International Journal of Heat and Mass Transfer 263 (2026) 128675

Contents lists available at ScienceDirect

International Journal of Heat and Mass Transfer

journal homepage: www.elsevier.com/locate/ijhmt

Development and assessment of the uncertainty-aware data-informed continual machine learning approach

Fabian Wiltschko *, Zhichao Zhang, Aurelian F. Badea, Xu Cheng

Institute of Applied Thermofluidics (IATF), Karlsruhe Institute of Technology (KIT), Kaisertrasse 12, 76131 Karlsruhe, Germany

ARTICLE INFO

Keywords:
Continual machine learning
Critical heat flux (CHF)
Uncertainty quantification
Deep, ensemble
Monte Carlo dropout
SWAG
Bayesian uncertainty quantification

ABSTRACT

For prediction of complex heat transfer phenomena, such as critical heat flux (CHF), machine learning models gain more attraction in the research community. Throughout the institutions, a large amount of experimental data exists, but the data is often confidential, so it is not possible to create a comprehensive database. Continual training of a machine learning model, shared from institution to institution, could tackle/mitigate this problem. Recently, a data-informed continual machine learning (DI-CML) approach has been developed. In the present work, this method is enhanced by considering the uncertainty of the machine learning model to filter the soft data, which is generated in that approach. The Monte Carlo Dropout method, deep ensembles and the stochastic weight averaging Gaussian (SWAG) are used to approximate the Bayesian uncertainty. Assuming that machine learning models provide low uncertainty if applied to data overlapping with the data used in its training, the generated soft data is first filtered for low uncertainty and then used to continue the training of the machine learning model. Doing so, the statistical validity domain of the trained model is identified without reconstructing or disclosing the original experimental distribution. Besides the parameter ranges, no additional information about the training data needs to be shared. Furthermore, the proposed method maintains predictive performance while reducing the amount of shared information compared to the DI-CML method.

1. Introduction

As the critical heat flux (CHF) plays an important role in heat transfer systems [1,2], reliable prediction has been in the focus of the research community for decades [3,4]. Most commonly, empirical correlations [4] and look-up tables [5] are used, with the known shortcomings. Also, the application of mechanistic models has been widely tested [6], but further improvement is required here. Recently, machine learning approaches were in the focus of the research community. Shortly, OECD/NEA organized a benchmark exercise [7], with over 30 international contributors. Also recently Ahmed [8] proposed an ensemble of neural networks for the prediction of the CHF. Nevertheless, neural networks for the prediction of the CHF have already been used in the 1990s [9,10]. A review of state-of-the-art AI methods for CHF prediction is published by Zhou et al. [11]. However, a machine learning model can only be as good as the underlying database used to train it. Experimental data is often confidential, or only parts of databases are shared with the community. Nowadays, throughout the research community, huge amounts of data exist, but creating a comprehensive database is challenging. On the other hand, the rise of machine learning models has the potential for significant advances in prediction of complex thermo-hydraulic and heat transfer phenomena – such as prediction of CHF. Usually, machine learning models yield accurate predictions, if applied within their training parameter range. This work aims to overcome the shortcomings of data availability throughout the research community. Ideally, a neural network for CHF prediction should be trained by a research institution, using their own high quality experimental data. This institution could share the machine learning model with the community, together with some information about the training parameter range. Later, another institution could continue to train that model, using their own experimental data and so on. Finally, a comprehensive machine learning model can be established, incorporating the knowledge and experience gained throughout the institutions. For continual learning, problems like “catastrophic forgetting” [12] must be solved. Recently, Song et al. [13] published the Data-Informed Continual Machine Learning (DI-CML) approach, which will be explained in Section 2. The purpose of this paper is to further enhance the DI-CML approach, reducing the amount of information about the

* Corresponding author.
E-mail address: Fabian.Wiltschko@kit.edu (F. Wiltschko).

https://doi.org/10.1016/j.ijheatmasstransfer.2026.128675
Received 31 October 2025; Received in revised form 9 March 2026; Accepted 10 March 2026
Available online 19 March 2026

training data that must be shared and improving the accuracy of the method. Accordingly, the proposed approach is conceptually different from generative replay, as it does not learn or disseminate a data generator; it only filters candidate samples using uncertainty information from the shared predictive model. In this context, "soft data" refers to synthetic input–output pairs generated without access to the original experimental records. These data points are not intended to reproduce the exact experimental distributions, nor are they generated by a learned generative model. Instead, they serve as candidate samples that are subsequently filtered based on the epistemic response of a previously trained predictive model. The objective is therefore not data generation or density estimation, but the identification of the model's statistical validity domain in the input space, enabling privacy-preserving continual training.

2. Data-informed continual machine learning approach

Song et al. [13] introduced the DI-CML approach, on which the present approach is based on. The approach is illustrated in Fig. 1. The idea is that any n^{th} researcher trains a n^{th} machine learning (ML) model, using the available experimental data. The n^{th} researcher shares the n^{th} machine learning model with the community, together with some information on the training database used: the distribution of the input variables $P_x(X)$ and the distribution of the error $P_\varepsilon(X)$ of the ML model. Another $(n+1)^{th}$ researcher may use the information on the database to generate artificial soft data, following the same distribution $P_x(X)$. Using the n^{th} ML model and the soft data input data, the soft output data is generated. This soft data is then combined with the experimental data that is available for the $(n+1)^{th}$ researcher. Now the $(n+1)^{th}$ ML model can be trained, using the combined database. That procedure can be repeated until a comprehensive ML model is established. Song et al. [13] presented two different ways to share the information on the distribution of the input variables $P_x(X)$: the probability density in form of a histogram with bins, or as a probability density distribution (PDF) function in the form of a Γ-function. Song et al. have demonstrated that a $(n+1)^{th}$ ML model can be trained, which is able to perform well on the new $(n+1)^{th}$ data, as well as on the initial n^{th} data. However, the success of the model strongly depends on the generated soft data. If, for instance, the description of the original data by the Γ-function is not accurate enough, some of the soft data points could be outside the statistically learned domain of the ML model. In that case, the soft output would not be reliable. Unfortunately, it is unknown in the current DI-CML approach, whether a given soft data point is reliable or not. Thus, reliability of the comprehensive machine learning model, after it has been trained by several researchers, could be questionable.

