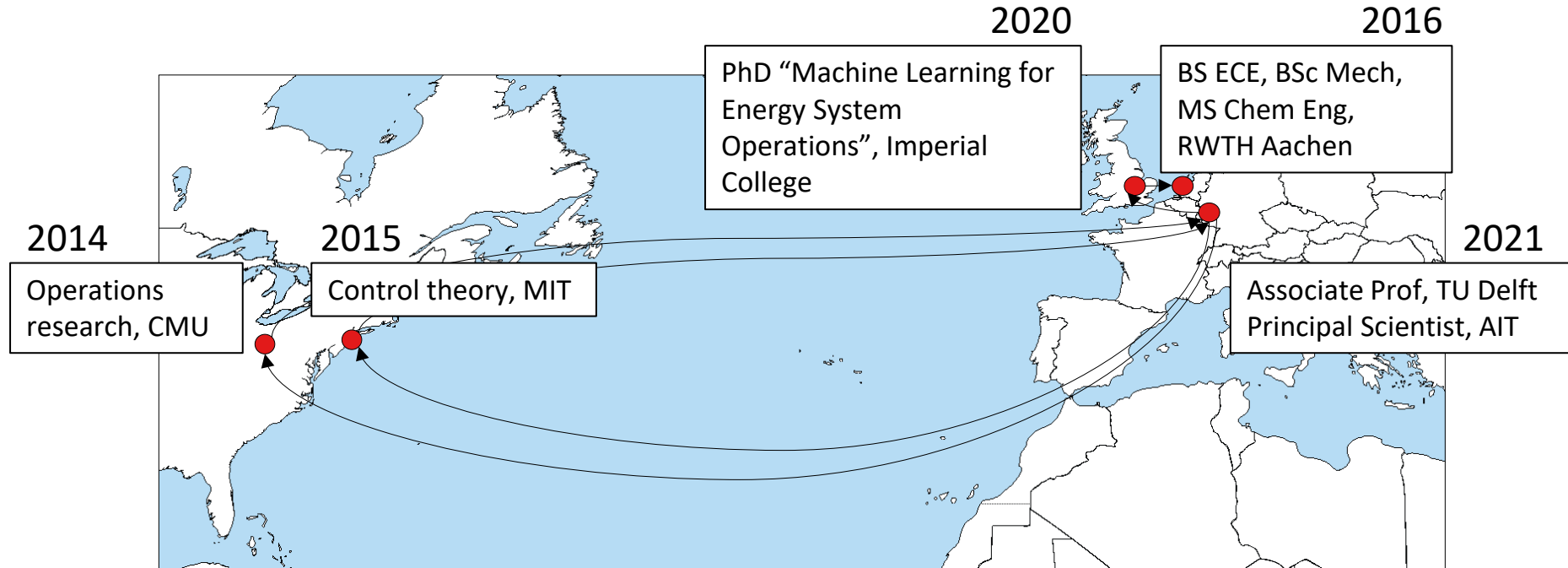


Power System Reliability with Deep Learning

ACM e-Energy'25
17-05

Prof. Dr. Jochen L. Cremer,
Associate Professor

Introduction



Delft AI Energy Lab

Mission & objective

- combine groundbreaking ML with the reliable theory of the physical energy system
- make energy systems sustainable, reliable, effective

Education

- EE4C12 ML for Electrical Engineering
- SET 3125 Machine Learning Workflows for Digital Energy Systems
- SC42150 Statistical Signal Processing
- SC42110 Dynamic Programming and Stochastic Control
- MOOC Digitalization of Intelligent and Integrated Energy Systems
- Crash course of “Data-science”

Research

- Supervised learning for real-time grid assessment
- Distributed learning for power system congestion management
- Data-driven grid models for electricity load and weather forecasts
- Characterizing healthy/normal trajectories of complex dynamical systems using dictionary learning
- From fast Fourier transform to fast reinforcement learning

Key innovations

- AI-based algorithms for grid operation
- Real-time security assessment and anomaly detection
- Real-time learning algorithms for control and security of complex dynamical systems

Team



Jochen Cremer



Peyman Mohajerin Esfahani



Haiwei Xie



Olayiwola Arowolo



Shabnam Khodakaramzadeh



Mert Karaçelebi



Basel Morsy



Mohammad Boveiri



Ali Rajaei



Jochen Stiasny



Viktor Zobernig



Demetris Chrysostomou



Luca Hofstadler



Betül Mamudi
Paul Bannmüller



Runyao Yu



Péline Cunat



AI Initiative

<https://www.tudelft.nl/ai/delft-ai-energy-lab>

[DAIEnergyLab](#)

DAI-Energy-Lab@tudelft.nl



Outline

Reliability management and data in control rooms

1. Introduction to reliability management
2. Machine learning approaches
3. Security assessment with cost-sensitive supervised learning

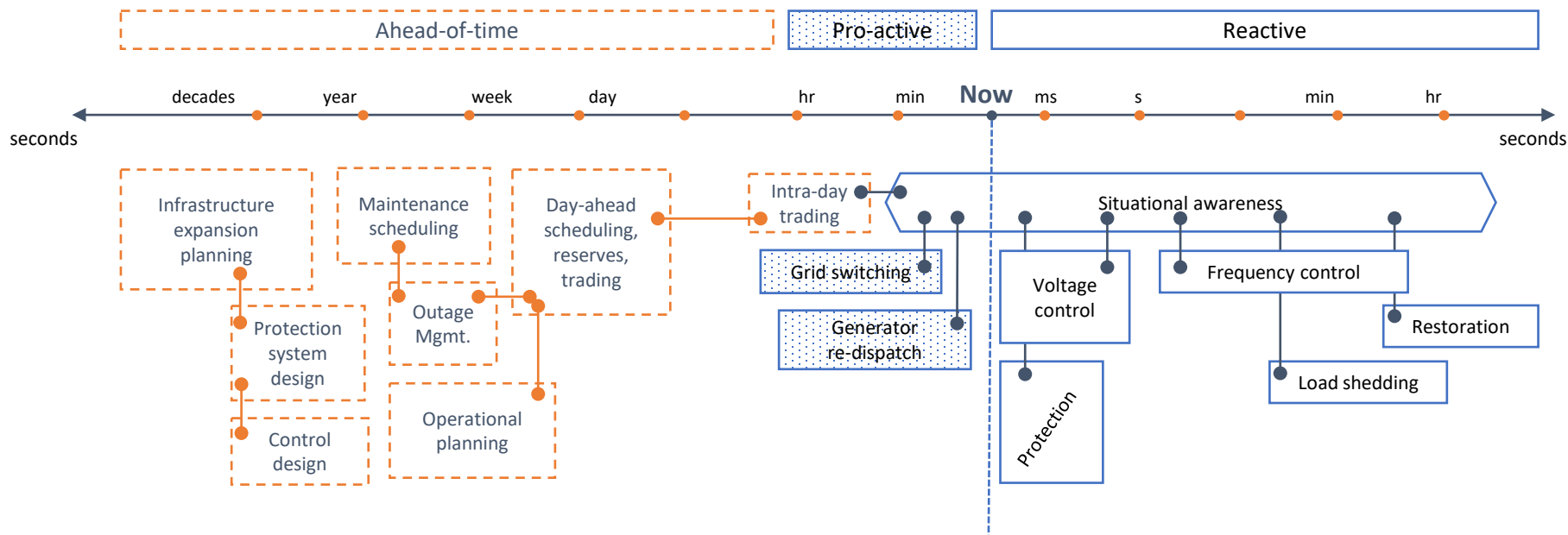
Learning models for secure system operation

4. Learning with domain knowledge
5. State estimation with graph neural networks
6. Weakly-supervised learning for secure operation
7. Challenges applying ML to reliability



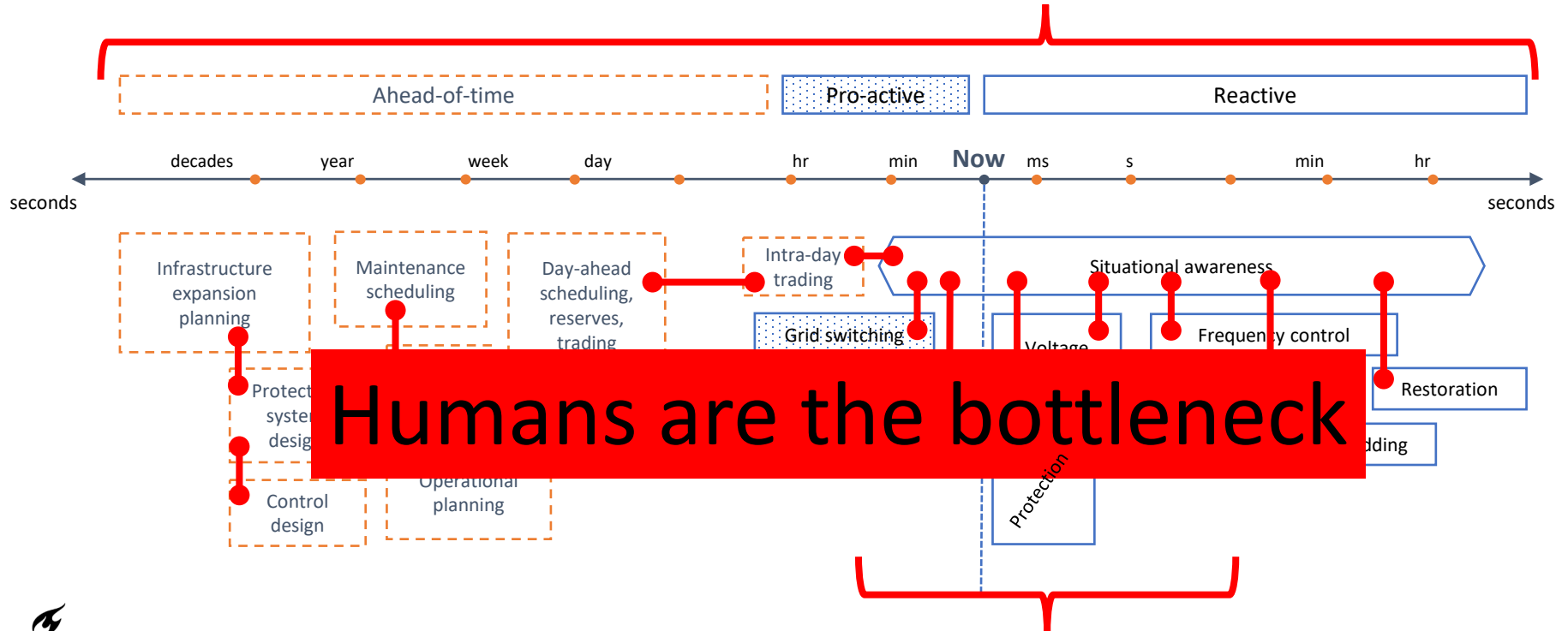
Experts are in charge to manually operate the power system based on experience and with the support of tools

A complex process



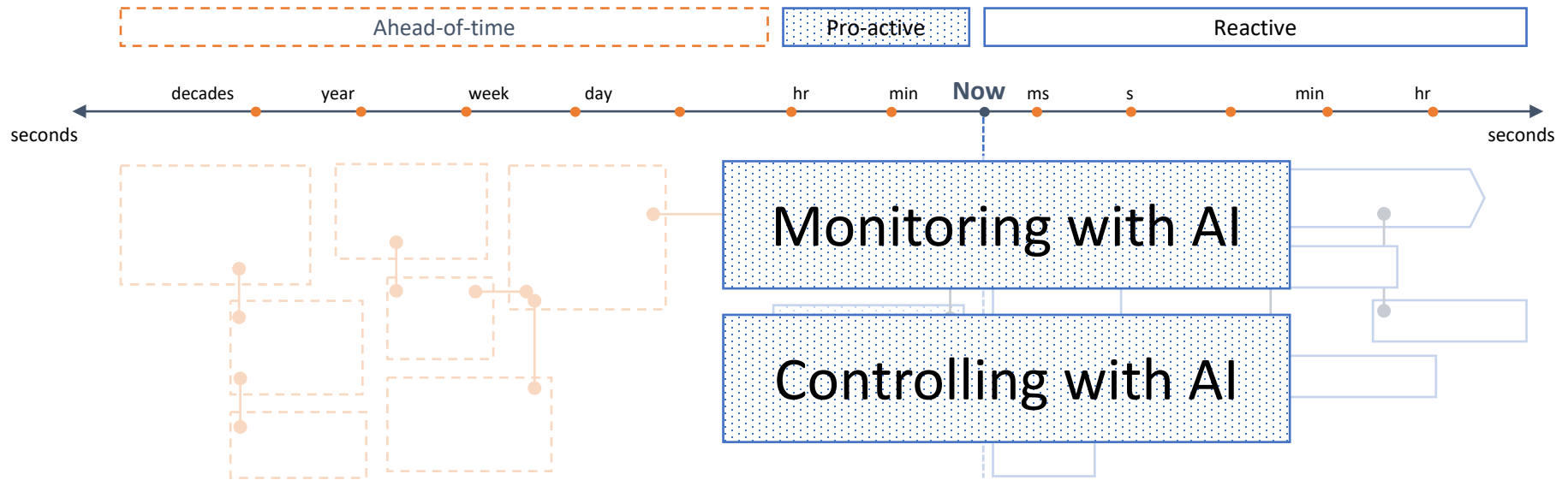
What's the issue?

Interdependencies
challenge manual rules



Human decisions are too slow

Automation first realised where urgently needed

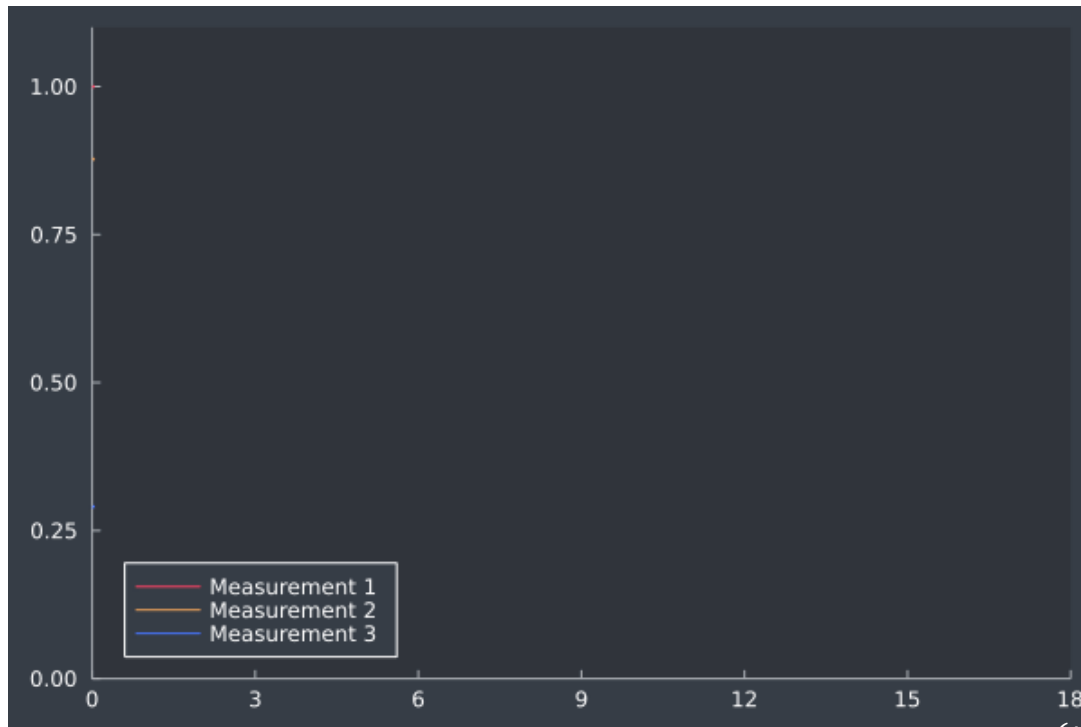


Real-time security assessment of disturbances



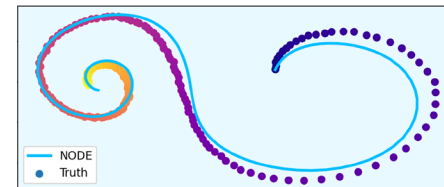
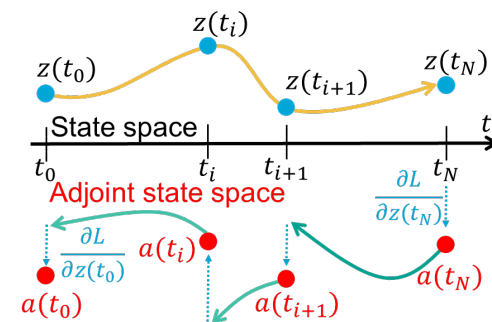
Mert Karaçelebi

Phase angles [norm]



Time [s]

Neural ordinary differential equations



[1] Mert Karaçelebi, Jochen L. Cremer "Online Neural Dynamics Forecasting for Power System Security", *International Journal of Electrical Power & Energy Systems* 2025

[2] Mert Karaçelebi, Jochen L. Cremer, "Predicting Power System Frequency with Neural Ordinary Differential Equations", *12th Bulk Power System Dynamics and Control Symposium and Sustainable Energy, Grids and Networks Journal*, 2025

Why do system operators require reliability monitoring?



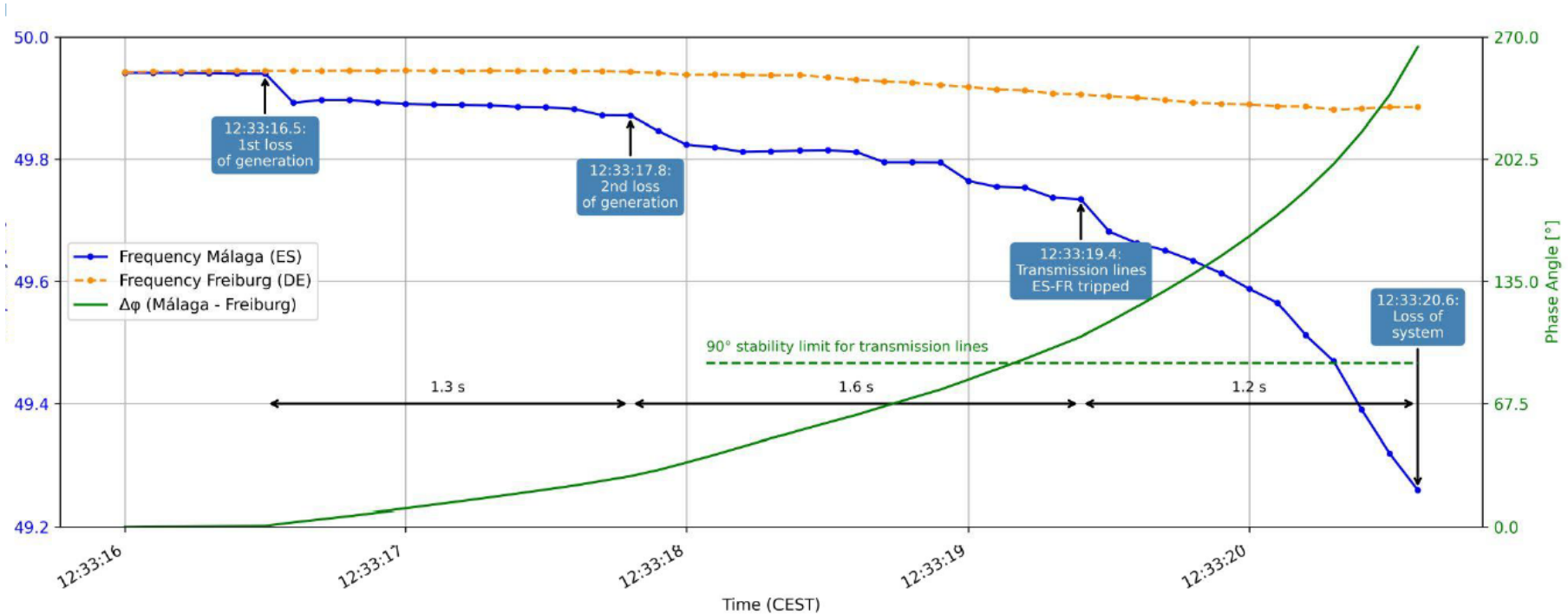
Houston, Texas 07 Feb 2021



Houston, Texas 16 Feb 2021

- Damages from the blackouts were estimated at **\$195 billion [3]**
- **Seconds away** from a total power blackout in Texas

Power blackout 28 April 2025 Spain/Portugal

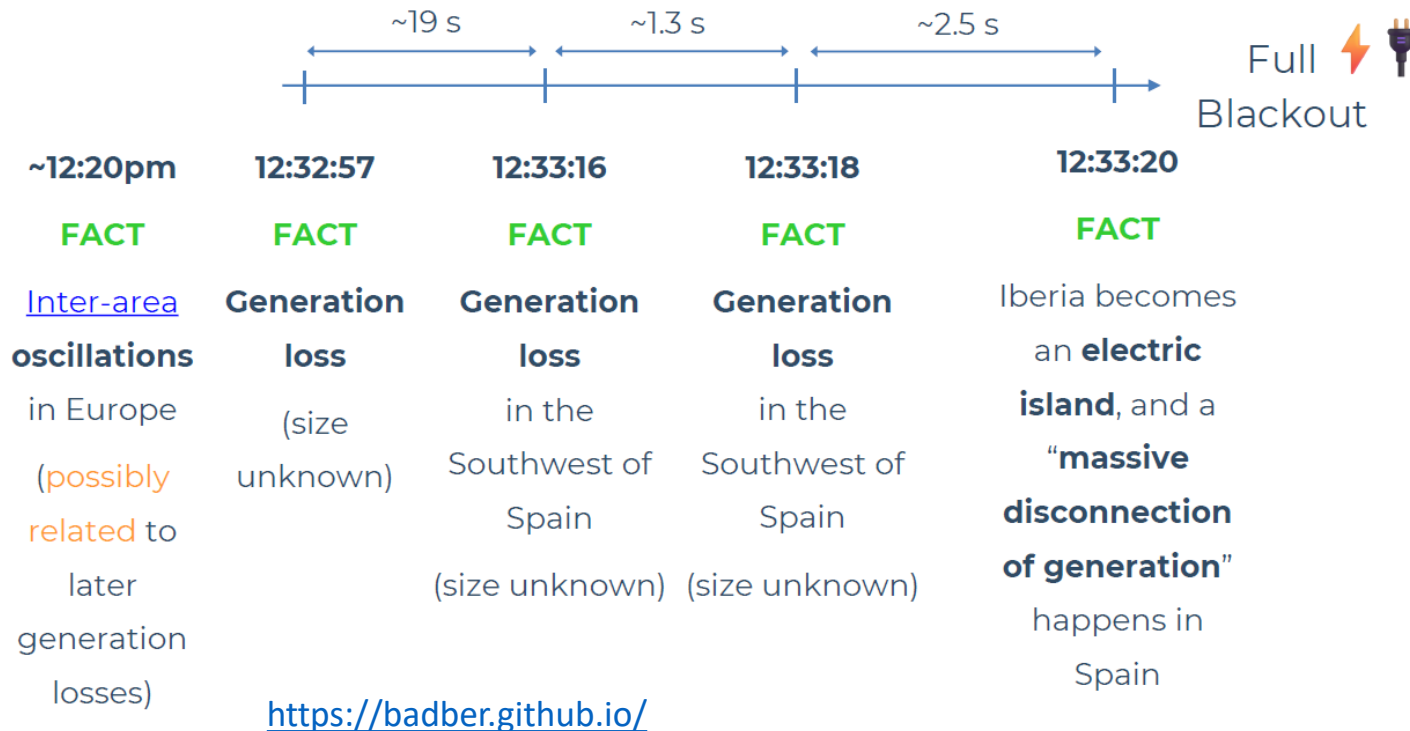


N-2?



