Deep Graph Learning of PMU Data for Real-Time Event Identification

Presenter: Yuxuan Yuan

Advisor: Zhaoyu Wang

Harpole-Pentair Assistant Professor

Iowa State University

IEEE PES GM 2021

Presentation Overview

- Introduction to PMU Data
- Deep Graph Learning for Real-Time Event Identification
  - Existing Work and Challenges
  - Graph Structure Parameterization
  - Multi-Scale Data Feature Extraction
- Conclusion and Future Work

Background

- This work is supported by the U.S. Department of Energy Office of Electricity under DEOD0000910. It focuses mainly on the use of real PMU data to develop real-time event identification using machine learning techniques.

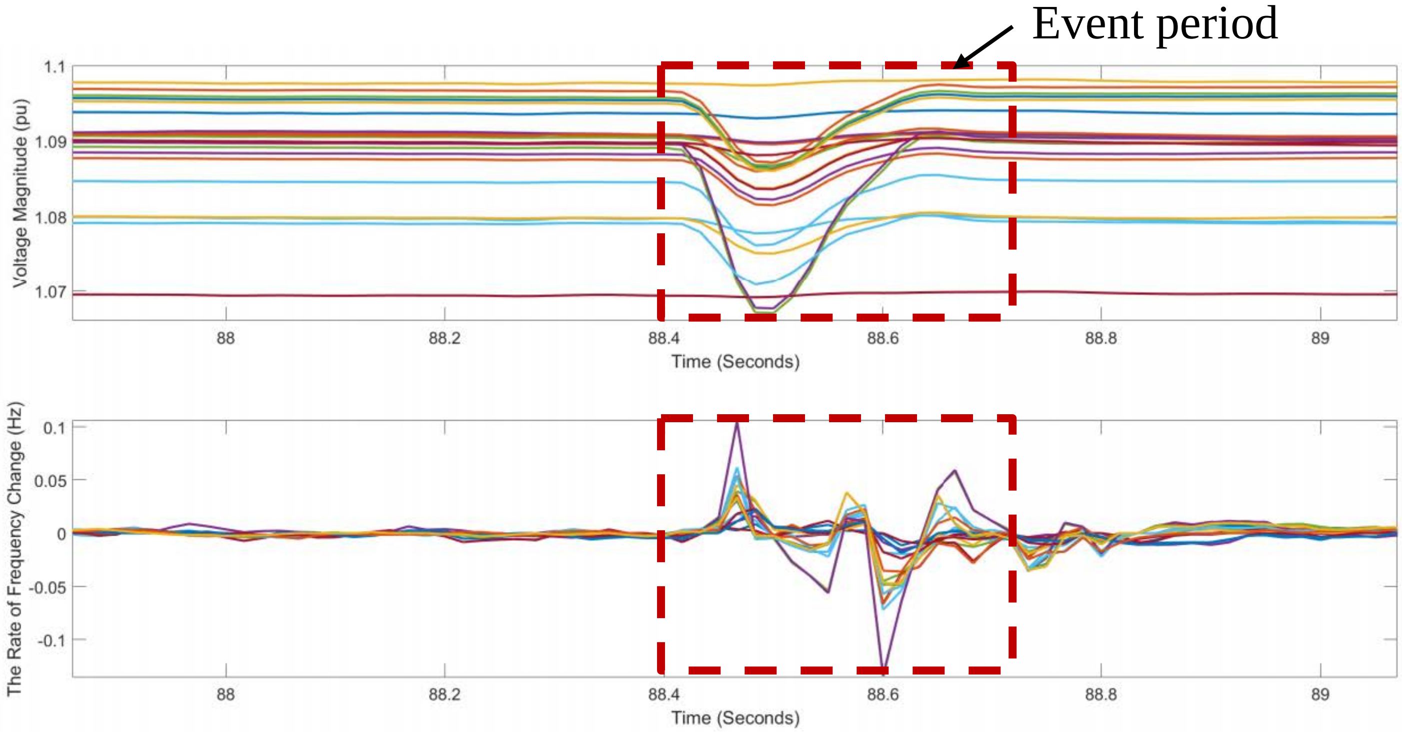
- The available PMU data is obtained from 440 PMUs installed across three U.S. transmission interconnections that include Texas, Western, and Eastern interconnection. Most data segment is archived at 30 frames/s and the remaining is archived at 60 frames/s. The total size of the dataset is more than 20 TB in Parquet form.

