NDT&E International 133 (2023) 102756

Contents lists available at ScienceDirect

NDT and E International

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

Convolutional neural networks for ultrasound corrosion profile time series regression

Sergio Cantero-Chinchilla [a,*], Christopher A. Simpson [a], Alexander Ballisat [b], Anthony J. Croxford [a], Paul D. Wilcox [a]

[a] *Department of Mechanical Engineering, University of Bristol, Bristol, BS8 1TR, UK*
[b] *Centre For Modelling & Simulation, Bristol, BS16 7FR, UK*

ARTICLE INFO

Keywords:
Non-destructive evaluation
Deep learning
Ultrasound
Convolutional neural networks
Corrosion
Wall thickness

ABSTRACT

A customised convolutional neural network (CNN) architecture is proposed in this paper to make estimations about the thickness values (minimum and mean) of corroded profiles from an ultrasonic time-series measurement (A-scan). The CNN architecture is determined after a hyper-parameter optimisation which leads to the best performing network. The model is trained using synthetic data and tested on both synthetic and experimental datasets. A comparison is made with (1) a state-of-the-art network for time series analysis and (2) conventional thickness estimation techniques such as peak envelope and threshold crossing. The proposed network provides an accurate estimation of the thickness values and outperforms the conventional techniques in both synthetic and experimental datasets. When compared to a conventional threshold-crossing technique for minimum thickness prediction, the proposed network is more consistent and less sensitive to changes in the threshold.

1. Introduction

In much safety critical infrastructure, monitoring material thickness is crucial. This is the case for the oil & gas sector, where replacement of components is determined by the remaining wall thickness. Typically, such assessments are made using periodic measurements with ultrasonic transducers [1]. The thickness of the part is then extracted using signal processing approaches such as the arrival time of the first reflection or the time that a certain amplitude is first crossed. In such approaches there is little knowledge of the nature of the surface and different corrosion mechanisms result in huge variations in the roughness. This variability in roughness causes significant uncertainty in the ultrasonically measured thickness and, ultimately, conservative approaches to replacement of components are used. More accurate and reliable thickness data at measurement locations would allow operators to make better informed decisions, potentially lowering uncertainty and costs. Deep learning [2] has significant potential to reduce thickness prediction errors. Most existing signal processing approaches do not address subtle features in an ultrasonic response, and simply perform a linear mapping of a single feature, arrival time, to thickness. Deep learning approaches can take advantage of complex signal information and indeed are specifically designed to do so. Historically, development has been focused on 2D domains such as images [3,4]. Notwithstanding this, there have been recent advances in the field of deep learning for time series analysis [5,6]. These have typically been applied to the problems of time series classification, for instance in automatic classification of heart rate irregularities [7], and have resulted in improvements in accuracy and computational time.

There are a range of potential deep learning approaches for the analysis of ultrasonic time series traces from corroded surfaces. These techniques can be split into discriminative and generative approaches [8]. *Generative* or model-based approaches leverage an unsupervised pre-training stage to generate representations of the time series [9], which can then be fed into a classifier or regression model. This can be compared to a *discriminative* approach which directly maps the inputs to the output variables. These inputs can either be features manually extracted from the time series or the raw time series itself. Discriminative approaches are often termed an end-to-end approach, as they include feature selection and optimisation as part of the learning process. A discriminative model has the advantage of generally being simpler to implement than a generative approach and potentially more accurate [10]. Furthermore, an end-to-end discriminative approach is typically less subject to human bias and enables the network to learn the most discriminative features for the task. Recent advancements have shown that a general purpose convolutional neural network (CNN) with a linear regression top layer can compete with problem-specific methods (e.g. VGG-16 or ResNet-50) [11].

[*] Corresponding author.
E-mail address: sergio.canterochinchilla@bristol.ac.uk (S. Cantero-Chinchilla).

https://doi.org/10.1016/j.ndteint.2022.102756
Received 16 May 2022; Received in revised form 13 September 2022; Accepted 8 October 2022
Available online 14 October 2022

In each layer of a CNN, one or more kernels are convolved with the input to produce the output. Recent work on 1D CNN architecture for time series classification [5,6,12,13] has highlighted the importance of kernel size on accuracy. For instance, Tang et al. [13] consider the time series data and the model's kernel size in relation to the underlying noise and pertinent signal frequency. They suggest that there is an optimum kernel size, which ensures that there is no fundamental loss of frequency information (overly small kernel) and minimises the amount of noise that is introduced into the convolutional window (overly large kernel). In the novel ROCKET architecture [5], the authors suggest that critical information is often found across multiple time series scales and uses varying kernel architectures (by varying kernel size, dilation and padding) to aggregate multi-scale contextual information without losing resolution. Fawaz et al. [8] investigated the influence of kernel size and network complexity with the results suggesting that an increasing complexity improves model performance. However, performance degrades when the number of parameters are increased to the point at which overfitting occurs. Therefore, a sufficiently complex network either defines or is a prerequisite for high predictive accuracy, particularly in the 1D time-series domain, suggesting that a sensible approach to a simple 1D CNN architecture would be to be build a shallow network that has a large kernel size. The current state-of-the-art architecture for time series analysis is InceptionTime [6], which is based on the Inception-Resnet (v4) 2D CNN classification architecture [14,15]. Notable features include the inception bottleneck for compressed feature representation [14], residual connections to improve deep network accuracy [16] and simultaneously training with three kernel sizes. The latter allows the model to sample information across three different time scales. The final predictions are made on an ensemble of five of these Inception networks (each model given equal weight in final prediction) to help reduce model variance. The high variance associated with an individual network is attributed to the residual connection [16]. While the model performs well on the UCR archive [17] (i.e. the largest publicly available repository of time series datasets), the authors show that some of the more sophisticated features such as the Inception bottleneck and residual connections do not actually help improve the model performance. Questions therefore remain over whether it is necessary to build an overly complex model for the task of time series classification and/or regression.

A number of deep learning works have been developed in the context of NDE, including signal denoising for defect classification [18] and artefact suppression for enhanced ultrasonic imaging results [19] both using autoencoders; damage detection directly from A-Scans [20]; and crack defect characterisation using CNNs [21]. However, uptake of deep learning on NDE is still comparatively low and lags far behind other domains, which is predominantly due to the requirement for large labelled datasets [22] that may be used for training the models. While there are techniques to mitigate this requirement (e.g. transfer learning, regularisation, and data augmentation), it is almost impossible to completely avoid this issue. In practice, thousands of reliably labelled instances of physically-measured data are rarely, if ever, available for NDE problems such as ultrasound corrosion profile analysis. This issue is magnified when these measurements need to be made across a wide, representative parameter space (e.g. corroded surfaces varying in both thickness and roughness). The only way this problem can feasibly be met is via some form of simulation where the precise parameters of the surface roughness can be controlled. Because ultrasonic testing involves wave propagation, accurate simulation via direct numerical simulation (e.g. FE) requires discretisation at subwavelength scale, which leads to extremely large models that have historically been prohibitive. However, the natural year-on-year increase in computational power combined with technical advances mean that simulation at the fidelity and scale required for producing training datasets is becoming tractable. Specifically, the development of extremely efficient code for full 3D time-domain FE simulations carried out on Graphics Processing Units (GPUs) [23] has enabled a step change in the quantity of model runs feasible.

