Uncertainty quantification in Artificial Neural Network: overview and application in nuclear industries

Data Science and Applied Mathematics Competency Centre

L. Lefebvre, M. Segond and G. Mestrot

{laurent.lefebvre, mathieu.segond, gaspard.mestrot}@framatome.com

Confidentiality

X This document contains Framatome's know-how

EXPORT CONTROL

AL = 0E001	ECCN = N

Goods labeled with "AL not equal to N" are subject to European or German export authorization when being exported within or out of the EU.
Goods labeled with "ECCN not equal to N or EAR99" are subject to U.S. reexport authorization. Even without a label, or with label: "AL:N" or "ECCN:N" or "ECCN:EAR99," authorization may be required due to the final whereabouts and purpose for which the goods are to be used.

FRAMATOME'S INFORMATION PROTECTION RULES

X C1 : This document and any and all information contained therein and/or disclosed in discussions supported by this document are **restricted**.

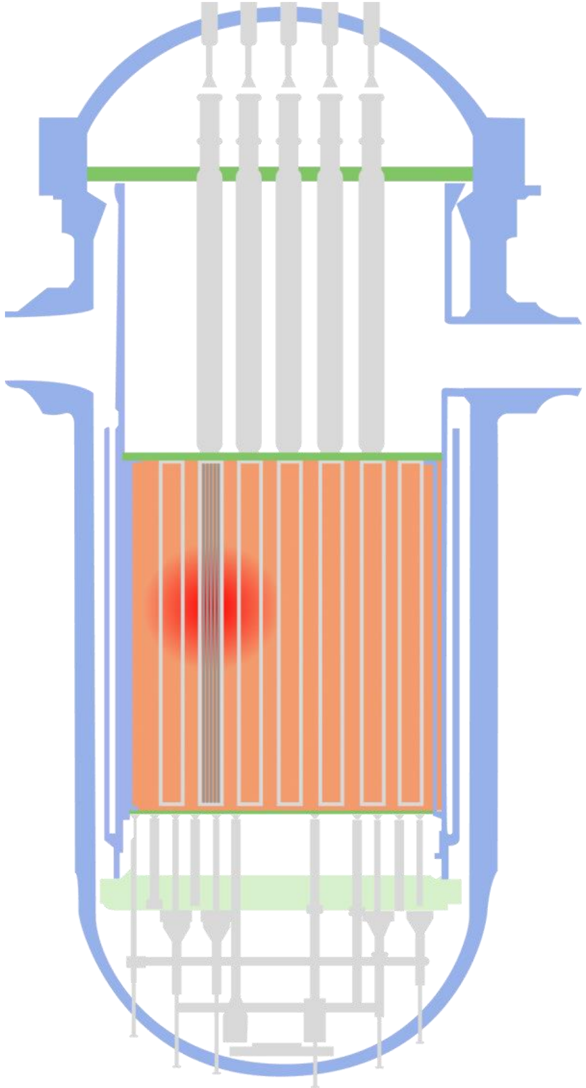
C2 : This document and any and all information contained therein and/or disclosed in discussions supported by this document are sensitive and **Framatome confidential,** such as its disclosure, alteration or loss are detrimental with a significant-to-high impact for Framatome.

The document, if disclosed, and any information it contains are intended for the sole attendees. The disclosure or reference to such information or document shall be made only on a proper judgment basis and by mentioning expressly "this information shall not be disclosed / transferred without prior consent".

C3 : This document and any and all information contained therein and/or disclosed in discussions supported by this document are classified **Framatome Secret**.

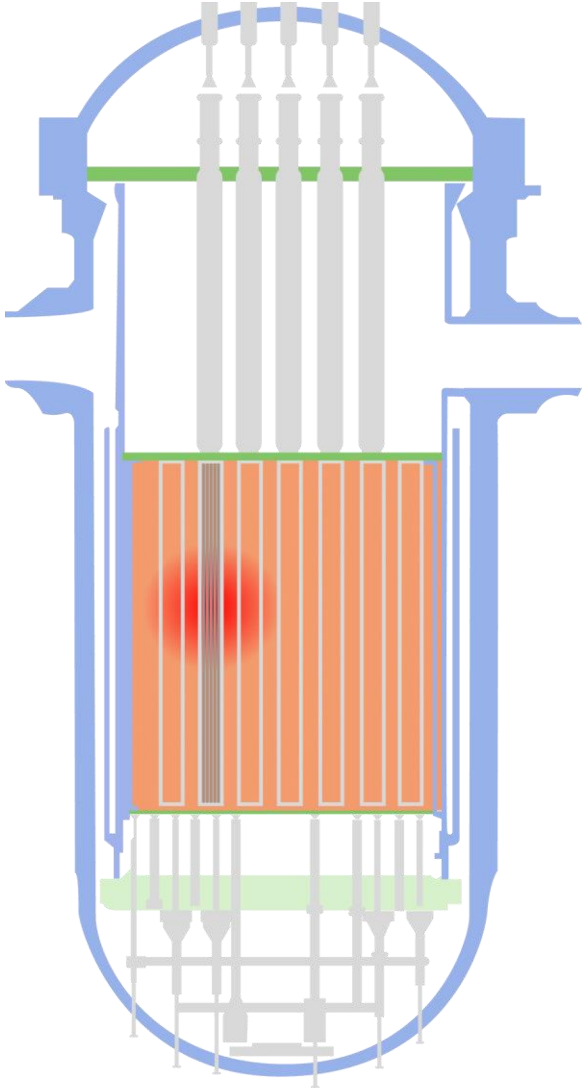
Each one must commit to keep secret any written or oral information disclosed during the meeting. It is forbidden to disclose it to any legal entity and any individual (including within Framatome) without prior consent of the meeting chairman.

Diving into the reactor core

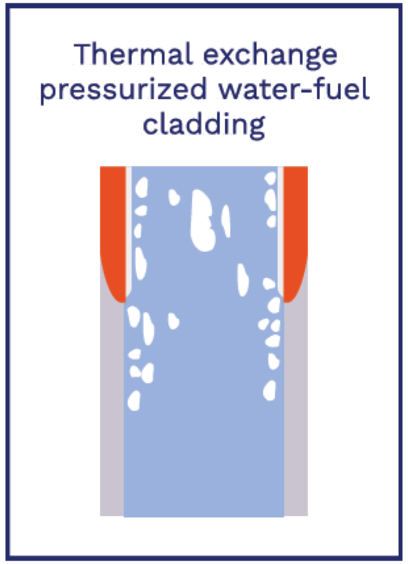


- Safe operation of nuclear reactors requires respecting a certain margin on physical quantities that are subject to regulatory safety criteria.
- Physical quantities like temperature, pressure or thermal power **can be measured**.
- More complex physical quantities can be **calculated** but **not directly measurable**.

Diving into the reactor core

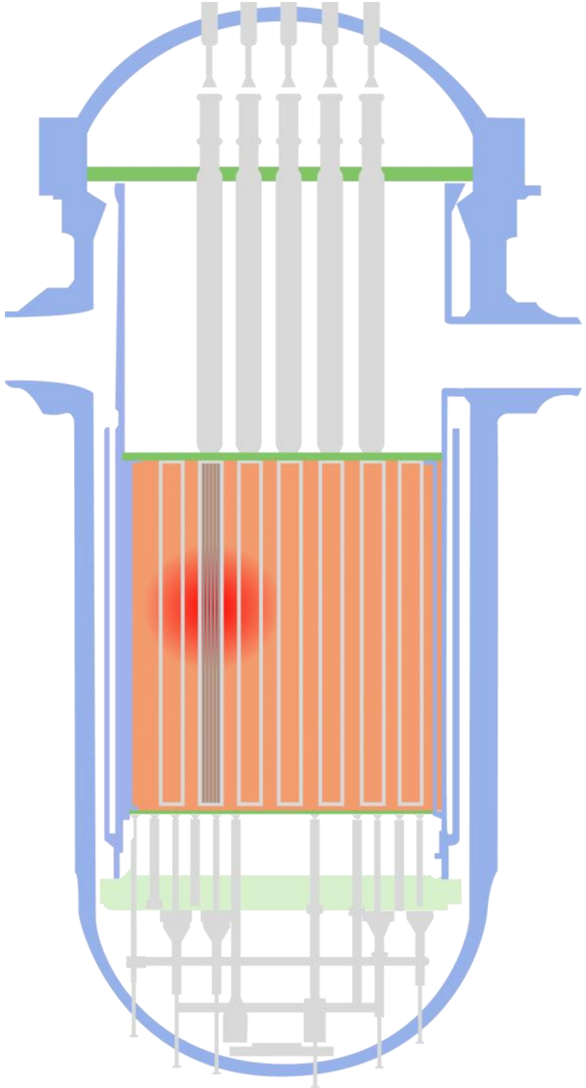


- Safe operation of nuclear reactors requires respecting a certain margin on physical quantities that are subject to regulatory safety criteria.
- Physical quantities like temperature, pressure or thermal power **can be measured**.
- More complex physical quantities can be **calculated** but **not directly measurable**.