Luis Badesa

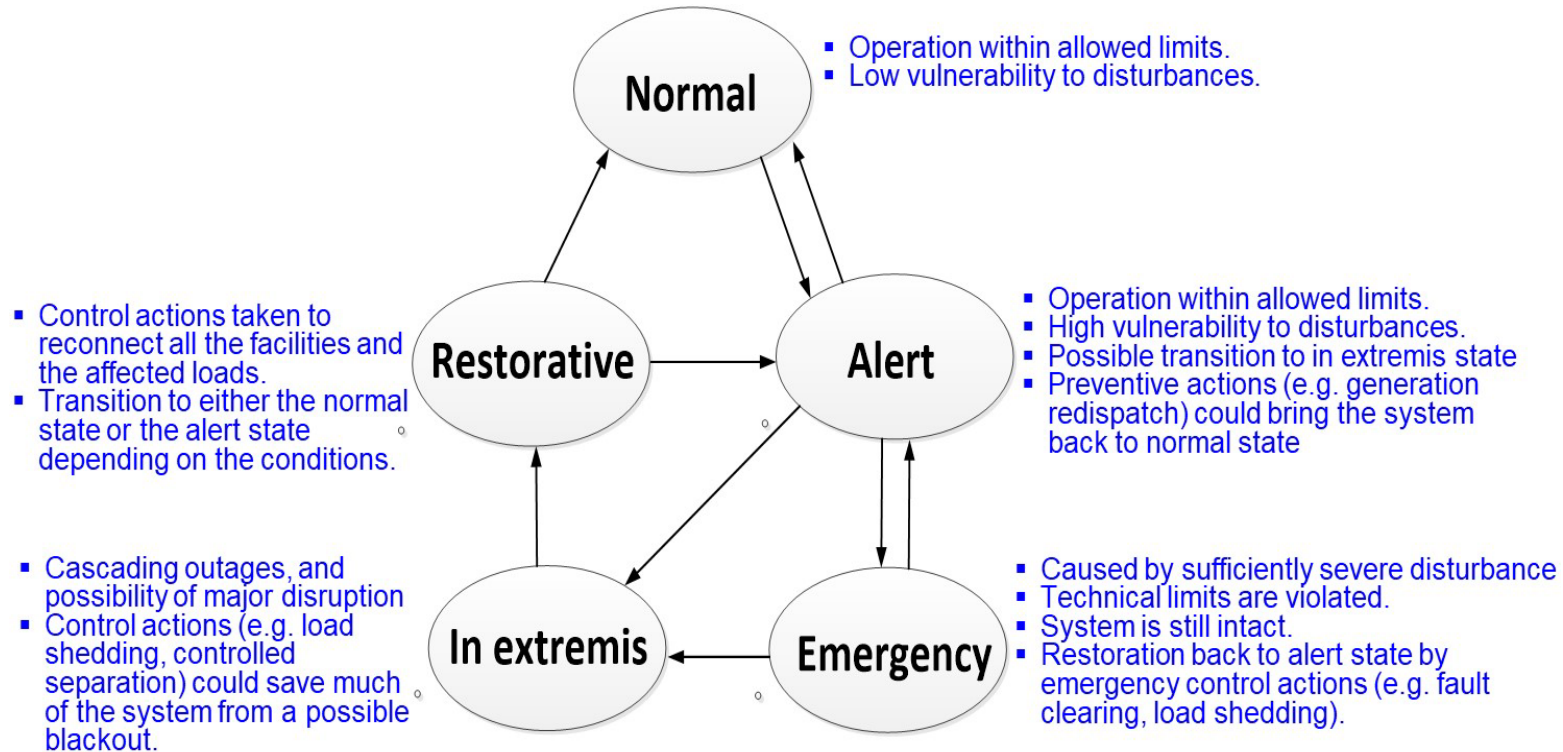
▶ 28th of April 2025 (CET)



Power system reliability

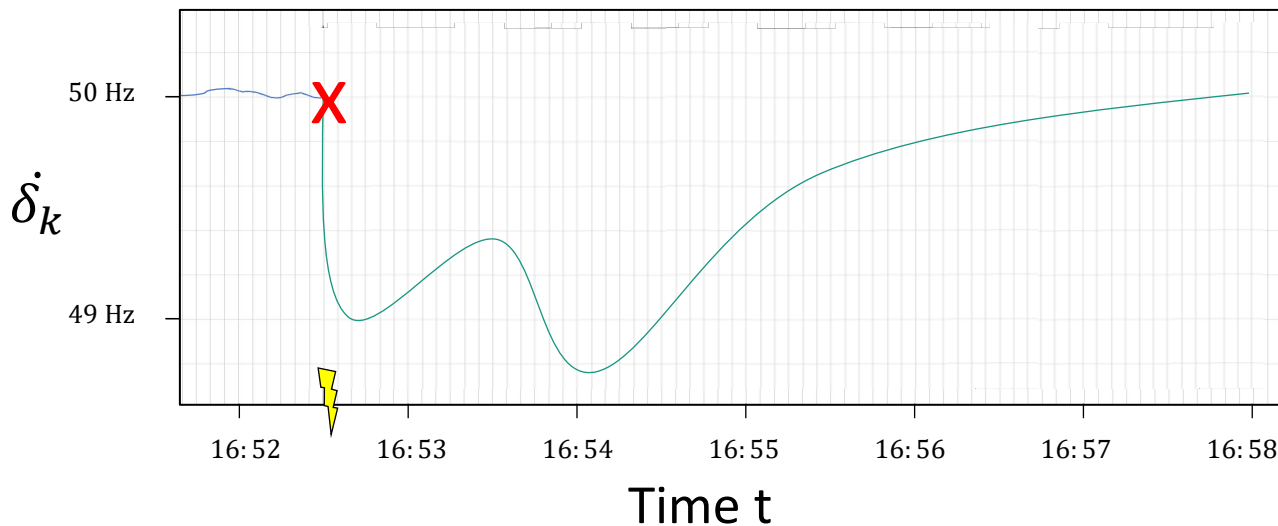
“...is the probability that an electrical power system can perform a required function under given conditions for a given time interval.”

Operating states of power systems



Conventional (offline) dynamic security assessment

Simulating time-response



Numerical integration

ODE system
$$\begin{cases} \dot{x} = f(x, t, x_0) \\ x_0 = (P_k^{16h}, Q_k^{16h}) \end{cases}$$

Forward Euler

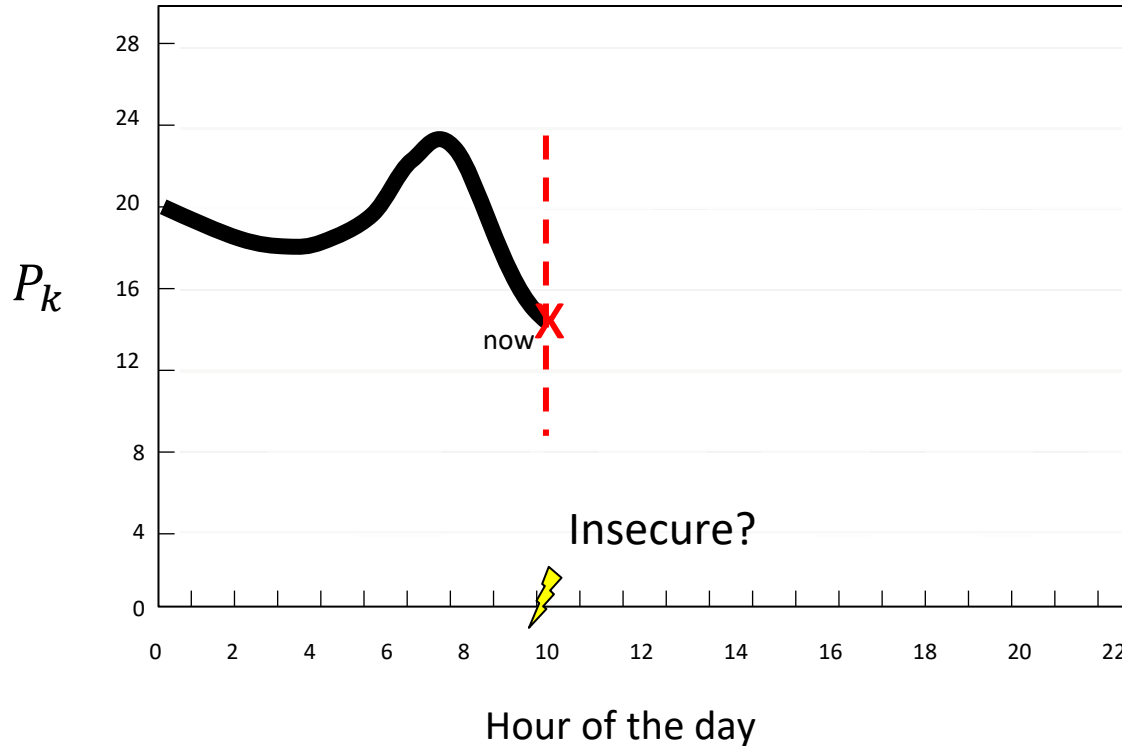
$$x_{k+1} = x_k + hf(x_k, t_k)$$

slow for large systems

Real-time dynamic security assessment

Objective: predicting security in real-time

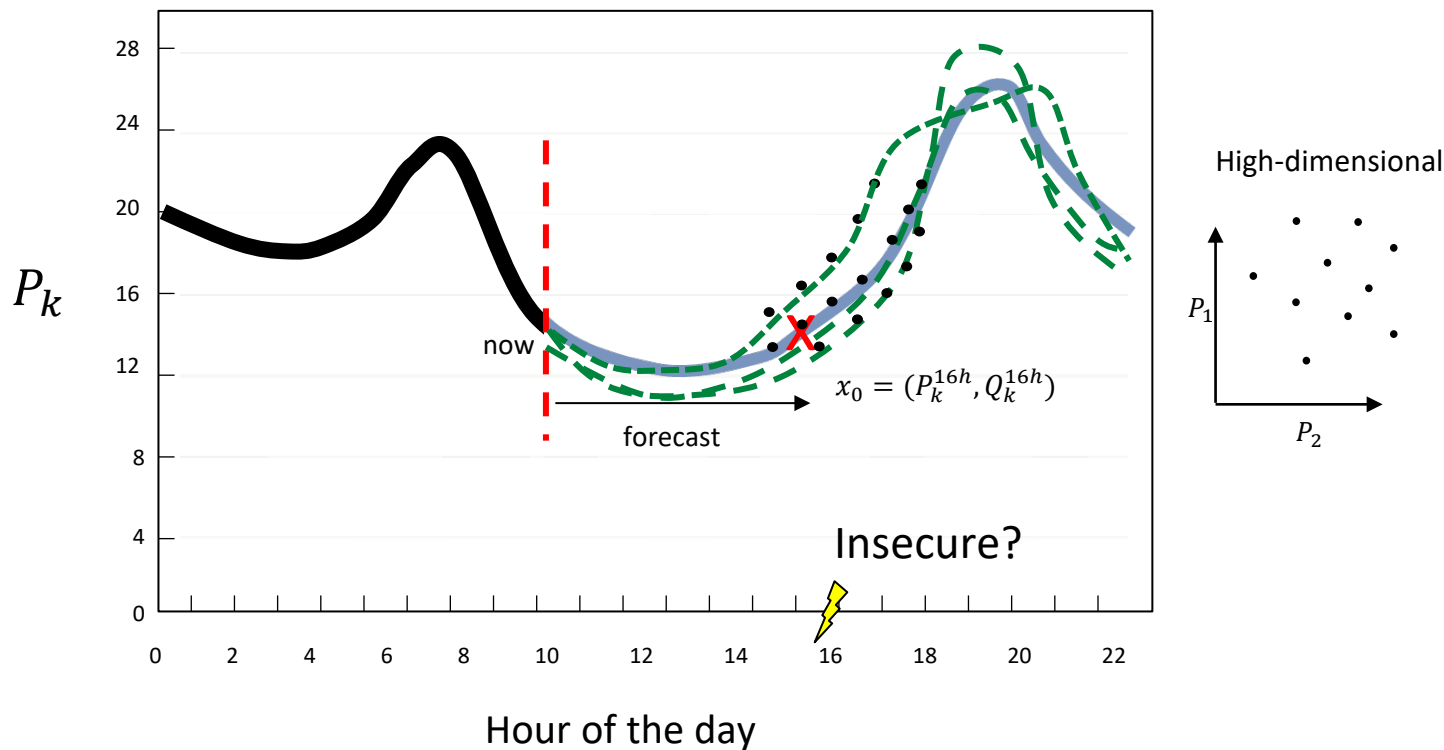
In response, use corrective actions in (near) real-time



(Preventive) real-time dynamic security assessment

For N-1 security

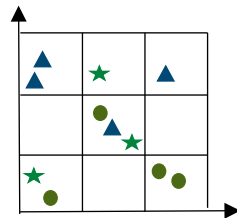
Preventive actions



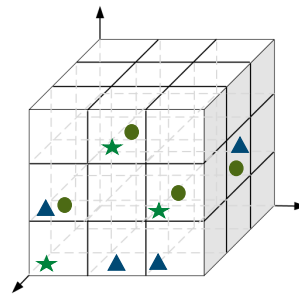
Curse of Dimensionality



1d: 3 regions



2d: 3^2 regions



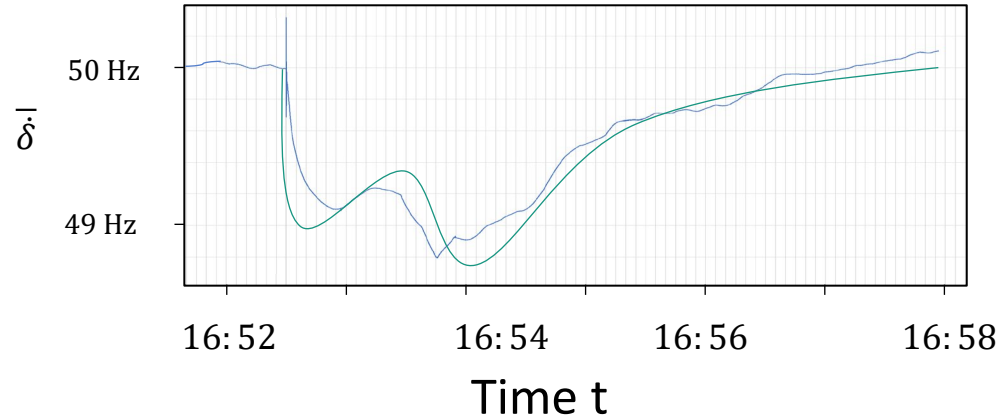
3d: 3^3 regions



1000d: hopeless

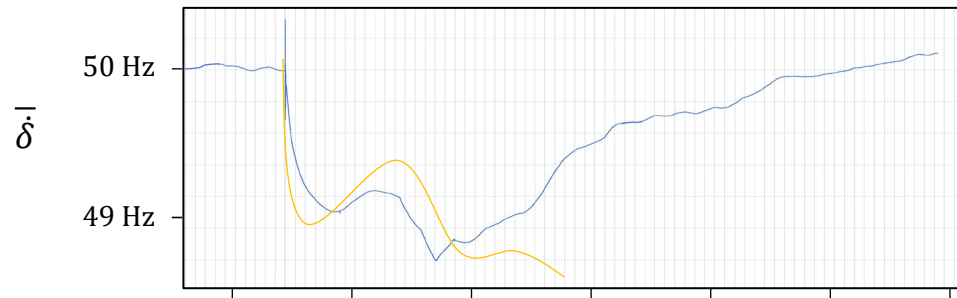
As dimensionality grows: fewer samples per region.

Security of power systems



secure

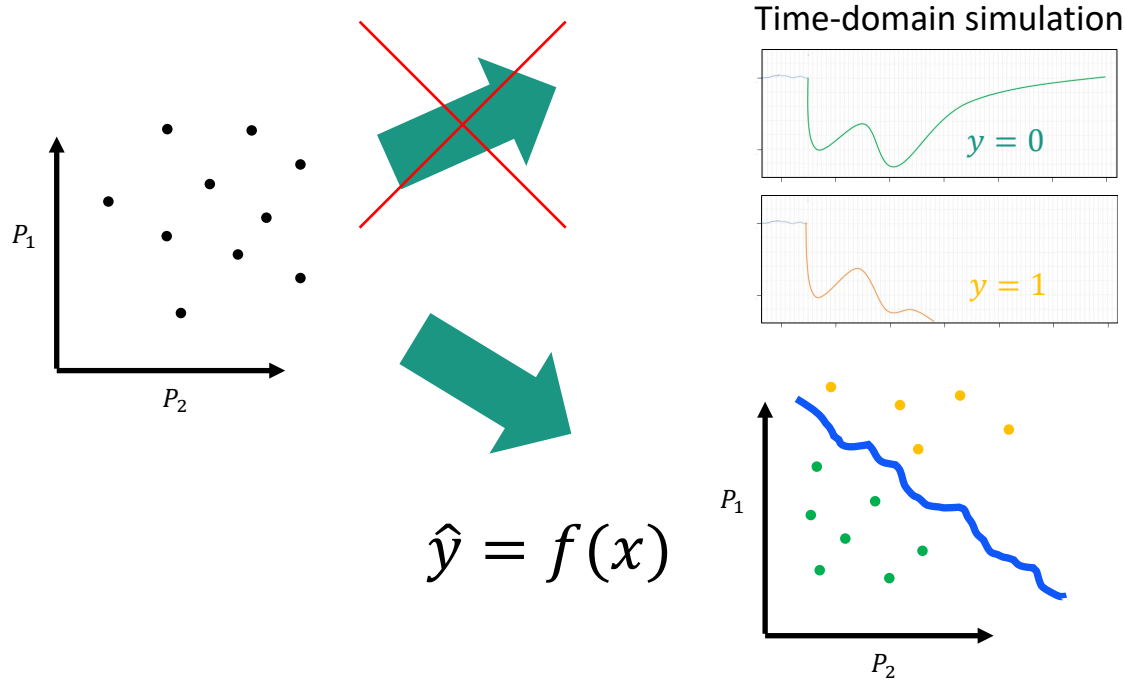
$$y = 0$$



insecure

$$y = 1$$

Machine learning model to predict security



How to train
and use f ?

Challenges for reliability management

- More extreme weather events
- Higher grid load in the system
- Higher uncertainty
- Highly complex problem

Opportunities for reliability management with AI

- Availability of better models and data (weather, grid data, etc)
- New AI techniques
- Once trained, models are quick in 'predicting', but challenges also exist

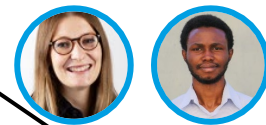
Outline

Reliability management and data in control rooms

1. Introduction to reliability management
2. Machine learning approaches
3. Security assessment with cost-sensitive supervised learning

Learning models for secure system operation

4. Learning with domain knowledge
5. State estimation with graph neural networks
6. Weakly-supervised learning for secure operation
7. Challenges applying ML to reliability



Federica
Bellizio

Olayiwola
Arowolo

Supervised Learning for Surrogate Models

Notation: Power system s , model m , parameter x

Objective: assess $m(x) \rightarrow y$ very fast and often

Surrogate approach

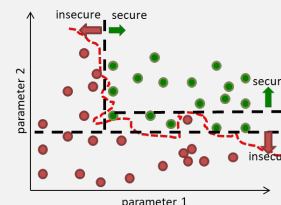
1. Generate a training dataset $\Omega^T = \{(x_i, y_i)\}_{i=1}^N$ where $y_i = m(x_i)$ from the full simulator
2. Train surrogate $f(x) \rightarrow \hat{y}$ with supervised loss $\sum_{i \in \Omega^T} \|y_i - \hat{y}_i\|$
3. Use $f(x_j)$ for new $j \notin \Omega^T$

Benefit: speed at inference

Applications

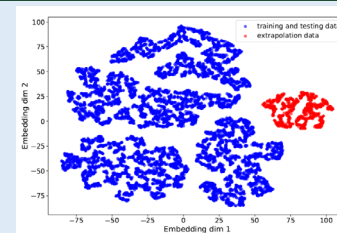
- Real-time dynamic security assessment ([8,9] and many others)

Two-dimensional example



Challenges

- What if s and m changes? e.g., topology changes
- What if the model is inaccurate $s \neq m$? e.g., inverter-based controls
- Need large, representative training data



Physics-Informed Learning

Objective: surrogate learning enhanced with physics knowledge from model m

Idea: Incorporate physics residual (e.g. from a PDE or simulator) to guide learning and improve generalization

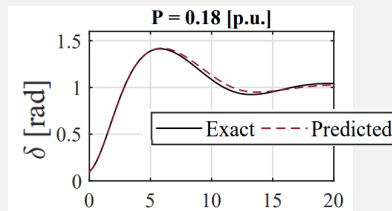
Physics-informed approach

1. Generate offline training dataset $\Omega^T = \{(x_i, y_i)\}_{i=1}^N$ with $y_i = m(x_i)$
2. Train surrogate $f(x) \rightarrow \hat{y}$ on composite loss $\sum_{i \in \Omega^T} \|y_i - \hat{y}_i\| + \mathcal{L}_{phys}(f(x_i), m)$
3. Use $f(x_j)$ for new $j \notin \Omega^T$

Benefits: Better generalisation performance with **fewer training samples**

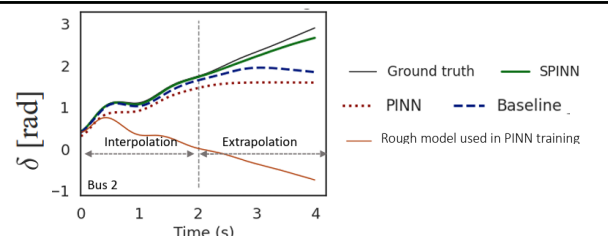
Applications

- Extrapolation in time-domain for dynamic analysis in power systems



Challenges

- Model inaccuracy $s \neq m$
- **Changes in s or m**
- Data sparsity
- Multi-loss scaling causes training instability
- Scaling issues to many physical loss terms in power systems



Weakly-Supervised (E2E) Learning

Objective: learn models $f(x)$ for downstream task even when exact labels $y_i = m(x_i)$ from the simulator m are unavailable, uncertain, or only indirectly defined.

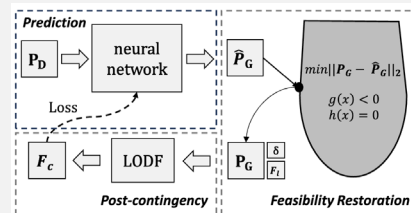
Approach

1. Generate many inputs $\Omega^T = \{(x_i)\}_{i=1}^N$
2. Model task loss $\sum_{i \in \Omega^T} \mathcal{L}(m(f(x_i)))$
3. Use $f(x_j)$ for new $j \notin \Omega^T$

Benefits: learning for computationally expensive or ill-defined problems

Applications

- Learn to predict effective inputs to OPF [13]
- Replace conventional solvers with NN [14]
- Distribution system state estimation [15]
- N-k security-constrained OPF [16]



Challenges

- Inexact supervision $s \neq m$ not so important as success defined by task-loss
- **System shift in s or m**
- Data coverage. Diverse samples are needed for generalization

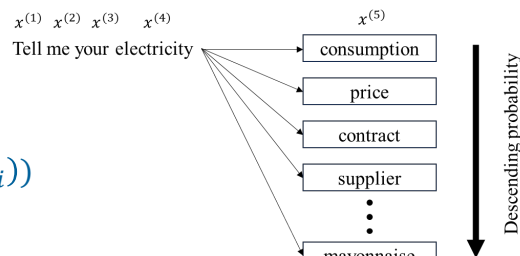
Self-Supervised Learning

Objective: Learn a **useful internal representation** from unlabeled data by solving a **pretext task** — no human-labeled or simulator-labeled outputs required.

Idea: instead of training on (x_i, y_i) train on auto-generated pseudo-labels or tasks constructed from structure x_i

Approach

1. Generate many inputs $\Omega^T = \{(x_i)\}_{i=1}^N$
2. Define self-supervised pretext loss $\mathcal{L}_{pretext}(f(x_i))$
3. Train encoder $\sum_{i \in \Omega^T} \mathcal{L}_{pretext}(f(x_i))$
4. Use $f(x)$ for downstream *task* (e.g. forecasting, OPF, estimation)



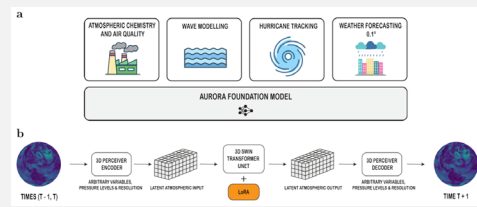
Benefits: Good initialization when little data, good transfer to downstream tasks

Challenges

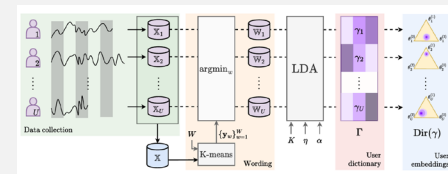
- Design pretext loss and model architectures with broad set of tasks, grid conditions, topologies
- Generate large data sets
- ...

Applications

- Natural Language Processing
- Weather foundational models
- Earth system foundational models [17]



Load forecasting of users [18]



Grid foundation models (GFM) [19]

[17] Bodnar, C., Bruinsma, W. P., Lucic, A., Stanley, M., Vaughan, A., Brandstetter, J., ... & Perdikaris, P. (2024). A foundation model for the earth system. *arXiv preprint arXiv:2405.13063*.

[18] Bölät, Kutay, and Simon Tindemans. "GUIDE-VAE: Advancing Data Generation with User Information and Pattern Dictionaries." *arXiv preprint arXiv:2411.03936* (2024).

