Self- and Domain-Awareness for Machine Learning in NDE and SHM

MOHAMMAD ALI FAKIH, PAUL D. WILCOX,
ANTHONY CROXFORD and SERGIO CANTERO-CHINCHILLA

ABSTRACT

Given any input, a machine learning (ML) algorithm will produce an output, but the confidence and accuracy of that output are currently not well understood or quantified. This poses serious risks in the case of automated safety-critical nondestructive evaluation (NDE) or structural health monitoring (SHM) applications. This work aims to provide a comprehensive approach for quantifying the uncertainty of ML models when used for inspection purposes (self-awareness), in addition to equipping these models with domain-awareness features (i.e., the ability to detect out-of-distribution or "bad" input data). An ensemble of models is trained as part of the procedure, and the reliability of the predictions is measured using the statistics of the models. This information is also used to assess the quality of the input data and evaluate whether it comes from an out-of-distribution source or falls within the domain that the models were trained on. However, this process encompasses making a good choice of the ML model's type and complexity to ensure model effectiveness and stability. It also involves understanding the trade-off between the ensemble size (number of models within the ensemble) and the prediction's uncertainty and statistical significance. A robust framework for the whole process is provided and discussed in this paper. The methodology is demonstrated using simulated ultrasound data for corrosion-profile assessment. The presented findings are a step forward towards having more confidence in using ML for NDE and SHM applications. Some current shortcomings are discussed and suggested for future investigation.

INTRODUCTION

The need for automated NDE is growing stronger with the great advancements in materials and production techniques. Machine learning has the potential to increase the NDE capacity and capability by both improving inspection accuracy and facilitating process automation by reducing the need for labour-intensive expert intervention. Such intervention is usually accompanied by high pressure on technical experts, longer inspection times, and therefore an increased probability of human error.

School of Electrical, Electronic and Mechanical Engineering, University of Bristol, Bristol, BS8 1TR, UK

However, ML cannot be used for the safety-critical NDE applications without:

1. self-awareness: having measures of how certain the predictions are (uncertainty quantification); and
2. domain-awareness: making sure the ML model knows if it is familiar with the measured data, i.e., data is in-distribution (InD), or is seeing something new, i.e., data is out-of-distribution (OOD).

Some authors of this paper have recently published a comprehensive literature review of deep learning applications in automated NDE [1]. In conjunction with several industrial partners, the authors outlined the fundamental requirements for using ML in NDE applications through a set of axioms. Further, Pyle et al. [2] have demonstrated that uncertainty quantification (UQ) can be achieved for deep learning models in the context of crack sizing for inline-pipe inspection. Two UQ approaches were applied and compared, namely, deep ensembles and Monte Carlo dropout. A technique was considered "well calibrated" if the prediction error is proportional to the estimated uncertainty for InD data, while "anomaly detection" was deemed successful if high uncertainty is assigned to OOD data. The results proved the superiority of the deep ensemble technique over the Monte Carlo dropout.

Conceptual simplicity and ease of implementation are two advantages of the deep-ensemble technique. However, there is a challenge in choosing (1) the hyperparameters of the individual deep-learning networks (model complexity) and (2) the number of models needed to ensure sufficient diversity, both while maintaining stable and statistically significant predictions. In line with the mentioned axioms and challenges, this paper proposes a complete framework for UQ and domain awareness of ML models when used for NDE/SHM purposes.

METHODOLOGY

The suggested solution is to train a big enough number of identical models (with different initialisations), called hereafter an ensemble, and rely on the statistics of the ensemble to get the desired self-awareness and domain-awareness features. Figure 1 shows the flowchart of the proposed methodology. The ensemble development process is all centred around the prior knowledge of the specific problem needs and the data available for training. This knowledge allows the ensemble designer to set prediction-precision criteria (PPC), which define the acceptable margin of prediction error for later ensemble-development stages (e.g., choosing appropriate model type/complexity and ensemble size).