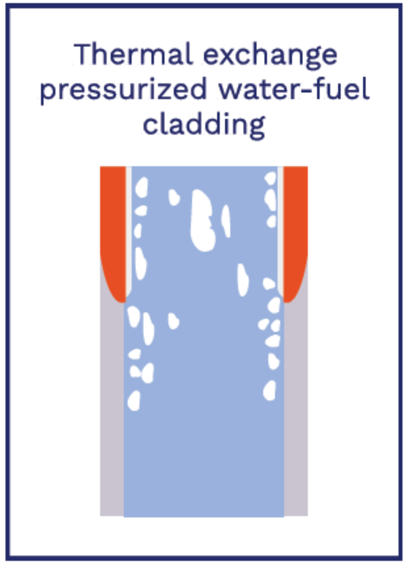


- Departure from Nucleate Boiling Ratio or DNBR is a physical quantity which gives the margin with respect to the boiling crisis in the reactor core that could appear in an accidental situation.

Diving into the reactor core



- Safe operation of nuclear reactors requires respecting a certain margin on physical quantities that are subject to regulatory safety criteria.
- Physical quantities like temperature, pressure or thermal power **can be measured**.
- More complex physical quantities can be **calculated** but **not directly measurable**.



- Departure from Nucleate Boiling Ratio or DNBR is a physical quantity which gives the margin with respect to the boiling crisis in the reactor core that could appear in an accidental situation.

- DNBR is assessed by numerical Thermal-Hydraulic (TH) 3D two-phases reference code on High Performance Computing (HPC) for the safety analysis report.

- DNBR is assessed by the simplified algorithm in the current embedded safety system.

Context

- **Thermal-Hydraulic 3D two-phases reference** calculation is expected to be **very close to the "reality"** but **cannot be used** in the safety system (too long, convergence issues, etc.). "It is very precise but too slow".

- **DNBR simplified algorithm** calculation by the safety system based on simple physics laws is **penalized** to be sure to assess the worst case. But it involves a **loss of margins.** "It is less precise but very fast".

- **The main idea** is to restore some margins (i.e. be close to the "reality") using **Deep Learning** (DL) techniques **embedded** in the safety system and to be very fast.

Questions

- How can we build such Deep Learning model?
- How can we estimate the performance of this model and the uncertainty of the prediction?
- Can we evaluate the quality/confidence of the uncertainty estimation?
- Can we apply these uncertainty methods for critical applications like in nuclear reactor core?

Content

1. Artificial Neural Network in nuclear safety system

2. Overview of uncertainty estimation methods in Deep Learning

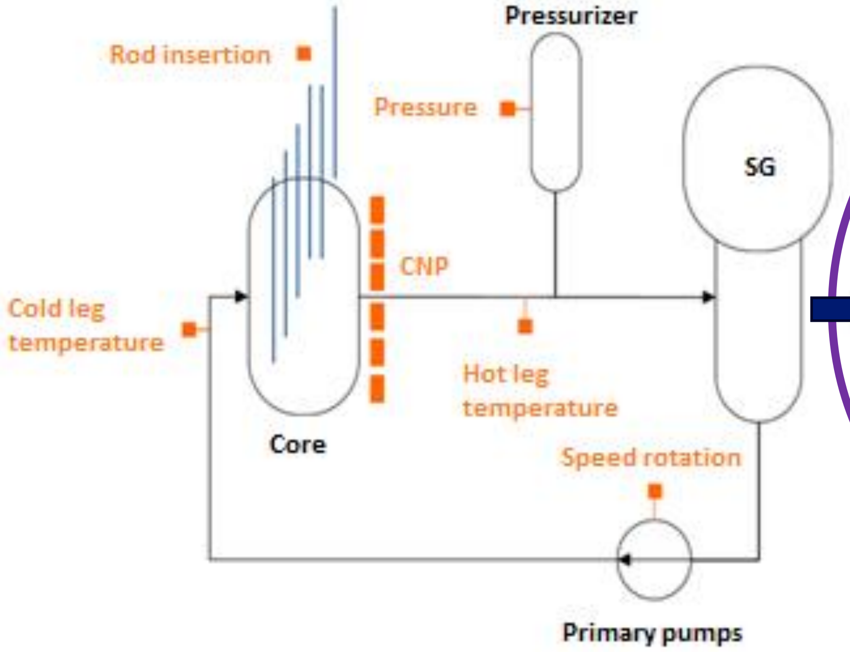
3. Application to DNBR safety system

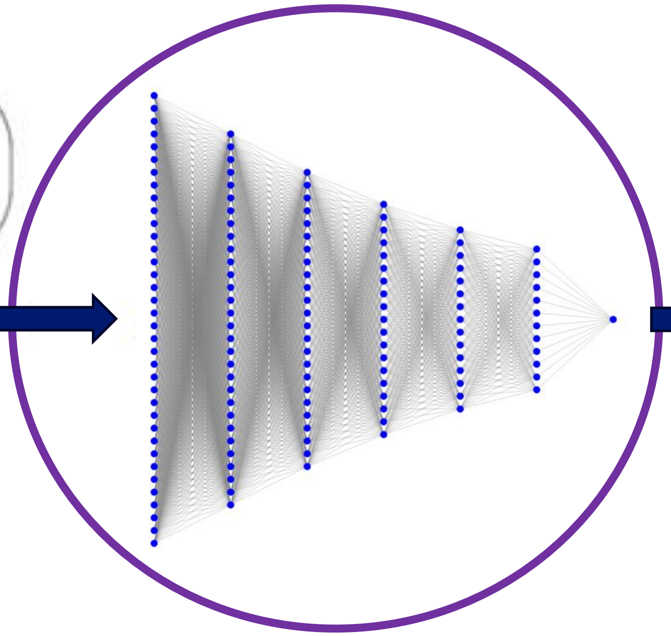
4. Conclusions and open questions for critical applications

Artificial Neural Network in nuclear safety system

Deep Learning in safety system

- DNBR Box is an Artificial Neural Network trained on millions of Thermal-Hydraulic 3D two-phases reference accidental simulations and implemented in a Field-Programmable Gate Array (FPGA) integrated circuit in order to predict the DNBR using the same inputs as the current simplified algorithm used in the safety system. “DNBR Box is very precise, can restore margins and it is very fast.”