[19] Hamann, H. F., Gjorgiev, B., Brunschweiler, T., Martins, L. S., Puech, A., Varbella, A., ... & Sobolevsky, S. (2024). Foundation models for the electric power grid. *Joule*, 8(12), 3245-3258.

Graph Neural Networks

Objective: Improve generalization performance in learning tasks on network-structured systems (like power grids)

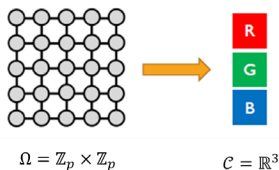
Idea: embedding graph topology directly into the model architecture as bias

Approach

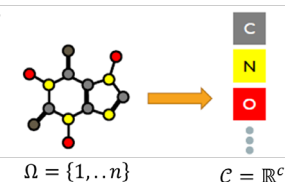
1. Construct graph $G = (V, \mathcal{E})$ with features on nodes and edges
2. Define f_{GNN} and learn with message passing on supervised loss $\sum_{i \in \Omega^T} \|y_i - \hat{y}_i\|$
3. Use $f(x_j)$ for new $j \notin \Omega^T$ or on unseen graphs G'

Benefits: Data efficient, generalisation to changes in topologies

Example: $p \times p$ RGB image



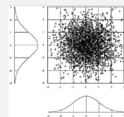
Example: molecular graph



Applications

- Graph neural solvers [20] for ACOPF [21]
- Distribution system state estimation [22]

Noisy measurements



Power flow equations

$$H(x) = \begin{bmatrix} P_1 - P_1(x) \\ Q_1 - Q_1(x) \\ \vdots \\ P_n - P_n(x) \\ Q_n - Q_n(x) \end{bmatrix}$$

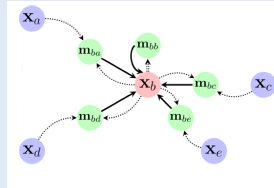
Topology



Challenges

- Model inaccuracy $s \neq m$
- Long-range dependencies are difficult to learn. *Power system topology is sparse*
- Challenging to learn for *global* problems (e.g. ACOPF)

Good to learn local relationships



Outline

Reliability management and data in control rooms

1. Introduction to reliability management
2. Machine learning approaches
3. Security assessment with cost-sensitive supervised learning

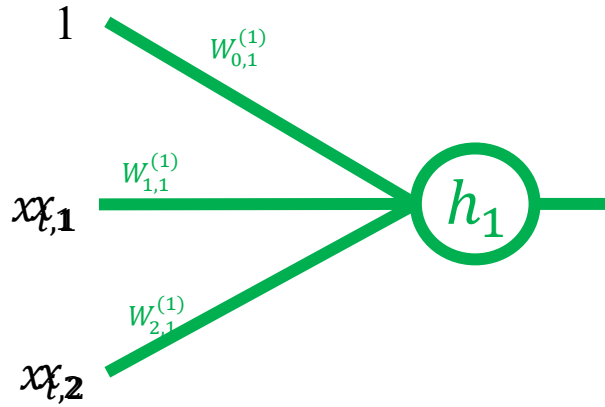
Learning models for secure system operation

4. Learning with domain knowledge
5. State estimation with graph neural networks
6. Weakly-supervised learning for secure operation
7. Challenges applying ML to reliability

Neural networks

One neuron

Here simplified notation
 $x_{i,1}$ to x_1

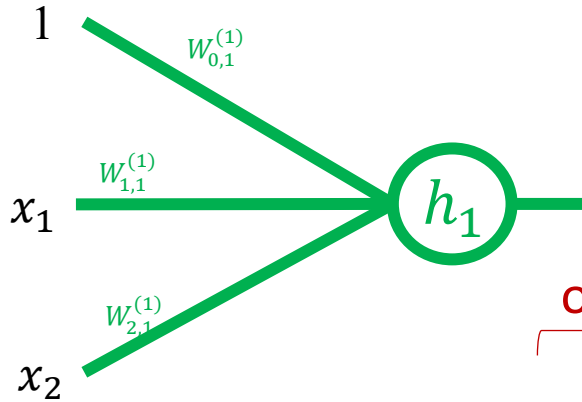


Activation function

$$h_1 = a(w_{0,1}^{(1)} + x_1 w_{1,1}^{(1)} + x_2 w_{2,1}^{(1)})$$

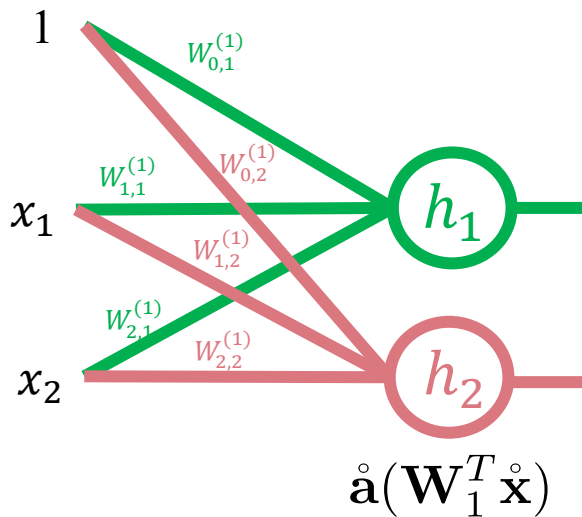
Neural networks

One neuron



$$h_1 = a(\underbrace{w_{0,1}^{(1)}}_{\text{out}_{h1}} + \underbrace{x_1 w_{1,1}^{(1)} + x_2 w_{2,1}^{(1)}}_{\text{in}_{h1}})$$

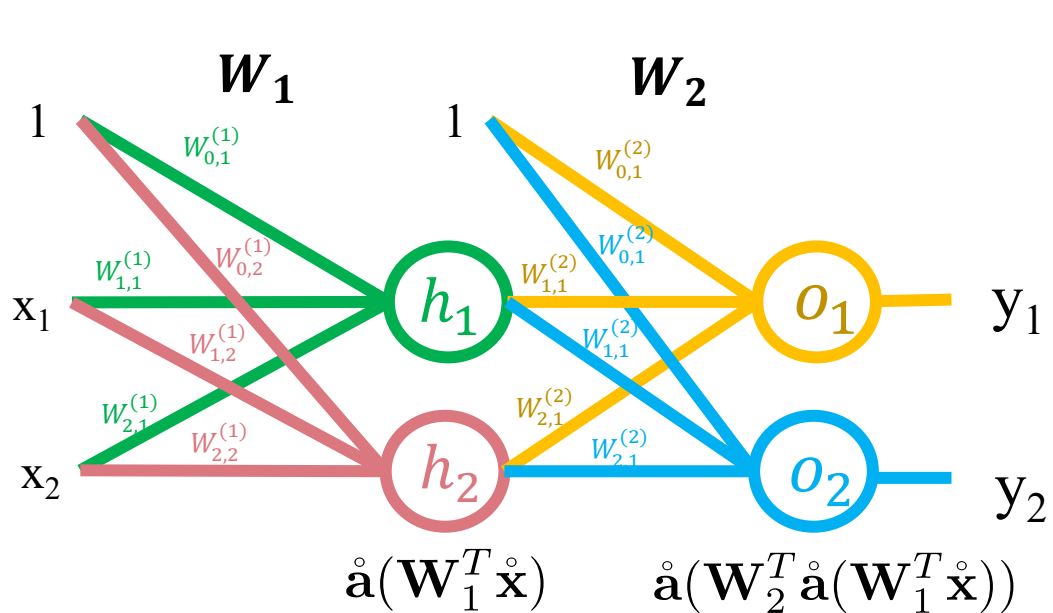
Compact model



$$\mathbf{W}_1 = \begin{bmatrix} W_{0,1}^{(1)} & W_{0,2}^{(1)} \\ W_{1,1}^{(1)} & W_{1,2}^{(1)} \\ W_{2,1}^{(1)} & W_{2,2}^{(1)} \end{bmatrix}$$

$(N + 1) \times u_1$

Multiple layers



$$\mathbf{W}_1 = \begin{bmatrix} W_{0,1}^{(1)} & W_{0,2}^{(1)} \\ W_{1,1}^{(1)} & W_{1,2}^{(1)} \\ W_{2,1}^{(1)} & W_{2,2}^{(1)} \end{bmatrix}$$

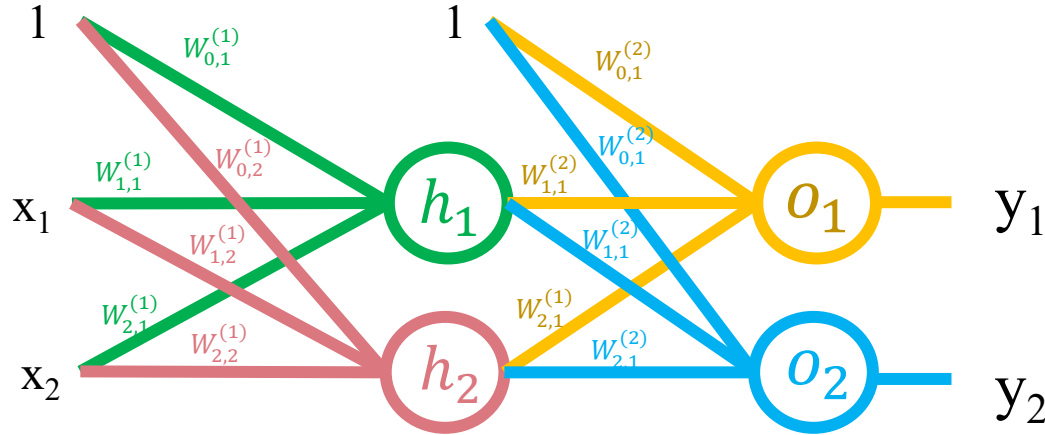
$(N + 1) \times u_1$

$$\mathbf{W}_2 = \begin{bmatrix} W_{0,1}^{(2)} & W_{0,2}^{(2)} \\ W_{1,1}^{(2)} & W_{1,2}^{(2)} \\ W_{2,1}^{(2)} & W_{2,2}^{(2)} \end{bmatrix}$$

$(u_1 + 1) \times u_2$

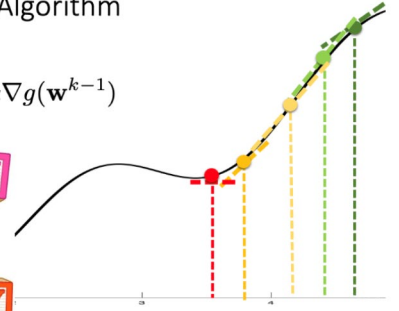
$$f(x) = \mathring{a}(\mathbf{W}_2^T \mathring{a}(\mathbf{W}_1^T \mathring{\mathbf{x}}))$$

Loss function



Gradient Descent Algorithm

$$\mathbf{w}^k = \mathbf{w}^{k-1} - \alpha \nabla g(\mathbf{w}^{k-1})$$



$$J = \frac{1}{|\Omega^T|} \sum_{i \in \Omega^T} \left((o_{i,1} - y_{i,1})^2 + (o_{i,2} - y_{i,2})^2 \right)$$

System operation



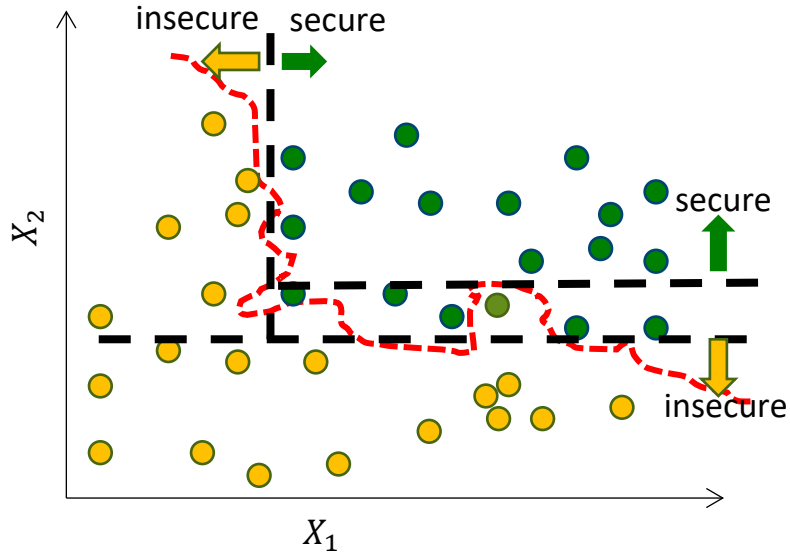
Experts are in charge to manually operate the power system based on **experience** and with **the support of tools**



Interpretable models

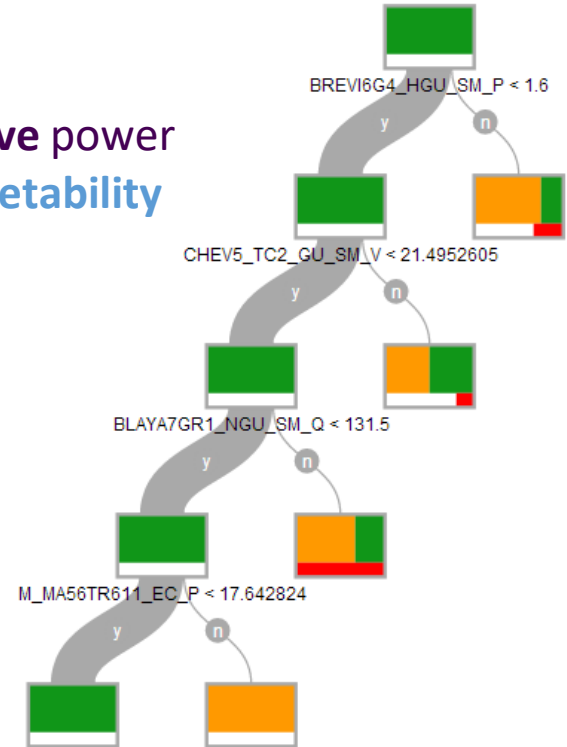
Decision Trees as a model?

Two-dimensional example

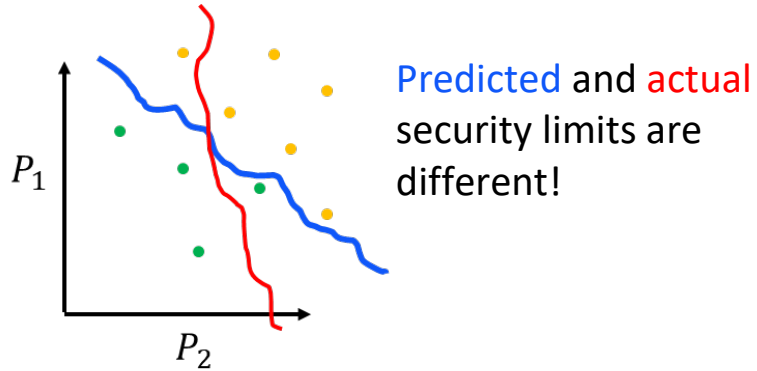


Decision trees:

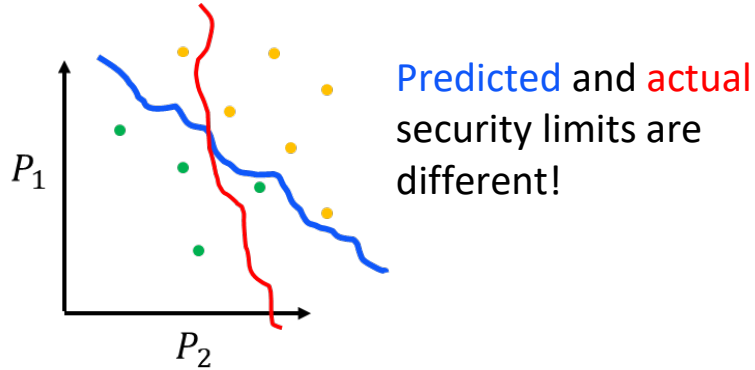
- Limited **expressive** power
- Fantastic **interpretability**



Metrics for classification



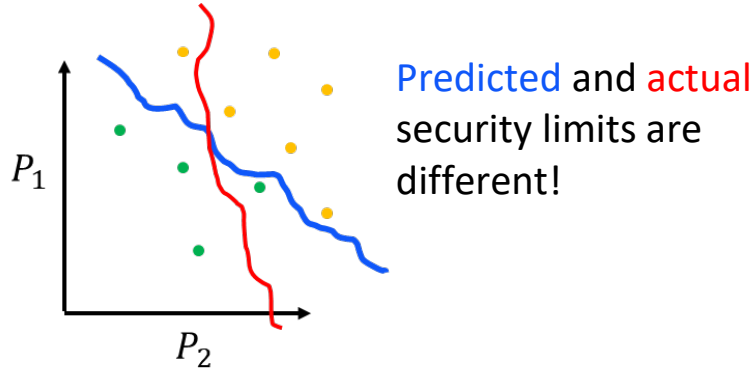
Metrics for classification



		Predicted Class	
		Positive	Negative
True Class	Positive	TP	FN
	Negative	FP	TN

Confusion Matrix

Metrics for classification



Two types of accurate predictions:

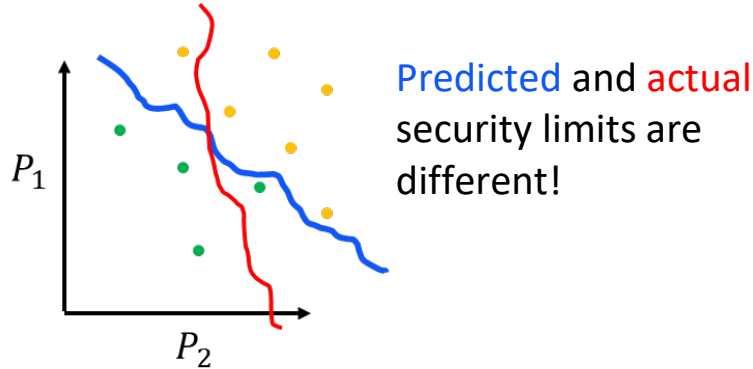
TN: Is secure and we think it is secure (GOOD)

TP: Is insecure and we think it is insecure (VERY GOOD!)

		Predicted Class	
		Positive	Negative
True Class	Positive	TP	FN
	Negative	FP	TN

Confusion Matrix

Metrics for classification



Two types of accurate predictions:

TN: Is secure and we think it is secure (**GOOD**)

TP: Is insecure and we think it is insecure (**VERY GOOD!**)

Two types of wrong predictions:

FP: Is secure but we think it is insecure (**BAD**)

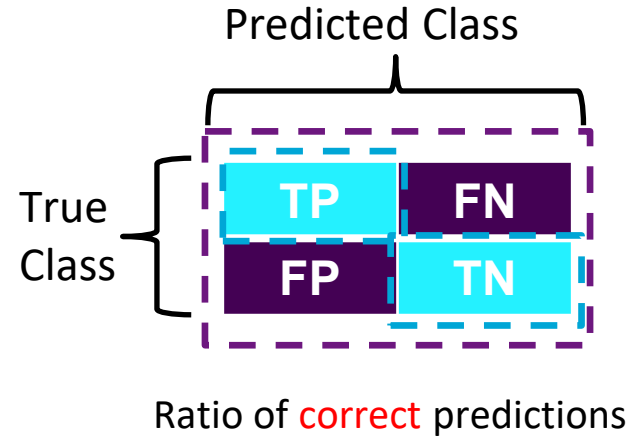
FN: Is insecure but we think it is secure (**VERY BAD!**)

		Predicted Class	
		Positive	Negative
True Class	Positive	TP	FN
	Negative	FP	TN

Confusion Matrix

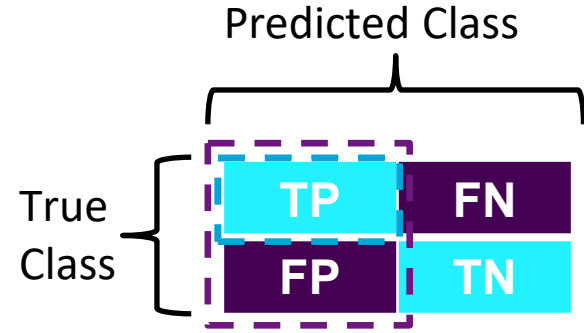
This can have a severe effect!

Metrics for classification



$$\text{Accuracy} = \frac{N_{TP} + N_{TN}}{N_{FP} + N_{TP} + N_{FN} + N_{TN}}$$

Metrics for classification

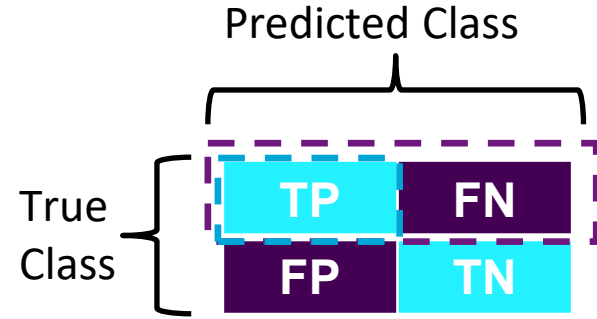


Ratio of correctly found **insecure** cases to **predicted insecure** predictions

$$\text{Accuracy} = \frac{N_{TP} + N_{TN}}{N_{FP} + N_{TP} + N_{FN} + N_{TN}}$$

$$\text{Precision} = \frac{N_{TP}}{N_{TP} + N_{FP}}$$

Metrics for classification



Ratio of correctly found **insecure** cases to **all insecure** cases

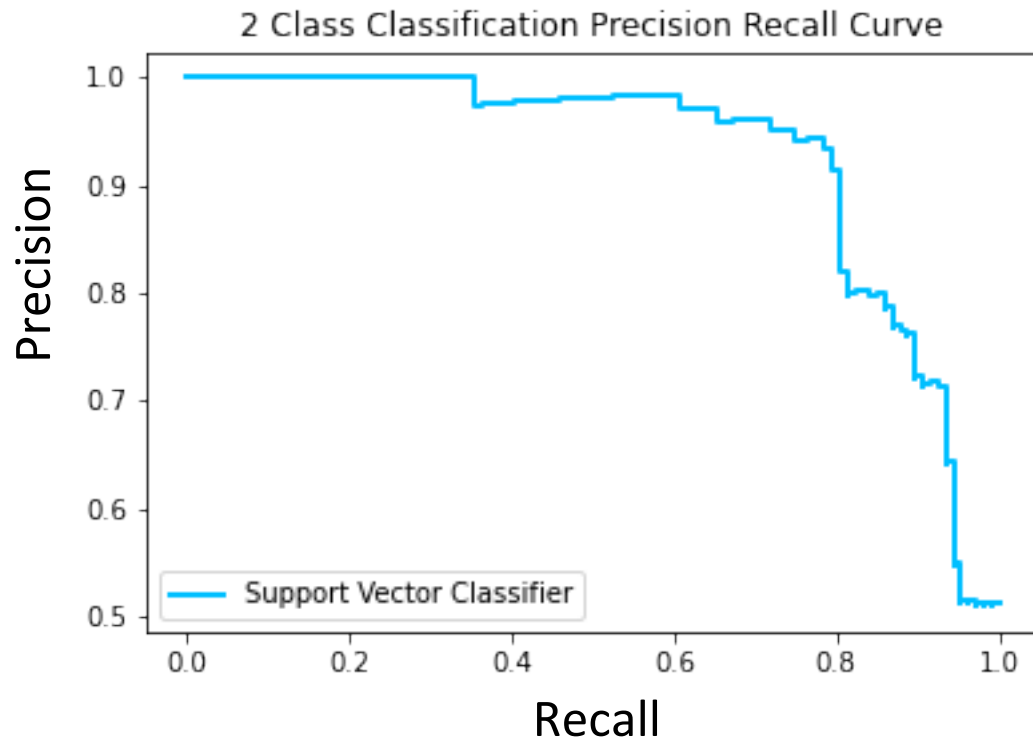
$$\text{Accuracy} = \frac{N_{TP} + N_{TN}}{N_{FP} + N_{TP} + N_{FN} + N_{TN}}$$

$$\text{Precision} = \frac{N_{TP}}{N_{TP} + N_{FP}}$$

$$\text{Recall} = \frac{N_{TP}}{N_{TP} + N_{FN}}$$

Precision vs Recall

When do we observe the highest performance?



Blackout predictions: Precision or Recall?



