Farm-wide virtual load monitoring for offshore wind structures via Bayesian neural networks

Nandar Hlaing[1,*], Pablo G. Morato[1], Francisco de Nolasco Santos[2], Wout Weijtjens[2], Christof Devriendt[2] and Philippe Rigo[1]

[1]Naval & Offshore Engineering, ArGEnCo, University of Liege, 4000 Liege, Belgium

[2]OWI-Lab, Vrije Universiteit Brussel, 1050 Brussels, Belgium

WORKSHOP in VUB

November 16, 2022, Brussels

PhairywinD

Belgian Offshore PhD Expertise

Why Bayesian neural networks?

- When lacking known grounds, Bayesian models express higher uncertainty in the extrapolation regime.

- Probabilistic interpretation of prediction models
  - predicted value
  - its confidence

- Joint uncertainty quantification
  - aleatory (physical) uncertainty
  - epistemic (model) uncertainty

PhairywinD
Belgian Offshore PhD Expertise

ABC of uncertainty quantification

Physical uncertainty

https://www.nytimes.com/2016/08/23/science/americas-first-offshore-wind-farm-may-power-up-a-new-industry.html

Aleatory

Measurement uncertainty

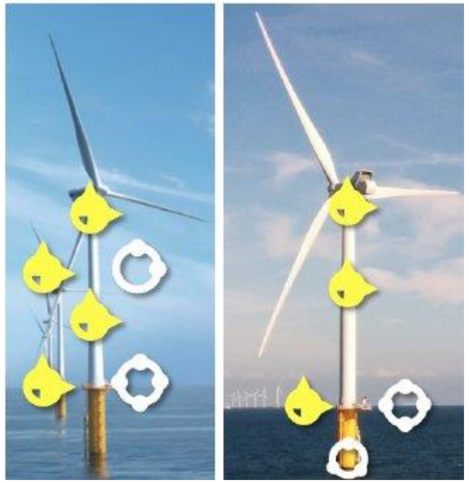
http://www.winspector.eu/news-and-events/oscillation-measurement-on-wind-turbine-blade/

Model uncertainty

https://www.smart-energy.com/renewable-energy/nrel-iea-release-open-access-plans-for-iea-15-mw-wind-turbine/

Epistemic

Statistical uncertainty



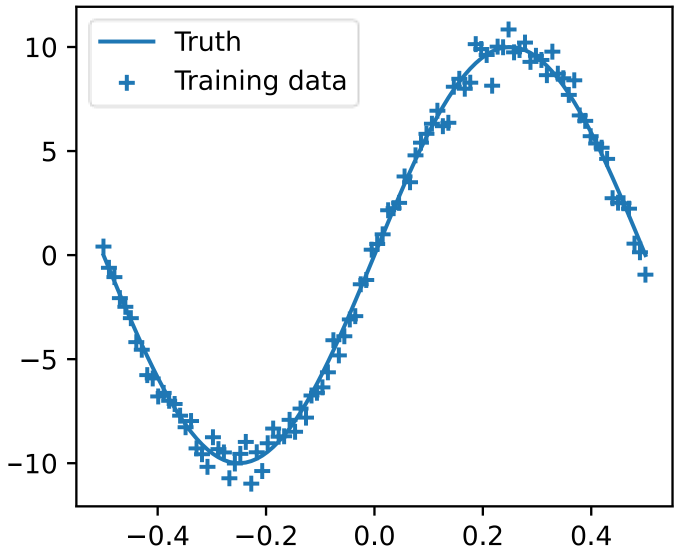
Source:https://www.researchgate.net/publication/275154568_MONITORING_THE_CONSUMED_FATIGUE_LIFE_OF_WIND_TURBINES_ON_MONOPILE_FOUNDATIONS

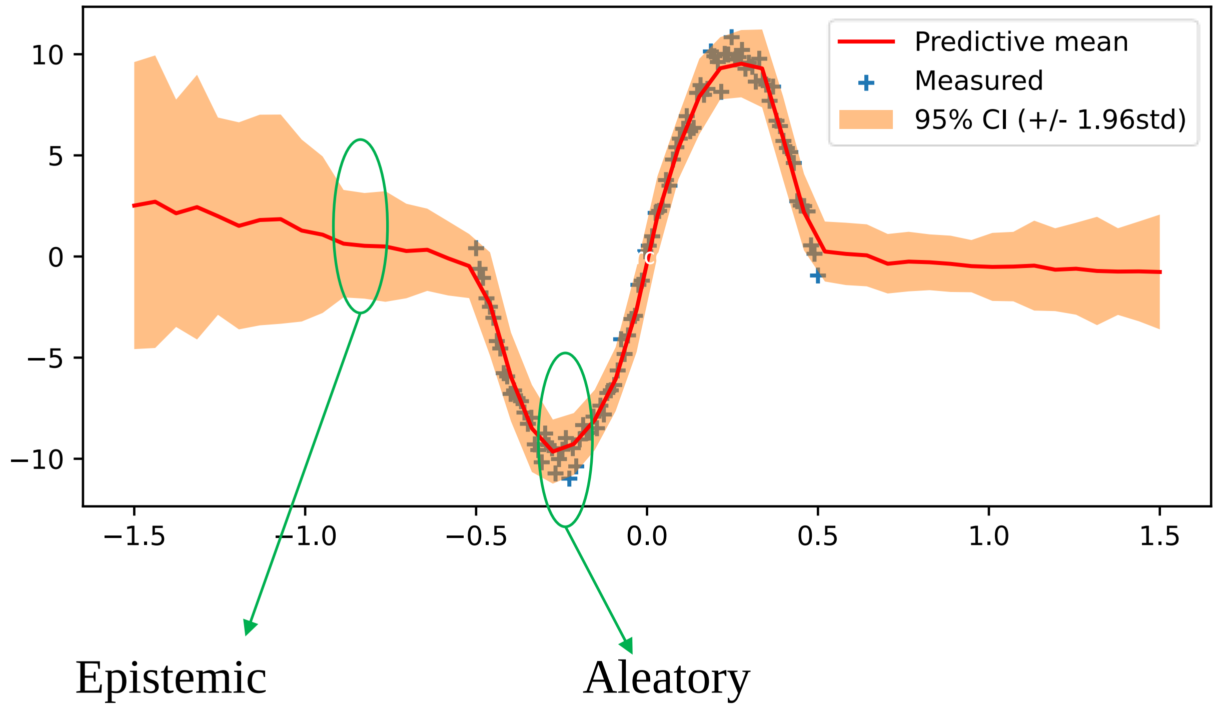
PhairywinD

Belgian Offshore PhD Expertise

Example BNN with a simple dataset

- A BNN modelling a sine function





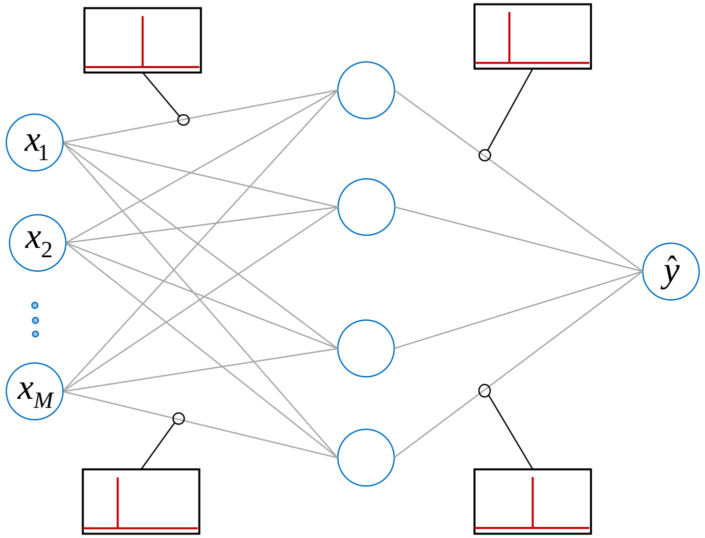
REF: https://matthewmcateer.me/blog/a-quick-intro-to-bayesian-neural-networks/