- A total of 6767 event labels recorded by utilities are utilized to provide the ground truths.

Introduction to Real-world PMU Data

	Interconnection A	Interconnection B	Interconnection C
Number of PMUs	212	43	188
PMU Reporting Rate (Samples/sec)	30	30, 60	30
Data Size	3 TB	5 TB	12 TB
Record Period	1 Year	2 Years	2 Years
No. of Data Files	2576	4365	10496
Total Number of Events	29	4854	1884
% of Good Data	66%	70% (30 samples/sec) 75% (60 samples/sec)	67%

Introduction to Real-world PMU Data



Challenges of Data-driven Event Identification

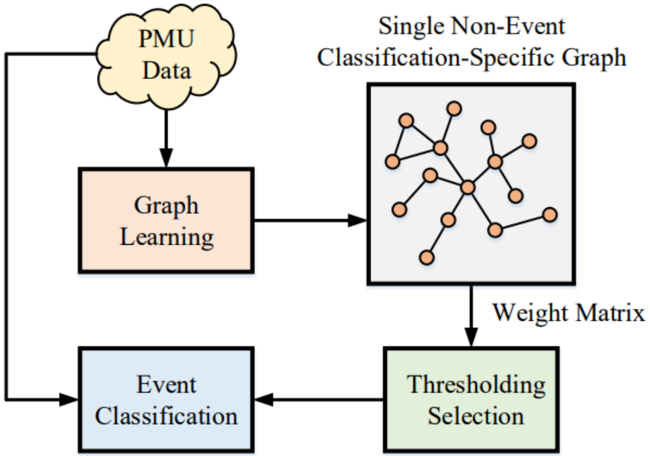
Challenges:

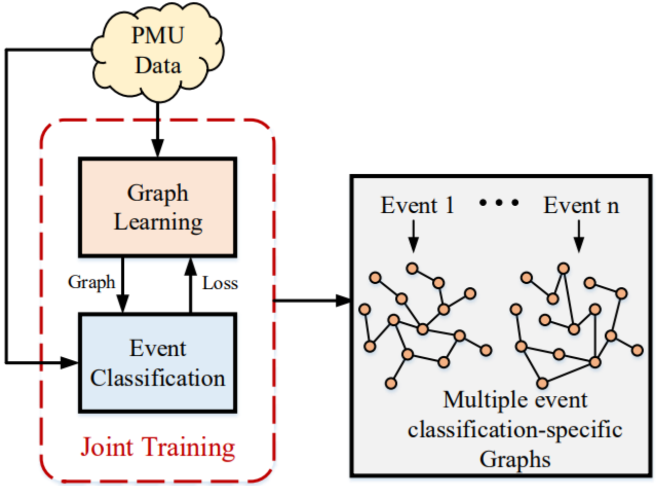
- Event identification based on a single PMU's data may be inaccurate and unreliable.
- How to take full advantage of all PMUs' data to improve the accuracy of event identification?
- The event identification model may suffer the curse of dimensionality if all PMUs' data is used.
- Feature reconstruction may be challenging if multiple PMUs' data is used. As the number of PMUs increases, the computational complexity of the feature reconstruction grows significantly, which impacts the real-time performance of the event identification model.

Solutions

Previous Solution:
- ✓ The graph learning task and event identification task are separated (suboptimal).
- ✓ The graphs are not event type-specific.
- ✓ Generating a single statistical graph for entire dataset (ignore the uncertainty of event locations).

