AI Modelling and Time-series Forecasting Systems for Trading Energy Flexibility in Electrical Distribution Grids

—

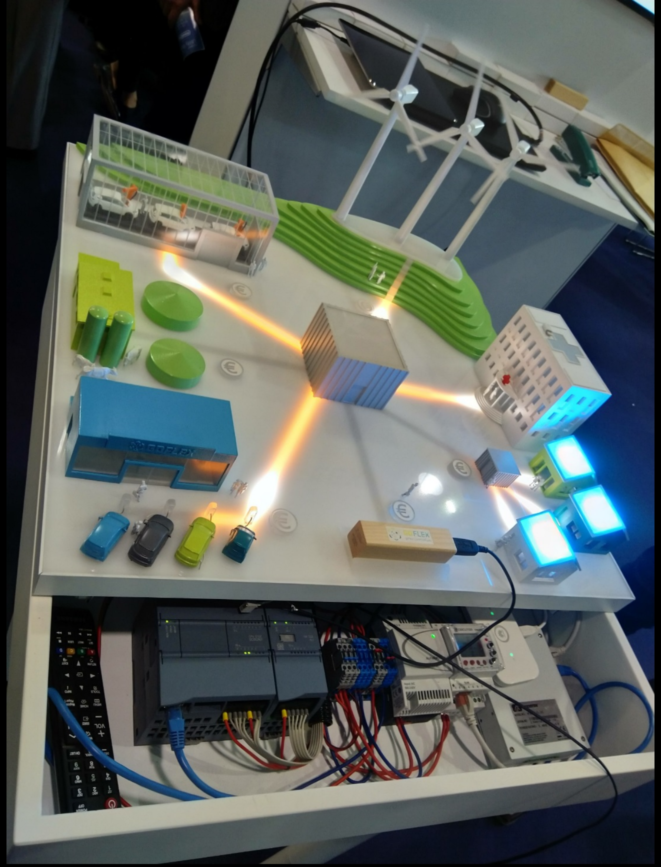
Francesco Fusco, PhD
IBM Research Europe, Ireland

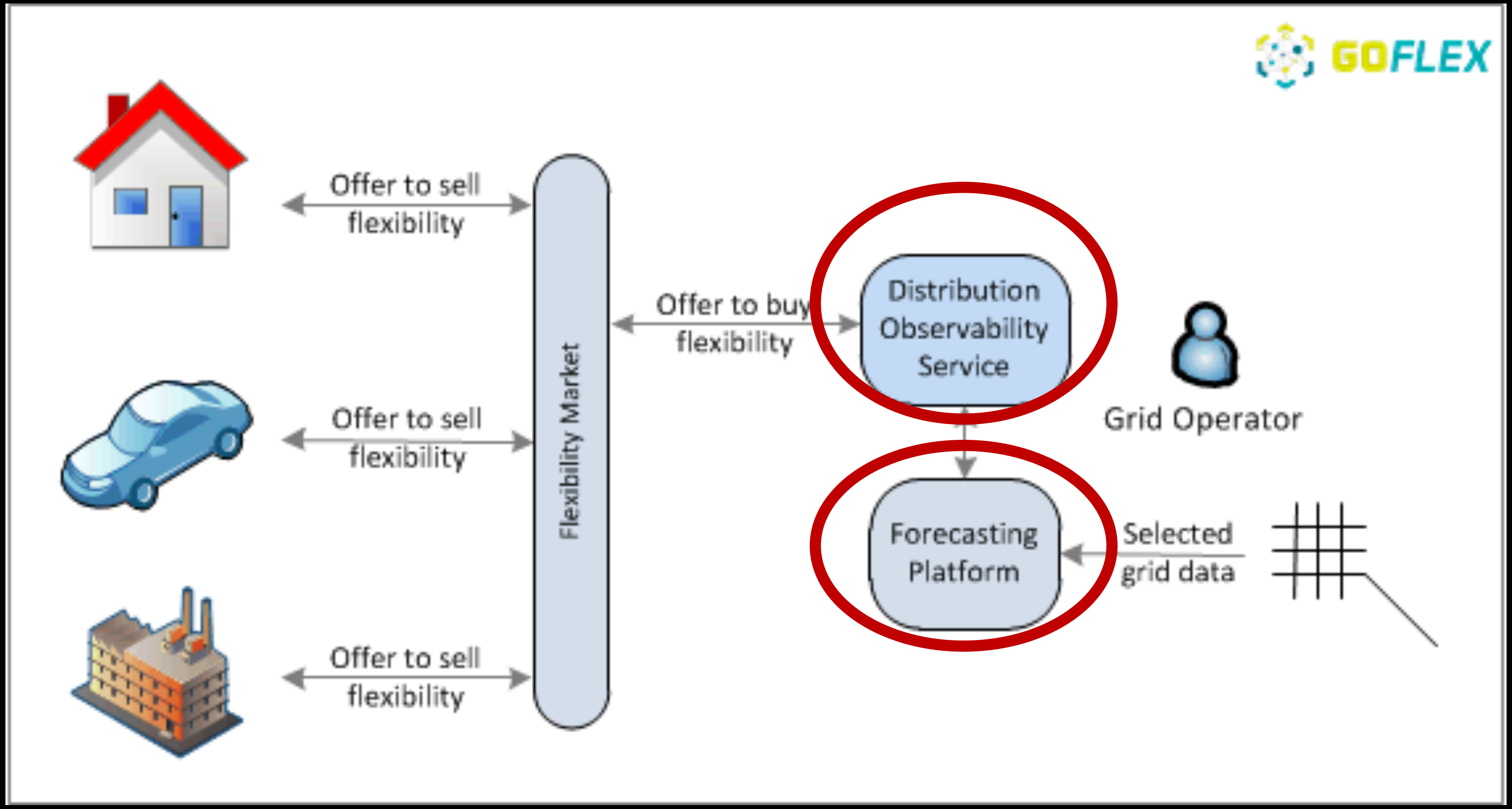
International Joint Conference on Green Digitalization
November 10th 2020

The GoFlex project

- EU-funded H2020 research project
- Develop solutions to enable an energy flexibility market
  - Distribution grid operator (DSO) buys flexibility to solve for local grid congestions
  - Accommodate larger shares of distributed (renewable) energy generation
- 3 demonstration sites at DSOs in Cyprus, Switzerland, Germany
- <u>IBM developed</u>
  - Scalable energy forecasting platform
  - AI-based grid modelling for DSO bidding decisions