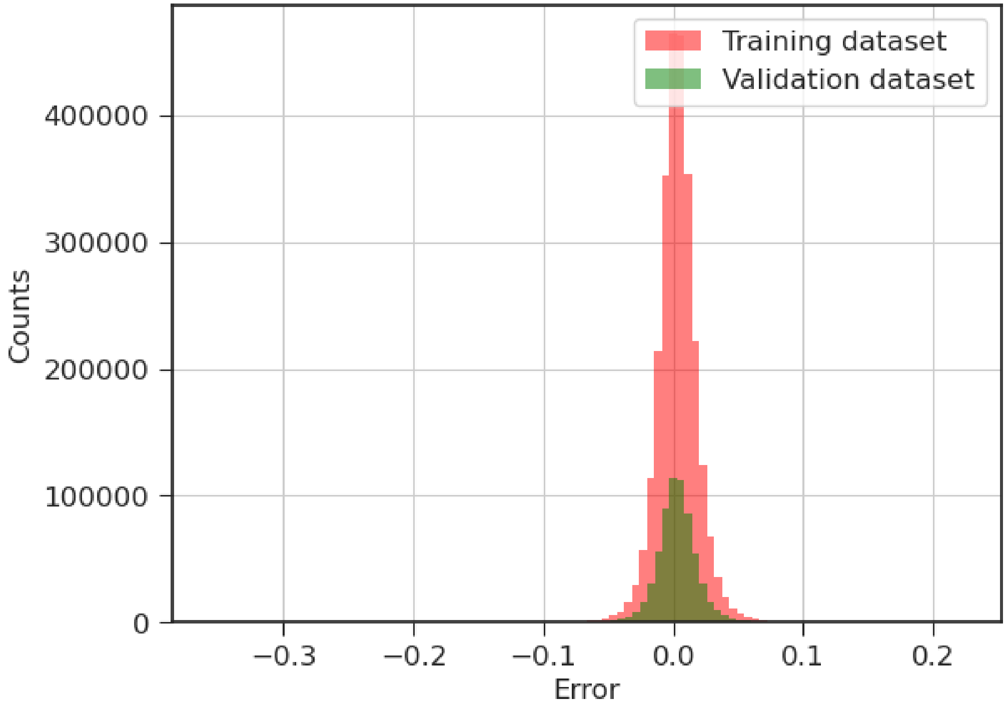


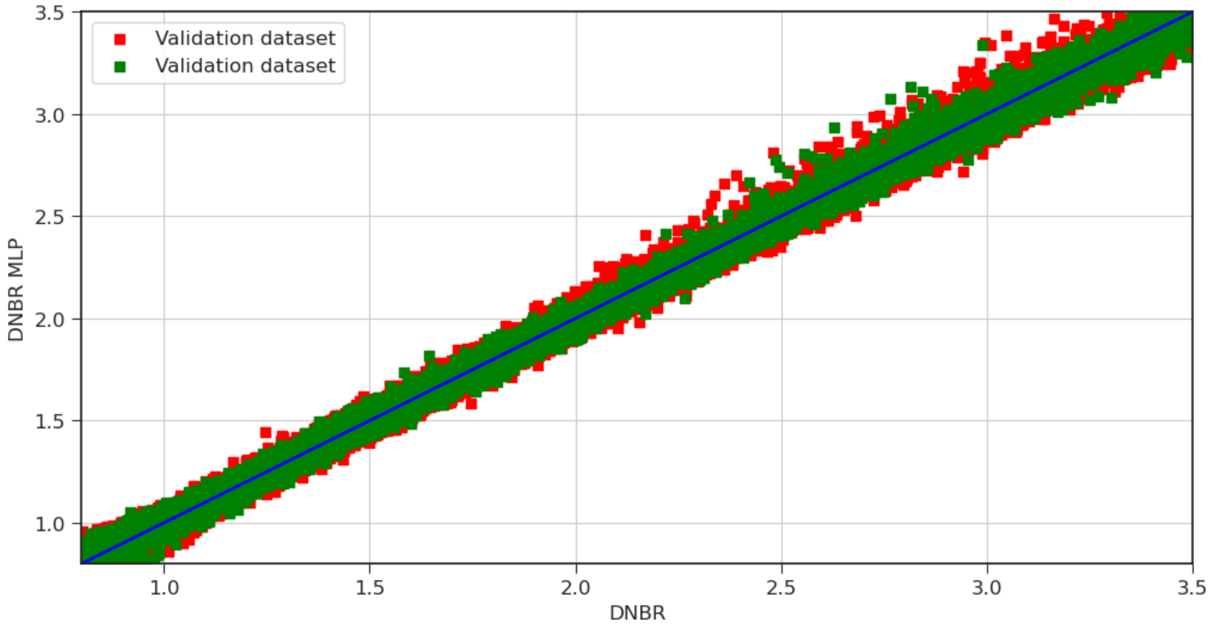


DNBR Box = Artificial Neural Network (ANN)

DNBR Box performances

- Learning and validation of DNBR box: database of millions of 3D core state-points issued from neutronics-Thermal-Hydraulics reference code simulations.
- Very good accuracy from performance metrics.

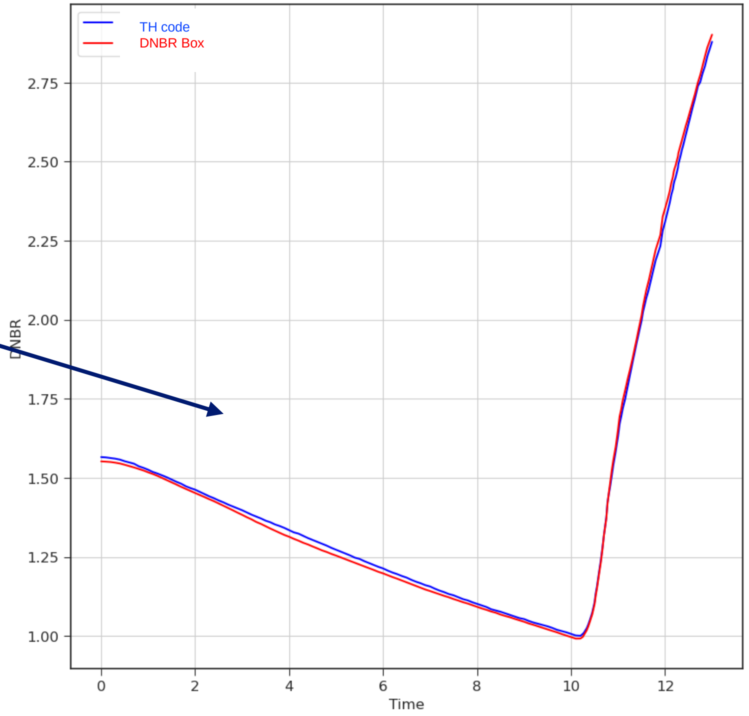




Metrics	Learning base (80 %)	Validation base (20 %)
Mean Squared Error (MSE)	0.000244	0.000246
$Q^2 = 1 - {}^{MSE}/_{Var[DNBR]}$	0.9994	0.9994

DNBR Box performances

- Use 3 other datasets to test DNBR Box:
  - Testing base (in-domain base);
  - Phenomenological base: accidental situation simulations;
  - Out-Of-distribution (OOD) base: real normal operating base (coming from an EDF nuclear power plant). DNBR Box is trained on accidental conditions, not normal ones.

