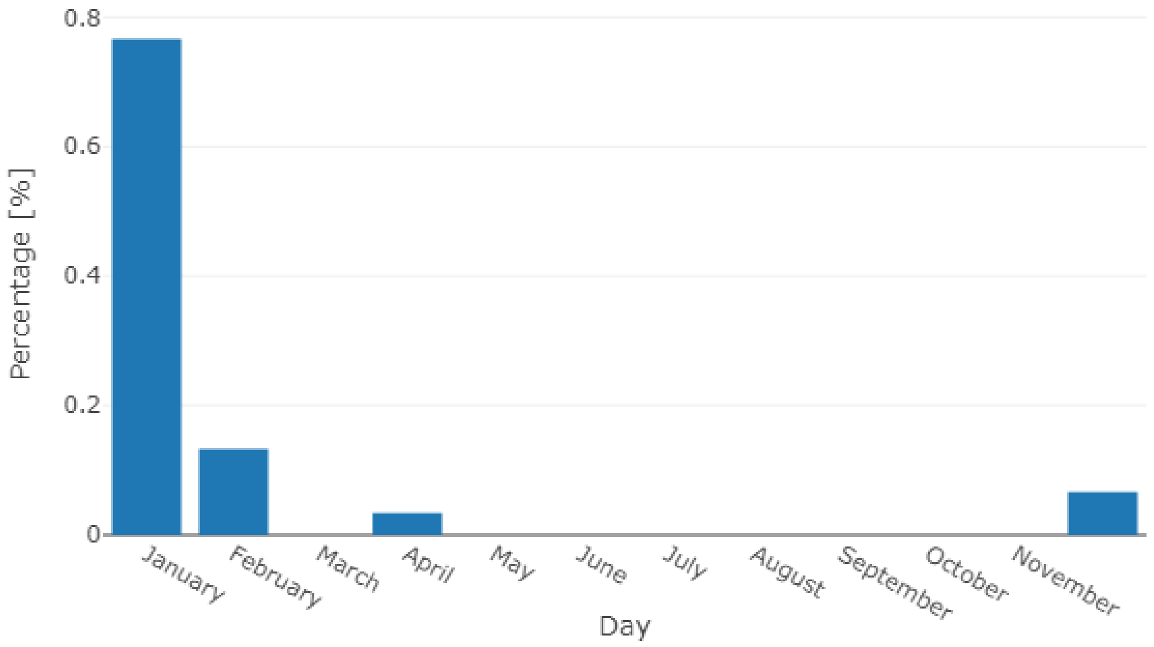
Statistical data exploration

Distribution of power usage within a week, values higher than 0.95 * peak value



Statistical data exploration

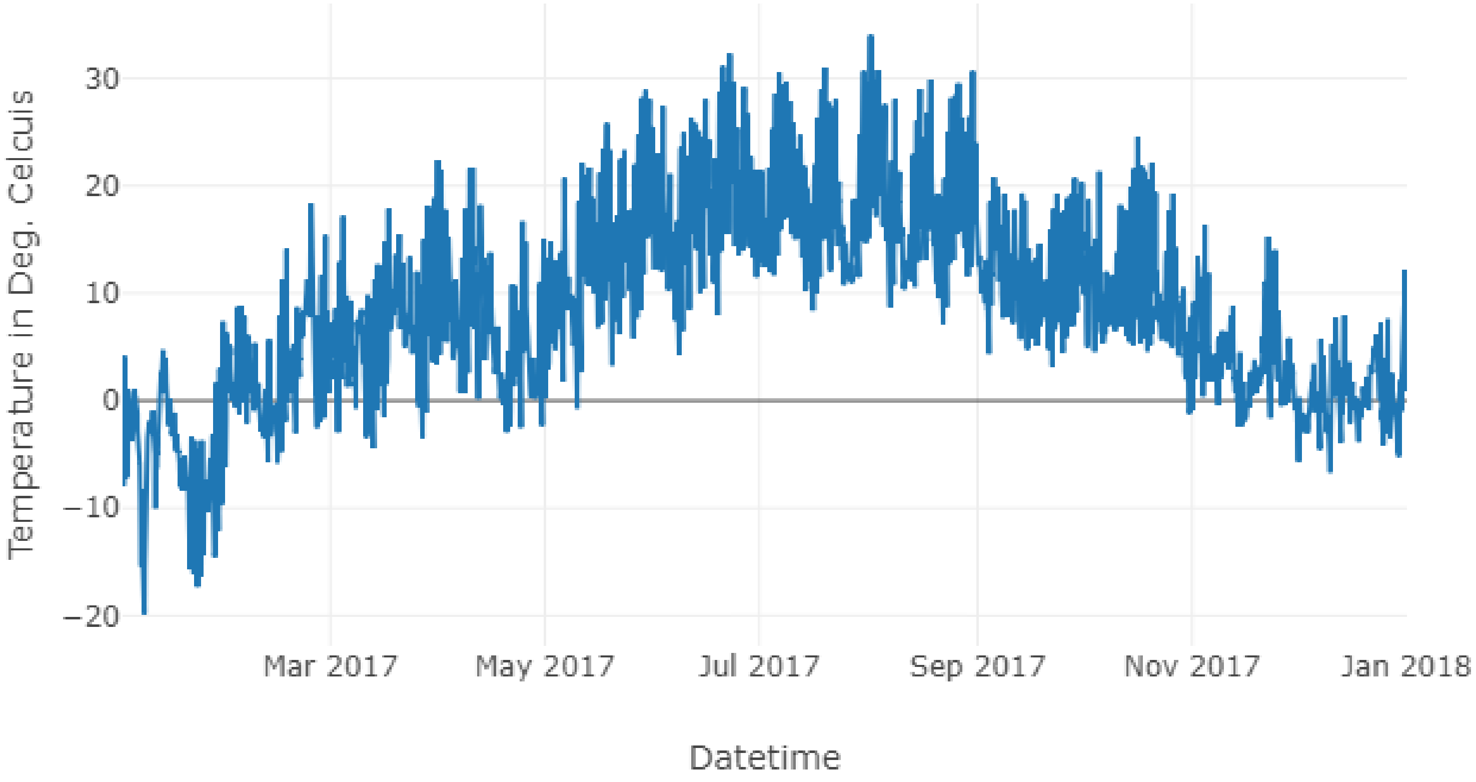
Distribution of power usage within a year, values higher than 0.95 * peak value



