INL/JOU-22-66939-Revision-0

Data coverage assessment on neural network based digital twins for autonomous control system

March 2023

Longcong Wang, Nam Dinh, Linyu Lin

Changing the World's Energy Future

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

INL/JOU-22-66939-Revision-0

Data coverage assessment on neural network based digital twins for autonomous control system

Longcong Wang, Nam Dinh, Linyu Lin

March 2023

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AR0000976

Data Coverage Assessment on Neural Network Based Digital Twins for Autonomous Control System

Longcong Wang[a,*], Linyu Lin[b], Nam Dinh[a]

[a]*Department of Nuclear Engineering, North Carolina State University, 2500 Stinson Dr, 27695, Raleigh, NC, United States*

[b]*Instrumentation, Controls, and Data Science, Idaho National Laboratory, 83415, Idaho Falls, ID, United States*

Abstract

In a recently developed Nearly Autonomous Management and Control (NAMAC) system, neural networks (NNs) are used to develop digital twins for diagnosis (DT-Ds). However, NNs are not usually considered extrapolation models and may result in large errors if they are applied to unseen data outside the training data (uncovered). In this study, we propose a data coverage assessment (DCA) to determine if the NN-based DT-Ds are extrapolated based on their epistemic uncertainty. The uncertainty quantification algorithms and uncertainty thresholds are selected based on the confusion matrix of classifying evaluation data into covered or uncovered data. To demonstrate the adaptability of the proposed framework, we applied it to a basic feedforward neural network and a more advanced recurrent neural network based on a more nonlinear database. Case studies show that the proposed framework can distinguish unseen data for both basic and advanced applications with proper uncertainty quantification algorithms and thresholds.

Keywords: Machine Learning, Neural Network, Digital Twin, Data Coverage Assessment

Acronyms

AE	Absolute Error
AUROC	Area under the ROC curve
DCA	Data Coverage Assessment
DE	Deep Ensembles

*Corresponding author

Email addresses: lwang59@ncsu.edu (Longcong Wang), linyu.lin@inl.gov (Linyu Lin), ntdinh@ncsu.edu (Nam Dinh)

DT	Digital Twin
DT-D	Digital Twin for Diagnosis
DT-P	Digital Twin for Prognosis
EBR-II	Experimental Breeder Reactor - II
FN	False negative
FNN	Feedforward Neural Network
FP	False positive
FPR	False positive rate
LOFA	Loss of Flow Accident
MCMC	Markov-Chain Monte Carlo
ML	Machine Learning
MSE	Mean Squared Error
NAMAC	Nearly Autonomous Management and Control
NLL	Negative log likelihood
NN	Neural Network
NPV	Negative predictive value
OoD	Out-of-Distribution
PPV	Positive predictive value
QoI	Quantity of Interest
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SGLD	Stochastic Gradient Langevin Dynamics
SSF	Safety Significant Factors
TH	Thermal-Hydraulics
TN	True negative
TNR	True negative rate
TP	True positive

TPR	True positive rate
UQ	Uncertainty Quantification

Symbols

θ	Model parameter
$\mathcal{D}$	Training data set
$\mathcal{D}_x$	Training input features
$\mathcal{D}_y$	Training output features
x	Input features
y	Output features
$p(\theta\|\mathcal{D})$	Posterior
$p(\theta)$	Prior
M	Number of NN in ensembles
σ	Standard deviation

1. Introduction

Developing autonomous control systems for nuclear reactors is a popular field of research [1–6]. A Nearly Autonomous Management and Control (NAMAC) system has been recently developed for advanced reactors [7] to support reactor operation by providing recommendations to reactor operators. It integrates knowledge extracted from a scenario-based model of the plant, plant operating procedures, technical specifications, and real-time measurements, based on which recommendations are derived.

The NAMAC system utilizes digital twins (DTs) to support these knowledge extraction and storage processes. A DT in the NAMAC system is *"a digital representation of a physical object or system that relies on real-time and past-history data to evaluate its complete states, to predict future transients, and to recommend appropriate control actions"* [7]. DTs in the current NAMAC demonstration are simplified as surrogates of reactor simulators. They take advantage of the impressive power of machine-learning (ML) algorithms in capturing highly nonlinear relations and predicting reactor behaviors under nominal and accidental conditions.

In current demonstrations of the NAMAC system, a neural network (NN) is used to develop two major DTs: DT for diagnosis (DT-D) and DT for prognosis (DT-P). DT-D recovers the complete reactor states. It infers the unobservable safety-significant factors (SSFs) based on observable reactor states. The estimated SSFs will be used by the DT-P to forecast future reactor states based on which the NAMAC system will recommend mitigation strategies to the operators under accident conditions. In addition to a feedforward neural network (FNN), a recurrent neural network (RNN), which is better at capturing the time-dependent system behaviors, has been used to develop the DT-D and DT-P [8–10].

A major issue of using NNs is that they are not considered extrapolation models. NNs are trained so that they can approximate the relationship between input and output features based on the training data. However, when a trained NN is applied to targets that are not included in the training data, the unknown relations stored in the new data cannot always be well captured. NN-based classifiers can overconfidently predict unrecognizable images with known labels [11, 12] and can fail to classify natural adversarial examples [13]. Similarly, when an FNN is applied to regression tasks, it may not extrapolate well in most nonlinear tasks [14]. Also, the FNN structure determines how it extrapolates [14]. Therefore, when an NN accurately fitting the training data is extrapolated, the prediction uncertainty and potential prediction error can become larger.

For safety-critical systems like nuclear reactors, DTs and DT-supported control system uncertainty need careful evaluation such that concerns about the risk and reliability of nuclear power plants can be resolved. Since NNs could result in a large uncertainty when they are extrapolated to unseen data points, it is important to determine if the NN-based DTs are applied to new data points within the extrapolation capability (covered) or without the extrapolation capability (uncovered). In this study, we propose a data coverage assessment (DCA) framework for defining the coverage conditions given a trained NN and its uncertainty information and testing if the definitions are valid given various target data points.

In this paper, Section 2 reviews relevant work and proposes that uncertainty-based method can be used for the DCA of a NAMAC DT-D. Section 3 introduces the techniques used to assess the data coverage. Section 4 discusses the proposed workflow for developing and optimizing the DCA. To demonstrate the proposed workflow, we formulate case studies and discuss the results in

Section 5.

2. Data Coverage Assessment

As mentioned, NNs are vulnerable to model extrapolation. Many fields, such as image classification, have already recognized similar issues and widely studied the detection of anomalous and out-of-distribution (OoD) samples [15, 16]. Numerous techniques have been developed, which can be supervised, unsupervised, or semisupervised based on whether the labels are available. Anomaly detection usually relies on some specific metric, which is usually the major contribution of a new technique. Based on the type of metric, those techniques can be grouped as probability-based, distance-based, uncertainty-based, and projection-based [15]. Most techniques are developed and tested based on image classification, so not all of them are suitable for nuclear engineering applications.

As NNs are widely used to solve engineering problems, researchers have noticed this issue and tried to address it. Bao et al. [17] assessed the data coverage by evaluating the coverage of physics involved in the processes. The physics is categorized in two scales: the global physics representing the macroscopic or global states, such as the dimension, boundary condition, and non-dimensional parameters representing the underlying physics, and local physics indicating microscopic or local states. The global and local physics will be evaluated separately, since local physics may not change while the global physics are extrapolated, and vice versa. In the NAMAC demonstration, the DT-D data coverage is assessed by checking the mutual information from training and testing scenarios [7]. The distributions of the quantity of interest (QoI) in training and test data are compared by calculating the symmetric Kullback-Leibler (KL) divergence. The prediction error and KL divergence are strongly correlated based on the calculated Pearson correlation coefficient. Gurgen and Dinh [18] used dynamic time warping (DTW) to measure the dissimilarity between the training and testing databases to evaluate whether the training data base covers the testing data base for physics-guided RNNs. The minimum DTW distance between a single testing time series and a training data base containing multiple time series characterizes the dissimilarity between the testing and training data bases. The prediction error increases as the dissimilarity characterized by the minimum DTW distance becomes larger. When NNs are used for flooding modeling, the

predictive uncertainty can detect data that the network does not know [19], and it is shown that the mean prediction interval width is larger for the unseen data.

Uncertainty-based methods have been successfully used for similar tasks, including some engineering applications [19–24]. By definition, the NN epistemic uncertainty can be caused by lack of data, so the epistemic uncertainty will become larger in the space not covered by the training data [21, 25], shown by the demonstrations of many uncertainty quantification (UQ) techniques developed for NNs [26, 27]. Therefore, we selected an uncertainty estimation to assess the data coverage of a single DT.

3. Technical Basis

3.1. Neural networks

NNs are generally a machine replicating the human brain to perform a particular task [28]. NNs connect numerous simple cells, which are called "neurons," and extract knowledge from the given data and environment through a process called "learning." The learned knowledge is stored in the connections between neurons, which are usually characterized by the weights and bias of the neurons.

An FNN is a classic NN type usually made of an input layer, an output layer, and several hidden layers. Each hidden layer may include hidden neurons connected sequentially. With input $\mathbf{x}$, a n-layer FNN will compute the output $\hat{\mathbf{y}}$ by:

$$\hat{\mathbf{y}} = f\left(\mathbf{w}_n f\left(\mathbf{w}_{n-1} \cdots f\left(\mathbf{w}_1 \mathbf{x} + \mathbf{b}_1\right) \cdots + \mathbf{b}_{n-1}\right) + \mathbf{b}_n\right) \tag{1}$$

where $\mathbf{w}_i$ is the weight matrix of the ith layer, $\mathbf{b}_i$ is the bias vector of the ith layer, and $f(\cdot)$ is the activation function.

Traditional NNs are not good at storing information at previous sequence steps, so they may not be a good fit to deal with time-dependent problems, such as predicting reactor transient behaviors. Therefore, RNNs, which take previous outputs as inputs, may fit better. However, classic RNNs can have trouble capturing long-term dependencies. Therefore, long short-term memory (LSTM) is proposed to address this issue [29]. A standard LSTM cell is shown in Figure 1, and includes

three gates: input, output, and forget. With the given input sequence, the layer state at time t can be computed by [30]:

$$f_t = \sigma\left(\mathbf{w}_{if}x_t + \mathbf{w}_{hf}h_{t-1} + \mathbf{b}_f\right) \tag{2}$$

$$i_t = \sigma\left(\mathbf{w}_{ii}x_t + \mathbf{w}_{hi}h_{t-1} + \mathbf{b}_i\right) \tag{3}$$

$$g_t = \tanh\left(\mathbf{w}_{ig}x_t + \mathbf{w}_{hg}h_{t-1} + \mathbf{b}_g\right) \tag{4}$$

$$o_t = \sigma\left(\mathbf{w}_{io}x_t + \mathbf{w}_{ho}h_{t-1} + \mathbf{b}_o\right) \tag{5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{6}$$

$$h_t = o_t \odot \tanh(c_t) \tag{7}$$

where h_t is the hidden state at time t, c_t is the cell state at time t, g_t represents the candidate hidden state, and i_t, f_t, and o_t represent input, forget, and output gates, respectively. The learnable parameters of an LSTM cell include the weights and bias of the gates.

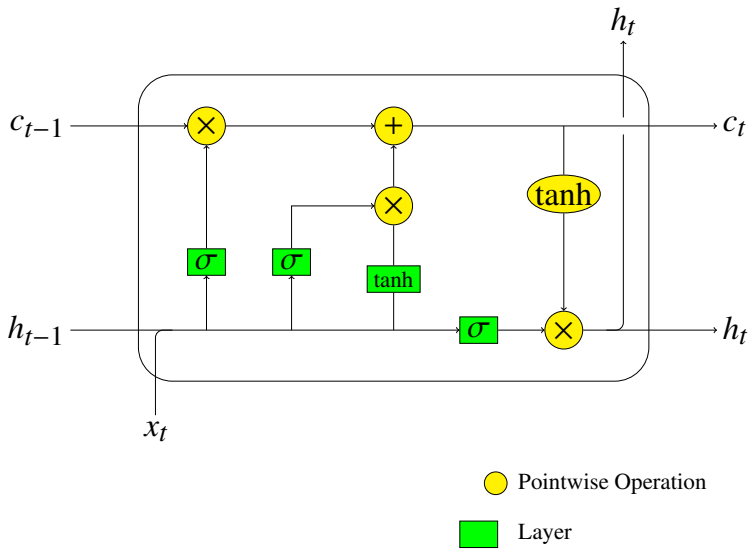


Figure 1: LSTM cell

3.2. Decomposition of uncertainty involved in neural networks

The uncertainties involved in NNs are usually categorized into: aleatoric uncertainty and epistemic uncertainty [31]. The definitions of aleatoric and epistemic uncertainties used in the NN UQ

are slightly different from those used in the UQ of traditional computational models.

The aleatoric uncertainty is caused by the inherent noise of training data [31]. For instance, the data collected from sensor readings may deviate from the true values. The aleatoric uncertainty cannot be reduced by collecting more training data, but improving the data quality may help minimize the aleatoric uncertainty. This type of uncertainty can be captured by NNs with built-in variance, which treats the NN predictions as probability distributions. Additionally, the probability distributions considered in NNs with built-in variance should be distinguished from the posterior distributions obtained using Bayesian inference.

