2021 IEEE Workshop on
Machine Learning for Power Systems

Neural Network Applications for Power Systems: Is it time to trust them?

Spyros Chatzivasileiadis
Associate Professor, DTU

Joint work with Andreas Venzke, Georgios Misyris, Jochen Stiasny, Rahul Nellikkath
and with Guannan Qu, Steven Low

IEEE PES Power & Energy Society®

IEEE

Machine Learning for Power Systems: Why?

1. **Extremely fast**
   - computation within only a **few milliseconds**

   100x – 1000x faster than conventional methods → we can run 1’000 possible scenarios and get a good estimate vs. running only 1 scenario

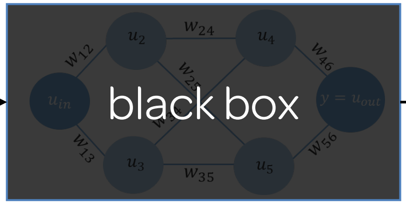
2. Good alternative if we do not have full knowledge of the actual model
   - Handle **very complex systems**
   - **Infer** from incomplete data

But: Would an Operator ever trust AI in the Control Room?

Our work: Remove the barriers and build **trustworthy AI-tools**

ML Barriers for Power systems

Input:
Operating point



secure/ insecure

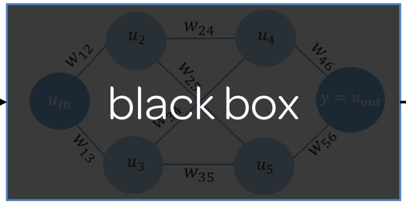
1. Why would we use a "**black box**" to decide about **a safety-critical application**?

2. **Accuracy is** a **purely statistical** performance metric. Who guarantees that the Neural Network can handle well previously unseen operating points?

3. Why would we depend on **discrete and incomplete data**, when we have developed **detailed physical models** over the past 100 years?

ML Barriers for Power systems

Input:
Operating point



secure/
insecure

1. Why would we use a "**black box**" to decide about **a safety-critical application**?

2. **Accuracy is** a **purely statistical** performance metric. Who guarantees that the Neural Network can handle well previously unseen operating points?

3. Why would we depend on **discrete and incomplete data**, when we have developed **detailed physical models** over the past 100 years?

Neural Network verification: guarantees for the NN performance!

→ *Remove dependence on the test database*

Physics-Informed Neural Networks

Neural Network Verification

for classification NNs in Power Systems

A. Venzke, S. Chatzivasileiadis. Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications. In *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 383-397, Jan. 2021, https://arxiv.org/pdf/1910.01624.pdf

V. Tjeng, K. Y. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," in International Conference on Learning Representations (ICLR 2019), 2019

Guiding Application:
Security Assessment with Neural Networks

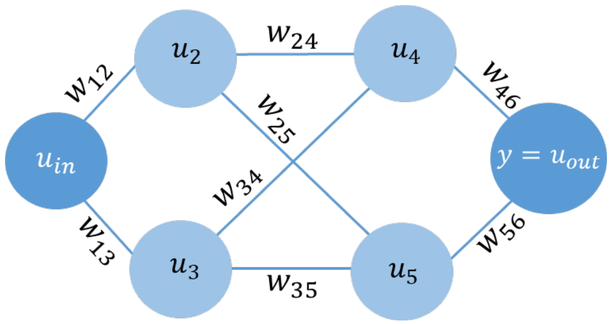
Approaches proposed up to now

Database of secure/insecure operating points

Any combination of **static/dynamic security** criteria

(e.g. N-1 & Small-signal stability)

1. Split the database in a training set and a test set



2. Train a neural network

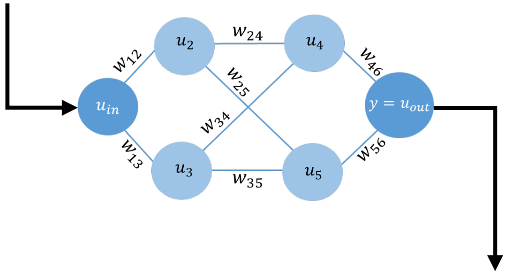
3. Test the neural network

4. Is accuracy high enough?

5. Use the NN

Input:

Operating point



secure/
insecure

NN Output:

Binary classification: **secure/insecure**

Extremely fast: up to 100x-1'000x faster

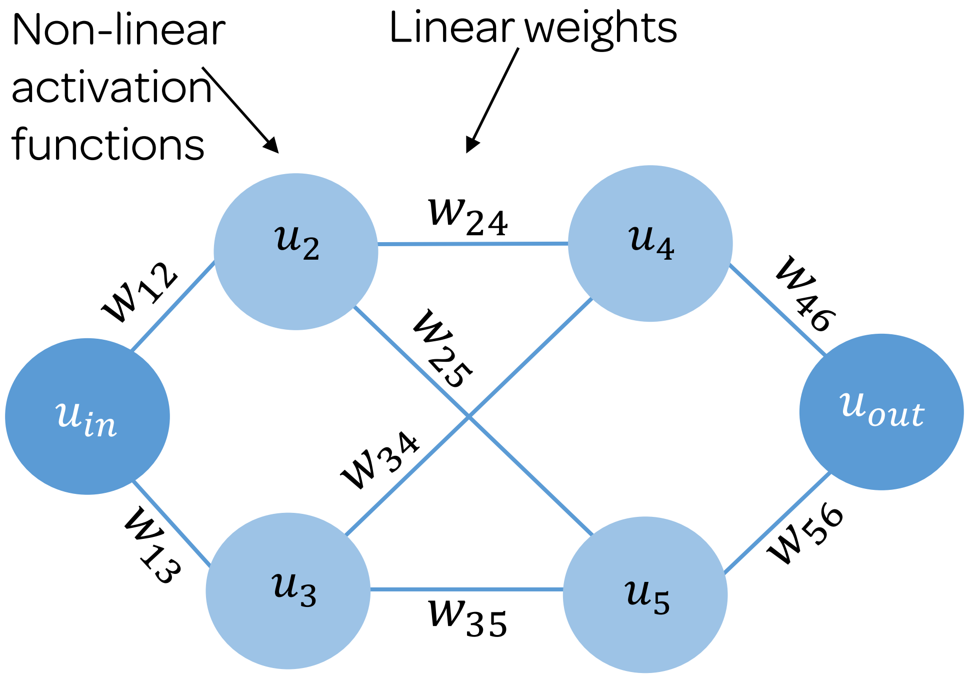
Neural Network Verification: HOW?

