Neural Network Verification for Power System Operation

Andreas Venzke, Spyros Chatzivasileiadis
Technical University of Denmark (DTU)

Motivation

- Electric power grid: the **largest machine** humans ever built

- Over the past few years: **explosion of the number of machine learning applications** in power systems
  - (Deep) neural networks, (deep) reinforcement learning, etc.
  - Can handle **high complexity extremely fast** → security assessment **100x -1'000x faster** compared with conventional methods

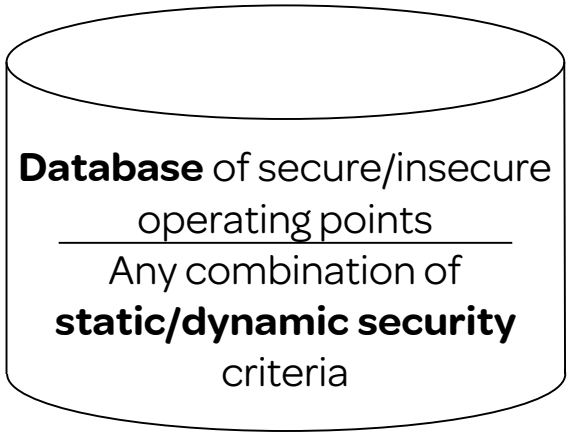
But:

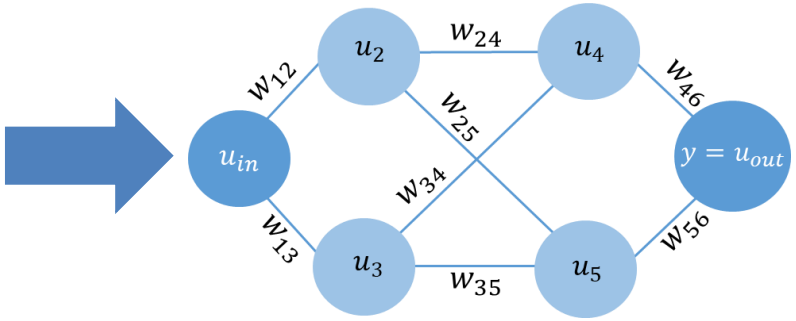
- Power systems are **safety-critical** systems:
  - "Black-box" methods for critical operations will never be adopted (e.g. neural networks for security assessment)

This talk:

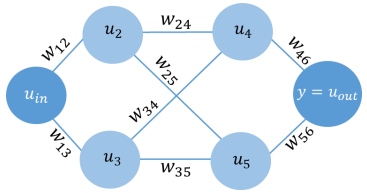
Neural network verification: neural networks are no longer a black-box

Security Assessment with Neural Networks





Use the NN



e.g. N-1 & Small-signal stability (Small-Signal Stab. up to now impossible to *directly* include in an OPF)

Train a neural network → "encode" all information about secure and insecure regions

NN Output:

Binary classification: **secure/insecure**

Neural Network Verification

A. Venzke, S. Chatzivasileiadis. Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications. *Accepted to IEEE Trans. Smartgrid*, 2020. https://arxiv.org/pdf/1910.01624.pdf

Why is neural network verification important?

- Neural networks have been shown to be extremely fast
  - Assess if an operating point is safe or unsafe 100x-1'000x faster (combination of different security criteria)
  - Application in optimization: find the optimal point >100x faster

- However, neural networks will never be applied in critical (power system) operations, if there are no guarantees about how they behave

- Until recently, the only way to assess the output of the neural networks was to **individually test** each input of interest and pass it through the neural network
  - Accuracy was purely statistical
  - **Challenge #1:** No way to guarantee what the output is for a **continuous** range of inputs

Evaluating Accuracy: Test Database

Traditionally:

- Split training database to e.g. 80% training samples and 20% test samples
- Train with the 80%
- Test with the 20%

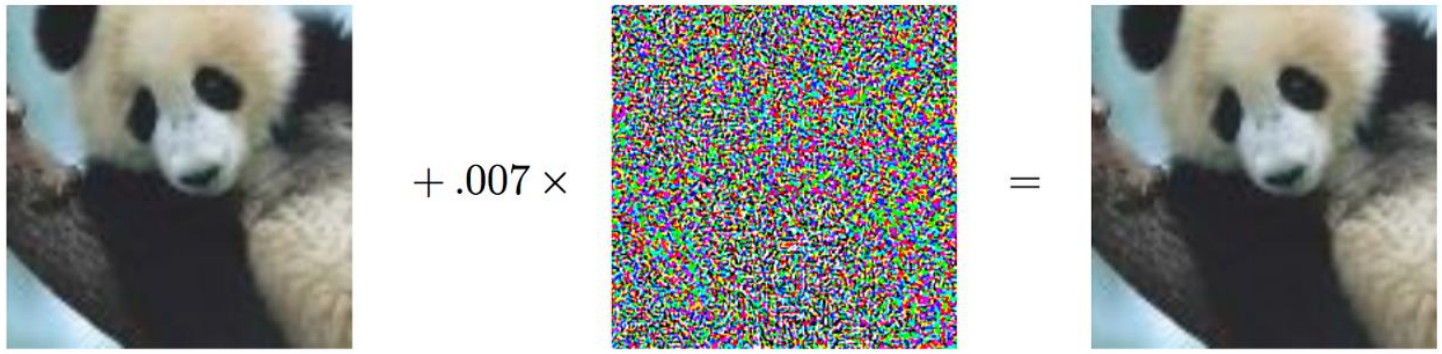
Modern toolboxes have this integrated and automatized →only need to provide a training database