While the DI-CML approach represents an important step toward continual learning with limited data sharing, it still requires explicit communication of the input-parameter distributions, either in the form of histograms or fitted probability density functions. Even without sharing individual experimental records, such distributions may reveal sensitive structural information, including preferred operating regions, parameter correlations, and experimental design priorities. In industrial and safety-critical applications, this level of information disclosure can already be considered confidential.

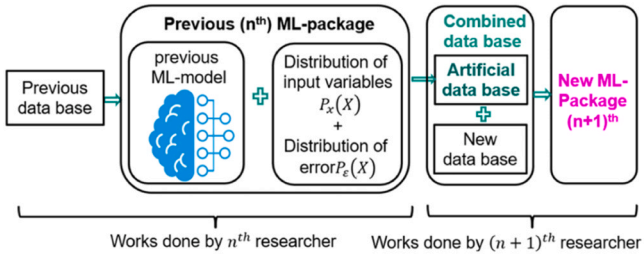

Fig. 1. Scheme of the DI-CML approach.

Therefore, it would be beneficial if the ML model would not only provide a prediction but additionally provide the uncertainty of the prediction. This could be used to sort out soft data points for which the model yields large uncertainty, avoiding incorporating bad information in the new combined database. In the following, the fundamentals and the methodology to filter the soft data by enhancing the DI-CML method are discussed.

3. Uncertainty-aware data-informed continual machine learning approach

As discussed in the previous section, correct generation of soft data is crucial for the DI-CML approach. The present work is motivated by the idea to figure out, which of the generated soft data points are "correct", that means they are like data points used for the training of the original neural network. A neural network generally yields good prediction for data it has seen in the training, or to data that is within the range of training parameters. Therefore, uncertainty of a neural network will also be low, if the given input data is within the range of training parameters. On the other hand, when the model is applied outside statistically well-supported regions of the training domain, epistemic uncertainty typically increases due to reduced familiarity of the learned mapping. Thus, quantification of the model uncertainty can be applied to figure out if a given soft data point comes from the training parameters range. The soft data can be filtered according to that, as explained in Section 3.4.

At the current state-of-the-art, approximate Bayesian uncertainty quantification methods are commonly applied to estimate epistemic uncertainty in neural networks[14]. A description of the general idea of approximate Bayesian posterior sampling methods can be found in MacKay [15] and in Neal [16]. It is also worth mentioning the PhD thesis of Gal [17], providing a comprehensive description of uncertainty quantification methods for neural networks. In the textbook of Bishop [18], all fundamentals are explained. The idea goes back to Bayes' theorem:

$$P(\theta|\mathscr{D}) = \frac{P(\mathscr{D}|\theta)\cdot P(\theta)}{P(\mathscr{D})} \tag{1}$$

with a hypothesis θ and the data $\mathscr{D}$. The belief in hypothesis θ, before seeing the data is called the prior, $P(\theta)$. The probability of the hypothesis after seeing the data, $P(\theta|\mathscr{D})$, is called the posterior. $P(\mathscr{D}\theta)$ is the likelihood, meaning how expected is the data $\mathscr{D}$, when hypothesis θ holds. $P(\mathscr{D})$ is called the evidence, which is the total probability of observing $\mathscr{D}$ under all possible hypotheses.

This theorem can be applied to neural networks. A neural network consists of weights w (and biases b), explaining the data $\mathscr{D} = \{x_i, y_i\}_{i=1}^{N}$ (N is the number of datapoints, x_i are the input features and y_i are the output features). In this sense, the neural network – more precisely the set θ of weights w (and bias b) – is understood as the hypothesis. Putting a prior over the weights $P(w)$, e.g. as a zero-mean Gaussian distribution, encodes the belief in hypothesis before seeing the data [19]. The likelihood (how well weights w and bias b explain the data $\mathscr{D}$) is written for a regression problem, assuming normal distribution of noise, as:

$$P(\mathscr{D}w) = \prod_{i=1}^{N} \mathscr{N}\left(y_i; f_w(X_i), \sigma^2\right) \tag{2}$$

where $\mathscr{N}(y_i; f_w(X_i), \sigma^2)$ is the probability density of observing actual data y_i, given that the neural network predicts $f_w(X_i)$ and the standard deviation of the noise in the data is σ. In this Gaussian regression setting, σ models aleatoric uncertainty, i.e. the inherent data noise [20]. The question of how plausible each setting of weights is, after seeing the data, is answered by the posterior:

$$P(w|\mathscr{D}) = \frac{P(\mathscr{D}|w)P(w)}{P(\mathscr{D})} \tag{3}$$

Predictions for new datapoints (X_i^*, y_i^*) are obtained from the posterior predictive distribution, which integrates over all plausible weights [18]:

$$P(y^*|X^*, \mathscr{D}) = \int P(y^*|X^*, w)P(w\mathscr{D})dw \tag{4}$$

The expected prediction is [18]:

$$\mathbb{E}[y^*|X^*, \mathscr{D}] = \int f_w(X^*)P(w\mathscr{D})dw \tag{5}$$

With a variance [18]:

$$\mathbb{V}[y^*|X^*, \mathscr{D}] = \mathbb{E}_{w\sim P(w\mathscr{D})}[\mathbb{V}(y^*|X^*, w)] + \mathbb{V}_{w\sim P(w\mathscr{D})}[\mathbb{E}(y^*|X^*, w)] \tag{6}$$

where $\mathbb{E}_{w\sim p(w|\mathscr{D})}[\mathbb{V}(y^*|X^*, w)]$ is the aleatoric uncertainty (noise of the data) and $\mathbb{V}_{w\sim p(w|\mathscr{D})}[\mathbb{E}(y^*|X^*, w)]$ is the epistemic uncertainty, i.e., the spread of predictions across plausible models or, in other words, the model uncertainty [20]. This needs to be approximated, since computation of the posterior $P(w|\mathscr{D})$ as well as the integration over the weights is impractical. In the following, different methods for Bayesian approximation are summarized.