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

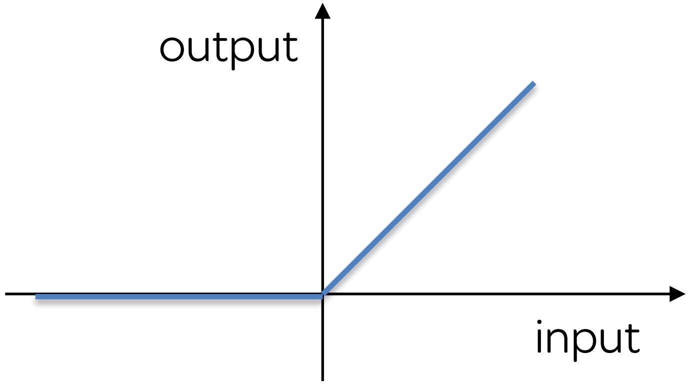
2. Formulate an **optimization** problem (MILP) and solve it → certificate for NN behavior

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

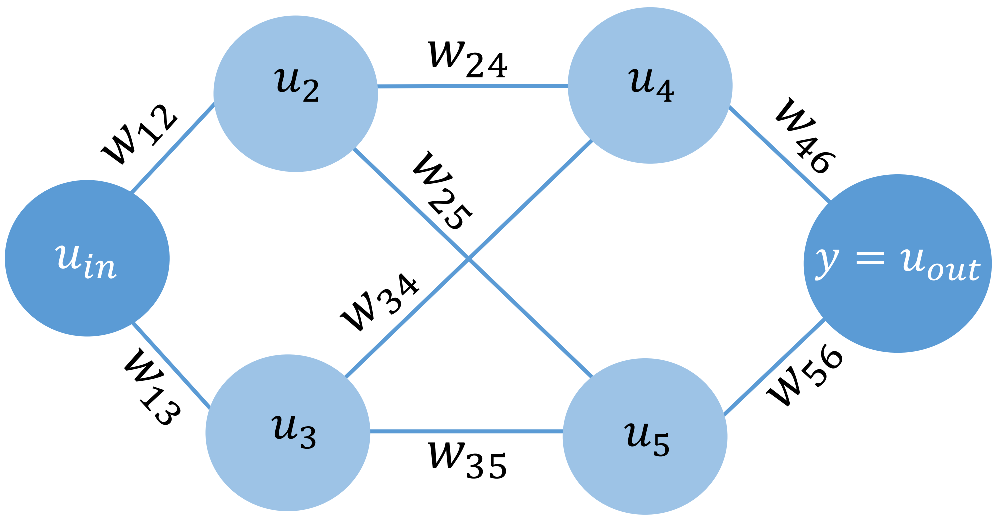
From Neural Networks to Mixed-Integer Linear Programming

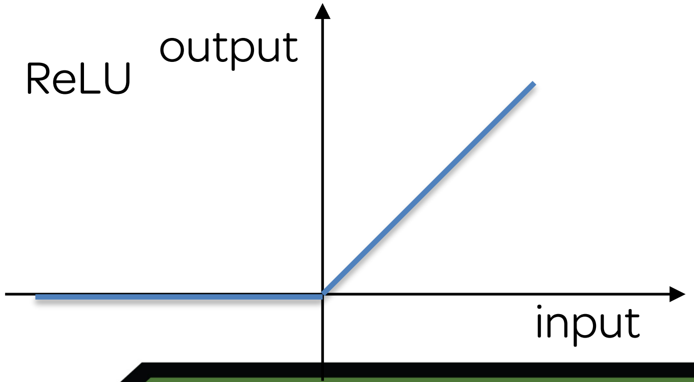


- Most usual activation function: ReLU
- ReLU: Rectifier Linear Unit



From Neural Networks to Mixed-Integer Linear Programming



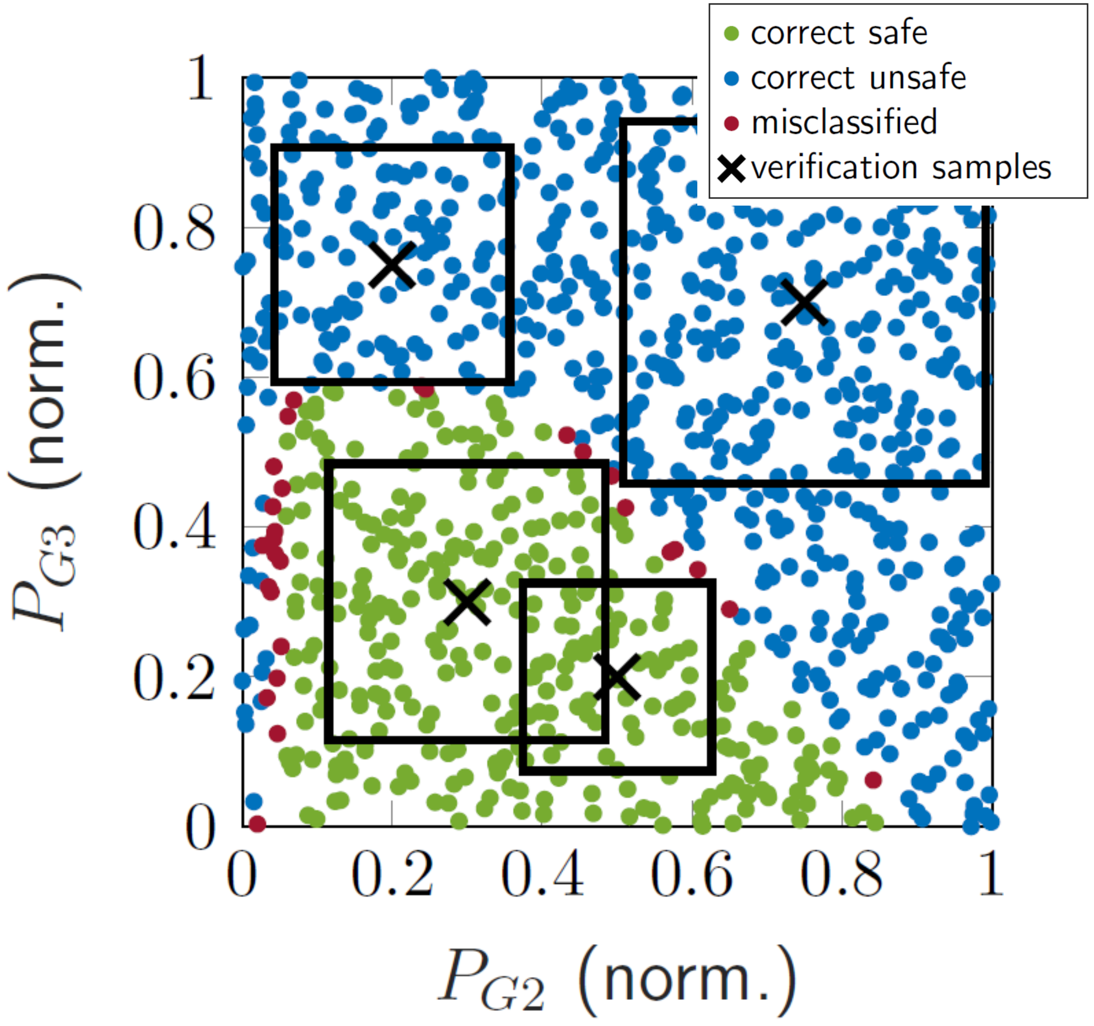


- Linear weights
- On every node: a non-linear activation function
  - ReLU: $\boldsymbol{u_j = \max(0, w_{ij}u_i + b_i)}$
- But ReLU can be transformed to a piecewise linear function with binaries