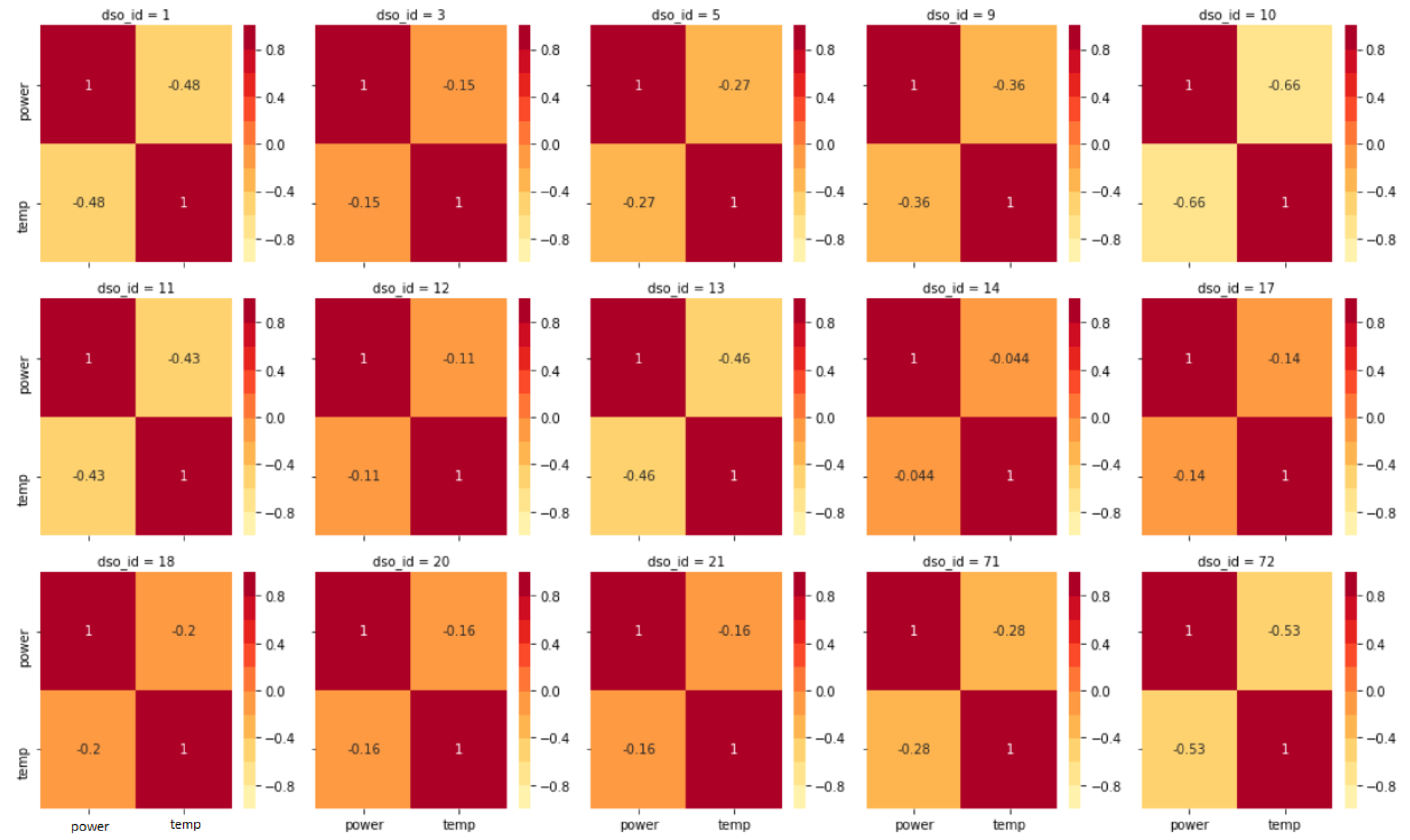
Weather Data

- Climate Data Center (Deutscher Wetterdienst)
- Hourly temperature observations in degree celsius