Challenge #2:
The test database determines the performance of the neural network. If the test data come from the same simulations as your training data, the accuracy can be deceivingly high. Would it be equally high in reality?

Ideally → use a different real-life dataset
Neural network verification overcomes this challenge too

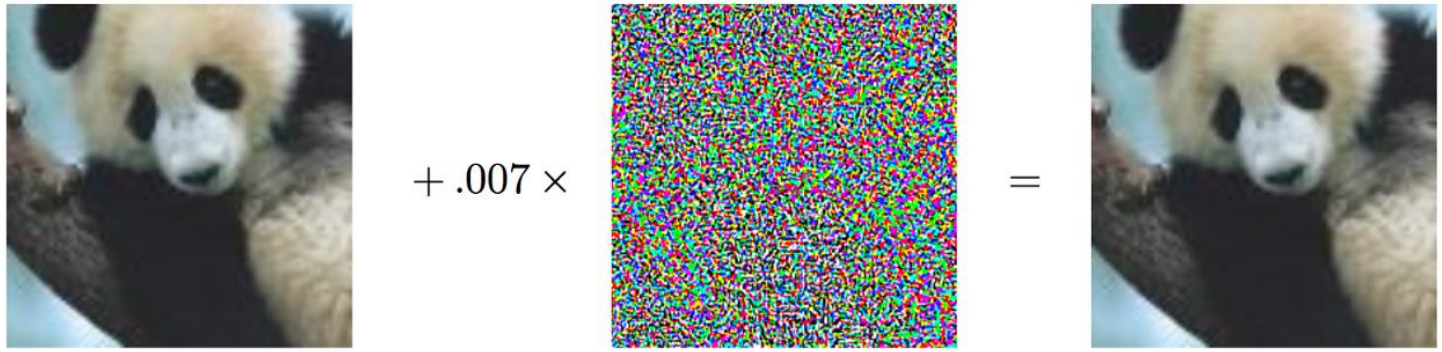
Adversarial examples

- Adversarial example: small perturbations lead to a false prediction



Adversarial examples

- Adversarial example: small perturbations lead to a false prediction



- **Challenge #3: No way to systematically identify adversarial examples**

Neural Network Verification: HOW?

1. Convert the neural network to a **set of linear equations with binaries**
    - The Neural Network can be included in a mixed-integer linear program

2. Formulate an optimization problem (MILP)

3. Solve the MILP to zero duality gap (find the global optimal) → certificate for the behavior for neural network

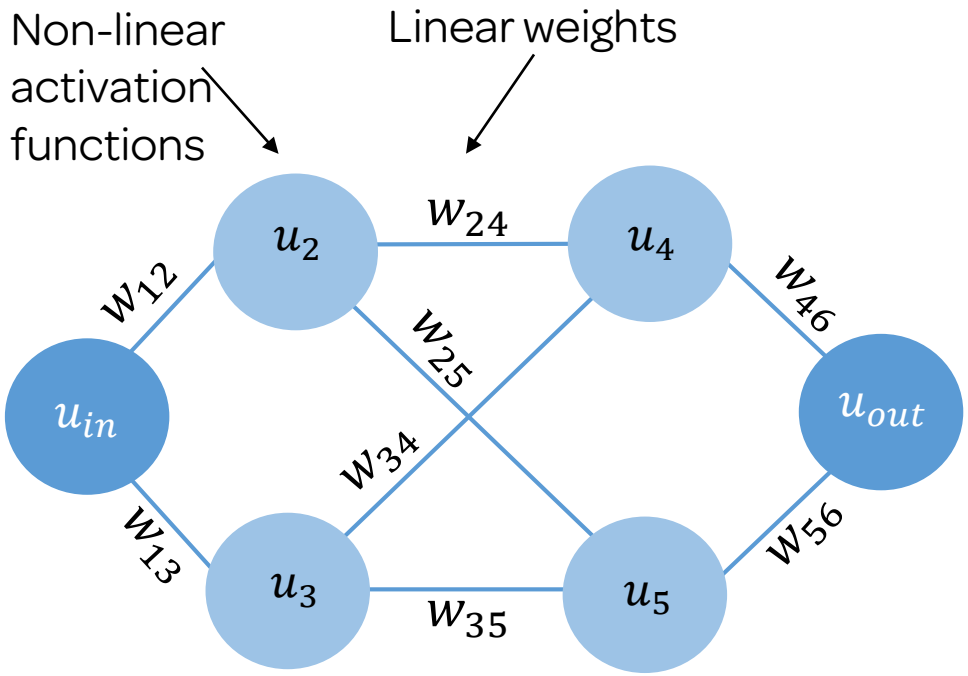
4. Assess if the neural network output complies with the ground truth