https://www.goflex-project.eu/

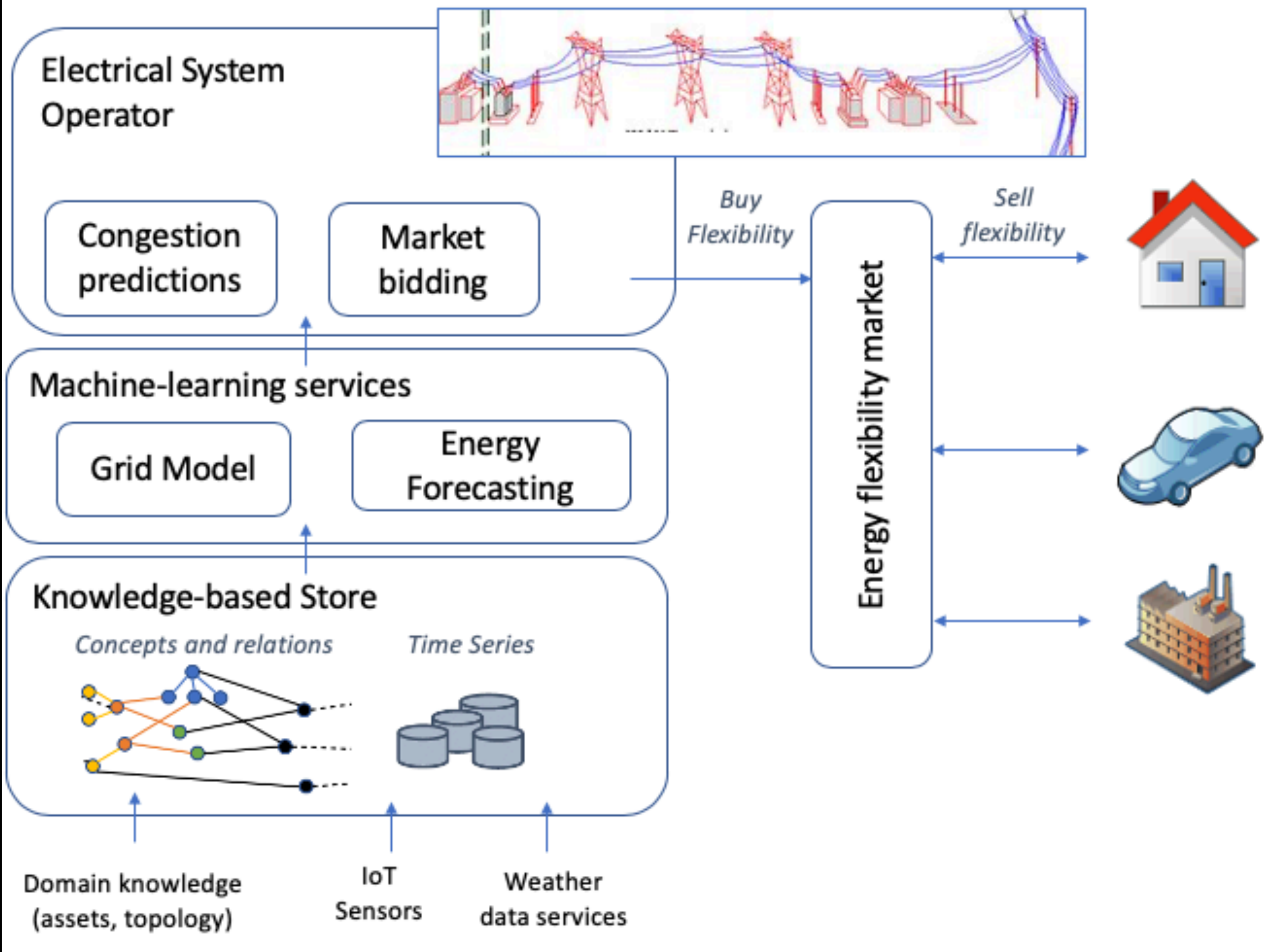




The GoFlex project

GOFLEX
goflex-community.eu

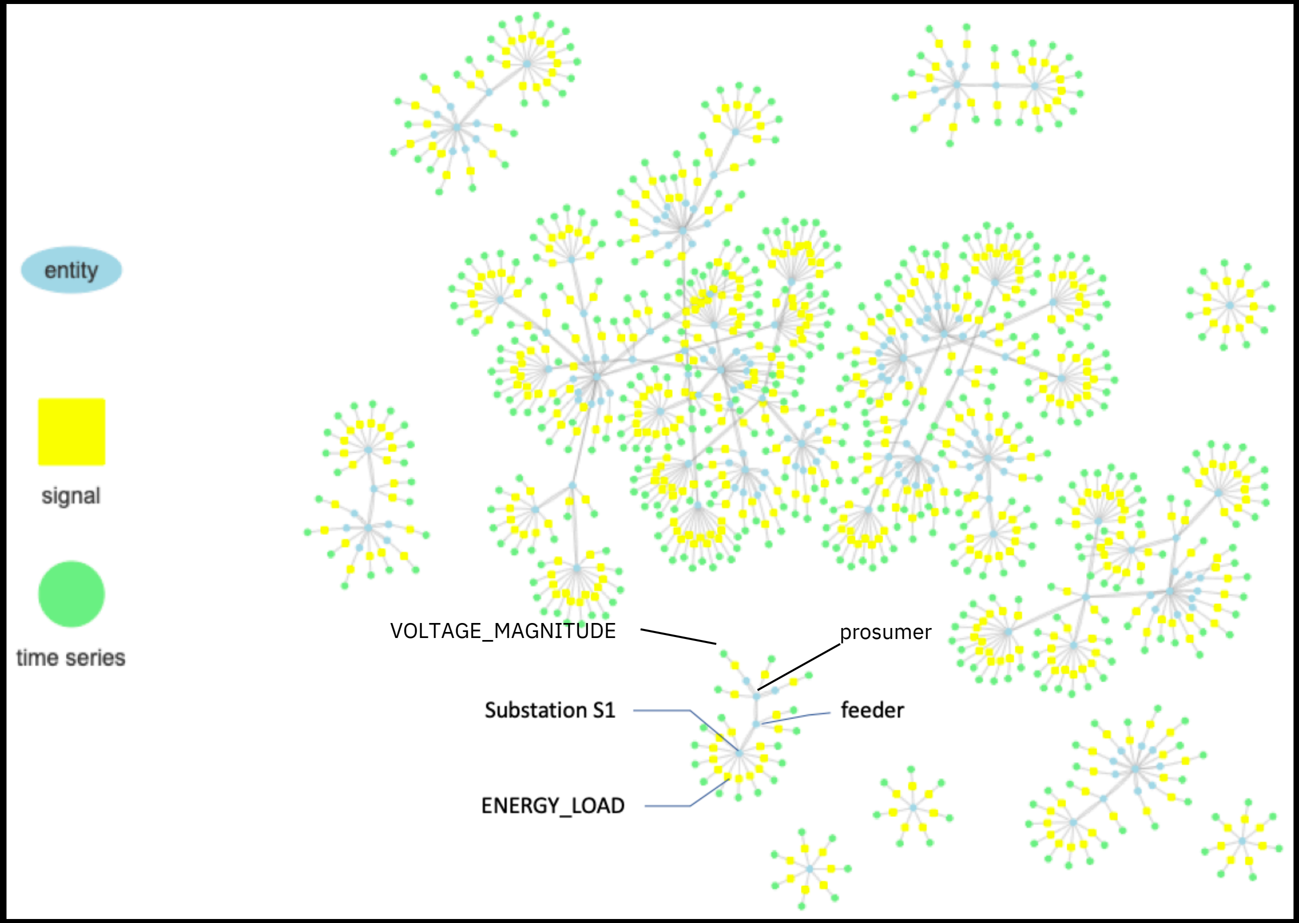
- Scalable energy forecasting platform
  - Collect IoT sensor data
  - Run time-series forecasting models of distributed consumption and generation
- AI-based grid modelling
  1. Predict grid issues
     - Input is energy forecasts
     - Estimate impact on the grid (load, voltages)
     - Compare with user-defined tolerance
  2. Predict required flexibility
     - Input is desired profile (load, voltages)
     - Estimate amount of energy flexibility (increase/decrease) to follow profile

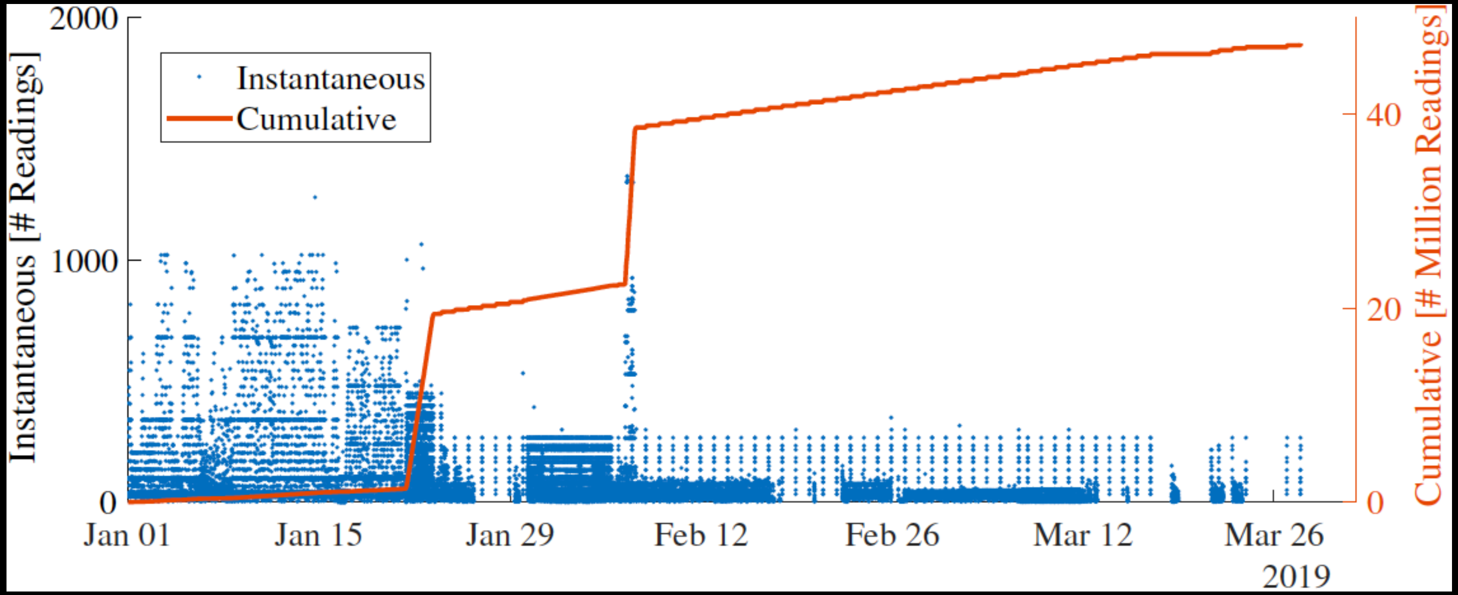


Leverage IoT data semantics:
- ***Scaling and Automating** time-series forecasting models*
- *Incorporate **domain knowledge** in AI Modelling*

Scalable time-series forecasting platform

- GoFlex use-case
  - Collect IoT data from prosumers, SCADA (1000+ points every 15 min)
  - https://github.com/GoFlexH2020/samples
  - Live energy forecasts
    - Distributed energy consumption and generation (solar, wind) at 200+ points
    - Refresh hourly for 24-hour windows at 15-minute time resolution
  - Continuous trial operation from Jan 2019 to March 2020





IBM Research Castor:
Scalable time-series forecasting platform

- IoT and semantic data management, deployment of AI time-series models on the Cloud
- Built-in parallelization and potentially infinite horizontal scalability
- Programmatic deployment of AI models based on semantics for model reuse and automation
- Full lineage of timeseries data, model forecasts, model (re-trained) versions.
- Transparent, customizable modelling ecosystem, supporting Python / R