⬇

MILP

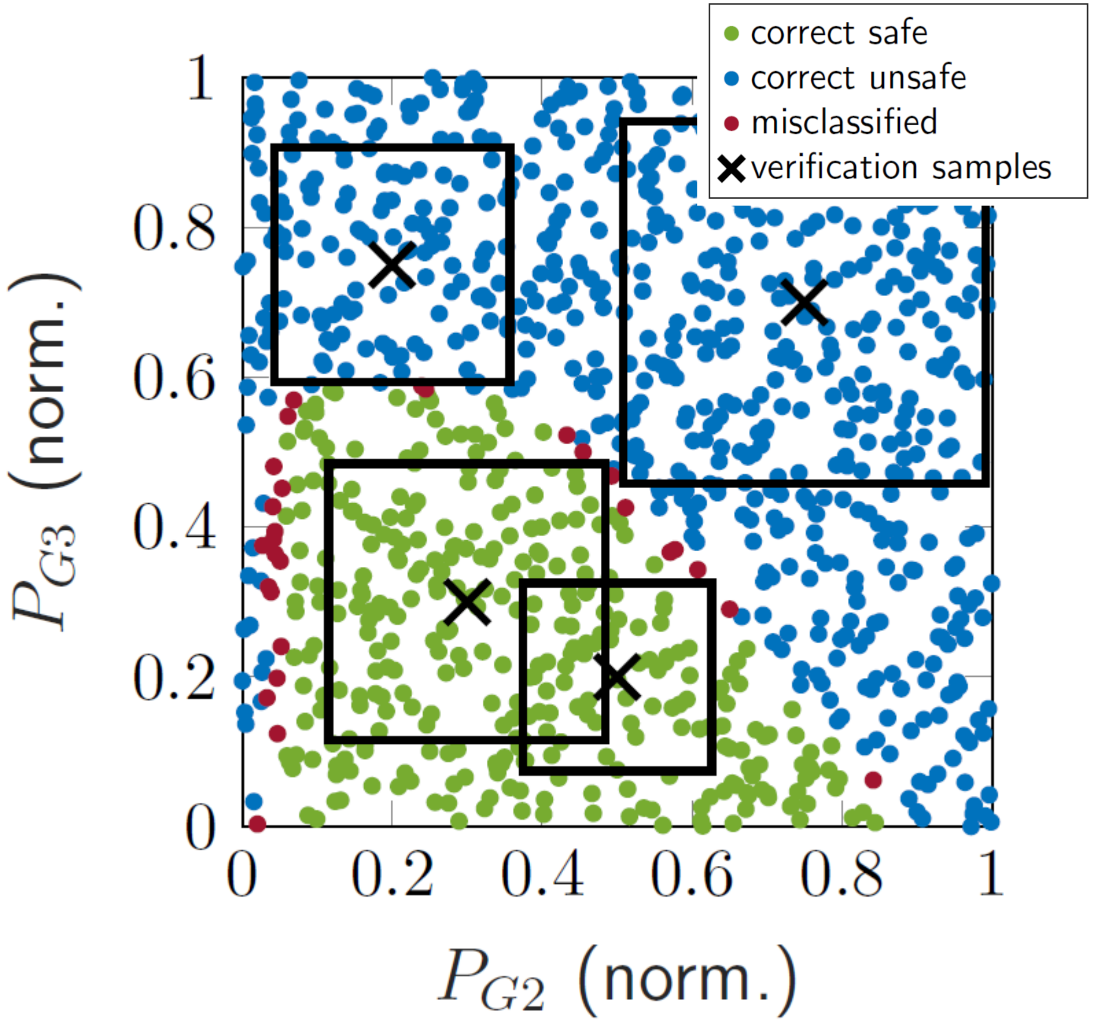
Certify the output for a continuous range of inputs



1. We assume a given input x_{ref} with classification “safe”

A. Venzke, S. Chatzivasileiadis. Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications. *IEEE Transactions on Smart Grid*, Jan. 2021. https://arxiv.org/pdf/1910.01624.pdf

Certify the output for a continuous range of inputs



1. We assume a given input x_{ref} with classification "safe"
2. Solve optimization problem: **Does classification change for *any* input within distance ε from $\mathbf{x_{ref}}$**?
3. If not, then **I can certify** that my neural network will classify the whole continuous region as "safe"
4. I can repeat this for other regions and different classifications

A. Venzke, S. Chatzivasileiadis. Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications. *IEEE Transactions on Smart Grid*, Jan. 2021. https://arxiv.org/pdf/1910.01624.pdf

Learning OPF:

Worst-case Guarantees for Regression Neural Networks

A. Venzke, G. Qu, S. Low, S. Chatzivasileiadis, Learning Optimal Power Flow: Worst-case Guarantees for Neural Networks. **Best Student Paper Award** at IEEE SmartGridComm 2020. https://arxiv.org/pdf/2006.11029.pdf

R. Nellikkath, S. Chatzivasileiadis, Physics-Informed Neural Networks for AC Optimal Power Flow
https://arxiv.org/abs/2110.02672 [code]

Quick Reminder: DC Optimal Power Flow

- **Objective**: find the minimum cost generation dispatch
- **Input**: Varying load demand at different nodes

- Considered constant: generator costs; system topology

OPF

Minimize Total Generation Cost

Subject to:

Total supply = Total load demand

Transmission line limits

Generator limits

Quick Reminder: DC Optimal Power Flow

- **Objective**: find the minimum cost generation dispatch
- **Input**: Varying load demand at different nodes

- Considered constant: generator costs; system topology

Several recent approaches in the literature that apply Neural Networks for solving the DC-OPF
- Demonstrate up to **100x speedup**
- But **no performance guarantees**

OPF

Minimize Total Generation Cost

Subject to:

Total supply = Total load demand

Transmission line limits

Generator limits

Provable Worst Case Guarantees for Neural Networks

- We developed an **Optimization Algorithm** that determines the worst-case violations of any neural network applied to such problems

- We **integrate**:
  - Neural network equations
  - **Physical equations**

- **Provable worst-case guarantees** of neural network performance

OPF

Minimize Total Generation Cost

Subject to:

Total supply = Total load demand

Transmission line limits

Generator limits

Worst violation over the **whole training dataset** (training+test set)

Our algorithm: **provable** worst-case guarantee over the **whole input domain**

	Empirical lower bound		Exact worst-case guarantee	
Test cases	ν_{g} (MW)	ν_{line} (MW)	ν_{g} (MW)	ν_{line} (MW)
case9				
case30				
case39				
case57				
case118				
case162				
case300				

ν_{g} Maximum violation of generator limits

ν_{line} Maximum violation of line limits

IEEE PES Power & Energy Society®

IEEE

Worst violation over the **whole training dataset** (training+test set)

Our algorithm: **provable** worst-case guarantee over the **whole input domain**

	Empirical lower bound		Exact worst-case guarantee	
Test cases	ν_g (MW)	ν_{line} (MW)	ν_g (MW)	ν_{line} (MW)
case9	2.5	1.8	2.8	1.9
case30	1.7	0.6	3.6	3.1
case39	51.9	37.2	270.6	120.0
case57	4.2	0.0	23.7	0.0
case118	149.4	15.6	997.8	510.8
case162	228.0	180.0	1563.3	974.1
case300	474.5	692.7	3658.5	3449.3

ν_g Maximum violation of generator limits

ν_{line} Maximum violation of line limits

Over the whole input domain **violations can be much larger** (here ~7x) compared to what has been estimated empirically on the dataset

IEEE PES Power & Energy Society®

IEEE

Worst violation over the **whole training dataset** (training+test set)

Our algorithm: **provable** worst-case guarantee over the **whole input domain**

	Empirical lower bound		Exact worst-case guarantee	
Test cases	ν_{g} (MW)	ν_{line} (MW)	ν_{g} (MW)	ν_{line} (MW)
case9	2.5	1.8	2.8	1.9
case30	1.7	0.6	3.6	3.1
case39	51.9	37.2	270.6	120.0
case57	4.2	0.0	23.7	0.0
case118	149.4	15.6	997.8	510.8
case162	228.0	180.0	1563.3	974.1
case300	474.5	692.7	3658.5	3449.3

ν_{g} Maximum violation of generator limits

ν_{line} Maximum violation of line limits

Our method provides **guarantees that no NN output will violate the line limits** over the whole input domain