In this paper, we present the first deep learning discriminative approach for ultrasonic thickness estimation in complex corrosion profiles. Note that only the cases where the corrosion is on the far surface relative to the transducer are addressed in this work, e.g. internal corrosion assessment from external measurement. The proposed method is comprised of CNNs whose architecture and hyper-parameters are optimised based on predictive model accuracy. Two models are developed that aim to predict statistical information (i.e. mean and minimum) about the thickness profile under the area of a transducer using ultrasonic time series as input. To mitigate the lack of labelled experimental data, an efficient GPU-based FE model of ultrasonic propagation is used to produce a large dataset consisting of synthetic ultrasound signals associated with a range of roughness and mean thickness parameters. These are used to train, validate and test 1D convolutional networks with the goal of finding the optimal model architecture that achieve the highest accuracy across a synthetic test set and a small subset of experimental measurements made on machined corrosion coupons. Before feeding the signals into the model, they are subject to signal pre-processing consisting of filtering the frequency content using a Gaussian function in the frequency domain. This pre-processing maximises similarity between the synthetic and experimental signals. The models are tested on both synthetic and experimental signals and compared against standard thickness estimation techniques: (1) peak envelope of the first echo for the mean thickness, and (2) threshold crossing for the minimum thickness. The use of the proposed approach for thickness prediction entails important benefits, needing less operator time for interpretation and supporting automation of repetitive human-dependent tasks.

The remainder of this paper is organised as follows: Section 2 describes the physics- and data-based models along with the conventional techniques that are used as a comparison; Section 3 illustrates the optimisation of the model and shows the performance of the optimal model in synthetic and experimental data, and in comparison with conventional methods; Section 4 discusses the model architecture and other data pre-processing details; finally Section 5 provides conclusion remarks.

2. Methodology

The proposed convolutional networks for thickness estimation in corrosion profiles require large model-based datasets. Therefore, the FE modelling technique adopted to generate data is introduced below before describing the convolutional network architectures to be optimised, the data pre-processing technique, and the methods that are used as a comparison.

2.1. Generation of synthetic data

The 3D FE models that provide ultrasonic data are constructed using a domain that is 20 mm by 20 mm square with a variable mean depth D. The top surface of the specimens is flat and parallel to the face of the ultrasonic transducer. The bottom surface profile is randomly generated through the following process: (1) choose a pair of values to describe amplitude of the corrosion roughness σ and correlation length λ; (2) generate a grid of random numbers normally distributed with standard deviation σ and take 2D FFT; (3) generate a Gaussian filter to yield the amplitude of the roughness and correlation length as σ and λ, respectively, and multiply by (2); and (4) take the inverse FFT of (3) to obtain the corrosion profile. The material (carbon steel) properties of all the specimens are: density 8050 kg/m^3, Young's modulus 200 GPa, and Poisson's ratio 0.29. The element size is determined by setting a maximum volume of a tetrahedral mesh to be $(V/15F_c)^3$, where F_c is the central frequency of the specimen and $V = 5980$ m/s is the wave propagation velocity in the material. The time step is set to $0.3l_{\min}/V$

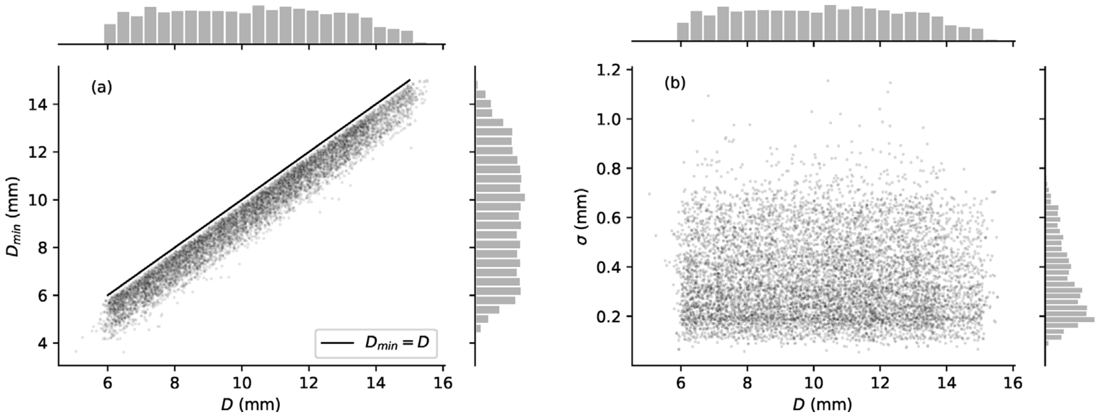


Fig. 1. Synthetic data parameter space in terms of mean thickness (D), with respect to (a) minimum thickness (D_{min}) and (b) amplitude of corrosion parameter (σ).

with l_{min} being the minimum length of an edge in the mesh, which leads to a sampling rate of 24.92 MHz. The transducer aperture is assumed to be either square or rectangular with their two dimensions varying independently to reduce sensitivity of the deep learning model to transducer size.

The parameter space is constituted by seven variables: (1) amplitude of corrosion parameter $\sigma \in [0.2, 0.8]$ mm; (2) correlation length of the corrosion $\lambda \in [0.3, 1.1]$ mm; (3) transducer x size $T_x \in [3, 9]$ mm; (4) transducer y size $T_y \in [3, 9]$ mm; (5) specimen mean depth $D \in [5, 15]$ mm; (6) transducer centre frequency $F_c \in [2, 5]$ MHz; and (7) transducer bandwidth $BW \in [0.5, 2.5]$ MHz. Note that any variation in parameters (1–5) requires a FE model evaluation while variation in (6–7) is performed in post-processing as a broadband signal is used as input load. To reduce the parameter space in the generation of the training datasets, a sensitivity study was run based on Sobol' indices [24]. The results [25] revealed that the most important parameter is the mean thickness D, followed by the frequency properties of the transducer (i.e. F_c and BW). The rest of parameters (i.e. σ, λ, T_x, and T_y) have very low sensitivity indices and therefore they are fixed in the final database to train the deep learning models. An additional sensitivity study was performed to optimise the transducer frequency parameters, which were found to be $F_c = 2.5$ MHz and $BW = 1$ MHz.

A total of 12,416 models were evaluated consisting of 26 different combinations of roughness parameters made by uniformly sampling $D \in [6, 15]$ mm for each (σ, λ) combination. It is important to note that each model is just a single realisation of those statistical parameters. Thus any two models with the same roughness parameters will have different surfaces. This means that the mean and minimum thickness parameters required to properly label ultrasonic data for model training must be calculated explicitly in the area directly below the sensor for each realisation. Thus the area under the aperture is assessed for each corrosion surface and aperture combination. The distribution of corrosion profile parameters under the aperture can be seen in Fig. 1.