Houston, Texas 07 Feb 2021



Houston, Texas 16 Feb 2021

$$\text{Precision} = \frac{N_{TP}}{N_{TP} + N_{FP}}$$

$$\text{Recall} = \frac{N_{TP}}{N_{TP} + N_{FN}}$$

Cost skewness: $C_{FN} \gg C_{FP}$

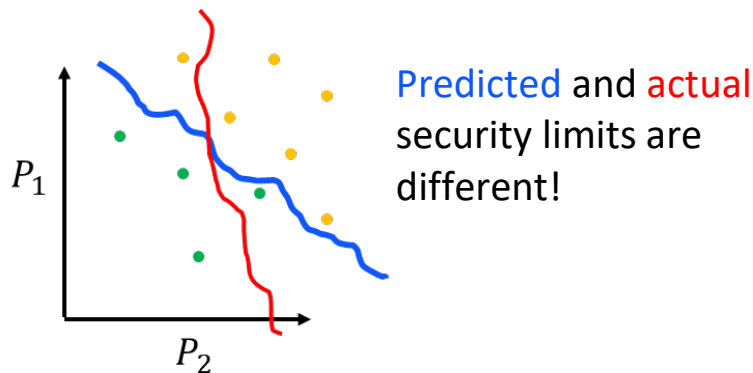
Problem

- The two different false predictions have different costs.



- Damages from the blackouts were estimated at \$195 billion
- Seconds away from a total power blackout in Texas

Metrics for classification



Two types of accurate predictions:

TN: Is secure and we think it is secure (**GOOD**)

TP: Is insecure and we think it is insecure (**VERY GOOD!**)

Two types of wrong predictions:

FP: Is secure but we think it is insecure (**BAD**)

FN: Is insecure but we think it is secure (**VERY BAD!**)

		Predicted Class	
		Positive	Negative
True Class	Positive	0	C_{FN}
	Negative	C_{FP}	0

Two issues

- Cost-skewness: $C_{FN} \gg C_{FP}$
- Class imbalance: $\pi_+ \ll \pi_-$

What a classifier can do

Classify points

- is x positive?

Rank points

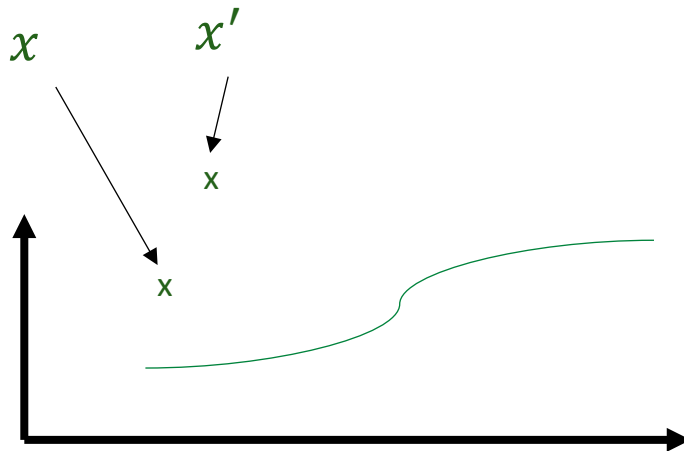
- Is x 'more positive' than x' ?

Output a score $s(x)$

- 'How positive' is x ?

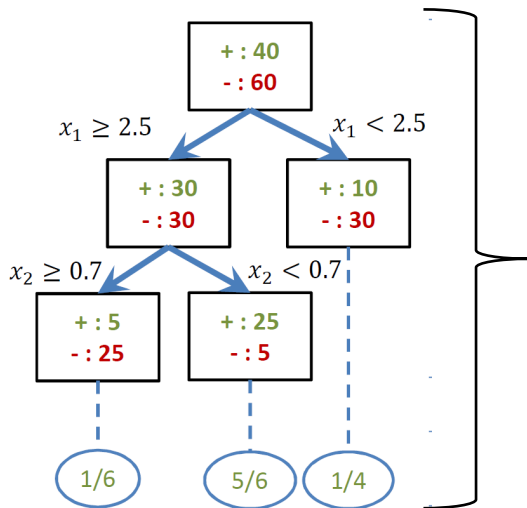
Output a probability estimate $\hat{p}(x)$

- What is the (estimated) probability that x is positive?



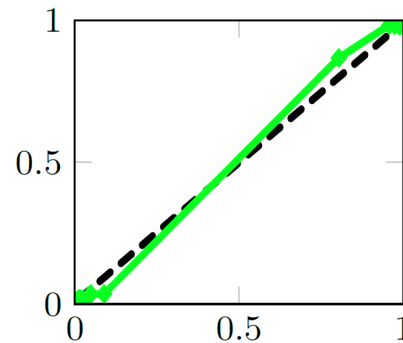
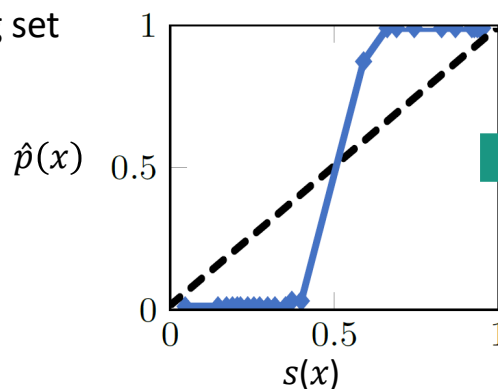
Probability estimation is not easy

- Scores $s(x) \in [0,1]$ as probability estimates $\hat{p}(x)$? **No!**

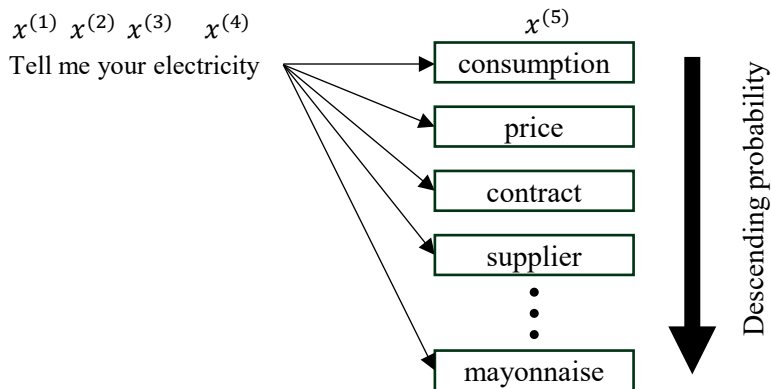


Based on
training set

Platt scaling: Find $\hat{p}(x) = \frac{1}{1+e^{As(x)+B}}$



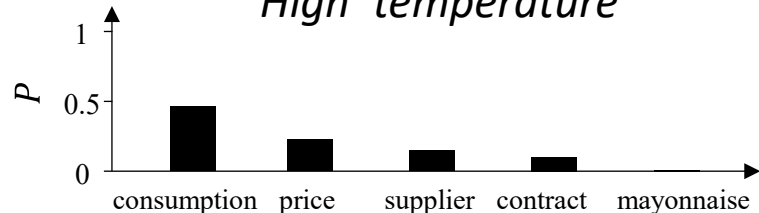
Calibration in Large Language Models



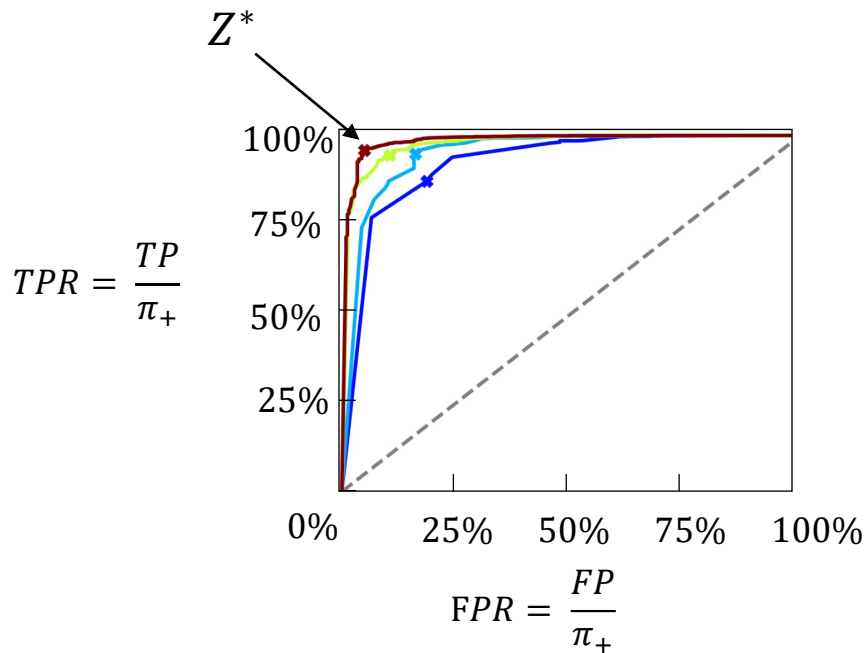
Low 'temperature'



High 'temperature'



Cost-sensitive learning



$$Z^* = \frac{\Pi_- C_{FP}}{\Pi_- C_{FP} + \Pi_+ C_{FN}}$$

↑ Probability of contingency ↑ 'Impact' of contingency

$\hat{p}(x) \geq Z^*$  predict secure
 $\hat{p}(x) < Z^*$  predict insecure

The risk of relying on machine learning

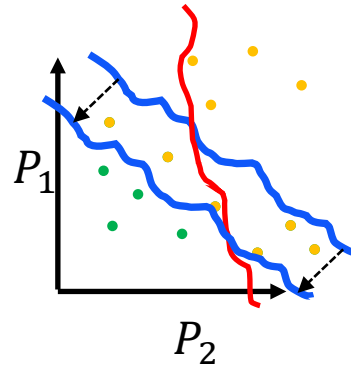
Step 1: Compute risks when predicting x_i as secure $\hat{y}_i = 1$ and insecure $\hat{y}_i = 0$

$$R_{\text{secure}} = p_i p_c \hat{p}(x_i) C_{FN}$$

$$R_{\text{insecure}} = p_i (1 - p_c) (1 - \hat{p}(x_i)) C_{FP}$$

Step 2: Predict with lowest residual risk

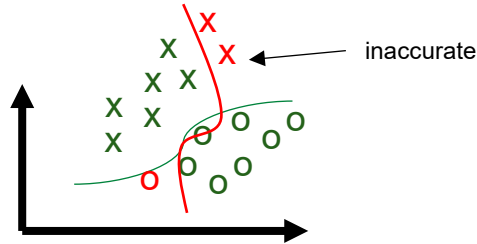
$$R_{\text{secure}} \vee R_{\text{insecure}}$$



Minimize risks by hybrid approach

Machine Learning

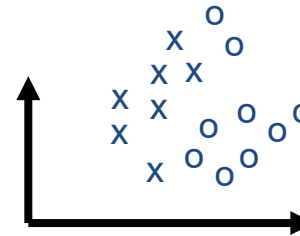
- Fast
- Sometimes inaccurate



+

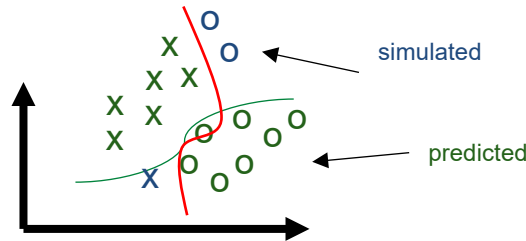
Simulator

- Slow
- Always accurate

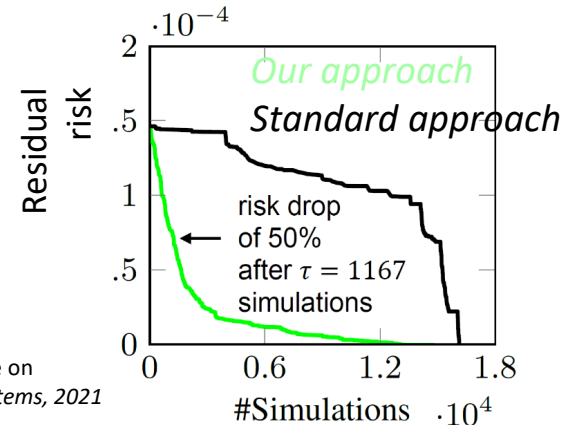


Probabilistic approach

=



Case study: French system



Outline

Reliability management and data in control rooms

1. Introduction to reliability management
2. Machine learning approaches
3. Security assessment with cost-sensitive supervised learning

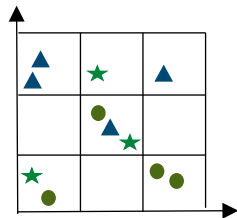
Learning models for secure system operation

4. Learning with domain knowledge
5. State estimation with graph neural networks
6. Weakly-supervised learning for secure operation
7. Challenges applying ML to reliability

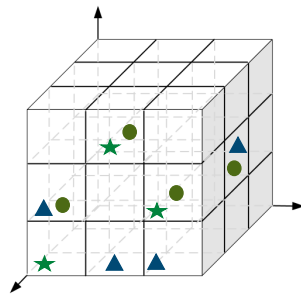
How to address curse of dimensionality?



1d: 3 regions



2d: 3^2 regions



3d: 3^3 regions



1000d: hopeless

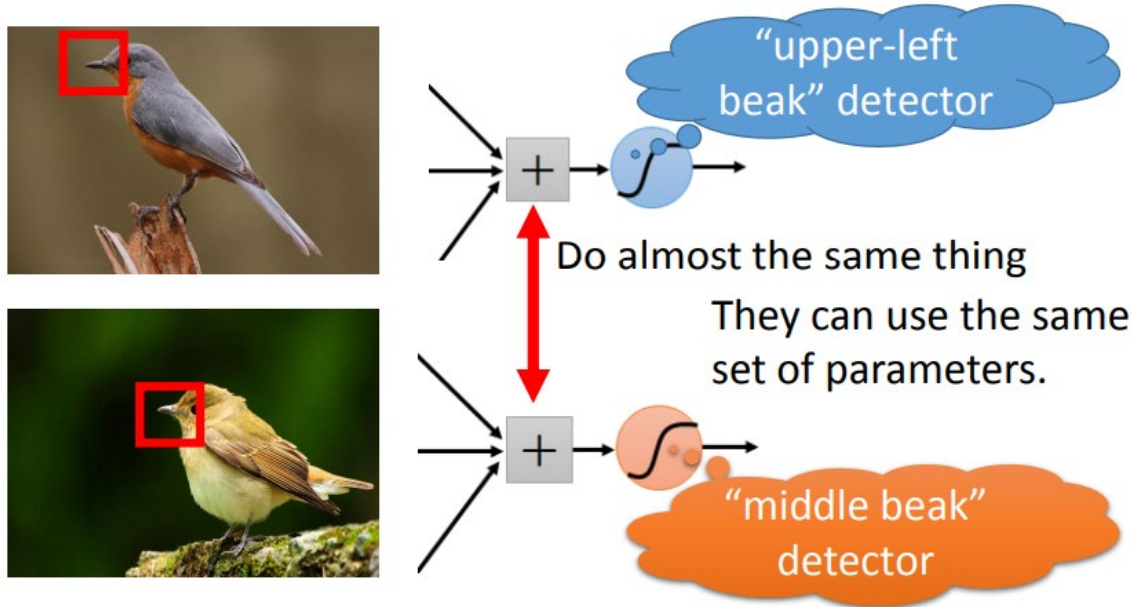
As dimensionality grows: fewer samples per region.

AI can well predict with images

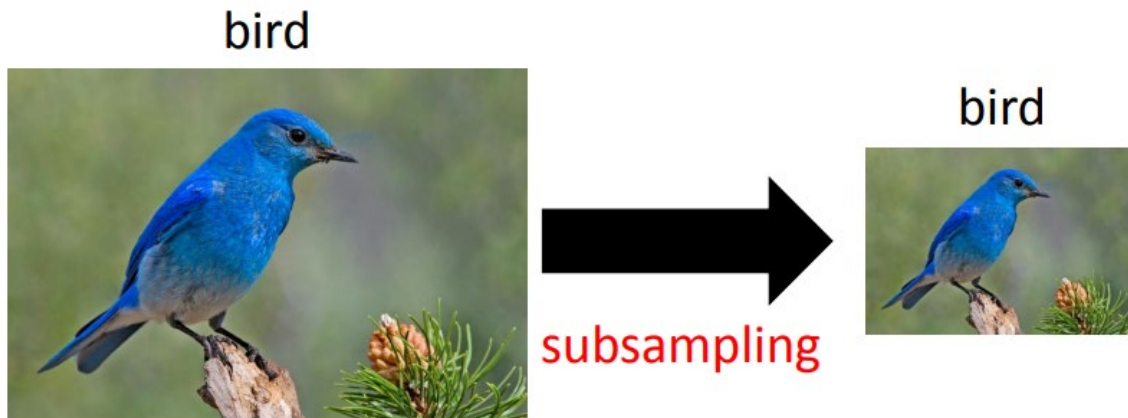
Property 1: Some patterns are much smaller than the whole image. A neuron does not have to see the whole image to discover the pattern.



Property 2: The same patterns appear in different regions. (translated invariance)



Property 3: Subsampling the pixels will not change the object. (Subsampling invariance)

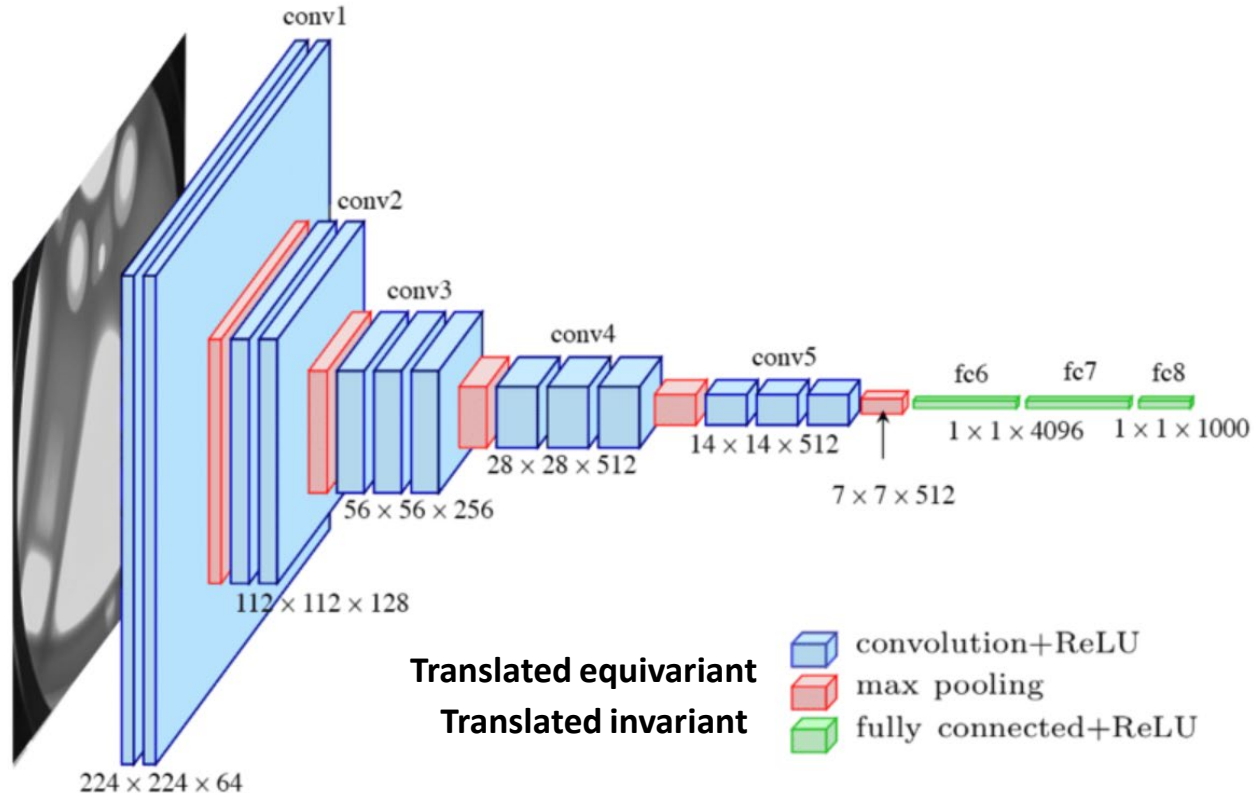


We can subsample the pixels to make image smaller

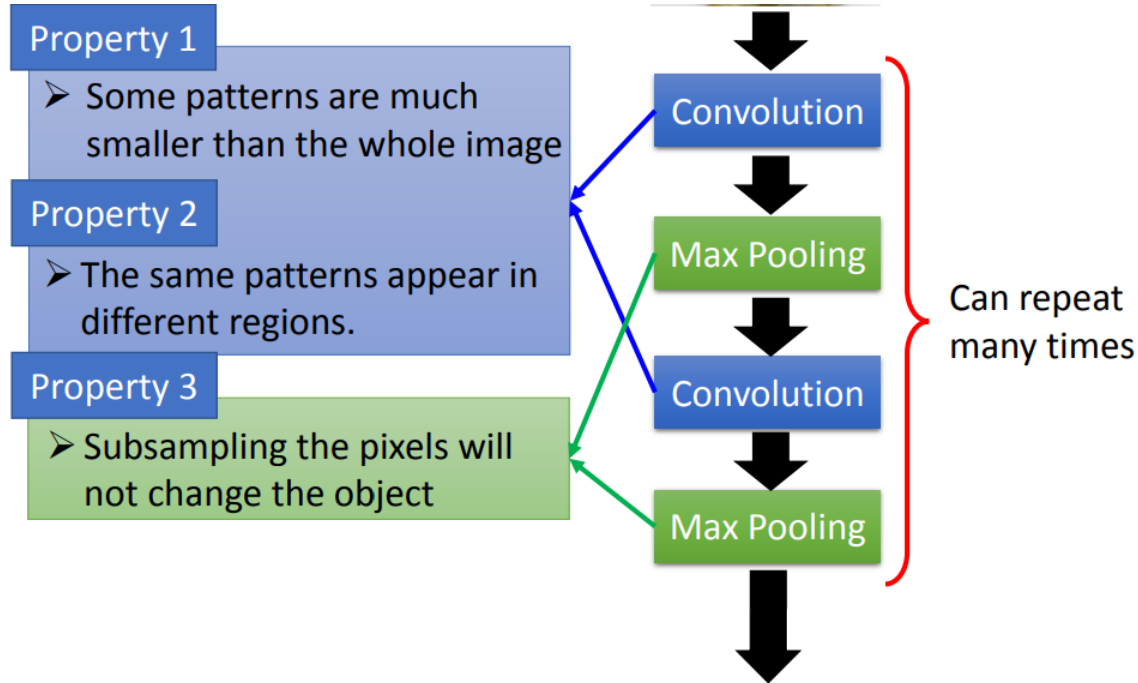


Less parameters for the network to process the image

How can CNN make this happen?



How can CNN make this happen?



CNN— Convolution layer

Stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
0	0	0	0	1	0
1	1	1	0	1	0
0	0	0	0	1	0

6×6 image

Those are the network parameters to be learned.

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1
Matrix

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2
Matrix



3	-1	-3	-1
-2	2	-1	-3
-2	-4	0	1
-1	0	-2	-1

CNN— Convolution layer

Stride=1

0	1	0	0	0	1
0	1	0	0	1	0
0	0	0	1	0	0
0	0	0	1	0	0
1	1	1	0	1	0
0	0	0	0	0	1

6×6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1
Matrix



0	1	-2	-1
-1	1	-1	-3
-1	-4	0	-1
-1	-1	-3	3

Property 2

The same patterns appear in different regions can be detected.