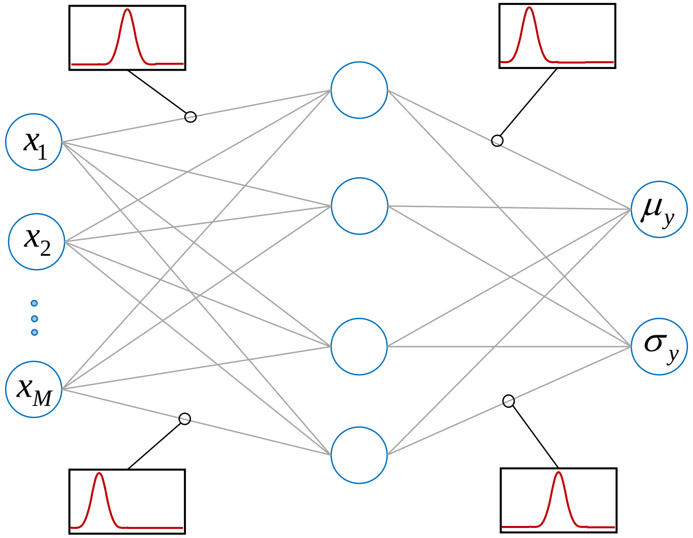
PhairywinD
Belgian Offshore PhD Expertise

Essentials of Bayesian neural networks

- Stochastic components
  - Stochastic weights and biases, and/or
  - Stochastic activations

- Stochastic output, i.e., a probability distribution
  - One parameter can be known.

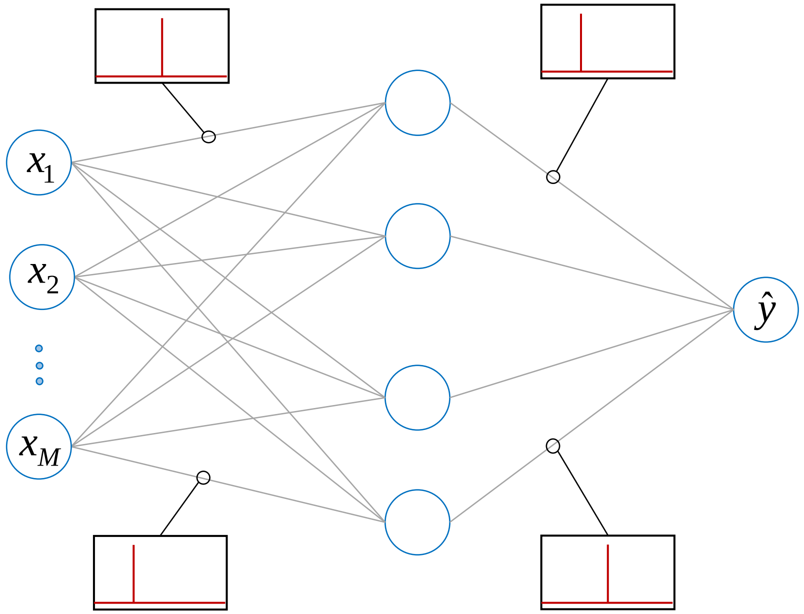


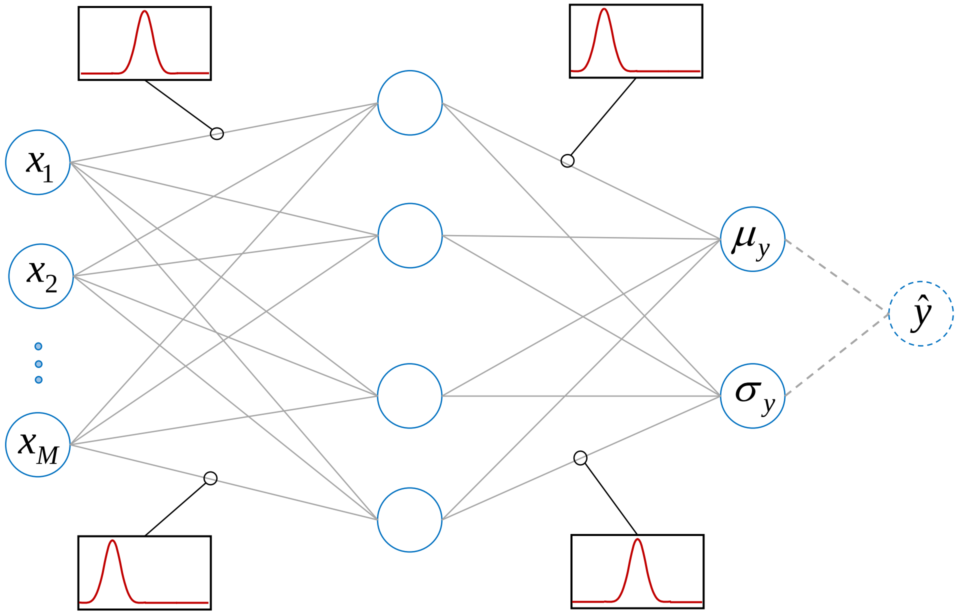


PhairywinD
Belgian Offshore PhD Expertise

Essentials of Bayesian neural networks

- Distribution layer





Essentials of Bayesian neural networks

- Bayesian inference

$$P(\theta|\, D) = \frac{P(D|\, \theta)P(\theta)}{\int P(D|\, \theta)P(\theta)d\theta}$$

$P(\theta)$ = prior distribution of model parameters (user-specified)

$P(D|\, \theta)$ = likelihood (D is the observed 'training' data)

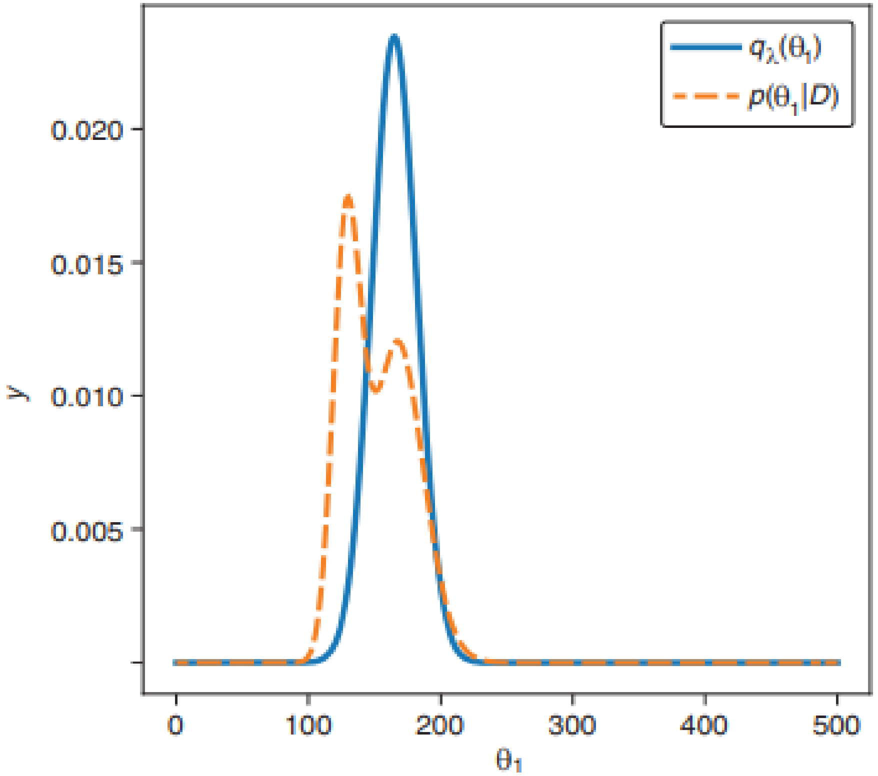
$P(\theta|\, D)$ = posterior distribution of model parameters

** High dimensional integral over the weights >> intractable posterior

Essentials of Bayesian neural networks

- Variational inference

- Objective of VI – to estimate unknown posterior distribution by a variational distribution, usually Gaussian $\mathcal{N}(\mu,\sigma)$.