In the present work, epistemic uncertainty is not interpreted as a calibrated posterior probability or confidence interval. Instead, it is used as a relative indicator of model familiarity with a given input region. The proposed filtering mechanism only requires that uncertainty increases consistently outside regions supported by the training data. Absolute probabilistic calibration, such as optimization of expected calibration error (ECE) or negative log-likelihood (NLL), is therefore not required for the intended purpose of validity-domain filtering.

3.1. Deep ensemble

A simple and straight forward approach for uncertainty quantification of a neural network is presented by Lakshminarayanan et al. [14]. The idea is simply that there are many different possible combinations of neural networks $y_m = f_{\theta_m}(X)$ (with θ_m as each set of w and b of the m-th network), that explain the data $\mathscr{D}$. Depending on the initialization of w and b before training, another possible neural network will be the result of the training. The predicted output value $\hat{y}$ of a neural network ensemble with M neural networks is:

$$\hat{y}(X) = \frac{1}{M}\sum_{m=1}^{M} f_{\theta_m}(X) \tag{7}$$

In the scope of the present work, aleatoric uncertainty is of less importance, but epistemic uncertainty needs to be evaluated, since this represents how uncertain the model is and thus gives us a clue whether the ensemble has been trained around the corresponding soft data. Epistemic uncertainty is calculated as the standard deviation of the predictions of the different models in the ensemble:

$$\sigma(X) = \sqrt{\frac{1}{M-1}\sum_{m=1}^{M}\left(f_{\theta_m}(X) - \hat{y}(X)\right)^2} \tag{8}$$

In this study, an ensemble of 100 independently initialized neural networks with identical architecture but different random seeds was trained. Epistemic uncertainty is computed as the standard deviation across ensemble predictions for each input sample.

3.2. Monte Carlo dropout

Another method for uncertainty quantification is proposed by Gal and Ghahramani [21]. The idea is the same, that there always exists many possible neural networks that explain given data $\mathscr{D}$. Just the implementation is different, making use of dropout layers [22]. Usually, dropout layers are applied in the training of a neural network, preventing overfitting by randomly dropping out a defined fraction of nodes in the corresponding fully connected layer. Usually, this is only used in training and when the model is used for prediction, the full network will be used. However, the dropout could be applied also for the prediction. Repeating the prediction with active dropout is the same as randomly drawing a network of a larger set of possible networks. Compared to the deep ensemble, the Monte Carlo dropout method requires only training of one neural network, instead of training of M different networks.

The mean prediction of T stochastic forward passes with dropout is calculated as:

$$\hat{y}(X) = \frac{1}{T}\sum_{t=1}^{T} f_{\theta_{\omega_t}}(X) \tag{9}$$

where ω_t is the dropout mask in the t-th forward pass. Again, the epistemic uncertainty is calculated as the standard deviation of the predictions of the T predictions:

$$\sigma(x) = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\left(f_{\theta_{\omega_t}}(X) - \hat{y}(X)\right)^2} \tag{10}$$

A dropout rate of 0.05 was applied in three hidden layers during training. For uncertainty estimation, $T = 100$ stochastic forward passes were performed at inference time with dropout activated, and epistemic uncertainty was calculated as the standard deviation across the sampled predictions.

3.3. Stochastic Weight Averaging-Gaussian

The Stochastic Weight Averaging-Gaussian (SWAG), another approximate Bayesian posterior sampling method is presented by Maddox et al. [23]. Based on stochastic weight averaging (SWA), where during the training of a neural network at given sample checkpoints, the current set of model parameters θ are stored, when the training has already converged and therefore different possible sets of θ are explored. For K collected checkpoints, the model parameters are averaged as:

$$\bar{\theta} = \frac{1}{K}\sum_{k=1}^{K}\theta_k \tag{11}$$

From the differences $(\theta_k - \bar{\theta})$, the sample covariance Σ is approximated, which yields a low-rank plus diagonal approximation of the weight posterior:

$$\Sigma = \frac{1}{K-1}\sum_{k=1}^{K}(\theta_k - \bar{\theta})(\theta_k - \bar{\theta})^\top \tag{12}$$

At model inference, S samples $\theta^{(s)}$ are drawn from a Gaussian distribution $\mathcal{N}(\bar{\theta}, \Sigma)$ and for each sample a prediction $f_{\theta^{(s)}}(x)$ is calculated. The mean prediction of S samples is calculated as:

$$\hat{y}(x) = \frac{1}{S}\sum_{s=1}^{S} f_{\theta^{(s)}}(x) \tag{13}$$

and the epistemic uncertainty is given by the standard deviation of these predictions:

$$\sigma(x) = \sqrt{\frac{1}{S-1}\sum_{s=1}^{S}(f_{\theta^{(s)}}(x) - \hat{y}(x))^2} \tag{14}$$

Same as for the Monte Carlo dropout method, only training of one neural network is required. Additionally, the dropout rate can be calibrated – or not applied – just as an optimal model would require. On the other hand, this approach requires a bit more programming for its implementation.