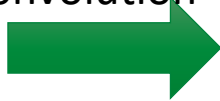
CNN— Max Pooling

Property 3

Subsampling the pixels will not change the object

6×6 image

convolution



3	-1	-3	-1
-2	2	-1	-3
-2	-4	0	1
-1	0	-2	-1

0	1	-2	-1
-1	1	-1	-3
-1	-4	0	-1
-1	-1	-3	3

Max pooling



3	-1
0	1

New images but smaller

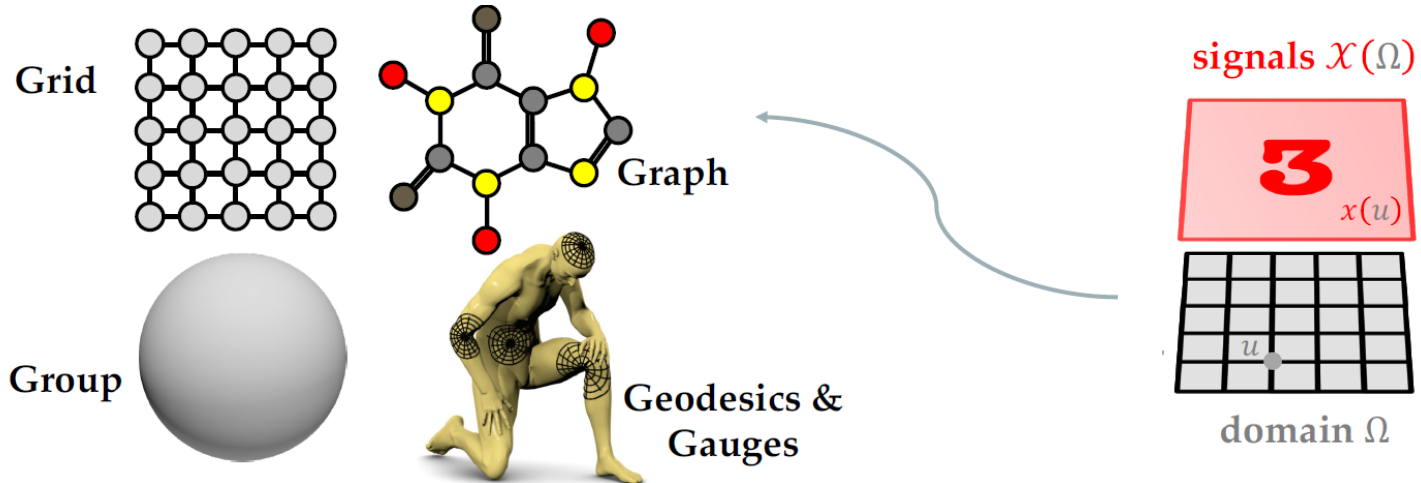
1	-1
-1	3

**Not much image-like data in power system
operation and planning...**

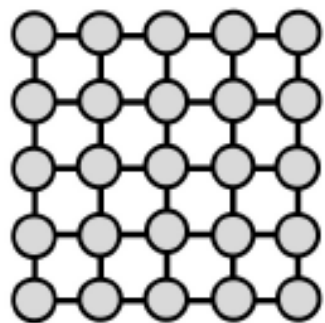
Geometric deep learning (GDL)

In GDL, data are **signals** x on **geometric domains** Ω

- The domain Ω is a set, possibly with additional structure
- A signal x on Ω is a function $\chi(\Omega, C) = \{x: \Omega \rightarrow C\}$
- C is a vector space whose dimensions are called channels



Graph Neural Networks (GNNs)

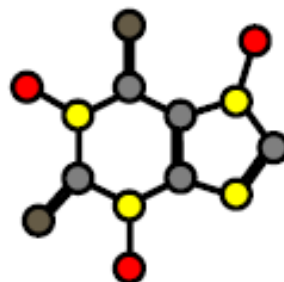


$$\Omega = \mathbb{Z}_n \times \mathbb{Z}_n$$



$$\mathcal{C} = \mathbb{R}^3$$

Example: $n \times n$ RGB image



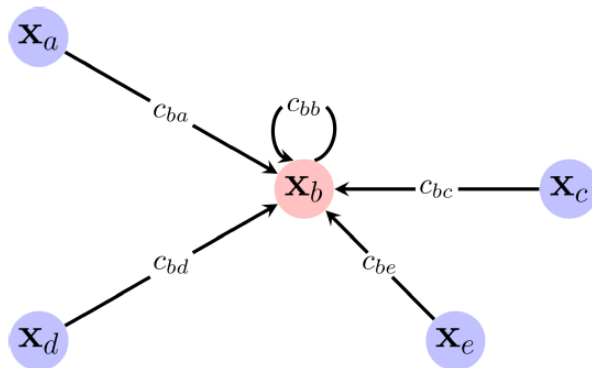
$$\Omega = \{1, \dots, n\}$$



$$\mathcal{C} = \mathbb{R}^m$$

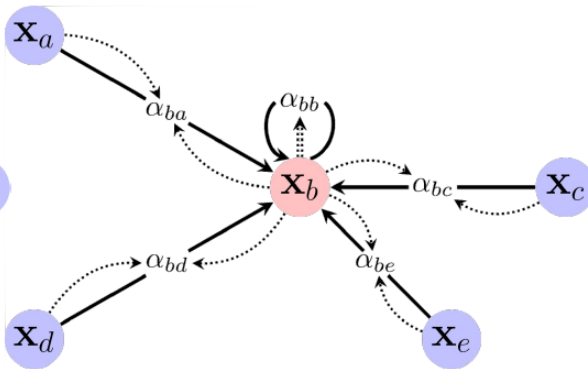
Example: molecular graph

The three 'flavours' of GNN layers



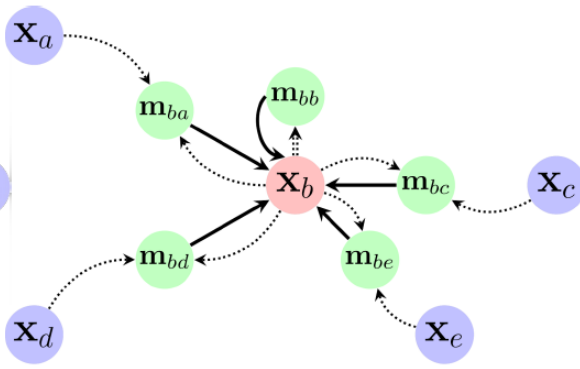
Convolutional

$$\mathbf{h}_i = \phi \left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij} \psi(\mathbf{x}_j) \right)$$



Attentional

$$\mathbf{h}_i = \phi \left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} a(\mathbf{x}_i, \mathbf{x}_j) \psi(\mathbf{x}_j) \right)$$



Message-passing

$$\mathbf{h}_i = \phi \left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} \psi(\mathbf{x}_i, \mathbf{x}_j) \right)$$

Increasing order of generality: *convolutional* $\subseteq$ *attentional* $\subseteq$ *message – passing*

Outline

Reliability management and data in control rooms

1. Introduction to reliability management
2. Machine learning approaches
3. Security assessment with cost-sensitive supervised learning

Learning models for secure system operation

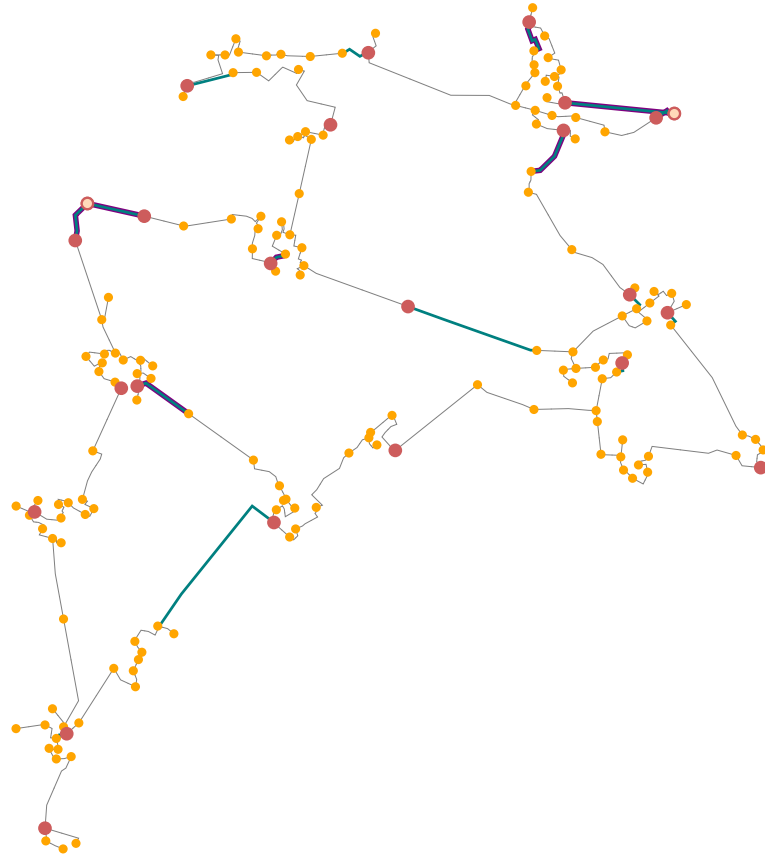
4. Learning with domain knowledge
5. State estimation with graph neural networks
6. Weakly-supervised learning for secure operation
7. Challenges applying ML to reliability

Distribution system state estimation

- Measurements $z \in \mathbb{R}^m$ with noise $\varepsilon \in \mathbb{R}^m$
- System state $x \in \mathbb{R}^{2n}$
- State estimation $f(z) \rightarrow x$
- Challenge: partial observable, scarce measurements $m \ll n$

Distribution system (MV)

- Trafo
- Lines
- MV/LV buses
- HV buses
- Power flow measurement
- Voltage measurements



Model of the power flow

$$x = [V, \varphi]$$

$$z = h(x) + \varepsilon$$

$$h(x) =$$

$$V_i = V_i$$

$$\varphi_i = \varphi_i$$

$$P_{ij \rightarrow} = -V_i V_j [\Re(Y_{ij}) \cos \Delta \varphi_{ij} + \Im(Y_{ij}) \sin \Delta \varphi_{ij}] + V_i^2 \left[\Re(Y_{ij}) + \frac{\Re(Y_{sij})}{2} \right]$$

$$P_{ij \leftarrow} = V_i V_j [-\Re(Y_{ij}) \cos \Delta \varphi_{ij} + \Im(Y_{ij}) \sin \Delta \varphi_{ij}] + V_j^2 \left[\Re(Y_{ij}) + \frac{\Re(Y_{sij})}{2} \right]$$

$$Q_{ij \rightarrow} = V_i V_j [-\Re(Y_{ij}) \sin \Delta \varphi_{ij} + \Im(Y_{ij}) \cos \Delta \varphi_{ij}] - V_i^2 \left[\Im(Y_{ij}) + \frac{\Im(Y_{sij})}{2} \right]$$

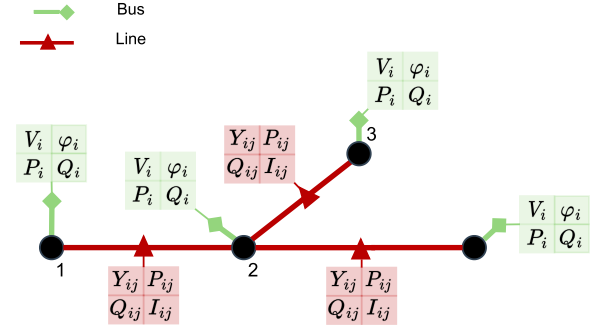
$$Q_{ij \leftarrow} = V_i V_j [\Re(Y_{ij}) \sin \Delta \varphi_{ij} + \Im(Y_{ij}) \cos \Delta \varphi_{ij}] - V_j^2 \left[\Im(Y_{ij}) + \frac{\Im(Y_{sij})}{2} \right]$$

$$I_{ij \rightarrow} = -\frac{P_{ij \rightarrow} - jQ_{ij \rightarrow}}{\sqrt{3}V_i e^{-j\varphi_i}}$$

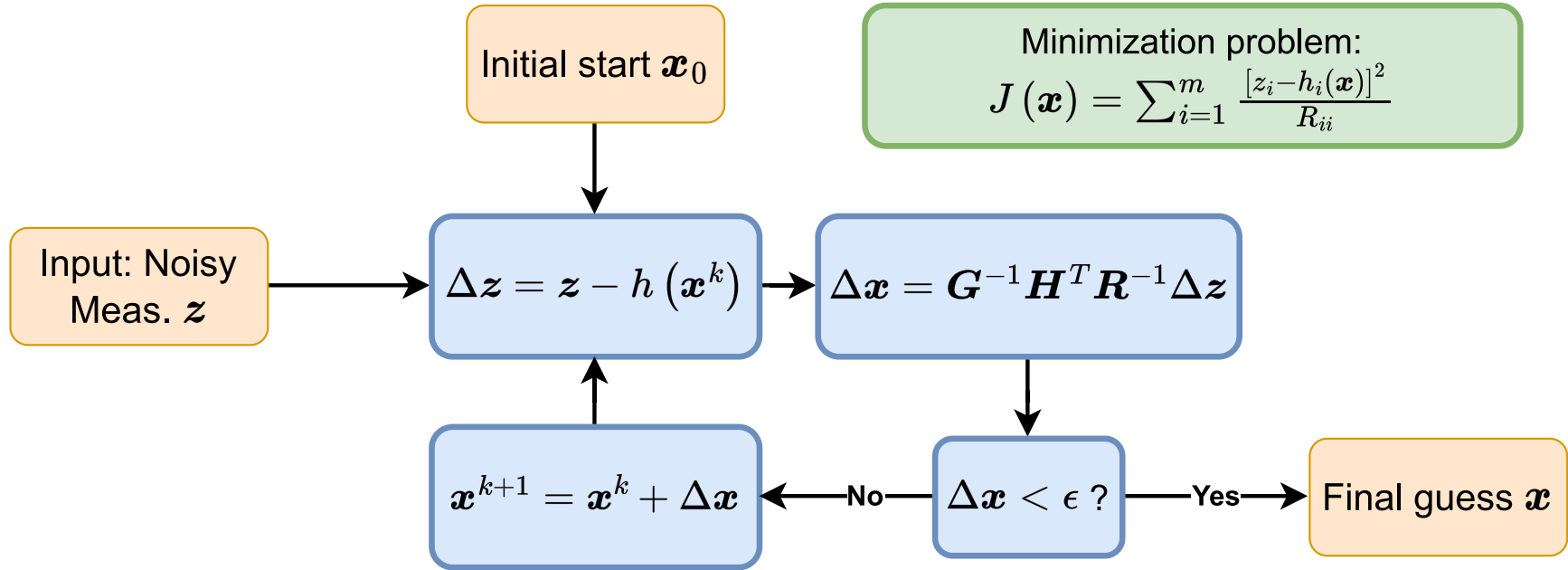
$$I_{ij \leftarrow} = -\frac{P_{ij \leftarrow} - jQ_{ij \leftarrow}}{\sqrt{3}V_j e^{-j\varphi_j}}$$

$$P_i = -\sum_{j \in N_x(i)} P_{ij \leftarrow} + P_{ij \rightarrow}$$

$$Q_i = -\sum_{j \in N_x(i)} Q_{ij \leftarrow} + Q_{ij \rightarrow}$$



Weighted least squares method

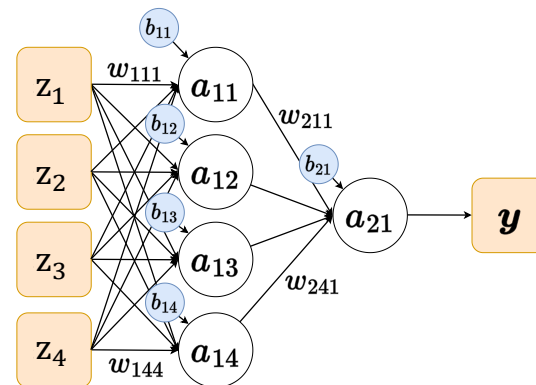


How accurate is $\mathbf{x}$?
Convergence?

Statistical learning

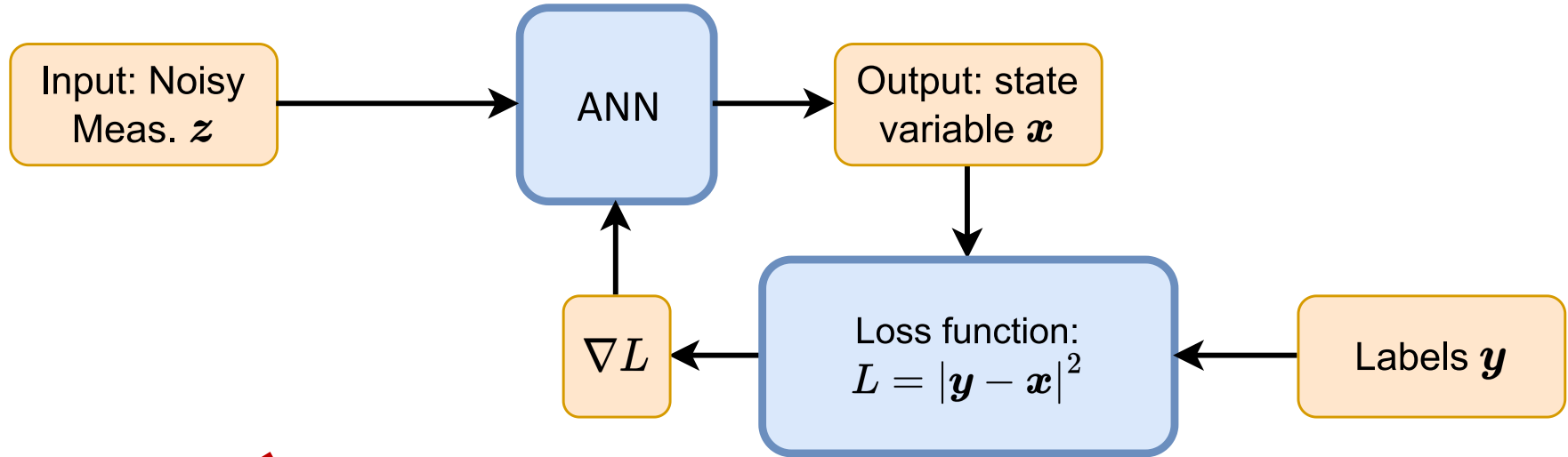
- Training set $\Omega^T = \{(z_1, y_1), (z_2, y_2) \dots (z_t, y_t)\}$ with t samples
- Inference problem is to find a function $f: Z \rightarrow Y$ such that $f(z) \sim y$
- Common loss function $L(f(z), y)$ for regression is the square loss

Artificial Neural Network (ANN)



$$f_{\theta}: Z \rightarrow Y$$

Supervised learning for state estimation



⚡ Labels are not known

⚡ Newton's method generates label with "errors" $\hat{y} = y + \gamma^N$

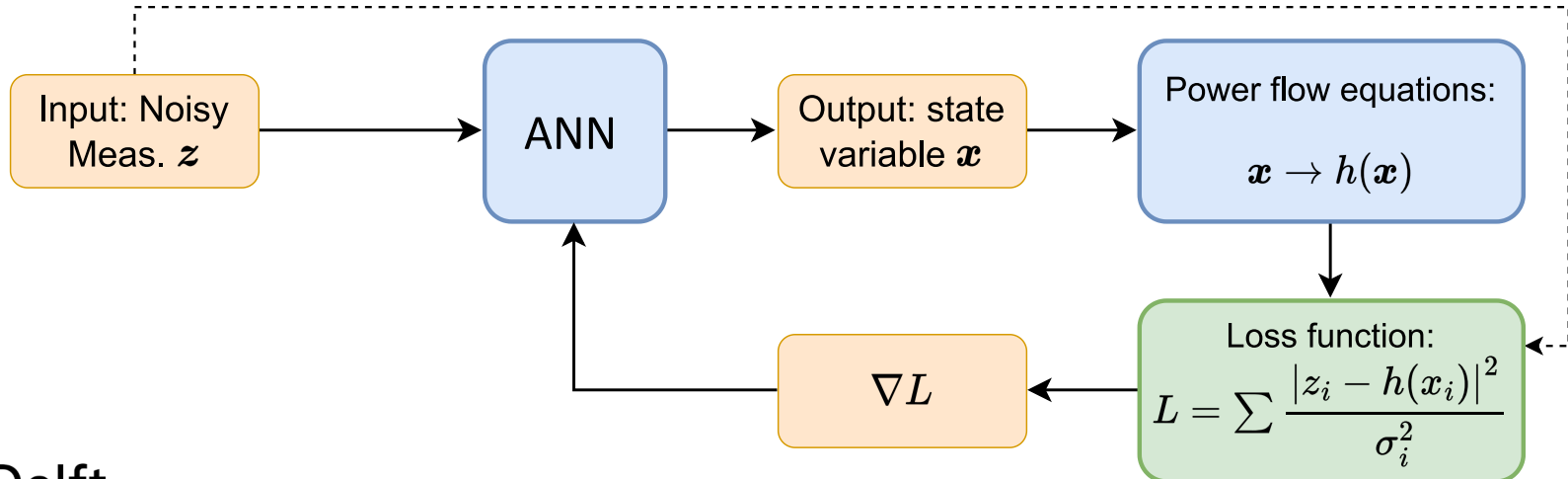
Weakly-supervised learning

- Inaccurate input and **output**
- Learn with inaccurate labels $\Omega^T = \{(z_1, \hat{y}_1), (z_2, \hat{y}_2) \dots (z_t, \hat{y}_t)\}$
- Design a loss function $L(f(z), \hat{y})$
- Objective: learning $f: Z \rightarrow Y$ such that $f(z) \sim y$



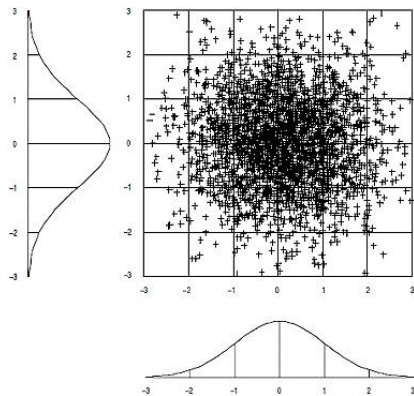
Weakly-supervised learning for state estimation

- ANN $f(z) \rightarrow x$
- Measurement function using power flow equations $h(x) \rightarrow \hat{z}$



Respect the structure of the domain

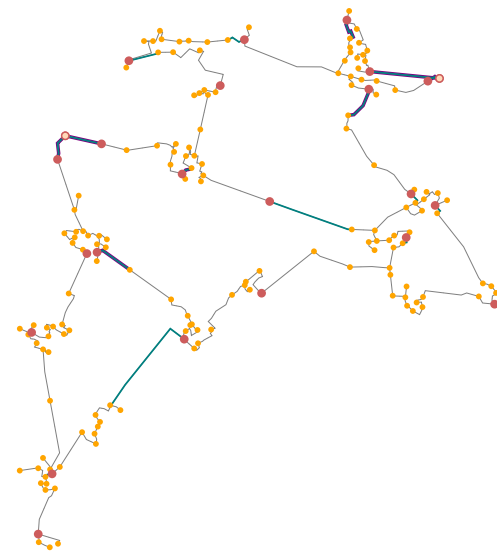
Noisy measurements



Power flow equations

$$h(x) = \begin{cases} V_i = V_i \\ \varphi_i = \varphi_i \\ P_{ij \rightarrow} = -V_i V_j [\Re(Y_{ij}) \cos \Delta \varphi_{ij} + \Im(Y_{ij}) \sin \Delta \varphi_{ij}] + V_i^2 \left[\Re(Y_{ij}) + \frac{\Re(Y_{sij})}{2} \right] \\ P_{ij \leftarrow} = V_i V_j [-\Re(Y_{ij}) \cos \Delta \varphi_{ij} + \Im(Y_{ij}) \sin \Delta \varphi_{ij}] + V_j^2 \left[\Re(Y_{ij}) + \frac{\Re(Y_{sij})}{2} \right] \\ Q_{ij \rightarrow} = V_i V_j [-\Re(Y_{ij}) \sin \Delta \varphi_{ij} + \Im(Y_{ij}) \cos \Delta \varphi_{ij}] - V_i^2 \left[\Im(Y_{ij}) + \frac{\Im(Y_{sij})}{2} \right] \\ Q_{ij \leftarrow} = V_i V_j [\Re(Y_{ij}) \sin \Delta \varphi_{ij} + \Im(Y_{ij}) \cos \Delta \varphi_{ij}] - V_j^2 \left[\Im(Y_{ij}) + \frac{\Im(Y_{sij})}{2} \right] \\ I_{ij \rightarrow} = -\frac{P_{ij \rightarrow} - jQ_{ij \rightarrow}}{\sqrt{3}V_i e^{-j\varphi_i}} \\ I_{ij \leftarrow} = -\frac{P_{ij \leftarrow} - jQ_{ij \leftarrow}}{\sqrt{3}V_j e^{-j\varphi_j}} \\ P_i = -\sum_{j \in \mathcal{N}_i(t)} P_{ij \leftarrow} + P_{i \rightarrow} \\ Q_i = -\sum_{j \in \mathcal{N}_i(t)} Q_{ij \leftarrow} + Q_{i \rightarrow} \end{cases}$$

Topology



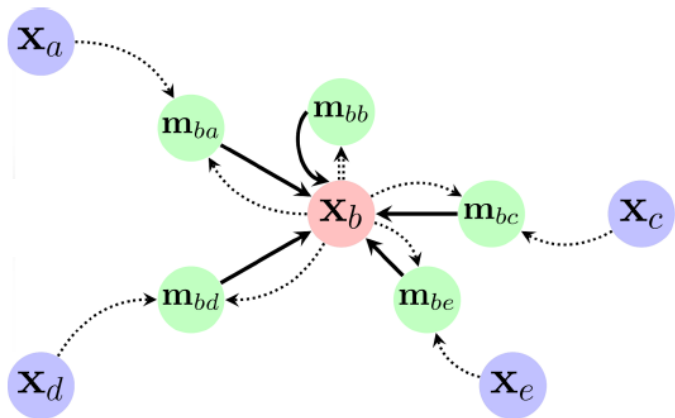
Locality on graphs: Neighbourhoods

- Consider graph $G = (V, E)$ where $E \subseteq V \times V$
- Adjacency matrix A with

$$a_{ij} = \begin{cases} 1, & (i, j) \in E \\ 0, & (i, j) \notin E \end{cases}$$

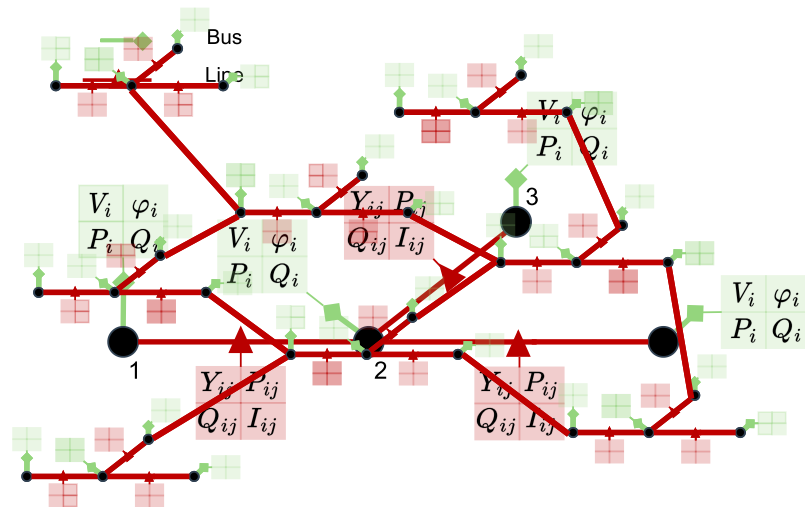
- (1-hop) neighbourhood $N_i = \{j: (i, j) \in E \cup (j, i) \in E\}$ for a node i
- Neighbourhood features $\mathbf{X}_{N_i} = \{\{x_j: j \in N_i\}\}$
- Local function, $\phi(x_i, \mathbf{X}_{N_i})$, operating over them.