How to measure similarity or divergence?

PhairywinD
Belgian Offshore PhD Expertise

Variational inference (in just a little bit of Maths :D)

$$\mathbb{KL}\left(q_\lambda(\theta) || P(\theta | D)\right) = \int q_\lambda(\theta) \log \frac{q_\lambda(\theta)}{P(\theta | D)} d\theta$$

$$\mathbb{KL}\left(q_\lambda(\theta) || P(\theta | D)\right) = \int q_\lambda(\theta) \log \frac{q_\lambda(\theta)}{\frac{P(\theta, D)}{P(D)}} d\theta$$

$$\mathbb{KL}\left(q_\lambda(\theta) || P(\theta | D)\right) = \int q_\lambda(\theta) \log P(D) d\theta - \int q_\lambda(\theta) \log \frac{P(\theta, D)}{q_\lambda(\theta)} d\theta$$

$$\mathbb{KL}\left(q_\lambda(\theta) || P(\theta | D)\right) = \log P(D) - \int q_\lambda(\theta) \log \frac{P(D | \theta) P(\theta)}{q_\lambda(\theta)} d\theta$$

$$\mathbb{KL}\left(q_\lambda(\theta) || P(\theta | D)\right) = -\int q_\lambda(\theta) \log \frac{P(\theta)}{q_\lambda(\theta)} d\theta - \int q_\lambda(\theta) \log P(D | \theta) d\theta$$

$$\mathbb{KL}\left(q_\lambda(\theta) || P(\theta | D)\right) = \int q_\lambda(\theta) \log \frac{q_\lambda(\theta)}{P(\theta)} d\theta - \int q_\lambda(\theta) \log P(D | \theta) d\theta$$

$$\mathbb{KL}\left(q_\lambda(\theta) || P(\theta | D)\right) = \mathbb{KL}\left(q_\lambda(\theta) || P(\theta)\right) - \mathbb{E}_{\theta \sim q_\lambda}\left[\log(P(D | \theta))\right]$$

PhairywinD
Belgian Offshore PhD Expertise

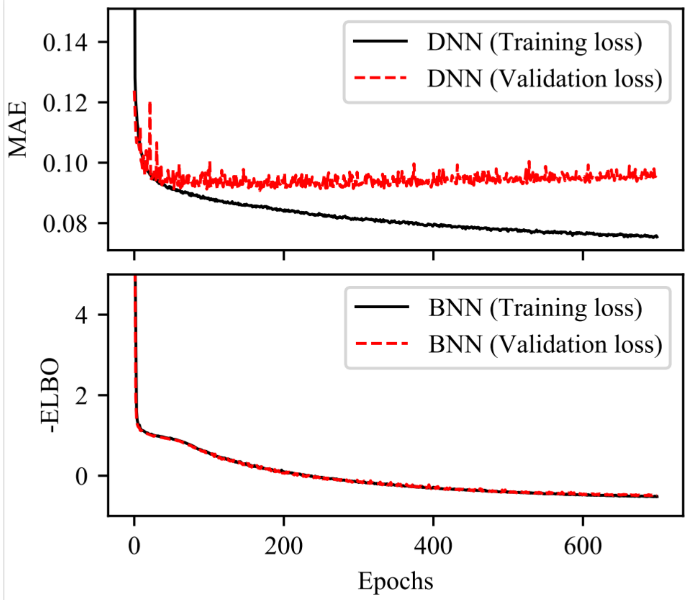
Variational inference

$$\lambda^* = \mathrm{argmin}\boxed{\left\{\mathbb{KL}\left(q_\lambda(\theta) \| P(\theta)\right) - \mathbb{E}_{\theta \sim q_\lambda}\left[\log(P(D|\theta))\right]\right\}}$$

Regularizer

Expected negative log-likelihood

- ✓ KL between variational and prior distributions.
- ✓ Prevents overfitting



Variational inference

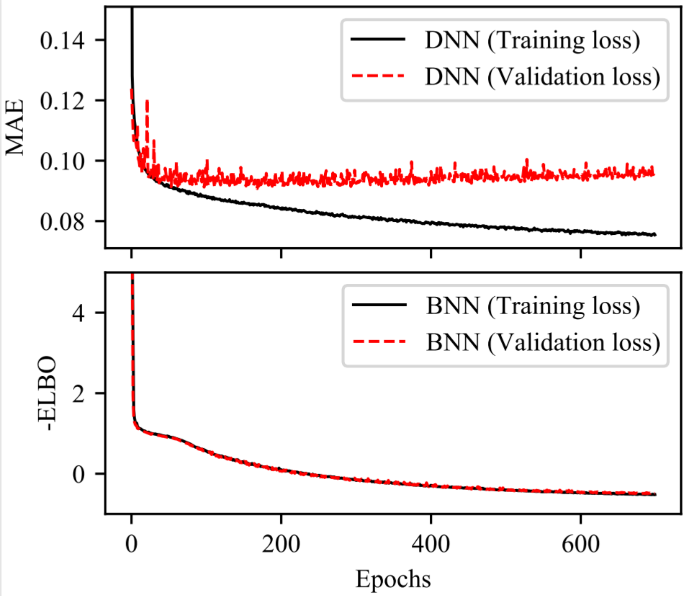
$$\lambda^* = \mathrm{argmin} \boxed{\left\{ \mathbb{KL}\left(q_\lambda(\theta) \| P(\theta)\right) - \mathbb{E}_{\theta \sim q_\lambda}\left[\log(P(D|\theta))\right] \right\}}$$

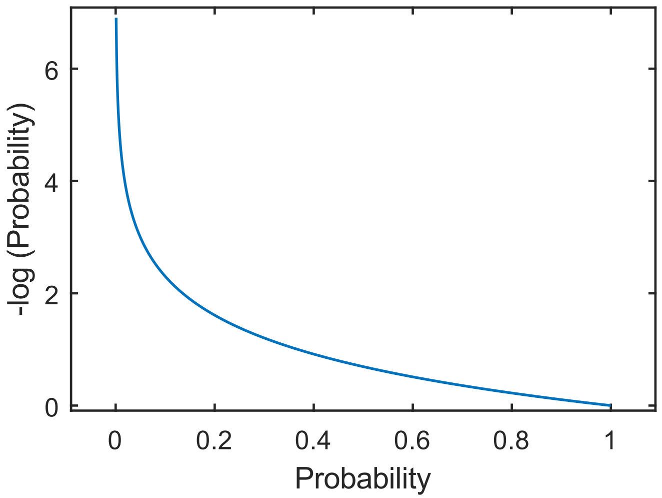
Regularizer

Expected negative log-likelihood

- ✓ KL between variational and prior distributions.
- ✓ Prevents overfitting.