Weight snapshots were collected starting at epoch 500 and stored every 5 epochs, with a maximum of 30 stored deviations used to construct the low-rank plus diagonal covariance approximation. Posterior sampling was performed using S = 100 sampled weight realizations to estimate epistemic uncertainty.

All uncertainty methods use the same base neural network architecture (three hidden layers with 200 neurons each, ReLU activation), Adam optimizer (learning rate 10^{-3}), batch size 50, and early stopping

with patience 100.

3.4. Iterative generation of high-quality soft data

The data n was used to train a machine learning model n. Now, artificial soft input data shall be generated and model n shall be used to predict soft output values. This soft data can be combined with new data $n+1$, which is available for another researcher, who can train machine learning model $n+1$ in the next step. To generate the soft data, in contrast to the DI-CML approach, for the present approach it is only necessary to know the range of each of the input features $X\{x_1, x_2, \ldots, x_n\}$ of data n. According to the minimum and maximum value of each of the input features, random data with a uniform distribution is generated, with same number as the number Z of datapoints in the database n. approximate Bayesian uncertainty estimation is used to determine the uncertainty $\sigma_i(X_i)$ for any of the soft data points. Now, for each of the soft data points can be judged, whether the uncertainty is acceptable or not – if not, the corresponding input lies in a region where the model is statistically unfamiliar. This situation may arise from extrapolation beyond the training range or from sparsely populated regions within the nominal parameter space, and in either case the predicted soft output of model n cannot be considered reliable. This is done according to a threshold value τ, so that only soft data which fulfills the following criteria is accepted:

$$\frac{\sigma_i(X_i)}{\widehat{y}_i(X_i)} < \tau \tag{15}$$

The value of the threshold value τ depends on the accuracy of model n, predicting data n. Obviously, one cannot expect the prediction of model n for artificial soft data to be more accurate than the original data, but it is an appropriate criterion to judge which soft data is similar to original training data, if the uncertainty of model n for an artificial data point is around a similar level as it is for original data. Soft data points which do not satisfy Eq. (15) are discarded, while the others are kept as *good* soft data points. In the next step, new soft data points are generated randomly, so that the total number of soft data points is again Z. Approximate Bayesian uncertainty estimation is applied, criterion of Eq. (15) is applied and so on. This is repeated until Z good soft data points are generated. The procedure is visualized in Fig. 2.

4. Evaluation of the uncertainty-aware data-informed continual machine learning approach

To evaluate the new approach, the database provided within the OECD/ NEA benchmark exercise [7] is used, which has also been used by Song et al. [13]. The original benchmark database contains 24,579 experimental data points. For consistency in the continual learning setup and to obtain equal-sized dataset partitions, one randomly selected data point was removed prior to splitting. All experiments reported in this study are therefore based on 24,578 samples. The covered parameter range is given in Table 1.

The full database is denoted as $\mathscr{D}_0$. To distinguish the two sampling configurations, superscripts are used such that $\mathscr{D}_i^{G1S}$ and $\mathscr{D}_i^{G2S}$ ($i \in \{1,2\}$) denote the respective dataset partitions for the sampling options G1S and G2S. For the evaluation of the new Uncertainty-Aware Data-Informed Continual Machine Learning (UA-DI-CML) approach, $\mathscr{D}_0$ is split into $\mathscr{D}_1$ and $\mathscr{D}_2$, testing two different sample options, as described in Song et al. [13]. The database $\mathscr{D}_0$ is split by the median value (1585 kg/m²s) of the mass flux.

G into two groups, where group 1 contains data with mass flux larger than the median value, and group 2 contains data with mass flux smaller than the median value. For the first sampling option, called G1S, the dataset partitions are denoted as $\mathscr{D}_1^{G1S}$ and $\mathscr{D}_2^{G1S}$. 80% of the data in group 1% and 20% of the data in group 2 are randomly selected and assigned to $\mathscr{D}_1^{G1S}$. The remaining data is assigned to $\mathscr{D}_2^{G1S}$. For the second sampling option, called G2S, the dataset partitions are denoted as $\mathscr{D}_1^{G2S}$ and $\mathscr{D}_2^{G2S}$. Group 1 is assigned to $\mathscr{D}_1^{G2S}$ and group 2 is assigned to $\mathscr{D}_2^{G2S}$. Thus, G1S represents a more realistic scenario, where different institutions have experimental data covering different ranges, but also overlapping with each other. G2S is a reference case, with a very strict split of the data at the median value, with no overlapping. It is also important to note that each database needs to be split into train data

Table 1
Parameter range of CHF database.

Parameter	Range
Pressure, $\boldsymbol{p}[\boldsymbol{MPa}]$	0.1 – 20.0
Mass flux, $\boldsymbol{G}[\boldsymbol{kg/m^2s}]$	8.2 – 7964.0
Local steam quality, $\boldsymbol{x_{out}}$ [–]	-0.5 – 1.0
Tube diameter, $\boldsymbol{D}[\boldsymbol{mm}]$	2.0 – 16.0