Two types of optimization problems:

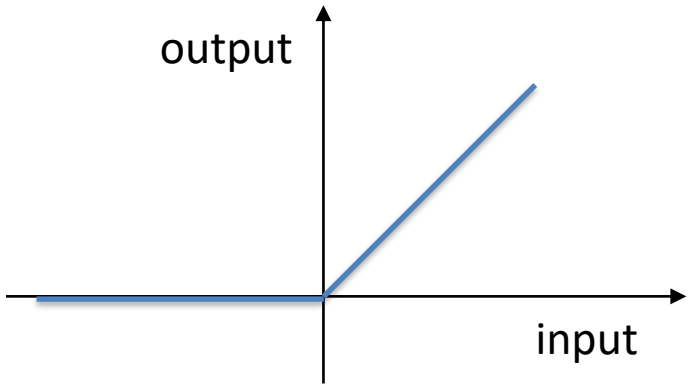
A. Certify that in a region around a given input x_{ref} the neural network maintains the same classification → guarantee that all input points (continuous range) in the neighborhood will be classified the same

B. Find the minimum distance from x_{ref} that the classification changes (possibility of adversarial examples)

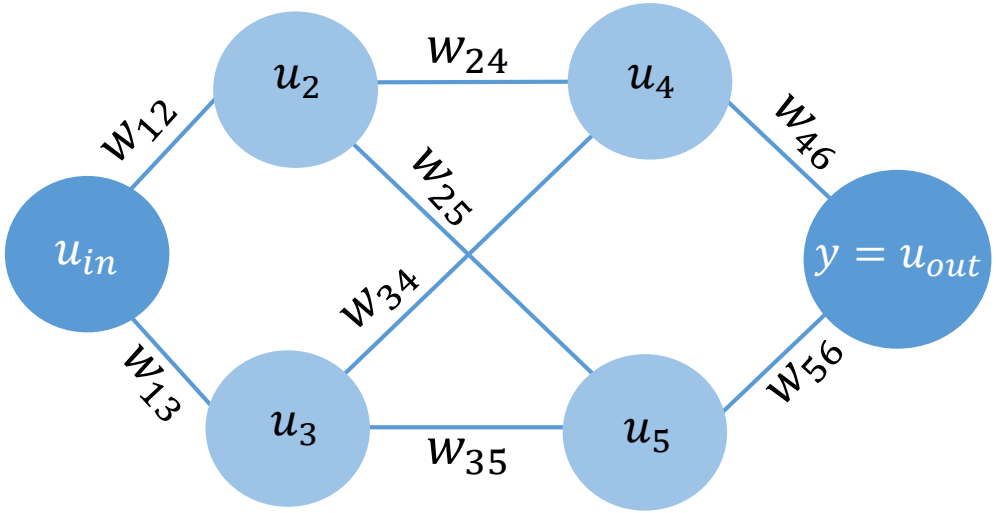
From Neural Networks to Mixed-Integer Linear Programming



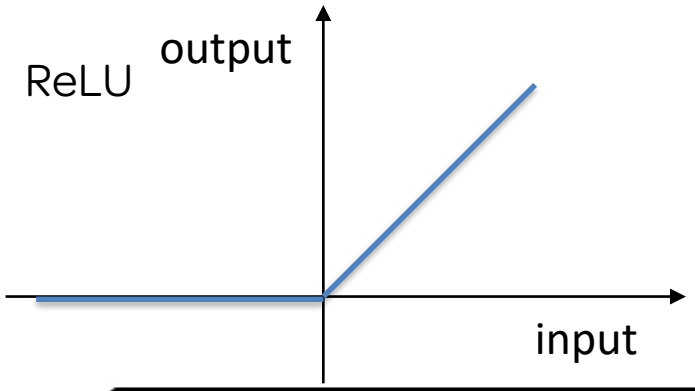
- Most usual activation function: ReLU
- ReLU: Rectifier Linear Unit