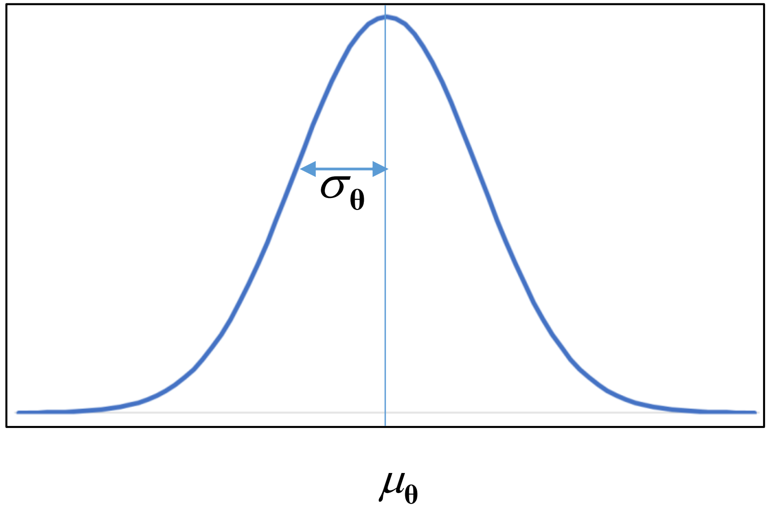
- ✓ Maximize the likelihood of the ‘labels’.
- ✓ One realization instead of expectation*.



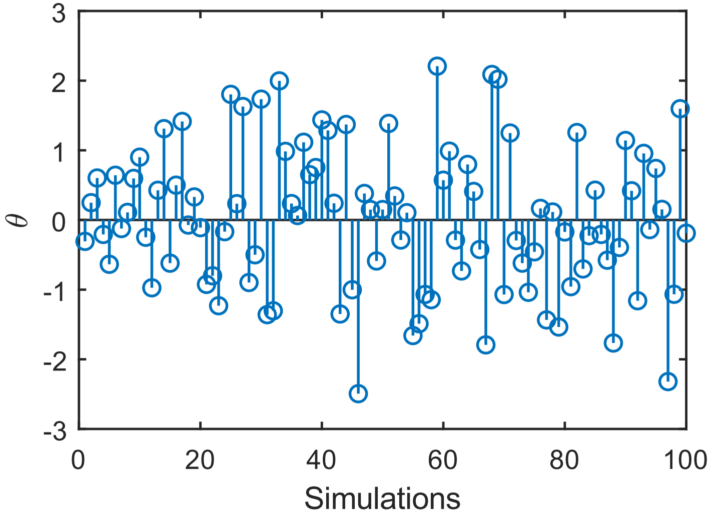


Training Bayesian neural networks

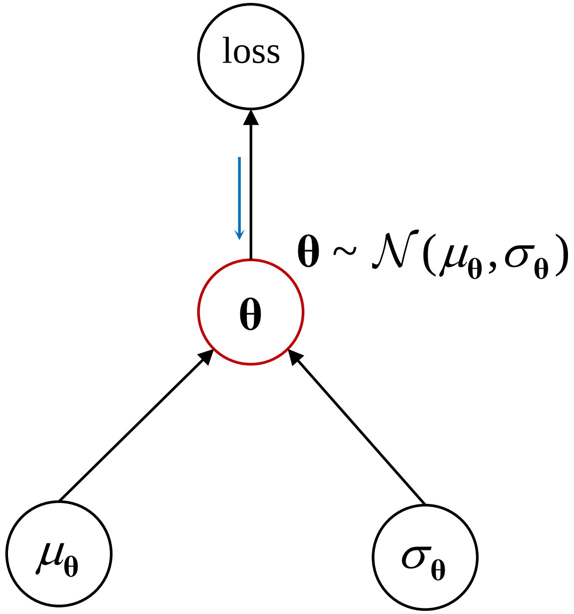
- Pick a prior distribution for the weights and/or biases.
- Draw a random sample from the distribution.
- Perform a forward pass.
- Compute the loss function (KL+NLL) given the random sample.
- Compute the derivative of the loss w.r.t the distribution parameters.
- Update the (posterior) distribution parameters.

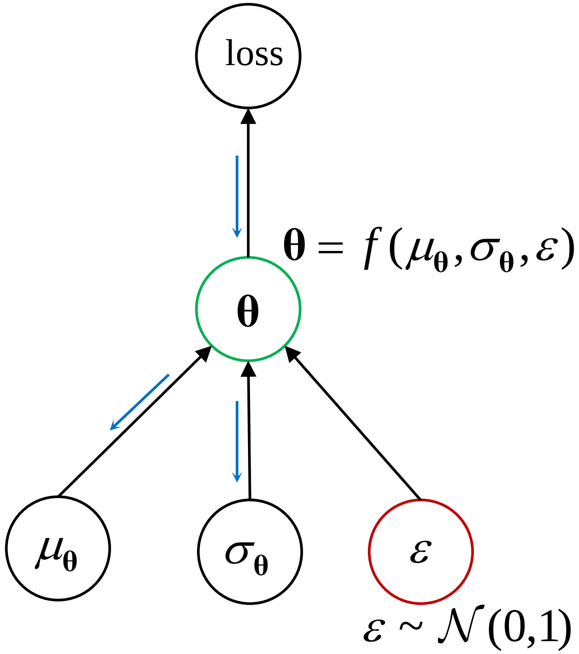


PDF: $p(\theta) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2}$



Training Bayesian neural networks





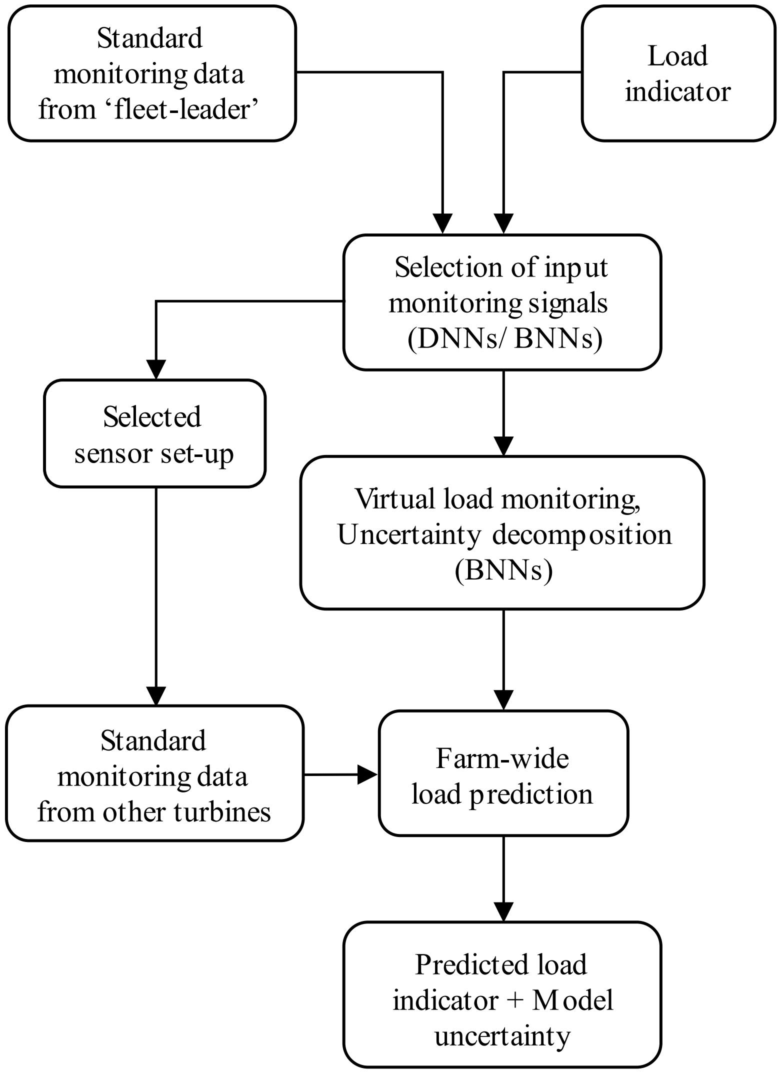
Reparametrization: $\boldsymbol{\theta} = \mu_{\boldsymbol{\theta}} + \sigma_{\boldsymbol{\theta}} \cdot \varepsilon$