2.2. Deep learning models for ultrasonic thickness estimation

2.2.1. Convolutional network architecture

In order to determine the optimum network architecture, a hyper-parameter search is performed. Note that the model optimisation is based on estimating the mean thickness (D) of the synthetic/experimental corrosion profiles through regression with the assumption that the same topology is good for both mean and minimum thickness estimation. Once the parameter search is completed, the optimal network structure is then used to train models to extract the mean and minimum thickness (D, and D_{min}) in the region under the transducer aperture.

A systematic search of the CNN hyper-parameter space is performed. The search varied the number of convolutional layers $L \in [1, 15] \subset \mathbb{N}$, the kernel size $k \in [3, 251] \subset \mathbb{N}$, and the application of max pooling between convolutional layers to help reduce complexity and prevent over-fitting. This resulted in 178 distinct models considering only valid model structures and that k is restricted to a subset of $[3, 251]$ which includes $k \in \{3, 5, 7, 11, 15, 21, 31, 51, 71, 91, 121, 151, 201, 251\}$. The number of filters f at each layer is fixed at $f = 64$ and batch normalisation (BN) is applied to the activation from each convolutional layer. Dropout is applied to a subset of near-optimal models of the hyper-parameter search to select the optimal model. The dropout probability is set to $p = 0.2$ and it is applied after the batch normalisation step on the final convolution layer to prevent over-fitting [26]. An example of a candidate network architecture used for the parameter search can be seen in Fig. 2.

The model-based dataset is split into (1) training data to optimise network weights; (2) validation data to test network performance during the training process; (3) and testing data to provide a final set of never before seen data to test final network performance. A train–validation–test split of 60%, 20%, 20% was used with each model being trained for 600 epochs (with a batch size of 128) using the Adam optimisation algorithm [27], which is a widely used adaptive extension of classic stochastic gradient descent. A flowchart summarising all the steps of the proposed optimisation methodology is shown in Fig. 3.

The CNNs are implemented in TensorFlow [28] using the Keras functional API, with a recursive 1D CNN model building algorithm being developed to help facilitate the parameter space search (www.github.com/casimp/undt-ai). Experience showed that all models converge before the specified 600 epochs and model check-pointing (recording network parameters throughout training) was used during training, with the best model (in terms of validation error) being reloaded at the end of training for testing purposes. These models are compared against a state-of-the-art 1D CNN architecture, namely InceptionTime [6]. This was originally developed for time series classification and has been adapted for regression as part of this work (www.github.com/casimp/undt-ai). InceptionTime uses three kernel sizes (i.e. 40, 20, 10) and while these can be modified, the aim is to compare an 'out-of-the-box' state-of-the-art model against simpler model architectures.

2.2.2. Optimising the dataset and model for experimental relevance

To ensure the developed models have performance that maps well to experimental data, efforts were made to transform the synthetic dataset into a form closer to that of experimental data. This is done as the synthetic data was generated with a broadband input signal, different to that seen experimentally. To match the frequency distribution in the synthetic data to experiment, filtering in the frequency domain of both synthetic and experimental time traces was applied by multiplying a

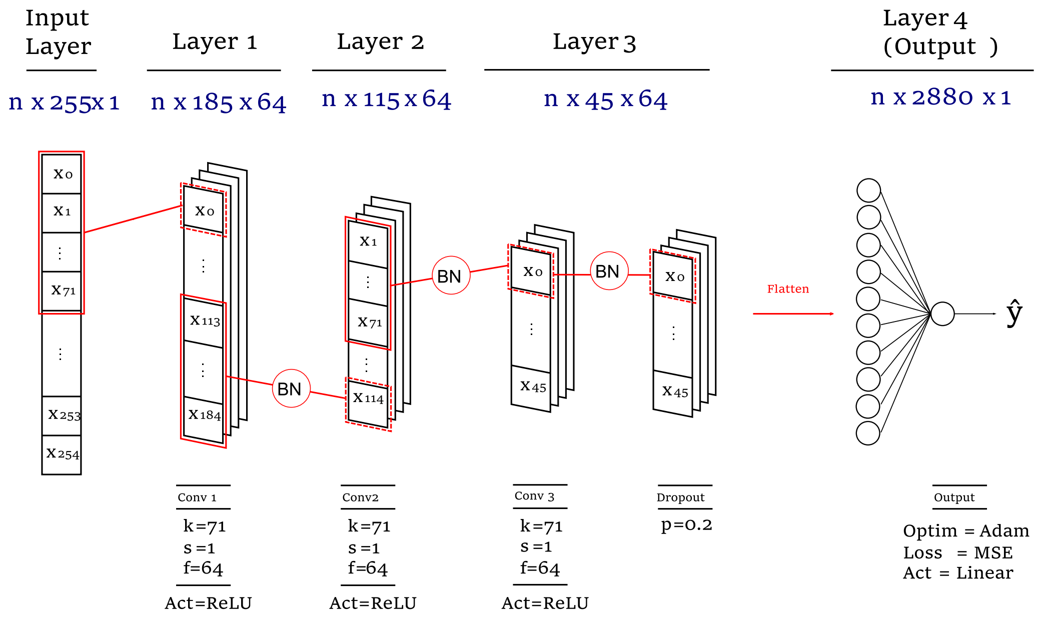


Fig. 2. Example of a four-layer CNN with three convolutional layers ($L = 3$), fixed kernel size of $k = 71$, and $f = 64$ filters. The model employs BN after the activation of each convolutional layer and dropout ($p = 20\%$) on the final layer.

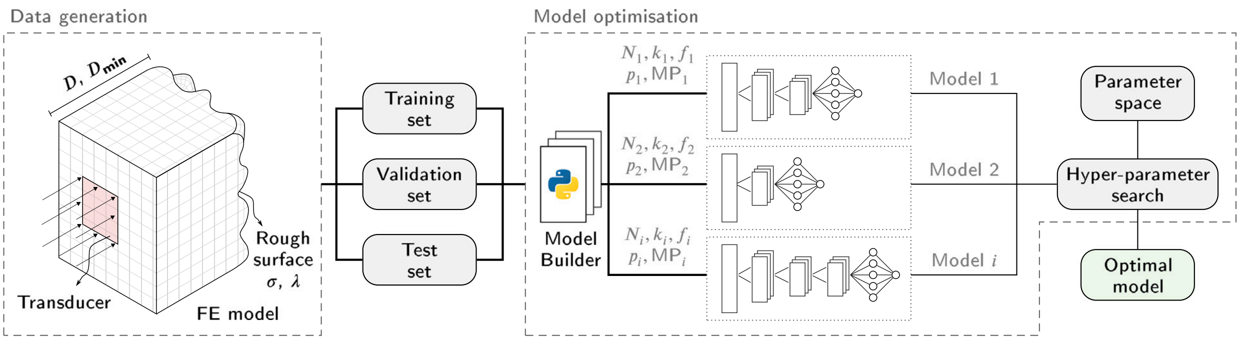


Fig. 3. Flowchart explaining the workflow of the data-driven thickness estimation approach to optimising the convolutional network architecture.

normalised Gaussian function by the magnitude of the spectrum, as follows:

$$\hat{X}(\omega) = X(\omega)\exp\left(-\frac{(\omega - F_c)^2}{2F_{\mathrm{std}}^2}\right) \quad (1)$$

where $\hat{X}(\omega)$ is the filtered frequency spectrum, $X(\omega)$ is the original spectrum, and F_c and F_{std} are the central frequency and width of the filter.