Our Solution:
- ✓ Joint learning of the latent interaction and the event identification model.
- ✓ The graphs are event type-specific.
- ✓ Generating one interaction graph for each single event.





Graphical PMU-based Event Identification

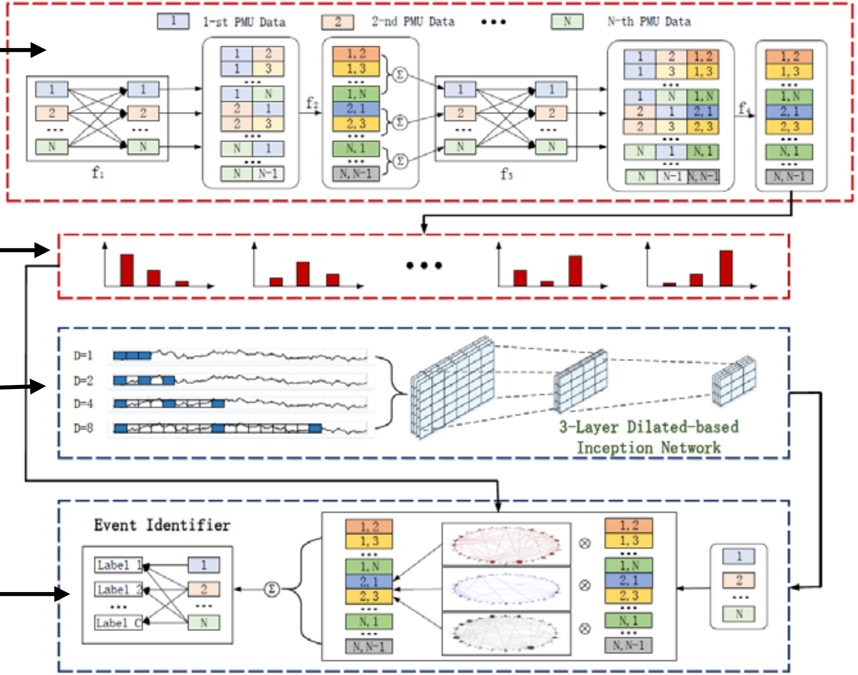
- The overall model follows an auto-encoder structure.
- Encoder: inferring the interaction graph given PMU data streams.
- Decoder: performing the event classifier by combining the features and the constructed graph