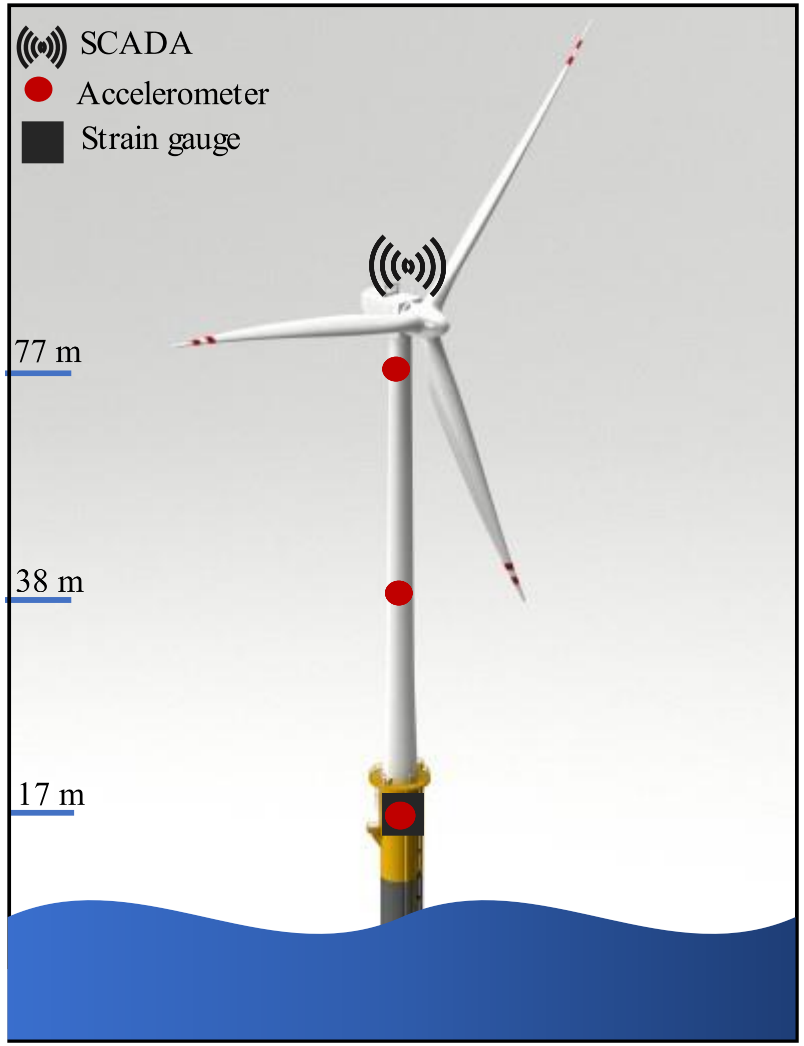
Some more practical things

- Implementation of BNNs in TensorFlow Probability
- Dense variational layers
  - DenseReparameterization
  - DenseFlipout
  - ..
- Convolutional variational layers
  - DenseReparameterization
  - DenseFlipout
  - ..
- Distribution layers
  - Sevaral ready-to-use distributions
  - [DistributionLambda]() for other distributions

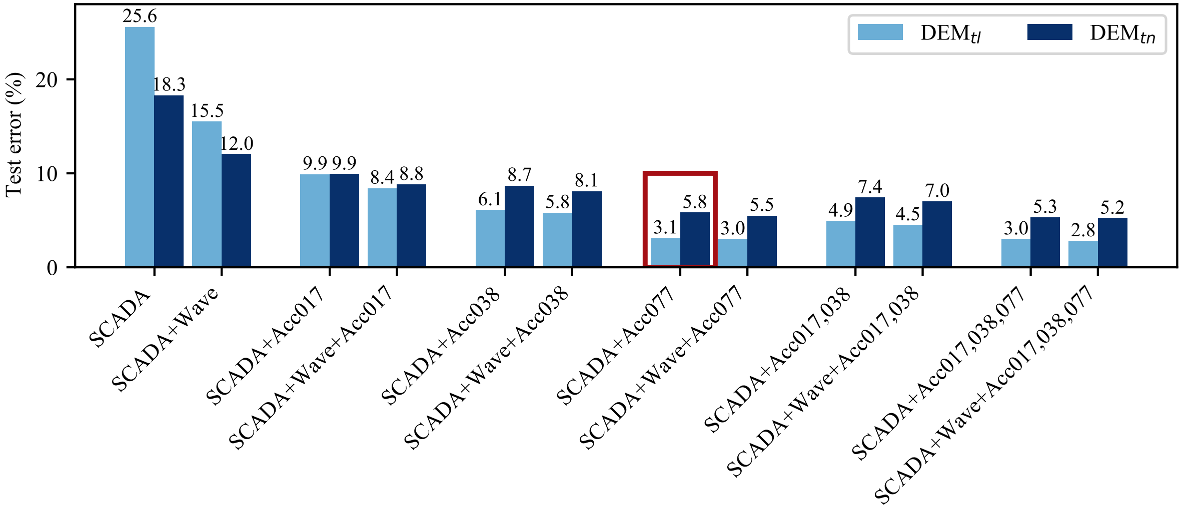
PhairywinD
Belgian Offshore PhD Expertise

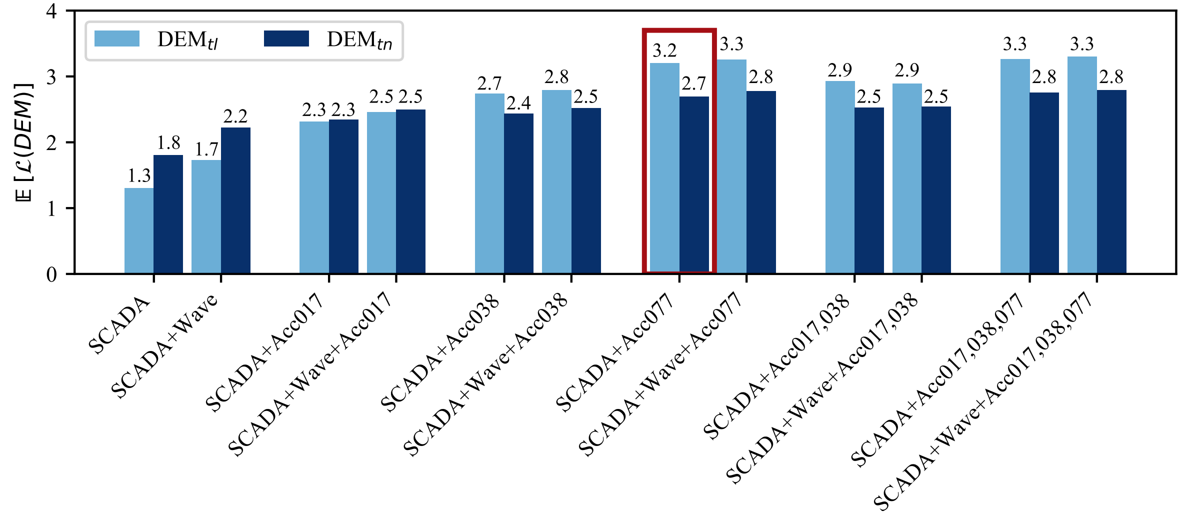
Farm-wide virtual load monitoring for OWTs via BNNs





