Mechanical Systems and Signal Processing 224 (2025) 112007

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

Transfer learning in guided wave testing of pipes

Mikolaj Mroszczak [a,*], Robin E. Jones [b], Peter Huthwaite [a], Stefano Mariani [c]

[a] NDE Group, Departament of Mechanical Engineering, Imperial College London, SW7 2AZ London, UK
[b] Guided Ultrasonics Limited, TW8 8HQ Brentford, UK
[c] Department of Civil, Chemical, Environmental and Materials Engineering — DICAM, University of Bologna, Viale del Risorgimento 2, Bologna 40136, Italy

ARTICLE INFO

Keywords:
Machine Learning
Guided Waves
Ultrasound
Non-Destructive Testing
Transfer Learning
Automation

ABSTRACT

Guided wave testing (GWT) is a non-destructive testing (NDT) technique extensively used for in-service testing of pipes that allows the inspection of tens of metres of pipe in either direction from a single sensor position. The aims are to identify and locate all physical features found along the pipe in the axial direction, and in particular the presence of defects, such as cracks or corrosion patches. However, the signals output by GWT of pipes are complex to interpret, making the quality of inspection highly dependent on the operator skills. Due to such signal complexities, at present there is a lack of automated procedures that can help operators in this task. Some of the recently developed machine learning (ML) algorithms are expected to possess the modelling capabilities required to address such a classification task, though they would typically need hundreds if not thousands of labelled input data for their training. This amount of experimental data is seldom available in the NDT field, particularly with regards to the damage cases. The main purpose of this article is to investigate whether, and how, it is possible to augment an available set of labelled experimental data with a synthetic dataset having characteristics that are similar but still distinct from the real ones. This is studied by training three different ML models with various combinations of actual and simulated data pertaining to GWT of pipes, the goal being the automated detection of reflections from pipe features within the inspection traces. The results demonstrate that when there is scarce availability of experimental data, substantial detection improvements can be achieved by pre-training the chosen ML model with synthetic data, before fine-tuning it on actual inspection data. In particular, the ML algorithm that is found to perform best for this task is a VGG-Net model, which is shown to yield false positive rates in the order of ~1.5 to 4 % at the fixed true positive rate of 99.7 %.

1. Introduction

Guided wave testing (GWT) is a well-established method used to perform non-destructive testing of pipes. By exciting an axially-propagating ultrasonic guided wave in the pipe, a full section of a pipeline at a range of up to 50 m can be inspected from a single access point [1,2]. This is a major time-saving and economically advantageous opportunity compared to performing many localised inspections with bulk ultrasound, especially when applied to buried or insulated pipes [3,4]. Guided wave inspection is typically performed by a qualified inspector who places the transducer ring [5] on the pipeline, performs the inspection and analyses the results,

* Corresponding author.
E-mail address: m.mroszczak19@imperial.ac.uk (M. Mroszczak).

https://doi.org/10.1016/j.ymssp.2024.112007
Received 19 December 2023; Received in revised form 21 July 2024; Accepted 29 September 2024
Available online 1 October 2024

flagging all pipe features (PFs), such as welds, bends and defects. If any PFs of concern are flagged, they are further inspected using one of the other Non-Destructive Testing (NDT) modalities (usually conventional ultrasonic testing or electromagnetic testing). This analysis requires a significant time commitment from trained personnel and may suffer from inconsistencies between the performance of different operators. As a result, it would be beneficial to develop a method for automated PF detection, which could help to alleviate the impact of possible human errors within the process.

One of the modern methods of automation of classification tasks is machine learning (ML). It involves the creation of an algorithm which maps the inputs to outputs imitating the human approach of learning on representative data [6]. In the most commonly envisioned NDT scenario, the input is either raw inspection data or extracted features thereof. The output is the classification as either a structural feature of interest or a non-feature. The machine learning model approximates the distribution of the training data in feature-space and uses it to classify new data. It is therefore necessary to provide sufficient data to accurately capture the underlying distribution and for the training data to belong to the same distribution (or at least a sufficiently similar distribution) as the intended use case. This leads to the main issue in ML for NDT, the scarcity of good quality data [7].

To solve any identification problem using ML, there is a need to gather data belonging to the distributions to be identified. In the NDT context the input data can be divided into the positive class, including samples containing a structural feature of interest, and negative, corresponding to all the other samples. Gathering both classes of data is effort intensive, as it is necessary to physically perform an inspection, but the difficulty is compounded in the case of the data containing defects, due to their rarity and that it only becomes clear that a defect is present after the inspection is complete. In similar scenarios in other domains, such as medical screening for tumours, the issue is somewhat less pronounced thanks to the systems that have been set up to ensure the continuity and shareability of the data [8], whereas in NDT much of the inspection data is proprietary and not available to researchers. Furthermore, there is no framework for gathering and sharing inspection data, despite initiatives such as DICONDE [9]. To combat that, much of the previous work has taken one of three approaches. The first approach is using simulated data, such as that generated by finite element modelling [10,11]. This has the advantage of being able to quickly generate large amounts of data, but it can suffer from difficulties in fully reflecting the intricacies of real-world scenarios [12]. In fact, an inspection can suffer from many sources of noise which are difficult to estimate and model, such as the effects of transducer coupling to the test piece, temperature variations, non-uniform mechanical properties of the sample and, for pipes, their contents [13]. As a result, data simulated using current approaches is often a poor representation of the true distribution. The second method is to use data measured from artificially manufactured defects [14,15]. In this way some of the noise sources (e.g. transducer coupling or temperature variations) are present, but there is still no guarantee that all the characteristics of real-life signals exist in the manufactured specimen. Additionally, the number of test-cases that can be produced by this approach is necessarily limited since manufacturing representative defects is costly. The third method is an attempt to employ transfer learning methods to adjust the training process, so the model can take the information out of the simulated data and apply it to the real dataset, bridging the so-called sim2real gap [16]. This approach acknowledges the differences between simulations and real data and builds the algorithm or pre-processes the data so that the model performs just as well on the real as on the simulated data. Transfer learning has been successfully applied to guided wave problems in the area of full wavefield reconstruction, where it has enabled generating large numbers of realistic wavefield simulations [17 18]. It has also been applied to defect localisation and severity assessment using guided waves [19 20]. However, in these works artificial defects rather than damage found on actual in-service inspections were considered. Finally, Zhang et al. presented an algorithm that can be trained to detect a defect using GWT in one structure and that can later be fine-tuned to work on a structure with a different material or geometry [21].

Despite the large body of research into the application of machine learning to ultrasonic inspections (see e.g. [22,10,23]), to the authors' knowledge this effort has not yet yielded any routine industrial application. This work aims to bridge the gap between research and industrial deployment by investigating the impact of the amount of real and simulated data to aid the ML training process in order to improve the performance on actual site data. In particular, the focus of the work is on the automated detection of PFs in guided wave testing of pipes, and the classification performance offered by three ML algorithms that are considered representative of the progress in deep learning research is evaluated. Namely, these are the Multilayer Perceptron (MLP) [24], here considered as the "basic" deep learning architecture, VGG-Net [25], a well-established network designed to analyse spatially dependent data, and U-Net [26], a state of the art image processing architecture. Other than shedding the light on the usefulness of simulated data for ML training, the specific industrial goal of the ML strategy developed in this work is to act as a filter in order to significantly limit the amount of pipe inspection data presented to qualified inspectors for the classification of PFs.

This paper is structured as follows: section 2 provides the information on the background of GWT of pipes and introduces the experimental design. It then describes the proposed ML architectures, the design of the ML training procedure and the metrics that are used to assess the performances. Section 3 summarizes the main results of the work by first comparing the performance offered by the employed algorithms at various sizes of simulated and real training data, and by then investigating which data samples are most often misclassified. Finally, Section 4 draws the salient conclusions of the study.

2. Background and Methodology

2.1. Guided wave testing of pipes

GWT is a modality of ultrasonic testing that follows much of the same principles as standard bulk wave ultrasound. It uses stress waves excited in the test sample and analyses the echoes resulting from the wave interaction with local acoustic impedance changes caused by PFs of interest. For pipe testing, frequencies in the order of 10 s of kHz are used to excite stress waves that are subject to the boundary conditions of the inner and outer surfaces of the pipe [27]. This results in a finite number of solutions for the wave equation at any given frequency [28] with the number rising with frequency. These solutions are known as guided wave modes, which propagate through the waveguide material via a prescribed shape. It is typically advantageous to use non-dispersive axisymmetric modes [29] such as the fundamental torsional mode T(0, 1) and the second longitudinal mode L(0, 2), which ensure lower dissipation of wave energy allowing for a significantly longer inspection range, and which simplify the interpretation of echoes in the measured signals. At present, the vast majority of pipe inspection systems in the market use the T(0, 1) mode since it offers a number of advantages over L(0, 2) [29].

Fig. 1 presents the schematic of a typical experimental setup for GWT of pipes, where an incident torsional wave generated by the transducer ring interacts with a non-axisymmetric defect, such as a corrosion patch. The resulting reflection is a mixture of axisymmetric and non-axisymmetric modes which are received by the ring. It should be noted that in pipe GWT it is possible to control the direction of both the transmitted wave and sensitivity to any received modes using multiple rings and suitable phase shifting [29]. While not perfect, this will greatly suppress the presence of reflections from the unwanted direction; this approach will be used throughout this paper.

Fig. 2 shows the typical data resulting from one such guided wave test performed with Guided Ultrasonics Ltd. (GUL) equipment [31], which transmits the T(0, 1) mode and receive both T(0, 1) (black traces in Fig. 2) and the first pipe flexural mode F(1, 2) (red traces in Fig. 2), which is non-axisymmetric. The probing mode is reflected by axisymmetric PFs (such as welds) as the same T(0, 1) axisymmetric mode (Fig. 2a), while the interaction with a non-axisymmetric PF causes the reflection to comprise both T(0, 1) and F(1, 2) as seen in Fig. 2b. For the purposes of practical inspections, the reflected T(0, 1) data is related to the percentage of cross-sectional area change (CSC) of the waveguide, while F(1, 2) gives an indication of the degree of circumferential localisation of the change [32].