Prior to training, each time series was normalised relative to the first back wall reflection peak, with the cross-talk also being cropped from the dataset to prevent the model over adapting to the amplitude of this feature. This was carried out as cross-talk signals are instrumentation dependent and often either gated and/or saturated, so networks should not rely on this information. The crop was applied to the synthetic and experimental datasets so that possible influence from the cross-talk in the analysed time series is minimised.

2.3. Comparison against conventional techniques

To meaningfully assess the performance of the proposed approach it is important that the predictions made using the optimal CNN are compared against a realistic baseline from a conventional technique. This is done by measuring the time of flight for the ultrasound signal, specifically, calculating the time of arrival of the back wall signal. The component thickness is related to the time of flight by the speed of sound within the material. There are many reasonable approaches for the calculation of the time of flight, in this case we use the envelope of the reflected pulse, defined using the Hilbert transform. The peak of the envelope is used to define the mean thickness under the aperture (D) and a threshold crossing used to define the minimum thickness (D_{min}). The threshold is defined here as 30% of the peak of the envelope, with this being the smallest value that could be reliable defined for all time traces in the synthetic dataset.

3. Results

3.1. Synthetic data

3.1.1. 1D CNN hyper-parameter grid search

The results from the hyper-parameter search are shown in Fig. 4, with the influence of kernel size and number of layers on the RMSE of the validation dataset being shown in Fig. 4a. The relation between the number of trainable parameters and RMSE is also depicted in Fig. 4b. When the number of layers is $L > 2$, increasing the kernel size and number of trainable parameters improves the model accuracy for any value of k. This suggests that the candidate models' performance generally improves with complexity. The optimal CNN is the model with $L = 3$, and $k = 71$ without max pooling. This architecture was then tested multiple times by training the models 10 times in order to check repeatability and model uncertainty. Moreover, dropout was

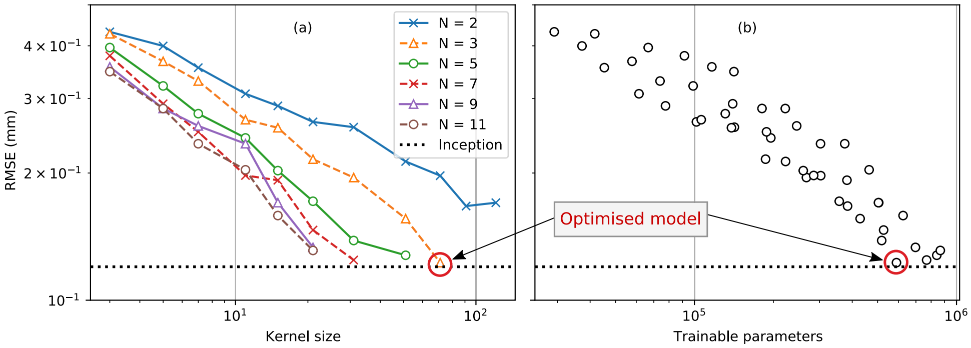


Fig. 4. Progression of RMSE against kernel size (a) and number of trainable parameters (c). A reference line showing the performance of InceptionTime is highlighted.

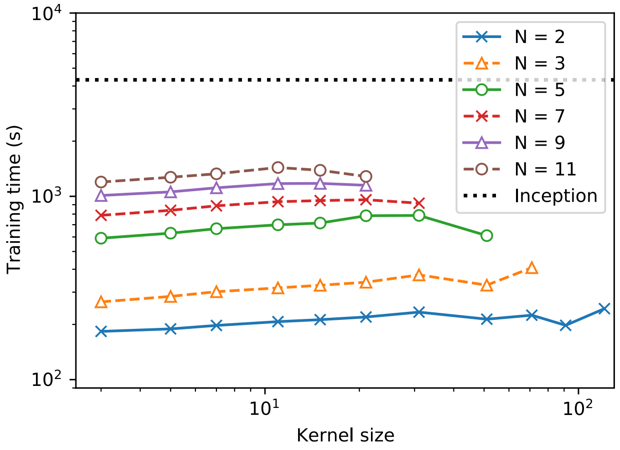


Fig. 5. Training time as a function of kernel size and number of layers. A reference line showing training time for InceptionTime is highlighted.

tested after the last convolutional layer before the fully connected layers and compared to the models without dropout. The results show that the mean RMSE of the optimal architecture using dropout was 0.11 mm and 0.12 mm without dropout. Therefore, dropout was incorporated into the optimal CNN architecture. Note that in addition to the improved performance, dropout increases regularisation and minimises over-fitting.

The results from the hyper-parameter search are also compared against results from the InceptionTime network. The InceptionTime ensemble model achieves slightly better test performance (RMSE = 0.11 mm) than any of the custom built models without dropout (as shown in Fig. 4). However, re-running the best performing model (L = 3, k = 71) and including dropout on the final layer produced a RMSE = 0.11 mm, which bring the RMSE in line with the performance of the more complex InceptionTime architecture. While a similar RMSE is achieved, the additional complexity of the InceptionTime network results in a training time of approximately 4000 s (see Fig. 5), this can be compared against a model with similar accuracy (L = 3, k = 71), which was trained in approximately 300 s on a NVIDIA GeForce RTX 2080, more than an order of magnitude faster. The training time of all models is presented in Fig. 5, here it can be seen that the training time is effectively insensitive to kernel size, with the change being defined by the number of layers (with all other aspects of the model architecture held constant). Thus in general the most sensible approach to network selection is to make use of the shallowest (minimum number of layers) network with a large kernel size. Doing this has the added benefit of limiting the required training dataset size for training [29,30].

3.1.2. Synthetic corrosion profile predictions

The optimal CNN architecture (shown in Fig. 2) has been used to train separate deep regression models for the prediction of (a) mean and (b) minimum synthetic corrosion profile thickness. Fig. 6 shows the predictions on the test set (which contains approximately 2000 test cases), with comparisons being made against predictions made using peak of the envelope and threshold crossing algorithms (for D and D_{min} respectively). The associated error distributions for each of these approaches and test cases are shown in Fig. 7. The RMSE for the CNN trained and evaluated with respect to mean thickness is 0.11 mm, compared to an RMSE of 0.65 mm for the peak of the envelope approach. The same network architecture trained and evaluated on the minimum thickness results in a RMSE of 0.2 mm, compared with an RMSE of 0.7 mm for the envelope thresholding approach.