A proper model type and general structure are chosen to ensure that efficient models are used within the ensemble. This is highly dependent on the data type (e.g., sensor signals, ultrasound/other images, etc.), data shape (e.g., signal length or image size), and data labels (e.g., known assessments corresponding to each sensor measurement). After these choices, interactive steps to decide the needed model complexity and number of models (ensemble size) are performed. These steps are computationally expensive since they require training and testing large numbers of models of various complexities. It is important to note that once such choices are made, they can be directly applied to similar problems without going through the same process (e.g., sensor data of a similar nature). Sanity checks regarding the validity of the ensemble design are then recommended.

The ensemble operation involves employing the trained models for prediction, UQ, and deciding the acceptance/rejection of the input data. This process does not require

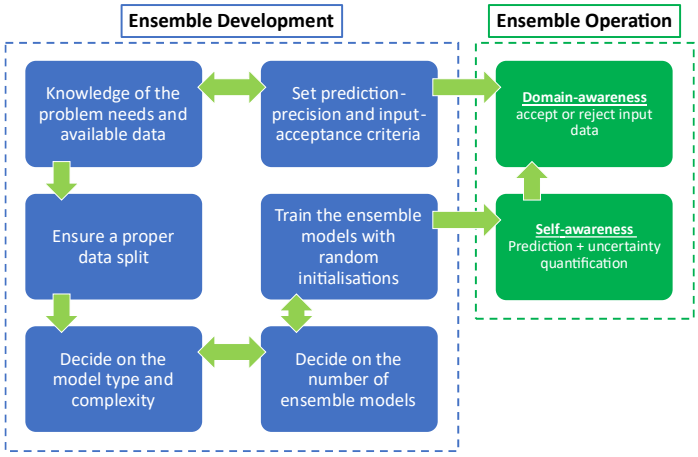

Figure 1. The overall flowchart of the proposed framework to develop and operate a self-aware and domain-aware ML ensemble suitable for NDE/SHM predictions.

high computational resources. The input-acceptance criteria (IAC) should be based on the quantified uncertainty and compliant with the preset PPC. These steps will be detailed in the subsequent sections using an example case study.

PROBLEM DESCRIPTION

The dataset used in this paper is described in a previous study [3]. The study used 1D convolutional neural networks (CNNs) to predict the thickness values of corrosion profiles using simulated ultrasonic A-scan measurements. It was demonstrated in [3] that the ML solution outperformed conventional ultrasound testing (UT) techniques for thickness evaluation. Part of this synthetic dataset is used in the current study, with a total of 12,415 cases of ultrasound responses from various corrosion profiles. The cases were varied by changing the roughness amplitude expressed as a root mean square (RMS) value, roughness correlation length, and mean thickness or depth (D).

Prediction-precision criteria should be set based on the inspection requirements of the problem at hand. The following criteria were specified for this study: the mean-thickness prediction error should be within ±0.15 mm over the mean thickness range of 5-16 mm (PPC: prediction error $\leq$ 0.15 mm).

ENSEMBLE SELF-AWARENESS

The rationale of the proposed approach is to train an ensemble of identical deep-learning networks (same architecture and hyperparameters) but with different random initialisation conditions, hereafter called deep ensemble (DE). If the complexity of the networks is good enough, the individual networks will learn distinct routes to map their inputs to their outputs. Figure 2 shows an example of the distribution of individual-network predictions in a DE, for an inspected parameter of interest (mean thickness D in this case). The overall prediction of the DE is defined as the mean of the predictions of all the M individual networks (μ_M). The uncertainty of the DE's prediction (U_M) can be related to a statistical measure of the variation in the individual predictions, which is