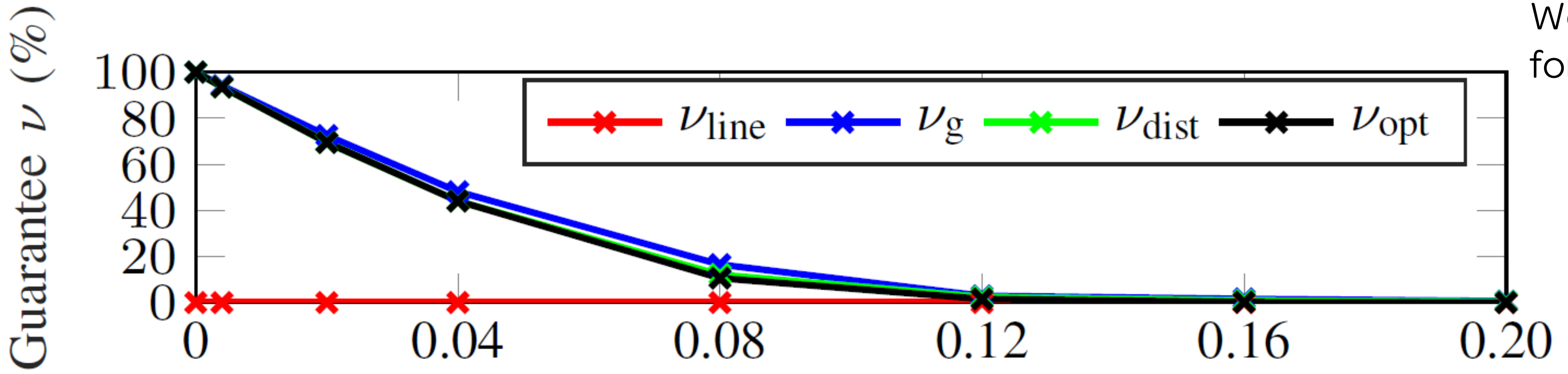
How can we reduce the worst-case violations?

- From our experiments with DC-OPF in 7 different test power systems, we observed that the **worst-case violations occur at the boundary of the input domain**

- Possible solution:

1. Train on a larger input domain
2. Use the NN on a subdomain of the original training input

Reducing the worst-case violations



(b) *case57*: Input domain reduction δ (–)

100% on y-axis = Worst-case violation for $\delta = 0$

- Input domain used for training

$$0.6\,\boldsymbol{p}_{\text{d}}^{\text{max}} \le \boldsymbol{p}_{\text{d}} \le 1.0\,\boldsymbol{p}_{\text{d}}^{\text{max}}$$

- Input domain for using the NN (and where worst-case violations were evaluated)

$$(0.6+\delta)\mathbf{p}_{\text{d}}^{\text{max}} \le \mathbf{p}_{\text{d}} \le (1.0-\delta)\mathbf{p}_{\text{d}}^{\text{max}}$$

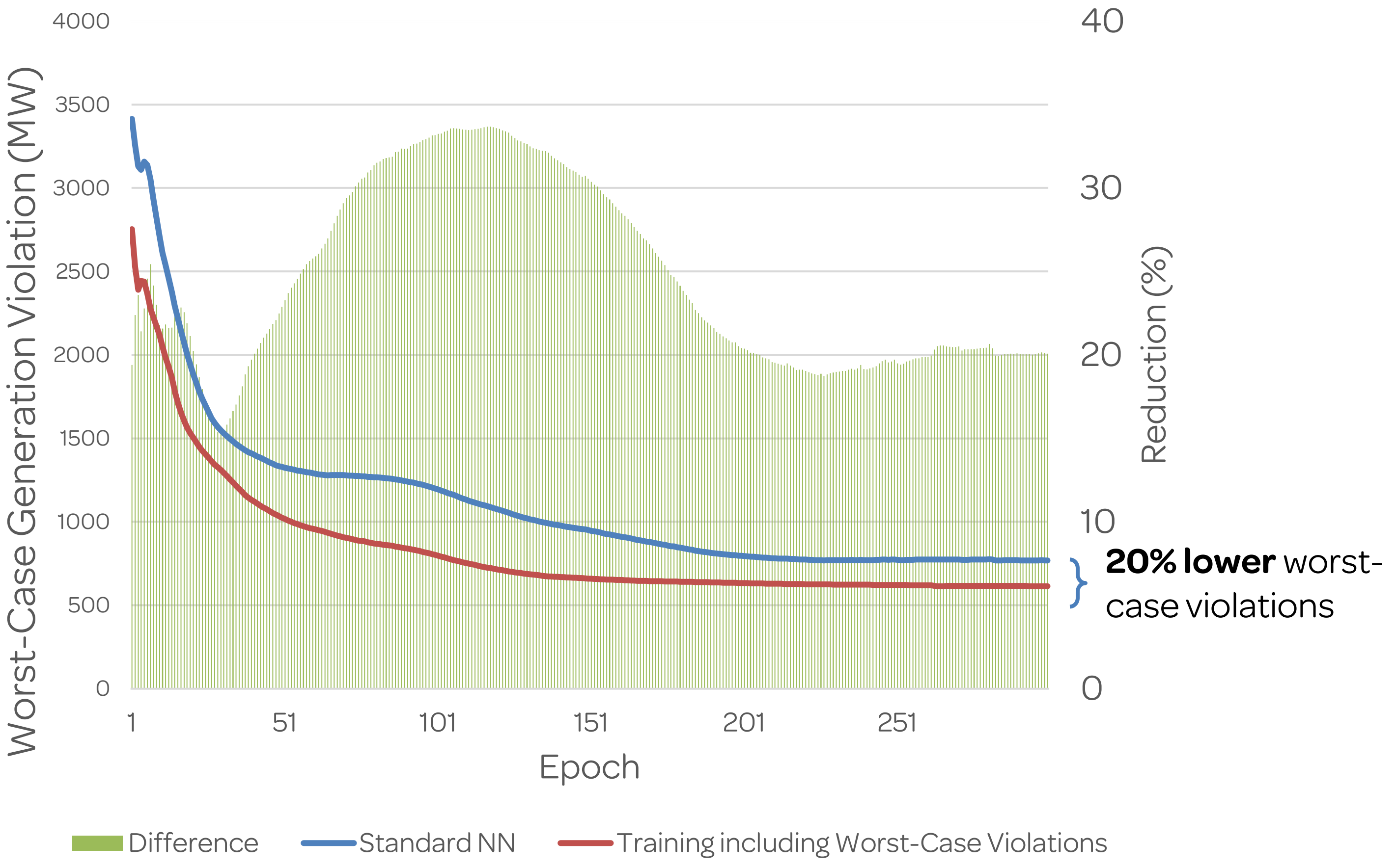
Example: If $\delta = 0.1$ then $0.7\,\boldsymbol{p}_{\text{d}}^{\text{max}} \le \boldsymbol{p}_{\text{d}} \le 0.9\,\boldsymbol{p}_{\text{d}}^{\text{max}}$

Next Step: Integrate the Worst-Case Guarantees <u>inside</u> the Neural Network Training

Preliminary Results

- Approach:
  1. Train for Epoch 1 (standard NN training, i.e. reduce mean absolute error)
  2. Determine the worst-case violations
  3. This is an optimization problem → I can **determine the gradients of the weights and biases to reduce the worst-case violations**
     - Steps 2 and 3 are completely independent of the training datasets. They consider the whole input region.
  4. **Update** the NN weights and biases based on the gradients of Step 3
  5. Train for Epoch 2 (standard NN training, i.e. reduce mean absolute error)
  6. Go to Step 2

