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
- IBM developed
  - Scalable energy forecasting platform
  - AI-based grid modelling for DSO bidding decisions

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

- **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

**Built on the IBM Cloud**

1. Live time-series data ingestion
2. Application semantics
3. Develop / customize AI models
4. Deploy AI models
5. Automatic execution of model train/score jobs
6. Automatic persistence of 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)
IBM Research Castor: Scalable time-series forecasting platform

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

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

<table>
<thead>
<tr>
<th>Site</th>
<th># Sensors</th>
<th># Models</th>
<th>Execution [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Germany</td>
<td>18</td>
<td>11</td>
<td>16.8</td>
</tr>
<tr>
<td>Switzerland</td>
<td>196</td>
<td>61</td>
<td>19.7</td>
</tr>
<tr>
<td>Cyprus</td>
<td>531</td>
<td>174</td>
<td>15.9</td>
</tr>
</tbody>
</table>

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

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

Table 3: System scalability analysis.

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

- **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
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

- AI-based modelling
  - 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)
  - **Flexibility**: extend as needed, grow with the data
  - **Modularity**: model different subsets of variables as desired (e.g. combine physics-based models and ML models)
  - **Scalability**: Sparse model
  - Naturally handle missing data: Use the same model for prediction or simulation problem

\[
p(X, Y) = \prod_{m=1}^{M} p(y_m | x_m) \prod_{k=1}^{K} p(X_k)
\]

AI-based grid modelling based on Probabilistic Graphs

- Gaussian assumption:
- Inference with sum-product algorithm
  - Messages from factor to variables:
    \[ h_{f_j \rightarrow x_i} = h_j - \sum_{k \in \mathcal{K}_j \setminus i} J^k_j (J_{x_k \rightarrow f_j} + J^k_{f_j})^{-1} (h_{x_k \rightarrow f_j} + h^k_j) \]
    \[ J_{f_j \rightarrow x_i} = J_j - \sum_{k \in \mathcal{K}_j \setminus i} J^k_j (J_{x_k \rightarrow f_j} + J^k_{f_j})^{-1} J^k_j. \]
  - Messages from variable to factors:
    \[ h_{x_i \rightarrow f_j} = \sum_{k \in \mathcal{K}_i \setminus j} h_{f_k \rightarrow x_i} \]
    \[ J_{x_i \rightarrow f_j} = \sum_{k \in \mathcal{K}_i \setminus j} J_{f_k \rightarrow x_i}. \]
  - Update variables:
    \[ \delta x_i = \left( \sum_{k \in \mathcal{K}_i} J_{f_k \rightarrow x_i} \right)^{-1} \left( \sum_{k \in \mathcal{K}_i} h_{f_k \rightarrow x_i} \right) \]
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. Scientific Data 4:170175 (2017)
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

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)
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


francfus@ie.ibm.com