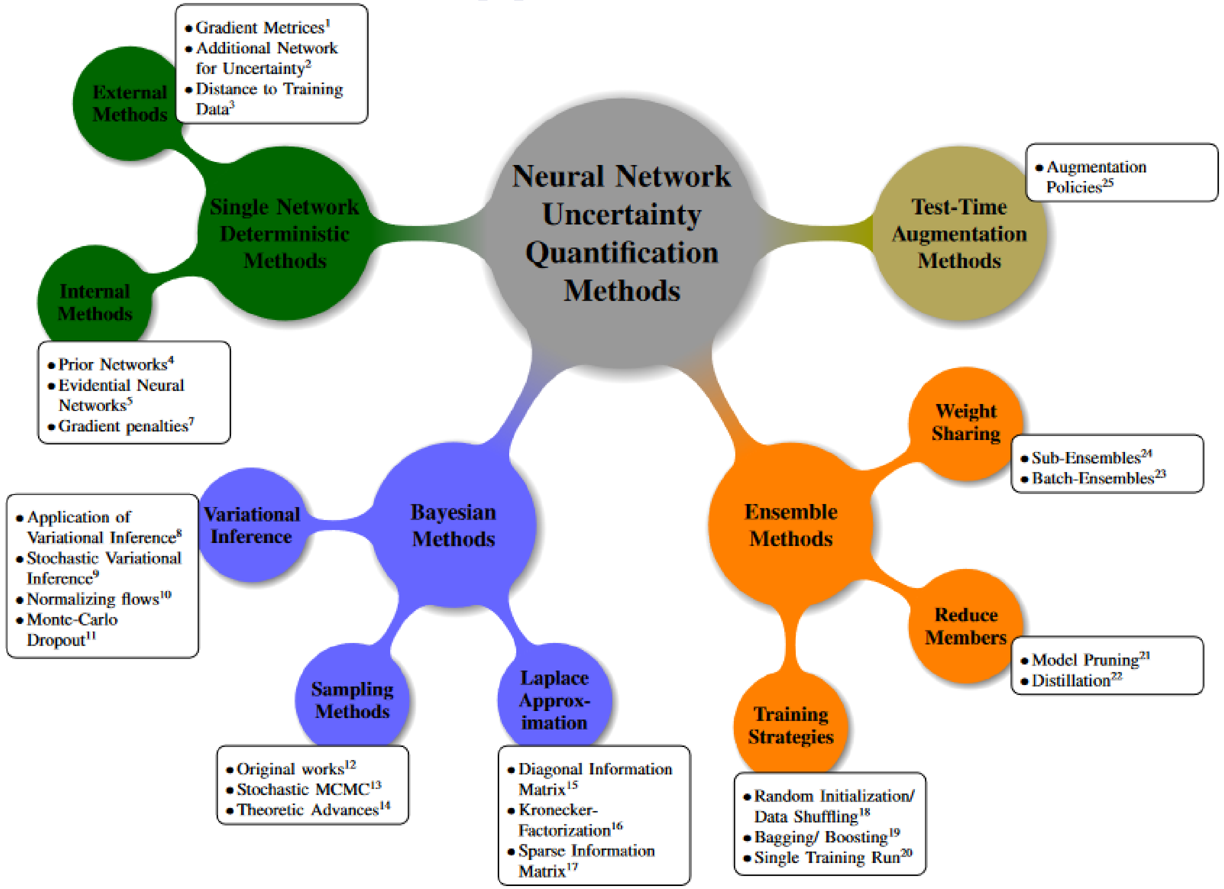


Metrics	Learning base (80 %)	Validation base (20 %)	Testing base	Phenomenological base	Out-Of-Distribution base
MSE	0.000244	0.000246	0.000836	0.000369	0.000045
Q^2	0.9994	0.9994	0.9971	0.9987	0.8330

Overview of uncertainty estimation methods in Deep Learning

Uncertainty quantification methods in Deep Learning

- Several Uncertainty Quantification (UQ) methods exist depending on the number of inference (also called forward pass) or based on the nature (deterministic or stochastic) of the model, as proposed by Gawlikowski et al [8].



- Differences in theoretical framework, in estimation time, in theoretical guarantees, in programming implementation, etc.

- Focus on:
  - Bayesian Neural Network (BNN) [1];
  - Deep Ensembles (DE) [2];
  - Monte-Carlo Dropout (MCDo) [3];
  - Mixture Density Network (MDN) [4, 9];
  - Conformal Calibration/Prediction (CC) (not exclusively dedicated to DL) [5, 6, 7].

Reference [8]: Jakob Gawlikowski et al., A survey of uncertainty in deep neural networks, arXiv:2107.03342, 2022.

Common theoretical framework

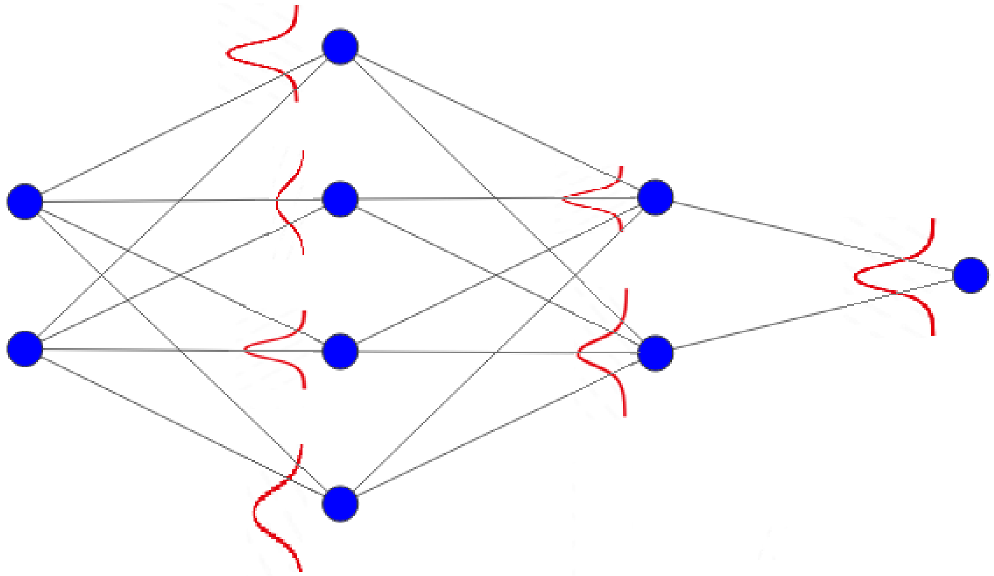
- Predictive distribution for a prediction y given a dataset D and model parameters θ is defined as:

$$p(y|x,D) = \int \underbrace{p(y|x,\theta)}_{\text{Data}} \underbrace{p(\theta|D)}_{\text{Model}} d\theta.$$

- $p(\theta|D)$ is intractable in pratice => Need approximations like variational inference, Markov Chain Monte-Carlo (MCMC), Laplace approximation, Monte-Carlo Dropout, etc.
- $p(y|x,D)$ is also intractable => Need approximations like Deep Ensembles, Mixture Density Network, etc.

Bayesian Neural Network

- Bayesian network methods treat weights in neural networks as probability distributions rather than fixed values.



- In Bayesian network [1]: $p(\theta|D)$ is approximated by another distribution $q(\theta)$ (variational inference) considered as Multivariate Normal distribution. The training consists in minimizing the error made by this approximation quantified by the Kullback-Leibler (KL) divergence:

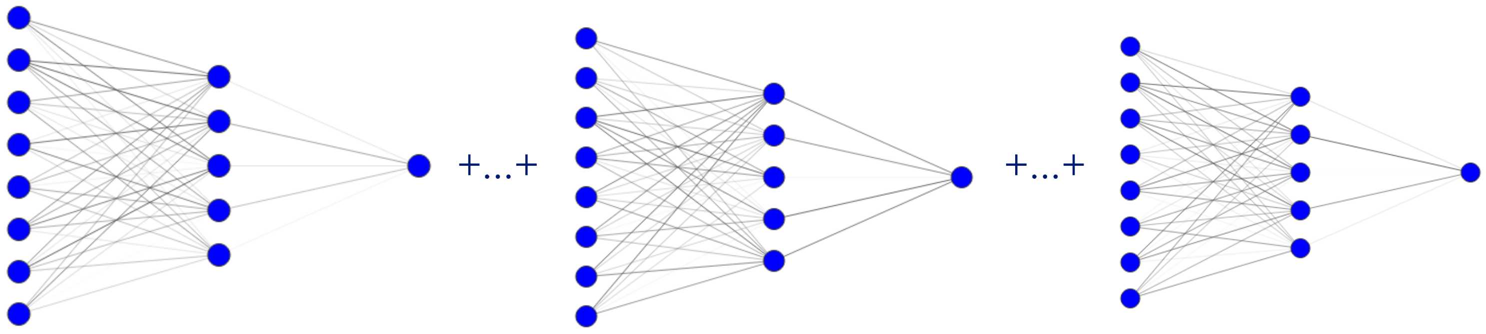
$$\mathrm{DL}(q||p) = \mathbb{E}_q\left[log\,\frac{q(\theta)}{p(\theta|D)}\right].$$

- But $p(\theta|D)$ is not known, so the training tries to minimize the Evidence Lower Bound (ELBO):

$$\mathrm{ELBO} = \mathbb{E}_q\left[log\,\frac{p(D|\theta)}{q(\theta)}\right]$$ where $p(D|\theta)$ is the likelihood.

Monte-Carlo Dropout

- In Monte-Carlo Dropout [3], the posterior distribution $p(y|x, D)$ is approximated by applying (Bernoulli) dropout during both training and inference and then performing multiple forward passes with different dropout masks m_t to obtain a distribution of predictions $y_t = f(x, \theta, m_t)$ for $t = 1, \dots, T$.