Farm-wide virtual load monitoring for OWTs via BNNs





Farm-wide application

- Multiple forward runs in BNNs
- Expectation of the predicted load
- Compute the overall and model uncertainty
- Compare against measured labels**

Farm-wide virtual load monitoring for OWTs via BNNs

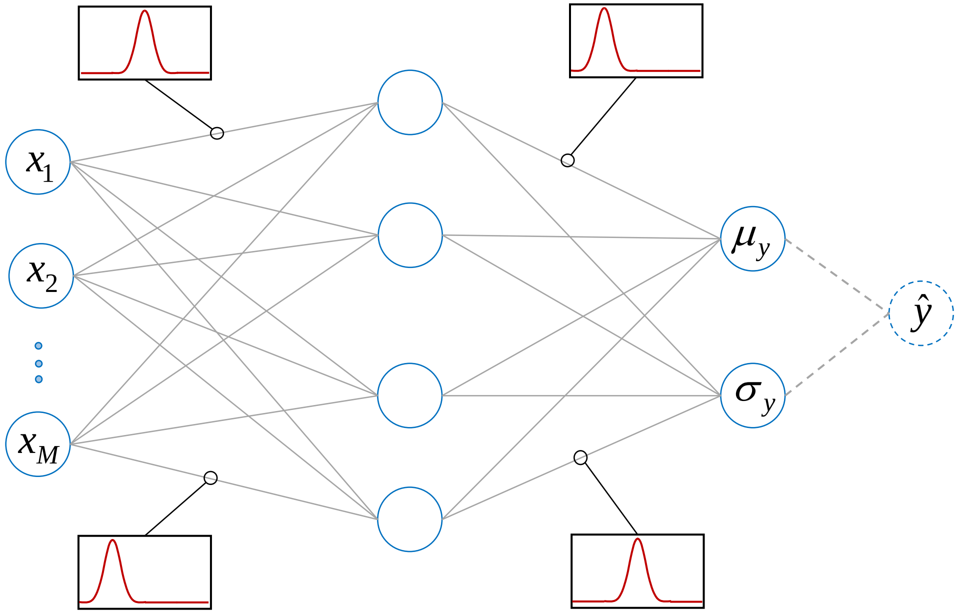
- Total variance theorem:

$$\mathbb{V}(\hat{y}|\,\mathbf{x}) = \underbrace{\mathbb{E}[\sigma_y^2|\,\mathbf{x}]}_{\text{Aleatory}} + \underbrace{\mathbb{V}(\mu_y|\,\mathbf{x})}_{\text{Epistemic}}$$

- μ_y and σ_y are also stochastic.

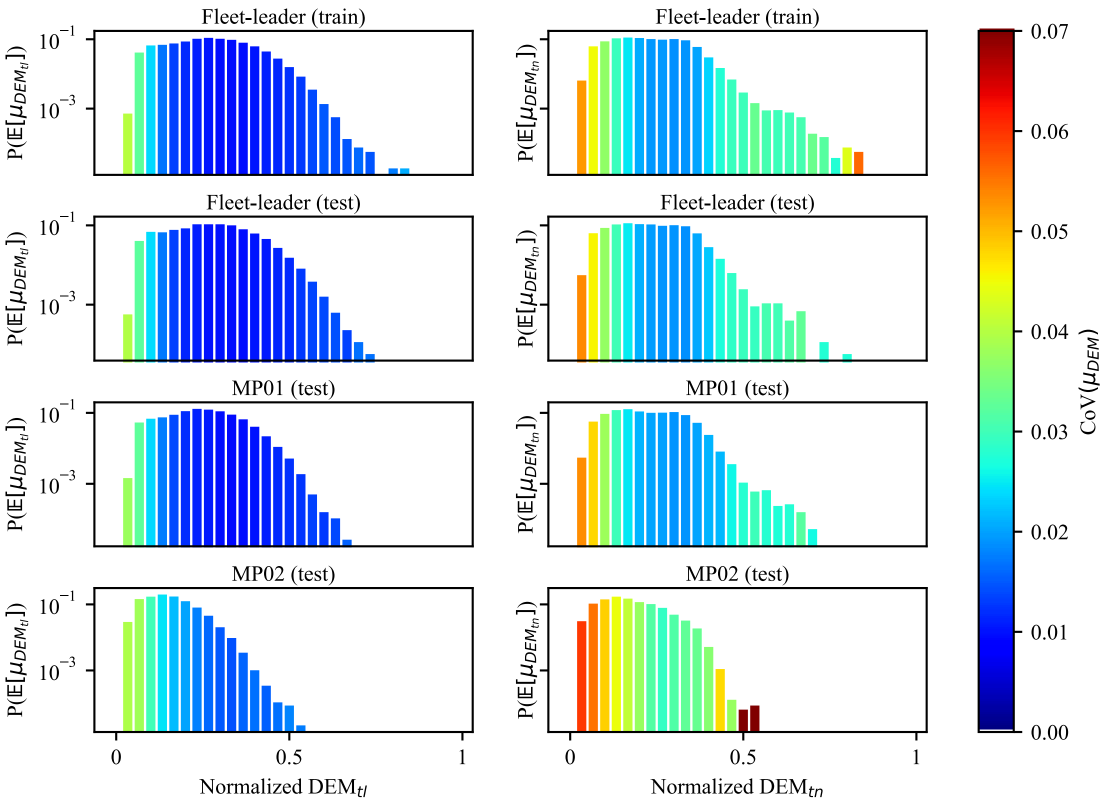
- Perfect model >> σ_y is the aleatory uncertainty.

- Model uncertainty from distribution of weights.

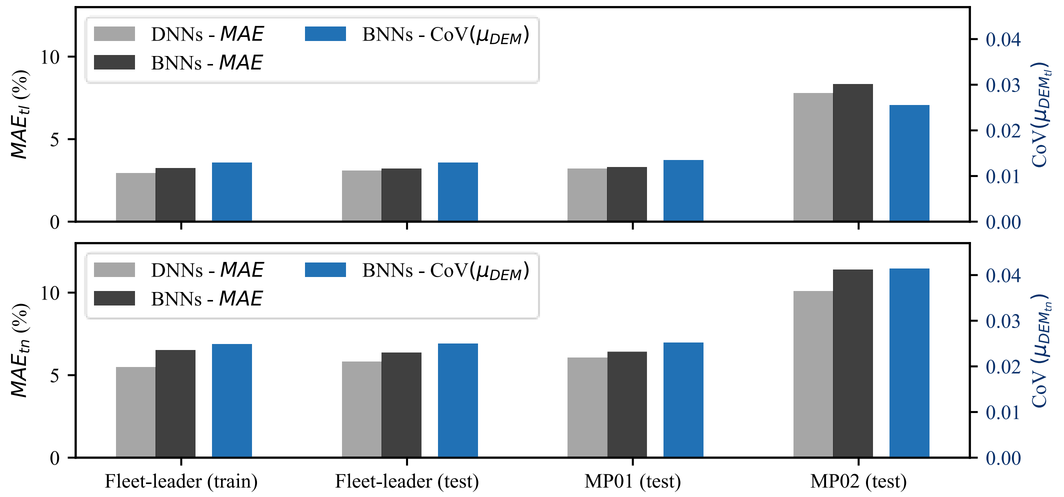


Farm-wide virtual load monitoring for OWTs via BNNs

Model uncertainty (BNN’s output only, no labels)