Typically, this data is presented to a trained inspector, who assesses the traces and differentiates between noise, benign PFs, and defects. The operator first needs to set and apply a distance-amplitude correction (DAC) curve [2] (dashed black trace in Fig. 3) to the data by using reflectors of known strength, which are typically welds. The data traces can be rescaled by dividing them by the corresponding DAC value, such that at any axial location along the pipe, a T(0, 1) reflection from a complete cut perpendicular to the axis would have a 100 % amplitude. The operator then sets a "defect call level" (dashed blue line in Fig. 3) on the T(0, 1) mode, above which a pulse would be considered a potential defect and identified for further investigation, whose exact value also depends on the background noise in the analysed traces.

Finally, the operator would make use of their experience to examine and classify all reflections exceeding the call level. In particular, they would consider the amplitude of T(0, 1), the ratio between that and F(1, 2), as well as the shape and frequency dependency of the reflection. The inherent risk in this procedure is that setting the call level too high can lead to some PFs being missed, while setting it too low requires the operator to manually analyse a vast amount of data. For this reason, the goal of this study is to develop a machine learning algorithm that can be fed with raw T(0, 1) and F(1, 2) traces and that can select all segments of such traces that could possibly contain a PF reflection. This information would then be given to the operator who would then classify the detected PFs.

2.2. Experimental design

2.2.1. Data

Machine learning is highly dependent on access to good quality data. Two sets of data are used in this study: a batch of experimental data acquired by inspecting many pipes in the field, and artificial data generated using finite element modelling (FEM) simulations. All real and simulated time-traces are split into segments each containing 128 values for T(0, 1) and F(1, 2), with such length corresponding to 1.0432 m two-way propagation distance at the T(0, 1) velocity, given the sampling rate of 200 kHz This value is chosen as a compromise between the need for spatial resolution, the ability to encompass the full reflection from a single PF, and the

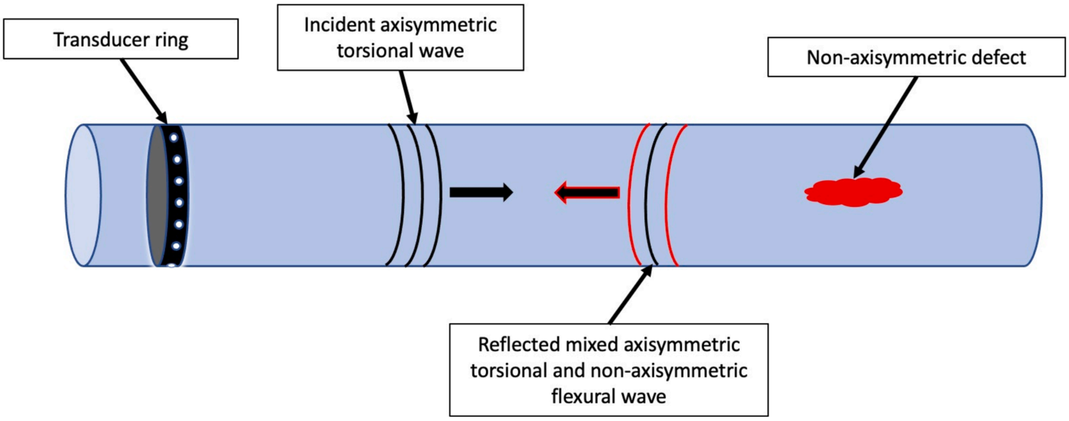
Transducer ring
Incident axisymmetric torsional wave
Non-axisymmetric defect
Reflected mixed axisymmetric torsional and non-axisymmetric flexural wave

Fig. 1. Schematic of a guided wave test of a pipe. The incident torsional wave (black) is reflected by the non-axisymmetric defect. The reflected wave is a mixture of torsional and flexural (red) modes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

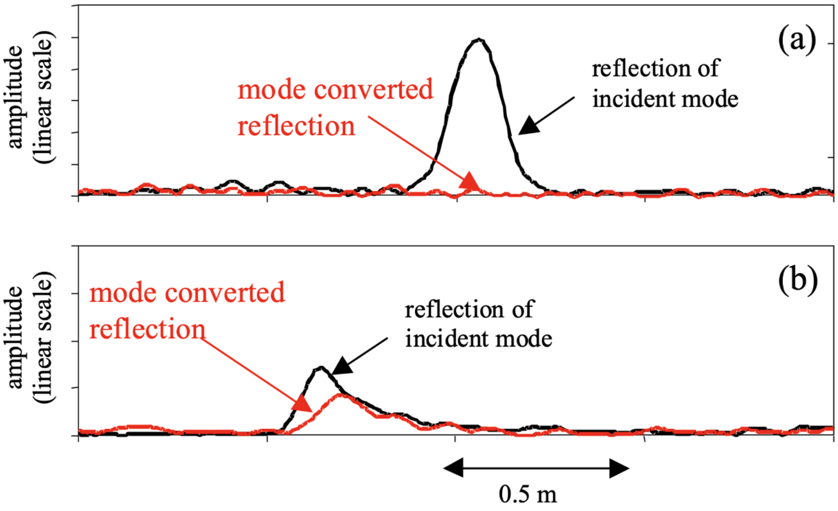

Fig. 2. Examples of signals collected via GWT of pipes. (a) shows the reflection from an axisymmetric PF. (b) shows the reflection from a non-axisymmetric PF such as a defect. Black and red traces correspond to T(0, 1) and F(1, 2) modes, respectively. Courtesy of [23]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

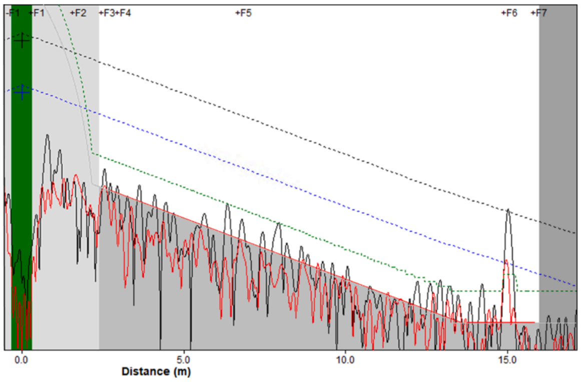

Fig. 3. Inspection of a generally corroded pipe. Black and red solid traces correspond to T(0, 1) and F(1, 2) modes, respectively. Weld DAC, call level and detection threshold (not discussed in here) are the dashed black, blue and green traces, respectively [28]. The sensor ring is at 0 m, while the feature at 15 m is a weld. The plot uses logarithmic scale on the y-axis. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

computational requirements, which rise with larger input data. Each of these segments corresponds to a single “data sample” (this is simply referred to as “sample” in the reminder of the article) that is used as input to the ML algorithms. In particular, each sample has the shape of (128, 2, 2), where the first dimension corresponds to time of acquisition, the second to T(0, 1) and F(1, 2) modes, and the third to enveloped and RF data. RF data is the raw amplitude of T(0, 1) or F(1, 2) time-trace, while the envelope is the magnitude of the analytic signal, which is the sum of the original RF signal, and its Hilbert transform multiplied by the imaginary number. Note that inspectors typically only use enveloped data, though the enveloping process necessarily removes the underlying frequency information. Since the amplitude of the unprocessed experimental data is meaningless, each numerical or experimental sample was normalised between -1 and 1 based on the maximum of the envelope of the T(0, 1) trace within the sample itself. This was done to ensure that each sample impacts the training of the neural network proportionally. In fact, if the amplitude variation between the samples is significant (as is the case in the unprocessed dataset), the training would likely overfit to high amplitude samples at the cost of low amplitude ones. Note that while it would likely be advantageous to normalise the data using the specific DAC curve, this is set manually by an inspector after some PFs are first detected and classified, hence it is not available at the first processing stage where the algorithm resulting from this research would be employed.

The real data was provided by GUL [31]. This consisted of traces collected over 55 inspections performed between the years of 2005 and 2012 on pipe diameters ranging from 4 to 36 in. Most of the inspections have been performed in a two-directional mode from the sensor position, resulting in two traces per inspection. Due to the complexity and uniqueness of the real-world pipe inspections, there is not much commonality in the data. The excitation frequency varies between 14 kHz and 50 kHz. The equipment used was either solid or inflatable transducer rings [34] working on 8 transducer channels. The inspected pipes included clean, buried, coated, and generally corroded ones, and contained a wide range of PFs (634 in total as detailed in Table 1), as annotated by qualified inspectors at time of testing. All inspection data is first processed using WavePro software [33] in order to export the T(0, 1) axisymmetric mode, the F(1, 2) non-axisymmetric mode, and the list of annotated PFs. The samples for the positive class of ML training are drawn by centring a rectangular window on each PF and shifting it by a small random number. The shifting is done to ensure that the PF is not always located at the midpoint of the sample and hence that the network performance is not dependent on the exact location of the PF in the sample. In fact, at time of testing the PF locations are clearly unknown and hence each PF is randomly positioned within its window. Also note that each PF only appears in a single sample to form the positive dataset. The negative class is drawn by selecting the segments of traces not containing PFs and thus composed solely of the background coherent and incoherent noise. Since the number of negative samples was significantly larger than the number of available PFs, the vast majority of those samples were discarded to achieve a class balance of roughly 3:1 ratio. This resulted in a “real” dataset size of 2400 samples, split between 634 positives and 1766 negatives. It is worth noting that such dataset contains a mixture of all excitation frequencies.