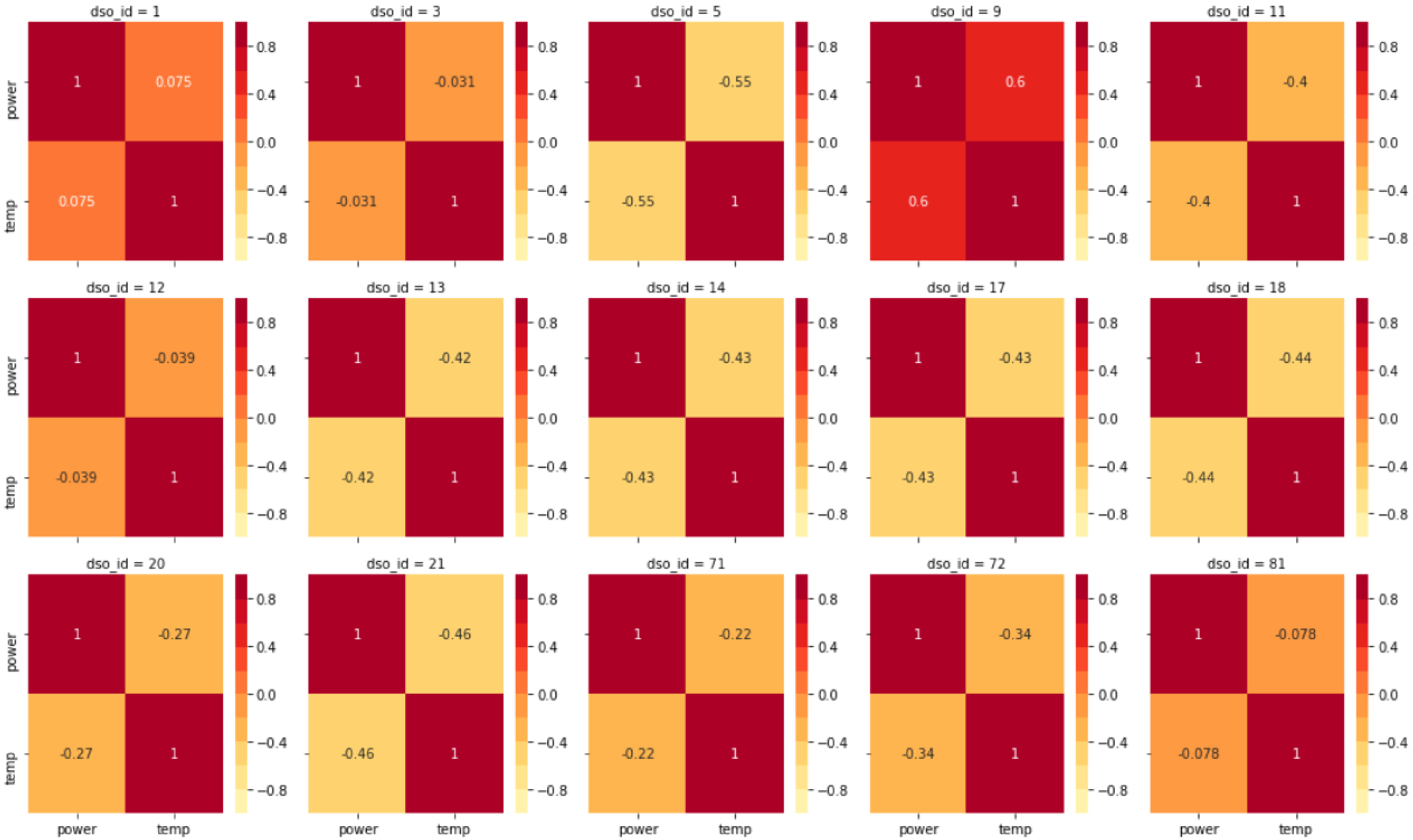
Weather Data



Correlation: Power and Temperature



Correlation: Power (Highest) and Temperature

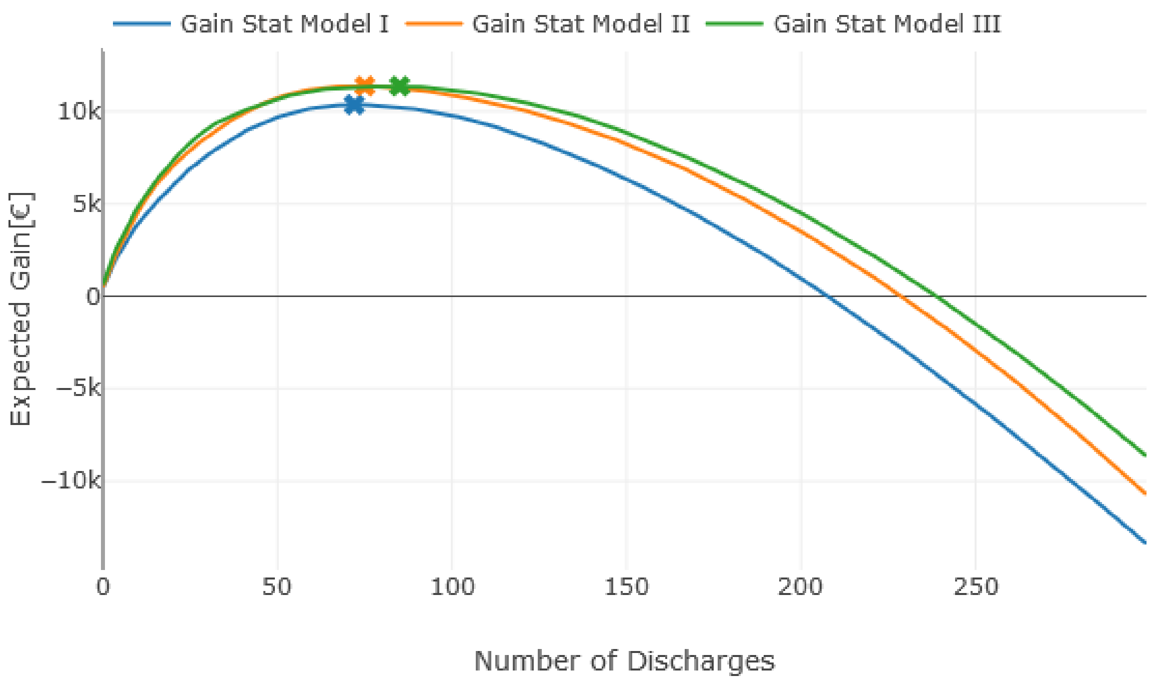


Statistical Model

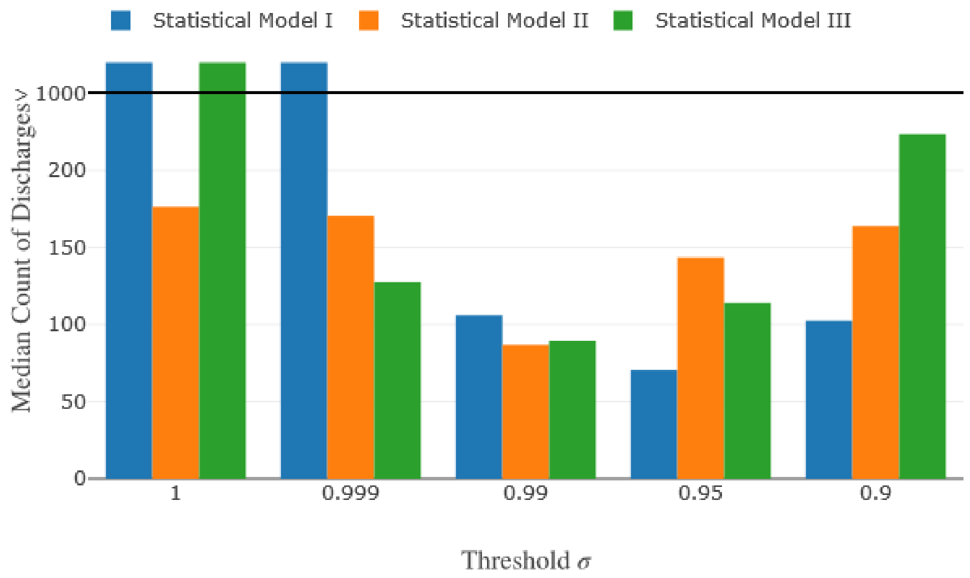
Idea:

$$P(\mathcal{P}(t,d,m) \geq \sigma) = P(\{T_\sigma = t\} \cap \{\tilde{D}_\sigma = d\} \cap \{M_\sigma = m\})$$
$$= \rho_{T,\sigma}(t) \cdot \rho_{\tilde{D},\sigma}(d) \cdot \rho_{M,\sigma}(m)$$
$$\approx \tilde{\rho}_{T,\sigma}(t) \cdot \tilde{\rho}_{\tilde{D},\sigma}(d) \cdot \tilde{\rho}_{M,\sigma}(m).$$