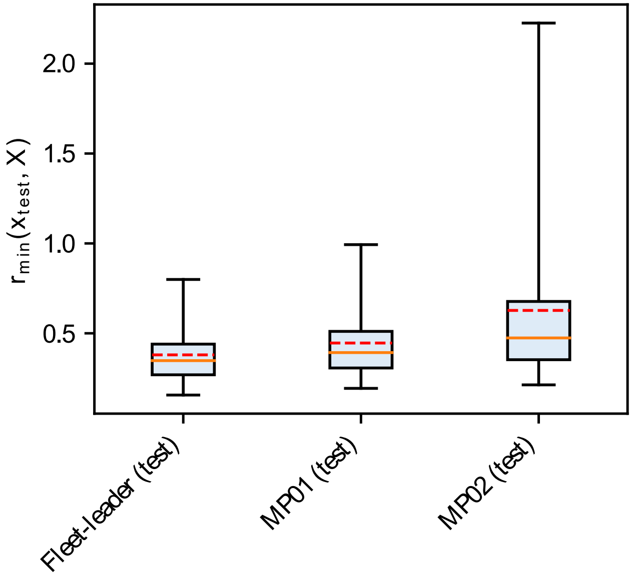


Comparison against labels (DNNs Vs BNNs)

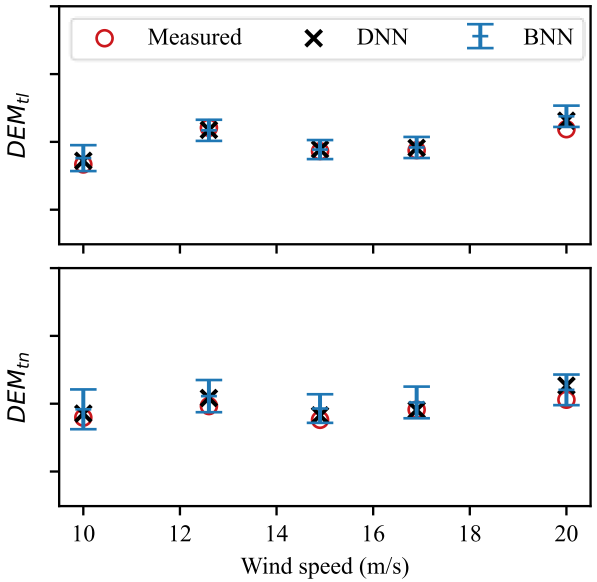


Farm-wide virtual load monitoring for OWTs via BNNs

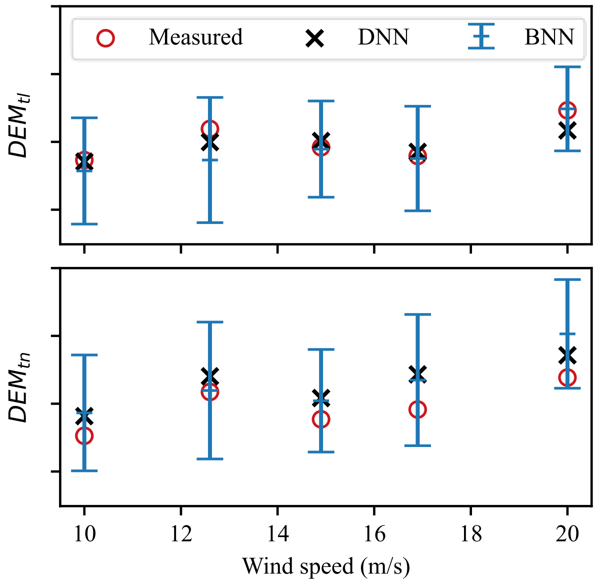
Mininum Eucleadian distances to the nearest training point



In-training point

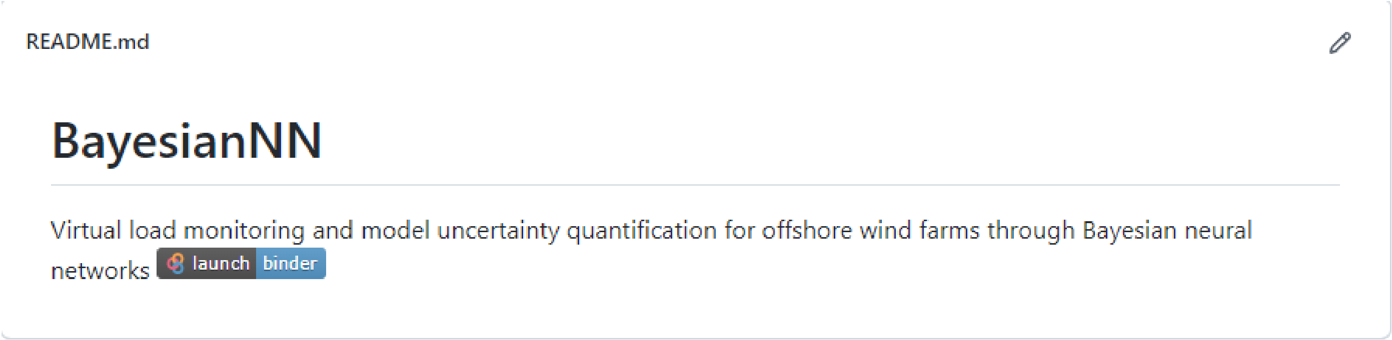


Out-of-training point



Hands-on experience

- Go to Github.com/Nandarhline.
- The name of the repository is ‘BayesianNN’.
- Launch binder in README.md

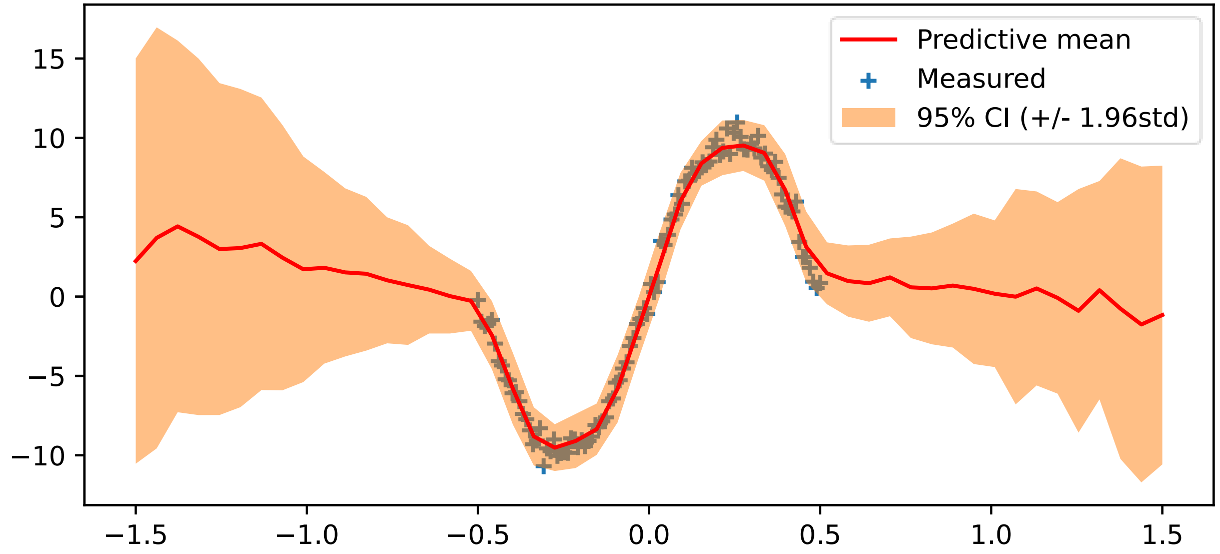


- Case(1): Synthetic data with constant noise
- Case(2): Adding training data from different regions
- Case(3): Synthetic data with varied noise