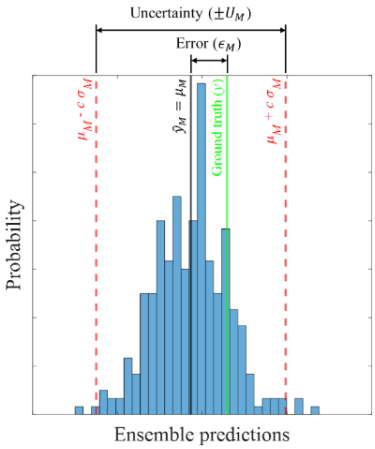

Figure 2. Example ensemble predictions of a single value (e.g., an inspected parameter of interest) to visualise the concept of the ensemble's overall prediction (μ_M), statistically defined uncertainty ($\pm c\ \sigma_M$), and actual prediction error.

generally a value related to their standard deviation (σ_M). The following equations provide definitions of the DE prediction, error, and uncertainty:

$$\hat{y}_M = \mu_M = \frac{1}{M}\sum_{m=1}^{M} \hat{y}_m; \tag{1}$$

$$\epsilon_M = \hat{y}_M - y; \tag{2}$$

$$U_M = c\ \sigma_M = c\sqrt{\frac{\sum_{m=1}^{M}(\hat{y}_m - \mu_M)^2}{M}}; \tag{3}$$

where M is the total number of models in the ensemble DE_M; $\hat{y}_M$ is the value predicted by the ensemble constituted of M single models; ϵ_M is the ensemble's prediction error; y is the ground truth of the predicted parameter; $\hat{y}_m$ is the value predicted by a single model "m" (the m^{th} member of the ensemble DE_M); U_M is the uncertainty of the ensemble prediction, where the ensemble is constituted of M single models; σ_M is the standard deviation of the M predictions by DE_M; and c is a multiplier based on the statistical relationship between σ_M and ϵ_M inferred using the validation dataset.

MODEL TYPE AND COMPLEXITY

Considering a typical deep-learning 1D CNN model (as the one that was used in the previous study [3]), there are several factors that would affect the complexity of the model. Along with other hyperparameters, the complexity in the model's architecture, for example, is dictated by the number and type of layers, kernel size, and number of filters. Since an optimisation of the CNN model was previously performed [3], the same general architecture of the CNN was adopted in this work: (i) a number of 1D convolutional layers followed by batch normalisation, (ii) a 20% dropout layer, (iii) a flattening layer, then (iv) a single output neuron. The models were created using a model-building algorithm for 1D CNNs (www.github.com/casimp/undt-ai) implemented in TensorFlow using the Keras functional API. For practical reasons, all

the hyperparameters were fixed except for two parameters, the number of convolutional layers (N_L) and the kernel size (k). The number of trainable parameters (N_{trP}) was used as a single-number proxy to measure the complexity of the model.

The ReLU activation function was used for the convolutional layers, while linear activation was used for the output layer. The Adam optimiser and the mean-squared error (MSE) loss function were employed to train the CNNs (with a batch size of 128). To prevent overfitting and ensure all the models are sufficiently trained, a patience of 350 was set as a stopping criterion. The patience value is case-specific and was chosen based on a separate study, which is not included here for brevity.

To study the effect of model complexity, an ensemble of 250 models was trained for 16 different model architectures. All the models were trained using the same training and validation datasets, but using different random initialisations. The performance of the ensembles was then evaluated over two distinct and unseen testing datasets, separately (Testing data # 1 and # 2) and when both are combined (all testing data). It should be noted here that all the testing examples in this section are InD data (in the sense that each parameter of the testing dataset lies within the range of that parameter's values in the training dataset). The obtained results are shown in Figure 3. The model architectures can be seen as blue labels on the top ticks of the horizontal axis, represented as $N_L \times k$ (e.g., 1×31 means $N_L = 1$ and $k = 31$).

Since only one parameter is being predicted in this problem, the root-mean-squared error (RMSE) of the predictions of a single model (m) for the whole testing dataset can be calculated. The ensemble mean and standard deviation of this mean error are then calculated.