The epistemic uncertainty is related to the lack of knowledge. Untrained NNs can make numerous possible predictions by assigning different values to model parameters [31]. If NNs are trained to fit the given training data, NN predictions are constrained. Predictions will converge to the truth where training data are provided. However, NNs will have more freedom to make predictions when lacking training data. Epistemic uncertainty can be used to describe how many possible predictions trained NNs may make. The epistemic uncertainty can be reduced by introducing new training data, which contains useful knowledge. In addition, the epistemic uncertainty will be impacted by the model structure, and the predictions made by larger networks will be more uncertain [32, 33]. It may be difficult to isolate the effect of each type of uncertainty from the total epistemic uncertainty.

3.3. Neural networks with built-in variance

The aleatoric uncertainty can be captured by explicitly predicting the variance of the output features [19, 34], as shown in Figure 2. When ordinary NNs are used for regression problems, they only predict the mean values, and the parameters are usually optimized by minimizing the mean squared error (MSE) between the predictions and the truth. NNs with built-in variance double the size of outputs, which consist of the mean μ and the variance σ^2 of the QoIs. To increase the numerical stability, the variance can be modeled as:

$$\sigma = \exp(\rho) \tag{8}$$

or

$$\sigma = \log(1 + \exp(\rho)) \tag{9}$$

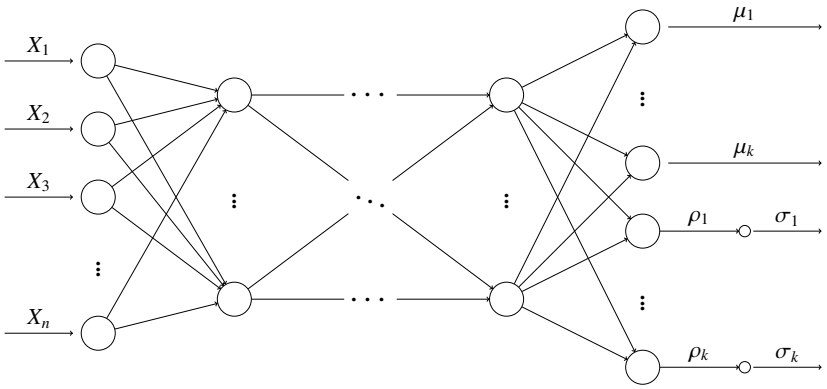


Figure 2: NN with built-in variance

Since NNs with built-in variance output both mean and variance, the MSE loss used for most regression problems is no longer a proper choice. Instead of minimizing the MSE loss, NNs with built-in variance can be trained by maximizing the likelihood $p(\mathcal{D}_y|\mathcal{D}_x, \theta)$.

If the outputs follow normal distributions and all the data instances are independent on each other, maximizing the likelihood is equivalent to minimizing the so-called negative log likelihood (NLL) loss function given in Equation 10.

$$\mathcal{L}_{\text{NLL}}(\mathcal{D}, \theta) = \frac{1}{N} \sum_{i=1}^{N} \left[\frac{\log \sigma_i^2}{2} + \frac{(y_i - \mu_i)^2}{2\sigma_i^2} \right] \tag{10}$$

However, the maximum likelihood estimation may be prone to overfitting. Another commonly used strategy is to maximize the posterior, which is the maximum-a-posterior estimation. The corresponding loss function with assumptions of normal priors is given as:

$$\mathcal{L}(\mathcal{D}, \theta) = \frac{1}{N} \sum_{i=1}^{N} \left[\frac{\log \sigma_i^2}{2} + \frac{(y_i - \mu_i)^2}{2\sigma_i^2} \right] + \lambda\theta^2 \tag{11}$$

The term $\lambda\theta^2$ is also known as the L2 regularization term. If the Laplace prior is assumed, it will become the so-called L1 regularization term.

NNs that explicitly model the output variance provide point-estimations of the aleatoric uncer-

tainty and mean output, but they cannot capture the epistemic uncertainty. Numerous algorithms have been developed to complete the NN UQ. We used the Bayesian learning methods [35] and deep ensembles [36] to close the NN UQ. Although the NN epistemic uncertainty is of more interest and assesses the data coverage, details of NNs with built-in variance are still provided here considering that many UQ techniques are implemented based on NNs with built-in variance.

3.4. Bayesian learning

Bayesian learning algorithms are widely used for NN UQ, which conducts Bayesian inference on NN parameters [19, 32, 35]. A Bayesian inference of NNs estimates the posterior distribution, $p(\theta|\mathcal{D})$, based on the given training data. Based on Bayes' theorem, the Bayesian posterior can be written as [35]:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}_y|\mathcal{D}_x, \theta)p(\theta)}{\int_\theta p(\mathcal{D}_y|\mathcal{D}_x, \theta')p(\theta')d\theta'} \propto p(\mathcal{D}_y|\mathcal{D}_x, \theta)p(\theta) \tag{12}$$

where $p(\theta)$ is the prior distribution and $p(\mathcal{D}_y|\mathcal{D}_x, \theta)$ is the likelihood term. The posterior predictive distribution against data point (x, y) can then be calculated as $p(y|x, \mathcal{D}) = \int p(y|x, \theta)p(\theta|\mathcal{D})d\theta$ and will represent the epistemic uncertainty. However, direct Bayesian inference for an NN of a practical size is intractable, since an NN contains numerous parameters of equal importance [26]. Therefore, numerous algorithms have been developed to make Bayesian learning on NNs practical. The popular Bayesian inference algorithms can be categorized into two types [35]: Markov Chain Monte Carlo (MCMC) methods and variational inference methods.

The MCMC methods sample the exact posterior of an NN. To make the MCMC computationally practical, many MCMC samplers are developed, such as stochastic gradient Langevin dynamics (SGLD) [37], preconditioned SGLD [38], stochastic gradient Hamiltonian Monte Carlo [39], and dropout Hamiltonian Monte Carlo [40]. However, the assumptions made for these methods may not always hold in practice, so they may only provide approximations of true posterior.

The variational inference methods are developed to learn an approximation scheme of the posteriors. The quality of the approximation is usually measured by the KL divergence between the variational distribution and the exact posterior. Minimizing the KL divergence is equivalent to maximizing the evidence lower bound, since the prior is fixed and usually not inferred (learned) [35].

3.4.1. Dropout as Bayesian approximation

Dropout is widely used to add regularization during training, which can reduce the risk of overfitting. It is then demonstrated in [41, 42] that dropout can serve as a Bayesian approximation to obtain the prediction uncertainty when it is used during prediction and evaluation. This approach is usually referred to as Monte Carlo dropout (MC dropout).

MC dropout can be categorized into variational inference [35], with Bernoulli distributions assigned to model parameters, and the normal loss functions used for training with dropout with additional L2 regularization are equivalent to the evidence lower bound for variational inference [41].

In practice, M stochastic forward passes through a trained NN with neurons randomly deactivated are performed, which is different from standard dropout. If MC dropout is applied to an NN treating the outputs as deterministic, the predicted mean is simply the average of all passes, which is also known as model averaging:

$$\mu(\mathbf{x}) \approx \frac{1}{M} \sum_{m=1}^{M} \mu_m(\mathbf{x}) \tag{13}$$

Only the epistemic uncertainty can be captured and is usually approximated by the variance of M samples [43]:

$$\sigma^2 \approx \frac{1}{M} \sum_{m=1}^{M} \left[\mu_m(\mathbf{x}) - \mu(\mathbf{x})\right]^2 \tag{14}$$

MC dropout can be easily implemented in FNNs compared to other Bayesian learning algorithms. It requires little additional knowledge and modeling effort. If dropout has already been used as a regularization technique, the uncertainty of a trained NN can be conveniently estimated by performing stochastic forward passes. In addition, training an NN with MC dropout is usually computationally cheaper than other variational inference algorithms [27].

However, MC dropout does not provide a high-quality estimation of the uncertainty. Although MC dropout is able to fit a 1D synthetic data set in terms of the root mean squared error, it may considerably underestimate the uncertainty [27].

3.4.2. Stochastic gradient Langevin dynamics

SGLD combining stochastic gradient algorithms and Langevin dynamics is proposed in [37] for the efficient UQ of large data sets. The proposed scheme adds a Gaussian noise to Robbins-Monro stochastic gradients:

$$\begin{aligned} \Delta\theta_t &= \frac{\epsilon_t}{2}\left[\nabla \log p(\theta_t) + \frac{N}{n}\sum_{i=1}^{n} \nabla \log p(y_{ti}|x_{ti}, \theta_t)\right] + \eta_t \\ \eta_t &\sim \mathcal{N}(0, \epsilon_t) \end{aligned} \tag{15}$$

where n is the batch size and ϵ_t is the step size (learning rate) decreasing towards zero at rates satisfying:

$$\sum_{t=1}^{\infty} \epsilon_t = \infty \qquad \sum_{t=1}^{\infty} \epsilon_t^2 < \infty \tag{16}$$

Typically, the step size will decrease exponentially as:

$$\epsilon_t = a(b + t)^{-\gamma} \tag{17}$$

where $\gamma \in (0.5, 1]$.

The mean output will be obtained by averaging the M samples weighted by learning rate:

$$f(\mathbf{x}) \approx \frac{\sum_{m=1}^{M} \epsilon_m f(\mathbf{x}, \theta_m)}{\sum_{m=1}^{M} \epsilon_m} \tag{18}$$

In this work, the QoIs are assumed to follow normal distributions and the step size will not change after the burn-in phase. Therefore, the predicted mean and the total variance can be calculated as:

$$\mu(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} \mu(\mathbf{x}, \theta_m) \tag{19}$$

$$\sigma_t^2(\mathbf{x}) = \left[\frac{1}{M}\sum_{m=1}^{M} \mu^2(\mathbf{x}, \theta_m) - \mu^2(\mathbf{x})\right] + \frac{1}{M}\sum_{m=1}^{M} \sigma^2(\mathbf{x}, \theta_m) \tag{20}$$

where $\frac{1}{M}\sum_{m=1}^{M} \mu^2(\mathbf{x}, \theta_m) - \mu^2(\mathbf{x})$ is treated as the epistemic uncertainty and $\frac{1}{M}\sum_{m=1}^{M} \sigma^2(\mathbf{x}, \theta_m)$ represents the aleatoric uncertainty.

3.5. Deep ensembles

It is proposed in [36] that ensembles of M networks, or deep ensembles (DE) for short, can be used to quantify the NN predictive uncertainty. Compared to Bayesian methods, DE is simpler to implement and requires fewer modifications to traditional NNs. To construct the ensembles, the NN parameters are randomly initialized, and the data instances are randomly shuffled. A relatively small ensemble size (e.g., M=10) is sufficient to obtain satisfactory performance [22]. To train a single NN, the NLL criterion is suggested for regression tasks [36]. In practice, the L2 regularization term is added to the NLL criterion to prevent overfitting.

During the prediction, the ensembles containing M NNs are treated as a uniformly-weighted mixture model, so the predictions are combined as:

$$p(y|\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} p(y|\mathbf{x},\theta_m) \tag{21}$$

For regression tasks, the prediction is a mixture, $\frac{1}{M}\sum_{m=1}^{M}\mathcal{N}\left(\mu(\mathbf{x},\theta_m),\sigma^2(\mathbf{x},\theta_m)\right)$. The mean and the total variance are given by:

$$\mu(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}\mu(\mathbf{x},\theta_m) \tag{22}$$

$$\sigma_t^2(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}\left[\sigma^2(\mathbf{x},\theta_m)+\mu^2(\mathbf{x},\theta_m)\right] - \mu^2(\mathbf{x}) \tag{23}$$

The total variance can be rewritten and decomposed as [44, 45]:

$$\sigma^2(\mathbf{x}) = \left[\frac{1}{M}\sum_{m=1}^{M}\mu^2(\mathbf{x},\theta_m) - \mu^2(\mathbf{x})\right] + \frac{1}{M}\sum_{m=1}^{M}\sigma^2(\mathbf{x},\theta_m) \tag{24}$$

where $\frac{1}{M}\sum_{m=1}^{M}\mu^2(\mathbf{x},\theta_m)-\mu^2(\mathbf{x})$ is treated as the epistemic uncertainty and $\frac{1}{M}\sum_{m=1}^{M}\sigma^2(\mathbf{x},\theta_m)$ represents the aleatoric uncertainty.

4. Data Coverage Assessment Workflow

4.1. Research hypothesis and assumptions

In this study, we propose that the epistemic uncertainty of NN predictions can be used to evaluate the data coverage. If the trained NN does not store sufficient knowledge against the target, the target will be considered uncovered, and the prediction error can become extremely large. This will serve as the fundamental hypothesis to be validated in this study.

We made the following assumptions:

- The simulation results obtained from GOTHIC are true, and the uncertainties of GOTHIC simulations are ignored.
- FNN and RNN are the only ML models used to develop DTs.
- The input features, output features, and NN architectures are fixed for DCA, and the possible uncertainty caused by feature selection and model structure are not considered.