- The final prediction is taken as the mean of the T stochastic forward passes, and the predictive uncertainty can be estimated from the variance of the predictions:

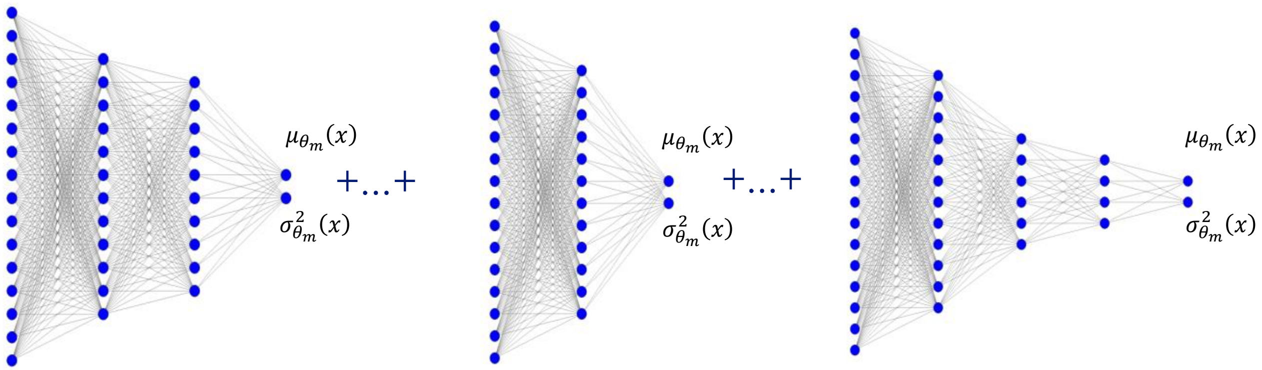
$$\begin{cases} \mathbb{E}[y] \approx \frac{1}{T}\sum_{t=1}^{T} y_t \\ Var[y] \approx \frac{1}{T}\sum_{t=1}^{T}(y_t - \mathbb{E}[y])^2 \end{cases}.$$

- The model is trained using common loss function (MSE, Negative Log-Likelihood for instance).

Deep Ensembles

- In Deep Ensembles [2], the posterior distribution $p(y|x,D)$ is approximated by learning multiple parameter settings (different neural structures, parameters initialization and training dataset) and averaging over resulting M models:

$$p(y|x,D) = \frac{1}{M}\sum_{m=1}^{M} p_{\theta_m}(y|x,\theta_m).$$



- $p_{\theta_m}(y|x,\theta_m)$ is approximated by a Gaussian distribution whose mean and variance are respectively the mean and variance of the mixture $M^{-1}\sum_m \mathcal{N}\left(\mu_{\theta_m}(x), \sigma^2_{\theta_m}(x)\right)$ given by:

$$\begin{cases} \mu(x) = M^{-1}\sum_m \mu_{\theta_m}(x) \\ \sigma^2(x) = M^{-1}\sum_m \left(\sigma^2_{\theta_m}(x) + \mu^2_{\theta_m}(x)\right) - \mu^2(x) \end{cases}.$$

- The models are trained using the Gaussian Negative Log-Likelihood (NLL) loss in order to learn μ_{θ_m} and $\sigma^2_{\theta_m}$.

Brief focus on aleatoric and epistemic uncertainties

- Predictive distribution $p(y|x,D)$ is in general separated into aleatoric uncertainty (also called data uncertainty) and epistemic uncertainty (also called model uncertainty).
- It means that we expect that far from the data, the epistemic uncertainty is higher than the aleatoric one.
- Deep Ensembles give a way to assess aleatoric and epistemic uncertainties:

$$\sigma^2(x) = M^{-1}\sum_m \left(\sigma^2_{\theta_m}(x) + \mu^2_{\theta_m}(x)\right) - \mu^2(x)$$

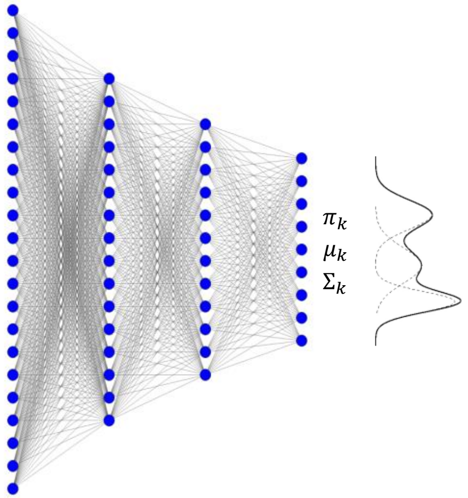
$$= M^{-1}\sum_m \sigma^2_{\theta_m}(x) + M^{-1}\sum_m \left(\mu^2_{\theta_m}(x) - \mu^2(x)\right)$$

$$= \underbrace{\mathbb{E}[\sigma^2_{\theta_m}]}_{\text{Aleatoric}} + \underbrace{Var[\mu^2_{\theta_m}(x)]}_{\text{Epistemic}}.$$

Mixture Density Network

- In Mixture Density Network [4, 9], the posterior distribution $p(y|x,D)$ is assumed to be composed of parameters constructing a Gaussian mixture mode:

$p(y|x,D) = \sum_{k=1}^{K} \pi_k(x)\mathcal{N}(y|\mu_k(x),\Sigma_k(x))$ where $\pi_k(x)$ are the mixture weights (probabilities) with K Gaussian, $\mu_k(x)$ the means and Σ_k the covariances.



- The final prediction is taken as the mean of the Gaussian mixture and the predictive uncertainty can be estimated from its variance:

$$\begin{cases} \mathbb{E}[y|x] = \sum_{k=1}^{K} \pi_k(x)\,\mu_k(x) \\ Var[y|x] = \underbrace{\sum_{k=1}^{K} \pi_k(x)\,\Sigma_k(x)}_{\text{Aleatoric}} + \underbrace{\sum_{k=1}^{K} \pi_k(x)\left\|\mu_k(x) - \sum_{i=1}^{K} \pi_i(x)\,\mu_i(x)\right\|^2}_{\text{Epistemic}}. \end{cases}$$

- The model is trained using the NLL of the observed data weighted by the mixture weights.

Conformal calibration/prediction

- It is based on the idea of using a training set to generate a set of ”conformity scores” that capture the similarity between a test example and the training examples. These conformity scores are used to quantify the uncertainty associated with each prediction.
- Split conformal prediction is the most widely-used version of conformal prediction.
- It is a straightforward way to generate prediction sets for any model. Suppose $(X_i, Y_i)_{i=1,\dots,n}$ and (X_{test}, Y_{test}) are i.i.d (or weaker condition of exchangeability).
- Then $\hat{q}$ is defined as:

$$\hat{q} = \inf\left\{q : \frac{|\{i : s(X_i, Y_i) \leq q\}|}{n} \geq \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right\}.$$

- The resulting prediction sets is defined as:

$C(X) = \{y : s(X, y) \leq \hat{q}\}$ where $s(X, y)$ are calibration or conformal scores.

- Then:

$$P\left(Y_{test} \in C(X_{test})\right) \geq 1 - \alpha.$$

- Cf. Conformal online model aggregation talk of Aaditya Ramdas.

Application to DNBR safety system

Quality of the uncertainty estimation

- Prediction Interval Coverage Probability (PICP) represents the percentage of test predictions that fall into a prediction interval, defined as:

$$PICP = \frac{c}{n}$$

where n is the total number of predictions and c the number of ground truth values that are actually captured by the predicted intervals.

- Mean Prediction Interval Width (MPIW) evaluates the average certainty of the model:

$$MPIW = \frac{1}{n}\sum_{i=1}^{n}\left(y_{upper_i} - y_{lower_i}\right).$$

- Signed Quantile Calibration Error (sQCE):

$$sQCE = \frac{1}{M}\sum_{m=1}^{M}\left(p_{obs}(\rho_m) - \rho_m\right)$$

where M is the number of confidence levels ρ_m and $p_{obs}(\rho_m)$ is the observed probability calculated as the fraction of predictions that fall into the ρ_m -confidence interval of their respective predictive distributions.

Results for OOD dataset

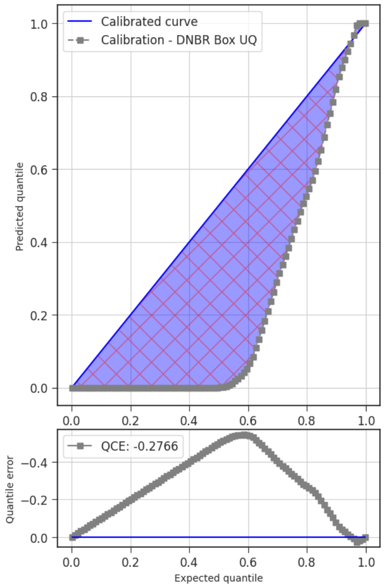
- DNBR Box is designed to be efficient during accidental conditions (it is trained on accidental core configurations). But in real life, the occurrence of transients is very rare, but cannot be ignored. So, DNBR Box will be used mostly in normal conditions, so in OOD.