Simulated data is generated using the Pogo FEM package [35]. A 5 m-long section of an 8 mm-thick 8-inch steel pipe is simulated using 2.67 mm cubic brick elements, therefore discretising the pipe cross-section via 3 elements through the wall thickness and 240 elements in the circumferential direction. By setting the Courant number to 0.3, a time step of 0.132 μs was selected. 0.5 m-long absorbing boundaries using the stiffness reduction method [36] are applied at the ends of the modelled pipe in order to minimise any reflection from the ends of the pipes, such that the model effectively becomes a finite section of an infinite pipe. 24 nodes equally spaced around the external circumference of the pipe and at a distance of 1 m from one end of the model are selected to simulate transducers. The nodes exert circumferential forces in order to excite a 5-cycle Hann-windowed T(0, 1) wave at a centre frequency of 17, 21, 25.5, 31 or 35 kHz, and record the tangential displacement throughout the simulated inspection. For each simulation, the amplitudes of the forces excited by the various nodes are randomly unbalanced in order to introduce some realistic coherent noise [37]. The transducer readings are reduced to groups of three, resulting in eight channels, to mimic the process followed in the real data processing. Next, the axisymmetric and non-axisymmetric modes traveling in both directions from the simulated sensor ring are extracted using the angular spectrum method [38], and only the data pertaining to the 4 m-long pipe direction is retained. Finally, the data is down sampled to 200 kHz using MATLAB resample function [39] in order to match the sampling used in the real data.

A total of 12,000 different simulations are solved, half of which include either a simulated defect or weld, while other types of PFs, such as supports or flanges, are not considered for the simulations. The defects are represented as through-thickness cracks of varying circumferential extents, as detailed in Table 2. A weld is represented as a circumferentially-uniform local increase in thickness described by a quadratic function of axial distance from the centre line of the weld as shown in Fig. 4. The thickness function is defined as:

$$wallThickness = 1 + \left(\left(\frac{weldCapHeight}{weldAxialSpan}\right)^2 * (weldAxialSpan^2 - (x - weldCentre)^2)\right)$$

The size of the weld in terms of span and cap thickness is selected randomly within the ranges given in Table 2. The resultant simulated positive dataset contains 4500 defects and 1500 welds, with each sample coming from a single simulation, for a total of 6000 positive samples. The simulated negative dataset also includes 6000 samples, again each of which being drawn from a single simulation.

Fig. 5 shows a few examples of the samples available in the experimental (plots (a, c, e, g, h)) and simulated (plots (b, d, f)) datasets. Plots (a) and (b) are drawn from the negative class. Both T(0, 1) and F(1, 2) in this class are typically at noise level, though the normalisation amplifies the small random oscillations as seen in figure. Plot (c) shows the reflections from an internal pitting defect causing wall thickness loss of about 30 %, while plot (d) is drawn from the simulation of a defect removing 13 % of the pipe CSC. Their reflections are both characterised by an F(1, 2) amplitude being very close to that of T(0, 1), as would be expected for a PF with small circumferential extent. Plots (e) and (f) correspond to welds, which are strong axisymmetric reflectors, thus the expected signature is a large T(0, 1) reflection with a very low F(1, 2) contribution. Plot (g) shows a flange signature, which is typically symmetrical with a smaller F(1, 2) contribution. Finally, plot (h) shows a bend signature. The bend is welded at its two ends, therefore one would expect to see two separate weld reflections, the first one being significantly stronger than the second one since T(0, 1) gets mode-converted as it propagates through the bend itself, and some relatively high F(1, 2) content due to the mode conversion.

Table 1
PFs included in the available experimental dataset.

Pipe Feature Type	Number	Brief Explanation	Example of Signature
Weld	293	Weld that joins two pipe sections	Fig. 5(e)
Support	164	Element that transfers the load from a pipe to the supporting structures	–
Defect	58	Damage such as a crack or a corrosion patch	Fig. 5(c)
Bend	41	Curved pipe section	Fig. 5(h)
Flange	36	Mechanical part used to join two pipe sections	Fig. 5(g)
False echo	28	Signal signature due to the imperfect removal of reflections originating from a PF located on the other direction of testing	–
Unidentified anomaly	10	Signature noted in the signal that does not correspond to an evident physical PF or to a false echo	–
Entrance into earth	4	Section where a pipe enters into ground (e.g., for road crossing)	–

Table 2
FEM simulation parameters.

Simulation Parameter	Value
Model pipe length	5 m
Pipe thickness	8 mm
Pipe external Diameter	203.2 mm (8 in.)
Element Shape	8-noded cubic element with 1 integration point
Element Size	2.67 mm
Number of Elements	8,089,920
Courant Number	0.3, using bulk longitudinal wave speed of 6055.3 m/s
Time Step	0.132 μs
Time-length of simulated inspection	0.0285 s
Absorbing boundaries	0.5 m, each side
Excitation frequency	[17, 21, 25.5, 31, 35] kHz
Excitation shape	5-cycle Hann-windowed
Number of transducers	24 excitation nodes, every 3 adjoining grouped
Material properties	E = 210 GPa, G = 80 GPa, ρ = 8000 kg/m^3
Shear wave speed in the model	3162.3 m/s
Defect type	Through thickness crack
Defect size	[3.6–18 %]
Weld size	[25–35] mm span, [3–3.5] mm cap thickness

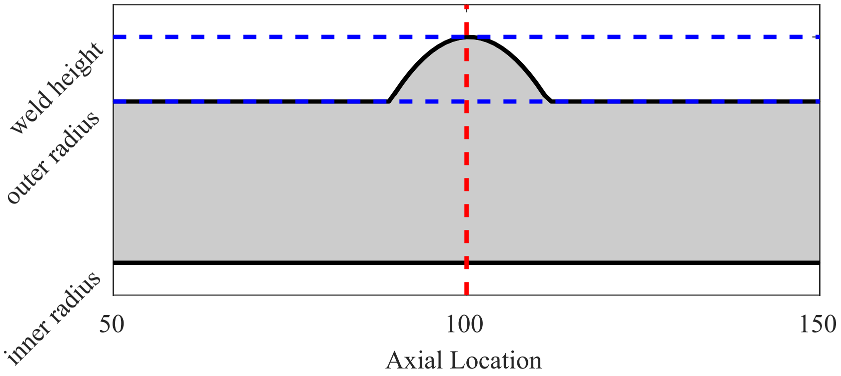

Fig. 4. Axial cross-section of a pipe wall containing a weld. The centre line in location 100 is marked with a red dashed line and the thickness of the weld is 40 % of wall thickness, marked between two blue lines. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.2.2. *Thresholding*

As discussed above, at present there is no standard procedure to automate the detection of PFs in guided wave-based inspection of pipes; instead, this relies on the experience of trained inspectors that would assess the absolute and relative enveloped amplitudes of both T(0, 1) and F(1, 2) modes. In particular, typically PFs are characterised either by a high T(0, 1) component (for axisymmetric PFs) or a high F(1, 2) to T(0, 1) ratio (for non-axisymmetric PFs). An attempt is made to mimic such a procedure, in order to form a baseline performance against which to assess the ML algorithms developed in this work. To this scope, thresholding on three different enveloped signal amplitudes are considered, namely on (*Th1*) T(0, 1) only, (*Th2*) the F(1, 2) to T(0, 1) ratio, and (*Th3*) on the following linear combination of T(0, 1) and F(1, 2): *0.7 x T(0, 1) + 0.3 x (F(1, 2)/T(0,1))*. Note that the linear combination of *Th3* is an arbitrary attempt to combine both information of *Th1* and *Th2* based on the experience of the authors on this subject. Also note that such thresholding methods are applied to the same ~1-meter-long segments in which the experimental traces are split, as explained in the previous section, though in this case only the enveloped, non-normalised, DAC-corrected traces are used.

An example of application of the proposed thresholding approaches is given with the aid of Fig. 6, which shows a fabricated set of T (0, 1) and F(1, 2) traces including three PFs at the positions indicated by the red dashed vertical lines. The leftmost PF has a high T(0, 1) component, the middle one has both T(0, 1) and F(1, 2) components at rather high levels, while the third PF has a weaker T(0, 1) reflection, though with a F(1, 2) component nearly at the same level as T(0, 1). Only as a matter of example, when the threshold is set to 0.6 (shown with a dotted black horizontal line in figure), the use of T(0, 1) alone (i.e., of *Th1*) would not detect the small rightmost PF, while the use of the F(1, 2) to T(0, 1) ratio (i.e., of *Th2*) would be insensitive to the large symmetric leftmost PF. The proposed linear combination of T(0, 1) and F(1, 2) (i.e., of *Th3*) shown in green would instead detect all three PFs at the considered threshold.

2.2.3. *Machine learning*

2.2.3.1. Architectures. Three neural network architectures are considered to perform the task considered in this study, namely MLP,