4.2. Proposed workflow

Based on the hypothesis, a development workflow for DCA is proposed in Figure 3, consisting of the following steps:

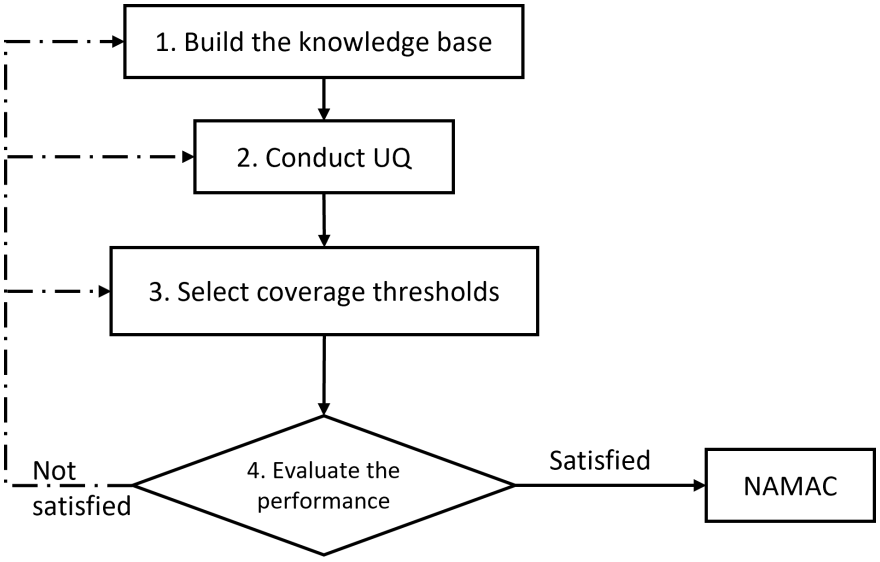


Figure 3: Development workflow for DCA

1) Step 1 aims to construct the knowledge base. The information required for the NN UQ will be collected in this step. To quantify the uncertainty of a given NN, its architecture will be analyzed. Since this study intends to assess the data coverage based on the provided training data, the data used to train the DT will be collected. Based on the preliminary analysis of the NN architecture and training data, proper UQ algorithms will be suggested. In addition, since the proposed DCA approach should be evaluated to guarantee its effectiveness, some evaluation data will be collected in this step based on expert knowledge.

2) With the developed knowledge base, Step 2 will quantify the NN prediction uncertainty. The NN UQ estimates the distributions of NN parameters based on the provided data and can be treated as a retraining process for the given NN. With fixed training data and NN architecture, the NN parameters will be retuned using the selected UQ algorithm, and the distributions of NN parameters can be obtained.

3) Step 3 aims to select the required coverage thresholds. To close the DCA, an uncertainty threshold is required. Moreover, an error threshold is required for evaluation purposes. In this study, these two thresholds are grouped as the coverage thresholds and more details will be provided in Section 4.3.

4) In Step 4, the developed DCA module will be evaluated based on its performance on the evaluation data. The evaluation will be based on whether data identified as not covered have high prediction errors and whether data identified as covered have low prediction errors. Proper evaluation metrics will be selected based on these two criteria, which will be further discussed in Section 4.3. Then the decision on whether the performance is satisfactory will be made. If the performance is not satisfactory, corrections and improvements will be made on Step 1, Step 2 and Step 3. If the performance is satisfactory, the developed DCA module will be used in the applications.

4.3. Coverage thresholds and evaluation metrics

A proper decision threshold is required to assess the data coverage condition. In this work, the outputs of DT-D consist of continuous variables, so the epistemic uncertainty will be characterized

by the standard deviation σ of the predictive distribution of μ caused by the epistemic uncertainty. The standard deviation will be calculated as:

$$\sigma = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\mu^2(\mathbf{x},\theta_m) - \mu^2(\mathbf{x})} \tag{25}$$

where $\mu(\mathbf{x},\theta_m)$ is the mean predicted by the m^{th} sample and $\mu(\mathbf{x})$ is the predicted mean averaged by Equations 13, 19, and 22. The decision regarding the data coverage condition will be made based on a coverage threshold selected based on the epistemic uncertainty, which is referred to as the uncertainty threshold. If the σ calculated for an unknown target $\mathbf{x}$ exceeds the uncertainty threshold, the target will be classified as not covered; otherwise, it will be classified as covered.

In this work, the coverage level is assessed with a binary scale. The assessment of data coverage level can be treated as a binary classification task that classifies whether a new target data is covered or uncovered. Therefore, a confusion matrix shown in Table 1 can be constructed to evaluate the performance of the proposed approach.

Table 1: Confusion matrix used to evaluate the DCA performance

		Estimated as	
		Covered	Uncovered
Reality is	Covered	True Positive (TP)	False Negative (FN)
	Uncovered	False Positive (FP)	True Negative (TN)

To complete the confusion matrix, the true coverage condition should be available. In some similar tasks, such as OoD detection, the true coverage condition is usually determined using some characteristics of samples, such as the labels, the ranges of input features, and the sources of the data. For instance, an image of fruit would not be considered covered by some images of vegetables. However, finding proper characteristics can be challenging in this study and may introduce extra uncertainty due to:

- The output space is continuous, so using the discretized labels to determine the true coverage condition is impractical.
- The control variables defining the transients, such as the pump ramping time and final pump

speed, are useful to evaluate the difference between two transients. However, the reactor may have similar behavior at the beginning of each transient and an NN trained on severe accidents may be able to provide accurate predictions on less severe accidents. Therefore, using those manipulated variables to determine the true coverage condition is not always reliable.

- Since the input features are also continuous, the ranges of input features can be alternative metrics to determine the true coverage condition. However, this method ignores the complex connections among the input features. In engineering applications, the input features are constrained by physical laws. Perturbing some input features may break the physics, which may not be detected by monitoring the ranges of the input feature. Therefore, although determining the coverage condition by checking whether the values of target inputs fall within the ranges of training inputs works well for many applications, we did not select it in this study due to this major concern.

One important observation is that well-trained NNs tend to fail when they are applied to uncovered data. Therefore, in this work, the NN performance on the target data, which is evaluated by the absolute error (AE) between the NN prediction and true value, is used to determine the true coverage condition by comparing the AE to an error threshold. If the AE is smaller than the threshold, the NN performs well and the target will be considered covered in this study. The AE is calculated as:

$$\mathrm{AE} = |\mu(\mathbf{x}) - y(\mathbf{x})| \tag{26}$$

where $\mu(\mathbf{x})$ is the predicted mean averaged by Equations 13, 19, and 22 and $y(\mathbf{x})$ is the value provided by GOTHIC simulations.

In this study, a target will be estimated as covered if the epistemic uncertainty is smaller than the uncertainty threshold. The actual coverage condition will be determined by comparing the actual prediction errors against the error threshold. If a target is estimated as covered and the real data coverage condition is also covered, it will be classified as a true positive (TP) based on Table 1. When DCA is deployed to DT-D systems of nuclear reactors, false positive (FP) is less acceptable with actual prediction errors larger than expected. For an FP, we are trusting our predictions and

assume that the prediction error is bounded, while the DT-D results may significantly deviate from the unknown true value. In a safety-critical system, failing to detect such huge prediction errors may finally lead to disastrous consequences. Therefore, FP is of more concern in this study.

Based on the confusion matrix, four evaluation metrics can be calculated to evaluate the proposed thresholds, as summarized in Equations 27-30. The true positive rate (TPR), also named sensitivity or recall, measures how well low-error predictions can be identified. The true negative rate (TNR), also named specificity, evaluates the DCA module based on how many high-error predictions can be captured. The positive predictive value (PPV), or precision, and the negative predictive value (NPV) assess how confident the model is in classifying the covered and uncovered data, respectively. Good coverage thresholds will lead to high scores in all these four metrics. However, classifying the true coverage condition by evaluating the model may increase the number of false negative (FN) predictions, since the NN predictions can accidentally agree with the truth even though the data are far away from the training data. As a result, good thresholds can sometimes lead to a low TPR and NPV. Under this condition, expert judgment must evaluate the adequacy of DCA.

$$\text{True positive rate (TPR)} = \frac{TP}{TP + FN} \tag{27}$$

$$\text{True negative rate (TNR)} = \frac{TN}{TN + FP} \tag{28}$$

$$\text{Positive predictive value (PPV)} = \frac{TP}{TP + FP} \tag{29}$$

$$\text{Negative predictive value (NPV)} = \frac{TN}{TN + FN} \tag{30}$$

4.4. Optimization objectives

The Receiver Operating Characteristic (ROC) curve can be used to visualize the performance of a binary classifier [46, 47]. It plots the TPR versus the false positive rate (FPR) by varying the corresponding threshold and qualitatively represents the trade-off between the costs and benefits of a classifier [48]. The point $(0, 1)$ represents a perfect classification, and a purely random classifier will produce a line connecting $(0, 0)$ and $(1, 1)$. To obtain the ROC curve, the TPR can be calculated

with Equation 27 and the FPR can be calculated as:

$$\text{False positive rate (FPR)} = \frac{FP}{FP + TN} \tag{31}$$

The area under the ROC curve (AUROC) can be used to quantitatively measure a classifier's performance [46]. The AUROC can be treated as a metric quantifying the probability that a classifier can correctly rank a true and false pair [47]. It is scale-invariant and uncertainty-threshold-invariant since it is independent of the absolute values of the selected classification metric and threshold. In this study, AUROC compares the performance of different UQ algorithms and selects the optimal one. In addition, the error threshold used to determine the true coverage conditions can also impact the AUROC. Therefore, the AUROC of each UQ method will first be maximized by optimizing the error threshold and then compared with each other to select the optimal UQ method. It should be noted that AUROC does not directly depend on the prediction error of a DT-D, while it is determined by whether the high-error predictions can be accurately captured. A DT-D cannot fit the evaluation data does not necessarily yield small AUROC.

The AUROC is independent of the uncertainty threshold, so it cannot determine the optimal uncertainty threshold. The optimal uncertainty threshold will be selected such that false predictions consisting of FPs and FNs are minimized. However, FPs and FNs may not respond to the change of coverage thresholds in the same way. In this study, the optimization of the uncertainty threshold will rely on a misclassification cost defined as [46]:

$$C = c_{FP}N_{FP} + c_{FN}N_{FN} \tag{32}$$

where N_{FP} is the number of FPs, N_{FN} is the number of FNs, c_{FP} is the cost of an FP, and c_{FN} is the cost of an FN. The cost coefficients, c_{FP} and c_{FN}, represent the importance of FP and FN, respectively. If FP is more harmful to the system, a higher c_{FP} should be used. Large c_{FN} indicates that FN is less acceptable than FP. The selection of c_{FP} and c_{FN} will be application oriented.

5. Case Study

5.1. Case formulation

This work is based on loss of flow accidents (LOFA) in Experimental Breeder Reactor II (EBR-II). In this study, the pump speed of Primary Sodium Pump #1 (PSP1) of EBR-II ramps down to initiate the accidents, and Primary Sodium Pump #2 (PSP2) may ramp up to mitigate the consequences. The detailed descriptions of the LOFA accidental scenarios and GOTHIC simulations used in this study can be found in Appendix B.

The development and optimization of DCA are demonstrated based on a DT-D in the NAMAC system, which restores the complete states of an EBR-II reactor in a real-time manner. Detailed descriptions of the NAMAC system and DT-D can be found in Appendix A. In this study, the DT-D is developed using FNN and RNN. The inputs of DT-D contain selected observable reactor state variables as summarized in Table 2. When FNNs are trained, the first-order time derivatives of the observable states are also used as inputs to partially capture the transient behaviors of the reactor. The output is the unobservable peak fuel centerline temperature, T_{PFCL}. If the output size is 1, only the epistemic uncertainty of T_{PFCL} is quantified. If the aleatoric uncertainty is quantified, the output size will be 2. Since the magnitude of the energy generation rate is much larger than other features, the inputs are pre-processed using min-max normalization. The training data in this study are generated using the GOTHIC EBR-II model [49]. Detailed descriptions of the scenarios of interest and the training data used in this study can be found in Appendix B.

This work formulates two case studies. In Case Study 1, the data coverage condition of an FNN-based DT-D is assessed. Two Bayesian-based UQ methods (MC dropout and SGLD) are used and the uncertainty threshold is selected based on the UQ results of training data. In Case Study 2, the DT-D is developed using RNN, and the uncertainty will be captured using DE and SGLD. The error threshold and uncertainty threshold are selected by optimizing the proposed DCA approach.

The case studies are designed to address:

1) Whether an epistemic uncertainty can be used to assess the data coverage condition?

Table 2: Selected observable state variables

Variable Number	Variable Name	Description
1	FL1	PSP1 mass flow rate
2	FL6	PSP2 mass flow rate
3	FL19	Z-pipe mass flow rate
4	TL8s1	High pressure lower plenum temperature
5	TL9s1	Low pressure lower plenum temperature
6	TL14s1	Upper plenum temperature
7	PS1	PSP1 speed
8	PS2	PSP2 speed
9	PH1	PSP1 head speed
10	PH2	PSP2 head speed
11	cv42C	Total core power generation
12	cv43C	Intermediate heat exchanger cooling power

The first question is about whether the epistemic uncertainty of an NN-based DT-D can characterize the data coverage condition. Case Study 1 will focus on answering this question, and Case Study 2 can be used to support the statement.

2) Which UQ technique should be used for DCA?

Numerous UQ algorithms can quantify the uncertainty of an NN, and different assumptions are made for those algorithms [31]. The uncertainties quantified using different algorithms may vary significantly, and some of them may perform better for DCA. Therefore, the optimal UQ technique should be properly selected based on its performance in the task of DCA. This question is formally discussed in Case Study 2.

3) How should the uncertainty threshold be selected?

A decision threshold is required to assess the data coverage condition. This question deals with two sub-questions: 1) how should epistemic uncertainty be characterized and 2) what is the corresponding threshold separating covered and uncovered data? This question is primarily addressed in Case Study 1 and further discussed in Case Study 2.

4) Is the proposed approach scalable?

No single DT in the NAMAC system can deal with all the possible conditions. Future studies

should investigate whether the proposed DCA approach can be scaled to DTs with different architectures and various training data. This question will be addressed by demonstrating the DCA on two case studies.