The uncertainty shades of μ in Figure 3 are based on statistical theory to estimate the actual μ from a population of 250 models (μ_{250} and σ_{250}). Although both are based on common statistical grounds, this statistical uncertainty is different from the suggested uncertainty quantification of a deep ensemble explained in the previous section. Specifically, the relationship between the estimated and true mean of an ensemble is given in Equation (4) [4]:

$$\hat{\mu}_M = \mu \pm \frac{\sigma}{\sqrt{M}} \approx \mu \pm \frac{\hat{\sigma}_M}{\sqrt{M}}; \tag{4}$$

where $\hat{\mu}_M$ and $\hat{\sigma}_M$ are, respectively, the estimated (calculated) mean and standard deviation from an ensemble of M models; and μ and σ are, respectively, the (unknown) true mean and standard deviation of the infinitely large population of all possible ML models that could have been trained from the available data. In the current problem, the true value of the standard deviation is unknown; hence, the best value of σ to be used in the estimated uncertainty was $\hat{\sigma}_M$.

It was observed that both the error and standard deviation initially decrease as the model complexity increases. The models' behaviour becomes chaotic for specific model architectures of higher complexity. The region of chaotic behaviour was identified by two features: (1) a sudden and pronounced increase in the ensemble's error and standard deviation, and (2) significantly different performance across two testing datasets.

Based on the results shown in Figure 3, it was decided to use a model architecture of $N_L = 2$ and $k = 91$. This architecture has shown the best performance before reaching the chaotic region. The performance of the chosen model type and architecture should obey the preset PPC, which is true in this case ($\mu_{M,\,\mathrm{RMSE}} < 0.15$ mm).

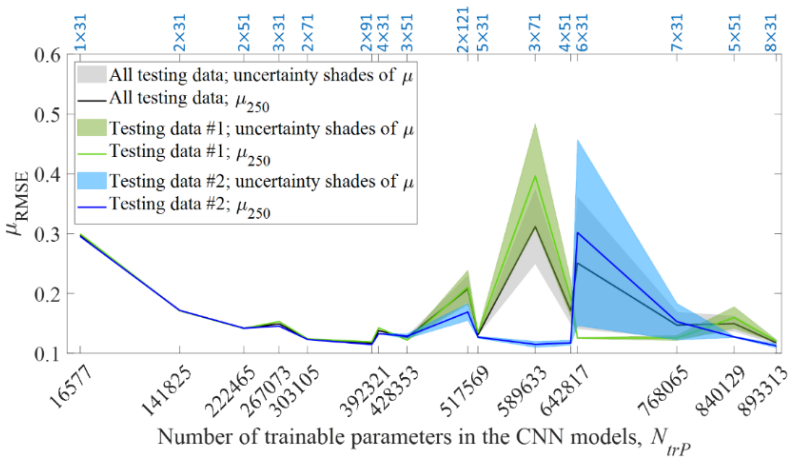

Figure 3. Performance of the DEs of various complexities (CNN architectures), measured using the number of trainable parameters.

ENSEMBLE SIZE

From a statistical point of view, the larger the number of models used, the better the estimation is. However, given the computational burden and resource consumption, it is vital to define a good-enough ensemble size M based on the design criteria.

Figure 4 shows the evolution of the mean absolute error of the DE ($\mu_{M,\mathrm{RMSE}}$) over three testing datasets (Test 1, Test 2, and OOD test) as M increases. The variation among the models is shown using the $\sigma_{M,\mathrm{RMSE}}$ shades. It is noticed that the prediction error plateaus after a DE size of 50 to 100 models; however, the error does not converge to zero as M increases. The prediction accuracy will always be limited by the available training data (amount and quality), in addition to the nature of the problem at hand. According to Figure 4b, an ensemble size $M \geq 14$ models agrees with the PPC ($\mu_{M,\mathrm{RMSE}} < 0.15$ mm). In this paper, it was preferred to use large ensemble sizes ($M \geq 250$ models) to observe other effects without concerns about the ensemble size.

Despite the difference in behaviour for two nominally identical InD testing datasets (Figure 4b), a test on the OOD dataset (Figure 4a) shows significantly higher values of the ensemble's mean error and standard deviation. This proposes that the discrepancy seen over the two InD datasets should not cause a problem in detecting OOD data. This will be further discussed in the next section.