From Neural Networks to Mixed-Integer Linear Programming

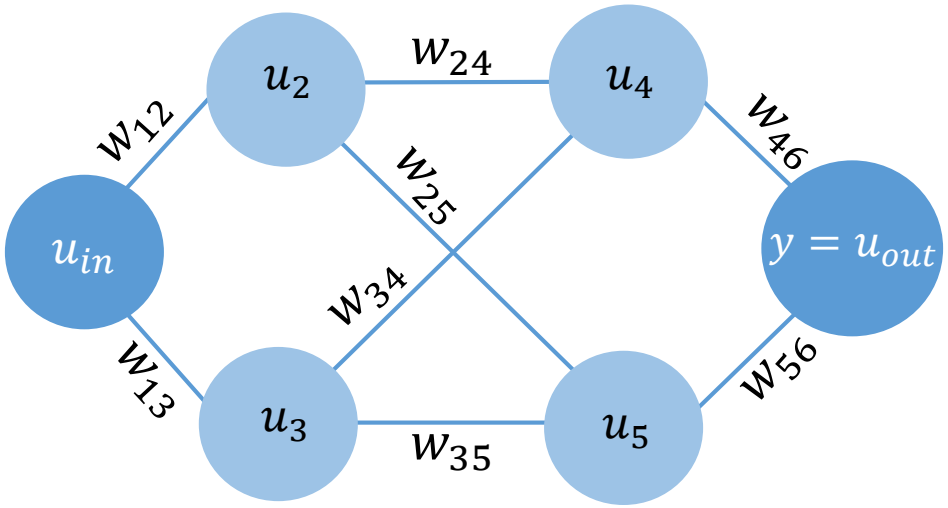


- Linear weights
- On every node: a non-linear activation function
  - ReLU: $\boldsymbol{u_j = \max(0, w_{ij}u_i + b_i)}$
- But ReLU can be transformed to a piecewise linear function with binaries



MILP

From Neural Networks to Mixed-Integer Linear Programming

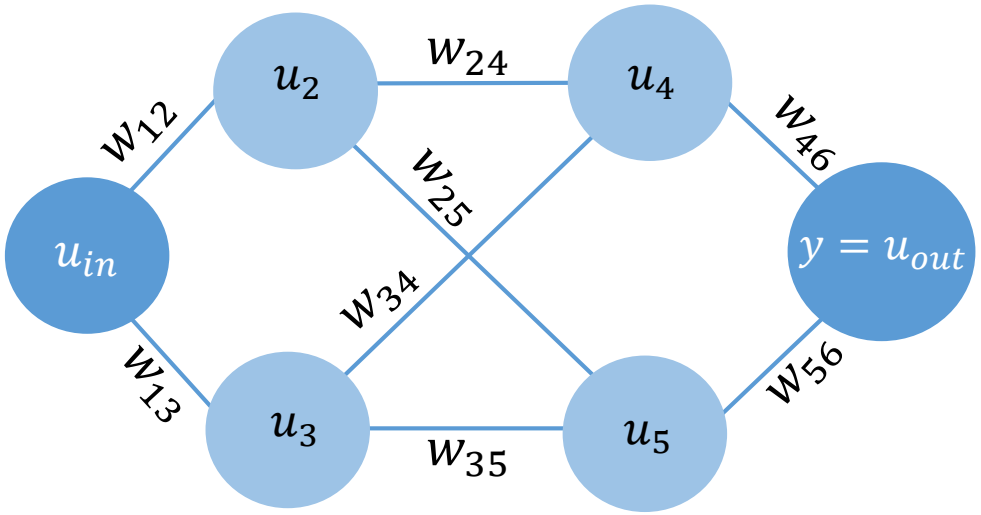


$$y = NN(x)$$

$$\begin{aligned}
&\hat{\mathbf{u}}_\mathbf{k} = \mathbf{W}_k \mathbf{u}_{\mathbf{k-1}} + \mathbf{b}_\mathbf{k} \\
&\mathbf{u}_k = \max(\hat{\mathbf{u}}_k, 0) \Rightarrow \\
&\mathbf{y} = \mathbf{u}_{out}
\end{aligned}
\begin{cases}
y_k \leq \hat{u}_k - \hat{u}_k^{\min}(1 - b_k) \\
u_k \geq \hat{u}_k \\
u_k \leq \hat{u}_k^{\max} b_k \\
u_k \geq 0 \\
b_k \in \{0,1\}^{N_k}
\end{cases}$$

Exact reformulation to MILP

From Neural Networks to Mixed-Integer Linear Programming



- Input: Active power gen. setpoints

$$\mathbf{x} = [p_{g1}, p_{gi}, \ldots, p_{gN}]^T$$

- Output
  - Binary classification: safe/unsafe
  - Output vector y with two elements:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