Encoder: Graph Inference

Encoder: Graph Sampling

Decoder: Data feature extraction

Decoder: Event identification

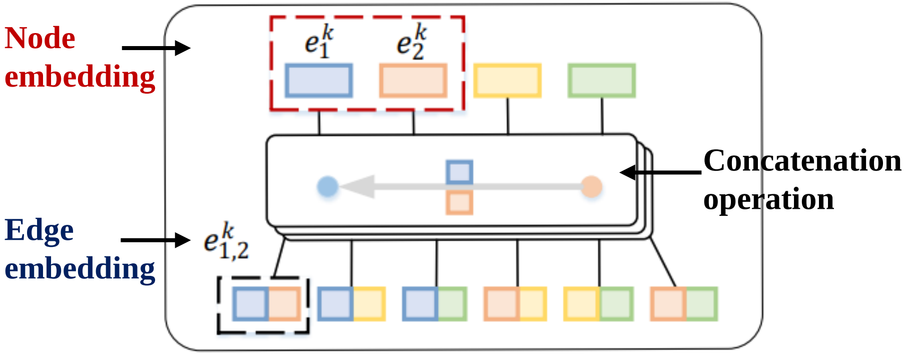


Graphical PMU-based Event Identification

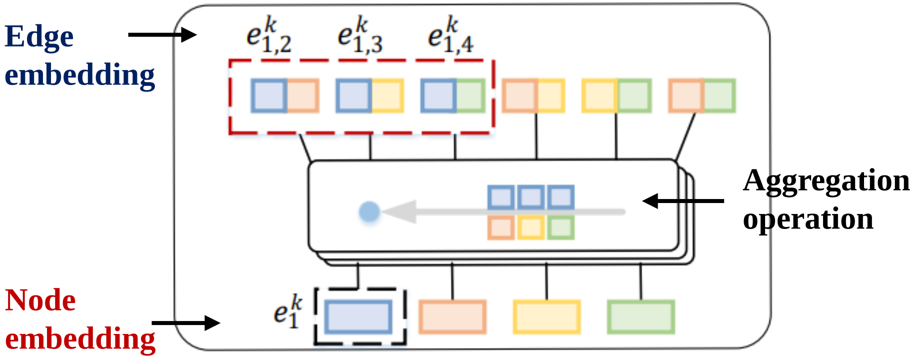
- Node: PMU
- Edge: The interdependence between two PMUs.
- Node/Edge Embedding: Using a vector to represent a node/edge.
- We have utilized the Bernoulli distribution to represent the graph structure.
- We have utilized the deep relational network for inferencing the latent relationship between different nodes.

$$e^{k}_{i,j} = f^{k}_{e}([e^{k}_{i}, e^{k}_{j}, x_{(i,j)}])$$

$$e^{k+1}_{i} = f^{k}_{n}([\sum_{i \in N_j} e^{k}_{(i,j)}, x_j])$$



(a) Node to Edge

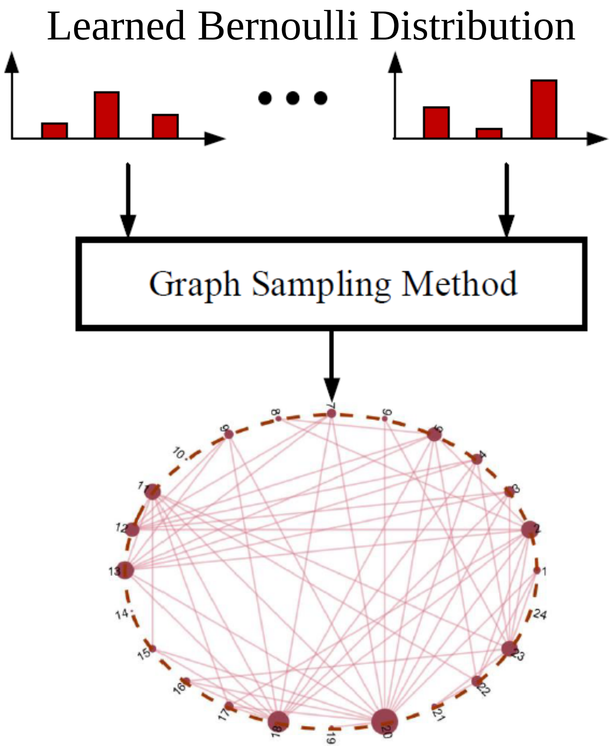


(b) Edge to Node

Graph Structure Parameterization (Encoder)

- For each event, one interaction graph is sampled from the learned Bernoulli distribution, which can handle the uncertainty of event locations.

- We have tested three different graphing sampling methods:
  - Stochastic Sampling (unweighted graph)
  - Deterministic Thresholding (unweighted graph)
  - Continuous Sampling (weighted graph)



Graph Structure Parameterization – Graph Sampling

Since Bernoulli distribution-based parameterization imposes a challenge on differentiability back-propagation process, we have utilized the Gumbel reparameterization technique:

$$z_{i,j} = \frac{\exp\left((log(e_{i,j}^{m}) + g_m)/\tau\right)}{\sum_{m=1}^{N} \exp\left((log(e_{i,j}^{m}) + g_m)/\tau\right)}$$

where, g_m is independent and identically distributed (i.i.d) sample drawn from Gumbel distribution with 0 location and 1 scale parameters, τ is a smooth coefficient and is assigned as 0.5 in this work.