Statistical Model – Theoretical Performance



Statistical Model – Practical Performance



Performance statistical model

Gain: ca. 20000€ per MW

SVM and NN – Data Preparation

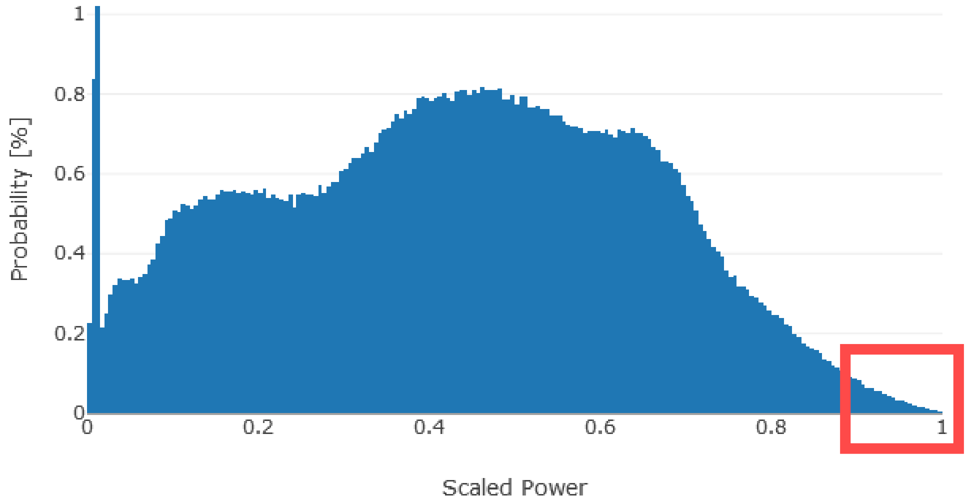
Idea: Create NN and SVM which uses weather data

Data Preparation:

- One Hot Encoding for datetime data
- Temperature data as continuous data

SVM and NN – Data Preparation

But: Dataset is highly unbalanced -> subsampling



SVM and NN – Configuration

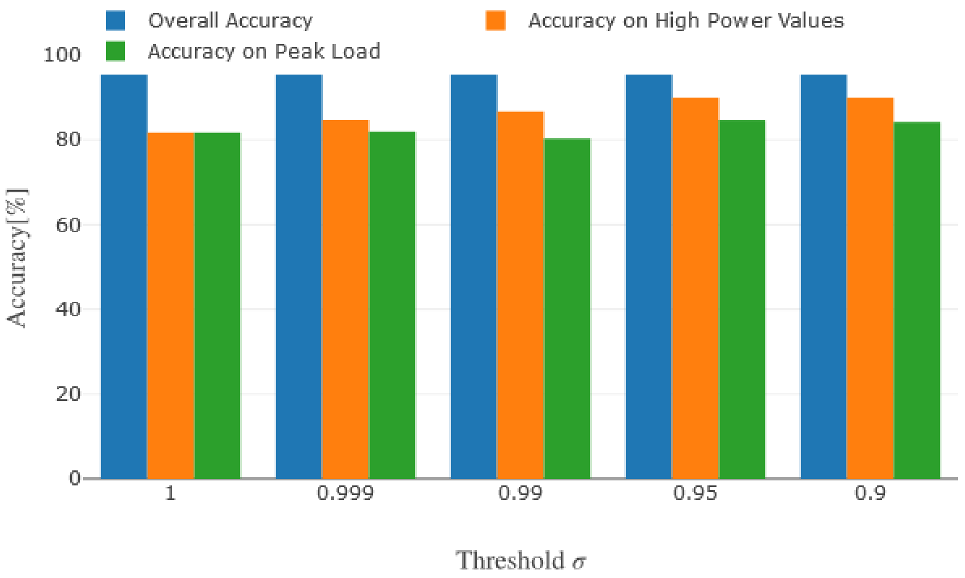
Support Vector Machine:

- Gaussian Kernel

Neural Network:

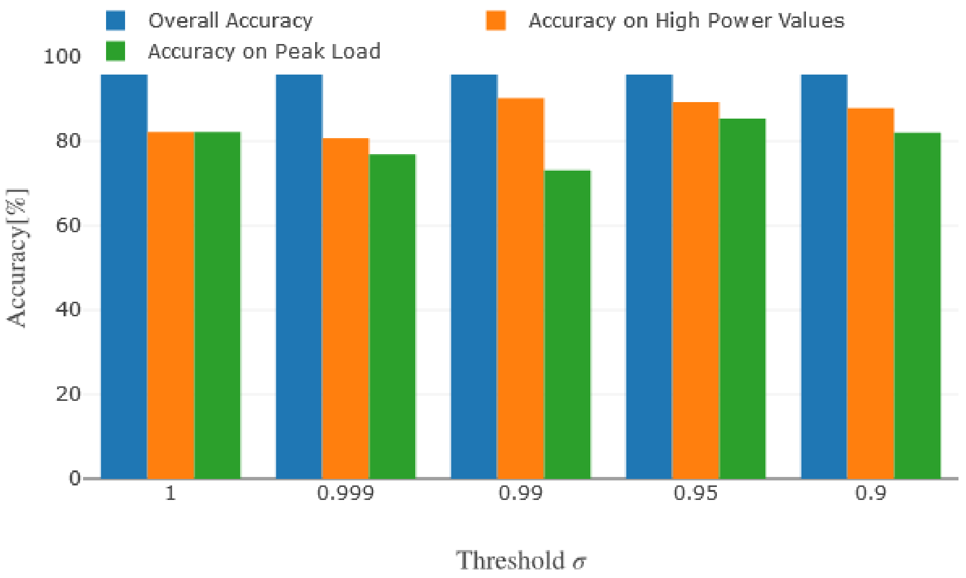
- 3 layers (12 – 8 – 1 neurons)
- 48 inputs (temperature and one hot encoded date)
- Relu activation function $\mathrm{f}(x) = \max(0, x)$ in first two layers
- Sigmoid in last layer

NN and SVM – Performance



Performance support vector machine

Gain: ca. 47000€ per MW



Performance neural network

Gain: ca. 38000€ per MW

ARIMA model

ARIMA model = Autoregressive Integrated Moving Average model

- Needs **stationary** data
- Integrated into Python library statsmodels

→ STEP 1: Make data stationary

→ Only predict difference between current and next value

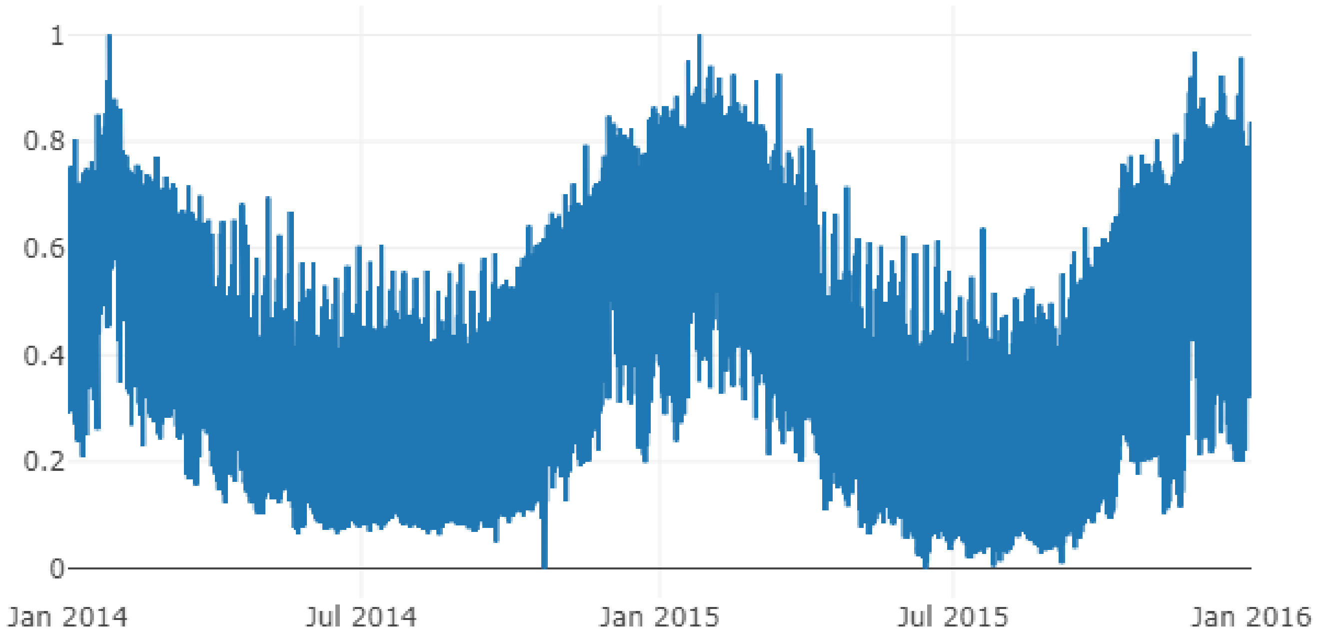
→ *Prediction for next value = average of the past two values + predicted difference*