Convolutional layers & message passing



$$\eta_i = \phi \left(x_i, \bigoplus_{j \in N_i} \psi(x_i, x_j) \right)$$

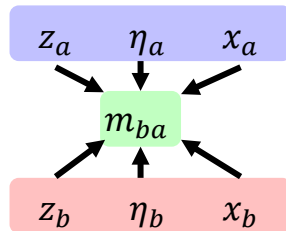
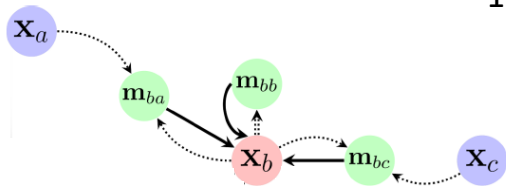
State estimation



Deep statistical solver

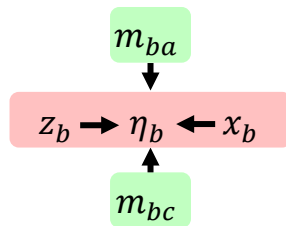
0. Initialize $x = x^0, \eta = \eta^0$

1. update edges



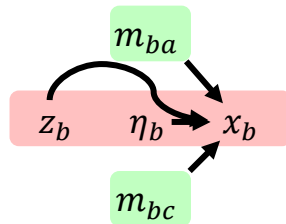
$$m_{ba} \leftarrow m_{ba} + \Delta t \times [\phi_{\theta}^{ba}(t, z_a, \eta_a, x_a) + \phi_{\theta}^{ba}(t, z_b, \eta_b, x_b)]$$

2. update vertices



$$\eta_b \leftarrow \eta_b + \Delta t \times \phi_{\theta}^b(t, z_b, \eta_b, x_b, m_{ba}, m_{bc}, m_{bb})$$

3. update label



$$x_b \leftarrow x_b + \Delta t \times \phi_{\theta}^{bx}(t, z_b, \eta_b, x_b, m_{ba}, m_{bc}, m_{bb})$$

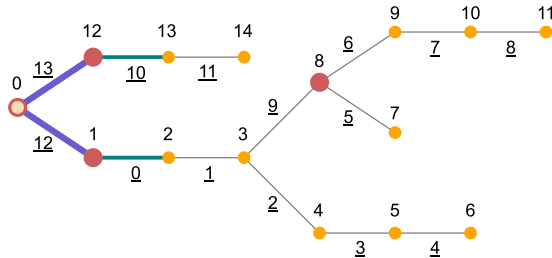
Perform $t = 1, \dots, T$



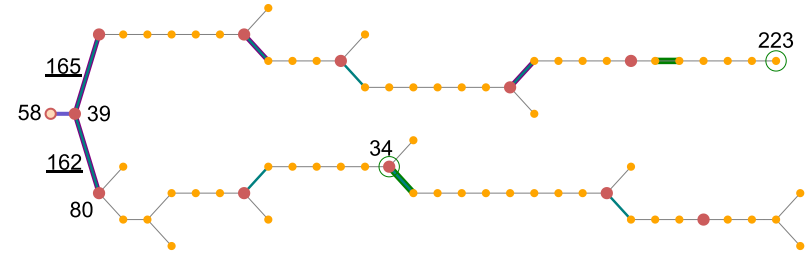
Benjamin Habib

Case study: power systems

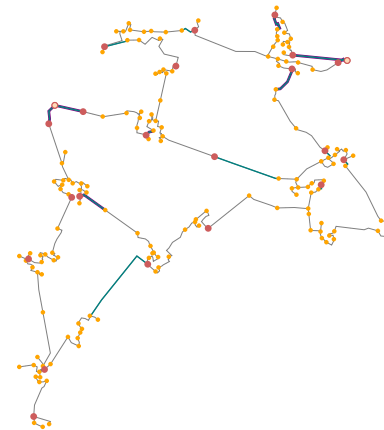
14-bus CIGRE MV grid from



70-bus Oberrhein MV sub-grid from



179-bus Oberrhein MV grid from



- Trafo
- Lines
- MV/LV buses
- HV buses
- Power flow measurement
- Voltage measurements
- Focus bus

Case study settings

Data generation

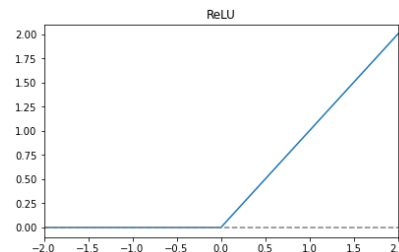
- 8640 days, with each 24 hours, +/- 15% around Gaussian on loads
- Balanced system, pandapower, AC power flow
- Measurement noise
 - 0.5% – 2% for the voltage and current measurement
 - 1% – 5% for the active and reactive power measurement
- Pseudomeasurement were generic load profiles
- Baselines
 - Weighted least square (WLS)
 - Feedforward Neural Network (FFNN)
 - supervised DSS^2

Model & hyperparameters

- Hyper-Heterogeneous Multi GNN
- Training 80%, validation 10%, testing 10%
- Grid search on learning rate λ , layer dimensions d , and layer numbers

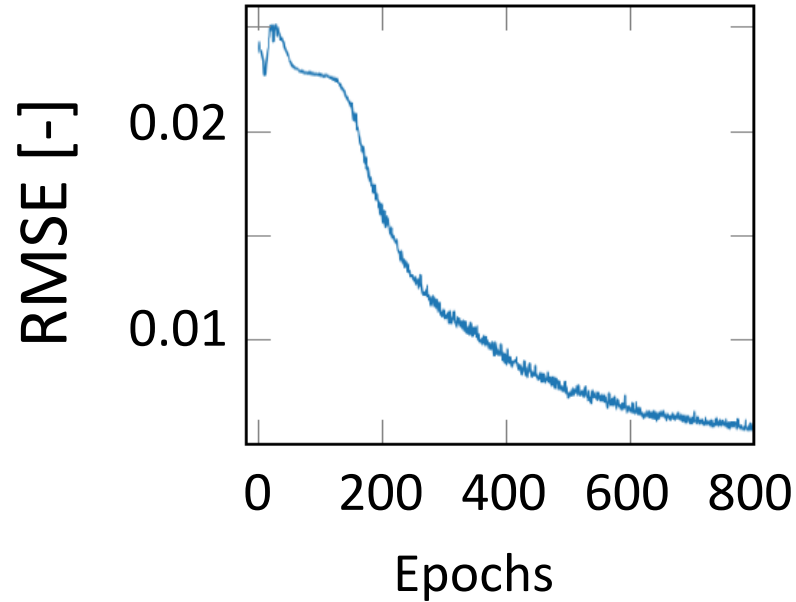
Assume stable system

$$L(z, x) = \sum_{i \in m} \frac{|z_i - h_i(x)|^2}{R_{ii}} + \lambda [\text{ReLU}(V - 1.05) + \text{ReLU}(0.95 - V) + \text{ReLU}(\text{loading} - 100) + \text{ReLU}(\varphi - 0.25) + \text{ReLU}(-0.25 - \varphi)]$$

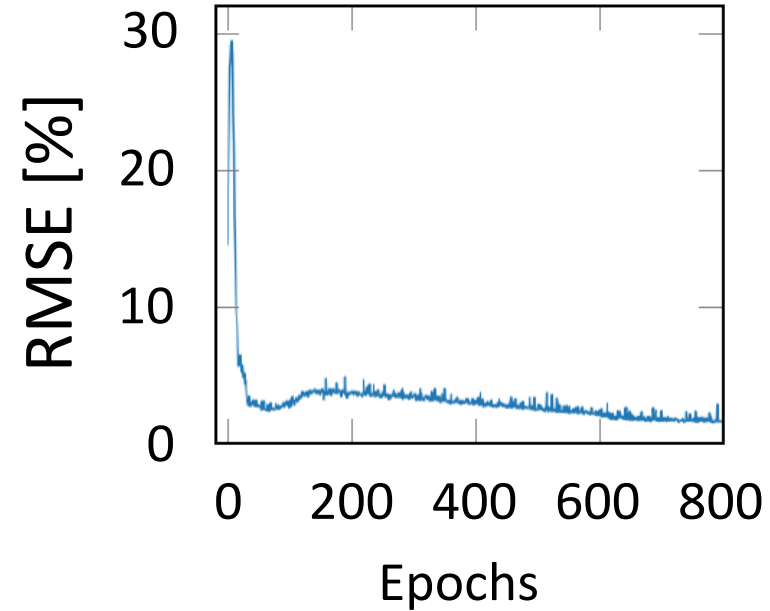


Training performance 14-bus system

Voltage levels

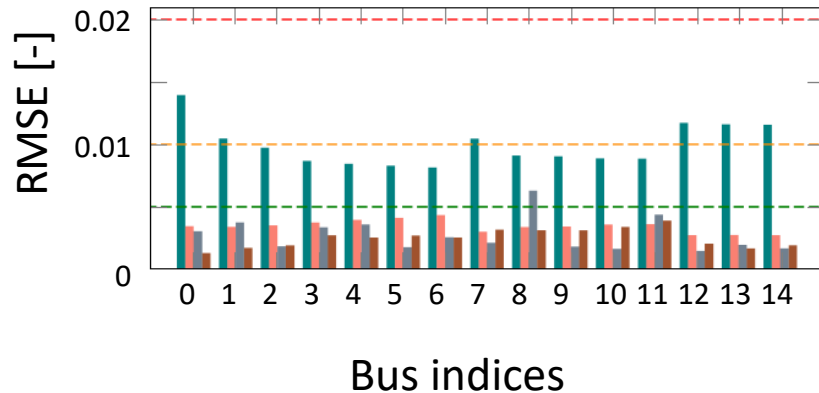


Line loadings

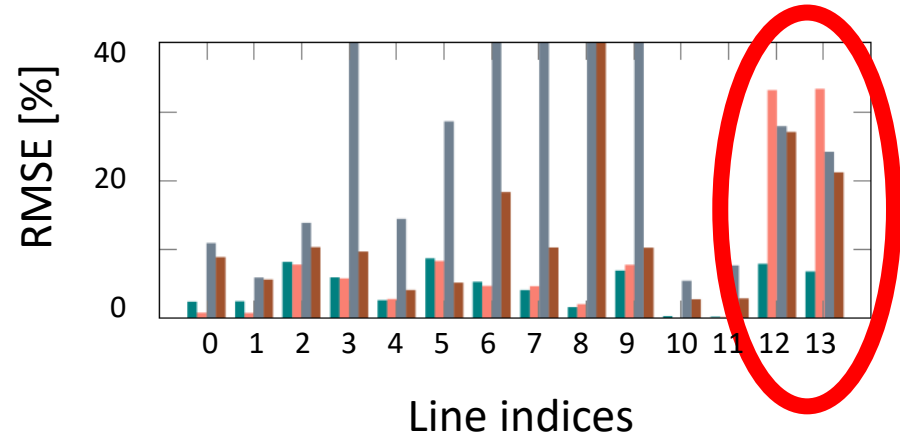


State estimation 14-bus system

Voltage levels



Line loadings



Model inaccuracies: assumed transformer = lines!

Accuracy

Metric \ approach	14-bus system			
	WLS	ANN	sup DSS^2	DSS^2
Voltage RMSE [10^{-3}]	10	3	3	3
Line loading RMSE [%]	3	42	13	4
Trafos loading RMSE [%]	5	39	14	8

Convergence

	14-bus system				70-bus Oberrhein			179-bus Oberrhein	
Metric \ approach	WLS	ANN	sup DSS^2	DSS^2	WLS	WLS*	DSS^2	WLS**	DSS^2
Voltage RMSE [10^{-3}]	10	3	3	3	31	6	2	10	2
Line loading RMSE [%]	3	42	13	4	17	15	2	6	3
Trafos loading RMSE [%]	5	39	14	8	39	24	3	4	4
Convergence [%]	100	100	100	100	25	100	100	53	100

- WLS did not converge in some instances (25%-50%)
- DSS^2 always ‘converges’ (produces a label)

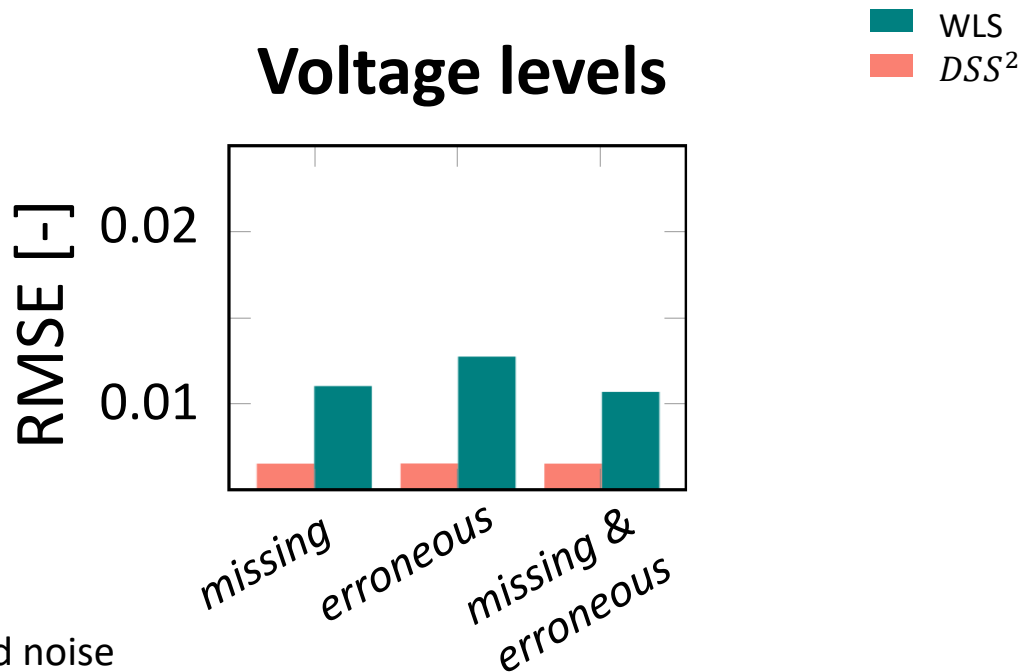
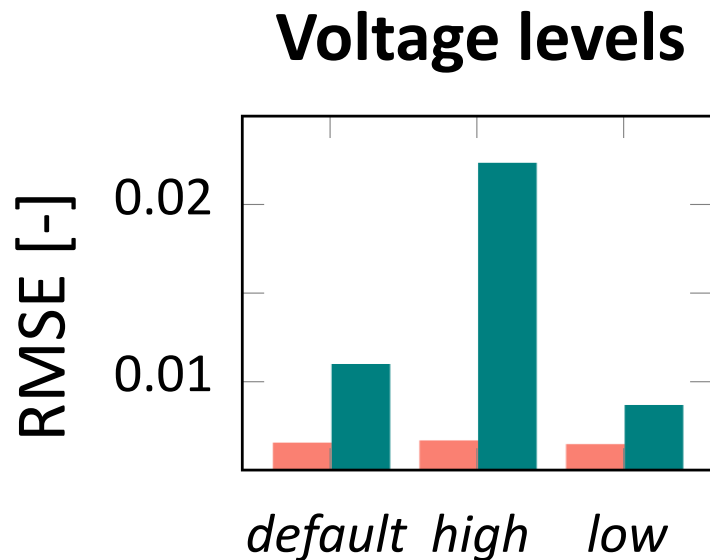
Computational ‘prediction’ time [ms]

	14-bus system				70-bus Oberrhein			179-bus Oberrhein	
Metric \ approach	WLS	ANN	sup DSS^2	DSS^2	WLS	WLS*	DSS^2	WLS**	DSS^2
Voltage RMSE [10^{-3}]	10	3	3	3	31	6	2	10	2
Line loading RMSE [%]	3	42	13	4	17	15	2	6	3
Trafos loading RMSE [%]	5	39	14	8	39	24	3	4	4
Convergence [%]	100	100	100	100	25	100	100	53	100
Computational time [ms]	86	4	5	6	123	161	26	1212	58

~10 ~2

- WLS increases significantly with system size
- DSS^2 increases moderately with system size

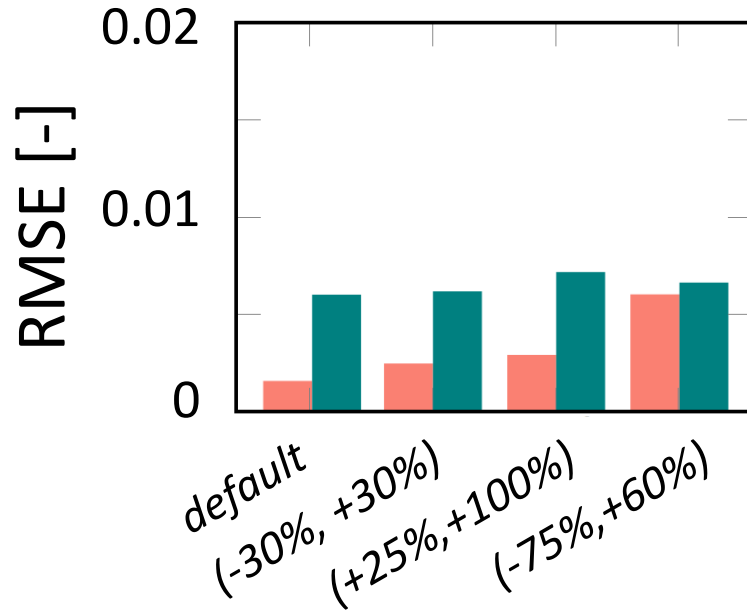
Noise and missing, erroneous data



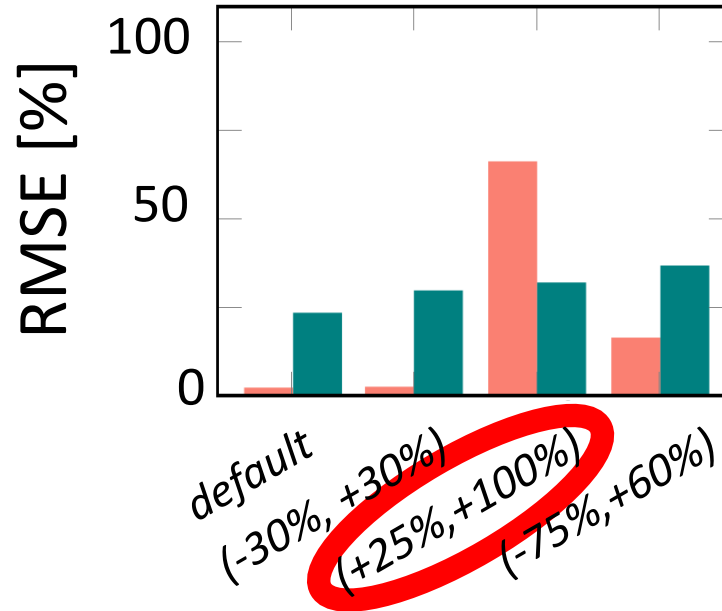
- DSS^2 successfully cancelled noise
- DSS^2 was not trained to handle such events
- GNN architecture increased the interpolation capabilities by incorporating the data symmetries w.r.t. the underlying graph

Increase in (generation, load)

Voltage levels



Line loadings



Limitation: high load levels

Outline

Reliability management and data in control rooms

1. Introduction to reliability management
2. Machine learning approaches
3. Security assessment with cost-sensitive supervised learning

Learning models for secure system operation

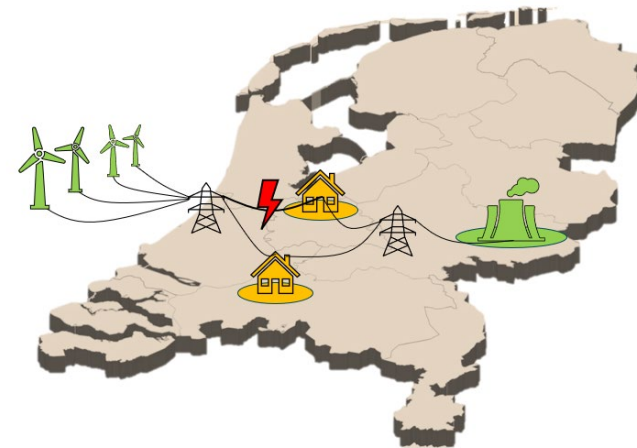
4. Learning with domain knowledge
5. State estimation with graph neural networks
6. Weakly-supervised learning for secure operation
7. Challenges applying ML to reliability



Bastien Giraud

Problem overview

- Growing grid complexity
 - Challenging to maintain N-1 security
- Increasing number of unforeseen weather events
- Need for N-k considerations to increase reliability
- **Problem:** Conventional approaches don't scale well with the number of simultaneous outages k



Security constrained optimal power flow (SCOPF)

Objective: minimize cost

Constraints: In = out
Generator limits
Line flow limits

Contingency Constraints: Line flow limits

$$\min_{n \in \Omega^G} \sum c_n P_{G_n}$$

$$B \cdot \delta = P_G - P_D$$

$$P_{G_n}^{min} < P_{G_n} < P_{G_n}^{max} \quad \forall n \in \Omega^G$$

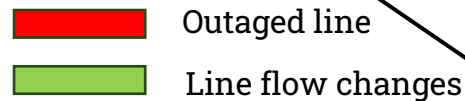
$$F_l^{min} < F_l < F_l^{max} \quad \forall l \in \Omega^L$$

$$F_l^{min} < F_l^c < F_l^{max} \quad \forall l \in \Omega^L, \forall c \in \Omega^C$$



Combinatorial
complexity

Conventional approaches

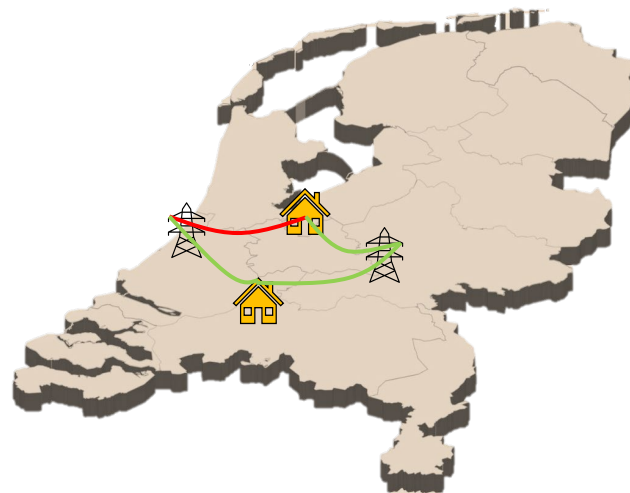


Solving a **large optimization** problem can be slow

- Benders decomposition
- Column and constraint generation algorithm with robust optimization
- Line outage distribution factors (**LODF**)