There is a noticeable negative bias on the minimum thickness predictions made by thresholding the pulse envelope (i.e. the predictions systematically under-estimate the true values). This bias is associated with the threshold that was chosen (i.e. 30%). A smaller threshold would likely reduce the observed bias but would prevent predictions on the noisier and thinner synthetic profiles and, while an offset could be applied to the entire prediction set, this level of manual data manipulation is unrealistic for real world applications, where the ground truth is not known.

3.2. Experimental data

3.2.1. Experimental measurements

To generate a comparable experimental test, a subset of five of the synthetic corrosion profiles were selected and machined from a low carbon steel block to allow for back-to-back comparison between synthetic and experimental measurements. The corrosion profiles were selected based on a requirement to cover the feature space in terms of D and roughness, σ. Rough (σ = 0.57 mm) and smooth (σ = 0.20 mm) profiles were selected with mean thicknesses of approximately 6 mm, 10 mm and 14 mm. There was limited scope to vary the correlation length λ due to the physical restrictions imposed by the milling tool; very short correlation lengths were therefore not sampled. Three of the five selected profiles are shown in Fig. 8, with Fig. 8a representing a smooth corrosion profile example and Figs. 8b and c being two relatively rough test cases.

Data was collected using a 6 mm diameter 5MHz panametrics Ultrasonic probe gel coupled to the flat top sample of the specimens. The probe was actuated with an Olympus pulser–receiver and the resulting signals were recorded with a Handyscope HS5 digital oscilloscope. Due to the limited size of the experimental dataset, high test accuracy for any given model does not necessarily reflect underlying test accuracy of that model-training data combination. The stochastic nature of deep neural networks introduces variability in the predictions and, across a small dataset, the impact of this is exacerbated, with relatively small perturbations in predicted values having a disproportionate effect on the suggested accuracy. To improve confidence in the reported error, each model was trained ten times, with random weight initialisation. The errors for all predictions across all runs are used to define the RMSE.

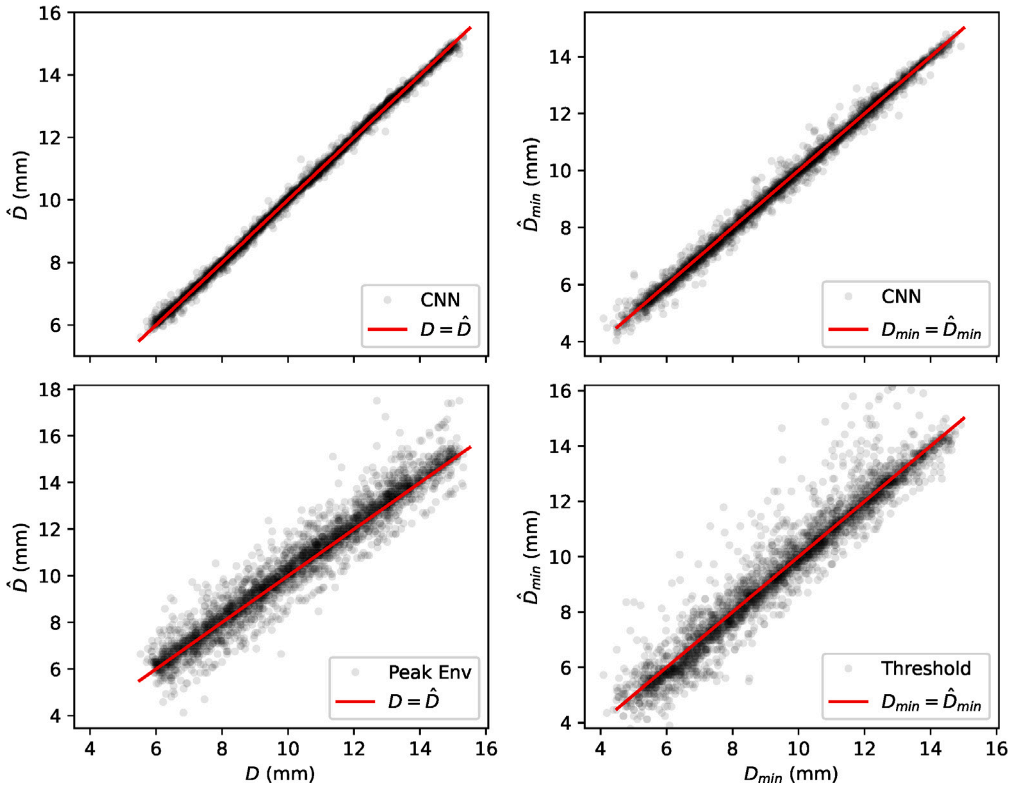


Fig. 6. Synthetic corrosion profile predictions for D and D_{min} using the optimal CNN architecture (separately trained models for D and D_{min}) and comparison against prediction using the pulse envelope approaches.

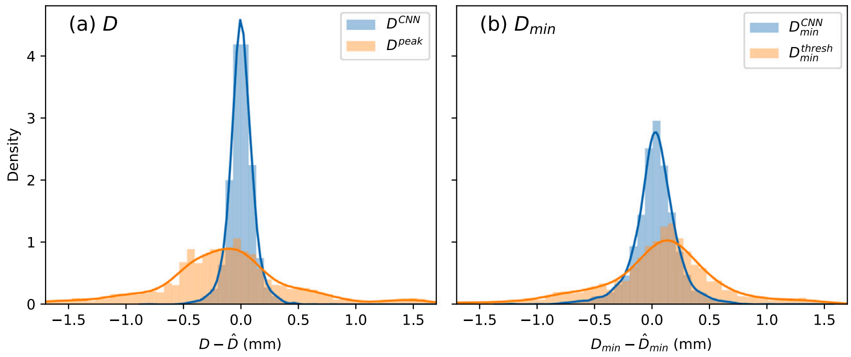


Fig. 7. Distribution of errors for (a) D and (b) D_{min} with predictions made using the optimal CNN and comparison against those produced using pulse envelope approaches.

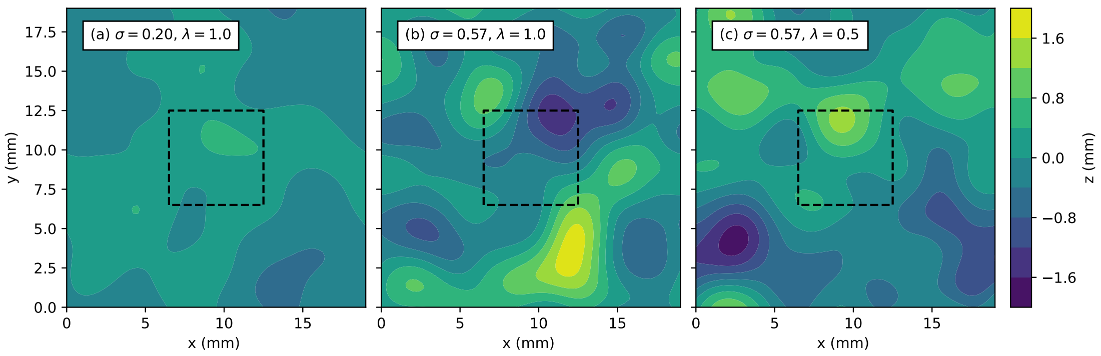


Fig. 8. Experimental corrosion profiles with (a) low roughness and long correlation length, (b) high roughness and long correlation length and (c) high roughness, short correlation length. The dashed lines define the aperture size and location.