→ STEP 2: Choose parameters

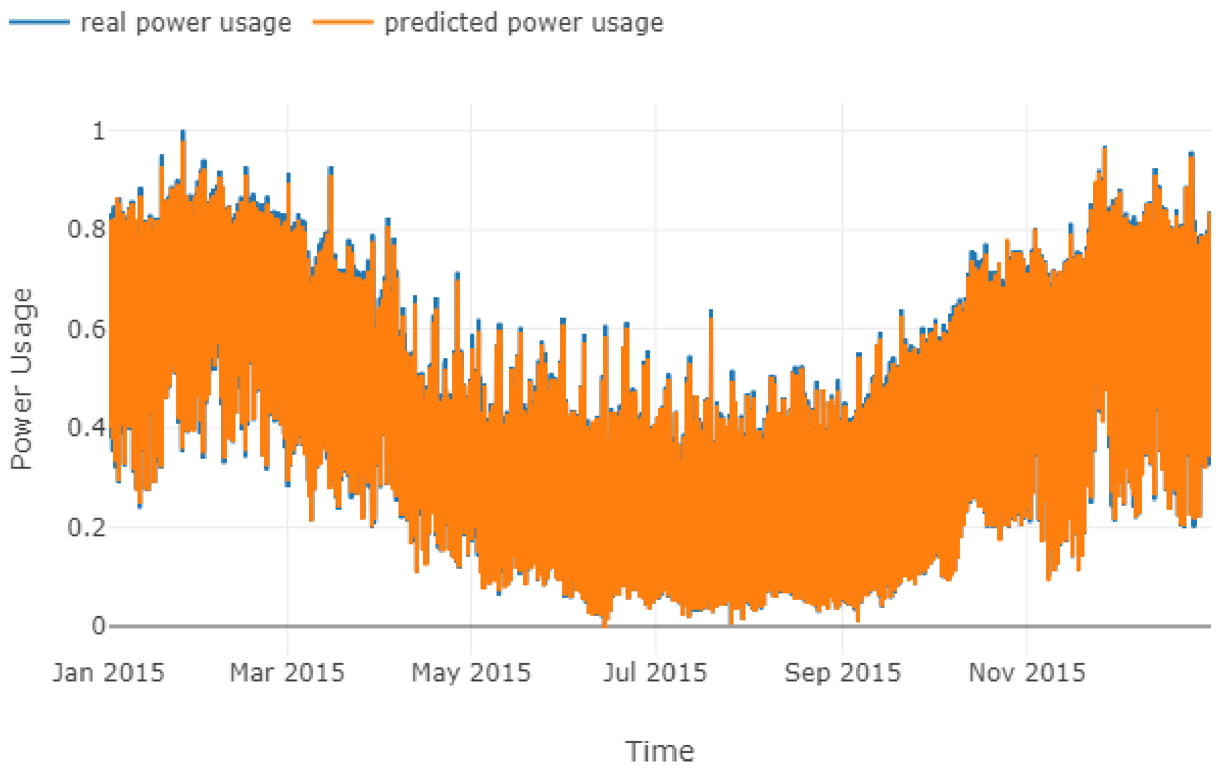
→ STEP 3: Fit ARIMA model and run prediction

ARIMA model

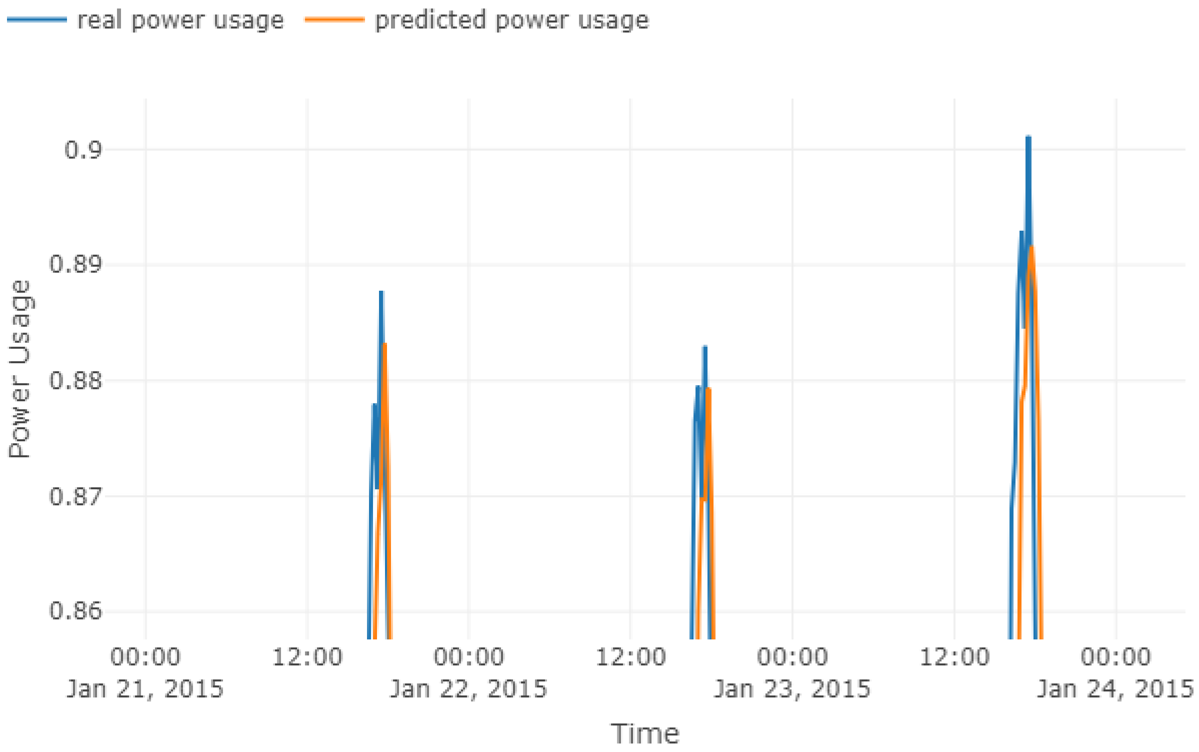
Avacon Netz GmbH, years of 2014 and 2015



ARIMA model prediction



ARIMA model prediction



Choosing peak values of ARIMA prediction

Initialize current_max and choose threshold T

<u>For every time intervall</u>

    if prediction for next interval > current_max * T:

        update current_max

        discharge batteries

Evaluation of ARIMA model

- → Choose different values for T
  - → How often is electricity fed into the grid?
  - → Is the maximum identified?
- → Compute gain/loss