- $y_1 \geq y_2$: safe
- $y_1 < y_2$: unsafe

Certify the output for a continuous range of inputs

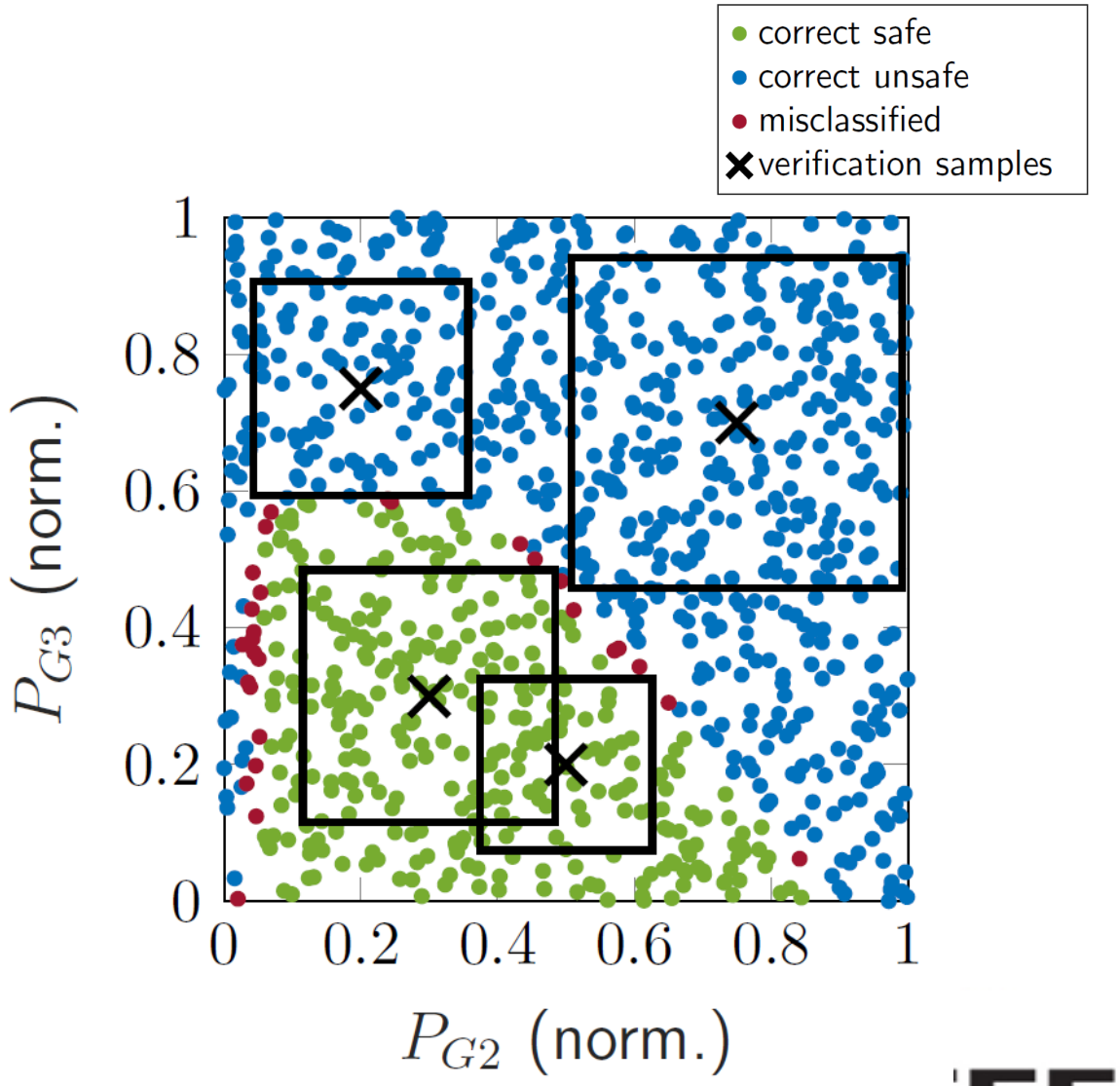
- We assume a given input x_{ref} with classification y: $y_1 > y_2$