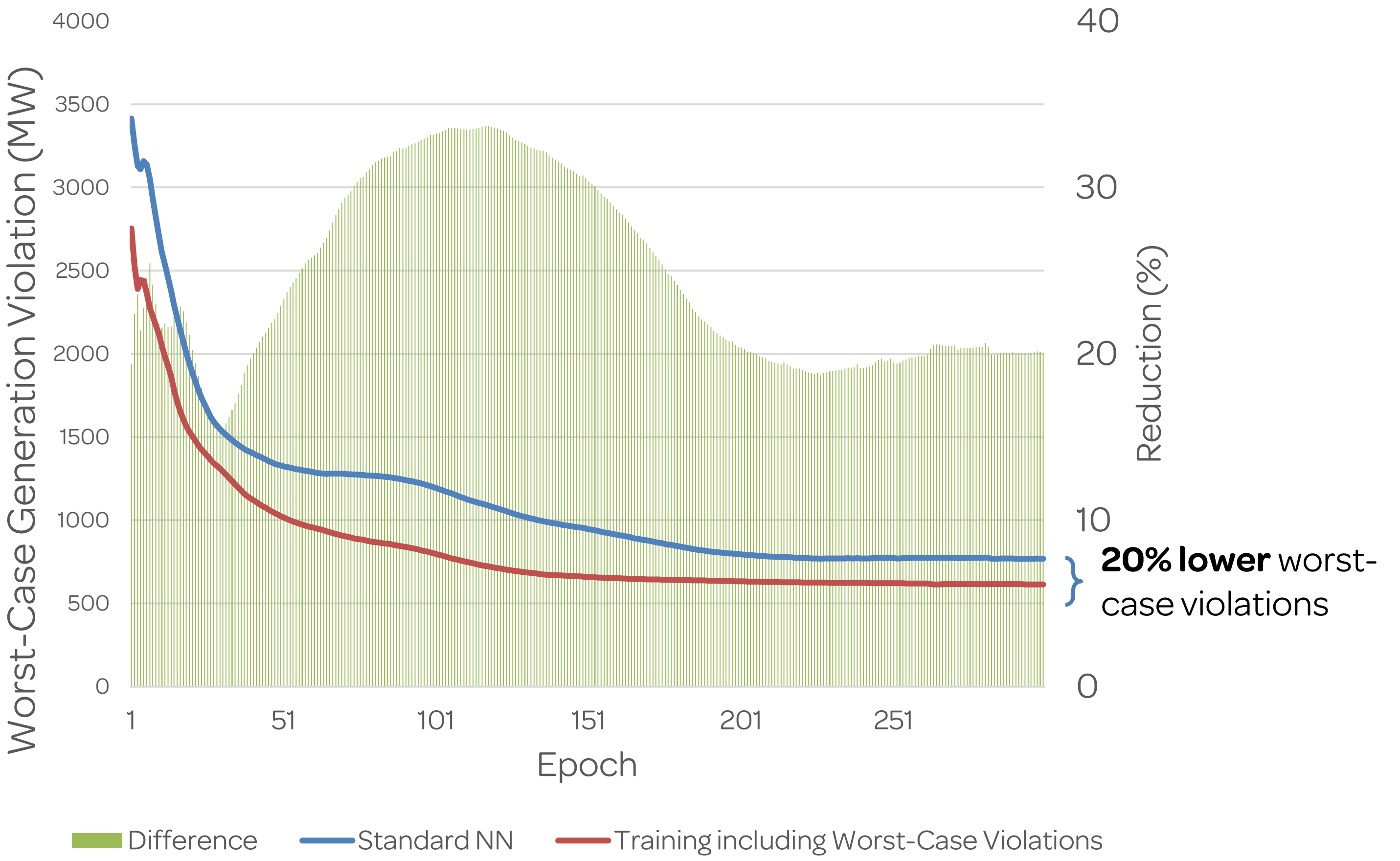
Standard NNs vs Training with Worst-Case Violations



Preliminary Results

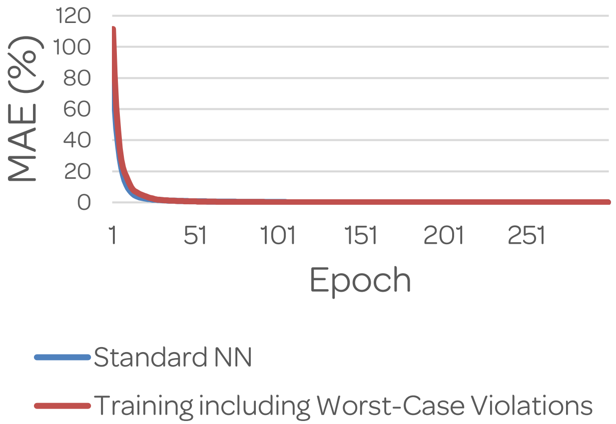
- NN training including worst-case violations consistently results to **~20% lower worst-case violations**
- Very low impact on the average performance: MAE is 0.04% higher

Standard NNs vs Training with Worst-Case Violations



Preliminary Results

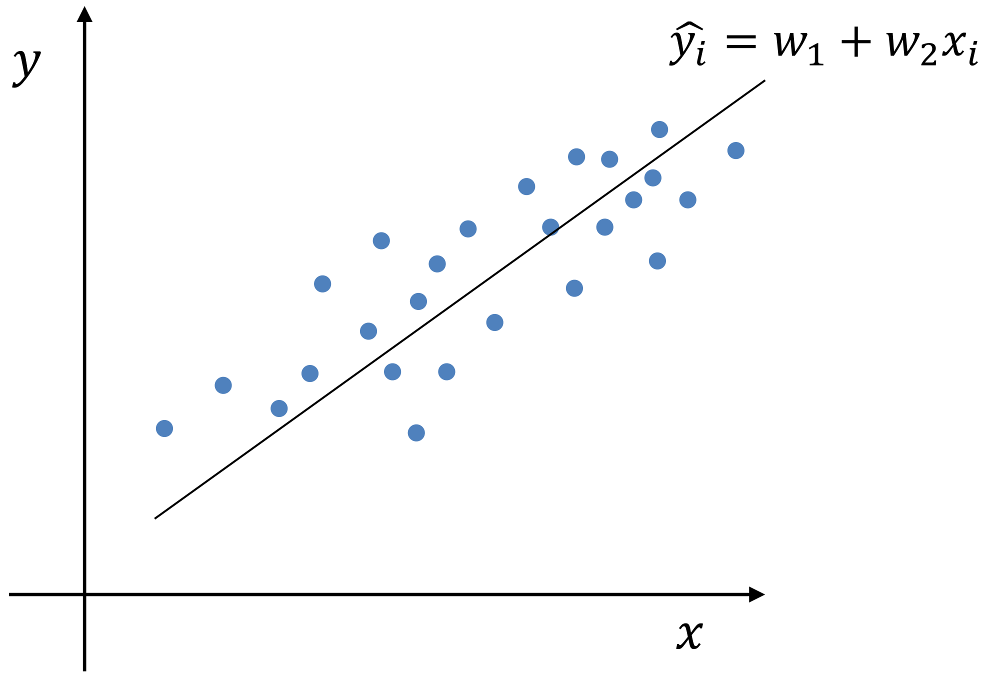
- NN training including worst-case violations consistently results to **~20% lower worst-case violations**
- Very low impact on the average performance: MAE is 0.04% higher



Physics-Informed Neural Networks for Power Systems

y_i: actual/correct value
$\hat{y}_i$: estimated value

Neural Networks: An advanced form of non-linear regression



Loss function: Estimate best w_1, w_2 to fit the training data

$$\min_{w_1, w_2} \quad \|y_i - \hat{y}_i\|$$

s.t.

$$\hat{y}_i = w_1 + w_2 x_i \qquad \forall i$$

Traditional training of neural networks required no information about the underlying physical model. Just data!