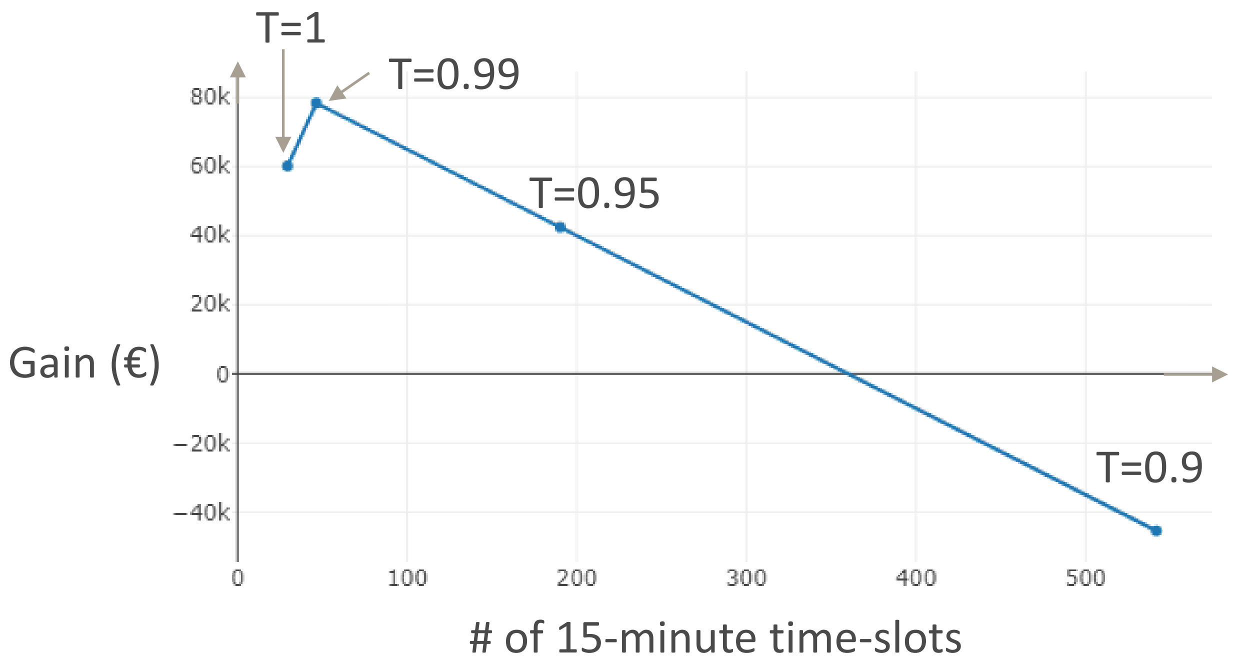
Evaluation of ARIMA model

Performance for four DSOs:

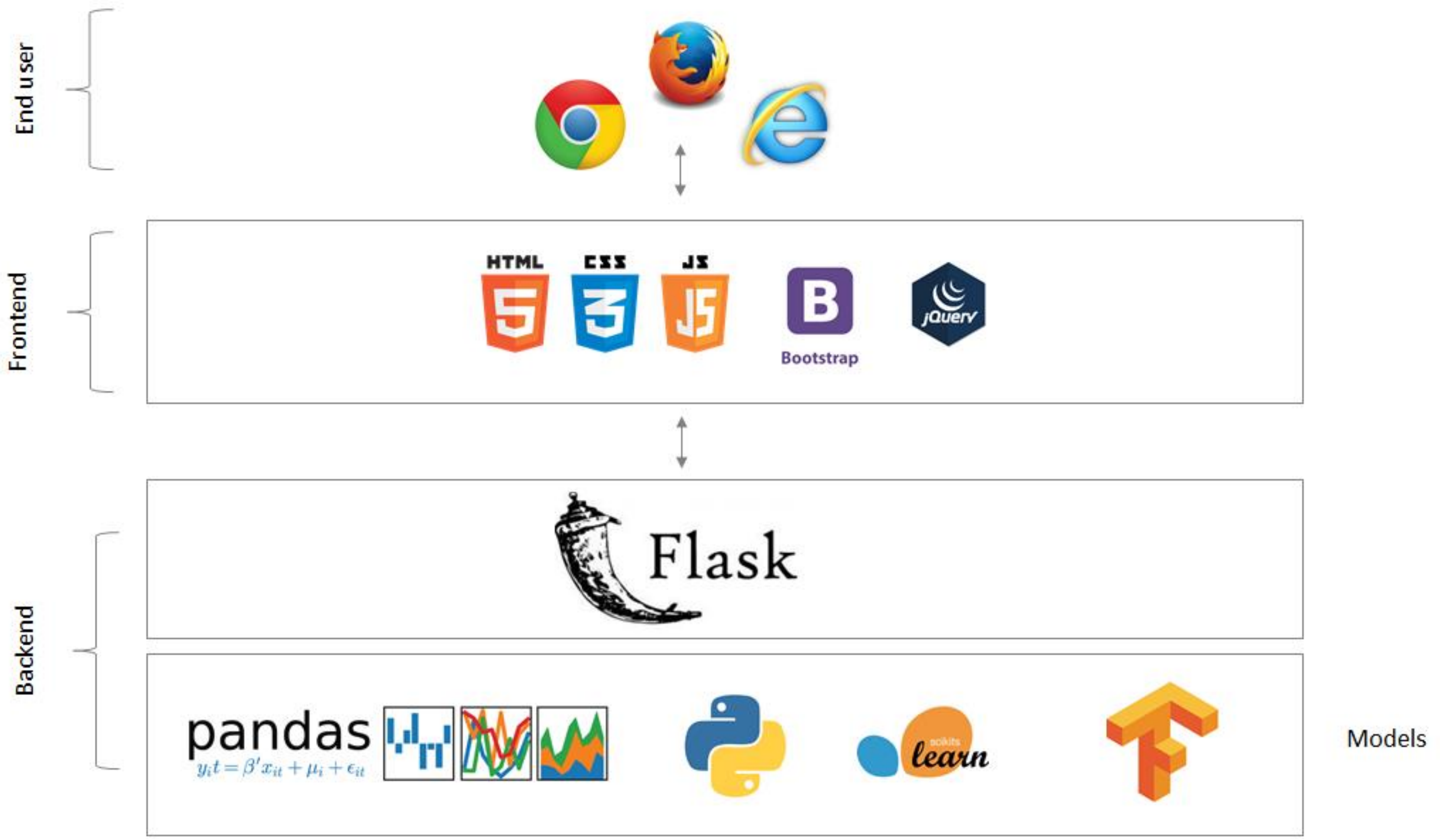
DSO ID		T				
		1	**0.99**	**0.95**	**0.9**	**0.85**
71	# of feed-ins	17	21	167	579	1194
	maximum hit?	1	1	1	1	1
	gain/loss	85750	84750	48250	-54750	-208500
72	# of feed-ins	23	33	76	270	600
	maximum hit?	1	1	1	1	1
	gain/loss	84250	81750	71000	22500	-60000
12	# of feed-ins	38	61	162	404	794
	maximum hit?	1	1	1	1	1
	gain/loss	80500	74750	49500	-11000	-108500
14	# of feed-ins	40	71	356	914	1945
	maximum hit?	0	1	1	1	1
	gain/loss	-10000	72250	1000	-138500	-396250
Average	**# of feed-ins**	**29,5**	**46,5**	**190,25**	**541,75**	**1133,25**
	maximum hit?	**0,75**	**1**	**1**	**1**	**1**
	gain/loss	**60125**	**78375**	**42437,5**	**-45437,5**	**-193312,5**