1. For distance ϵ evaluate if input x exists with different classification y_2

$$\max_{x,y} \quad \mathbf{y_2 - y_1}$$

$$\text{s.t.} \quad y = NN(x)$$

$$|\mathbf{x - x_{ref}}|_\infty \leq \epsilon$$



Examples of NN verification problems

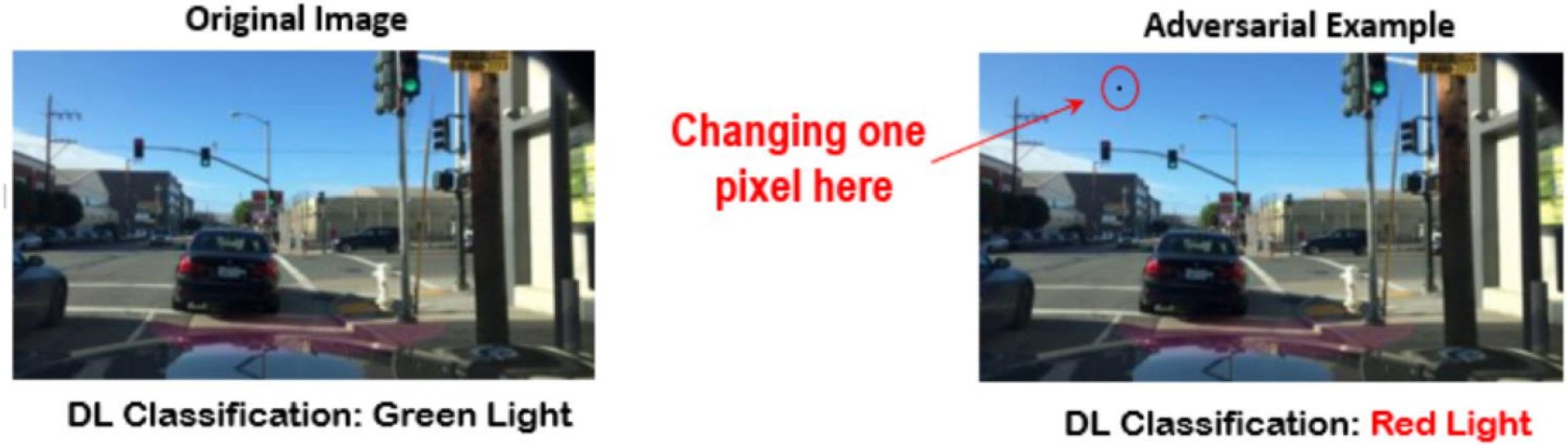
1. Find the **minimum distance** from the zero input $\boldsymbol{x_{ref} = 0}$ that classification becomes **secure**

	Actual insecure region around $x_{ref} = 0$ (ground truth)	NN verification (80% sparsity, and power balance constraint)
162-bus:	66.2%	65.5%

2. Find the **maximum wind infeed** for which the NN determines a **safe dispatch**

	Actual maximum wind infeed (ground truth)	NN verification (80% sparsity, and power balance constraint)
9-bus:	92.5%	89%

Adversarial examples in safety-critical systems



source: Wu et al. A game-based approximate verification of deep neural networks with provable guarantees. arXiv:1807.03571.

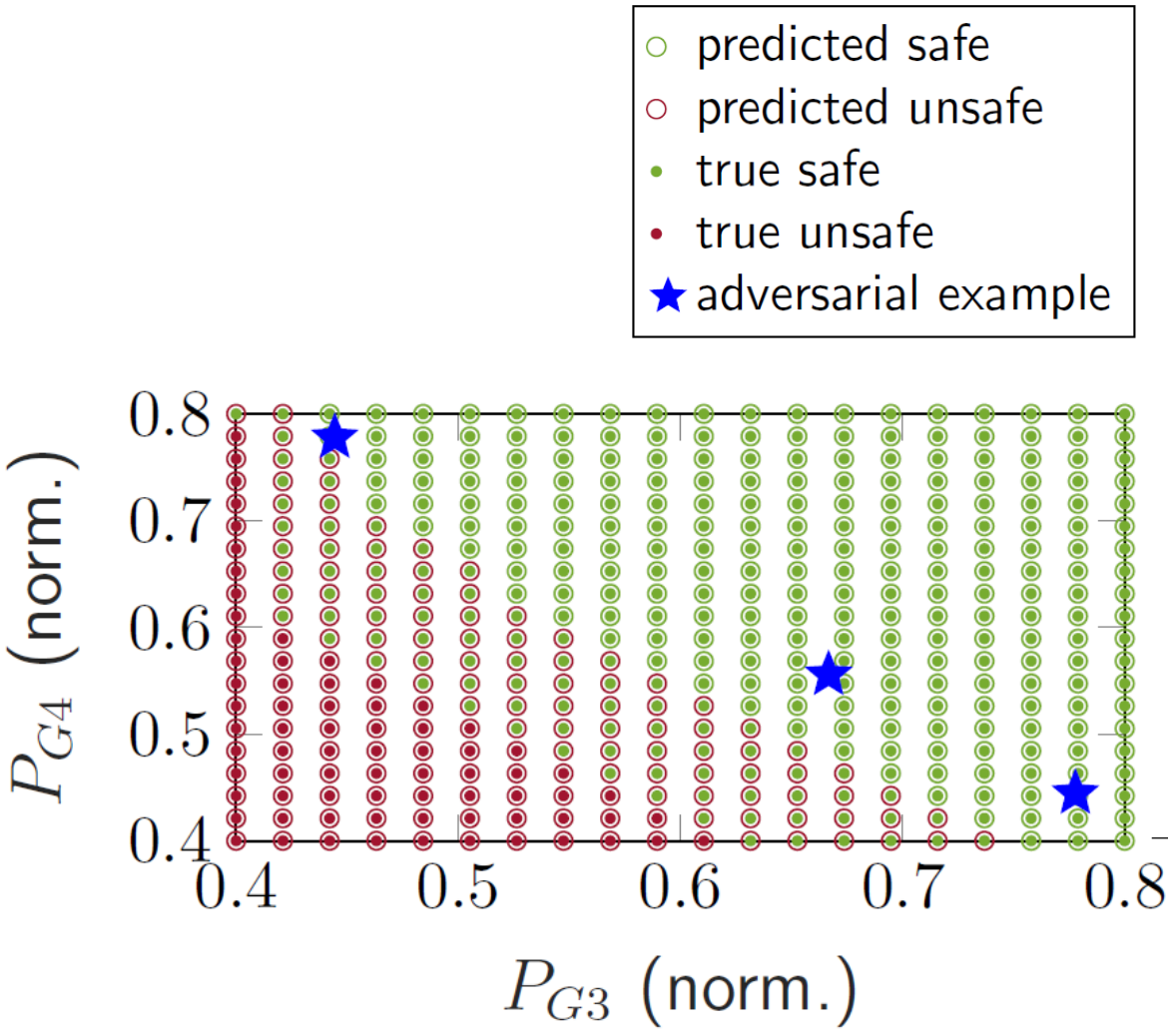
- Adversarial examples exist in many (deep) learning applications
- Major barrier for adoption of machine learning techniques in safety-critical systems!