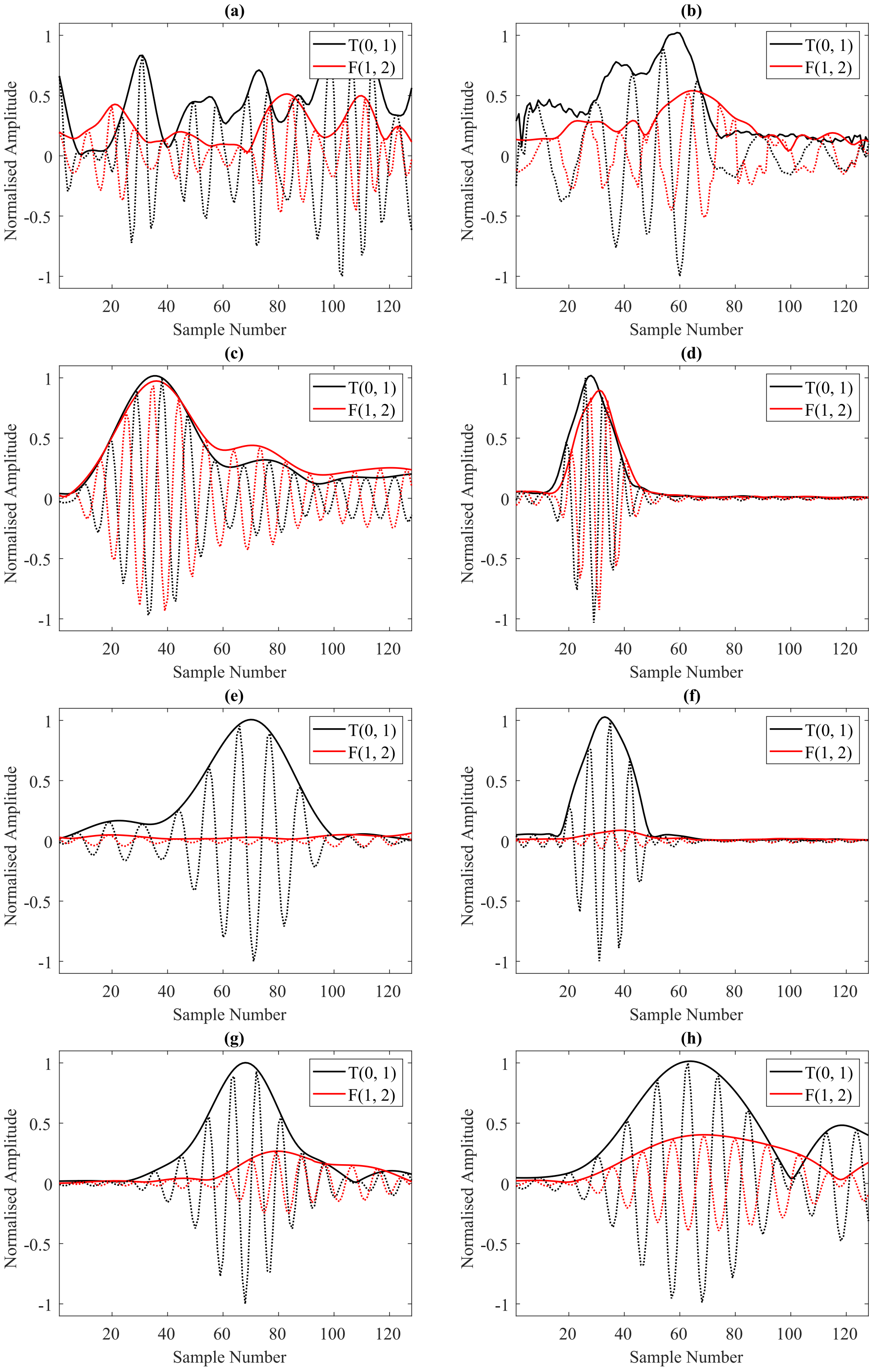

Fig. 5. Examples of the data samples used for training the ML models. (a) Experimental pristine, (b) simulated pristine, (c) experimental pitting-type corrosion defect, (d) simulated defect, (e) experimental weld, (f) simulated weld, (g) experimental flange and (h) experimental bend.

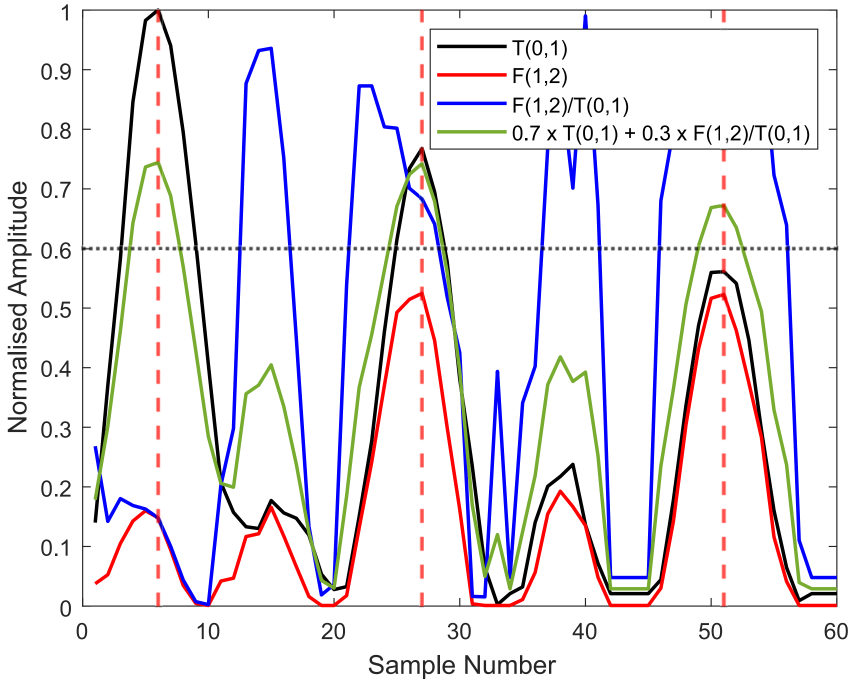

Fig. 6. Example of application of thresholding on a fabricated dataset of T(0, 1) and F(1, 2) traces. The locations of three PFs are indicated with red dashed lines and a tentative threshold is shown as a black dotted line. Thresholding is performed independently on the enveloped signals shown as black, blue and green solid lines. A successful detection occurs when any of these signals exceeds the threshold in the vicinity of a PF. Crossing the threshold away from PFs is a false positive, not crossing it in the vicinity is a false negative. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

VGG-Net, and U-Net. MLP is the traditional fully-connected neural network [24], therefore it is considered as the benchmark. The hyperparameters of the MLP network were tuned by sequentially increasing its complexity while assessing the performance, and the final implementation is given in Fig. 7. As the input data is three-dimensional: time, T(0, 1)/F(1, 2) and RF/envelope, it is first flattened to be input into a series of four dense layers, which are followed by a 40 % dropout layer and two additional dense layers. The activation function is ReLU [40]at all layers except from the output, where it is sigmoid.

VGG-Net is a convolutional neural network comprised of blocks of convolutional layers connected with Max Pooling layers [25]. Fig. 8 shows its final implementation, with the "feature extractor" section of the network only consisting of three blocks of two convolutional layers with input sizes sequentially reduced by a factor of 2, since deeper networks tended to overfit to the training set. The "classifier" portion is then similar to that used for MLP, with the addition of a flattening layer that makes the inputs compatible with fully connected layers.

U-Net was first introduced in 2015 for biomedical image segmentation [26], and was then used in a variety of contexts including non-destructive testing [41,42,43], the discriminator parts of generative adversarial networks [44,45], image denoising [46] or speech

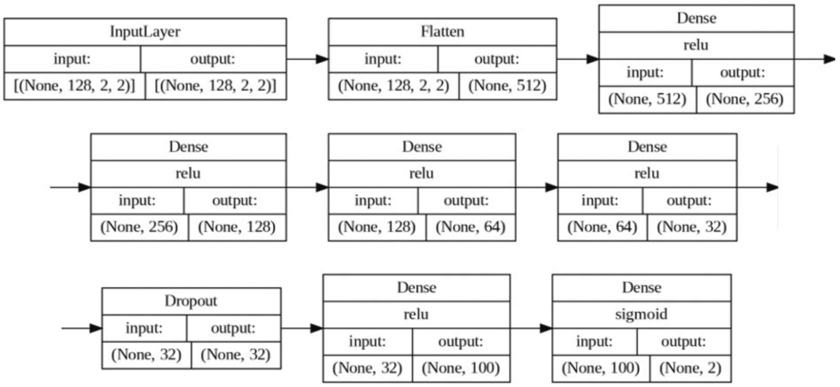

Fig. 7. MLP implementation utilised in this work. The first two rows represent the feature extractor section of the network, while the third row is the classifier head.

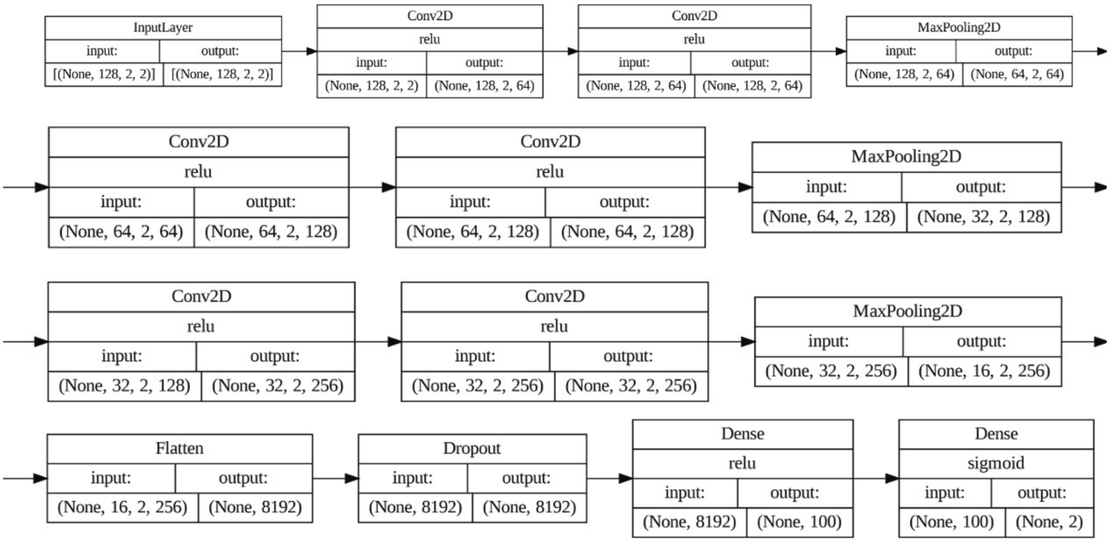
InputLayer
input: [(None, 128, 2, 2)]
output: [(None, 128, 2, 2)]
Conv2D relu
input: (None, 128, 2, 2)
output: (None, 128, 2, 64)
Conv2D relu
input: (None, 128, 2, 64)
output: (None, 128, 2, 64)
MaxPooling2D
input: (None, 128, 2, 64)
output: (None, 64, 2, 64)
Conv2D relu
input: (None, 64, 2, 64)
output: (None, 64, 2, 128)
Conv2D relu
input: (None, 64, 2, 128)
output: (None, 64, 2, 128)
MaxPooling2D
input: (None, 64, 2, 128)
output: (None, 32, 2, 128)
Conv2D relu
input: (None, 32, 2, 128)
output: (None, 32, 2, 256)
Conv2D relu
input: (None, 32, 2, 256)
output: (None, 32, 2, 256)
MaxPooling2D
input: (None, 32, 2, 256)
output: (None, 16, 2, 256)
Flatten
input: (None, 16, 2, 256)
output: (None, 8192)
Dropout
input: (None, 8192)
output: (None, 8192)
Dense relu
input: (None, 8192)
output: (None, 100)
Dense sigmoid
input: (None, 100)
output: (None, 2)

Fig. 8. VGG-Net architecture implemented for this work. Rows 1–3 represent the feature extractor section of the network, while row 4 is the classifier head.

enhancement [47]. The architecture is based on a contraction path followed by a symmetric expansion path. Given *n* layers in the contraction–expansion section of the architecture, the shapes of layers 1 and n, 2 and n-1… are identical which allows for the introduction of additive skip connections between them. This allows the architecture to maintain the feature extraction capability of a deep neural network while keeping the information directly contained in the raw data, such as the amplitude. In the original work by Ronneberger et al. the architecture design was motivated by its need to be trained on small number of annotated images [26], tackling the problem of segmentation. As the task required in this work is classification rather than segmentation, U-Net is used as the feature extraction element of the full architecture and that is followed by a set of fully connected layers to perform classification. In particular, the implemented contraction path for U-Net was identical to the VGG-Net presented in Fig. 8. That was followed first by the corresponding symmetric expansion path, then by the classification head, which again was identical to that used for VGG-Net and shown in Fig. 8.