5.2. Case Study 1—DCA on FNN

5.2.1. Experiment setups

In Case Study 1, the DT-D is developed using FNN. The FNN UQ hyperparameters are summarized in Table 4 and the DT-Ds are developed using PyTorch [30]. As summarized in Table 3, Case Study 1 uses one training data set and three evaluation data sets. The Training Set, Evaluation Set 1, and Evaluation Set 2 are sampled from the database "aa01." Compared to episodes in the training data set, more severe accident transients are included in Evaluation Set 1. The accident transients involved in Evaluation Set 2 are similar to those used to train the DT. Data in Evaluation Set 3 come from database "ba01," where PSP2 does not ramp up to mitigate the accidents.

Table 3: Training and evaluation data used in Case Study 1

Data set	Database	Number of episodes	Characteristics	
Training	aa01	42	Trigger temperature ($°C$):	606.00, 630.26, 651.48
			Final pump speed:	> 130%
Evaluation 1	aa01	6	Trigger temperature ($°C$):	651.48, 700.00
			Final pump speed:	105%, 111%, 121%
Evaluation 2	aa01	4	Trigger temperature ($°C$):	618.13, 639.35
			Final pump speed:	137%, 141%
Evaluation 3	ba01	3	Pump ramping time (s):	1, 113.68, 451.71
			Final pump speed:	0

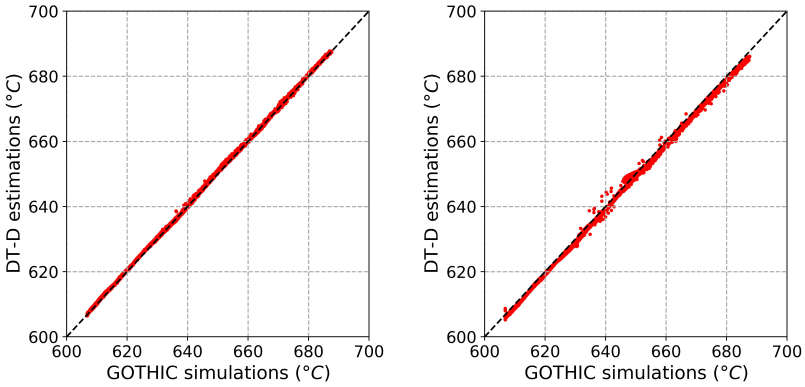
5.2.2. UQ of FNN-based DT-Ds

In this case study, the FNN uncertainty is quantified using two different methods: MC dropout and SGLD. First, the effectiveness of Bayesian methods in capturing the training data behaviors

Table 4: Selected FNN hyperparameters in case study 1

Model Name	FNN-MCD	FNN-SGLD
Number of hidden layers	4	4
Hidden size	128, 128, 128, 128	128,128,128,128
Input size	24	24
Output size	1	2
Learning rate	1×10^{-4}	$10^{-4}\left(5 \times 10^{3} + n\right)^{-0.6}$
Batch size	64	128
Weight decay	1×10^{-7}	1×10^{-5}
Optimizer	Adam	SGLD
Maximum epochs/burn-in steps	20,000	800,000
Loss function	MSE	NLL
Number of NNs	10,000	1000
UQ method	MC dropout (drop rate = 0.1)	SGLD

is checked. The comparison between the mean T_{PFCL} estimated by Bayesian learning and the GOTHIC simulation results is depicted in Figure 4. After UQ, DT-D results agree well with the true values. MC dropout performs better than SGLD in terms of the mean prediction and SGLD results slightly underestimate the T_{PFCL}.



(a) FNN-MCD (b) FNN-SGLD

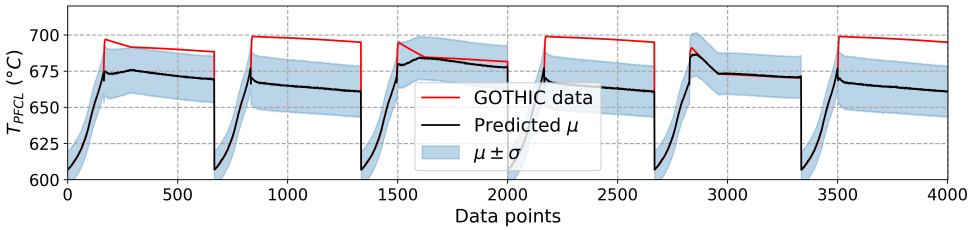
Figure 4: Model fitting on training data of Case Study 1

To study the performance of the inferred FNNs on different target data, the FNNs are implemented to the three evaluation data sets. Figure 5 shows how an FNN inferred using MC dropout fits the evaluation data and Figure 6 depicts the predictions made by the FNN using SGLD. The x-axes of Figure 5 and Figure 6 are labeled with "data points" representing the time-discretized

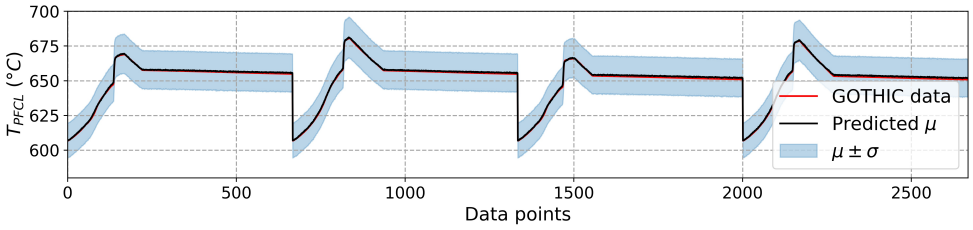
predictions of episodes. As observed from Figure 5b and Figure 6b, both models can make accurate predictions against Evaluation Set 2 with low epistemic uncertainty. However, the discrepancies between the predicted μ and true values of Evaluation Sets 1 and 3 are larger, as shown in Figure 5a, Figure 5c, Figure 6a, and Figure 6c. In addition, the trained NNs are less confident in predicting the T_{PFCL} of transients in Evaluation Sets 1 and 3 considering the large prediction uncertainty. Using SGLD, the DT-D estimations can easily become nonphysical if it is used on Evaluation Set 3, and the epistemic uncertainty suggests that the DT-D is extremely uncertain about its predictions, which can be observed from Figure 6c.

Compared to the episodes in the training set, all the episodes in Evaluation Set 1 and Evaluation Set 2 start from the same steady state and are initialized by the same type of PSP1 malfunction. Therefore, at the beginning of each episode, the DT-Ds can accurately predict the reactor behaviors with relatively low uncertainty. After the PSP1 stops ramping down, the epistemic uncertainty becomes larger for the episodes in Evaluation Set 1, since the implemented mitigation strategies are not covered by those used in the Training Set. The episodes in Evaluation Set 2 are similar to those in the Training Set and are within the prediction capability of the developed DT-Ds, so the prediction errors and epistemic uncertainty against Evaluation Set 2 are the smallest. For the episodes in Evaluation Set 3, mitigation strategies are not injected and the PSP1 malfunction varies, so they are significantly different from the episodes used for training. As a result, the developed DT-Ds only fit the very beginning of the third episode in Evaluation Set 3. The prediction errors and epistemic uncertainty against Evaluation Set 3 are the largest.

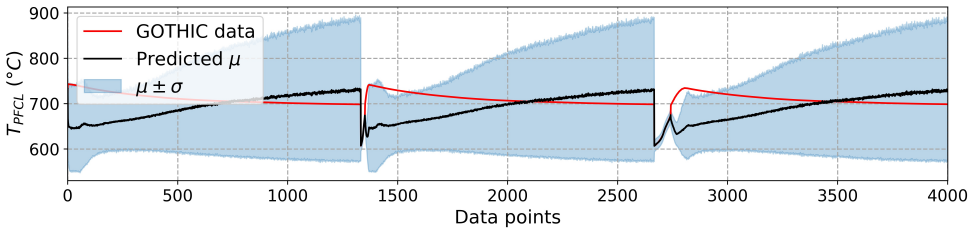
The DT-D prediction error is plotted against the epistemic uncertainty estimated in Figure 7, and the correlations between them are visually evaluated. Figure 7a, Figure 7d and Figure 7f show that the σ and prediction error are positively correlated. As the epistemic uncertainty elevates, the corresponding prediction error becomes larger. Moreover, if σ is low, the prediction error is usually much lower. Based on Figure 7b and Figure 7e, when σ is about 13.5 °C for MC dropout and about 10 °C for SGLD, most prediction errors are below 5 °C, and no clear correlation between the σ and prediction error can be extracted. In addition, the DT-D prediction can sometimes agree with the GOTHIC data when the uncertainty is large, as shown in Figure 7c. As depicted in Figure 5c, the predicted transient behaviors of these data do not follow the trends simulated by



(a) Evaluation 1



(b) Evaluation 2



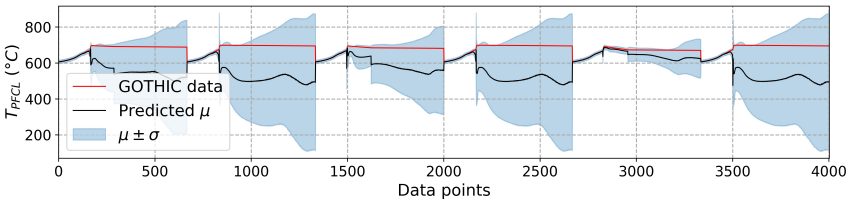
(c) Evaluation 3

Figure 5: Model fitting of FNN-MCD on evaluation data of Case Study 1

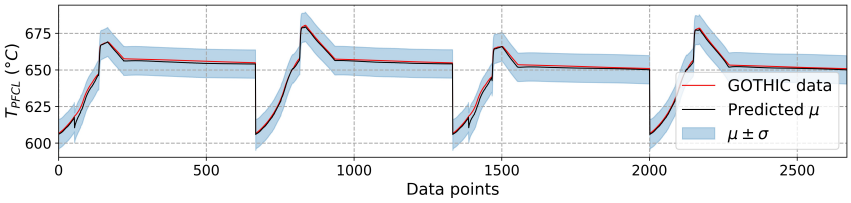
GOTHIC. Therefore, although the NNs accurately predict the T_{PFCL} of these data, they should still be considered not covered. Based on the graphical evaluation, it can be summarized that, if σ is low, small prediction error would be expected. For predictions with huge σ, they usually cannot be trusted considering the huge difference between them and the true values and should not be considered well covered. The uncovered and covered data can be separated from each other by selecting a proper uncertainty threshold. However, how the uncertainty threshold should be selected is still subject to investigation.

5.2.3. Selection of coverage thresholds

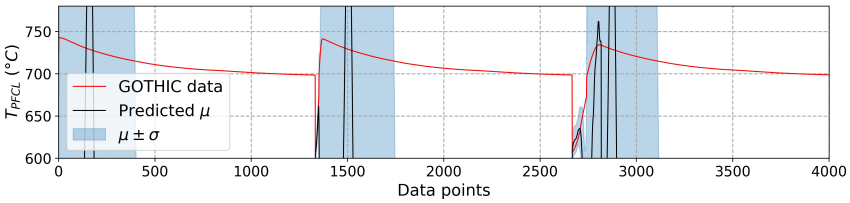
As discussed, an uncertainty threshold is required to close the workflow. Moreover, an error threshold is required to evaluate the DCA performance. In Case Study 1, we derive these two thresholds based on the training results, since we expect optimal performance when the DT-Ds are



(a) Evaluation 1



(b) Evaluation 2



(c) Evaluation 3

Figure 6: Model fitting of FNN-SGLD on evaluation data of Case Study 1

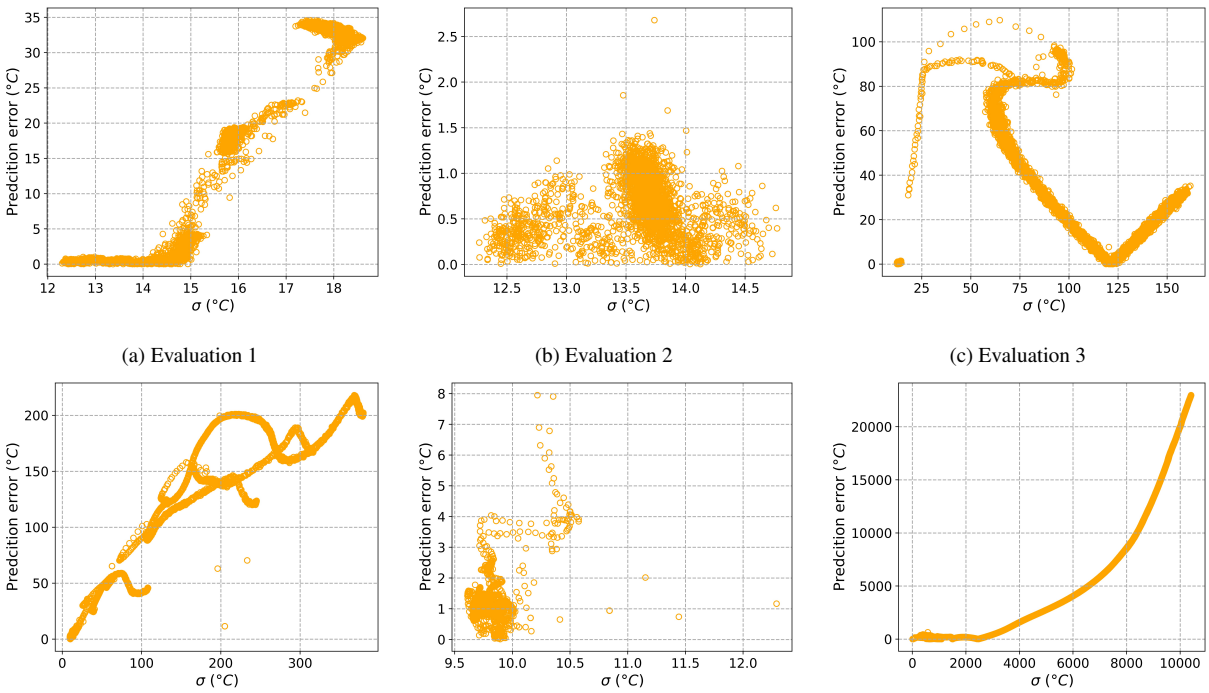
applied to the training data. If the epistemic uncertainty exceeds the uncertainty against training data, the corresponding data should be classified as uncovered. The error threshold will be selected based on the DT-D errors against the training data to determine the true coverage condition.