Evaluation of ARIMA model

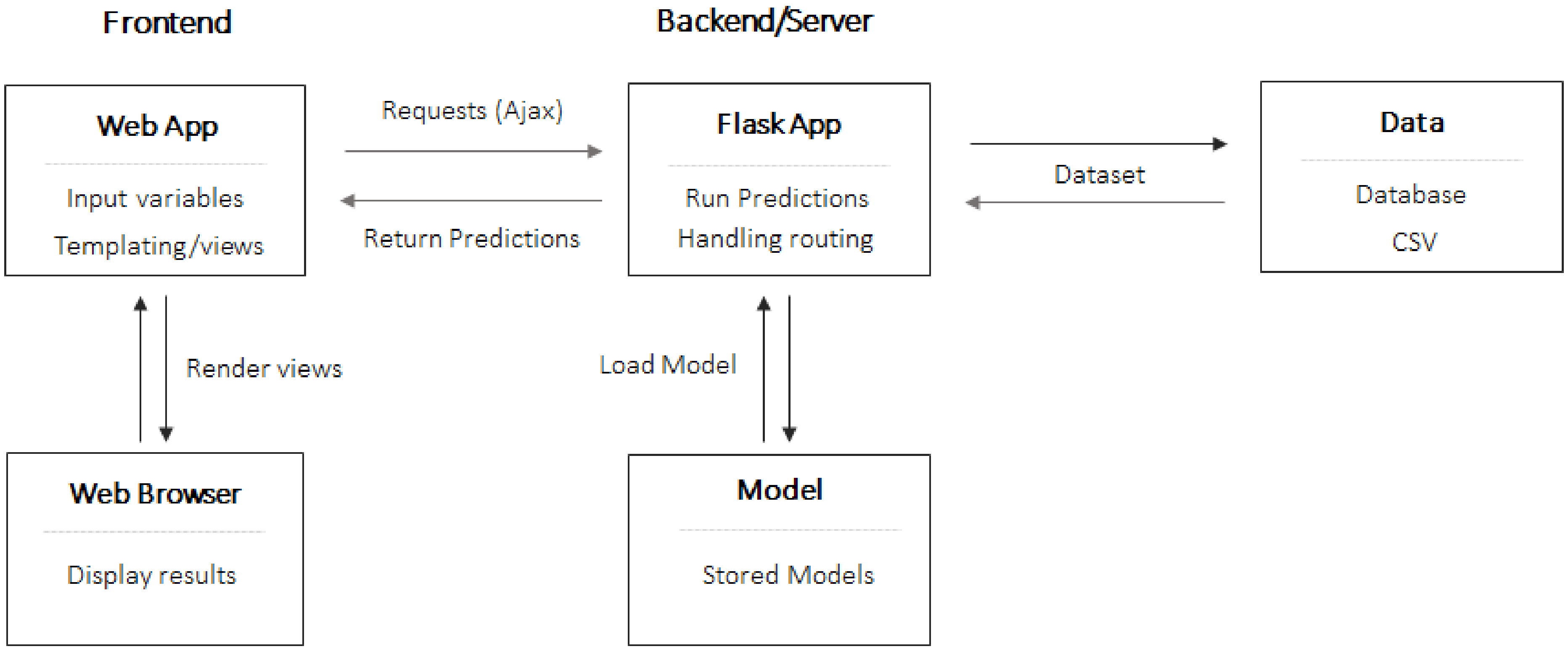
Performance curve for four DSOs:



Web Application: Architecture



Web Application: Application Process flow



Conclusion

- Statistical Model:
  - Gain: ca 20000€ per MW
  - Baseline performance
  - Ready for production
- NN and SVM:
  - Gain: ca 40000€+ per MW
  - Room for improvement (optimize hyperparameters, use live data, …)
- ARIMA Model:
  - Gain: ca 78000€ per MW (to be taken with a grain of salt!)
  - Needs live data
- Web Application:
  - Demonstration of our models

Thank you for your attention!