Systematically identify adversarial examples

- We assume a given input x_{ref} with classification y: $y_1 > y_2$

2. Minimize distance ϵ from $\mathbf{x}_{\text{ref}}$ to input $\mathbf{x}$ with classification y_2

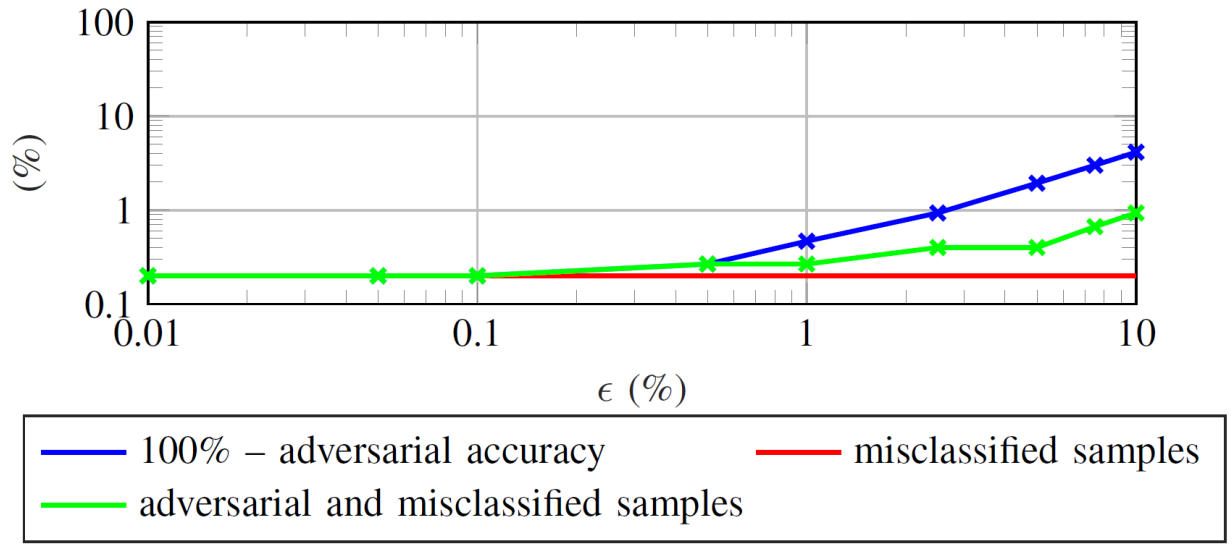
$$\begin{aligned} \min_{\mathbf{x},\mathbf{y},\epsilon} \quad & \epsilon \\ \text{s.t.} \quad & \mathbf{y} = NN(\mathbf{x}) \\ & |\mathbf{x} - \mathbf{x}_{\text{ref}}|_\infty \leq \epsilon \\ & y_2 \geq y_1 \end{aligned}$$



Systematically identify adversarial robustness, when NN accuracy is not a dependable metric