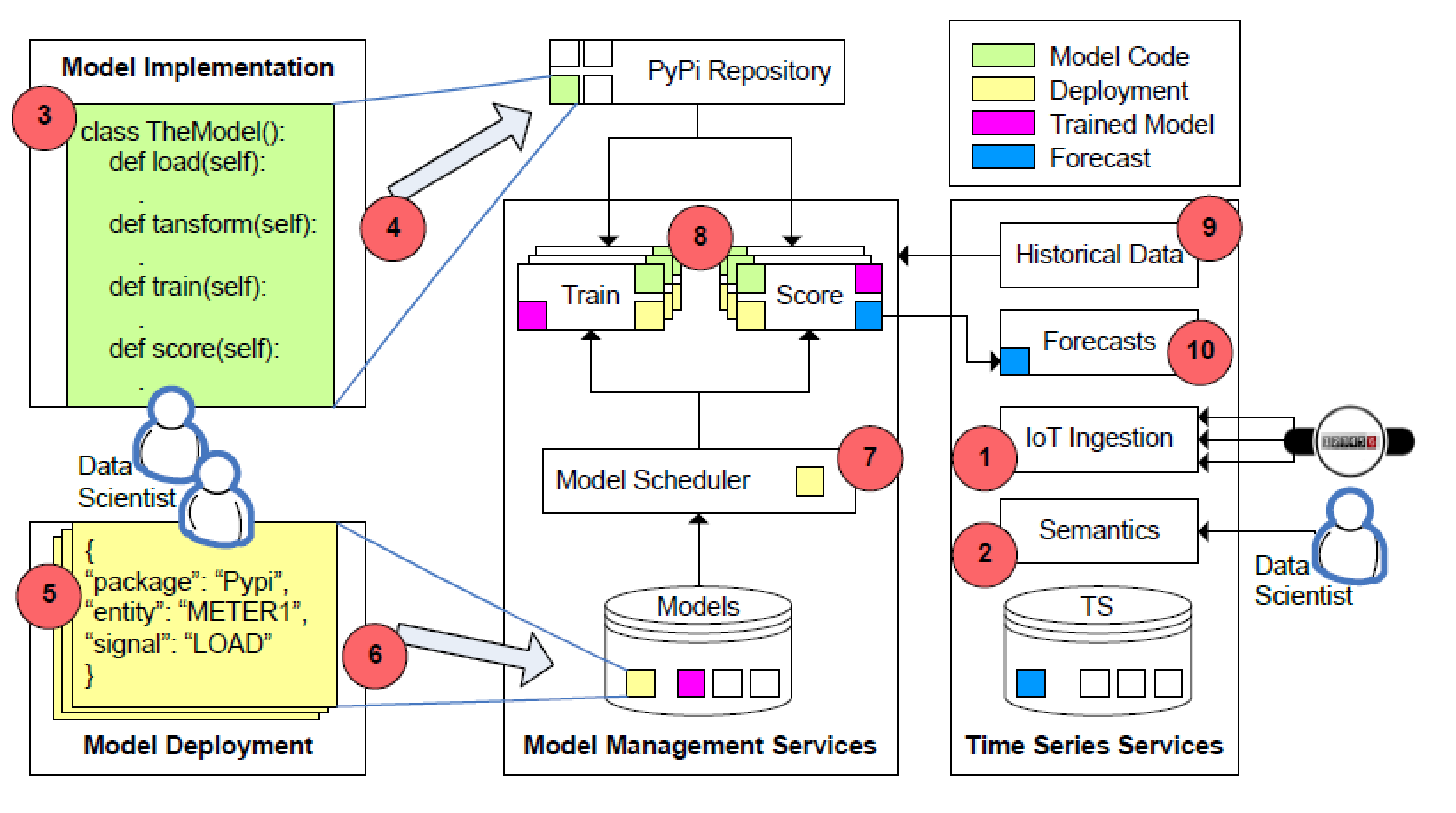


Figure 1: IBM Research Castor Overview and Modelling Flow

Built on the IBM Cloud

1 Live time-series data ingestion

2 Application semantics

3 4 Develop / customize AI models

5 6 Deploy AI models

7 8 Automatic execution of model train/score jobs

9 10 Automatic persistence trained model versions, time-series predictions

IBM Research Castor:
Scalable time-series forecasting platform

- Separation between implementation and model deployment
  - Implement modelling steps (feature engineering, model selection, etc.) based on abstract *semantic context* (“solar generation”, “substation load”, ...)
  - Specify semantic context instance in deployment configuration

- Implement one model, deploy many (automate)

Listing 1 Model Implementation Pseudocode

```
class MyModel (ModelInterface):
  def __init__ (self, context, task,
                modelId, modelVersion,
                user_params)
  def load():
    x = getTimeseries(context.entity,
                      context.signal,
                      start, end)
    w = getWeather(context.entity.lat,
                   context.entity.long,
                   start, end)
  def transform():
    x = merge(x, w)
    x = align_data (x, user_params.frequency)
    generate_lagged_features(x)
    // other features
  def train():
    start, end = user_params.train_period
    self.load(), self.transform()
    return( LinearRegression.fit(x))
  def score():
    start, end = now(),now()+dt(hours=24)
    self.load()
    model = get
    return( moc
```

Listing 2 Example of Model Deployment Configuration

```
{
 ”context”: {”entity”: <value>, ”signal”:<value>},
 ”model_name”: <value>,
 ”dist_name”: <myModelCodePackage>,
 ”dist_ver”: ”1.0.0”,
 ”module”: <myModelCodeRoutines>,
 ”training_deployment”: {
     ”time”: ”2019-03-01T00:00:00+00:00”,
     ”repeatEvery”: ”1_week” },
 ”scoring_deployment”: {
     ”time”: ”2019-03-01T00:00:00+00:00”,
     ”repeatEvery”: ”1_hours” },
 ”user_parameters”: {
     ”frequency”: ”15T,
     ”train_time”: {”2018-01-01”, ”2019-01-01”} },
}
```

IBM Research Castor:
Scalable time-series forecasting platform

- The GoFlex use
  - Approx. 250 models running live every 15 min

Site	# Sensors	# Models	Execution [s]
Germany	18	11	16.8
Switzerland	196	61	19.7
Cyprus	531	174	15.9

Table 2: Size and performance in deployed systems. Model execution refers to the average duration of a scoring job.

- Scalability tests
  - Up to 27K jobs / hour at no performance degradation

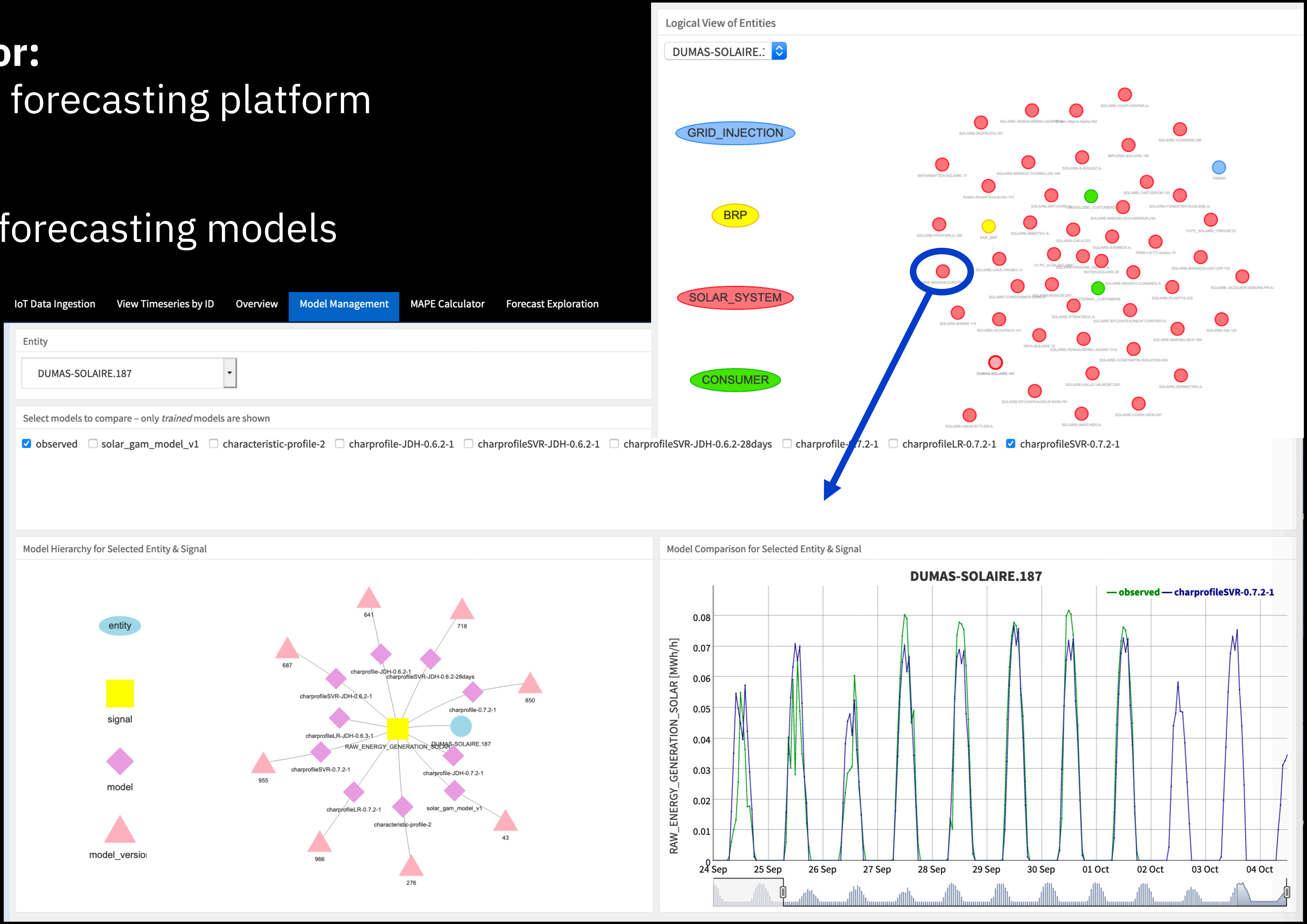
Parallel Jobs	# Jobs/hour	Job Duration [s]
10	5,600	6.4
50	18,900	9.5
100	22,300	16.1
150	26,900	20.1
175	27,600	22.8
200	26,700	27.0

Table 3: System scalability analysis.

Eck, B. and Fusco, F. and Gormally, G. and Purcell, M. and Tirupathi, S. *“Scalable Deployment of AI Time-series Models for IoT”*. AI4IoT Workshop at IJCAI 2019

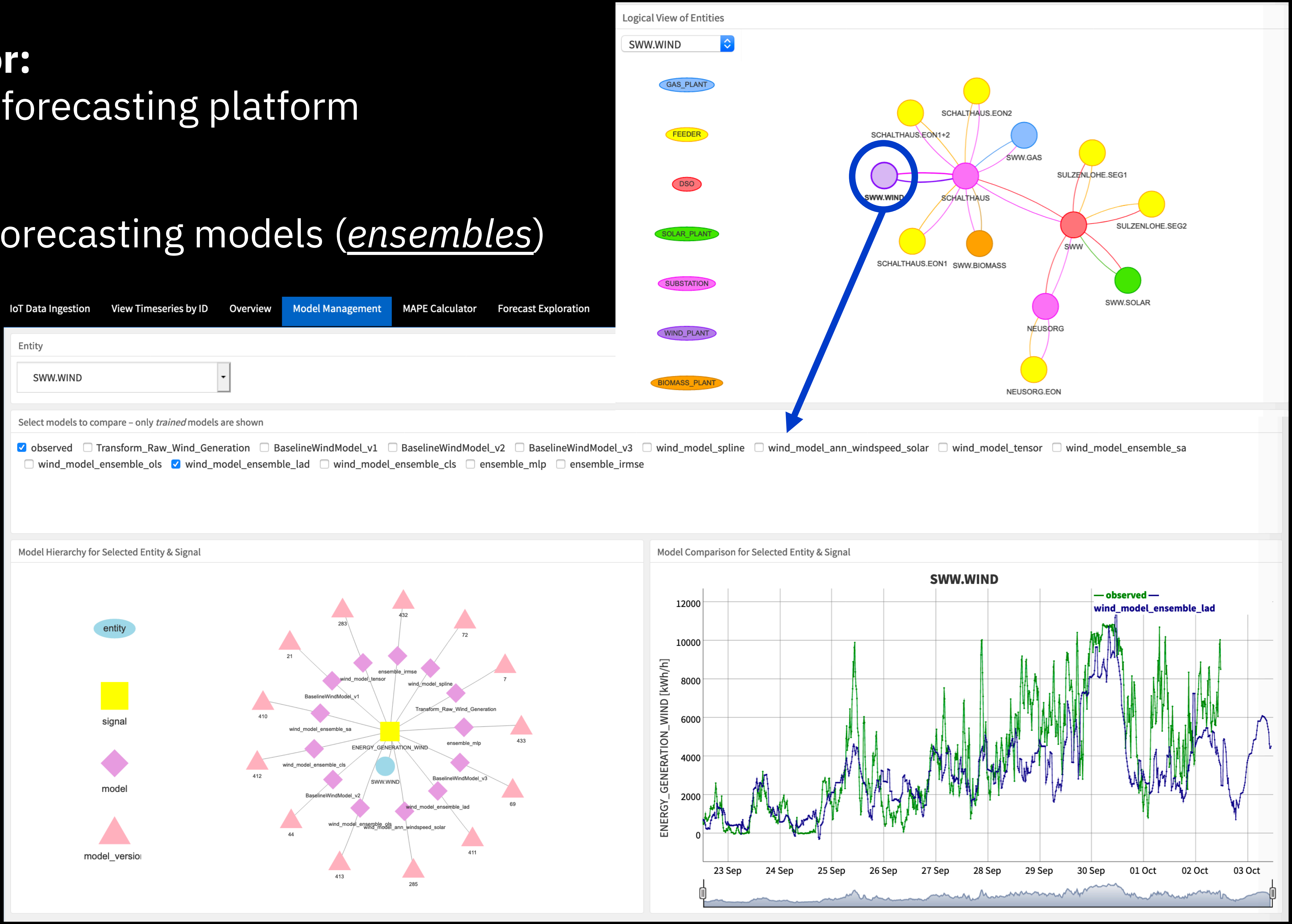
IBM Research Castor:
Scalable time-series forecasting platform