PhairywinD
Belgian Offshore PhD Expertise

Hands-on experience

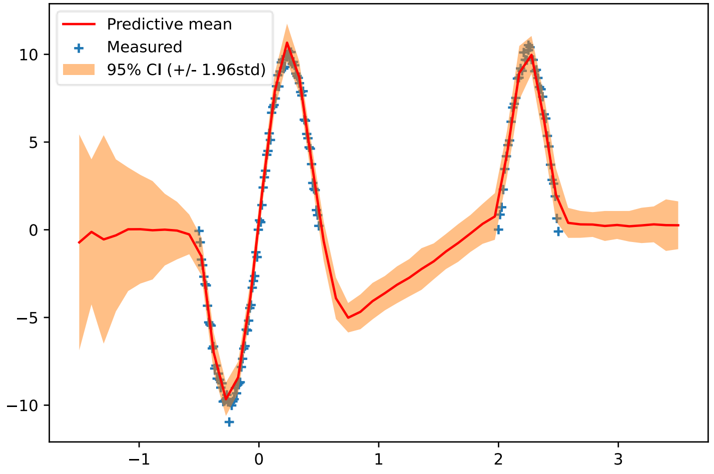
- Case(1): Synthetic data with constant noise



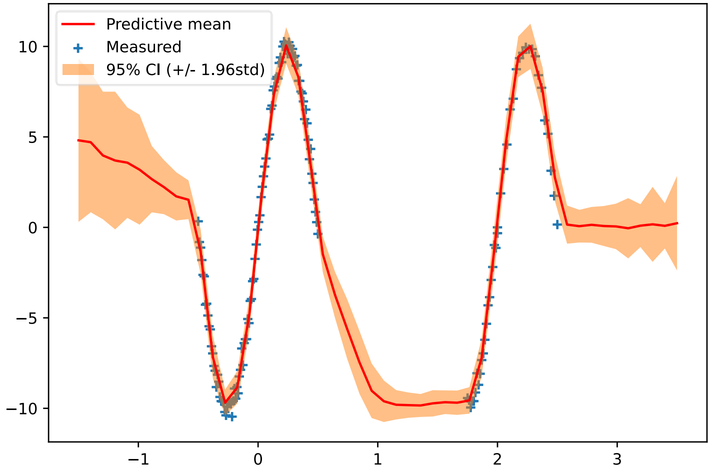
PhairywinD
Belgian Offshore PhD Expertise

Hands-on experience

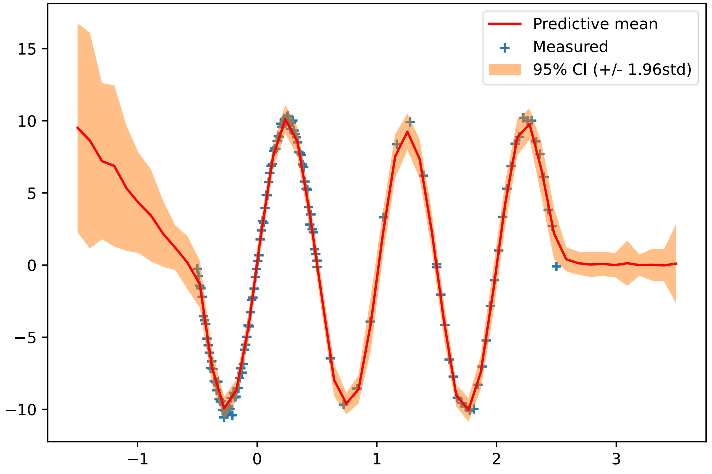
- Case(2): Adding training data from different regions



20 points between 2 and 2.25
20 points between 2.25 and 2.5



20 points between 1.75 and 2
20 points between 2 and 2.5

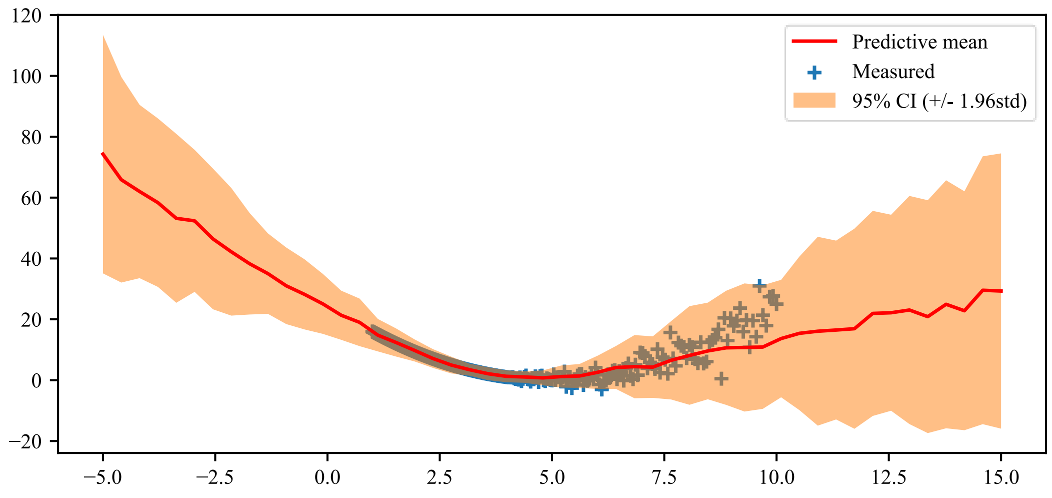


10 points between 0.5 and 1.5
30 points between 1.5 and 2.5

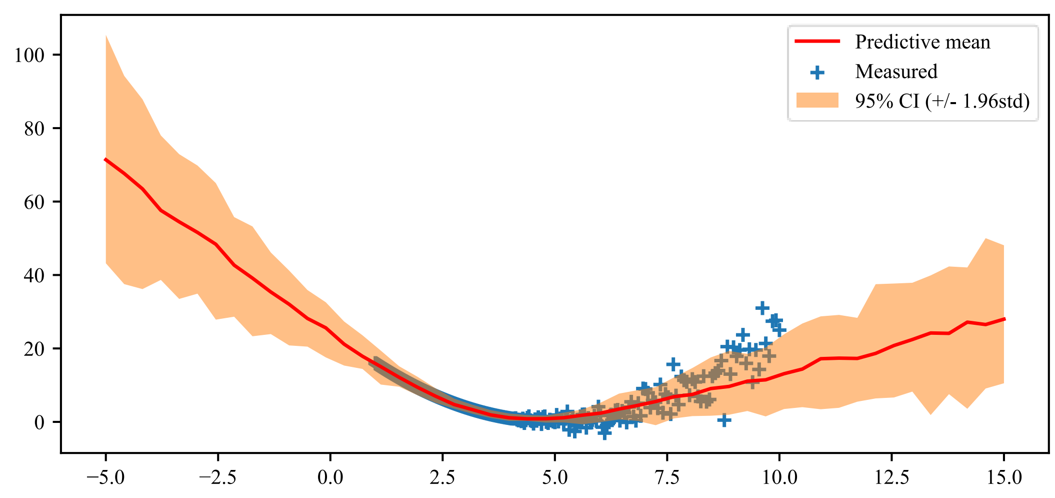
PhairywinD
Belgian Offshore PhD Expertise

Hands-on experience

- Case(3): Synthetic data with varied noise



Overall predictive uncertainty



Model uncertainty

PhairywinD
Belgian Offshore PhD Expertise

Future work

- To identify new optimal training points.
- To compare with kernel-based methods, e.g., Gaussian processes.
- To implement in decision-making for life-cycle management.

PhairywinD
Belgian Offshore PhD Expertise

Partners

With the support of Energy Transition Fund