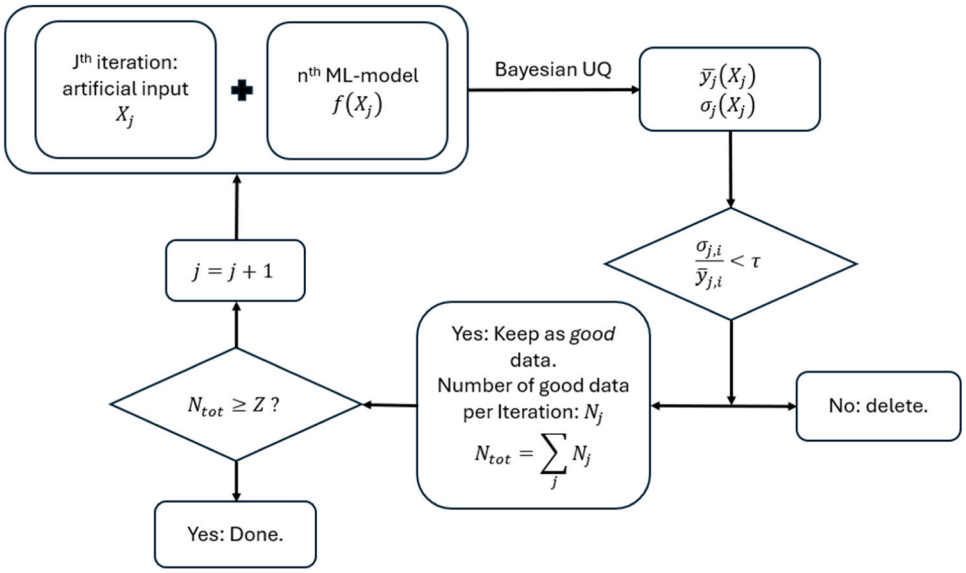

Fig. 2. Uncertainty-aware soft data filtering approach.

(containing 80% of the data) and test data (containing 20% of the data). The data is split in that way, that $\mathscr{D}_{0,train} = \mathscr{D}_{1,train} + \mathscr{D}_{2,train}$, and respectively of the test data, to strictly separate train data from test data throughout the whole evaluation. Table 2 summarizes sampling options.

The reference model MLM0 is trained using $\mathscr{D}_0$. Further, a nth model is called MLM1 and trained with $\mathscr{D}_1$. MLM1 is also a reference model. For sake of clarity, only the standard neural network (exactly as in Song et al. [13]) is used here, no matter which of the uncertainty quantification (UQ) methods is applied, so that only two reference models MLM1-G1S and MLM1-G2S according to the sampling have to be considered. The (n + 1)th model is called MLM2, and according to the applied UQ method, model IDs are summarized in Table 3.

The accuracy of MLM0 and the different MLM1 with respect to the different databases are summarized in Table 4, where the mean error μ and the standard deviation of errors σ_e are defined as:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\frac{100 * (\hat{y}_i - y_i)}{y_i} \tag{16}$$

$$\sigma_e = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{100 * (\hat{y}_i - y_i)}{y_i} - \mu\right)^2} \tag{17}$$

Note that the presented values are slightly different from the values on Song et al. [13], who presented the accuracy against the full database, while in the present work the accuracy versus the test data is shown.

Especially for the sampling option G2S, where MLM1 is evaluated on input combinations in $\mathscr{D}_2^{G2S}$ that are not represented in its training subset $\mathscr{D}_1^{G2S}$, thereby constituting an extrapolative evaluation setting, larger prediction deviations are observed. For the sampling option G1S, MLM1 yields better results, since it has been seeing data in the same range. Nevertheless, the performance of MLM1 is always worse on $\mathscr{D}_0$ and $\mathscr{D}_2$, compared to MLM0. Goal of this approach is to train a model MLM2, which yields accurate results on all the three databases – without having access to the original $\mathscr{D}_1$.

4.1. Estimation of the threshold uncertainty τ

Now, artificial soft input data X_{soft} must be generated. MLM1 needs to be used to create soft output values y_{soft}. Application of approximate Bayesian posterior sampling method shall be used to filter out soft data with large model uncertainty, to keep only reliable data points for training of MLM2. As described in Section 3.4, a threshold value τ needs to be established, determining which level of uncertainty is still acceptable. The accuracy of MLM1 against $\mathscr{D}_1$ gives us an idea of how high τ can be minimally – obviously, cannot be expected to be less uncertain about artificial input data X_{soft}, than is has been in its original training data. Fig. 3 shows the predicted uncertainties σ_i scaled by the actual predicted values $\hat{y}_i$, for the deep ensemble (DE) method, for sampling option G1S, over the predicted CHF values ($\hat{y}_i$). The distribution looks similar for both sampling methods and all the UQ methods, so that presentation of one Figure as an example is sufficient. For small CHF, the uncertainty of MLM1 increases, while the uncertainty remains almost constant for CHF values of larger than around 4000 kW/m^2. The data on the Figure is split into groups with equal spacing in the CHF and the 90th percentile is marked with a red cross (so that 90% of datapoints have smaller uncertainty than that value). Then, a function of the following form is fitted to these red data points:

Table 2
Identification of the cases considered in this study.

	Original database	1st sampling option	2nd sampling option
Case-ID	O	G1S	G2S

Table 3
Identification of the different (n + 1)th models.

UQ Method for filtering	Deep Ensemble	Monte Carlo Dropout	SWAG
Model-ID	MLM2-DE	MLM2-MCD	MLM2-SWAG

Table 4
Accuracy of the reference ML models against the unseen test data.

	MLM0		MLM1-G1S		MLM1-G2S	
	μ	σ_e	μ	σ_e	μ	σ_e
$\mathscr{D}_0$	3.47	24.32	6.29	31.53	-24.74	113.56
$\mathscr{D}_1$	2.77	20.5	5.58	25.19	0.23	13.67
$\mathscr{D}_2$	4.16	27.16	6.99	36.78	-49.71	156.08