- Example of solar forecasting models



IBM Research Castor:
Scalable time-series forecasting platform

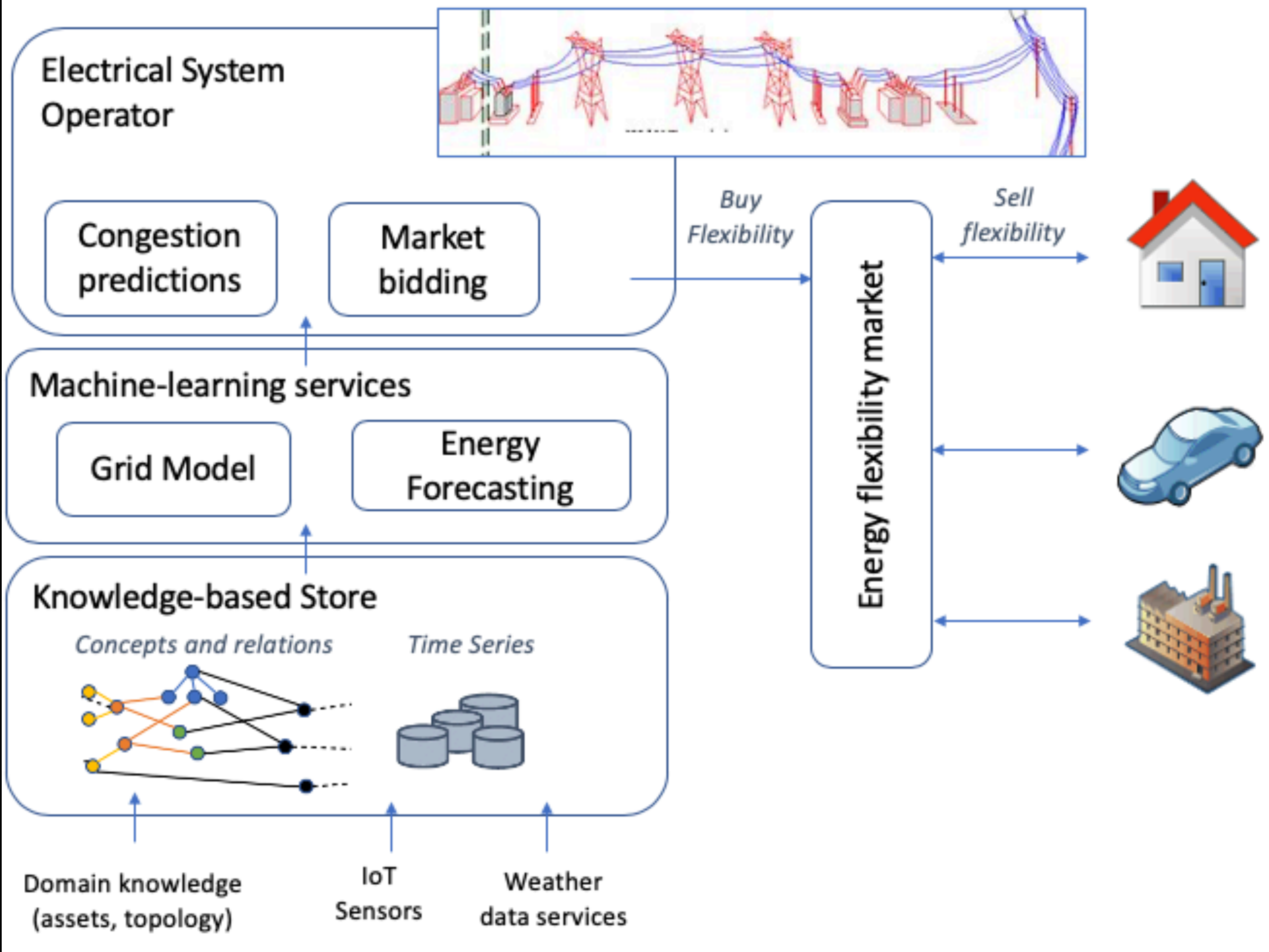
- Example of wind forecasting models (*ensembles*)



The GoFlex project

goflex-community.eu

- <u>Scalable energy forecasting platform</u>
  - Collect IoT sensor data
  - Run time-series forecasting models of distributed consumption and generation
- <u>AI-based grid modelling</u>
  1. Predict grid issues
     - Input is energy forecasts
     - Estimate impact on the grid (load, voltages)
     - Compare with user-defined tolerance
  2. Predict required flexibility
     - Input is desired profile (load, voltages)
     - Estimate amount of energy flexibility (increase/decrease) to follow profile



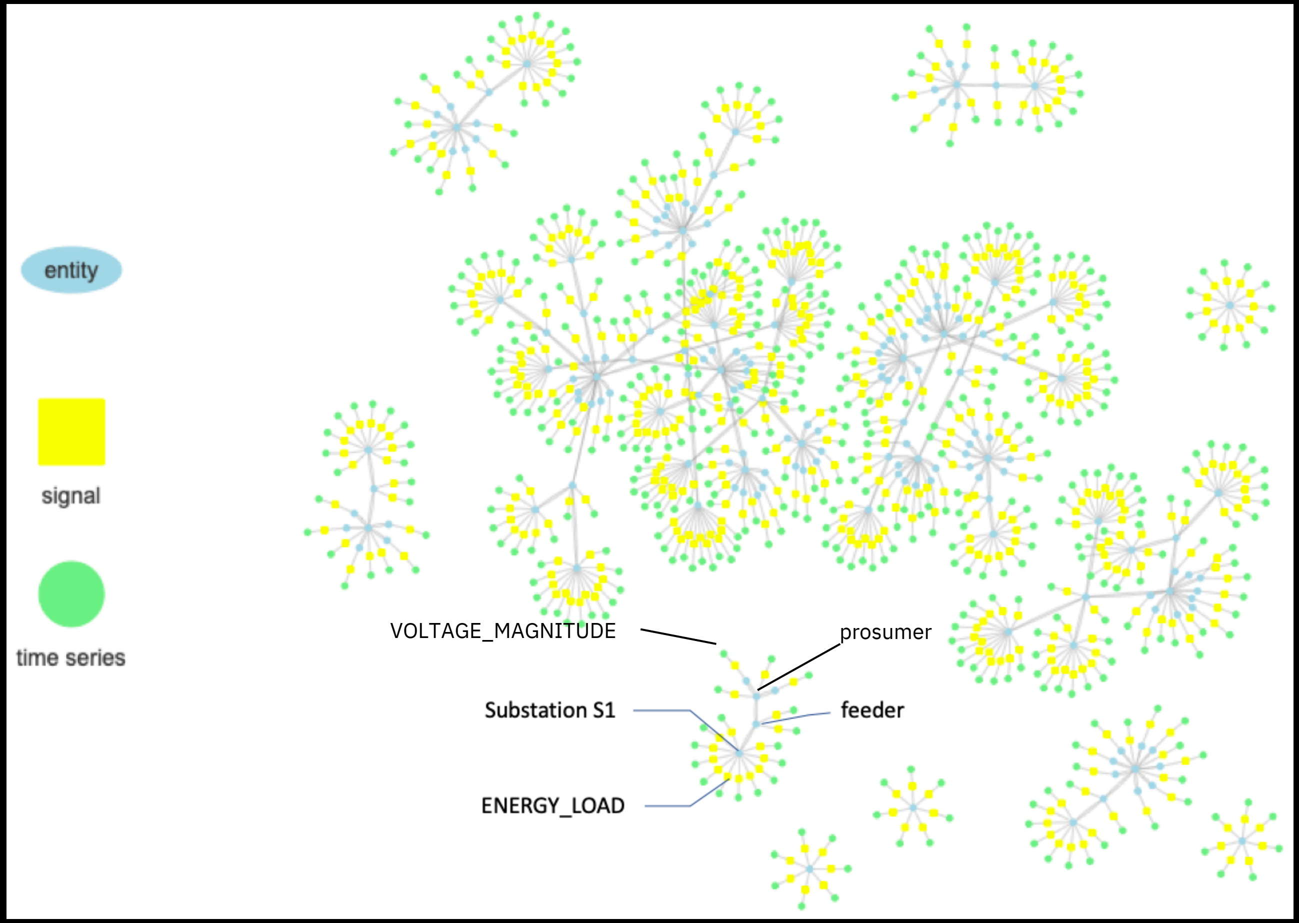
Leverage IoT data semantics:
- ***Scaling and Automating** time-series forecasting models*
- *Incorporate **domain knowledge** in AI Modelling*