The mean value is a common metric to characterize the behaviors of a set of instances. The mean quantity of N instances, $\bar{\psi}$, is simply defined as:

$$\bar{\psi} = \frac{1}{N} \sum_{i=1}^{N} \psi^i \tag{33}$$

where ψ will be either σ or AE in this study.

However, the σ and AE against training data usually fluctuate around the mean value, so using mean values as metrics can be misleading. In this work, the $m\%$ bound is an alternate way to



(a) Evaluation 1 (b) Evaluation 2 (c) Evaluation 3

(d) Evaluation 1 (e) Evaluation 2 (f) Evaluation 3

Figure 7: Correlation between epistemic uncertainty and prediction error (a,b,c: FNN-MCD; d,e,f: FNN-SGLD)

derive the thresholds. The $m\%$ bound, ψ_m, is defined as the quantity satisfying:

$$\frac{m}{100} \approxeq \frac{\sum_{i=1}^{N} I\left(\psi^i, \psi_m\right)}{N} \tag{34}$$

where N is the total number of instances and $I\left(\psi^i, \psi_m\right)$ is defined as:

$$I\left(\psi^i, \psi_m\right) = \begin{cases} 1, \text{if } \psi^i \leq \psi_m \\ 0, \text{if } \psi^i > \psi_m \end{cases} \tag{35}$$

In Case Study 1, ψ_{99} will be selected for the corresponding thresholds.

5.2.4. DCA performance evaluation on FNNs

Using AE_{99} and σ_{99} as the uncertainty and error thresholds, the DCA performance is quantified in Table 5. Based on the definitions of σ_{99} and AE_{99}, some training data will be labelled as uncovered. Therefore, we expect a small number of false classifications and consider them accept-

able. As observed from Table 5, if the three evaluation sets are treated separately, most metrics in the three evaluations are satisfied, except for TNR and NPV in Evaluation Set 2, and TPR and PPV in Evaluation Set 3. If the three evaluation sets are considered as a whole, both models can distinguish uncovered data from covered data, and the four metrics selected are larger than 0.95.

Table 5: Confusion matrix and evaluation metrics (using AE_{99} and σ_{99} as thresholds)

UQ Algorithm	Data set	N_{TP}	N_{TN}	N_{FP}	N_{FN}	TPR	TNR	PPV	NPV
MC Dropout	Evaluation 1	1426	2442	40	94	0.94	0.98	0.97	0.96
	Evaluation 2	2584	0	79	5	1.00	0.00	0.97	0.00
	Evaluation 3	88	3809	6	98	0.47	1.00	0.94	0.97
	Overall	4098	6251	125	197	0.95	0.98	0.97	0.97
SGLD	Evaluation 1	768	3159	42	33	0.96	0.99	0.95	0.99
	Evaluation 2	2493	57	104	14	0.99	0.35	0.96	0.80
	Evaluation 3	3	3987	4	7	0.30	1.00	0.43	1.00
	Overall	3264	7203	150	54	0.98	0.98	0.96	0.99

Evaluation Sets 2 and 3 represent two extreme situations: most data in Evaluation Set 2 can be covered by the training data and most data in Evaluation Set 3 are uncovered. For Evaluation Set 2, more than 95% of predictions are TPs. The amount of the other three types of predictions is small, so the relatively low TNR and NPV may be misleading. Using MC dropout, no prediction in Evaluation Set 2 is grouped as TN and the number of FNs is extremely low, so TNR and NPV are even 0. Moreover, the maximum prediction error of FPs is below 5 °C, so the performance may still be considered acceptable.

In Evaluation Set 3, most predictions are TNs. It is worth mentioning that the NN predictions of uncovered data can sometimes agree with the true value, which leads to a relatively large number of FNs. Since more than 95% of the data in Evaluation Set 3 are not covered by the training data, the numbers of TPs and FPs are low. As a result, TPR and PPV are relatively low for Evaluation Set 3. Therefore, the present method based on confusion numbers may not apply to extreme cases with a very small amount of positive or negative cases. The requirements on data amount are valid since data coverage can usually be easily detected in extreme scenarios, while this study focuses on transients with both covered and uncovered data.

5.2.5. Summary of Case Study 1

This case study investigates whether epistemic uncertainty can be used to assess the data coverage condition. Epistemic uncertainties of FNN-based DT-Ds are quantified using Bayesian learning. The qualitative analysis finds that there may exist some threshold that can separate accurate predictions from unreliable predictions.

To evaluate the DCA performance against the three evaluation sets, the values of the uncertainty and error thresholds are tentatively derived based on the training results. The data in Evaluation Sets 2 and 3 are unbalanced, so poor performance in terms of some evaluation metrics may be misleading and compromise the effectiveness of the proposed DCA approach. When the three evaluation sets are treated as a single data set, the data are more balanced, and all the evaluation metrics are satisfied. It should be noted that the results obtained based on the confusion matrix may not apply to extreme cases with all covered/uncovered data.

5.3. Case Study 2—DCA on RNN and optimization of DCA

5.3.1. Experiment setups

To check the scalability of the proposed framework and demonstrate the optimization process, Case Study 2 is formulated using GOTHIC data generated by the improved EBR-II model. The DT-D is developed using RNN, which is better at capturing the time-series data and is used in the demonstration of advanced NAMAC system [9]. As shown in Table 6, one training data set and one evaluation data set are used in Case Study 2. The training and evaluation data consist of more complicated scenarios used in [9] to make the scope of this study consistent with that of the developed NAMAC system. Detailed episode descriptions can be found in Appendix B.

Table 6: Training and evaluation data used in Case Study 2

Data set	Database	Number of episodes	Characteristics		
			PSP1 final speed	PSP2 final speed	Trigger time (s)
Training	Q8	54	22.2%	$\geq$ 145.8%	50 – 100
			$>$ 22.2%	100% – 150%	50 – 100
Evaluation	Q8	14	$<$ 22%	100% – 150%	50 – 100
			22.2%	$\leq$ 143.8%	50 – 100

5.3.2. UQ of RNN-based DT-Ds

In Case Study 2, the DT-D is developed using RNN utilizing LSTM cells. DE and SGLD can estimate the RNN uncertainty. The selected hyperparameters are summarized in Table 7. If DE is used, the ensembles are constructed by randomly initializing the NNs. In this study, the initial NN parameter values are sampled from five different distributions, as summarized in Table 8. In addition, the MSE criterion is compared to the NLL criterion which is recommended for DE [36].

Table 7: Selected RNN hyperparameters in Case Study 2

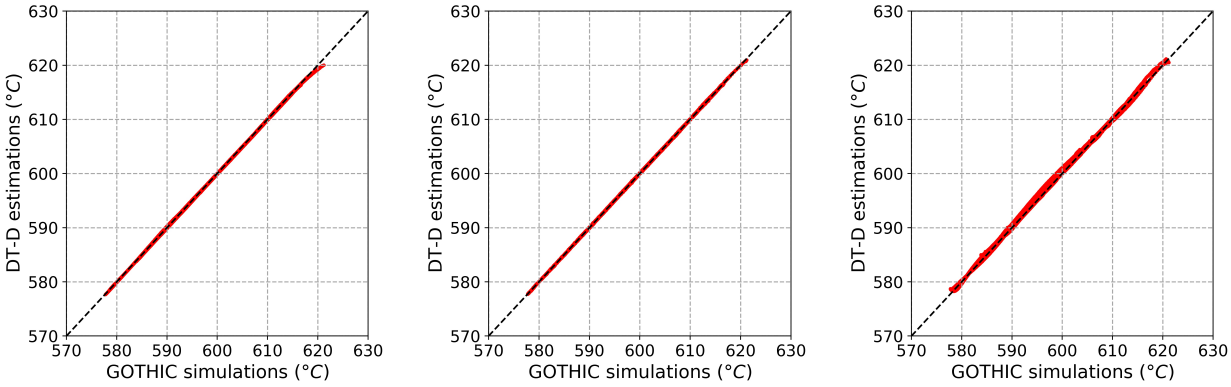
Model Name	RNN-DE	RNN-DE-var	RNN-SGLD
Number of LSTM layers	2	2	2
Hidden size	128, 128	128, 128	128, 128
Input size	12	12	12
Output size	1	2	2
Sequence length	10	10	10
Learning rate	0.01 (Initial)	0.01 (Initial)	$10^{-4}\left(5\times10^{3}+n\right)^{-0.6}$
Batch size	1024	1024	64
Weight decay	1×10^{-7}	1×10^{-7}	1×10^{-5}
Optimizer	Adam	Adam	SGLD
Sequence length	10	10	10
Maximum epochs/burn-in steps	10,000	10,000	1,740,000
Loss function	MSE	NLL	NLL
Number of NNs	10	10	400
UQ algorithm	DE	DE	SGLD

Table 8: Initialization of NNs in DE

Initialization	Number of NNs
$\mathcal{U}\left(-\sqrt{\frac{1}{\text{hidden\_size}}}, \sqrt{\frac{1}{\text{hidden\_size}}}\right)$	2
$\mathcal{U}(-0.1, 0.1)$	2
$\mathcal{U}(-0.2, 0.2)$	2
$\mathcal{N}\left(0, 0.1^2\right)$	2
$\mathcal{N}\left(0, 0.2^2\right)$	2

As shown in Figure 8, the predicted μ of training data after UQ can agree well with the truth. Figure 9 depicts the results when the RNNs are applied to the evaluation data set. Although the DT-Ds are well calibrated against the training data, many predictions of data in the evaluation set

become unreliable, and unreliable predictions usually have more significant uncertainty, which are consistent with the observations in Case Study 1.



(a) RNN-DE (b) RNN-DE-var (c) RNN-SGLD

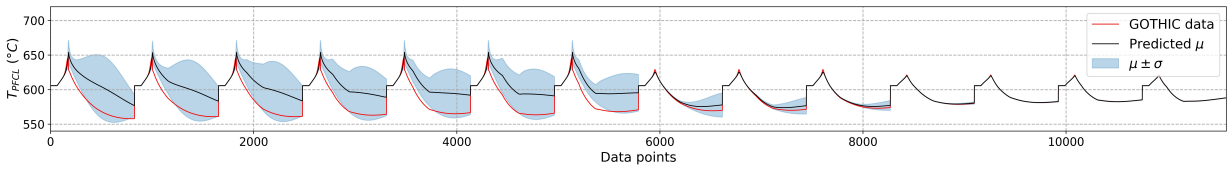
Figure 8: Model fitting on training data of Case Study 2

5.3.3. Selection of coverage thresholds

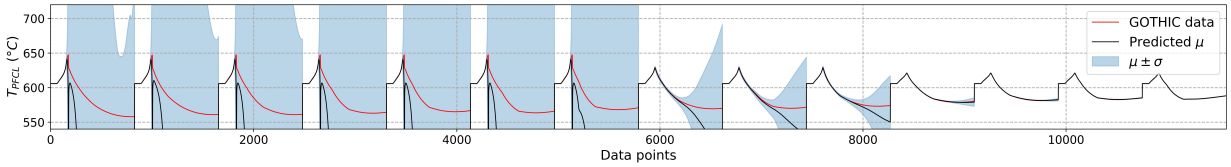
σ_{99} and AE_{99} are used as the initial guess of coverage thresholds. As shown in Figure 10, with the initial guess, all three methods can accurately capture the behaviors of data classified as covered based on a graphical comparison. However, the number of false classifications is high, and the four metrics are only around 0.9, as summarized in Table 10. Therefore, the uncertainty estimator and coverage thresholds can be tuned to obtain an optimal performance. This section demonstrates how the DCA can be optimized.

First, the UQ technique performance is assessed using the ROC curve and corresponding AUROC, based on which the optimal UQ technique is selected. Using AE_{99} as the error threshold, the ROC curves of different UQ methods are compared in Figure 11. All the methods perform well in terms of the AUROC. Although DE methods outperform SGLD, the AUROC of SGLD is still larger than 0.9, which indicates that SGLD can still be a good method to assess the data coverage.