Methods	Mean squared error (MSE)	Mean absolute error (MAE)	Prediction interval coverage probability (PICP) at 95 %	Mean prediction interval width (MPIW) at 95 %	Signed Quantile Calibration Error (sQCE)	Negative Log Likelihood (NLL)
DNBR Box	**0.000045**	**0.0061**	-	-	-	-
MDN	0.000075	0.0076	100.00	0.0792	0.212	-3.81
BNN	0.000486	0.028	**94.40**	**0.0563**	-0.282	-3.33
MCDo	0.004052	0.064	100.00	0.204	0.236	-2.90
DE	0.000490	0.016	100.00	0.101	**0.0983**	-3.51
CC	**0.000045**	**0.0061**	100.00	0.0627	0.206	**-4.05**

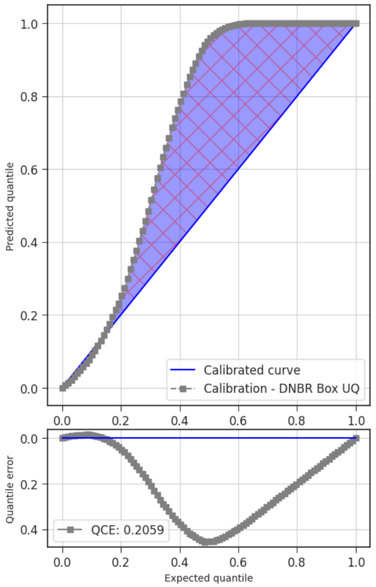
- No clear insight of a better method with respect to another, as already observed in literatures [17, 19, 21, 24, 25, 26].

Results for OOD dataset

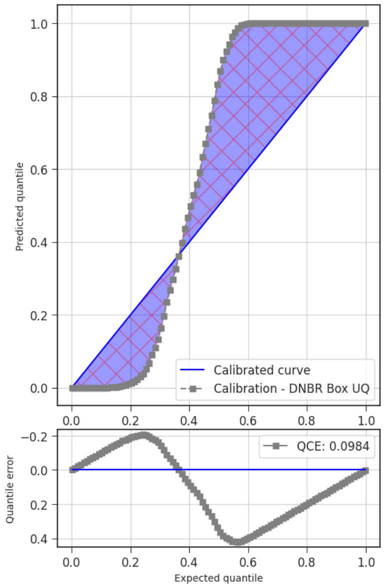
- All the methods don’t give calibrated uncertainty (cf. Gaël Varoquaux talk).



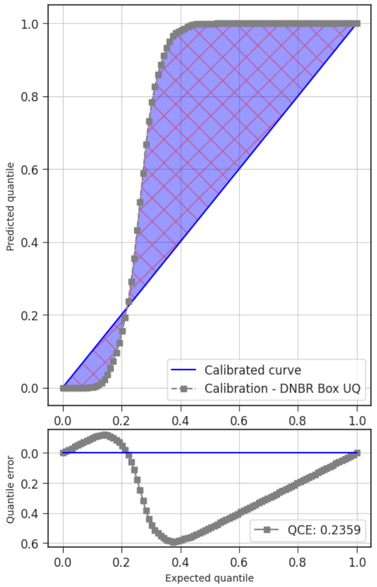
BNN



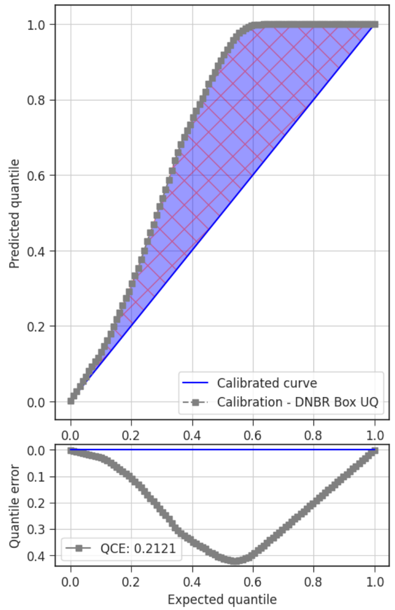
CC



DE



MCDo



MDN

Results for OOD dataset

Features	MDN	BNN	MCDo	DE	CC
Prediction accuracy	✓ (very good)	× (small bias)	× (model is too small)	✓ (very good)	✓ (best)
Quality of the uncertainty estimation	× (over-estimation)	✓ (good estimation)	× (bad-estimation)	× (over-estimation)	× (as expected because the model is not calibrated on this dataset; works perfectly on calibrated dataset)
Well-calibrated	×	×	×	×	× (as expected because the model is not calibrated on this dataset; works perfectly on calibrated dataset)
Easy to implement	✓	×	✓	✓	✓
Easy to train	✓	×	✓	✓	✓
Easy to use	× (increase the output dimension)	× (increase the number of parameters)	✓	× (heavy computations)	✓

Conclusions and open questions for critical applications

Conclusions

- It is possible to assess complex physical quantities like DNBR using DL techniques.
- DNBR Box computes the DNBR as precise as the TH reference code and faster than the simplified algorithm used in the safety system.

- There are many Uncertainty Quantification methods for DL with pros and cons.
- For DNBR Box, there is no clear agreement about the technique to use.

Open questions

- For critical applications like in nuclear safety, aeronautics, autonomous vehicle, medicine, etc. it is important (even mandatory regarding the safety authorities) **to be confident** on the **predictions** of the model and on **its uncertainties.**

- Can we train a model to predict with a good accuracy and in the same time to estimate with a good confidence the uncertainty?

- Can we get a theoretical bound of the error?

- Can we trust the uncertainty estimation in test time, OOD time?

- Can we implement these methods on specific hardware like Field Programmable Gate Array (FPGA)?

Open questions

- To tackle these problems, Eline Pot will begin a PhD thesis with Framatome (M. Segond, L. Lefebvre), EDF R&D (M. Keller) and CentraleSupélec (J. C. Pesquet).

- We plan to:
    - study some current UQ methods, perform benchmarks (in-domain and OOD), discuss some performance metrics, etc.
    - develop some new UQ methods if necessary to be compliant with embedded system.
    - study of Lipschitz regularity of deep neural networks [10, 11] for robustness to noisy inputs data.

framatome

Thank you

Bibliography

- [1] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, Daan Wierstra, Weight Uncertainty in Neural Networks, arXiv:1505.05424, 2015.
- [2] Balaji Lakshminarayanan, Alexander Pritzel, Charles Blundell, Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles, arXiv:1612.01474, 2017.
- [3] Yarin Gal, Zoubin Ghahramani, Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, arXiv:1506.02142, 2015.
- [4] Christopher M. Bishop, Mixture Density Networks, Neural Computing Research Group Report, 1994.
- [5] Anastasios N. Angelopoulos, Rina Foygel Barber, Stephen Bates, Theoretical Foundations of Conformal Prediction, arXiv:2411.11824, 2024.
- [6] Anastasios N. Angelopoulos, Stephen Bates, A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification, arXiv:2107.07511, 2022.
- [7] V. Vovk, A. Gammerman, and G. Shafer, Algorithmic Learning in a Random World. Springer, 2005.
- [8] Jakob Gawlikowski et al., A survey of uncertainty in deep neural networks, arXiv:2107.03342, 2022.
- [9] Sungjoon Choi, Kyungjae Lee, Sungbin Lim, Songhwai Oh, Uncertainty-Aware Learning from Demonstration using Mixture Density Networks with Sampling-Free Variance Modeling, arXiv:1709.02249, 2017.