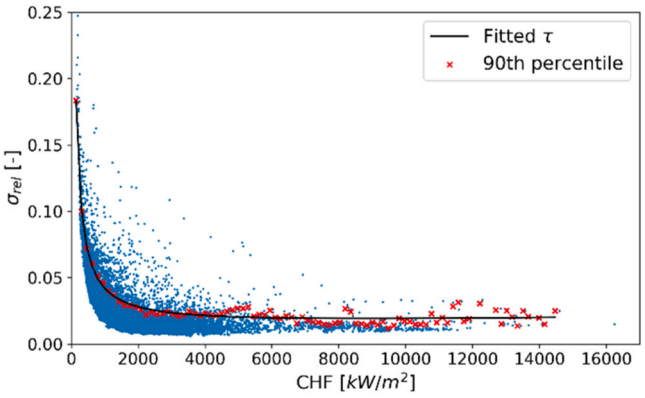

Fig. 3. Relative model uncertainty and fitted threshold function.

$$\tau(\hat{y}_i) = \frac{a}{(\hat{y}_i - b)} + c\hat{y}_i + d \tag{18}$$

This is simply the best representation of the distribution of the uncertainty σ_{rel}, and could be a constant, linear, exponential or any other function, when the method would be applied to another database or another physical problem. The threshold function $\tau(\hat{y}_i)$ must be fitted for each of the UQ methods, as well as for both sampling options. An overview is presented in Table 5. Compared to the information on the histograms or the Γ-PDF functions, as in the DI-CML approach, for the present approach only the threshold function $\tau(\hat{y}_i)$, which means a description on how accurate one can expect the MLM1 model to be, must be shared.

The use of relative uncertainty, defined as the ratio of epistemic uncertainty to the predicted value, may lead to increased values at very small predicted CHF levels. This behavior is not an artifact of the proposed method but reflects the observed performance characteristics of the trained model. Importantly, the threshold function $\tau(\hat{y}_i)$ is fitted directly to the uncertainty distribution obtained on the original training data, such that filtering is performed relative to the model's demonstrated reliability rather than based on absolute uncertainty magnitudes. Consequently, regions exhibiting low relative uncertainty are

Table 5
Fitted parameters of threshold function $\tau(\hat{y}_i)$.

UQ and Sampling	a	b	c	d
DE – G1S	29.95	-28.39	3.77e-7	1.25e-2
DE – G2S	16.61	40.94	3.5e-8	1.82e-2
MCD – G1S	91.4	-9.88	2.16e-6	1.14e-2
MCD – G2S	88.33	-14.04	2.85e-6	3.36e-3
SWAG – G1S	49.75	1.9	1.53e-6	1.24e-2
SWAG – G2S	10.49	74.95	4.44e-7	3.578e-3

interpreted as statistically familiar to the model, even if sparsely populated, while regions with elevated relative uncertainty are conservatively excluded.

4.2. Exploration of original training data distribution

Initially, soft data X_{soft} is generated randomly, with uniform distribution, as shown exemplarily of G in Fig. 4. In the present study, input data are tube diameter D, pressure p, mass flux G and local steam quality x_{out}. The generated soft data for all the 4 variables is within the range of $\mathscr{D}_1$ and has the same number of points as in $\mathscr{D}_1$.

Now, applying DE, MCD and SWAG, the predictions $\hat{y}_{i,soft}(X_{i,soft})$ and the corresponding model uncertainties $\sigma_{i,soft}(X_{i,soft})$ are calculated for all the N datapoints. Soft data points which do not satisfy:

$$\sigma_{rel,i,soft}(X_{i,\ soft}) = \frac{\sigma_{i,\ soft}(X_{i,\ soft})}{\hat{y}_{i,\ soft}(X_{i,\ soft})} < \ \tau(\hat{y}_i) \tag{19}$$

are discarded. Then, new random soft data are generated uniformly, so that the total number is again the same as in $\mathscr{D}_1$. This process is repeated until only soft data remain, satisfying the criterion (19). The distribution of the final soft data, compared to the original data $\mathscr{D}_1$ can be seen in Fig. 5 for all the input features, exemplarily for DE method with sampling G2S. The statistical validity domain of the trained model is revealed to some extent. This happens since the machine learning model yields very certain predictions when applied to a region of input features already experienced during the training. In contrast to the method proposed by Song et al. [13], in the present method, also the combination of the different input variables should be implicitly considered. The training data distribution is also discovered by the other UQ methods, for the sampling method G1S and for the sampling method G2S, yielding only slightly different distributions.

It is important to emphasize that the distributions shown in Fig. 5 do not represent a reconstruction of the original experimental database. The displayed soft data remain synthetic and are generated independently of the confidential measurements. The filtering procedure only identifies regions in the input space where the predictive model yields consistently low epistemic uncertainty, which reflects the model's statistical familiarity rather than the exact sampling density, discrete experimental conditions, or proprietary measurement patterns. Exact reproduction of the original input distributions is neither intended nor desirable, as it would contradict the privacy objective of the proposed approach.

4.3. Results

Discovering the distribution of the training data is already promising. Now, actual accuracy of MLM2 with the different UQ methods applied in soft data generation process is compared to each other and are also compared to results of the DI-CML method. Theaccuracy of different models against different data is shown in Table 6 for G1S and in Table 7 for G2S, again with the unseen test data. The results of DI-CML method [13] (the variant with uniform distribution of p, D and x_{out}) are included for comparison.

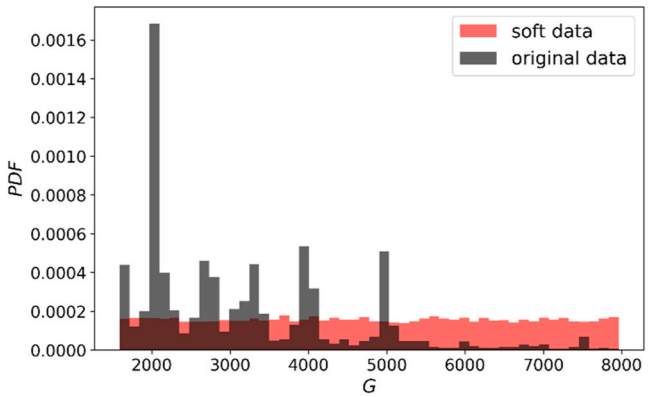

Fig. 4. Initial distribution of soft data G (G2S).