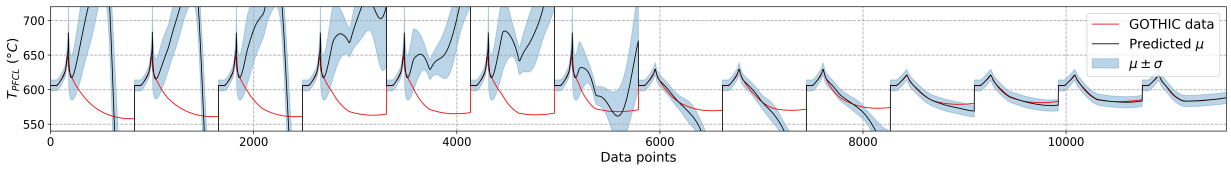
In this work, the true coverage condition is determined using an error threshold, so the ROC curve may vary if the error threshold is changed. The AE_{99} derived for different algorithms are not the same and may not maximize the AUROC. Therefore, to select the optimal UQ algorithm, it is



(a) RNN-DE

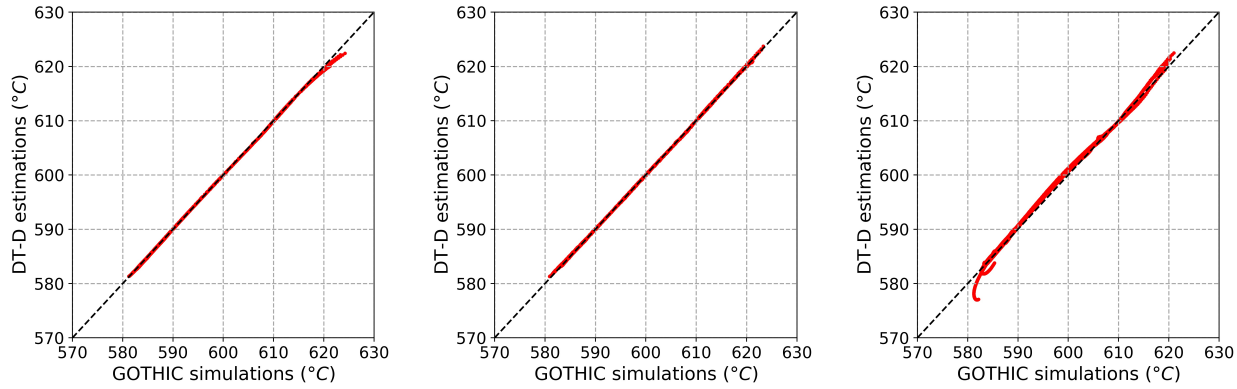


(b) RNN-DE-var



(c) RNN-SGLD

Figure 9: Model fitting on evaluation data of Case Study 2



(a) RNN-DE (b) RNN-DE-var (c) RNN-SGLD

Figure 10: Model fitting on covered target data of Case Study 2 using RNN (using σ_{99} as the uncertainty threshold)

necessary to find the maximal AUROC by optimizing the error threshold. Since the NAMAC system intends to provide recommendations to reactor operators and should maintain reactor safety, it is unreasonable to select a large error threshold. Therefore, how the AUROC changes as the error threshold sampled from 0.1 °C to 10 °C varies is studied and summarized in Figure 12. RNN-DE-var shows the largest maximal AUROC, followed by RNN-DE, and RNN-SGLD has the lowest maximal AUROC. The maximal AUROC of DE using NLL criterion is around 1.0 if the error

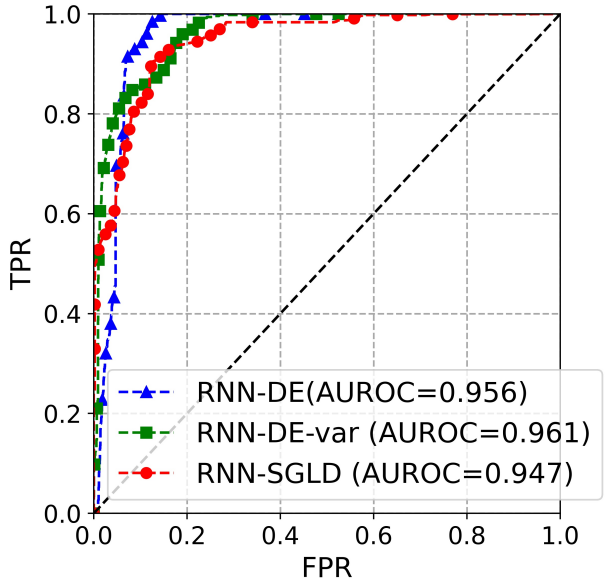


Figure 11: ROC curve and AUROC (using AE_{99} as error threshold)

threshold is larger than 1.5 °C, under which condition it is almost a perfect data coverage assessor. Although the SGLD performance is worse than the other two methods, its AUROC still exceeds 0.96, so it can still be an acceptable method for DCA. In summary, DE with NLL loss outperforms the other two methods and will be considered as the optimal UQ algorithm. Although RNN-DE-var is the optimal model, it will still be compared to other models in the following discussions for investigation purposes.

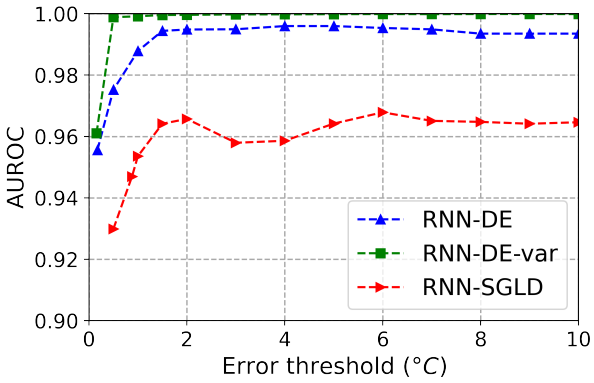


Figure 12: AUROC using different error thresholds

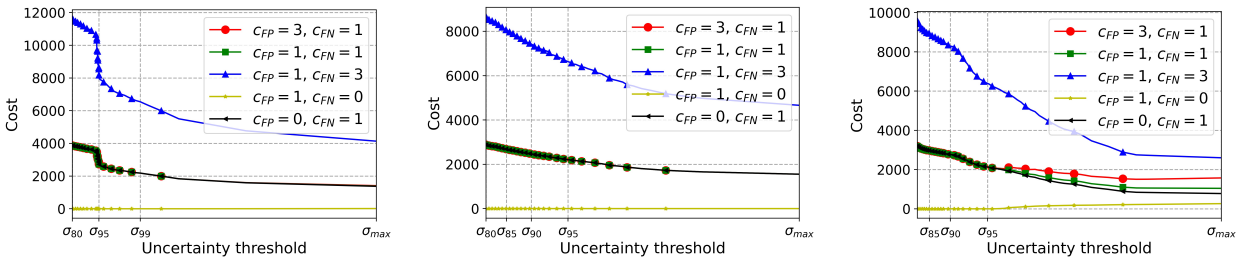
For each model, an optimal error threshold will be determined. As shown in Figure 12, the AU-

ROC of RNN-DE-var does not significantly increase if the error threshold is larger than 1.5 °C, so 1.5 °C is taken as the optimal error threshold for RNN-DE-var. Considering the trade-off between AUROC and DT-D error, 2 °C is used as optimal error threshold for RNN-DE and RNN-SGLD.

The uncertainty threshold is a critical hyperparameter in DCA. The ROC curve is plotted by varying the uncertainty threshold, so the AUROC does not inform people of what uncertainty threshold should be selected. The optimal uncertainty threshold will be determined based on the misclassification cost defined in Equation 32. The uncertainty threshold is initially proposed to be derived based on the training results, so the uncertainty threshold is firstly sampled from σ_{80} to σ_{max} (σ_{100}). The cost coefficients indicate how false predictions are weighted. As summarized in Table 9, five sets of cost coefficients are used in this study to vary the weight to false predictions. As depicted in Figure 13, the number of FPs will increase and FNs will decrease if the uncertainty threshold becomes larger. Thus the optimal uncertainty threshold may vary if false predictions are weighted differently. In this work, the safety of the nuclear reactor is vital, so FPs are more harmful and may lead to disastrous consequences. Therefore c_{FP} should be larger than c_{FN}. If not specified, $c_{FP} = 3$ and $c_{FN} = 1$ will be used in the following discussions.

Table 9: Cost coefficients used in Case Study 2

c_{FP}	c_{FN}	Interpretation
1	1	FPs and FNs are of equal concern
3	1	FPs are of more concern
1	3	FNs are of more concern
1	0	Only FPs are of concern
0	1	Only FNs are of concern



(a) RNN-DE (b) RNN-DE-var (c) RNN-SGLD

Figure 13: Misclassification cost with different uncertainty thresholds (using the optimal error threshold)

Although DE is superior to SGLD in terms of AUROC, the misclassification costs using DE are comparable to the SGLD cost due to the large number of FNs. Using DE, the ensembles of predictions against training data have low variance, so the uncertainty threshold derived based on the training results would be small. As a result, numerous target data that can be accurately predicted are not assessed as covered due to the low uncertainty threshold, and the number of FNs becomes huge. If the uncertainty threshold continues to increase beyond the range $[\sigma_{80}, \sigma_{max}]$, it is reasonable to expect that the number of FNs and the total misclassification cost will decrease.

Figure 14 illustrates how misclassification costs change when the uncertainty threshold is selected from an extended range. As the uncertainty threshold increases from σ_{max}, the SGLD cost continues to increase and those of DE will decrease. In addition, the minimal DE costs will be much smaller than the SGLD cost. However, the optimal uncertainty threshold for the DE method will be much larger than the training uncertainty. It should be noted that true epistemic uncertainty is usually unknown and all the UQ techniques only provide estimations based on specific assumptions, so it is difficult to select the uncertainty threshold based on expert judgment. The epistemic uncertainty against the training data might be the best knowledge people can get. Thus, people should be careful when an uncertainty threshold much greater than σ_{max} is selected.

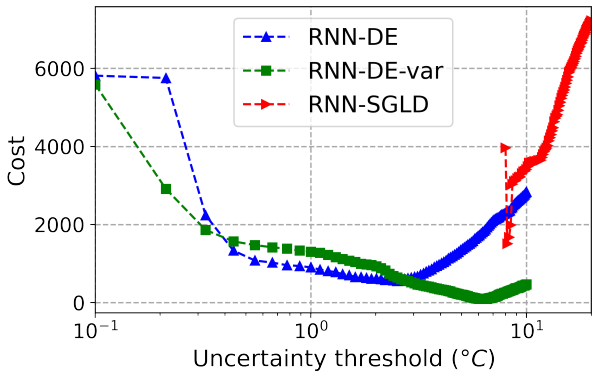


Figure 14: Misclassification cost using large uncertainty thresholds ($c_{FP} = 3$, $c_{FN} = 1$)

5.3.4. Performance evaluation of DCA on RNNs

The confusion matrix results with the initial guess and after each step of optimizations are summarized in Table 10. Using AE_{99} and σ_{99}, which are derived based on the training results, the

RNN-DE performance is the best but is still not satisfactory. With a given UQ algorithm, the optimization process starts from the optimization of the error threshold. The optimal error threshold leads to the maximal AUROC and effectively reduces the number of FPs. However, it significantly increases the number of FNs. To reduce the number of false predictions, the uncertainty threshold is optimized by minimizing a defined misclassification cost. If the optimal uncertainty threshold is searched based on training results, the misclassification costs of DE methods do not see the local minimum, as shown in Figure 13, and the number of FPs remains large after the optimization. Therefore, the search range for the uncertainty threshold is extended beyond the training results. With the extended range, the misclassification costs of DE methods can be reduced to low levels. For SGLD, the initial guess of the uncertainty threshold (σ_{99}) is already close to the optimal one, so the optimization of the uncertainty threshold does not significantly improve the performance.

After the optimization process, RNN-DE-var shows the best performance, which is consistent with the results of the AUROC comparison. Using the optimal coverage thresholds, RNN-DE-var can effectively distinguish a covered target from an uncovered target with few misclassifications. The four evaluation metrics selected are also close to 1.

Table 10: Confusion matrix and evaluation metrics

Model	Error Threshold	Uncertainty Threshold	N_{TP}	N_{TN}	N_{FP}	N_{FN}	TPR	TNR	PPV	NPV
RNN-DE	AE_{99}	σ_{99}	3156	6995	479	948	0.77	0.94	0.87	0.88
	Optimal	σ_{99}	3635	5766	0	2177	0.63	1.00	1.00	0.73
	Optimal	Optimal (training results)	4438	5753	13	1374	0.76	1.00	1.00	0.81
	Optimal	Optimal (extended range)	5744	5603	163	68	0.99	0.97	0.97	0.99
RNN-DE-var	AE_{99}	σ_{99}	2990	7151	978	459	0.87	0.88	0.75	0.94
	Optimal	σ_{99}	3968	5805	0	1805	0.69	1.00	1.00	0.76
	Optimal	Optimal (training results)	4219	5805	0	1554	0.73	1.00	1.00	0.79
	Optimal	Optimal (extended range)	5687	5802	3	86	0.99	1.00	1.00	0.99
RNN-SGLD	AE_{99}	σ_{99}	3082	6995	909	592	0.84	0.88	0.77	0.92
	Optimal	σ_{99}	3803	6502	188	1085	0.78	0.97	0.95	0.86
	Optimal	Optimal (training results)	4045	6469	221	843	0.83	0.97	0.95	0.88
	Optimal	Optimal (extended range)	4032	6471	219	856	0.82	0.97	0.95	0.88

6. Conclusion

In this work, data coverage conditions of NN-based DT-Ds in the NAMAC system are assessed based on the epistemic uncertainty of neural networks. The quantified epistemic uncertainty is characterized by the standard deviation of the predictive distribution and compared to an uncertainty threshold to assess the data coverage condition. A confusion matrix is constructed to evaluate and optimize the DCA performance. Different UQ algorithms are compared, and the optimal one is selected based on AUROC. The uncertainty threshold is optimized by minimizing a misclassification cost.