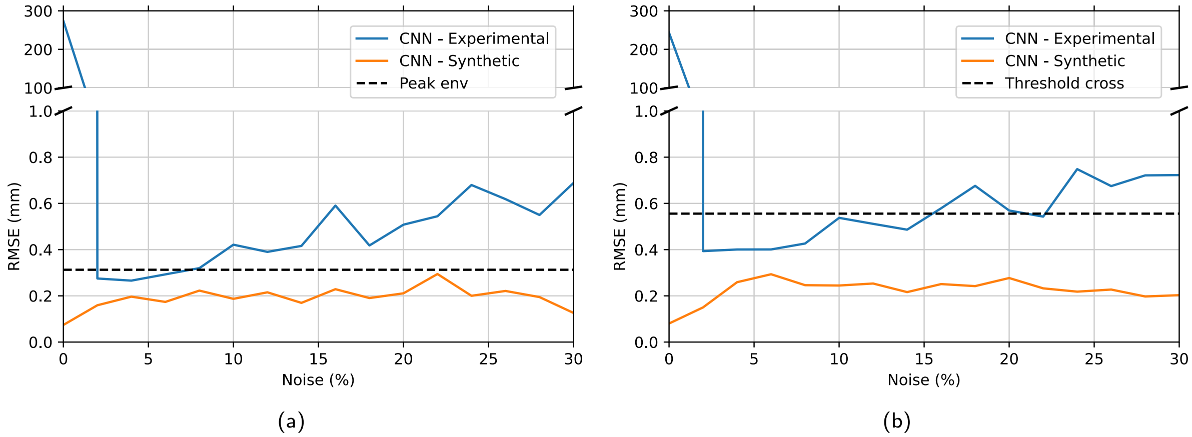


Fig. 9. RMSE levels of (a) mean and (b) minimum thickness estimation of models trained at different noise levels in comparison with the corresponding numerical data to the experimental samples. The RMSE of the CNN predictions is compared against the RMSE of the pulse envelope and threshold crossing approaches.

3.2.2. Comparison against classical methods

The optimised model architecture was trained using the previously discussed frequency corrected synthetic dataset with applied noise varying from 0% to 30% of the signal maximum. Note that this percentage refers to the standard deviation of normally distributed noise. The standard deviation varies with respect to the maximum signal amplitude, so the definition of noise is analogue to the signal-to-noise ratio as it considers the relation between noise-free signal amplitude and noise amplitude. The addition of noise, which acts as a regularisation technique [2,31], was essential to mimic the reality (compared to simulation) of experimental data and the values were taken much higher than those seen experimentally (approximately 1% of the signal maximum) to explore the effect on network performance. The results from this analysis can be seen in Fig. 9.

Models trained on noise-free data are highly unstable. Adding noise to the synthetic training data significantly improves the accuracy of the predicted experimental values, with improvements occurring for individual models trained with 2% additive Gaussian white noise applied to the synthetic training signals. Simultaneously, there is a reduction in the accuracy of the same model tested on the synthetic test set. Applying noise at a level of 2% to the combined dataset increases the synthetic test RMSE from 0.07 mm to 0.18 mm for the mean thickness and from 0.08 mm to 0.16 mm for minimum thickness compared to noise free training; the associated RMSE on the experimental dataset notably reduces from 273.86 mm to 0.28 mm and from 242.98 mm to 0.39 mm for mean and minimum thickness estimation respectively. As expected, the predictions on experimental data have higher RMSE than the predictions on the corresponding synthetic data. This is as a result of the increasing complexity of the experimental signals stemming from, for example, subtle differences in the real transducer geometry with respect to the simulated one. Increasing the noise level above 2% makes the RMSE for both mean and minimum thickness gradually increase for the synthetic and experimental test signals. A similar level of experimental performance was observed for applied noise levels from 2 to 6%, suggesting that the response is not overly sensitive to the noise level near the optimum. Note that the optimum level of noise yields RMSE in experimental test signals consistently lower than that obtained through the peak envelope and 30% threshold crossing (black dashed lines in Fig. 9) in the noise interval between 2 to 6%.

Predictions for the experimental mean and minimum thickness have been made using the optimised model architecture trained on combined raw synthetic and frequency-processed training data, transformed using 2% additive Gaussian noise. The results of these predictions are shown in Fig. 10 for each of the ten runs that were completed for each model. The results are compared against predictions made using the peak of the pulse envelope (D) and a pulse envelope threshold cross (D_{min}). In both cases the CNN outperforms the conventional approach, with lower RMSE being recorded for both D and D_{min} (refer to Fig. 9).

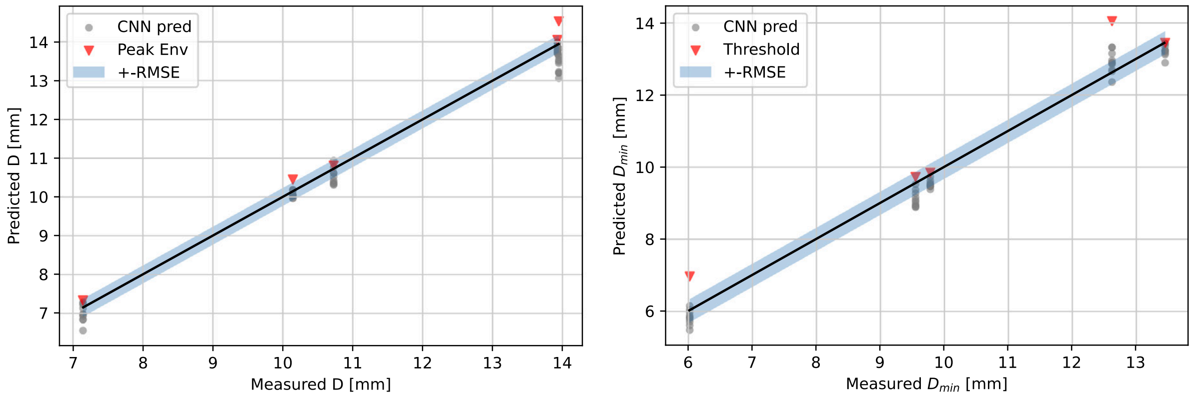
4. Discussion

For FE-generated synthetic datasets, a well optimised CNN can provide a noticeable improvement in predictive accuracy when compared to more conventional approaches (e.g. peak of the envelope). This improvement is related to (1) the complex set of features that the CNN discovers through the adaptively learnt filters and (2) the non-linear relationships between these features; both points are synonymous with convolutional networks. The complex feature selection is further illustrated in the following paragraphs, with an example of a ReLU activated, filtered time series being given in Fig. 11; this indicates the number of features of interest within the ultrasound time series.