On first glance, no substantial performance differences are observed among the five MLM2 variants, independent of whether sampling method G1S or G2S is used. Importantly, this comparable predictive performance is achieved while significantly reducing the amount of information that must be shared between institutions, which constitutes the primary objective of the proposed UA-DI-CML framework. The proposed UA-DI-CML framework reduces the amount of shared information compared to DI-CML while maintaining comparable predictive performance. The new approach presented in this study requires only a description of the expected accuracy of the model MLM1. Further, in the present approach it is guaranteed that only soft data points, for which MLM1 yields predictions with low uncertainty are used. In the DI-CML approach, also soft data points with a large uncertainty could be eventually used, encoding bad information into MLM2. This would result in a less reliable comprehensive machine learning model, after several researchers have applied the approach, using their data. Even though the differences are small, the Monte Carlo Dropout method most often yields the slightly more accurate MLM2. With simple implementation and –low computational effort (especially compared to the DE method), the MCD method is a promising approach for continual learning.

This study is evaluated on a single CHF benchmark dataset and two predefined sampling scenarios, so broader validation on additional thermo-hydraulic datasets, for different physical phenomena and transfer settings as well is still needed. In addition, the uncertainty estimates from Deep Ensembles, Monte Carlo Dropout, and SWAG are used here as relative indicators of model familiarity rather than as fully calibrated probabilistic confidence measures. The proposed soft-data filtering strategy identifies a statistical validity domain, but its performance may depend on the chosen uncertainty threshold. Finally, although UA-DI-CML reduces information sharing compared with DI-CML, it does not provide a formal privacy guarantee in the cryptographic or differential-privacy sense.

5. Conclusion

The present study deals with an improvement of the recently published DI-CML approach. By evaluating the uncertainty of a machine learning model MLM1 on artificially generated soft data, it shall be ensured that only reliable data is used for the training of further models MLM2. This is another step towards a comprehensive machine learning model, which incorporates knowledge available in different institutions throughout the community. In the present study, three different approximate Bayesian posterior sampling methods have been applied to filter artificial soft data. All the three methods are reliable, the differences in prediction accuracy of the new machine learning models MLM2 are marginal. However, with respect to computational and implementation effort, the Monte Carlo Dropout method provides a favorable balance between efficiency and sufficiently informative epistemic uncertainty for the purpose of validity-domain filtering. A clear advantage compared to the DI-CML method is that besides the range, no information on the database $\mathscr{D}_1$ is required. Only the uncertainty of a given MLM1 must be shared with the community, which in turn also documents the reliability of the used data. Further it is observed that the statistical validity domain of MLM1 can be approximated in terms of model familiarity, without reconstructing or disclosing the original confidential experimental distribution., by iteratively generating new random soft data and discarding those which yield predictions with large uncertainties.

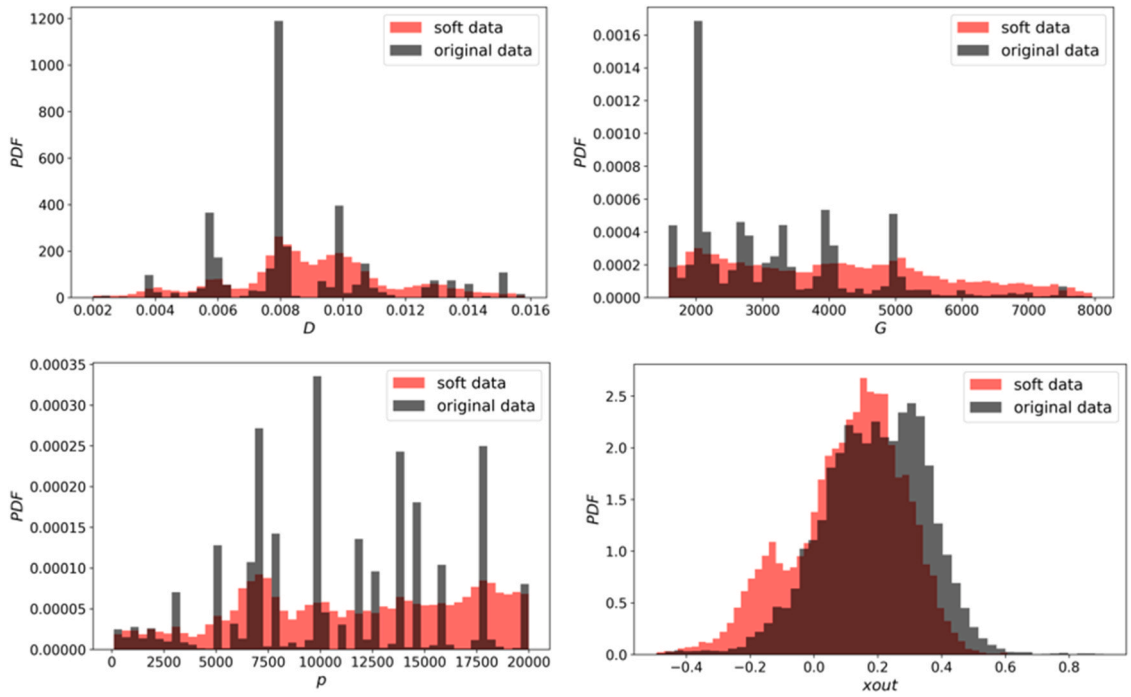

Fig. 5. Final soft data distribution (G2S) with low uncertainty according to deep ensemble uncertainty quantification.

Table 6
Accuracy of different $(n+1)^{th}$ machine learning models for sampling G1S.