Bibliography

- [10] Patrick L. Combettes and Jean-Christophe Pesquet, Lipschitz certificates for layered network structures driven by averaged activation operators, SIAM Journal on Mathematics of Data Science, 2(2):529–557, 2020.
- [11] Kavya Gupta, Fateh Kaakai, Beatrice Pesquet-Popescu, Jean-Christophe Pesquet, and Fragkiskos D. Malliaros, Multivariate lipschitz analysis of the stability of neural networks. Frontiers in Signal Processing, 2, 2022.
- [12] Javier Antoran, James Urquhart Allingham, and Jose Miguel Hernandez-Lobato, Depth uncertainty in neural networks, 2020.
- [13] G. E. Hinton D. E. Rumelhart and R. J. Williams, Learning Internal Representations by Error Propagation. Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations, MIT Press, 1986.
- [14] Nikita Durasov, Timur Bagautdinov, Pierre Baque, and Pascal Fua, Masksembles for uncertainty estimation, 2021.
- [15] Vikranth Dwaracherla, Xiuyuan Lu, Morteza Ibrahimi, Ian Osband, Zheng Wen, and Benjamin Van Roy, Hypermodels for exploration, 2020.
- [16] Gianni Franchi, Andrei Bursuc, Emanuel Aldea, Severine Dubuisson, and Isabelle Bloch, Encoding the latent posterior of bayesian neural networks for uncertainty quantification, 2021.
- [17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger, On calibration of modern neural networks, 2017.

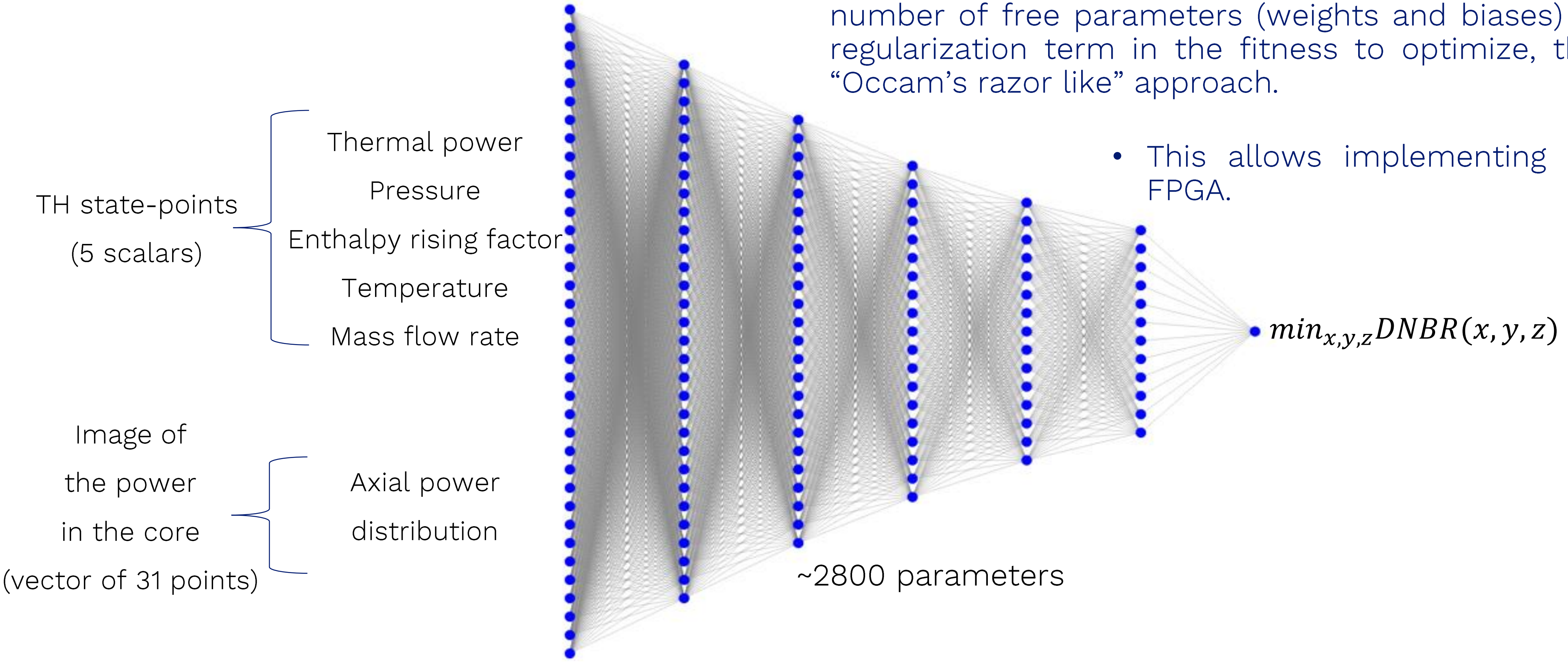
Bibliography

- [18] Wenchong He and Zhe Jiang, A survey on uncertainty quantification methods for deep neural networks: An uncertainty source perspective, 2023.
- [19] V. Kuleshov, N. Fenner, and S. Ermon, Accurate uncertainties for deep learning using calibrated regression. In International Conference on Machine Learning, pages 2796–2804. PMLR, 2018.
- [20] Juho Lee, Yoonho Lee, Jungtaek Kim, Eunho Yang, Sung Ju Hwang, and Yee Whye The, Bootstrapping neural processes, 2020.
- [21] Hendrik A. Mehrtens, Alexander Kurz, Tabea-Clara Bucher, and Titus J. Brinker, Benchmarking common uncertainty estimation methods with histopathological images under domain shift and label noise, 2023.
- [22] Dmitry Molchanov, Alexander Lyzhov, Yuliya Molchanova, Arsenii Ashukha, and Dmitry Vetrov, Greedy policy search: A simple baseline for learnable test-time augmentation, 2020.
- [23] Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy, Epistemic neural networks, 2023.
- [24] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek, Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019.
- [25] Florian Seligmann, Philipp Becker, Michael Volpp, and Gerhard Neumann, Beyond deep ensembles: A large-scale evaluation of bayesian deep learning under distribution shift, 2023.
- [26] Brian Staber and Sébastien Da Veiga, Benchmarking bayesian neural networks and evaluation metrics for regression tasks, 2023.

DNBR Box neural structure

- Neural structure hyper-optimization using genetic algorithm: find the best model with respect to the Mean Squared Error (MSE) while minimizing the complexity of the neural network, i.e. the number of free parameters (weights and biases) by introducing a regularization term in the fitness to optimize, thus following an “Occam’s razor like” approach.

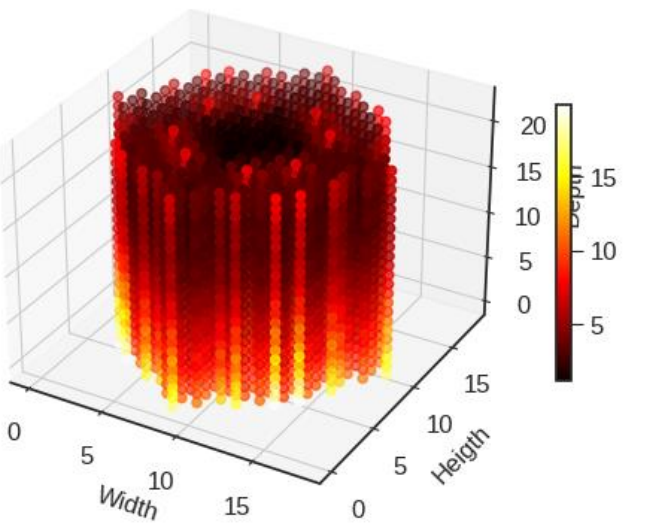
- This allows implementing DNBR Box in a FPGA.



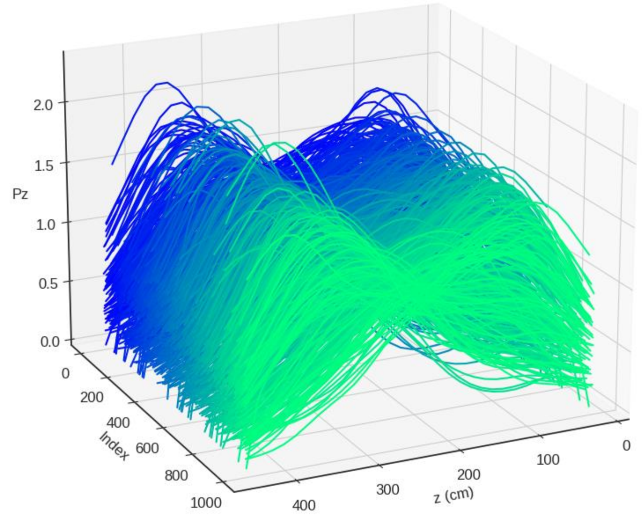
DNBR Box training phase

- Lack of online physical modeling must be compensated by a big and representative database to train and validate the model.
- A database is sampled on 4-loops Nuclear Power Plant designs (1450 MWe and 1300 MWe reactors):
  - o 1000 Thermal-Hydraulic state-points ⊗ 6000 axial power distributions computed by the core neutronics-TH 3D simulation code => 6 x 10^6 data 3D core simulations.
  - o Data filtering on DNBR range of interest for safety.
  - o Leads to 3 x 10^6 core state-points data => big data set to process.
  - o Representative of accidental/incidental transients.