Physics Informed Neural Networks

- Automatic differentiation: derivatives of the neural network output with respect to the input can be computed during the training procedure

- A differential-algebraic model of a physical system can be included in the neural network training*

- Neural networks can now exploit knowledge of the actual physical system

- Machine learning platforms (e.g. Tensorflow) enable these capabilities

*M. Raissi, P. Perdikaris, and G. Karniadakis, Physics-Informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", Journal of Computational Physics, vol.378, pp. 686-707, 2019

Physics-Informed Neural Networks for Power Systems

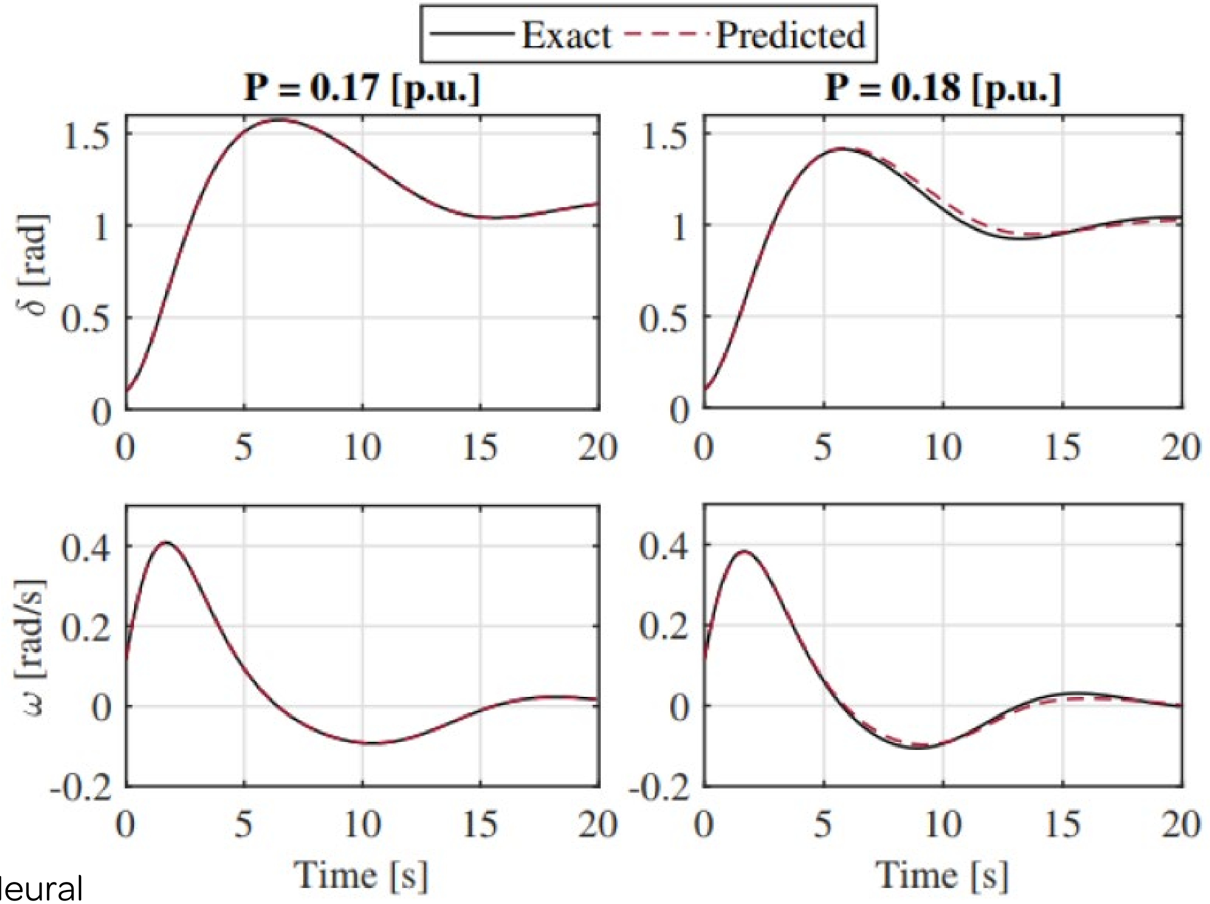
"Original"
Loss function

$$\boxed{\min_{\mathbf{W},\mathbf{b}} \quad \frac{1}{|N_\delta|}\sum_{i\in N_\delta}|\hat{\delta}-\delta^i|^2} + \frac{1}{|N_f|}\sum_{i\in N_f}|f(\hat{\delta})|^2 \tag{6a}$$

$$s.t. \qquad \hat{\delta} = NN(t, P_m, \mathbf{W}, \mathbf{b}) \tag{6b}$$

$$\dot{\hat{\delta}} = \frac{\partial\hat{\delta}}{\partial t}, \qquad \ddot{\hat{\delta}} = \frac{\partial\dot{\hat{\delta}}}{\partial t} \tag{6c}$$

$$f(\hat{\delta}) = M\ddot{\hat{\delta}} + D\dot{\hat{\delta}} + A\sin\hat{\delta} - P_m \tag{6d}$$



G. S. Misyris, A. Venzke, S. Chatzivasileiadis, Physics-Informed Neural Networks for Power Systems. Presented at the Best Paper Session of IEEE PES GM 2020. https://arxiv.org/pdf/1911.03737.pdf

Physics-Informed Neural Networks for Power Systems

"Original"
Loss function

"Physics-Informed"
term

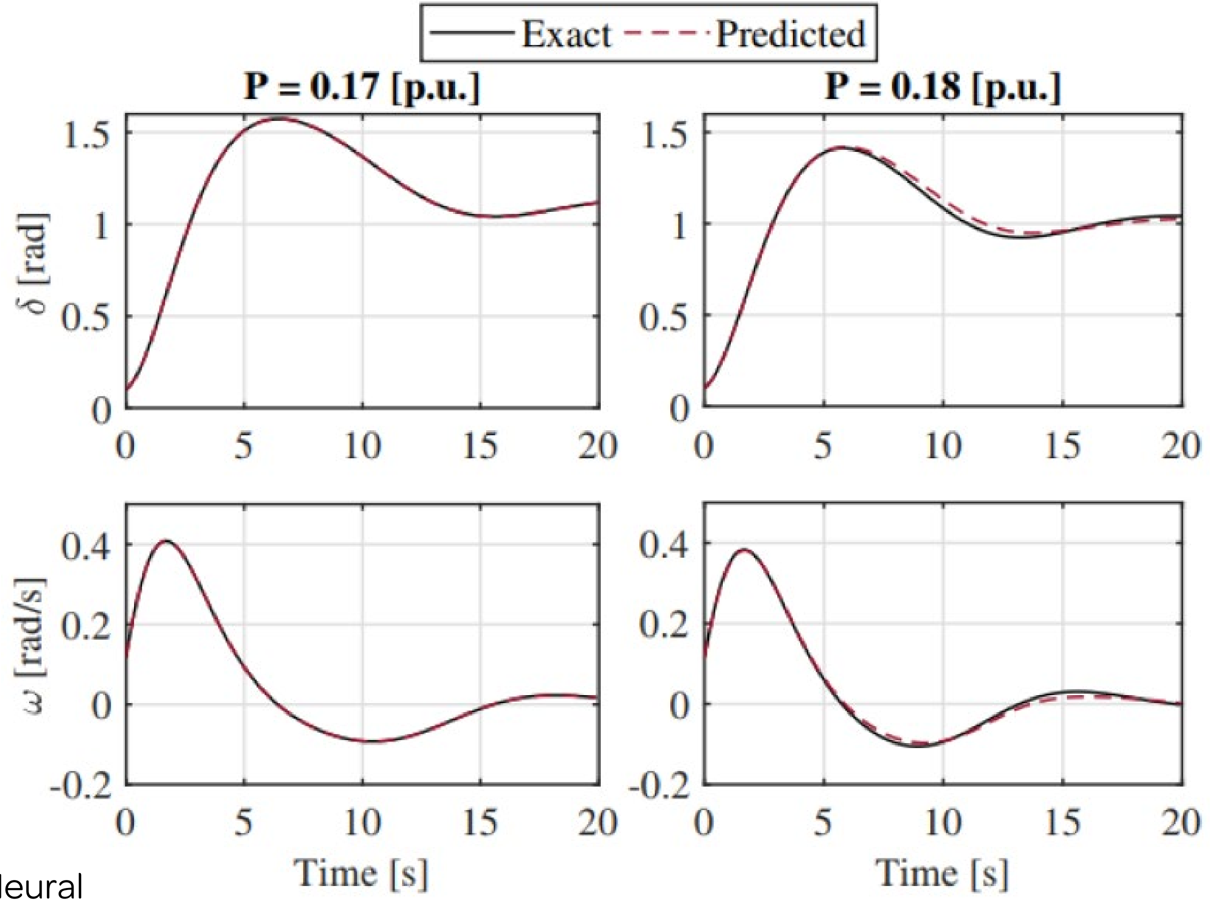
$$\min_{\mathbf{W},\mathbf{b}} \quad \frac{1}{|N_\delta|}\sum_{i\in N_\delta}|\hat{\delta}-\delta^i|^2 + \boxed{\frac{1}{|N_f|}\sum_{i\in N_f}|f(\hat{\delta})|^2} \tag{6a}$$