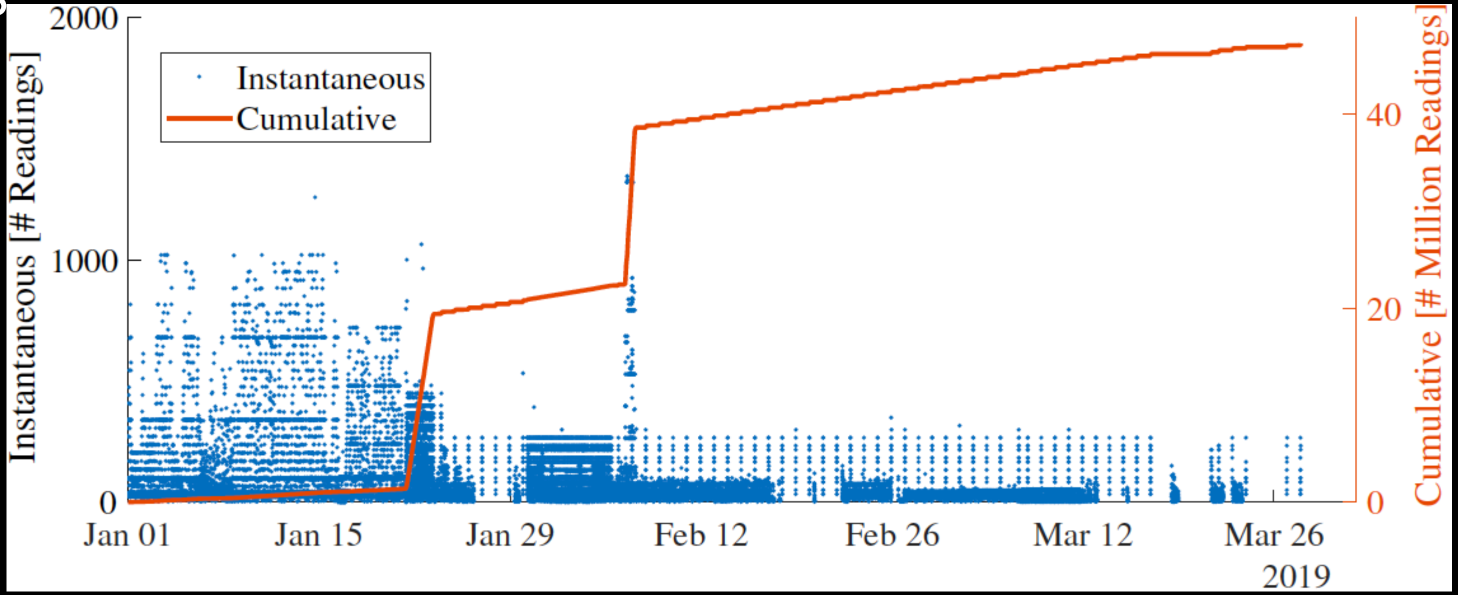
AI-based grid modelling

- GoFlex use-case
  - Predict impact of distributed energy generation on grid assets (e.g. voltage, line loading)
  - Estimate required flexibility for desired asset behavior (e.g. voltage profile)

- Limitations of power-flow models
  - Require detailed, accurate physical parameters
  - Difficult to maintain under changing grid

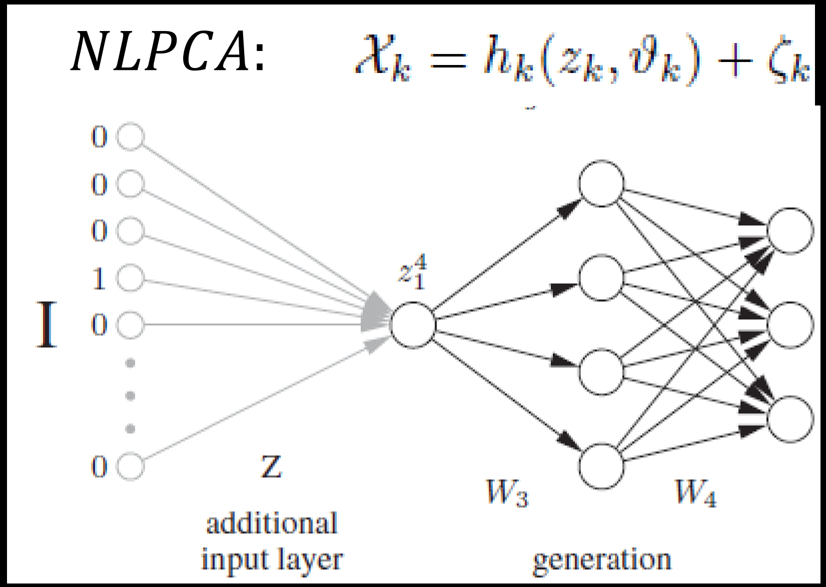
- <u>AI-based modelling</u>
  - Leverage sensor data
  - Don’t ignore available knowledge, however incomplete (connectivity, physics)



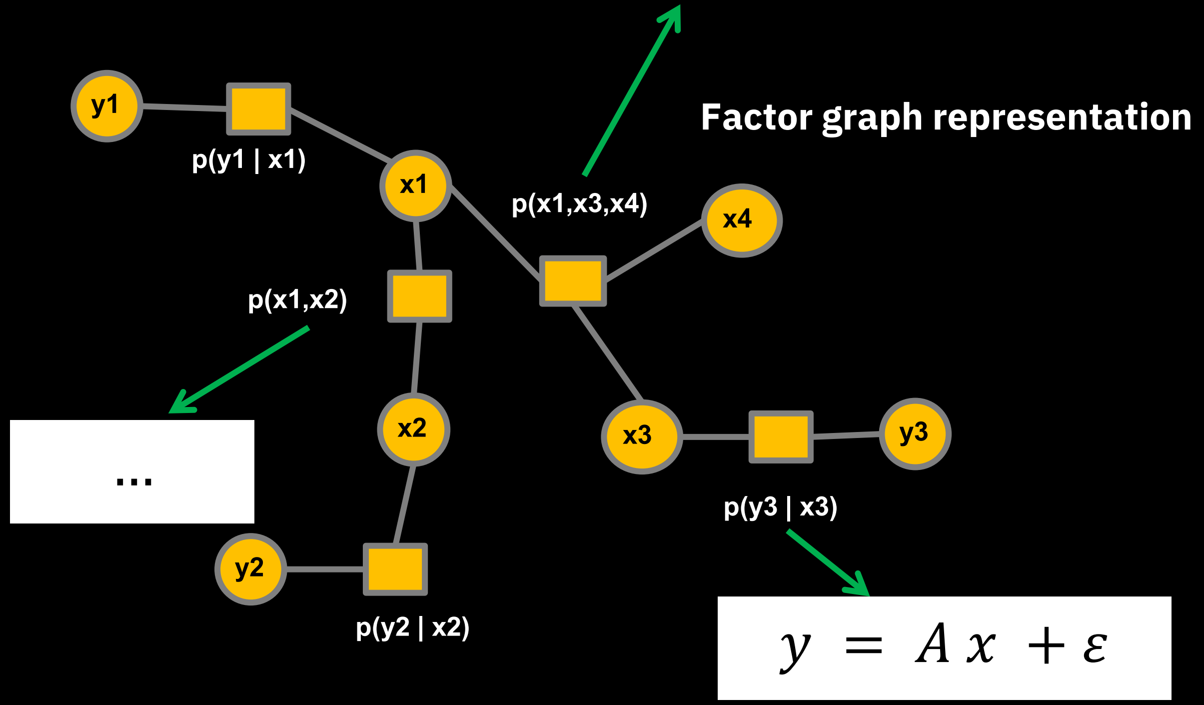


AI-based grid modelling based on Probabilistic Graphs

- (Gaussian) Probabilistic Graphs (*)
  - Embed domain knowledge (factorization)
  - <u>Flexibility</u>: extend as needed, grow with the data
  - <u>Modularity</u>: model different subsets of variables as desired (e.g. combine physics-based models and ML models)
  - <u>Scalability:</u> Sparse model
  - Naturally handle missing data: Use the same model for prediction or simulation problem



$$p(\mathcal{X}, \mathcal{Y}) = \prod_{m=1}^{M} p(y_m|\mathcal{X}_m) \prod_{k=1}^{K} p(\mathcal{X}_k)$$



(*) Fusco, F. *Probabilistic Graphs for Sensor Data-Driven Modelling of Power Systems at Scale.* Data Analytics for Renewable Energy Integration Workshop at the European Conference on Machine Learning (ECML) 2018

AI-based grid modelling based on Probabilistic Graphs

- Gaussian assumption:

$$p(\mathcal{X},\mathcal{Y}) \propto \prod_{m=1}^{M} e^{-\frac{1}{2}[y_m - f_m(\mathcal{X}_m)]^\top R_m [y_m - f_m(\mathcal{X}_m)]} \prod_{k=1}^{K} e^{-\frac{1}{2} g_k(\mathcal{X}_k)^\top S_k g_k(\mathcal{X}_k)}$$

- Inference with sum-product algorithm
  - Messages from factor to variables:

$$h_{f_j \to x_i} = h_j - \sum_{k \in \mathcal{K}_j \setminus i} J_j^{jk} (J_{x_k \to f_j} + J_j^{kk})^{-1} (h_{x_k \to f_j} + h_j^k)$$
$$J_{f_j \to x_i} = J_j - \sum_{k \in \mathcal{K}_j \setminus i} J_j^{jk} (J_{x_k \to f_j} + J_j^{kk})^{-1} J_j^{kj},$$

$$J_m = F_m^\top R_m^{-1} F_m$$
$$h_m = F_m^\top R_m^{-1} (y_m - f_m(\overline{\mathcal{X}}_m))$$

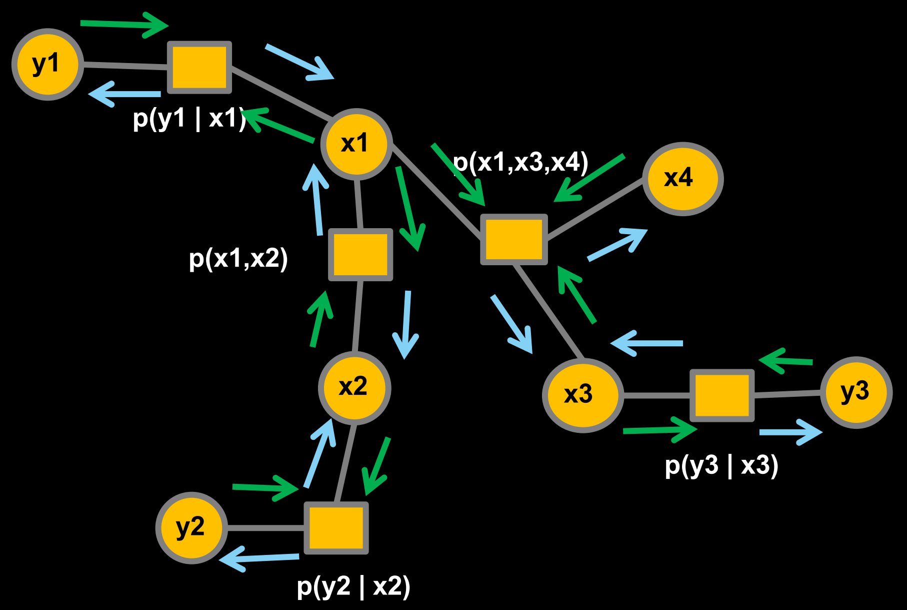
$$J_k = G_k^\top S_k^{-1} G_k$$
$$h_k = G_k^\top S_k^{-1} (\mathcal{X}_k - g_k(\overline{\mathcal{X}}_k))$$