4.1. On the optimal network architecture

For deep regression for ultrasound time series analysis, deeper networks do not provide additional improvements in accuracy beyond $L = 3$, where L is the number of convolutional layers. In general, adding more layers and making a network deep enhances its ability to learn non-linear relationships and mappings. On a fundamental level, the corrosion profile depth is related linearly to the front wall to back wall time of flight and the speed of sound. There is likely an upper limit to the amount of complex non-linearity required. Once a network is sufficiently deep to learn the required non-linear mappings, increasing the depth would not be expected to degrade performance (i.e. the network should simply be able to learn an identity mapping [16]), providing that sufficient regularisation is provided. Note that this is at the cost of greater train time and unnecessary complexity. This is consistent with our observations, with only a small drop-off in performance for deeper networks with the same complexity in terms of trainable parameters.

While a considerable effort has been made to dissect and explain an optimal, simplified design architecture for this type of time series regression problem, it is also clear that state-of-the-art models such as InceptionTime network do appear to work as a plug-and-play approach for achieving optimal or near optimal predictive accuracy. The only real question over their implementation for this type of problem is whether they are unnecessarily complex, if a far simpler model can achieve the same accuracy. The authors of the InceptionTime network acknowledge that even for their own time-series classification tasks, the advanced features of the network do not provide an additional performance boost [6]. A simpler architecture is more easily explainable (e.g. the network weights and activations are more accessible) and the training time can be significantly lower (in this work, by more than an order of magnitude). The latter point is particularly important if a model requires multiple iterative runs while the dataset is tuned and explored. It therefore seems reasonable to suggest that a very simple general-purpose 1D convolutional network may, in many situations, strike



(a) (b)

Fig. 10. Model predictions of (a) mean and (b) minimum thickness for each experimental sample at 2% noise. The uncertainty bands represent the ±RMSE.

a better balance between simplicity in implementation and training speed. The authors of the current work therefore provide a simple toolkit based on the findings from the current research to easily build this type of network [www.github.com/casimp/undt-ai/].

4.2. On the experimental application

While a deep learning approach has been shown to be very successful when training and testing on synthetic data, transfer of the model to the experimental data was a challenge. The optimised CNN architecture trained on frequency-corrected data resulted in an unacceptably high error in the experimental data set (and far worse than that from conventional techniques). Fundamentally, although the simulation produces superficially similar ultrasonic responses to experiment, there are a number of known discrepancies which cause subtle differences, for instance: (1) the simulation is based on transducers with rectangular footprints over which the sensitivity is uniform, while the experiment utilises circular transducers with unknown (but generally non-uniform) sensitivity; (2) in the simulation the transducer is not explicitly modelled, but in experiment the transducer presents a finite acoustic load which damps surface waves. The accuracy in the experimental prediction was ultimately improved through additive Gaussian white noise; this resulted in mean and minimum coupon thickness predictions that had a RMSE that was approximately 50% smaller than those made using the pulse envelope techniques, with an optimal noise level of 2%. Adding noise resulted in an improved predictive accuracy, which can be understood by considering (a) how the model will learn in the presence of high levels of noise and (b) the underlying physical validity of the synthetic data.

As an increasing amount of noise is added to the data, the number of discriminative features on each time trace will reduce until the only reliable feature will be the peak of the first reflection. Adding noise would therefore be expected to seriously compromise the model's ability to learn and limit the complexity of the model. As a simple test of this hypothesis we can compare the activations from all filters for the first layer of models trained on synthetic data with 2% and 6% Gaussian noise and the raw synthetic data. This comparison is shown in Fig. 11 where a histogram of the locations of all (i.e. f = 64) activated values are illustrated. Fig. 11 effectively depicts where the features of interest are within a given time series. As expected, a model trained on noisy data tends to focus more on the region around the first reflection as the noise level increases. A model trained on the raw synthetic data leverages features across the full time series (multiple reflections, etc.), and will have learnt more complex interactions. However, this potentially makes it more susceptible to physically unrealistic features in the synthetic time trace. While the region of focus is more limited when the model has been trained on noisy data, the CNN is still able to learn valuable features that translate across from the synthetic to experimental data and, critically, the performance of the network still outperforms predictions produced using the peak of the envelope (or associated threshold). The network has still been able to learn features and their non-linear mappings to allow it to outperform a conventional approach. Furthermore, in terms of minimum thickness predictions, the CNN not only outperforms a thresholding of the pulse envelope, but is also more reliable. Small changes in the pulse threshold cause large changes in the predicted thickness and are difficult to reliably and automatically set.

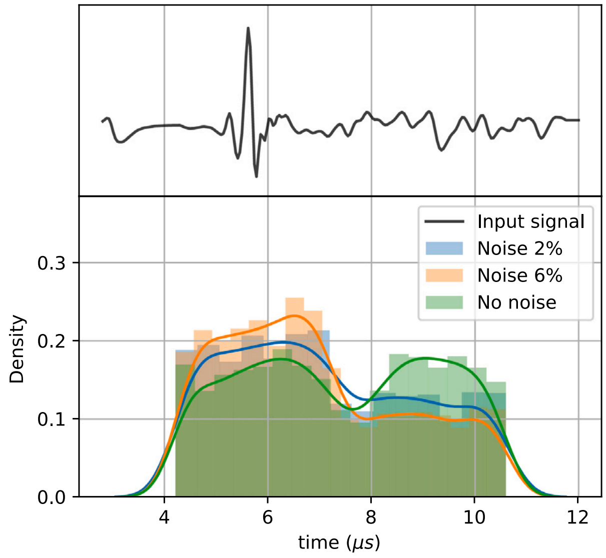


Fig. 11. Location of activated values (bottom panel) for a single time trace (top panel) from the first layers of a CNN trained on noisy data and noise-free data.

In summary, our training data is not sufficiently physically realistic to be a drop in replacement for experimental data; it is providing the network with features that are unreliable for thickness prediction on experimental data. By adding noise we reduce the feature space, forcing the model to only learn from the most significant and reliable features, which fortunately translate well to the experimental dataset. In order to use deep learning for NDE there must almost necessarily be an element of this type of synthetic data creation or augmentation. This study has highlighted both the potential of this approach (in terms of the high predictive accuracy) and also what will likely be a common pitfall, which is to say a critical, and comparatively nuanced (to the human eye) divergence between the physical response of the synthetic data and the real world response. Datasets must ideally either achieve an even higher level of physical realism than displayed here or, practitioners must accept that training on synthetic data is only the first step in either (a) a transfer learning process [32] or (b) a more complex data manipulation and augmentation process.