ENSEMBLE DOMAIN-AWARENESS

One dangerous aspect of conventional machine/deep learning models is that a model can provide the user with a prediction no matter what input it is given, as long as it fits the expected input shape (e.g., signal length or image size and format). Hence, one of the most important features of a convenient ML solution for NDE is to be a domain-aware technique.

Figure 4b shows that the deviation among model predictions significantly increases when introducing OOD data to the ensemble, demonstrating that the standard deviation can provide a tool to estimate the accuracy and validity of the prediction. Figure 5a

shows a scatter plot of all the available data in terms of two corrosion-profile parameters, namely, RMS and D (blue: InD; and grey: OOD).

The prediction errors and ensemble standard deviation of both the InD and OOD datasets are presented in Figure 5b-c. The results demonstrate significantly high errors and uncertainties for the OOD predictions when far enough from the InD data ranges. Since one does not have access to the ground truth in practice, the error would not be available for examination. Hence, the input-acceptance criteria (IAC) should be set over accessible prediction parameters like σ_M or σ_M/μ_M. In the current stage, a trial was made to set visual thresholds for σ_M to accept or reject the input data (check Figure 5d). It can be said, for this specific case, that a threshold of $\sigma_M < 0.1$ mm is a good data-acceptance criterion which does not disagree with the PPC.

In practice, the DE designer is supposed to train using all the available data to avoid limiting the range of applicability. This means there would not be a set of OOD data to check the validity of the chosen IAC. Therefore, a robust way of setting the IAC should be established.

CONCLUSION

This paper presented a comprehensive methodology to develop a self-aware and domain-aware machine-learning (ML) solution to fulfil the safety-critical needs of nondestructive evaluation and structural health monitoring. The proposed solution involves training an ensemble of identical ML models using different initialisation conditions. This allows taking the mean of the ensemble outcomes as the final prediction, and the standard deviation as a measure to quantify the uncertainty. The ensemble statistics are also used to determine the goodness of the input data and decide whether it is within the domain that the models were trained on. This procedure requires a wise selection of the ML model's complexity within a hyperparameter range that guarantees model stability. Another important aspect is to acquire an understanding of the number of models required in the ensemble to ensure robustness and strong statistical significance. Despite the strengths of the developed framework, further work is still needed to enhance its robustness and establish well-defined practices for both ensemble development and operation.

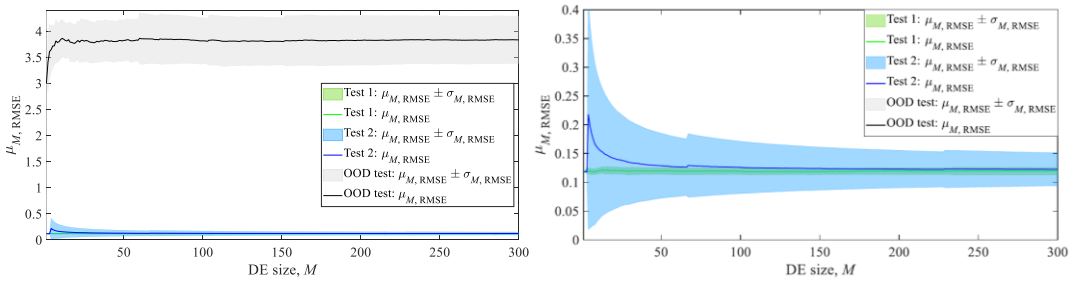

Figure 4. Convergence, over three distinct testing datasets (one of which is out of the training distribution), as the size of the ensemble increases: (a) prediction accuracy (mean absolute error); and (b) a zoom-in to Test 1 and 2.