  - Messages from variable to factors:

$$h_{x_i \to f_j} = \sum_{k \in \mathcal{K}_i \setminus j} h_{f_k \to x_i}$$
$$J_{x_i \to f_j} = \sum_{k \in \mathcal{K}_i \setminus j} J_{f_k \to x_i},$$

  - Update variables:

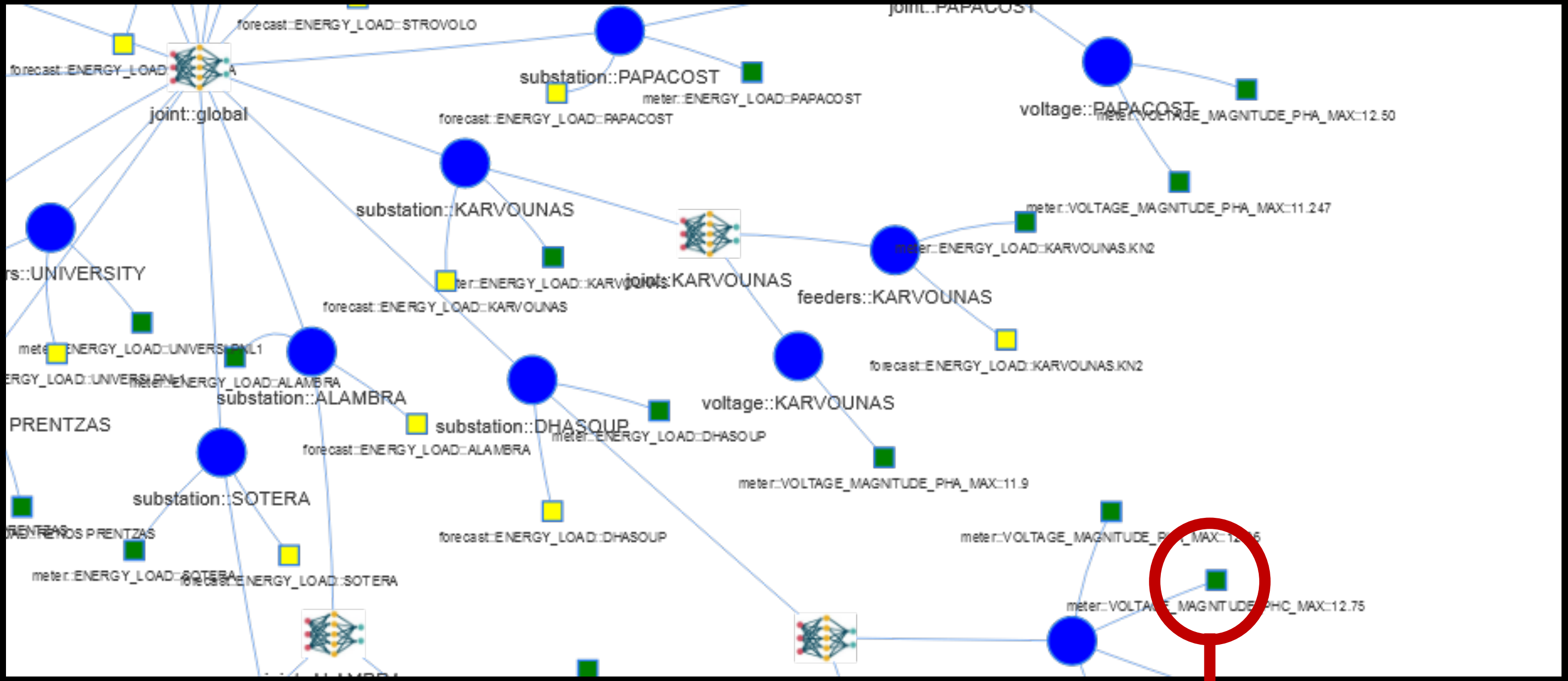
$$\delta x_i = \left(\sum_{k \in \mathcal{K}_i} J_{f_k \to x_i} \right)^{-1} \left(\sum_{k \in \mathcal{K}_i} h_{f_k \to x_i} \right)$$

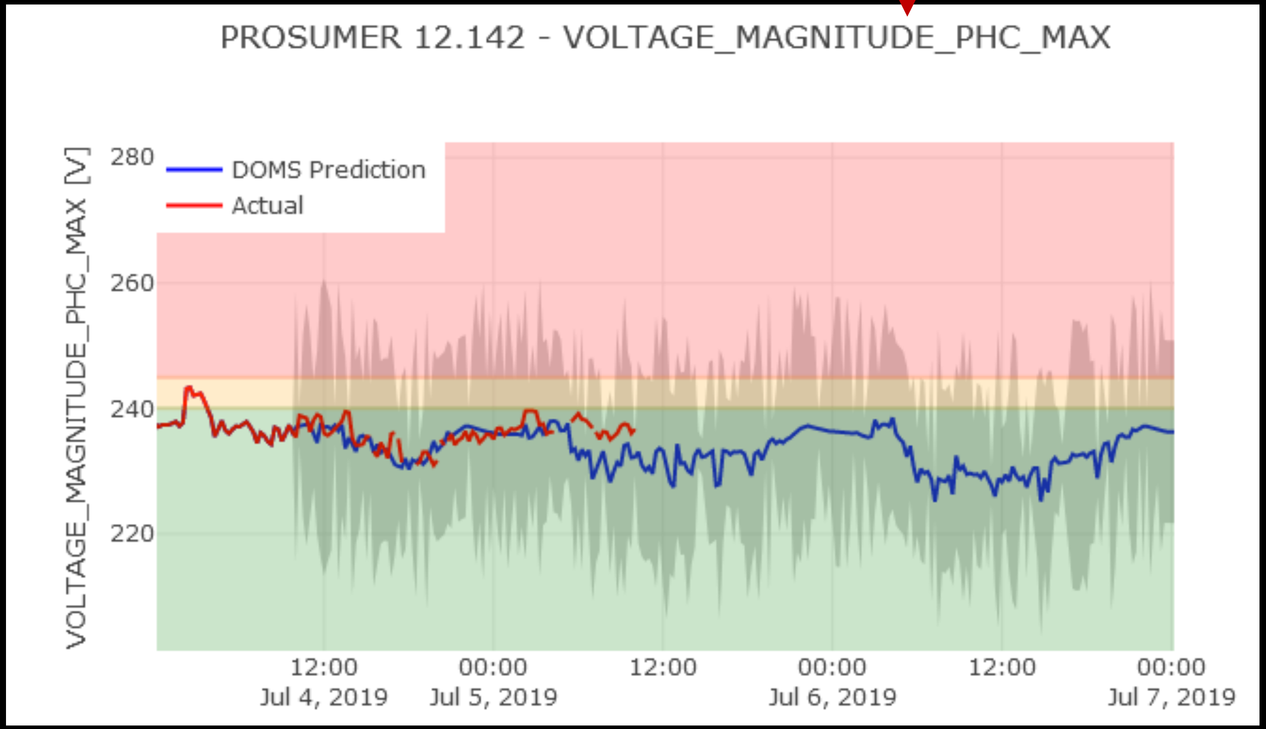


AI-based grid modelling based on Probabilistic Graphs

- GoFlex Demonstration site in Cyprus
  - Grid model of 15 substations, 29 feeders, 41 prosumers (voltage)
  - Graphical model composed of 16 NLPCA neural network models (1 x substation + 1 global)
  - Receive energy forecasts of substation / feeder loads and distributed renewable generation
  - Estimate voltage at prosumers

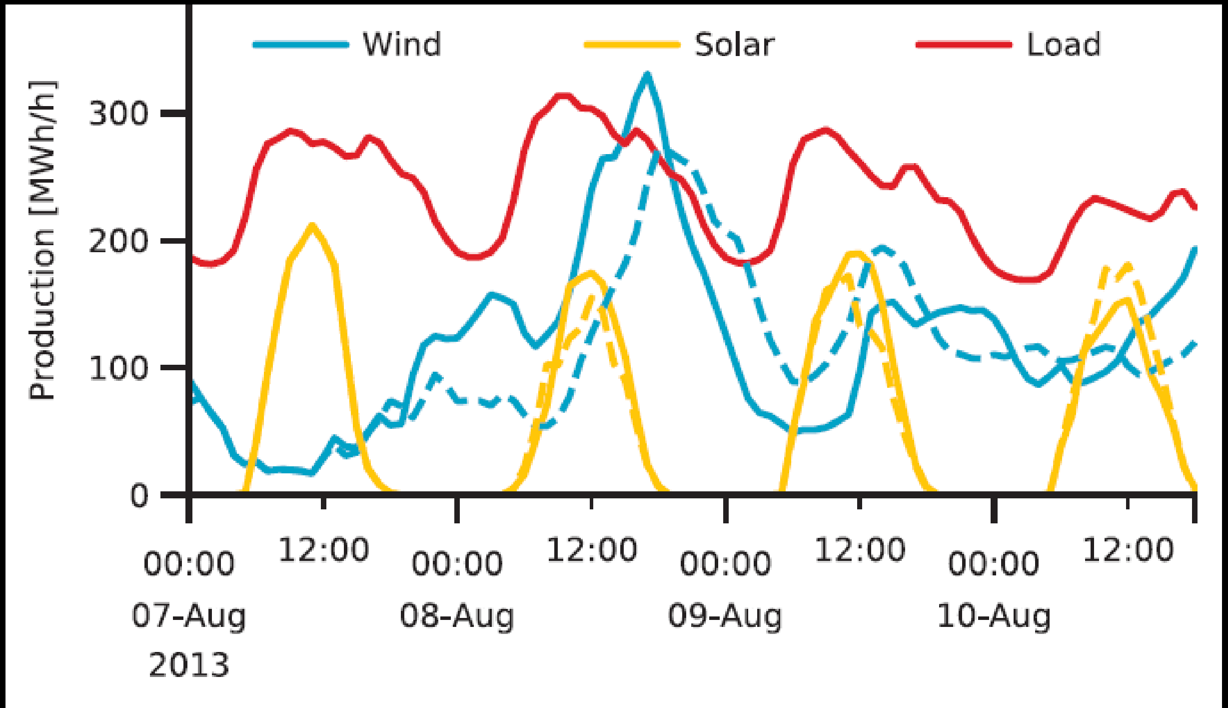
- If voltages outside bounds, simulate the graphical models to estimate energy variation required for desired profile