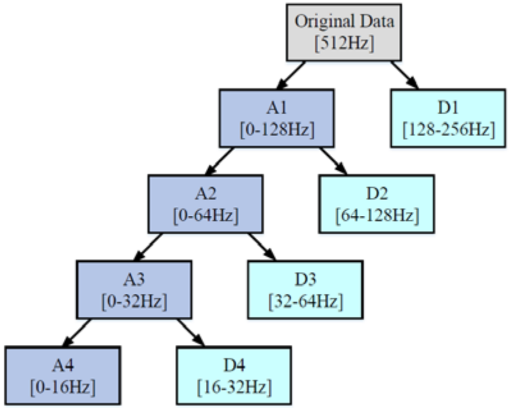
Multi-Scale Data Feature Extraction (Decoder)

Previous Solution:

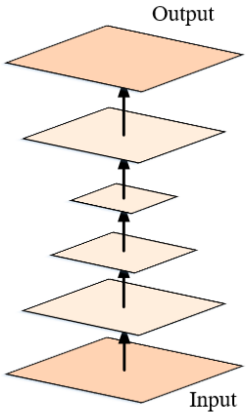
- ✓ Wavelet transform-based multi-resolution analysis (high computation burden).
- ✓ MTF-based feature reconstruction (high computation burden).
- ✓ Standard CNN-based feature extractor (only capture single-scale feature).

Our Solution:

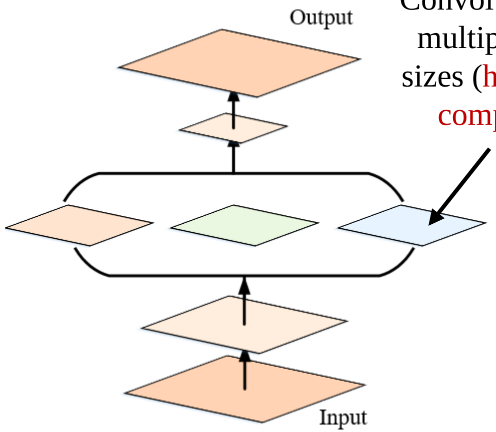
- ✓ Using **inception-based model** to capture multi-scale data features.
- ✓ Using **dilated convolution** layer to replace standard convolution layer for reducing the complexity of the model.



Wavelet Transform



Standard Feature Extractor



Convolutions with multiple kernel sizes (high model complexity)

Inception-based Model

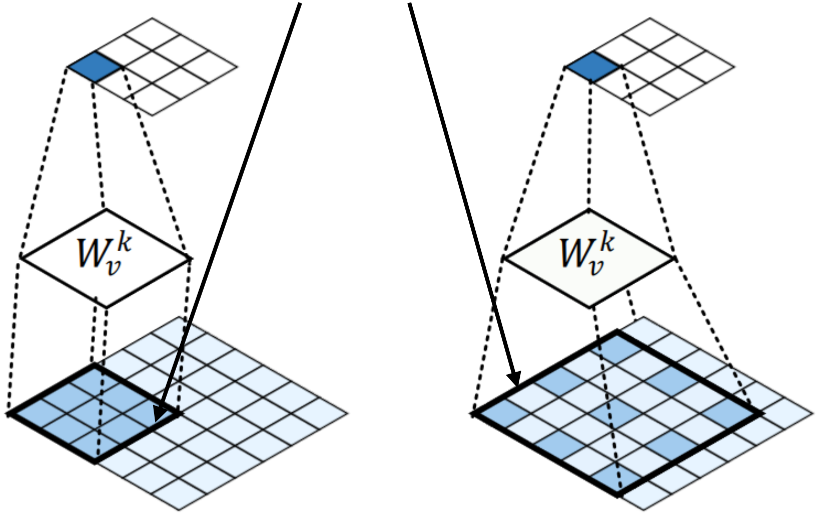
Multi-Scale Data Feature Extraction (Decoder)

- The main idea of dilated convolution is to insert zeros between two consecutive features in the the convolutional kernels, which significantly increases the receptive filed.