Machine learning approaches often rely on **labeled** training data

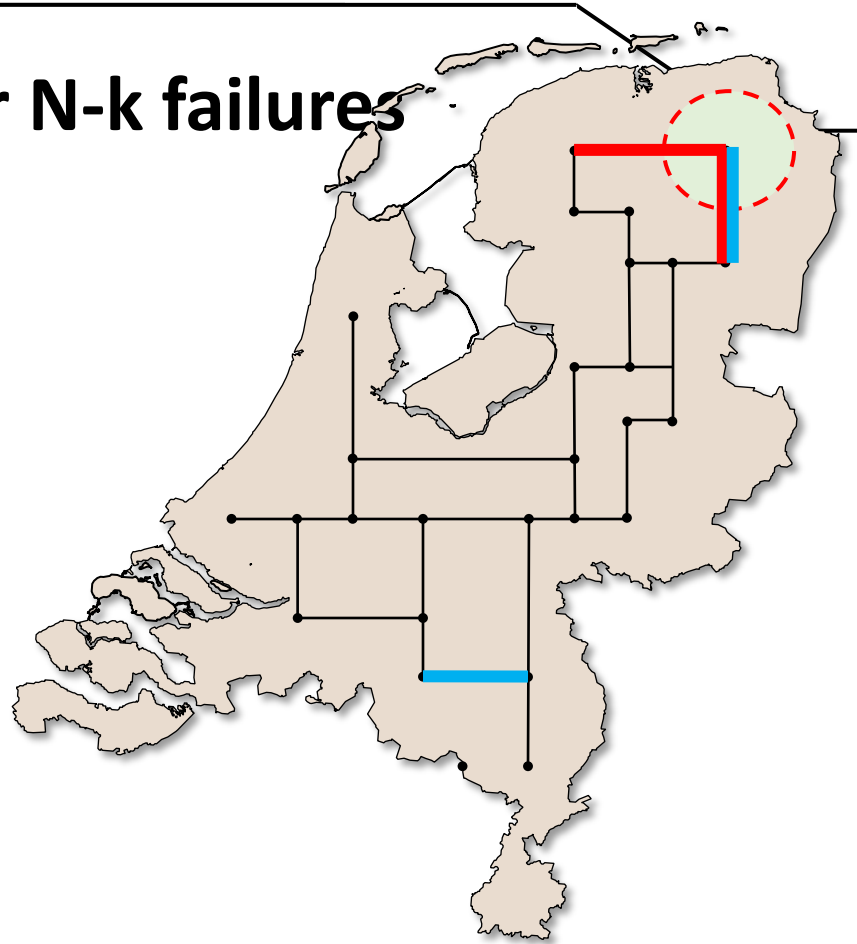
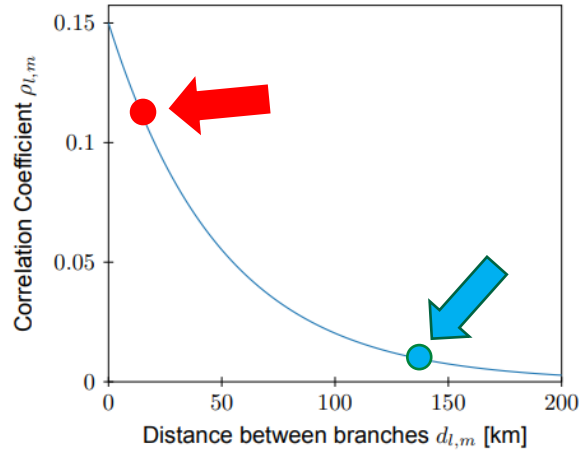
- Intractable for increasing k



$$F^c = F^0 + LODF_{N-k} \times F^0$$

Probabilistic security for N-k failures

- Compute probabilities of all contingencies
- Spatial correlation between line outages
- Compute joint probabilities using a copula analysis



Proposed constraint-driven approach

Main advantages

- Weakly-supervised -> so **no labeled data** needed
- Never actually solve an SCOPF

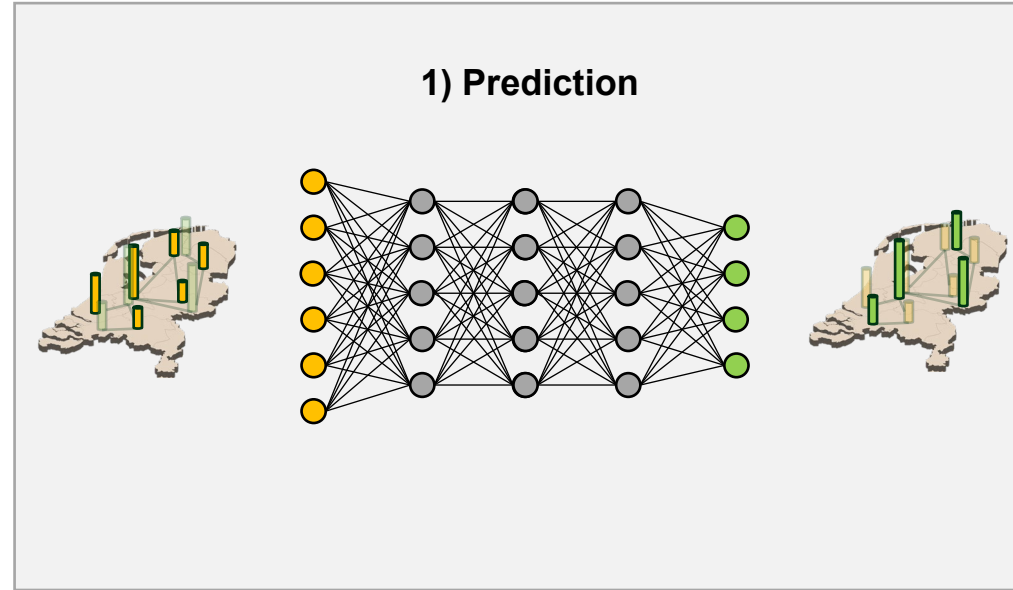
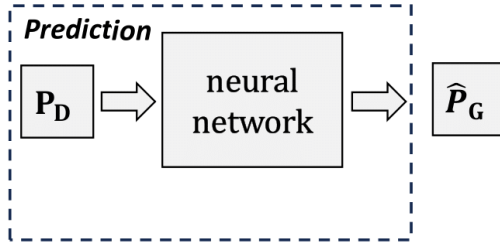
Contributions

- The deterministic constraint-driven approach to approximate N-k SCOPFs, considering all line contingencies using LODFs.
- The computational graph memory reduction for fast and efficient implementation.
- The probabilistic security assessment to formulate a N-k risk-based security criterion, providing an alternative to the current deterministic N-1 security criterion.

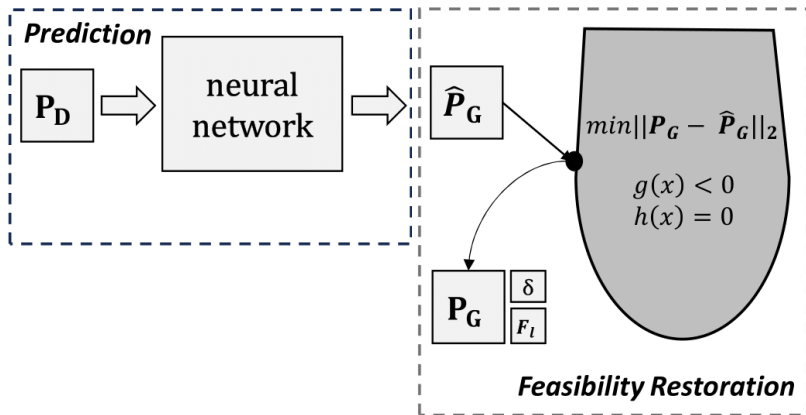


LODF = line outage distribution factor
SCOPF = security constrained optimal power flow

Proposed constraint-driven approach



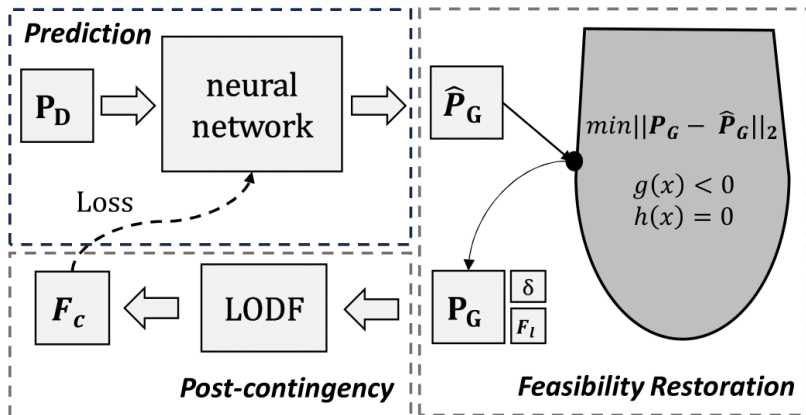
Proposed constraint-driven approach



2) Feasibility Restoration

- With $\hat{P}_{G_n}$ compute predicted line flow $\hat{F}_l^0$
- Prediction might **violate** DC PF equations
- Map prediction to feasible region constrained by DC PF equations

Proposed constraint-driven approach

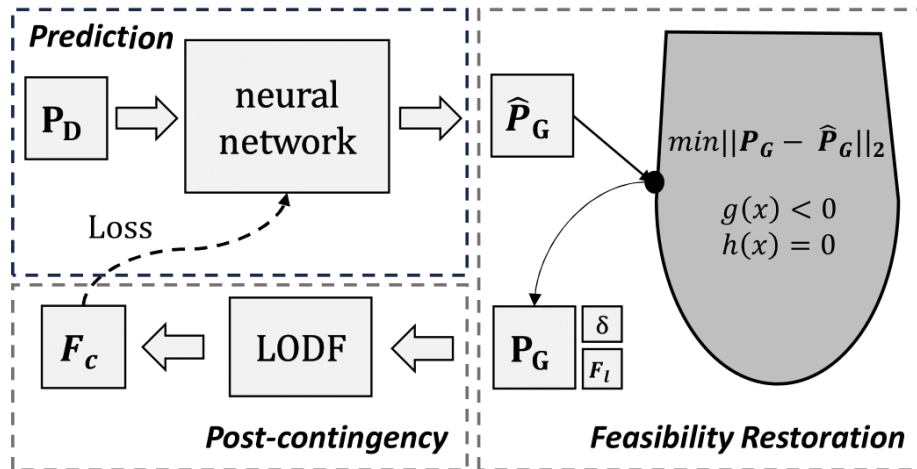


3) Post-contingency

$$F^c = F^0 + LODF_{N-k} \times F^0$$

$$F_l^{min} < F_l^c < F_l^{max} \quad \forall l \in \Omega^L, \forall c \in \Omega^C$$

Proposed constraint-driven approach



1) Dispatch cost

$$\lambda_c \sum P_G c_G$$

2) Line flow violation pre-contingency

$$\lambda_0 \|ReLU(|\hat{F}^0| - F^{max})\|_1$$

3) Line flow violation post-contingency

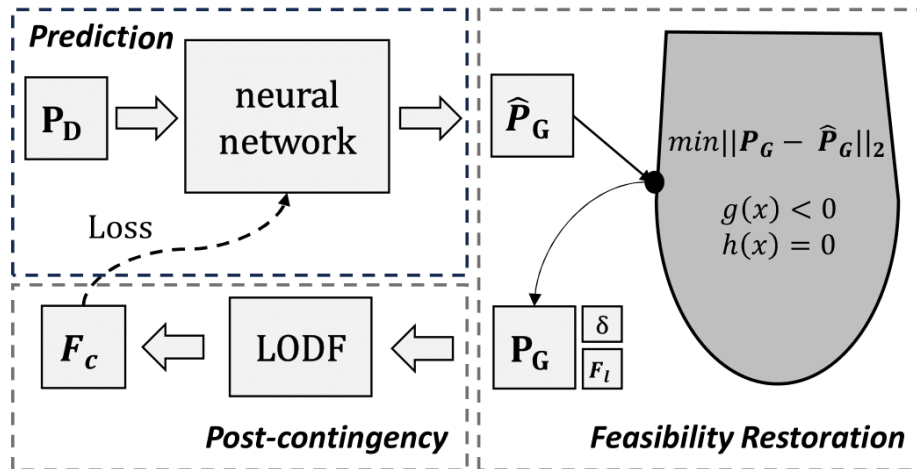
$$\lambda_1 \|ReLU(|F^c| - F^{max})\|_1$$

4) Power imbalance

$$\lambda_2 \|\sum \hat{P}_G - \sum P_D\|_1$$

$$Loss = \lambda_c \sum P_G c_G + \lambda_0 \|ReLU(|\hat{F}^0| - F^{max})\|_1 + \lambda_1 \|ReLU(|F^c| - F^{max})\|_1 + \lambda_2 \|\sum \hat{P}_G - \sum P_D\|_1$$

Proposed constraint-driven approach



1) Dispatch cost

$$\lambda_c \sum P_G c_G$$

2) Line flow violation pre-contingency

$$\lambda_0 \|ReLU(|\hat{F}^0| - F^{max})\|_1$$

3) Line flow violation post-contingency

$$\lambda_1 \|\pi_{N-k} \cdot ReLU(|F^c| - F^{max})\|_1$$

4) Power imbalance

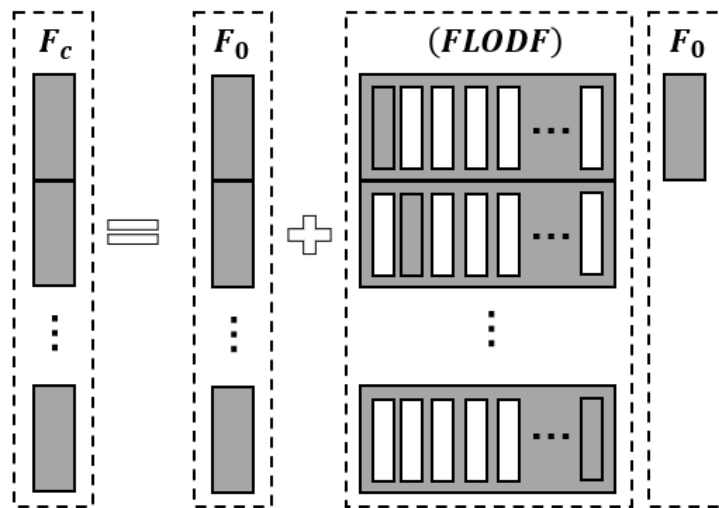
$$\lambda_2 \|\sum \hat{P}_G - \sum P_D\|_1$$

$$Loss = \lambda_c \sum P_G c_G + \lambda_0 \|ReLU(|\hat{F}^0| - F^{max})\|_1 + \lambda_1 \|\pi_{N-k} \cdot ReLU(|F^c| - F^{max})\|_1 + \lambda_2 \|\sum \hat{P}_G - \sum P_D\|_1$$

Sparsity LODF matrix


 Non-zero values
 Zeros

$$F^c = F^0 + LODF_{N-k} \times F^0$$

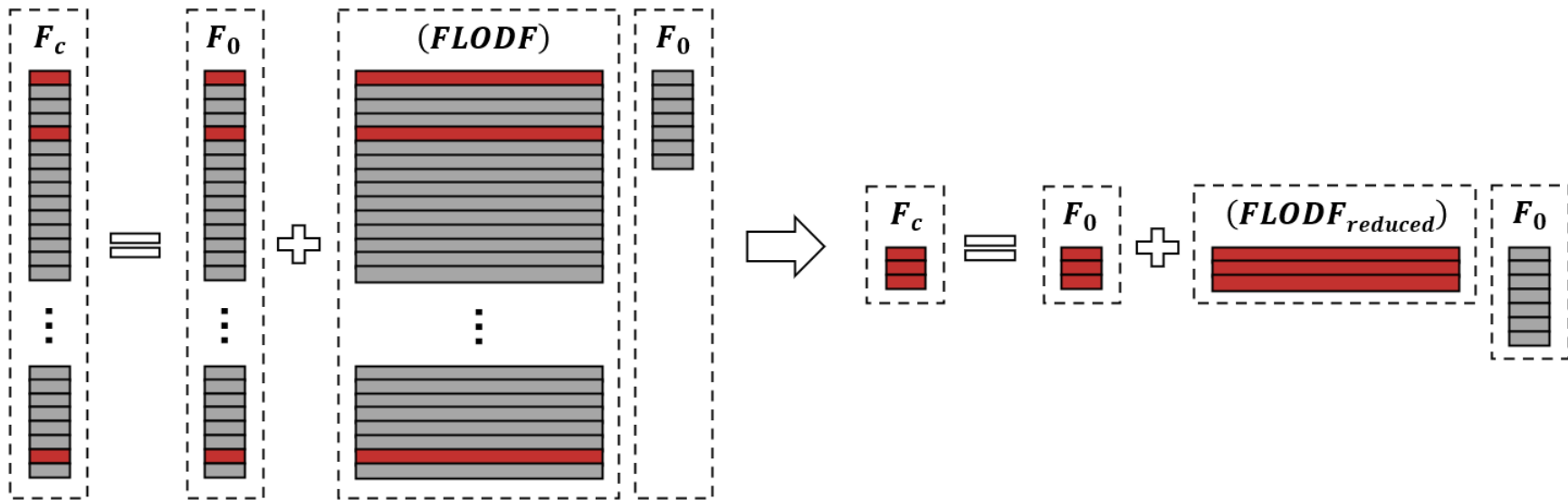


FLODF = 'Full LODF'

	k	1	2	3
39-bus sparsity [%]		98.5	97.6	97.5
118-bus sparsity [%]		99.6	99.3	99.0

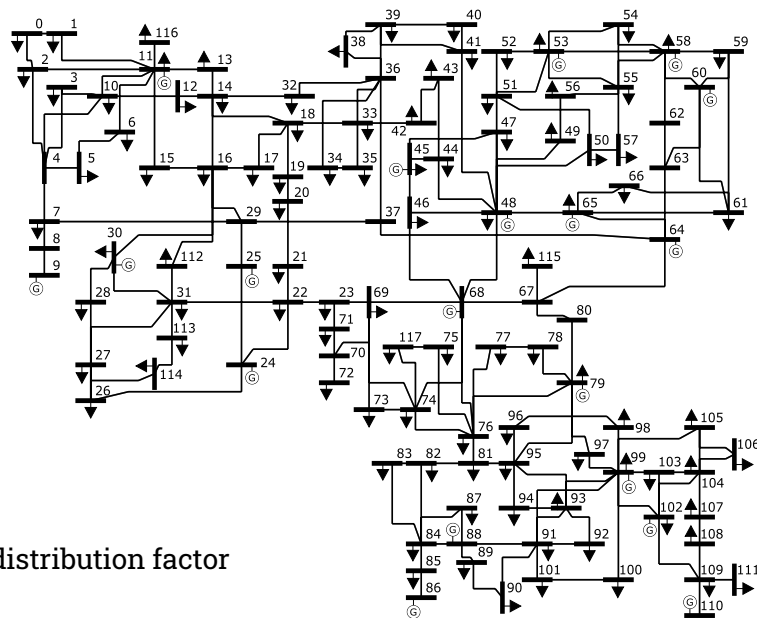
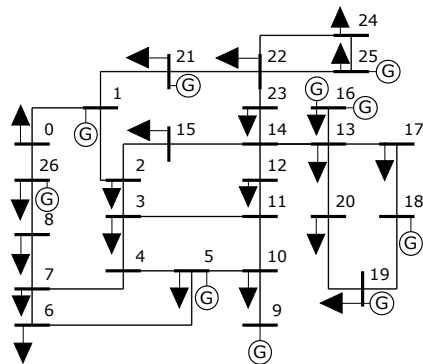
LODF = line outage distribution factor

Reducing the graph

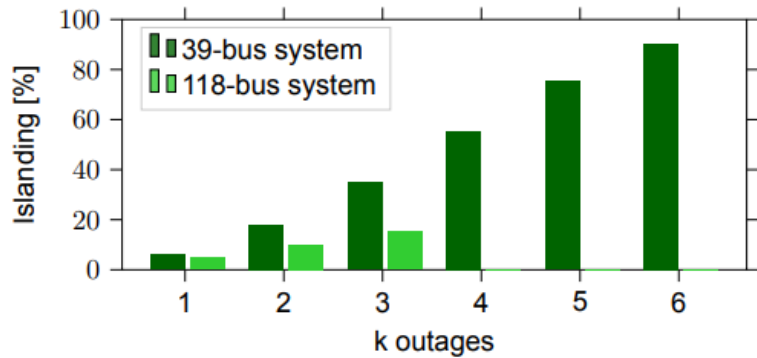
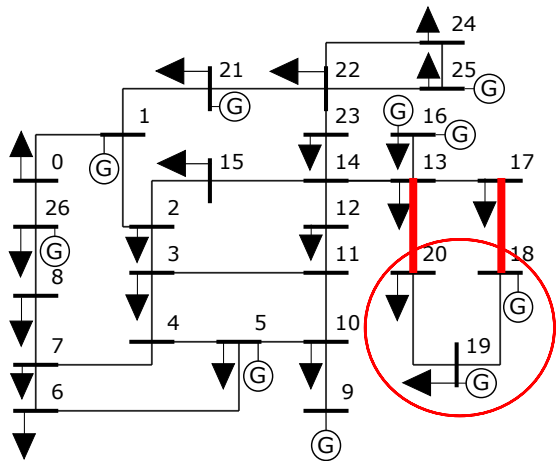


Case studies

- IEEE 39-bus and 118-bus test systems
- $k = \{1, 2, 3\}$
- Baseline: iterative contingency screening with LODFs
- Code: <https://github.com/TU-Delft-AI-Energy-Lab/Constraint-Driven-SCOPF>

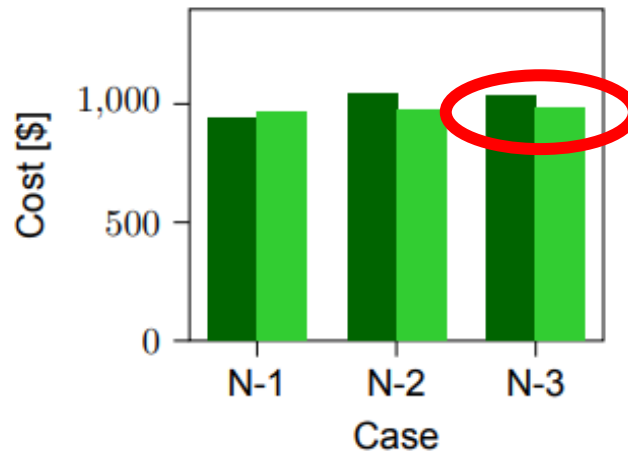
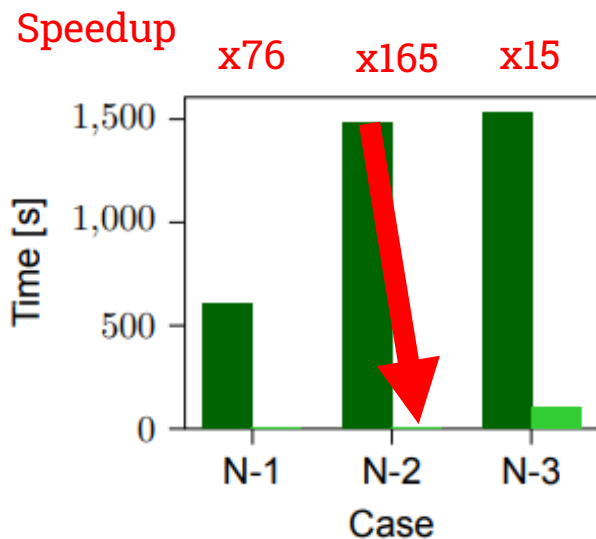