2.2.3.2. Training design. This work is designed to test whether the addition of simulated data in the process of training any of the ML algorithms discussed above improves its performance for the detection of PFs in actual pipe inspection data. To this end, a transfer learning procedure between simulated and experimental data is implemented by devising the ML training as a two-stage process in which each architecture is first pre-trained on simulated data and is then fine-tuned on experimental data, with the model retained at the end of each training stage being that giving the lowest validation loss. The final trained network is then evaluated on a testing set formed by randomly selecting 480 experimental samples (i.e., 20 % of the full real dataset) among those not used for training or validation, and with a prescribed class balance of approximately 3:1 between negative and positive samples (355 and 125, respectively).

In order to assess the impact of the amount of real and simulated training data on the performance achieved by each algorithm, all architectures are trained on 16 different combinations of real and simulated dataset sizes, as detailed in Table 3 where each combination is denoted “ID”, and are evaluated on the same testing set. For each ID, the training is repeated 5 times on different randomly selected simulated and real training sets (always excluding the real samples used in the testing set), with the validation sets drawn at random as 20 % of their respective training sets, in the attempt to incorporate the effects of random factors affecting the training procedure, such as the influence of the specific training set being used and of the random initialisation of all parameters.

Table 4 lists some additional training parameters that are kept constant across all ML training. The numbers of epochs set for training on both simulated and real datasets are selected as to be sufficiently large for the training and validation loss curves to plateau for the largest datasets used in the study. Binary cross-entropy is used as a standard loss function for binary classification problems [48]. Adam optimiser [49] is selected as it is used in the majority of the novel ML research (over 320 times more prevalent than

Table 3
Combinations of real (nReal) and simulated (nSims) training dataset sizes used in this study.

ID	nReal	nSim	ID	nReal	nSim	ID	nReal	nSim	ID	nReal	nSim
1	0	0	5	270	0	9	810	0	13	1800	0
2	0	900	6	270	900	10	810	900	14	1800	900
3	0	2700	7	270	2700	11	810	2700	15	1800	2700
4	0	10,000	8	270	10,000	12	810	10,000	16	1800	10,000

Table 4
ML training parameters.

Parameter	Value
Number of epochs on simulation set	30
Number of epochs on real set	30
Loss function	Binary Cross-entropy [48]
Optimiser	Adam [49]
Learning rate	5e-5 initially, exponentially decaying
Kernel Size	(60, 2)

Adafactor, the second most common optimiser [50]). Adam is a self-tuning optimiser, therefore the user only needs to select an initial learning rate before that is tuned internally during training. The initial learning rate is set to 5e-5 after some trial and error. Finally, the kernel size is set to (60, 2). The second dimension is set to 2 so T(0, 1) and F(1, 2) are considered for each convolution, while the length-dimension of 60 proved to perform best experimentally in the 10–100 range.

2.3. Metrics

The metrics used for the evaluation of the performance of a binary classifier vary widely. The ones used in this work are Area Under Receiver Operating Curve (AUROC) and False Positive Rate (FPR) at 99.7 % True Positive Rate (TPR), which in the remainder of the article is referred to as FPR@1TPR. The Receiver Operating Curve (Fig. 9) plots TPR vs FPR while sweeping the detection threshold [51]. As such, it has a monotonic trend between points (0, 0) and (1, 1), corresponding to a threshold that classifies every sample as negative and a threshold that classifies every sample as positive, respectively. The point corresponding to a perfect classifier is (0,1). Note that both FPR and TPR can be either expressed in the 0 to 1 range or, equivalently, in percentage between 0 and 100 %. AUROC as a metric has the advantage of being agnostic to the selected threshold, as well as taking into consideration both TPR and FPR, both of which are of interest in a defect detection problem. However, as explained in the Section “Guided wave testing of pipes”, the objective of this study is to achieve a very high TPR value, i.e., to only miss very few true PFs, such that virtually all data related to pipe PFs would be detected and shown to a human inspector. This motivated the choice of using FPR@1TPR as an additional metric. It must be noted that since the experimental testing set only includes a small number of positive samples, the FPR@1TPR value obtained from any trained algorithm is highly dependent on the smallest value assigned to a positive sample.

3. Results

3.1. Performances of the proposed algorithms

The performance offered by the three thresholding approaches and by the three ML architectures when trained with various sizes of simulated and real data is discussed below. Table 5 gives AUROC and FPR@1TPR when thresholding is applied to all available 2400

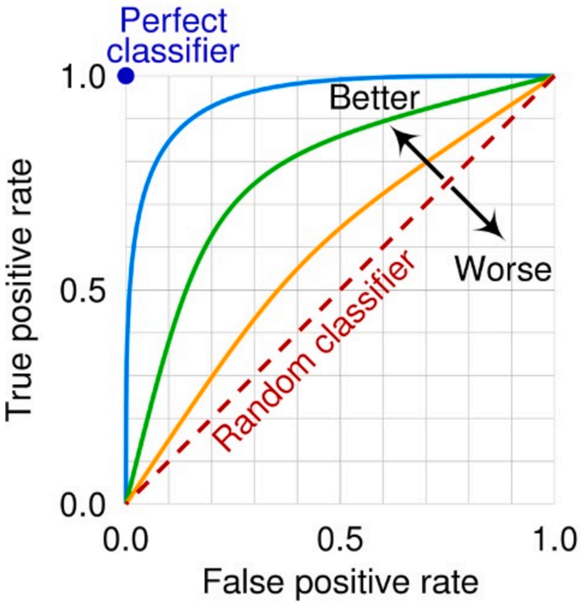

Fig. 9. Schematic description of Receiver Operating Curves.

experimental data samples. The best results overall are obtained when using *Th1*, i.e., when thresholding on T(0, 1) only. However, despite its relatively high AUROC sized at 0.9912, the FPR@1TPR at 0.4246 essentially indicates that there exist some PFs giving low T (0, 1) reflections which can only be flagged by thresholds that would also call an excessively large number of false positives, i.e., almost one false positive for every two negative samples. The extremely poor performance of *Th2* shows that the sole use of F(1, 2) cannot reliably discriminate between positive and negative samples. This is expected, since all axisymmetric PFs such as welds do not produce significant non-axisymmetric reflections. Finally, the attempt to combine the amplitudes of T(0, 1) and F(1, 2) into a single parameter via *Th3* did not yield improvements over the sole use of T(0, 1), as both AUROC and FPR@1TPR for *Th3* are slightly worse than those of *Th1*. These results essentially suggest that the experience of trained operators on the analysis of signals acquired by GWT of pipes cannot be replaced by an approach that only makes use of the amplitudes of T(0, 1) and F(1, 2) time-traces, and that their shapes should also be considered.

Fig. 10 gives the performances of the three ML architectures evaluated on the same, randomly selected testing set of 480 experimental samples and across the 16 combinations of training and testing set sizes (IDs), as given in Table 3. For each ID, the metrics obtained from the 5 separate training instances are summarized in a boxplot where the central mark indicates the median, while the bottom and top edges indicate the 25th and 75th percentiles, respectively. Fig. 10(a and b) show that MLP does not possess the required modelling capabilities to reliably characterise true and negative samples, and despite the expected general improvement as more simulated and real input data are employed for its training, its best overall performance remains significantly inferior to that of *Th1* in terms of both AUROC and FPR@1TPR. It is worth emphasizing here that the input to MLP and the other ML architectures significantly differ to that of thresholding, as in the latter case samples are not normalised.

Fig. 10(c and d) shows that when VGG-Net is solely trained on simulated data (i.e., IDs 1 to 4), the performance on experimental data is extremely poor. When, instead, a relatively low number of 270 real samples are used for fine-tuning (IDs 5 to 8), both AUROC and FPR@1TPR gradually improve as the amount of simulated data is increased. Then, once a more substantial number of real samples are available, the benefit of pre-training the model with simulated data starts to fade. In fact, the performance at IDs 13 to 16, where 1800 real samples are used, is essentially flat as the number of simulated samples is increased from 0 to 10000. Such performance is significantly superior to that offered by thresholding on T(0, 1) (i.e., *Th1*), with AUROC surpassing 99.8 % and FPR@1TPR yielding 2.1 %. In order to investigate whether this overall behaviour and quality of performance are specific to the particular testing set used to produce Fig. 10, the same procedure is repeated in five more instances by randomly selecting five additional testing sets, and the resulting FPR@1TPR are displayed in the left column of Fig. 11. All plots confirm the conclusions drawn above. It is also worth noting that for each of the six testing sets depicted in Fig. 10(c and d) and in the left column of Fig. 11, when 1800 real samples are used for training (IDs 13 to 16) the use of simulated data has a negligible effect, whereby when only up to 810 real samples are available, the performance of the classifier can be improved by the addition of simulated samples. This is particularly evident in Fig. 11(c). Unsurprisingly, the plots also show that the FPR@1TPR varies as different testing sets are evaluated, and it ranges between ∼1.5 and ∼4 % when 1800 real samples are used for training.