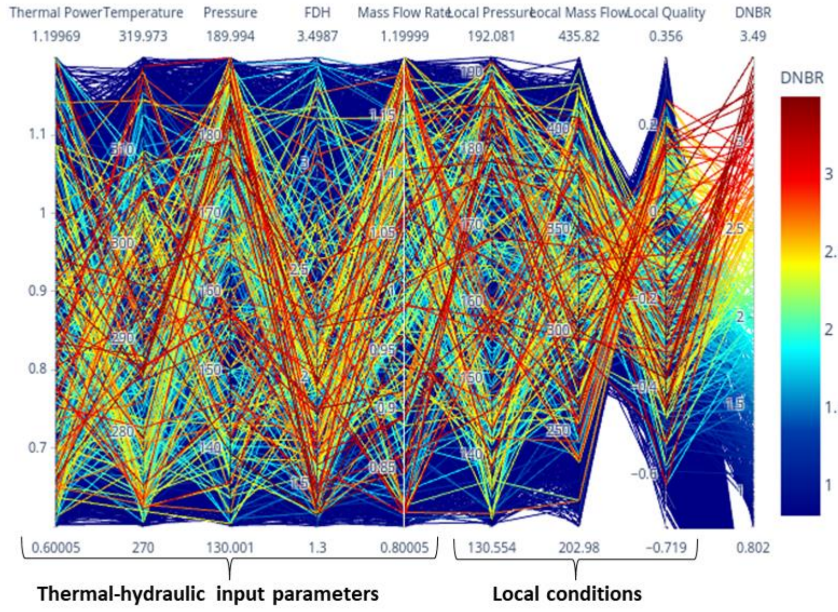
True 3D DNBR



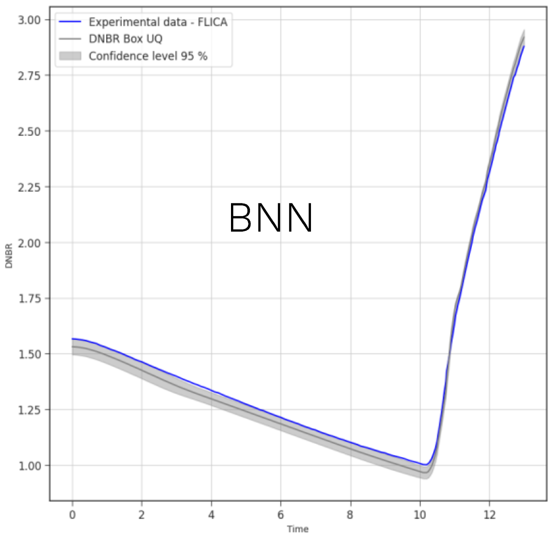
Axial power distributions

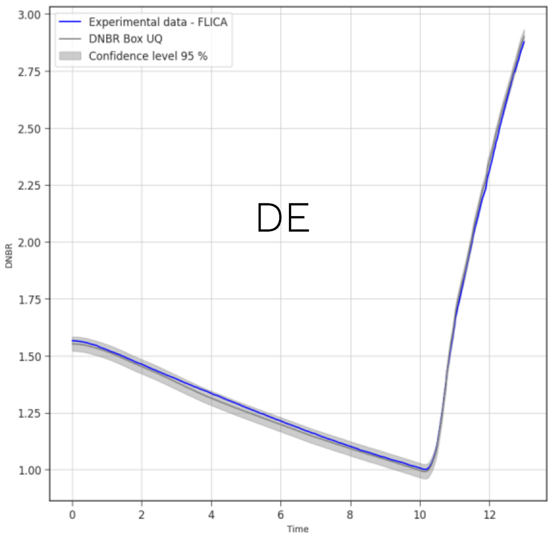


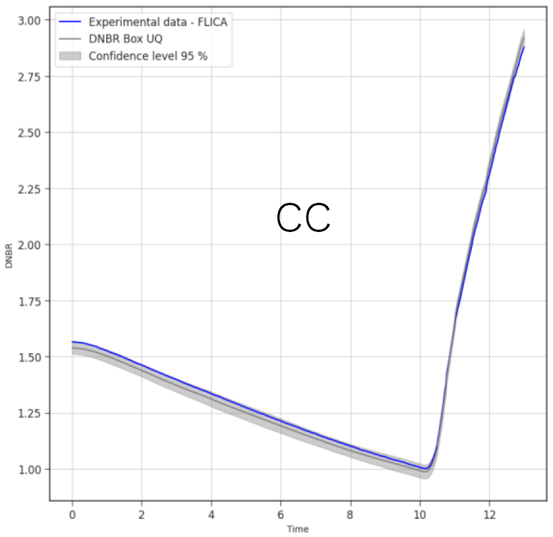
Cobweb

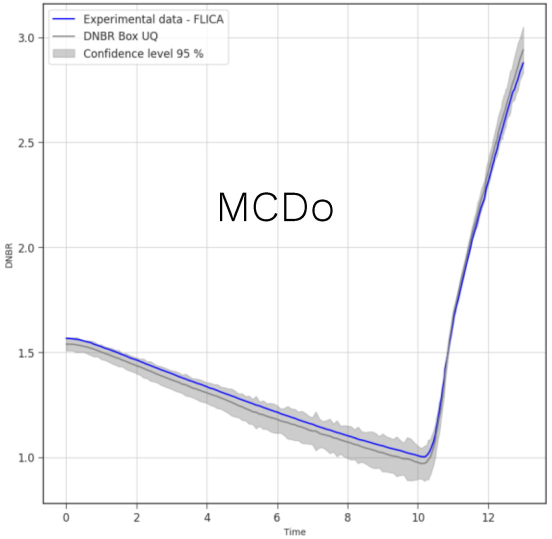


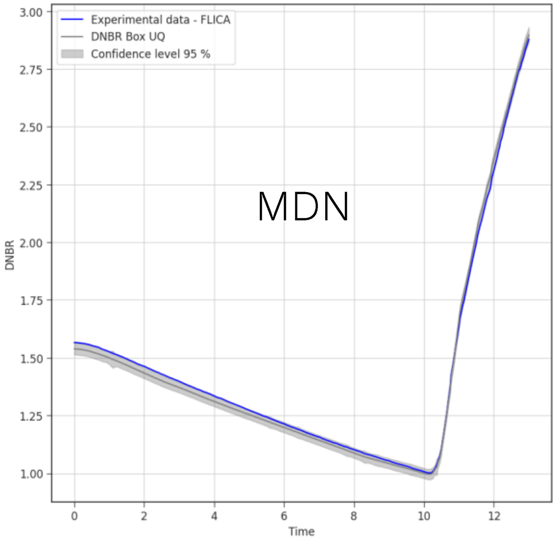
Benchmark on phenomenological dataset











Benchmark on phenomenological dataset

- No clear insight.

Methods	Mean squared error (MSE)	Mean absolute error (MAE)	Prediction interval coverage probability (PICP) at 95 %	Mean prediction interval width (MPIW) at 95 %	Signed Quantile Calibration Error (sQCE)	Negative Log Likelihood (NLL)
DNBR Box	**0.00037**	**0.0167**	-	-	-	-
MDN	0.000555	0.021	56.35	**0.0481**	-0.306	-2.60
BNN	0.00137	0.035	23.20	0.0575	-0.463	-0.74
MCDo	0.000201	**0.0125**	**94.47**	0.120	-0.233	-2.75
DE	0.00069	0.024	78.45	0.0605	-0.333	-2.87
CC	**0.00037**	0.0167	89.50	0.0627	**-0.126**	**-3.41**