Islanding



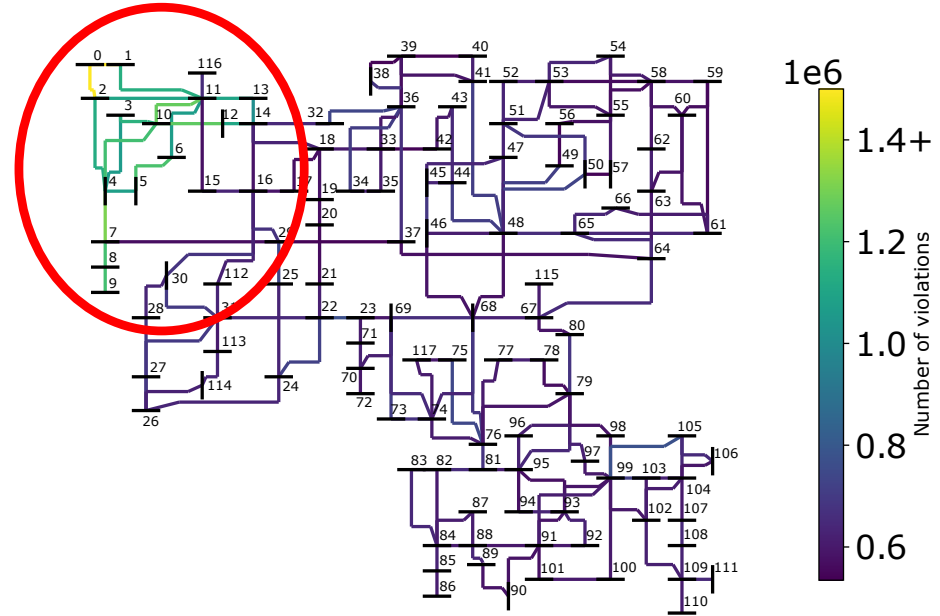
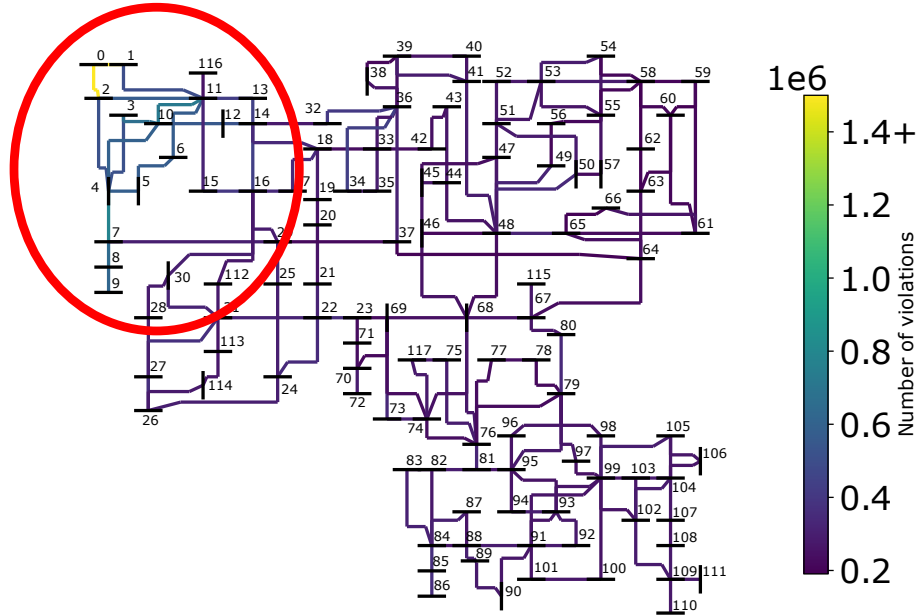
Removing islanding cases

Performance 118-bus system

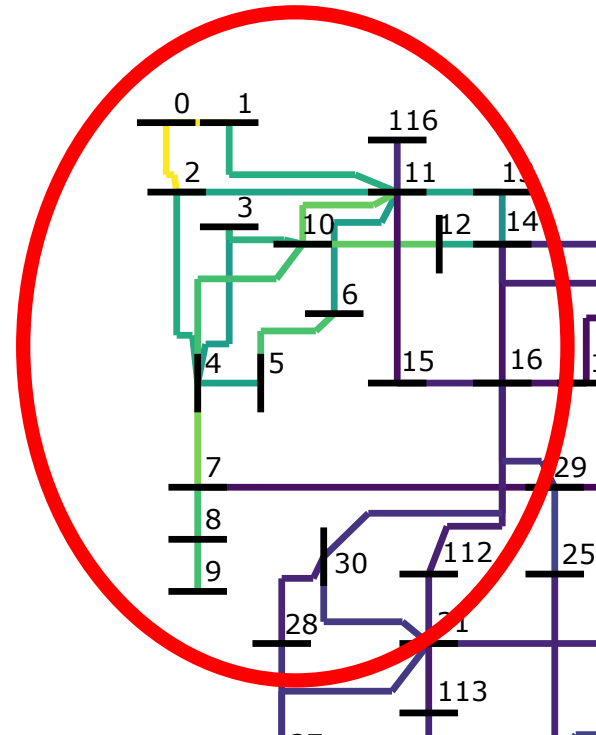
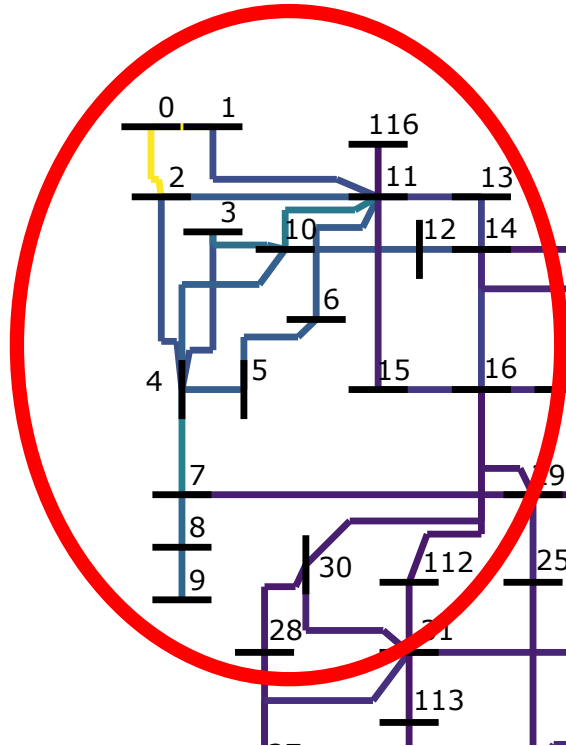


N-3 proposed approach

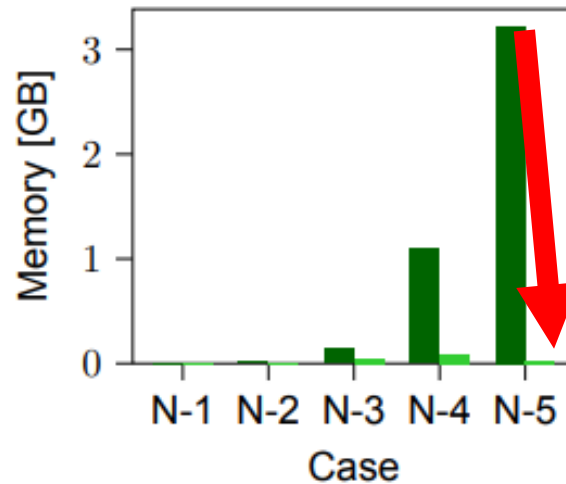
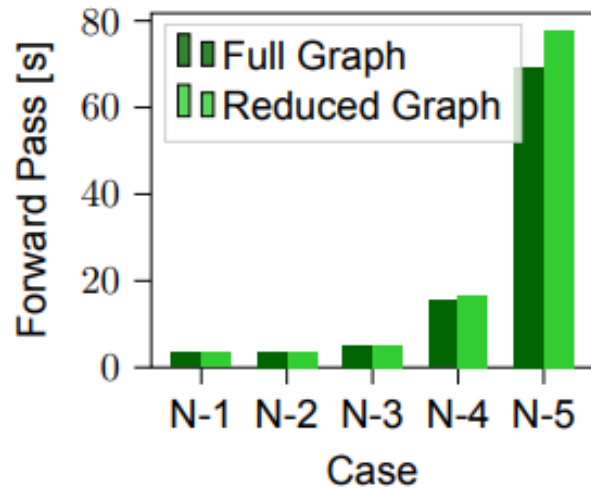
N-3 baseline



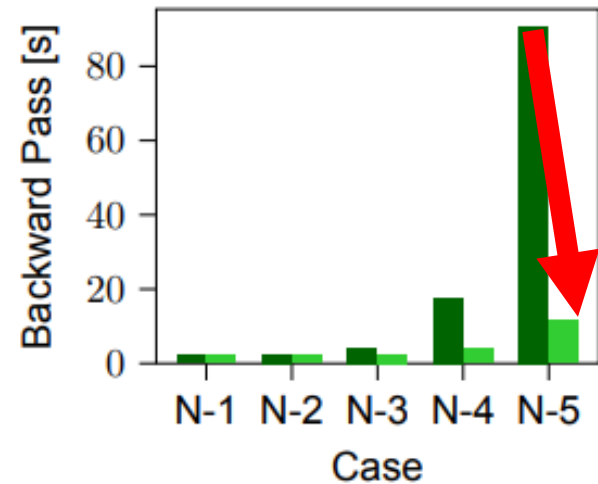
N-3 proposed approach



Reducing computational graph



Reduction in memory



Reduction in
computation time

Outline

Reliability management and data in control rooms

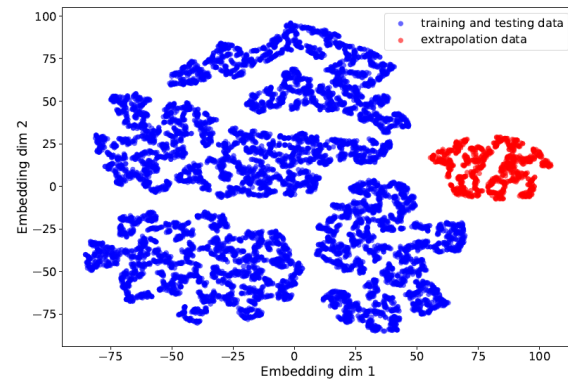
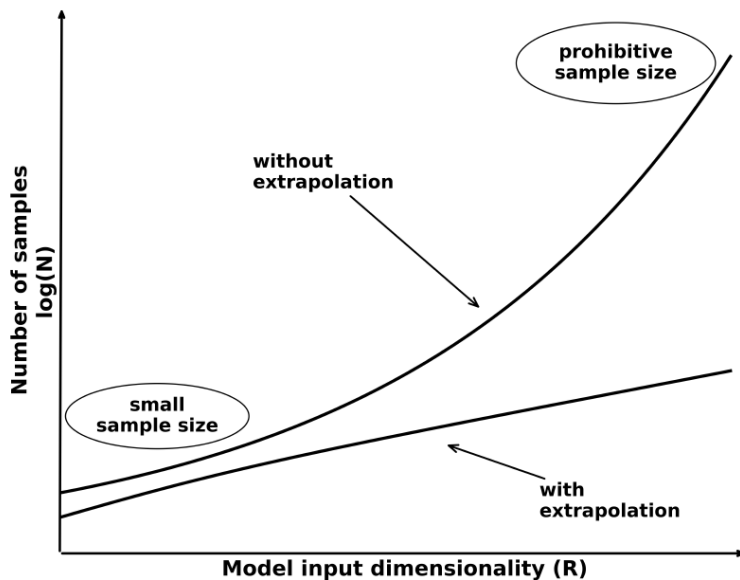
1. Introduction to reliability management
2. Machine learning approaches
3. Security assessment with cost-sensitive supervised learning

Learning models for secure system operation

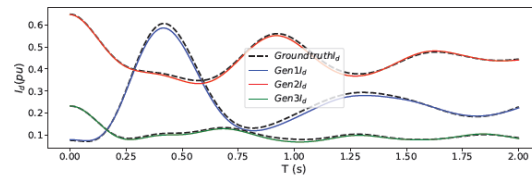
4. Learning with domain knowledge
5. State estimation with graph neural networks
6. Weakly-supervised learning for secure operation
7. Challenges applying ML to reliability

Generalisation to changes in s or m

The model performs well not just on training data, but on **unseen scenarios** — new grid states, topologies, contingencies, or time horizons.

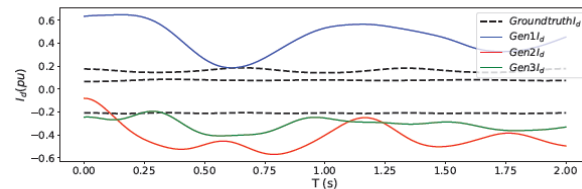


Extrapolation in continuous domain



(a) I_d current trajectory

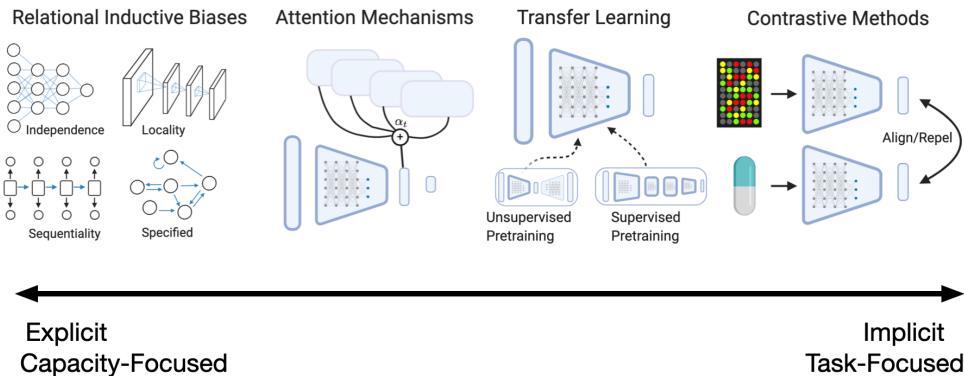
Extrapolation in nonlinear domain (discrete)



(a) I_d current trajectory

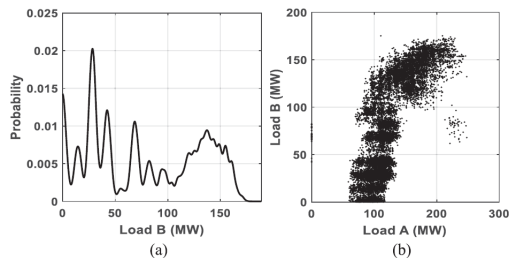
Challenge: Data-efficiency

- Data efficiency is critical
- Embedding **inductive bias** and learning **task-aware representations** helps supervised models generalise better — even with limited labels.

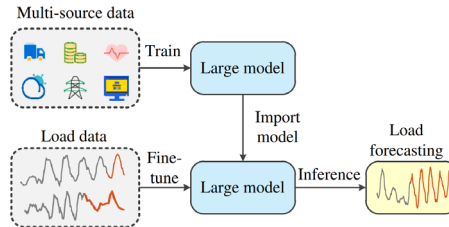


Sampling synthetic data & use real-data

Snapshot sampling



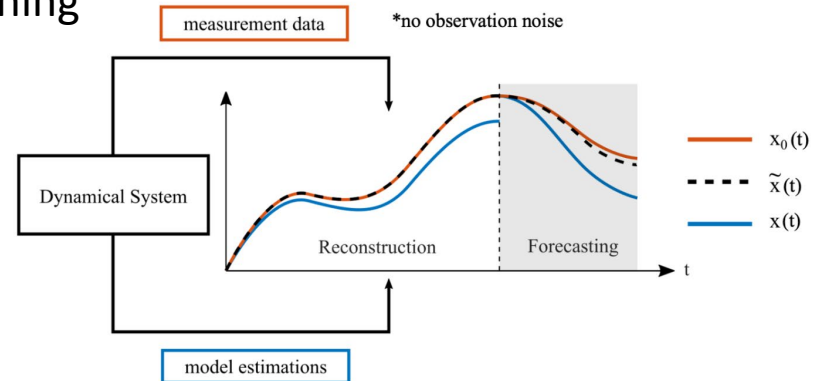
Time-series foundational models



Model inaccuracy $s \neq m$ (data quality issues)

"All models are wrong, but some are useful", George E. P. Box

- Example challenges
 - Distribution: Inaccurate transformer-tap positions
 - Transmission: Converter-based control models are unknown
- Possible techniques: Parameter estimation to develop probabilistic and deterministic models, discrepancy learning



Conclusions

- For many decades, AI has been investigated for power system reliability -> demonstrating promising ideas
- Promising: New techniques, availability of data, models, industry R&D commitments

Open research challenges

- Handling changes in data, and model inaccuracy -> Adaptive GNNs
- Curse of dimensionality -> Self-supervised learning
- Addressing risks, confidence, and trust in ML models
- A large amount of data is needed
- Integrating various concepts

Thank you

Speaker

Jochen Cremer

Associate Professor IEPG, TU Delft

Web: <https://www.tudelft.nl/ai/delft-ai-energy-lab>

Personal www.jochen-cremer.com

Email: j.l.cremer@tudelft.nl

Code: <https://github.com/TU-Delft-AI-Energy-Lab>



References & code

Weakly supervised learning for power systems (Example code: <https://github.com/TU-Delft-AI-Energy-Lab/Deep-Statistical-Solver-for-Distribution-System-State-Estimation>)

- Bastien Giraud, Ali Rajaei, Jochen L. Cremer “Constraint-Driven Deep Learning for N-k Security Constrained Optimal Power Flow”, *Electric Power System Research and 2024 IEEE Power System Computation Conference* [Code: <https://github.com/TU-Delft-AI-Energy-Lab/Constraint-Driven-SCOPF>]
- B. Habib, E. Isufi, W. v. Breda, A. Jongepier and Jochen L. Cremer, “Deep Statistical Solver for Distribution System State Estimation,” *IEEE Transactions on Power Systems*, 2023, doi: 10.1109/TPWRS.2023.3290358.

Cost-sensitive learning

- Dariush Wahdany, Carlo Schmitt, Jochen L. Cremer, “More than Accuracy: End-To-End Wind Power Forecasting that Optimises the Energy System”, *Electric Power System Research*, 2023
- A. Bugaje, J. L. Cremer, M. Sun, G. Strbac, “Selecting DT Models for Security Assessment using ROC- and Cost-Curves”, *Energy and AI*, 2021: 100110.
- J. L. Cremer, G. Strbac, “A Machine-learning based Probabilistic Perspective on Dynamic Security Assessment” *International Journal of Electrical Power & Energy System*, 2020

Interpretable models (Example code: <https://github.com/JochenC/From-optimization-based-machine-learning-to-interpretable-security-rules-for-operation>)

- J. L. Cremer, I. Konstantelos, G. Strbac, “From Optimization-based Machine Learning to Interpretable Security Rules for Operation”, *IEEE Transactions on Power Systems*, 2019
- J. L. Cremer, I. Konstantelos, S. H. Tindemans, G. Strbac, “Data-driven Power System Operation: Exploring the Balance between Cost and Risk”, *IEEE Transactions on Power Systems*, 2018

Fast training of models

- Mert Karaçelebi, Jochen L. Cremer “Online Neural Dynamics Forecasting for Power System Security”, *International Journal of Electrical Power & Energy Systems* 2025
- Mert Karaçelebi, Jochen L. Cremer, “Predicting Power System Frequency with Neural Ordinary Differential Equations”, *12th Bulk Power System Dynamics and Control Symposium and Sustainable Energy, Grids and Networks Journal*, 2025

Generalisation challenge:

- Olaiyiwola Arowolo, Jochen Stiasny, Jochen Cremer, “Exploring the Extrapolation Performance of Machine Learning Models for Power System Time Domain Simulations”, *12th Bulk Power System Dynamics and Control Symposium and Sustainable Energy, Grids and Networks Journal*, 2025



Thank you for your attention

Jochen Cremer
j.i.cremer@tudelft.nl

Reliability Indicators

Loss of load expectation (LOLE)

- Expected amount of time demand can not be supplied [h/y]
- $\sum$ (Outage duration x probability)

Expected energy not supplied (EENS)

- Amount of energy expected not to be supplied during that period [MWh/y]
- $\sum$ (Energy not supplied x probability)



$$t_1 * \pi_1 = 2 \text{ h/y}$$

$$E_1 * \pi_1 = 3 \text{ MWh/y}$$



$$t_1 * \pi_1 = 0 \text{ h/y}$$

$$E_1 * \pi_1 = 0 \text{ MWh/y}$$



$$t_1 * \pi_1 = 0 \text{ h/y}$$

$$E_1 * \pi_1 = 0 \text{ MWh/y}$$



$$t_1 * \pi_1 = 3 \text{ h/y}$$

$$E_1 * \pi_1 = 4 \text{ MWh/y}$$

+

$$\text{LOLE} = 5 \text{ h/y}$$

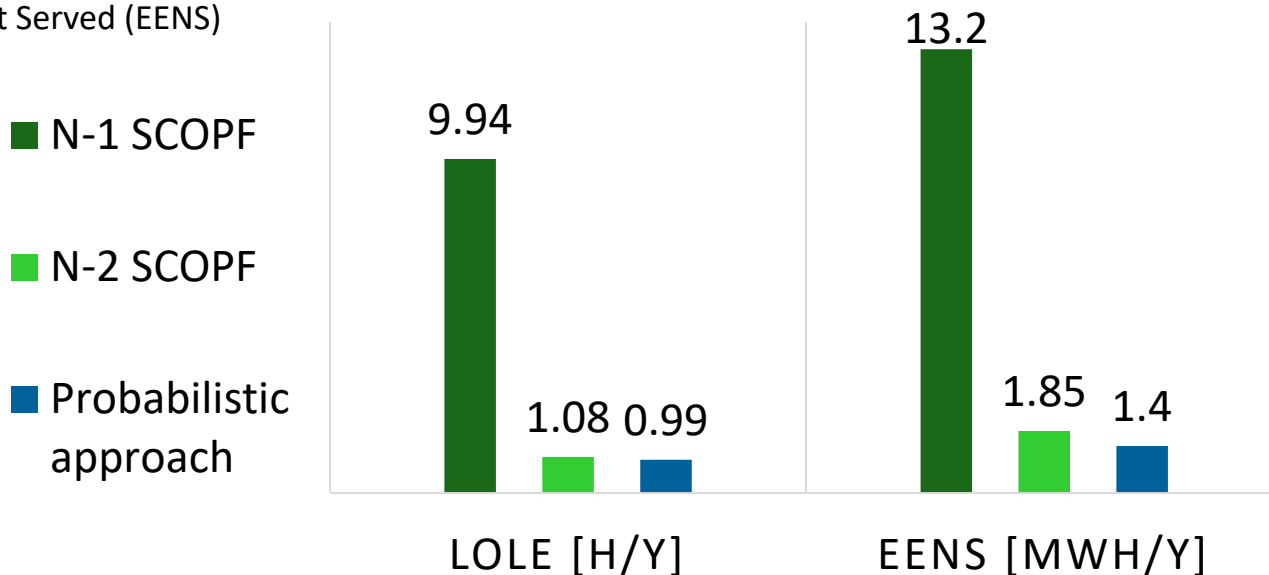
$$\text{EENS} = 7 \text{ MWh/y}$$

Probabilistic security assessment

Proposed probabilistic security enhances reliability

Compare reliability indices

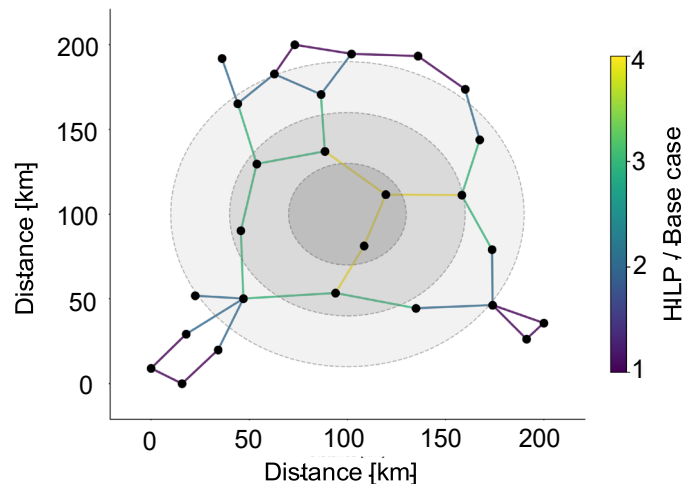
- Loss of Load Expected (LOLE)
- Expected Energy not Served (EENS)



Extreme event

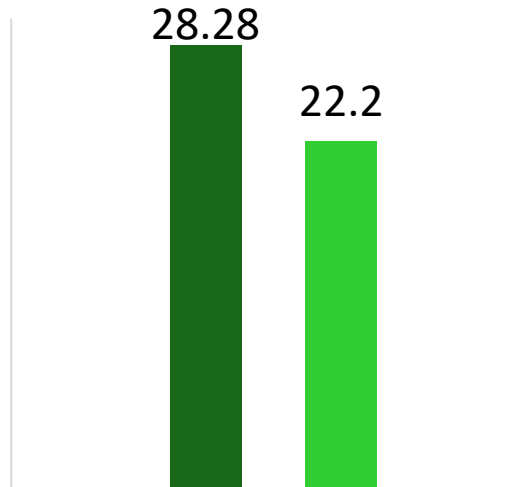
- Individual probabilities change due to an earthquake
- Recompute joint probabilities
- Recompute reliability indices

Potential for increased
resiliency



■ N-2 SCOPF

■ Probabilistic approach



EENS [MWH/Y]

Performance 118-bus system

- Evaluate ability to identify line violations
- **Only** consider single, double or triple line outages
- Post-cont violations [%] indicates the percentage of samples where line violations occur

Proposed approach
Baseline