N-1 and Small-signal stability assessment for the **162-bus system**

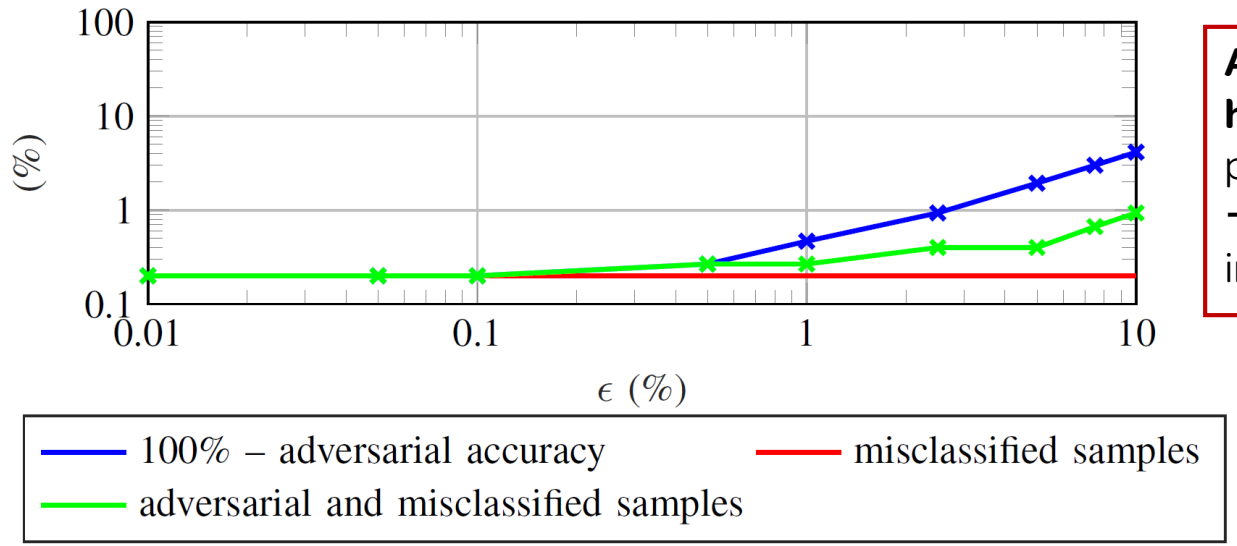
- **Highly unbalanced dataset**: only 1.36% of the samples in the secure region
- NN accuracy evaluates to 99%, even if all the 'safe' samples are misclassified as 'unsafe'
- Rigorous methods to identify adversarial robustness are necessary



Systematically identify adversarial robustness, when NN accuracy is not a dependable metric

N-1 and Small-signal stability assessment for the **162-bus system**

- **Highly unbalanced dataset**: only 1.36% of the samples in the secure region
- NN accuracy evaluates to 99%, even if all the 'safe' samples are misclassified as 'unsafe'
- Rigorous methods to identify adversarial robustness are necessary



Adversarial robustness can also help us determine the NN performance **for uncertain inputs** → e.g. how the NN performs if wind infeed changes within range ε

Challenges

- **Tractability** for large neural networks
  - Up to now, we have verified NNs with 4 layers and 100 nodes at each layer (NN used for the 162-bus system)
  - We require weight sparsification, bound tightening, and ReLU pruning (remove binary variables) to maintain tractability

- **Connect verification with ground truth assessment**
  - Currently, we can first certify the neural network output, and we should then assess if this output is correct (i.e. that the NN can be trusted in real operation)
  - Now working on a verification procedure that will be integrated in the training of the neural network → NN training will offer a certificate of performance (no more statistics!)

- **Retraining** is necessary to avoid adversarial examples
  - The **quality of the training database is crucial** for good performance!

Wrap-up: Neural network verification

- NN verification **unlocks a world of new opportunities** for practical applications in power systems
  - **Removes the barrier** for neural network applications in safety-critical power system operations

- **Certify the behavior of neural networks**: formal guarantees

- **Systematically identify adversarial examples**

Thank you!

Spyros Chatzivasileiadis
Associate Professor, PhD
www.chatziva.com
spchatz@elektro.dtu.dk

Neural Network Verification:
A. Venzke, S. Chatzivasileiadis. Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications. *Accepted to IEEE Trans. Smartgrid*, 2020.
https://arxiv.org/pdf/1910.01624.pdf

From Neural Networks and Decision Trees to MILP:
A. Venzke, D. T. Viola, J. Mermet-Guyennet, G. S. Misyris, S. Chatzivasileiadis. Neural Networks for Encoding Dynamic Security-Constrained Optimal Power Flow to Mixed-Integer Linear Programs. 2020. https://arxiv.org/pdf/2003.07939.pdf
L. Halilbašić, F. Thams, A. Venzke, S. Chatzivasileiadis, and P. Pinson, "Data-driven security-constrained AC-OPF for operations and markets," *PSCC* 2018. [.pdf]
See presentation video of 20PESGM2230 (panel session "Learning to Optimize Transmission Systems")

Physics-Informed Neural Networks:
G. S. Misyris, A. Venzke, S. Chatzivasileiadis, Physics-Informed Neural Networks for Power Systems. Accepted at IEEE PES GM 2020. https://arxiv.org/pdf/1911.03737.pdf
J. Stiasny, G. S. Misyris, S. Chatzivasileiadis, Physics-Informed Neural Networks for Non-linear System Identification applied to Power System Dynamics. 2020. https://arxiv.org/pdf/2004.04026.pdf
See presentation video of 20PESGM1831 ("Best Conference Papers on Planning, Operation, and Energy Markets")

Some code available at:
www.chatziva.com/downloads.html