$$s.t. \qquad \hat{\delta} = NN(t, P_m, \mathbf{W}, \mathbf{b}) \tag{6b}$$

$$\dot{\hat{\delta}} = \frac{\partial\hat{\delta}}{\partial t}, \qquad \ddot{\hat{\delta}} = \frac{\partial\dot{\hat{\delta}}}{\partial t} \tag{6c}$$

$$\boxed{f(\hat{\delta}) = M\ddot{\hat{\delta}} + D\dot{\hat{\delta}} + A\sin\hat{\delta} - P_m} \tag{6d}$$

Swing equation



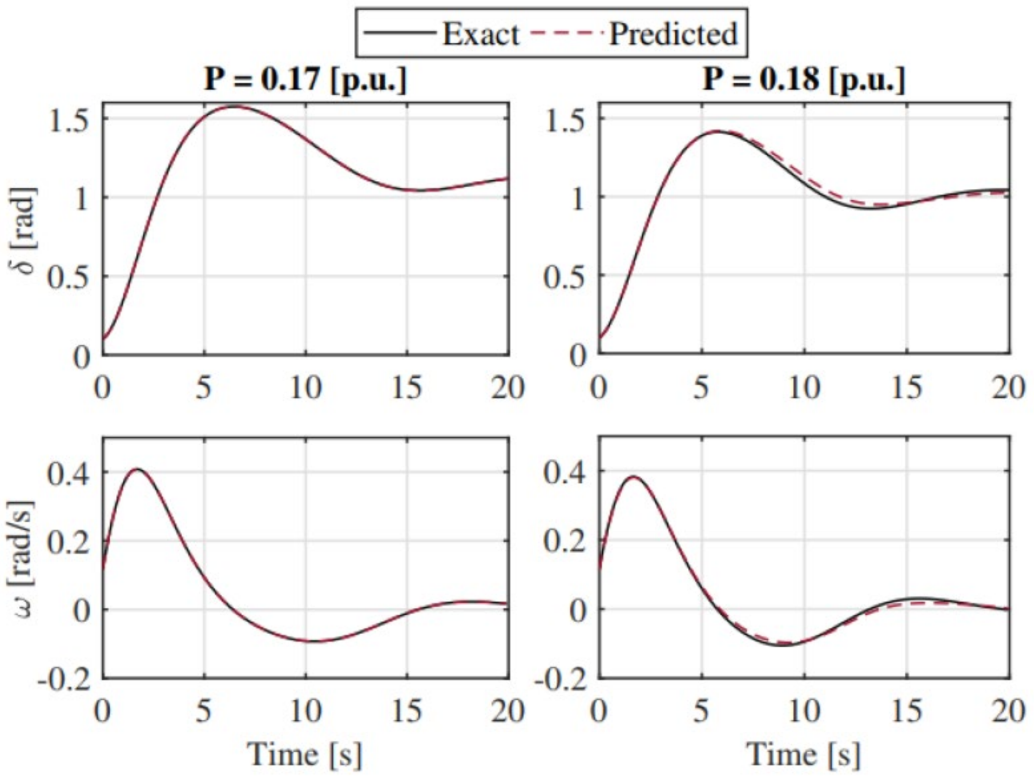
G. S. Misyris, A. Venzke, S. Chatzivasileiadis, Physics-Informed Neural Networks for Power Systems. Presented at the Best Paper Session of IEEE PES GM 2020. https://arxiv.org/pdf/1911.03737.pdf

IEEE PES Power & Energy Society®

IEEE

Physics-Informed Neural Networks for Power Systems

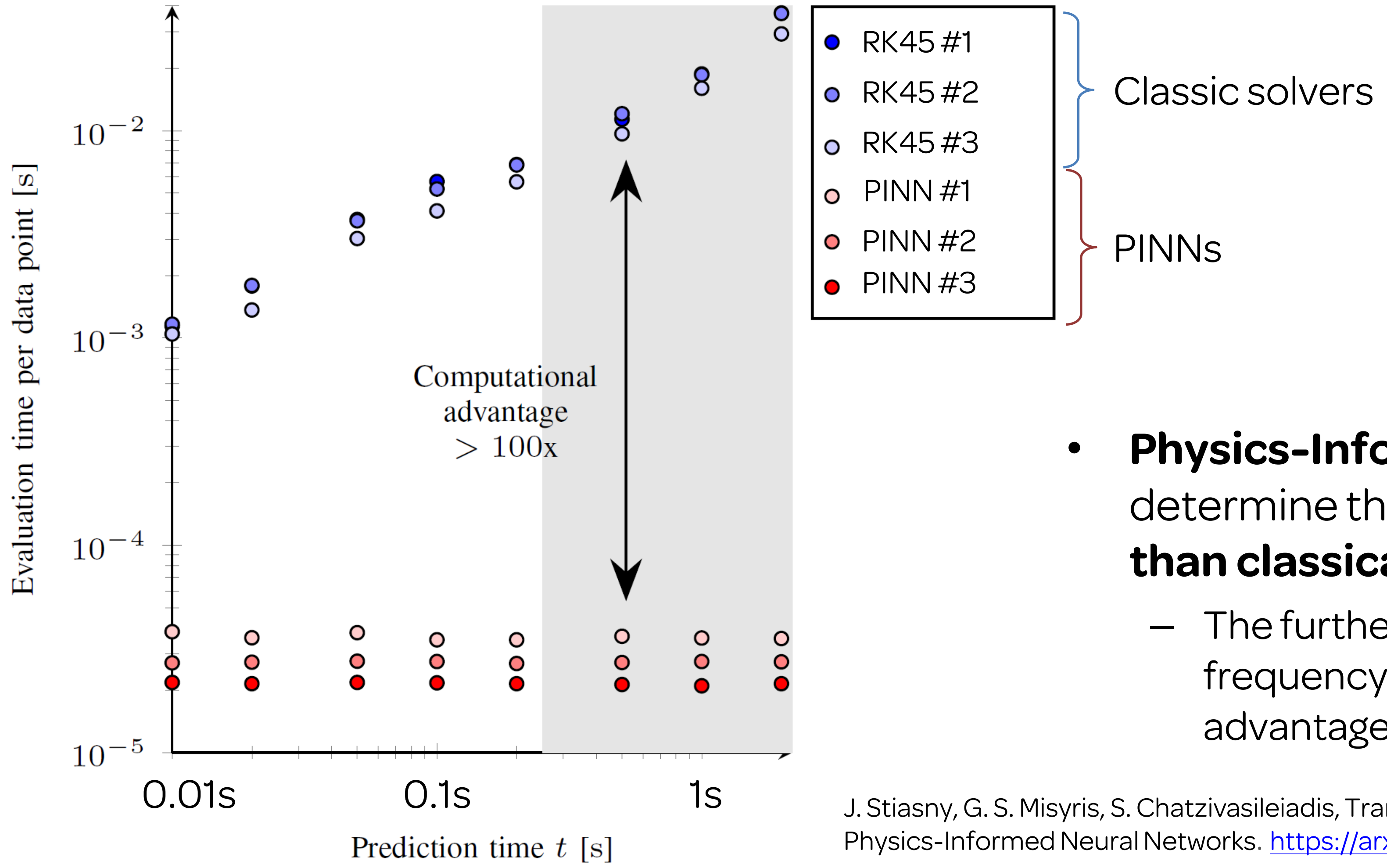
- Physics-Informed Neural Networks (PINN) could potentially replace solvers for systems of differential-algebraic equations in the long-term
  - **Probable power system application: Extremely fast screening of critical contingencies**
- In our example: PINN 87 times faster than ODE solver
- Can **directly estimate** the rotor angle at **any** time instant