- Based on the figure, it is clear a dilated 3×3 convolutional kernel with d = 2 has a similar receptive field with a standard 5×5 convolutional kernel.

- d is a dilation rate that defines a spacing between the values in a convolutional kernel.

Same kernel size but larger receptive field



Standard Convolution Layer

d-2 Dilated Convolution Layer

Hyperparameter Tuning

- Hyperparameter: Adjustable parameter whose value is used to control the learning process.

- Hyperparameters are tuned via a grid search strategy: the number of hidden neurons, the threshold of graph sampling, the smooth coefficient of Gumbel-Max technique, dilated rate, the number of graph-layer.

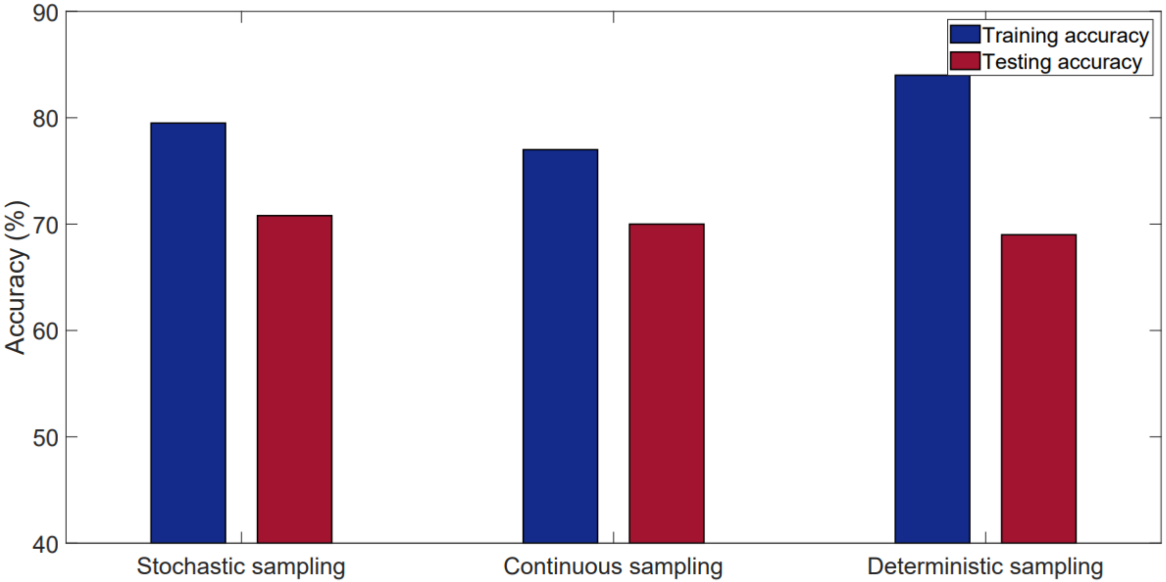
- The proposed method is verified using the data of one interconnection. The event logs are utilized as the ground truths (around 9600 data samples: 4800 event samples + 4800 normal operation samples).

- We perform a temporal 70/15/15 split for training, validation, and testing, respectively.