Finally, Fig. 10(e and f) shows that the employed U-Net algorithm struggles to yield a consistent performance, which, on average, unexpectedly deteriorates as the number of real samples used for the fine-tuning of the model is increased. This is confirmed by the results shown in the right column of Fig. 11, where U-Net is tested on the same five additional testing sets described above. Only in one instance, i.e., the case of Fig. 11(j), the expected behaviour is obtained, and a similar performance as VGG-Net is achieved. This unpredictable behaviour of U-Net was extensively investigated as the hyperparameters of the model were sequentially varied, though the investigation remained inconclusive and VGG-Net was chosen to carry out the study described in the next section. It is worth noting, however, that U-Net provides the overall best performance when only simulated data are used (i.e., IDs 1 to 4).

3.2. Investigation on misclassified pipe features

It is of practical interest to investigate which PFs are more difficult to identify correctly using the VGG-Net model trained with the largest sets of real and simulated data considered in this work (i.e., on the scenario indicated as ID = 16). In order to make this study statistically more relevant, training and testing on VGG-Net at ID = 16 was performed for 44 additional randomly selected testing sets. Again, for each testing set the training was repeated five times. The full database obtained by adding these additional results to those described in the previous section consists of 50 different testing sets, each including 125 PFs selected at random among the available 634, with Table 6 giving statistics on the numerosity of the various types of PFs included in each testing set. Since each testing set is evaluated by five independently trained models, this equates to 31,250 real positive samples being tested. A study on the composition of this ensemble of experimental samples shows that each of the 634 PFs appears at least 20 times.

The rule that is used to determine the threshold against which to mark the false negatives in each of the 250 sets of results (as obtained from the 50-by-5 VGG-Net training and testing instances) is to set it to the value giving a FPR of 2 % in the specific set. Following this procedure, a total of 784 positive samples (∼2.5 % of 31250) are misclassified as negative, with 32 out of the total 634 PFs (∼5 %) being misclassified at least once. The composition of this set of 32 PFs is given in Table 7, where PFs are listed in descending

Table 5
Performance of the three proposed thresholding approaches across all 2400 experimental data samples.

	Th1	Th2	Th3
AUROC	0.9912	0.2769	0.9662
FPR@1TPR	0.4246	0.9927	0.4732

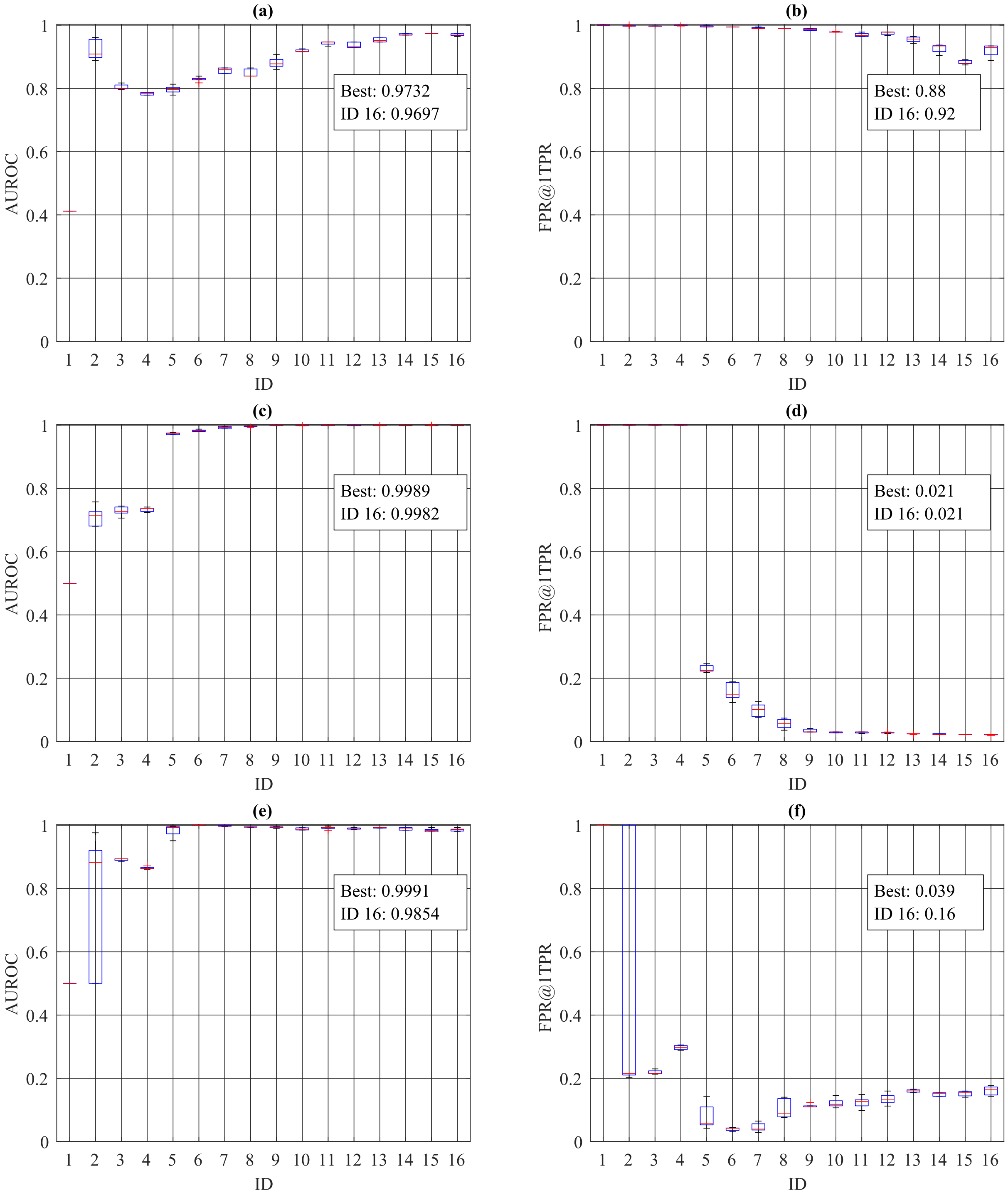

Fig. 10. AUROC of MLP (a), FPR@1TPR of MLP (b), AUROC of VGG-Net (c), FPR@1TPR of VGG-Net (d), AUROC of U-Net (e), FPR@1TPR of U-Net (f). On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the '+' marker symbol.

order according to the ratio between misclassified cases and their total number. The table shows that false echoes, defects and supports are the PFs that are most liable to be misclassified based on their signatures on T(0, 1) and F(1, 2) signals. This can be easily explained, as false echoes are often marked by operators based on the presence of large PF reflections in the time-traces propagating in the other

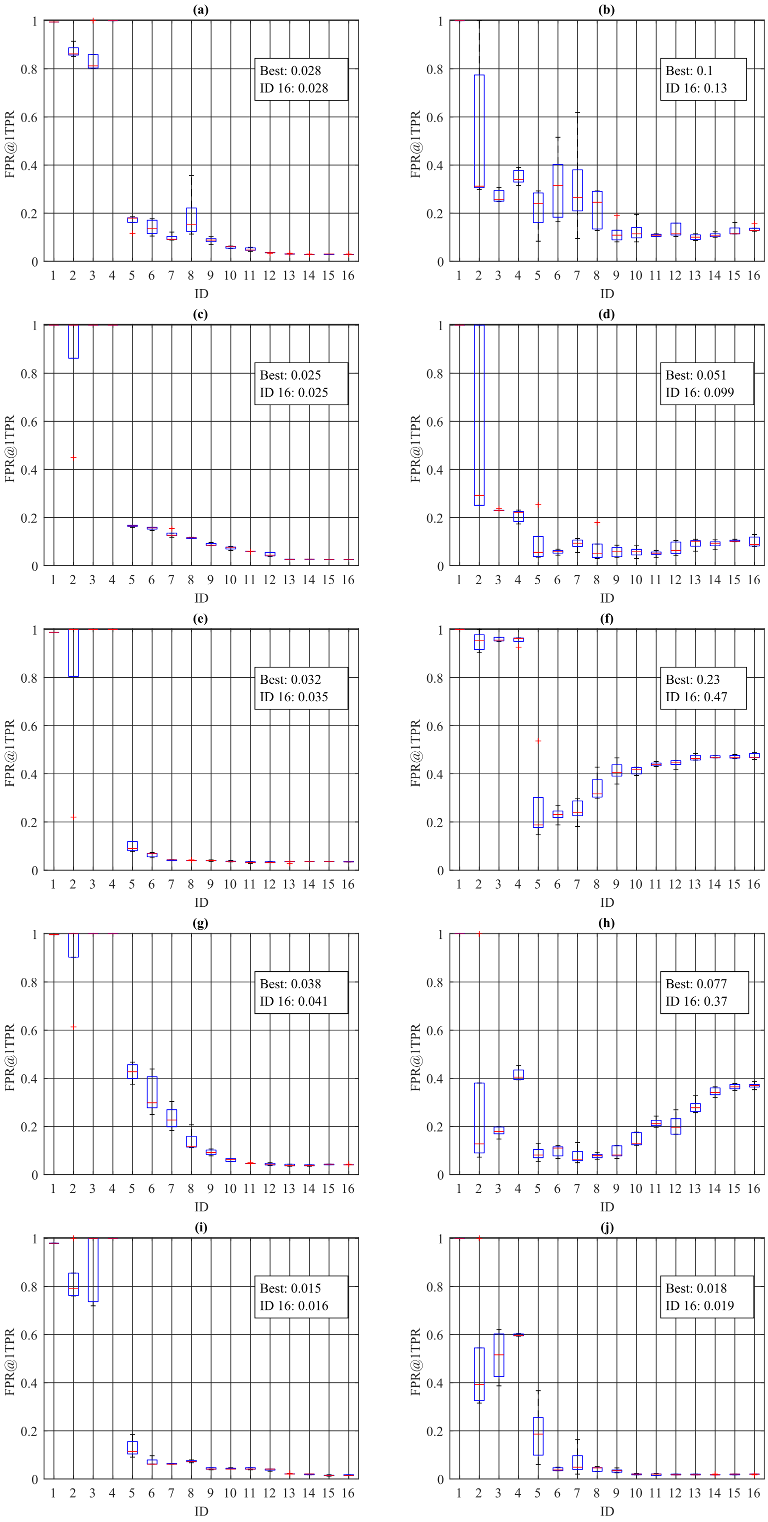

Fig. 11. FPR@1TPR as VGG-Net (a, c, g, e, i) and U-Net (b, d, f, h, j) are evaluated on five different testing sets. The same testing set is used in each row of plots. Boxplots are as in Fig. 9.