AI-based grid modelling based on Probabilistic Graphs
Application to European transmission grid

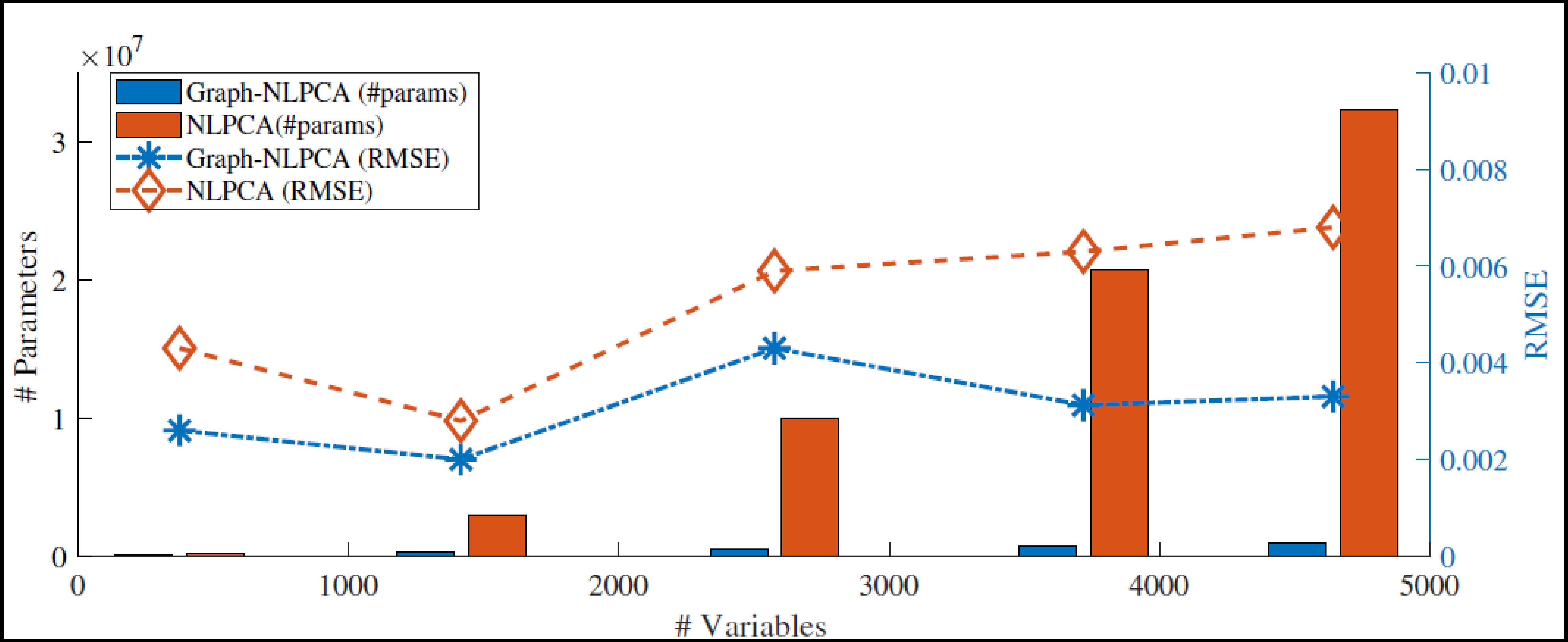
- Re-Europe **(*)**
  - 1354-nodes Europe transmission grid
  - Hourly timeseries of electrical demand, wind and solar generation
- Simulate power flow (Matpower):
  - Generate timeseries of voltage and active / reactive power
  - Add Gaussian noise with st. dev. : 10^{-3} MW (power), 10^{-5} p.u. (voltage)
- Total 8124 timeseries (1354 x 6) variables



(*) Jensen, T.V., Pinson, P.: Re-europe and a large-scale dataset for modeling a highly renewable european electricity system. Scientic Data 4:170175 (2017)

AI-based grid modelling based on Probabilistic Graphs
Application to European transmission grid

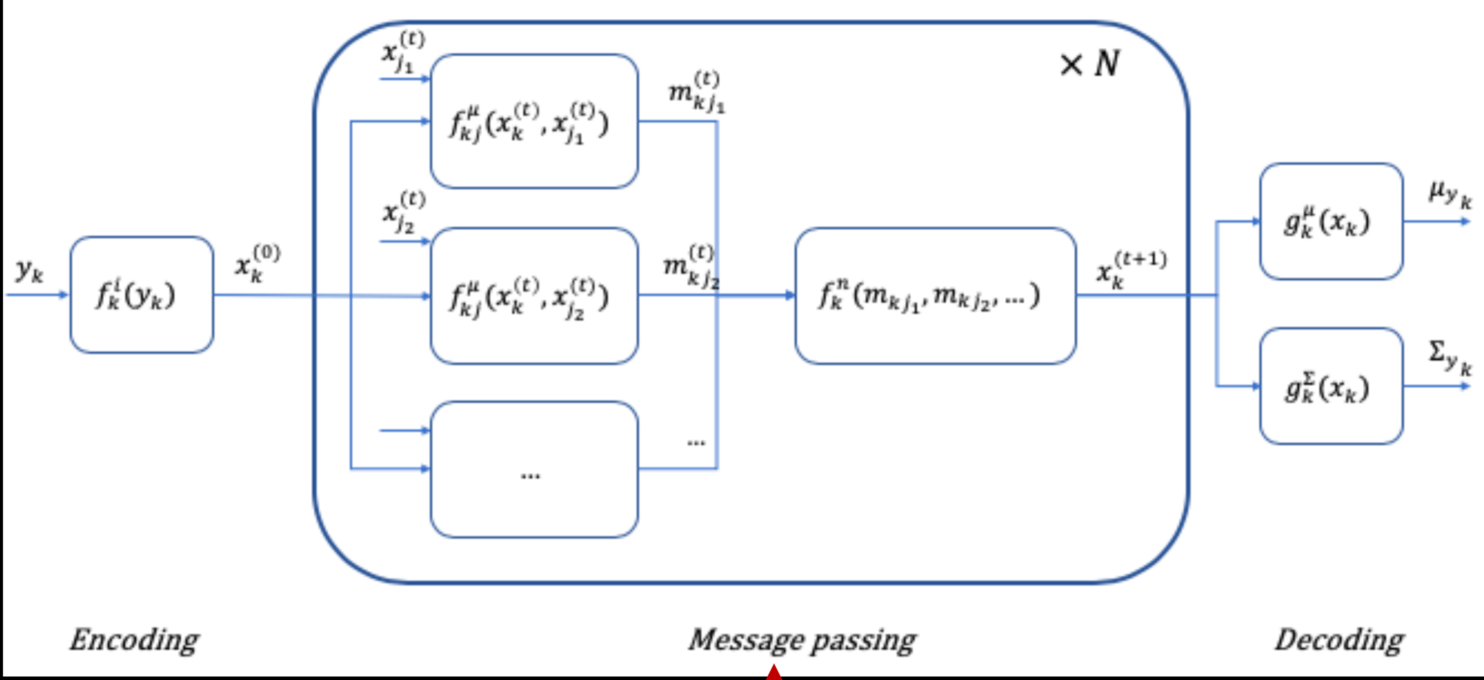
- Centralised NLPCA model deteriorates as dimensionality increases with limited training data



AI-based grid modelling based on Probabilistic Graphs

Graph Neural Networks

- Limitations of proposed graphical model
  - Derivation of principled belief propagation is not trivial for non-linear, loopy graphs
  - Belief propagation message-passing algorithms are not fully parallelizable and iterative in nature
- Use *Graph Neural Networks*
  - Message-passing as feedforward neural network
  - Learn belief propagation from the data with standard gradient-based algorithms
  - Inference is a feedforward pass



Encode

$$x_k^{(0)} = f_k^i(y_k)$$

Message Passing

$$m_{kj}^{(t)} = f_{kj}^e(x_k^{(t)}, x_j^{(t)}) \quad \forall j \in \mathcal{N}(k)$$

Message node k to node j

$$x_k^{(t+1)} = f_k^n(m_{kj}, \forall j \in \mathcal{N}(k)),$$

Aggregate messages into node k

Decode

$$\mu_{y_k} = g_k^{\mu}(x_k)$$
$$\Sigma_{y_k} = g_k^{\Sigma}(x_k)$$

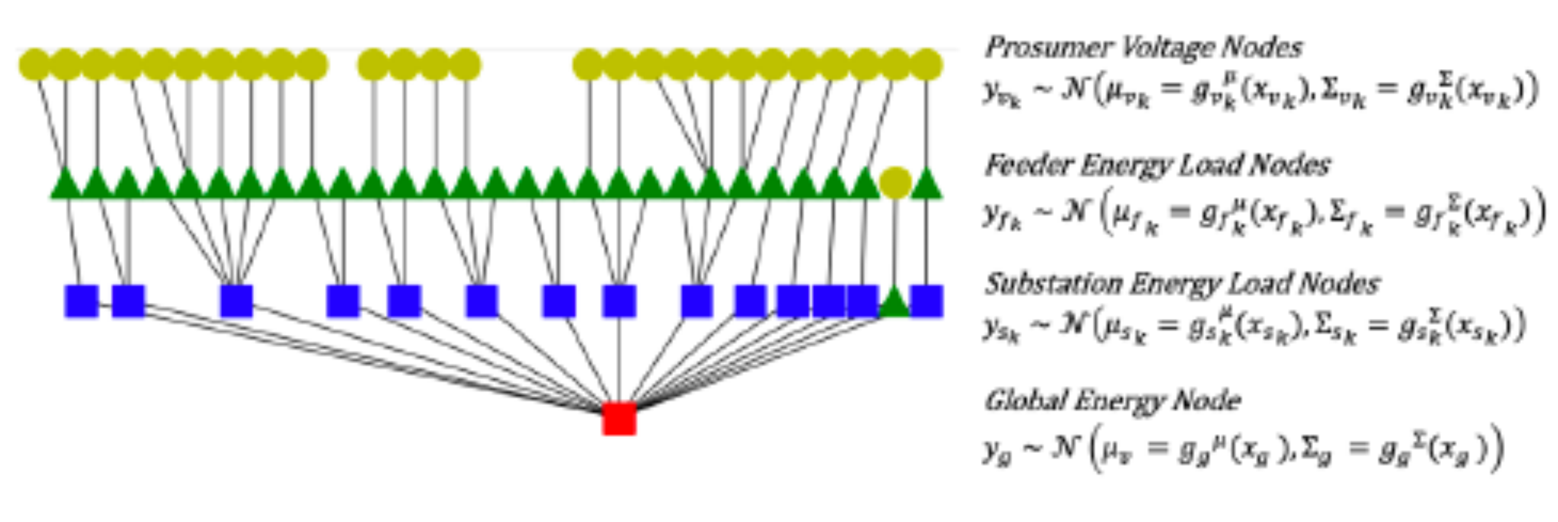
$$\mathcal{L} \propto \sum_{t=1}^{T} \sum_{k=1}^{N} \frac{1}{2} \log |\Sigma_{y_k}^t| + \frac{1}{2}(y_k^t - \mu_{y_k}^t)^\top {\Sigma_{y_k}^t}^{-1} (y_k^t - \mu_{y_k}^t)$$