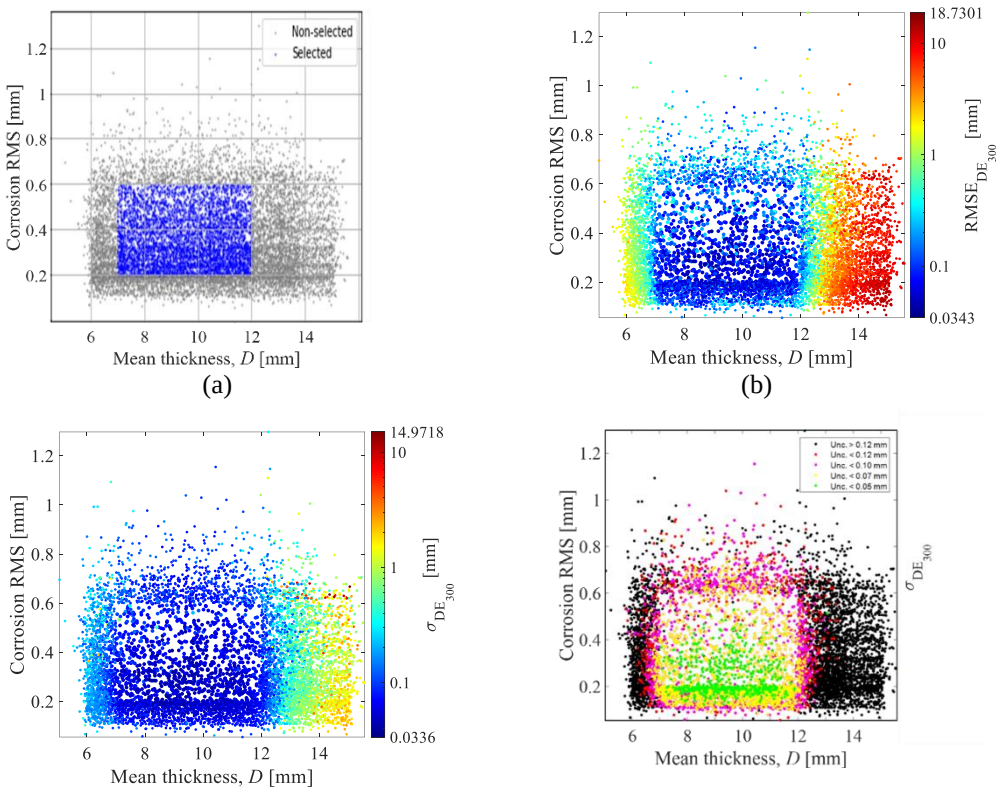

Figure 5. Scatter plots of the selected InD and OOD testing datasets: (a) **blue:** in-distribution data examples selected for ensemble development; **grey:** OOD distribution data examples; (b) ensemble RMSE error ($\mathrm{RMSE}_{\mathrm{DE}_{300}}$) calculated for each data example, using the predictions of 300 DE models; (c) ensemble standard deviation ($\sigma_{\mathrm{DE}_{300}}$) calculated for each data example; and (d) $\sigma_{\mathrm{DE}_{300}}$ thresholds set visually to accept or reject the input data.

ACKNOWLEDGEMENTS

This work was funded by the Core Research Programme of the UK Research Centre for Non-Destructive Evaluation (RCNDE). The work was partly carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol – http://www.bris.ac.uk/acrc/.

REFERENCES

1. Cantero-Chinchilla, S., P.D. Wilcox, and A.J. Croxford, *Deep learning in automated ultrasonic NDE–developments, axioms and opportunities.* NDT & E International, 2022. **131**: p. 102703.
2. Pyle, R.J., et al., *Uncertainty quantification for deep learning in ultrasonic crack characterization.* IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 2022. **69**(7): p. 2339-2351.
3. Cantero-Chinchilla, S., et al., *Convolutional neural networks for ultrasound corrosion profile time series regression.* NDT & E International, 2023. **133**: p. 102756.
4. Snedecor, G.W. and W.G. Cochran, *Statistical methods.* 8th ed. 1989, Ames: Iowa State University Press.