Model	$\mathscr{D}_0$		$\mathscr{D}_1$		$\mathscr{D}_2$	
	μ	σ_e	μ	σ_e	μ	σ_e
MLM2-b	4.0	25.99	3.25	22.67	4.74	28.91
MLM2-PDF	4.31	25.74	3.77	22.64	4.84	28.51
MLM2-DE	4.27	27.21	3.35	25.57	5.19	28.74
MLM2-MCD	4.84	25.89	3.84	22.55	5.84	28.81
MLM2-SWAG	4.97	26.12	4.09	22.75	5.86	29.08

Table 7
Accuracy of different $(n+1)^{th}$ machine learning models for sampling G2S.

Model	$\mathscr{D}_0$		$\mathscr{D}_1$		$\mathscr{D}_2$	
	μ	σ_e	μ	σ_e	μ	σ_e
MLM2-b	4.69	27.39	3.11	22.69	6.27	31.33
MLM2-PDF	5.58	28.99	3.83	22.79	7.34	33.99
MLM2-DE	4.72	27.25	3.06	21.82	6.38	31.69
MLM2-MCD	3.72	26.74	1.94	22.01	5.51	30.65
MLM2-SWAG	4.78	27.18	4.13	22.51	5.42	31.14

CRediT authorship contribution statement

Aurelian F. Badea: Writing – review & editing. **Xu Cheng:** Writing – review & editing, Methodology. **Zhichao Zhang:** Validation, Software, Investigation, Conceptualization. **Fabian Wiltschko:** Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Fabian Wiltschko reports was provided by Karlsruher Institut fÜr Technologie (KIT). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Bruder, G. Bloch, T. Sattelmayer, Critical heat flux in flow boiling - review of the current understanding and experimental approaches, Heat Transf. Eng. 38 (3) (2017) 347–360.
[2] X. Cheng, U. MÜller, Review on critical heat flux in water cooled reactors, Forschungszentrum Karlsruhe GmbH, Karlsruhe, Germany (2003).
[3] D.D. Hall, I. Mudawar, Critical heat flux (CHF) for water flow in tubes - I. Compilation and assessment of world CHF data, Int. J. Heat Mass Transf. 43 (2000) 2573–2604.
[4] D.D. Hall, I. Mudawar, Critical heat flux (CHF) for water flow in tubes - II. Subcooled CHF correlations, Int. J. Heat Mass Transf. 43 (2000) 2605–2640.
[5] D.C. Groeneveld, J.Q. Shan, A.Z. Vasic, L.K.H. Leung, A. Durmayaz, J. Yang, S. C. Cheng, A. Tanase, The 2006 CHF look-up table, Nuclear Eng. Des. 237 (2007) 1909–1922.
[6] Y. Liu, W. Liu, L. Gu, J. Shan, L. Zhang, X. Su, Existing DNB-type CHF mechanistic models and relations with visualized experiments in forced convective flow boiling: a review, Progress Nuclear Energy 148 (2022).
[7] J.-M. Le Corre, G. Delipei, X. Wu, X. Zhao, Benchmark on artificial intelligence and machine learning for scientific computing in nuclear engineering. Phase 1: Critical Heat Flux Exercise Specifications, OECD Publishing, NEA Working Papers,, Paris, France, 2024.
[8] I. Ahmed, I. Gatti, E. Zio, Optimized ensemble of neural networks for the prediction of critical heat flux, Nuclear Eng. Des. 439 (2025).
[9] S.K. Moon, S.H. Chang, Classification and prediction of the critical heat flux using fuzzy theory and artificial neural networks, Nuclear Eng. Des. 150 (1994) 151–161.
[10] S.K. Moon, W.-P. Baek, S.H. Chang, "Parametric trends analysis of the critical heat flux based on artificial neural networks, Nuclear Eng. Des. 163 (1996) 29–49.
[11] W. Zhou, S. Miwa, H. Wang, K. Okamoto, Assessment of the state-of-the-art AI methods for critical heat flux prediction, Int. Commun. Heat Mass Transf. 158 (2024).
[12] K. James, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rasu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell, Overcoming catastrophic forgetting in neural networks, in: Proceedings of the National Academy of Sciences, 114, Imperial College London, United Kingdom,, March 2017.
[13] M. Song, F. Wiltschko, X. Liu, A.F. Badea, X. Cheng, Development and assessment of the data-informed continuous machine learning approach, Int. J. Heat Mass Transf. (2025).
[14] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, Proceed. 31st Int. Conf. Neural Inform. Process. (2017). December 4 - 9.
[15] D.J.C. MacKay, A practical Bayesian Framework for Backpropagation Networks, Neural Computation 4 (1992) 448–472.
[16] R.M. Neal, Bayesian learning for neural networks, University of Toronto,, Toronto, Canada, 1995.
[17] Y. Gal, Uncertainty in deep learning, Depertment of Engineering, University of Cambridge,, Cambridge, Great Britain, 2016.

[18] C. Bishop, Pattern recognition and machine learning, Springer,, New York, USA, 2006.
[19] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural networks, Proceed. 32nd Int. Conf. Mach. Learn. (2015). July 6-11.
[20] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision? Proceed. 31st Conf. Neural Inform. Preces. Syst. (NIPS) (2017). December 4-9.
[21] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: representing model uncertainty in deep learning, Proceed. 33rd Int. Conf. Mach. Learn. 48 (2016). June 19 - 24.
[22] N. Srivastava, G. Hinton, A. Krizhevsky, S. Ilya, S. Ruslan, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[23] W.J. Maddox, T. Garipov, P. Izmailov, D. Vetrov, A.G. Wilson, A simple baseline for bayesian uncertainty in deep learning, Proceed. 33rd Annu. Conf. Neural Inform. Proces. Syst. (2019). December 8-14.