Code is available on GitHub: https://github.com/jbesty

G. S. Misyris, A. Venzke, S. Chatzivasileiadis, Physics-Informed Neural Networks for Power Systems. Presented at the Best Paper Session of IEEE PES GM 2020. https://arxiv.org/pdf/1911.03737.pdf

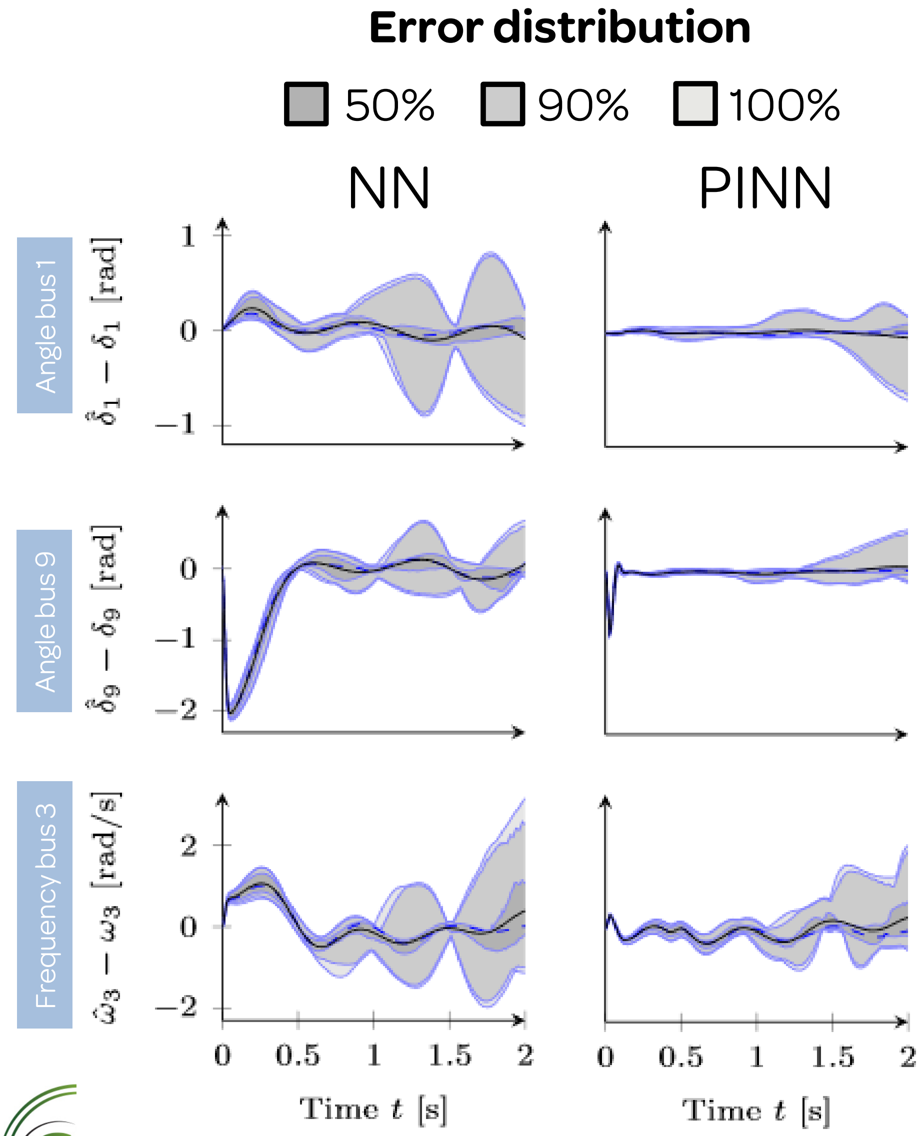
Computation time:
Classical numerical solvers vs. Physics-Informed NNs



- **Physics-Informed Neural Networks** can determine the outputs more than **100x faster than classical numerical solvers**
  - The further we look in time, e.g. what is the frequency at t=1s, the larger the computational advantage is

J. Stiasny, G. S. Misyris, S. Chatzivasileiadis, Transient Stability Analysis with Physics-Informed Neural Networks. https://arxiv.org/abs/2106.13638 [code]

Accuracy:
Standard Neural Networks (NN) vs Physics-Informed NNs (PINN)



- **PINNs result in lower errors** than standard neural networks
- The error increases as we look further into the future
- PINNs can deliver an excellent screening tool, i.e. to very quickly assess if critical scenarios are secure or not.
  - To determine the exact numerical values, classic solvers are still very valuable

J. Stiasny, G. S. Misyris, S. Chatzivasileiadis, Transient Stability Analysis with Physics-Informed Neural Networks. https://arxiv.org/abs/2106.13638 [code]

Wrap-up

- Neural network verification (and worst-case guarantees) can **build the missing trust** for neural network applications in power systems

- Physics Informed Neural Networks can take advantage of the rich information about existing power system models inside neural network training

- This talk:

1. Verification of Classification Neural Networks for Power Systems
    - Application in Security Assessment
2. Provable worst-case guarantees for Regression Neural Networks
    - Application in OPF (and probably in many other optimization programs)
3. Physics-Informed Neural Networks for Power Systems

Thank you!

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

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

A. Venzke, G. Qu, S. Low, S. Chatzivasileiadis, Learning Optimal Power Flow: Worst-case Guarantees for Neural Networks. **Best Student Paper Award** at IEEE SmartGridComm 2020.[.pdf | slides | video]

G. S. Misyris, A. Venzke, S. Chatzivasileiadis, Physics-Informed Neural Networks for Power Systems. Presented at the **Best Paper Session** of IEEE PES GM 2020. https://arxiv.org/pdf/1911.03737.pdf

R. Nellikkath, S. Chatzivasileiadis, Physics-Informed Neural Networks for AC Optimal Power Flow
https://arxiv.org/abs/2110.02672 [code]

J. Stiasny, G. S. Misyris, S. Chatzivasileiadis, Transient Stability Analysis with Physics-Informed Neural Networks. https://arxiv.org/abs/2106.13638 [code]

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