Fusco, F. and Eck, B. and Gormally, R. and Purcell, M. and Tirupathi, S. "*Knowledge- and Data-driven Services for Energy Systems using Graph Neural Networks". IEEE Conference on Big Data 2020.*

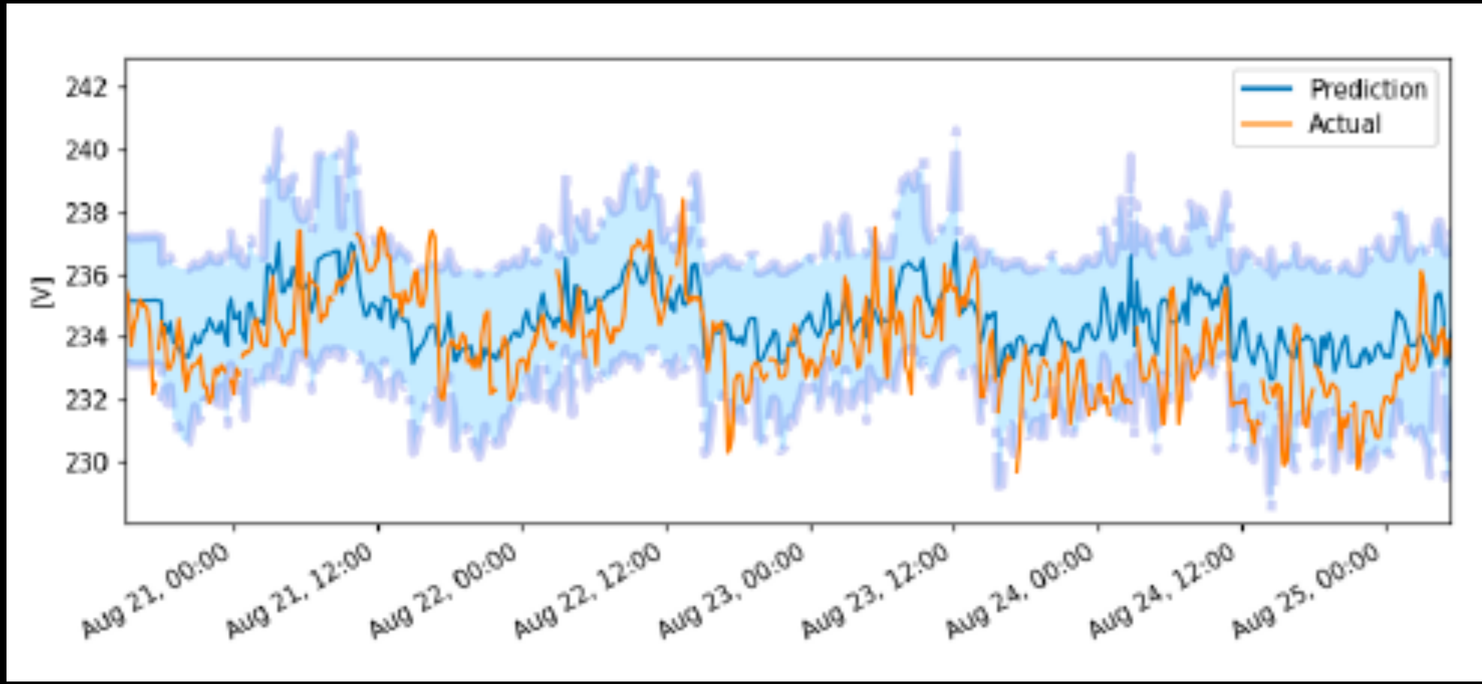
AI-based grid modelling based on Probabilistic Graphs

Graph Neural Networks

- GoFlex Demonstration site in Cyprus
  - Grid model of 15 substations, 29 feeders, 41 prosumers (voltage)
  - Graphical model composed of 16 NLPCA neural network models (1 x substation + 1 global)
  - Voltage prediction problem (solved as data imputation)

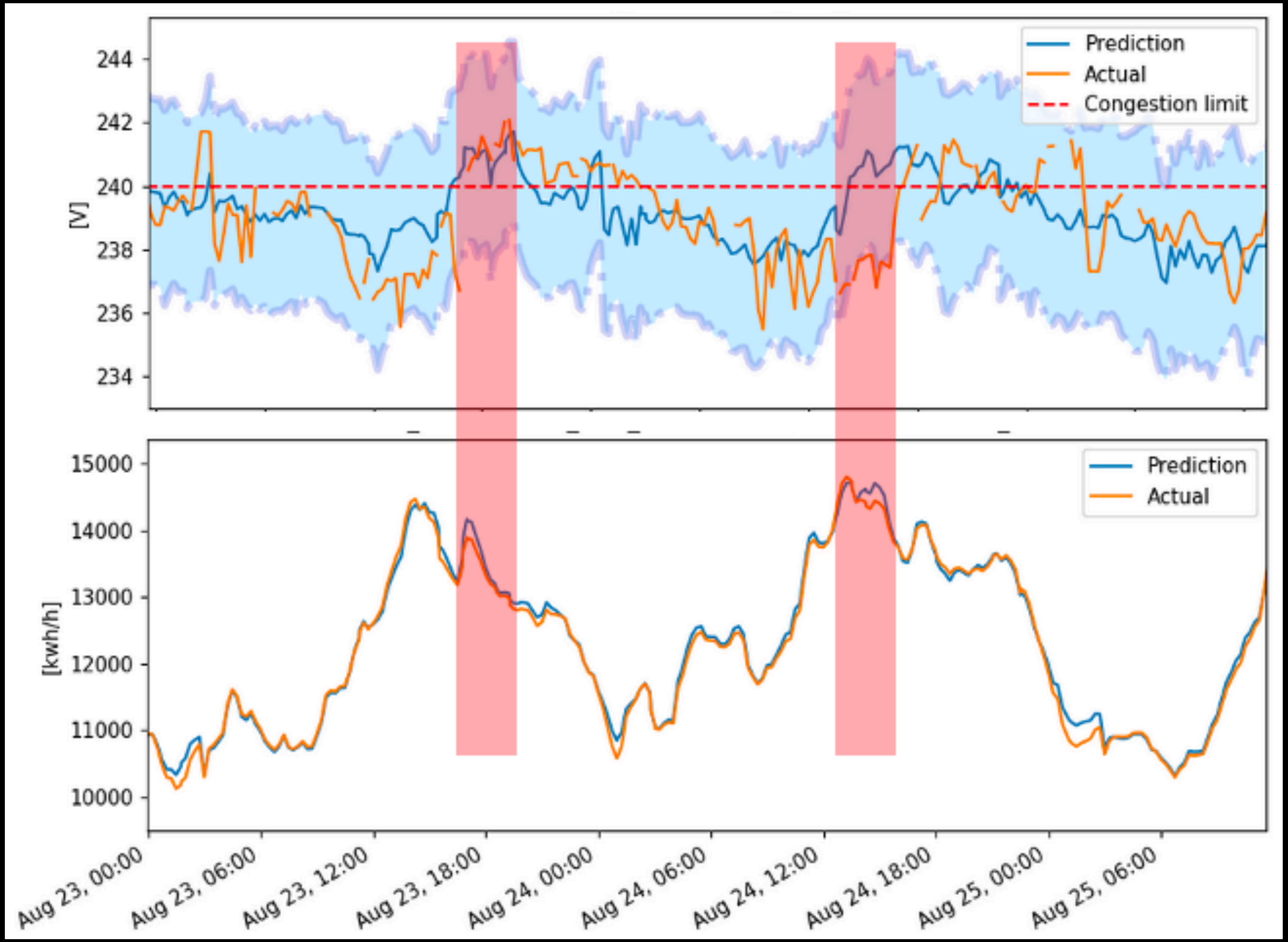


Model	# Layers	# MP	# Params	MAPE	RMSE
GNN	2	5	143, 464	0.81%	2.42
GNN	3	5	169, 144	0.82%	2.45
GNN	2	8	142, 832	0.80%	2.40
GNN	3	8	169, 144	0.81%	2.41
MLP	2	–	1, 271, 440	0.80%	2.40
MLP	3	–	2, 118, 760	0.73%	2.21
AE	2	–	2, 407, 088	0.72%	2.17
AE	3	–	3, 796, 840	0.70%	2.13



AI-based grid modelling based on Probabilistic Graphs
Graph Neural Networks

- GoFlex Demonstration site in Cyprus
  - Example of using AI grid model to generate flexibility bids to avoid congestion



Thank you, Questions ?

—

- References

  - Fusco, F. and Eck, B. and Gormally, R. and Purcell, M. and Tirupathi, S. "Knowledge- and Data-driven Services for Energy Systems using Graph Neural Networks". IEEE Conference on Big Data 2020.
  - Eck, B. and Fusco, F. and Gormally, G. and Purcell, M. and Tirupathi, S. "Scalable Deployment of AI Time-series Models for IoT". AI4IoT Workshop at IJCAI 2019
  - Eck, B. and Fusco, F. and Gormally, G. and Purcell, M. and Tirupathi, S. "*AI Modelling and Time-series Forecasting Systems for Trading Energy Flexibility in Distribution Grids*". ACM e-Energy 2019
  - Fusco, F. "*Probabilistic Graphs for Sensor Data-Driven Modelling of Power Systems at Scale*". Data Analytics for Renewable Energy Integration Workshop at the European Conference on Machine Learning (ECML) 2018
  - Fusco, F. and Gormally, R. and Tirupathi, S. "*Power systems data fusion based on belief propagation*". IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe) 2017

francfus@ie.ibm.com