In this work, two case studies are formulated. The training and evaluation data are generated by GOTHIC simulations under different scenarios. FNNs and RNNs are used to develop the DT-D. The epistemic uncertainty of an FNN-based DT-D is quantified using the MC dropout and SGLD, based on which uncovered data can be distinguished from covered data. With an initial guess of the coverage thresholds based on the training results, results from four evaluation metrics all exceed 0.95, indicating a satisfied performance of the proposed DCA approach. For an RNN-based DT-D, the epistemic uncertainty is determined by DE with MSE loss, DE with NLL loss, and SGLD. With the thresholds derived from the training results, the DCA performance is not satisfactory and needs further improvements. By optimizing the thresholds, the DCA performance is significantly improved, and most evaluation metrics reach 0.95. The results after optimization suggest that DE with NLL loss outperforms both SGLD and DE with MSE loss.

The case studies suggest that epistemic uncertainty can be used to assess the data coverage of NN-based DT-Ds. However, the coverage thresholds derived based on training results only apply to FNN-based DT-D developed for scenarios with two manipulated variables. For an advanced NN like RNN used for more complicated scenarios, the thresholds need optimization to achieve the same level of accuracy as the DCA results for FNNs. Future works will focus on more accurate characterization of epistemic uncertainty and other potential methods for DCA, such as distance-based and density-based methods.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is performed with the support of the ARPA-E MEITNER program under the project entitled: "Development of a Nearly Autonomous Management and Control System for Advanced Reactors" and using a GOTHIC license provided by Zachry Nuclear Engineering, Inc. GOTHIC incorporates technology developed for the electric power industry under the sponsorship of the Electric Power Research Institute. The authors would also like to thank Drs. Xu Wu, Maria Avramova, Ralph Smith, Jeffrey Lane, Paridhi Athe and Pascal Rouxelin for their comments, suggestions and support.

This manuscript has been authored by Battelle Energy Alliance, LLC under Contract No. DE-AC07-05ID14517 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes.

References

[1] R. E. Uhrig, L. H. Tsoukalas, Multi-agent-based anticipatory control for enhancing the safety and performance of generation-iv nuclear power plants during long-term semi-autonomous operation, Progress in Nuclear Energy 43 (2003) 113–120.

[2] M. G. Na, B. R. Upadhyaya, X. Xu, I. J. Hwang, Design of a model predictive power controller for an sp-100 space reactor, Nuclear science and engineering 154 (2006) 353–366.

[3] B. Upadhyaya, K. Zhao, S. Perillo, X. Xu, M. Na, Autonomous control of space reactor systems (no. doe/id/14589), University of Tennessee, Knoxville, TN (United States) (2007).

[4] K. Hu, J. Yuan, Multi-model predictive control method for nuclear steam generator water level, Energy Conversion and Management 49 (2008) 1167–1174.

[5] M. C. Darling, G. F. Luger, T. B. Jones, M. R. Denman, K. M. Groth, Intelligent modeling for nuclear power plant accident management, International Journal on Artificial Intelligence Tools 27 (2018) 1850003.

[6] D. Lee, A. M. Arigi, J. Kim, Algorithm for autonomous power-increase operation using deep reinforcement learning and a rule-based system, IEEE Access 8 (2020) 196727–196746.

[7] L. Lin, P. Athe, P. Rouxelin, M. Avramova, A. Gupta, R. Youngblood, J. Lane, N. Dinh, Development and assessment of a nearly autonomous management and control system for advanced reactors, Annals of Nuclear Energy 150 (2021) 107861.

[8] J. Lee, L. Lin, P. Athe, N. Dinh, Development of the machine learning-based safety significant factor inference model for diagnosis in autonomous control system, Annals of Nuclear Energy (2021) 108443.

[9] L. Lin, P. Athe, P. Rouxelin, M. Avramova, A. Gupta, R. Youngblood, J. Lane, N. Dinh, Digital-twin-based improvements to diagnosis, prognosis, strategy assessment, and discrepancy checking in a nearly autonomous management and control system, Annals of Nuclear Energy 166 (2022) 108715.

[10] A. Gurgen, L. Lin, N. Dinh, Development and assessment of physics-guided machine learning framework for prognosis system, in: Transactions of 2020 ANS Virtual Winter Meeting, volume 123, 2020, pp. 272–275.

[11] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: International Conference on Learning Representations, 2015. URL: http://arxiv.org/abs/1412.6572.

[12] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 427–436.

[13] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, D. Song, Natural adversarial examples, arXiv preprint arXiv:1907.07174 (2019).

[14] K. Xu, M. Zhang, J. Li, S. S. Du, K.-I. Kawarabayashi, S. Jegelka, How neural networks extrapolate: From feedforward to graph neural networks, in: International Conference on Learning Representations, 2021. URL: https://openreview.net/forum?id=UH-cmocLJC.

[15] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, D. Song, Anomalous example detection in deep learning: A survey, IEEE Access 8 (2020) 132330–132347. doi:10.1109/ACCESS.2020.3010274.

[16] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K. R. Müller, A unifying review of deep and shallow anomaly detection, Proceedings of the IEEE (2021) 1–40. doi:10.1109/JPROC.2021.3052449.

[17] H. Bao, N. T. Dinh, J. W. Lane, R. W. Youngblood, A data-driven framework for error estimation and mesh-model optimization in system-level thermal-hydraulic simulation, Nuclear Engineering and Design 349 (2019) 27–45.

[18] A. Gurgen, N. T. Dinh, Development and assessment of a reactor system prognosis model with physics-guided machine learning, Nuclear Engineering and Design 398 (2022) 111976. URL: https://www.sciencedirect.com/science/article/pii/S0029549322003272. doi:https://doi.org/10.1016/j.nucengdes.2022.111976.

[19] P. Jacquier, A. Abdedou, V. Delmas, A. Soulaïmani, Non-intrusive reduced-order modeling using uncertainty-aware deep neural networks and proper orthogonal decomposition: Application to flood modeling, Journal of Computational Physics 424 (2021) 109854. URL: http://www.sciencedirect.com/science/article/pii/S0021999120306288. doi:https://doi.org/10.1016/j.jcp.2020.109854.

[20] H. Palacci, H. Hess, Scalable natural gradient langevin dynamics in practice, 2018. arXiv:1806.02855.

[21] N. Tagasovska, D. Lopez-Paz, Single-model uncertainties for deep learning, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/73c03186765e199c116224b68adc5fa0-Paper.pdf.

[22] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, J. Snoek, Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/8558cb408c1d76621371888657d2eb1d-Paper.pdf.

[23] M. Combalia, F. Hueto, S. Puig, J. Malvehy, V. Vilaplana, Uncertainty estimation in deep neural networks for dermoscopic image classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 744–745.

[24] T. Zhou, T. Han, E. L. Droguett, Towards trustworthy machine fault diagnosis: A probabilistic bayesian deep learning framework, Reliability Engineering & System Safety 224 (2022) 108525. URL: https://www.sciencedirect.com/science/article/pii/S095183202200179X. doi:https://doi.org/10.1016/j.ress.2022.108525.

[25] A. Shafaei, M. Schmidt, J. Little, A Less Biased Evaluation of Out-of-distribution Sample Detectors, in: BMVC, 2019.

[26] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural networks, in: Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37, 2015, pp. 1613–1622.

[27] P. Myshkov, S. Julier, Posterior distribution analysis for bayesian inference in neural networks, in: Workshop on Bayesian Deep Learning, NIPS, 2016.

[28] S. Haykin, Neural Networks and Learning Machines, Pearson Education, 2009.

[29] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (1997) 1735–1780. URL: https://doi.org/10.1162/neco.1997.9.8.1735. doi:10.1162/neco.1997.9.8.1735. arXiv:https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf.

[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information

Processing Systems, 2019, pp. 8024–8035.

[31] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, S. Nahavandi, A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Information Fusion 76 (2021) 243–297. URL: https://www.sciencedirect.com/science/article/pii/S1566253521001081. doi:https://doi.org/10.1016/j.inffus.2021.05.008.

[32] Y. Gal, Uncertainty in Deep Learning, Ph.D. thesis, University of Cambridge, 2016.

[33] R. M. Neal, Bayesian learning for neural networks, Ph.D. thesis, University of Toronto, 1995.

[34] D. A. Nix, A. S. Weigend, Estimating the mean and variance of the target probability distribution, in: Proceedings of 1994 ieee international conference on neural networks (ICNN'94), volume 1, IEEE, 1994, pp. 55–60.

[35] L. V. JOSPIN, W. BUNTINE, F. BOUSSAID, H. LAGA, M. BENNAMOUN, Hands-on bayesian neural networks-a tutorial for deep learning users, ACM Comput. Surv 1 (2020).

[36] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: NIPS, 2017, pp. 6405–6416. URL: http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.

[37] M. Welling, Y. W. Teh, Bayesian learning via stochastic gradient langevin dynamics, in: Proceedings of the 28th international conference on machine learning (ICML-11), 2011, pp. 681–688.

[38] C. Li, C. Chen, D. Carlson, L. Carin, Preconditioned stochastic gradient langevin dynamics for deep neural networks, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), 2016, pp. 1788–1794. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11835/11805.

[39] T. Chen, E. Fox, C. Guestrin, Stochastic gradient hamiltonian monte carlo, in: International conference on machine learning, PMLR, 2014, pp. 1683–1691.

[40] S. Hernández, D. Vergara, M. Valdenegro-Toro, F. Jorquera, Improving predictive uncertainty estimation using dropout–hamiltonian monte carlo, Soft Computing 24 (2020) 4307–4322.

[41] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning, 2016, pp. 1050–1059.

[42] S. Bachstein, Uncertainty Quantification in Deep Learning, Master's thesis, Universität Ulm, 2019.

[43] D. Zhang, L. Lu, L. Guo, G. E. Karniadakis, Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems, Journal of Computational Physics 397 (2019) 108850.

[44] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision?, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/2650d6089a6d640c5e85b2b88265dc2b-Paper.pdf.

[45] S. Depeweg, J.-M. Hernandez-Lobato, F. Doshi-Velez, S. Udluft, Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 1184–1193. URL: http://proceedings.mlr.press/v80/depeweg18a.html.

[46] A. P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern recognition 30 (1997) 1145–1159.

[47] J. A. Hanley, B. J. McNeil, The meaning and use of the area under a receiver operating characteristic (roc) curve., Radiology 143 (1982) 29–36.

[48] T. Fawcett, An introduction to roc analysis, Pattern recognition letters 27 (2006) 861–874.

[49] J. Lane, J. Link, J. King, T. George, S. Claybrook, Benchmark of gothic to ebr-ii shrt-17 and shrt-45r tests, Nuclear Technology (2020) 1–17.

[50] Y. Cai, B. Starly, P. Cohen, Y.-S. Lee, Sensor data and information fusion to construct digital-twins virtual machine tools for cyber-physical manufacturing, Procedia manufacturing 10 (2017) 1031–1042.

[51] S. M. Cetiner, M. D. Muhlheim, A. Guler Yigitoglu, R. R. J. Belles, M. S. Greenwood, T. J. Harrison, R. S. Denning, C. A. Bonebrake, G. Dib, D. Grabaskas, et al., Supervisory control system for multi-modular advanced reactors, Technical Report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States); Pacific . . . , 2016.

[52] J. H. Lee, A. Yilmaz, R. Denning, T. Aldemir, Use of dynamic event trees and deep learning for real-time emergency planning in power plant operation, Nuclear Technology (2018).

[53] T. S. Sumner, T. Y. Wei, Benchmark specifications and data requirements for EBR-II shutdown heat removal tests SHRT-17 and SHRT-45R, Technical Report, Argonne National Lab.(ANL), Argonne, IL (United States), 2012.

[54] EPRI, Gothic thermal hydraulic analysis package, version 8.3 (qa), 2019.

Appendix A. Nearly autonomous management and control system

The Nearly Autonomous Management and Control (NAMAC) system refers to a system developed to output recommendations based on human knowledge, digital twins (DTs), and specific operational workflow [7]. In the NAMAC system, DTs are essential components used to support the NAMAC functions. DTs are developed based on the knowledge base using machine learning (ML) algorithms. The key DTs in the NAMAC system include:

- DT for Diagnosis (DT-D): The DT-D is used to evaluate the complete states of a physical system. In a physical system, some information, such as the states of unobservable variables and fault conditions, cannot be directly measured. Data-driven methods, such as artificial neural networks and support vector machine, are becoming popular in the DT-D development. Those methods are applied to capture the relationship between correlated variables such that the complete states can be restored. Figure A.15 shows the scheme of data-driven DT-D model used in the NAMAC system.

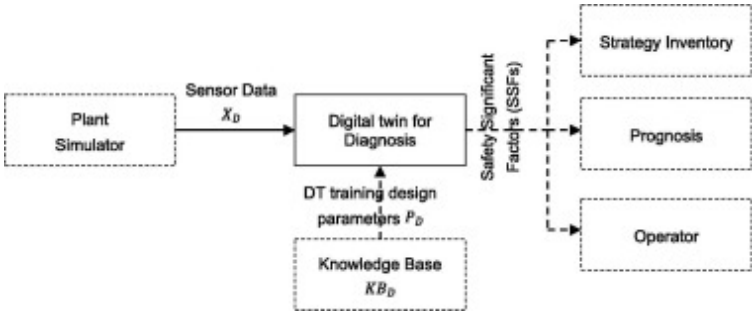


Figure A.15: Scheme of data-driven DT-D model [7]

- DT for Strategy Inventory (DT-SI): The major objective of DT-SI is to provide a list of available actions based on the diagnosed system states and the design limits of the system. Many approaches can be used to implement DT-SI, such as grid-based algorithms and artificial intelligence declarative approaches.