Numerical Results

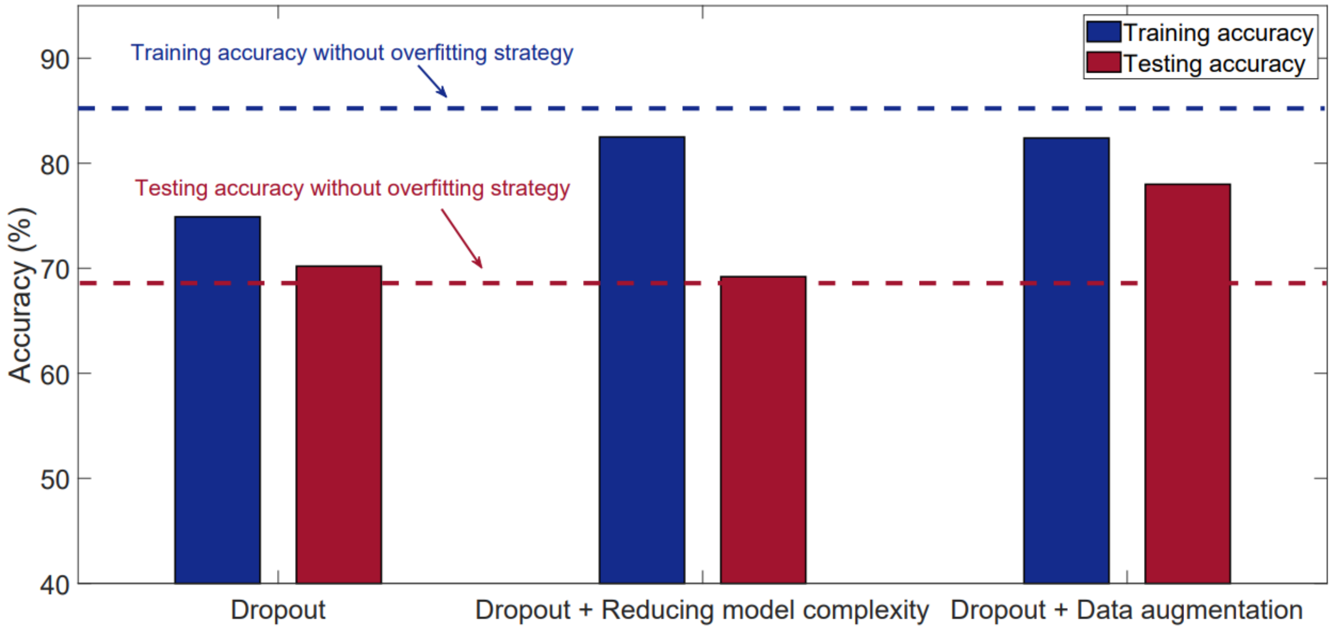
- We compare three different event identification methods:
  - ✓ Our proposed method: graph neural network-based method with interaction graphs.
  - ✓ Non-deep learning method without interaction graphs: support vector machine (SVM).
  - ✓ Deep learning method without interaction graphs: CNN-based method.
- All methods are evaluated using the mean absolute percentage error (MAPE).
- The average online computation time for performing the proposed method is around 0.0156 s (using a standard PC with an Intel(R) Xeon(R) CPU running at 4.10GHZ and with64.0GB of RAM and an Nvidia Geforce GTX 1080ti 11.0GB GPU).

Comparison of Three Graph Sampling Methods



- In this case, the deterministic thresholding method shows a slightly better performance than two other sampling methods.
- The difference between the training and testing accuracy indicates the overfitting problem.

Comparison of Three Methods to Prevent Overfitting



- Based on different overfitting strategies, the training accuracy decreases from around 84% to around 82%; the testing accuracy increases from 68% to around 78%.

Comparison of Three Event Identification Methods

Method	Testing accuracy
Proposed method	78%
CNN-based method	60%
Support vector machine (SVM)	63%

- This table summarizes the event classification testing accuracy of the proposed model and existing two methods.

- Based on testing accuracy, the proposed method has a better performance (78%) than other methods ({60%,63%}) in this case, indicating that data-driven inference of interaction graphs is effective.

Conclusion and Future Work

- PMUs provide high-granularity and synchronized measurements, including voltage and current phasor, frequency, and frequency variation, which enables capturing most dynamics of power systems.

- We demonstrated how to use multiple PMU data streams together with deep learning for identifying system events.

- In the future, this work will be extended by integration with semi-supervised learning and federated learning techniques and to deal with the event mismatch and data privacy problems prevalent in real-world grids.

Thank you!

Q&A