Table 6
Statistics on the numerosity of the various types of PFs included in each of the 50 testing sets.

Pipe Feature Type	Mean	Standard Deviation	Maximum	Minimum
Weld	59.08	6.01	69	45
Support	34.28	4.83	48	24
Defect	11.3	3.14	19	6
Bend	8.16	1.98	12	4
Flange	6.84	2.39	13	2
False echo	5.8	2.17	11	2
Unidentified anomaly	2.22	1.40	6	0
Entrance into earth	0.94	0.91	3	0

direction of test, information that is not given to VGG-Net. Similarly, often supports give very weak signal reflections, and they are marked based on their physical presence as confirmed by visual inspections. Finally, reflections from very small defects can be barely distinguishable from background noise, which may force an operator to conservatively flag an indication despite significant uncertainty. On the other side, welds, bends, flanges and entrances into earth usually give very distinctive and significant reflections, hence the low probability of VGG-Net missing any of them.

With regards to the 32 PFs that have been misclassified at least once, it is of interest to investigate those with a high prevalence rate (i.e., the number of times a sample has been misclassified divided by the number of times it has been tested), since those with low prevalence can be filtered out via ensemble methods [52] The attention here is devoted to defects, since they are the PFs of most concern if missed. Of the 7 misclassified defects listed in Table 7, only two have prevalence rates exceeding 20 %, and their signatures are shown in Fig. 12. Interestingly, they both come from the same inspection, whose WavePro [33] trace is displayed in Fig. 13. The inspection of the original trace shows that the marked position of defect (a) (denoted as + F6 by the operator) is slightly off and most of the major F(1, 2) signature of the defect is absent from the trace presented to the ML algorithms. Notably, when the windowing segment labelled (a) in Fig. 13 is moved ~0.6 m to the left, and when such better-centred version of this defect is fed to all VGG-Net models used in this investigation at the set thresholds corresponding to 2 % FPR, a 100 % true positive rate is obtained. More challenging, instead, is to find reasons for the consistent misclassification of the defect signature shown in Fig. 12(b) as well as in the red box labelled (b) in Fig. 13, which is characterised by relatively similar levels of T(0, 1) and F(1, 2) reflections. This highlights one of the fundamental issues plaguing virtually all ML algorithms currently used worldwide, i.e., their black-box nature, and therefore suggests to focus on ML explainability [53] for future research.

4. Conclusion

This article has proposed a transfer learning framework that allows augmenting an experimental dataset collected via GWT of pipes with synthetic data produced via FEM in order to train a ML algorithm to inspect the signals and to detect reflections from pipe features. The transfer learning between synthetic and real data is achieved by first pre-training the chosen ML model on the simulations, and then fine-tuning it via additional training on the set of experimental signals. In particular, three types of ML models have been considered for the task, namely MLP, VGG-Net and U-Net, and their performances have been also compared to those given by classical thresholding approaches. Unexpectedly, VGG-Net was found to yield more consistent results than U-Net, while they both significantly outperformed MLP and thresholding. Restricting the analysis to the VGG-Net results, the investigation has shown that when scarce amounts of real data are available, significant gains in the detection performance can be obtained by employing the suggested pre-training approach. In particular, the performance monotonically improves and eventually plateaus as the synthetic dataset is increased, at which point further improvements can only be obtained by enlarging the size of the experimental dataset. However, once a sufficiently large number of real data are available for training, the benefit of pre-training the model on simulations starts to fade. This was shown to occur for this particular application once 1800 data samples (roughly equivalent to 1800 m of inspected pipes) are fed to the VGG-Net model, at which point false positive rates in the order of ~1.5 to 4 % at the fixed true positive rate of 99.7 % are achieved. From a practical standpoint, the adoption of this model would greatly reduce the amount of data that needs

Table 7
Number and percentage of misclassified PFs according to the specific PF type, using the VGG-Net model trained with 10,000 simulated and 1800 experimental samples (i.e., the scenario indicated as ID = 16).

	Misclassified cases (at least once)	Total number	Percentage (%)
False Echo	4	28	14.3
Defect	7	58	12.1
Support	17	164	10.4
Weld	4	293	1.4
Bend	0	41	0
Flange	0	36	0
Unidentified anomaly	0	10	0
Entrance into earth	0	4	0

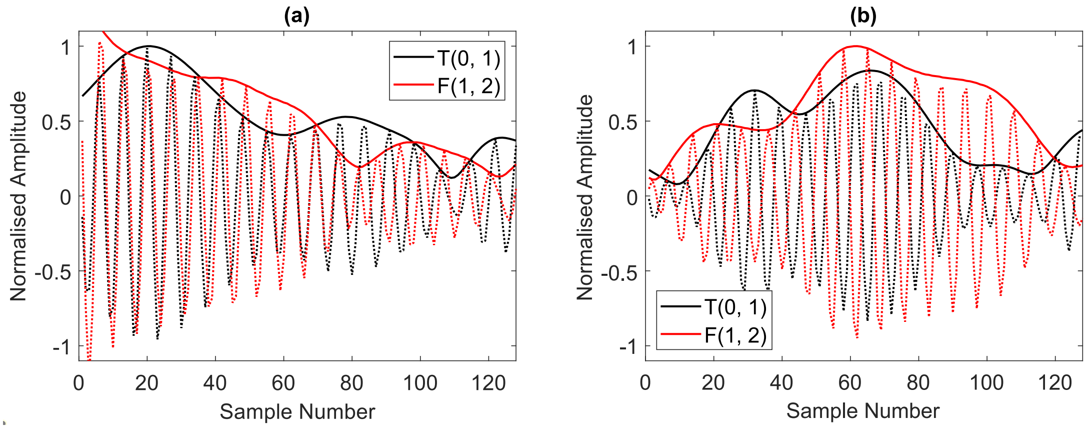

Fig. 12. Samples of defects that are consistently misclassified as negative samples by VGG-Net (prevalence rates of 91 %(a) and 98 %(b)).

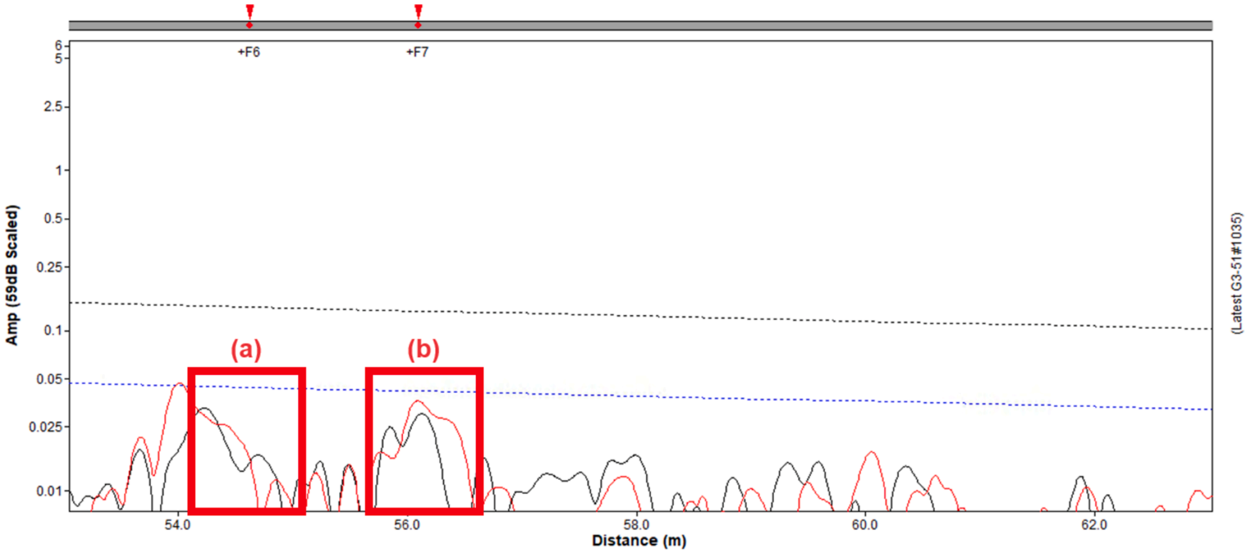

Fig. 13. Portion of the original inspection trace containing the defects shown in Fig. 11. Defects are marked as + F6 and + F7 on the trace and they are located at a distance of approximately 54 and 56 m from the sensor position. The segments of trace fed to the ML algorithms are highlighted in red boxes. WavePro, courtesy of (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
Reproduced from [25].

to be manually inspected by qualified operators for the classification of the detected pipe features. Future studies will focus on automatising this subsequent classification step, although it is expected that additional experimental samples would be required to better characterise the signatures of those pipe features that are less frequently found, such as flanges, bends, and, most importantly, defects.

A further investigation was carried out to pinpoint specific pipe features that VGG-Net would consistently miss. They included two of the 58 defects available in the experimental dataset, although it was later found that the position of one of the two had been misreported by the inspector in the original inspection trace. Once that defect position was corrected, VGG-Net was actually able to detect it. It is more concerning instead that all efforts devoted to understand why the other defect was left undetected remained inconclusive. This highlights the importance of being able to explain the decisions made by any given ML model, therefore suggesting to focus on ML explainability as a further potential avenue of future research in this project.