- DT for Prognosis (DT-P): The DT-P is developed to provide forecasts of system states based on historic information and current states. Data analytical models [50], model-based methods combined with probabilistic risk analysis framework [51] are implemented in similar

applications. Data-driven models are now gaining popularity because of their high accuracy and low computational cost [52]. In the developed NAMAC system, NNs are used to develop the DT-P. Figure A.16 depicts the scheme of data-driven DT-P model used in the NAMAC system.

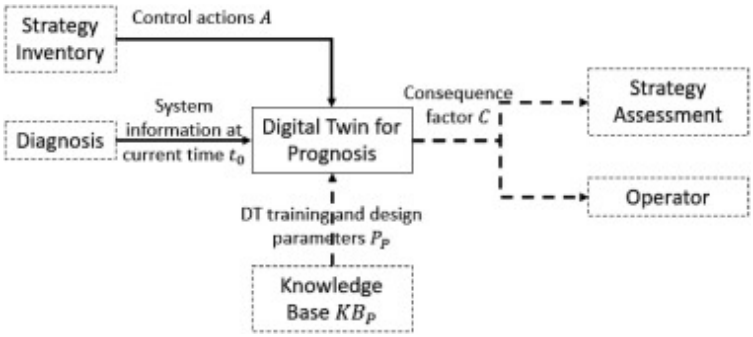


Figure A.16: Scheme of data-driven DT-P model [7]

- DT for Strategy Assessment (DT-SA): The DT-SA ranks the values of control actions based on the possible consequences predicted by the DT-P. In addition to the future states, preference structure is also input to the DT-SA.

The NAMAC system has been demonstrated based on the Experimental Breeder Reactor II (EBR-II) automatic controls during a single loss-of-flow accident scenario. The corresponding operational workflow of the NAMAC system is shown in Figure A.17. In the demonstration, a GOTHIC plant simulator models the EBR-II reactor. The plant simulator is used to simulate the reactor behaviors during both normal and accidental conditions, which will be used as training and testing data to develop the DTs.

The sensory data are collected from the reactor and utilized by the DT-D. Processing the observable states, the DT-D recovers the complete conditions of the reactor. The unobservable safety-significant factors estimated by the DT-D are then used by:

1) DT-P to forecast the future of reactor

2) DT-SI to list the available actions

3) discrepancy checker for continuous monitoring.

The complete reactor states restored by DT-D and available control actions identified by DT-SI are fed to DT-P. The DT-P can then predict the future reactor states over a certain time range. Based on the DT-P outputs, the optimal strategies can be determined by DT-SA.

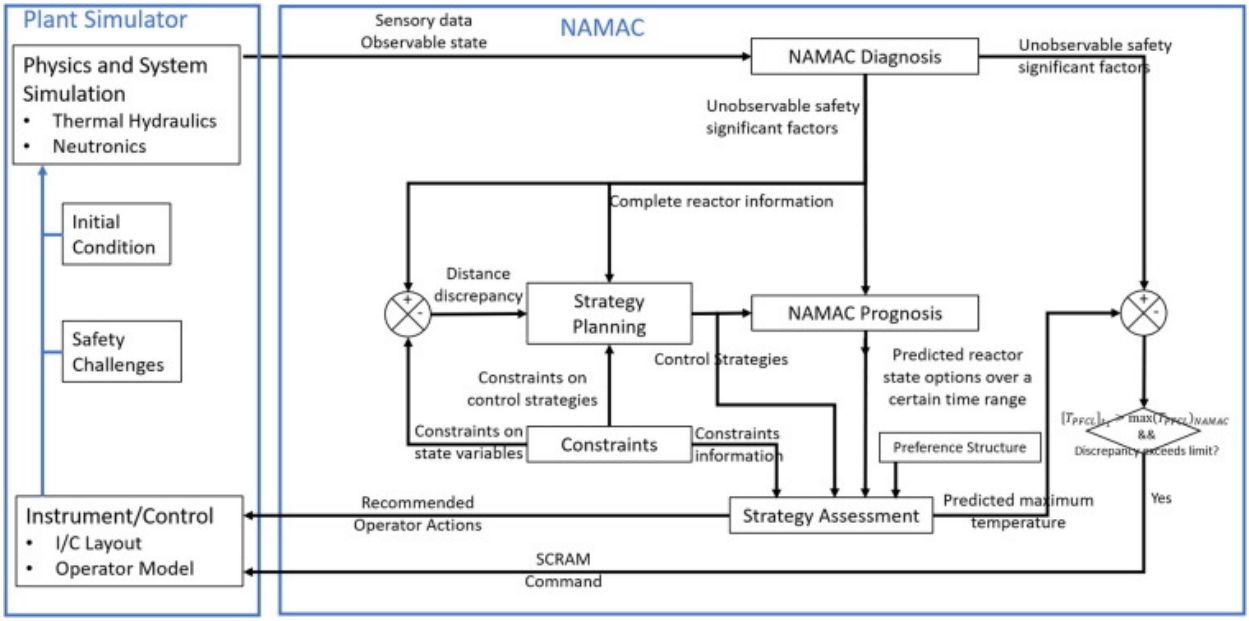


Figure A.17: Operational workflow of the NAMAC system [7]

In the operational workflow, DT-D is one of the essential DTs, since it estimates the unobservable safety-significant factors to restore the complete reactor states. DT-P is another essential component, its outputs are used to determine the recommended operator actions that may be adopted to control the plant. A SCRAM signal may be generated if no operator action satisfying the safety limits can be recommended or the discrepancy between the DT-D and DT-P predictions exceeds the limit. If the reliability of DT-D and DT-P is not accurately assessed, the reactor may fail to SCRAM under accidental conditions, and severe accidents may be caused by the harmful actions recommended. Therefore, the trustworthiness of these DTs should be properly evaluated.

NNs are now gaining popularity due to their impressive regression power, so they have been selected to develop DT-D and DT-P models. In this demonstration, both the DT-D and DT-P are developed using feedforward neural networks [7]. Considering that the data used in the demonstration come from transient episodes, recurrent neural networks (RNNs), which are more suitable to deal with time-series data, are used to develop the DT-D model[8]. Since the development of a

traditional RNN does not guarantee the conservation of physics, a physics-guided RNN improves the DT-P performance [10]. To evaluate the trustworthiness of those NN-based DTs, DCA should be integrated into the NAMAC system.

Appendix B. EBR-II transient scenarios and GOTHIC simulations

The EBR-II was a fast reactor located in Idaho and operated by Argonne National Laboratory [53]. It was initially used to demonstrate a breeder reactor requiring only U-238. The primary coolant system of EBR-II is illustrated in Figure B.18. The sodium in the primary tank is transferred to the reactor by two primary sodium pumps (PSPs). The sodium then flows through the inner core and extended core regions from the high-pressure inlet plenum. The hot sodium is mixed in the upper plenum and fed into the intermediate heat exchanger (IHX) through the Z-pipe. An auxiliary electromagnetic (EM) pump is equipped in the Z-pipe. The sodium will flow back to the primary pool after exiting the IHX and will be drawn by PSPs again. The IHX will transfer the core heat to the intermediate loop and finally to the steam generators using the sodium in the intermediate loop.

In the NAMAC system, the behaviors of EBR-II in this work are simulated by GOTHIC [54]. GOTHIC is a thermal-hydraulic analysis tool bridging the gap between system-level thermal-hydraulic codes and computational fluid dynamics codes [49]. The simplified EBR-II model constructed using GOTHIC is depicted in Figure B.19. The GOTHIC EBR-II model has been benchmarked to SHRT-17 and SHRT-45R tests. It is demonstrated that GOTHIC simulations and experimental results agree well for the reactor power, reactivity feedback, flow rate for primary pumps, and temperatures of core outlet and Z-pipe inlet [49].

The NAMAC system is demonstrated based on case studies designed for a single loss-of-flow accident of EBR-II. The reactor is initially operating at nominal power, and the two PSPs work with nominal rotational speed. The accident transients are triggered by PSP #1 (PSP1) coasting down. The NAMAC system finds available control actions and recommends the best one by predicting the consequences of each action. The available control actions considered in the NAMAC demonstration are increasing the rotational speed of PSP #2 (PSP2).

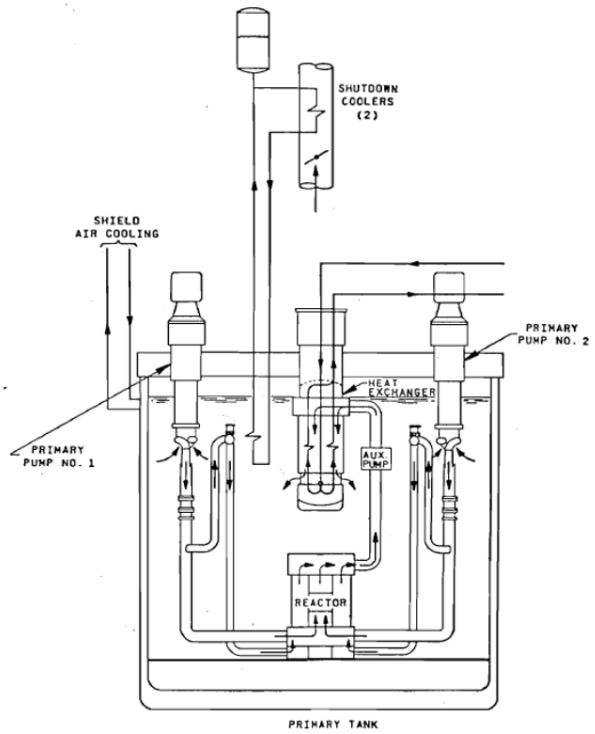


Figure B.18: EBR-II primary tank [53]

In this work, three databases are used. Table B.11 summarizes the characteristics of these databases where the pump speeds are represented in terms of the nominal values. The episodes in databases "aa01" and "ba01" are used in the Q4 NAMAC demonstration. They are generated using the "simp007" model. The "simp016" model provides data used in the Q8 NAMAC demonstration, which are referred as the Q8 data base and are used in Case Study 2.

For the episodes in database "ba01", PSP1 ramps down from the nominal state, and the rotational speed of PSP2 is not increased to mitigate the accidents. The episodes will be characterized by the final PSP1 speed and the ramping time. 32 different PSP1 final speeds and 32 different PSP1 ramping times are randomly sampled from the given range. In total, 1,024 episodes are generated using GOTHIC.

For the episodes in database "aa01," the PSP1 ramps down to initialize the accidents, and the rotational speed of PSP2 will be increased to the designated value if the trigger temperature is reached. The ramp-down behaviors of PSP1 are the same for all the episodes, and the episodes vary in terms of the trigger temperature and PSP2 final speed. 1,024 episodes are simulated by

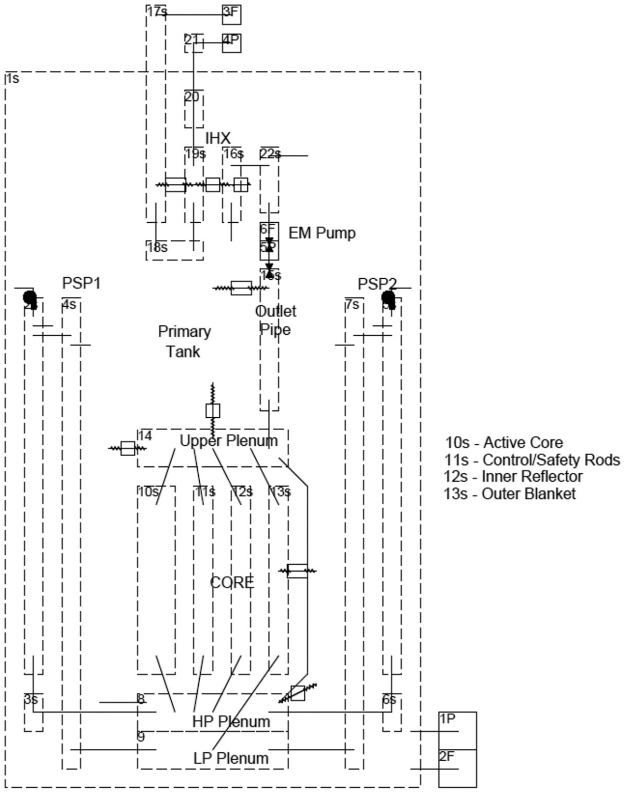


Figure B.19: Simplified GOTHIC EBR-II model [49]

randomly sampling the trigger temperature and PSP2 final speed.

The episodes in database "Q8" start from PSP1 ramping down within a fixed time. The four characteristics, PSP1 final speed, PSP2 final speed, PSP2 ramping up trigger time, and EM pump status, are randomly sampled from the given ranges. 5,000 episodes are generated for database "Q8."

Table B.11: Descriptions of databases used

Database	EBR-II Model	Scenario Description	Manipulated Variables
ba01	simp007	PSP1 ramps down	PSP1 final speed [0%, 100%]
		PSP2 keeps constant	PSP1 ramping time [0 sec, 500 sec]
aa01	simp007	PSP1 ramps down	PSP2 final speed [100%, 150%]
		PSP2 ramps up	Trigger temperature [600°C, 700°C]
Q8	simp016	PSP1 ramps down	PSP1 final speed [0%, 100%]
		PSP2 ramps up	PSP2 final speed [100%, 150%]
			Trigger time [50 sec, 100 sec]
			EM pump status (On or Off)