5. Conclusions

A deep learning approach has been proposed in this paper to characterise thickness information (either minimum or mean thickness) in complex corroded materials from ultrasonic data. A customised architecture is developed through a hyper-parameter optimisation, whereby the number of layers, the kernel size, and the presence of dropout are chosen. The optimal network architecture is then tested using both synthetic and experimental data. A comparison is provided against standard deep learning models for time series analysis (i.e. InceptionTime) and classical methods to extract thickness information. The following conclusions are drawn:

- CNNs offer the potential for a step change in corrosion profile predictive accuracy and reliability, with testing on an FEA generated synthetic dataset highlighting a four-fold reduction in RMSE compared to a peak of the envelope approach.
- A relatively shallow network is required, which is likely related to a minimal requirement for complex non-linear relationships to model a fundamentally linear relationship between time of flight and profile thickness. More sophisticated architectures such as InceptionTime provide a slight performance improvement at the cost of a considerable increase in complexity.
- A model trained on synthetic data struggles to produce the same predictive accuracy on experimental data acquired on nominally identical profiles. This further demonstrates the need for hyper-real data for CNNs, which can capture nuanced feature representations and relationships and are susceptible to learning features which are systematically present in synthetic data not physically realistic.
- Adding Gaussian noise to a synthetic profile prevents the networks from over-adapting to low-level features, focusing the network on the key discriminative feature, the first back wall reflection. This then allows the network the generalise across to the experimental dataset.

CRediT authorship contribution statement

Sergio Cantero-Chinchilla: Conceptualization, Methodology, Software, Investigation, Data curation, Validation, Formal analysis, Visualization, Writing – original draft. **Christopher A. Simpson:** Conceptualization, Methodology, Software, Investigation, Visualization. **Alexander Ballisat:** Conceptualization, Methodology, Software, Investigation, Data curation. **Anthony J. Croxford:** Conceptualization, Writing – review & editing, Supervision. **Paul D. Wilcox:** Conceptualization, Writing – review & editing, Resources, Project administration.

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.

Data availability

The authors do not have permission to share data.

Acknowledgements

The work reported is part of a pilot project (Grant number 100374) funded by Lloyd's Register Foundation, UK and the Alan Turing Institute Data-Centric Engineering Programme. This 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] Barshinger J, Pellegrino B, Nugent M. Ultrasonic sensor system for wall-thickness monitoring. Insp J 2016;22(2):2–7.
[2] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.
[3] Rawat W, Wang Z. Deep convolutional neural networks for image classification: A comprehensive review. Neural Comput 2017;29(9):2352–449.
[4] Medak D, Posilović L, Subašić M, Budimir M, Lončarić S. Automated defect detection from ultrasonic images using deep learning. IEEE Trans Ultrason Ferroelectr Freq Control 2021.
[5] Dempster A, Petitjean F, Webb GI. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. Data Min Knowl Discov 2020;34(5):1454–95.
[6] Fawaz HI, Lucas B, Forestier G, Pelletier C, Schmidt DF, Weber J, Webb GI, Idoumghar L, Muller P-A, Petitjean F. InceptionTime: Finding AlexNet for time series classification. Data Min Knowl Discov 2020;34(6):1936–62.
[7] Li J, Chen Z-Z, Huang L, Fang M, Li B, Fu X, Wang H, Zhao Q. Automatic classification of fetal heart rate based on convolutional neural network. IEEE Internet Things J 2018;6(2):1394–401.
[8] Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller P-A. Deep learning for time series classification: a review. Data Min Knowl Discov 2019;33(4):917–63.
[9] Längkvist M, Karlsson L, Loutfi A. A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognit Lett 2014;42:11–24.
[10] Ng AY, Jordan MI. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In: Advances in neural information processing systems. 2002, p. 841–8.
[11] Lathuilière S, Mesejo P, Alameda-Pineda X, Horaud R. A comprehensive analysis of deep regression. IEEE Trans Pattern Anal Mach Intell 2019;42(9):2065–81.
[12] Kashiparekh K, Narwariya J, Malhotra P, Vig L, Shroff G. ConvTimeNet: A pre-trained deep convolutional neural network for time series classification. In: 2019 international joint conference on neural networks. IJCNN, IEEE; 2019, p. 1–8.
[13] Tang W, Long G, Liu L, Zhou T, Jiang J, Blumenstein M. Rethinking 1D-CNN for time series classification: A stronger baseline. 2020, arXiv preprint arXiv:2002.10061.
[14] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, p. 1–9.
[15] Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, inception-ResNet and the impact of residual connections on learning. In: 31st AAAI conference on artificial intelligence. 2017.
[16] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 770–8.
[17] Dau HA, Bagnall A, Kamgar K, Yeh C-CM, Zhu Y, Gharghabi S, Ratanamahatana CA, Keogh E. The UCR time series archive. IEEE/CAA J Autom Sin 2019;6(6):1293–305.
[18] Munir N, Park J, Kim H-J, Song S-J, Kang S-S. Performance enhancement of convolutional neural network for ultrasonic flaw classification by adopting autoencoder. NDT & E Int 2020;111:102218.
[19] Cantero-Chinchilla S, Wilcox PD, Croxford AJ. Deep learning in automated ultrasonic NDE–developments, axioms and opportunities. NDT & E Int 2022;131:102703.
[20] Guo Y, Xiao Z, Geng L, Wu J, Zhang F, Liu Y, Wang W. Fully convolutional neural network with GRU for 3D braided composite material flaw detection. IEEE Access 2019;7:151180–8.
[21] Pyle RJ, Bevan RL, Hughes RR, Rachev RK, Ali AAS, Wilcox PD. Deep learning for ultrasonic crack characterization in NDE. IEEE Trans Ultrason Ferroelectr Freq Control 2021.
[22] Cantero-Chinchilla S, Wilcox PD, Croxford AJ. A deep learning based methodology for artefact identification and suppression with application to ultrasonic images. NDT & E Int 2022;126:102575.
[23] Huthwaite P. Accelerated finite element elastodynamic simulations using the GPU. J Comput Phys 2014;257:687–707.
[24] Archer G, Saltelli A, Sobol IM. Sensitivity measures, ANOVA-like techniques and the use of bootstrap. J Stat Comput Simul 1997;58(2):99–120.
[25] Ballisat A, Wilcox P, Croxford A. Model based optimisation of ultrasonic corrosion measurement. In: Structural health monitoring 2019: Enabling intelligent life-cycle health management for industry internet of things (IIOT) - Proceedings of the 12th international workshop on structural health monitoring, Vol. 1. 2019, p. 933–8.

[26] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15(1):1929–58.

[27] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.

[28] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, et al. Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). 2016, p. 265–83.

[29] Lei F, Liu X, Dai Q, Ling BW-K. Shallow convolutional neural network for image classification. SN Appl Sci 2020;2(1):1–8.

[30] Miao S, Xu H, Han Z, Zhu Y. Recognizing facial expressions using a shallow convolutional neural network. IEEE Access 2019;7:78000–11.

[31] Zur RM, Jiang Y, Pesce LL, Drukker K. Noise injection for training artificial neural networks: A comparison with weight decay and early stopping. Med Phys 2009;36(10):4810–8.

[32] Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C. A survey on deep transfer learning. In: International conference on artificial neural networks. Springer; 2018, p. 270–9.