CRediT authorship contribution statement

Mikolaj Mroszczak: Writing – original draft, Visualization, Software, Investigation, Formal analysis, Data curation. **Robin E. Jones:** Writing – review & editing, Supervision. **Peter Huthwaite:** Writing – review & editing, Supervision, Resources, Project

administration, Methodology, Funding acquisition, Conceptualization. **Stefano Mariani:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Robin E. Jones reports a relationship with Guided Ultrasonics Limited that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The research has been funded jointly by EPSRC grant number EP/S023275/1 and Guided Ultrasonics Ltd.

Data availability

Data will be made available on request.

References

[1] X. Zang, Z.-D. Xu, H. Lu, Z. Chen, Z. Zhenwu, Ultrasonic guided wave techniques and applications in pipeline defect detection: a review, Int. J. Press. Vessel. Pip. 206 (2023).
[2] D.N. Alleyne, B. Pavlakovic, M.J.S. Lowe, P. Cawley, Rapid long-range inspection of chemical plant pipework using guided waves, in: AIP Conference Proceedings, 2001, pp. 180–187.
[3] F. Lyu, X. Zhou, Z. Ding, X. Qiao and D. Song, "Application Research of Ultrasonic-Guided Wave Technology in Pipeline Corrosion Defect Detection: A Review," Coatings, vol. 14, no. 3, 2024.
[4] A. Demma, D. Alleyne, B. Pavlakovic, Testing of buried pipelines using guided waves, Bahrain, Manama, 2005.
[5] Guided Ultrasonics Ltd., "EFC Solid Rings," 2023. [Online]. Available: https://www.guided-ultrasonics.com/product/efc-solid-rings/. [Accessed 19 Jun 2023].
[6] C. Janiesch, P. Zschech, K. Heinrich, Machine learning and deep learning, Electron. Mark. 31 (3) (2021) 685–695.
[7] I. Virkkunen, T. Koskinen, O. Jessen-Juhler, J. Rinta-aho, Augmented Ultrasonic Data for Machine Learning, J. Nondestr. Eval. 40 (2021).
[8] Medical Imaging Technology Association, "DICOM - Digital Imaging and Communication in Medicine," 2023. [Online]. Available: https://www.dicomstandard.org. [Accessed 19 Jun 2023].
[9] ASTM, Standard Practice for Digital Imaging and Communication in Nondestructive Evaluation (DICONDE), Conshohocken: ASTM International, 2023.
[10] S. Uhlig, A. Ikin, S. Frank, T. Constanze, M. Wolff, A review of synthetic and augmented training data for machine learning in ultrasonic non-destructive evaluation, Ultrasonics 134 (2023).
[11] S. McKnight, S.G. Pierce, E. Mohseni, C. MacKinnon, C. MacLeod, T. O'Hare, C. Loukas, A comparison of methods for generating synthetic training data for domain adaption of deep learning models in ultrasonic non-destructive evaluation, NDT&E International 141 (2024).
[12] R.J. Pyle, R.L.T. Bevan, R.R. Hughes, R.K. Rachev, A. Ait Si Ali, P.D. Wilcox, Deep learning for ultrasonic crack characterization in NDE, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 68 (5) (2020) 1854–1865.
[13] P. Cawley, F. Cegla, A. Galvagni, Guided waves for NDT and permanently-installed monitoring, Insight 54 (11) (2012) 594–601.
[14] N.J. Shipway, P. Huthwaite, M.J.S. Lowe, T.J. Barden, Using ResNets to perform automated defect detection for Fluorescent Penetrant Inspection, NDT and E Int. 119 (2021).
[15] R.J. Pyle, R.L.T. Bevan, R.R. Hughes, A.A.S. Ali, P.D. Wilcox, Domain adapted deep-learning for improved ultrasonic crack characterization using limited experimental data, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 69 (4) (2022) 1485–1496.
[16] S. Hofer, K. Bekris, A. Handa, J. C. Gamboa, M. Mozifian, F. Golemo, D. Fox, K. Goldberg, J. Leonard, C. Karen Liu, J. Peters, S. Song, P. Welinder and M. White, "Sim2Real in Robotics and Automation: Applications and Challenges," IEEE Transactions on Automation Science and Engineering, vol. 18, no. 2, 2021.
[17] K. Supreet Alguri, J.B. Harley, Transfer learning of ultrasonic guided waves using autoencoders: A preliminary study. In AIP Conference Proceedings, 2019.
[18] K. Supreet Alguri, C. C. Chia and J. B. Harley, "Sim-to-Real: Employing ultrasonic guided wave digital surrogates and transfer learning for damage visualization," Ultrasonics, vol. 111, 2021.
[19] B. Zhang, X. Hong and Y. Liu, "Multi-Task Deep Transfer Learning Method for Guided Wave-Based Integrated Health Monitoring Using Piezoelectric Transducers," IEEE Sensors Jourlan, vol. 20, no. 23, 2020.
[20] W. Liu, Z. Tang, F. Lv, X. Chen, An efficient approach for guided wave structural monitoring of switch rails via deep convolutional neural network-based transfer learning, Meas. Sci. Technol. 34 (2022).
[21] B. Zhang, X. Hong, Y. Liu, Distribution adaptation deep transfer learning method for cross-structure health monitoring using guided waves, Struct. Health Monit. 21 (3) (2022) 757–1308.
[22] S. Cantero-Chinchilla, P.D. Wilcox, A.J. Croxford, Deep learning in automated ultrasonic NDE – Developments, axioms and opportunities, NDT&E International 131 (2022).
[23] Y. Ying, J.H. Garrett, I.J. Oppenheim, L. Soibelman, J. Harley, J. Shi, Y. Jin, Toward data-driven structural health monitoring: application of machine learning and signal processing to damage detection, J. Comput. Civ. Eng. 27 (6) (2013) 667–680.
[24] F. Murtagh, Multilayer perceptrons for classification and regression, Neurocomputing 2 (5–6) (1991) 183–197.
[25] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in International Conference on Learning Representation, San Diego, 2014.
[26] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional Networks for Biomedical Image Segmentation, MICCAI 2015 (2015) 234–241.
[27] D.N. Alleyne, P. Cawley, The excitation of Lamb waves in pipes using dry-coupled piezoelectric transducers, J. Nondestr. Eval. 15 (1996) 11–20.
[28] K.F. Graff, Wavemotion in elastic solids, Ohio State University Press, Belfast, 1975.
[29] D.N. Alleyne, T. Vogt, P. Cawley, The choice of torsional or longitudinal excitation in guided wave pipe inspection, Insight 51 (7) (2009) 373–377.
[30] P. Cawley, M. Lowe, D. Alleyne, B. Pavlakovic, P. Wilcox, Practical long range guided wave inspection-applications to pipes and rail, Mater. Eval. 61 (1) (2003) 66–74.
[31] Guided Ultrasonics Limited, "GUL Homesite," 2023. [Online]. Available: https://www.guided-ultrasonics.com. [Accessed 19 Jun 2023].
[32] D.N. Alleyne, M.J.S. Lowe, P. Cawley, The reflection of guided waves from circumferential notches in pipes, J. Appl. Mech 65 (3) (1998) 635–641.
[33] Guided Ultrasonics Ltd., "WavePro4 Software," 2023. [Online]. Available: https://www.guided-ultrasonics.com/product/wavepro4-software/. [Accessed 18 July 2023].
[34] F. C. R. Marques and A. Demma, "Ultrasonic guided waves evaluation of trials for pipeline inspection," in 17th World Conference on Nondestructive Testing, Shanghai, 2008.
[35] P. Huthwaite, Accelerated finite element elastodynamic simulations using the GPU, J. Comput. Phys. 257 (2014) 687–707.
[36] J.R. Pettit, A. Walker, P. Cawley, M. Lowe, A Stiffness Reduction Method for efficient absorption of waves at boundaries for use in commercial Finite Element codes, Ultrasonics 54 (7) (2014) 1868–1879.
[37] S. Mariani, S. Heinlein, P. Cawley, Compensation for temperature- dependent phase and velocity of guided wave signals in baseline subtraction for structural health monitoring, Struct. Health Monit. 19 (1) (2020) 26–47.

[38] J. Davies, P. Cawley and M. J. S. Lowe, “Long Range Guided Wave Pipe Inspection – the Advantages of Focusing,” 17th World Conference on Nondestructive Testing, vol. 6, p. 6, 2008.
[39] The Mathworks Inc., “resample,” 2023. [Online]. Available: https://www.mathworks.com/help/signal/ref/resample.html. [Accessed 10 November 2023].
[40] A. F. Agarap, “Deep learning using rectified linear units (relu),” arXiv preprint arXiv:1803.08375, 2018.
[41] L. Yang, H. Wang, B. Huo, F. Li, L. Yanhong, An automatic welding defect location algorithm based on deep learning, NDT and E Int. 120 (2021).
[42] B.C.F. Oliveira, A.A. Seibert, V.K. Borges, A. Albertazzi, R.H. Schmitt, Employing a U-net convolutional neural network for segmenting impact damages in optical lock-in thermography images of CFRP plates, Nondestructive Testing and Evaluation 36 (4) (2021) 440–458.
[43] Q. Luo, B. Gao, W.L. Woo, Y. Yang, Temporal and spatial deep learning network for infrared thermal defect detection, NDT and E Int. 108 (2019).
[44] C. Wu, Y. Zou, Z. Yang, U-GAN: Generative Adversarial Networks with U-Net for Retinal Vessel Segmentation, in: 2019 14th International Conference on Computer Science & Education (ICCSE), 2019, pp. 642–646.
[45] E. Schonfeld, B. Schiele, A. Khoreva, A U-Net Based Discriminator for Generative Adversarial Networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8207–8216.
[46] R. Komatsu, T. Gonsalves, Comparing U-Net Based Models for Denoising Color Images, AI 1 (4) (2020) 465–486.
[47] R. Giri, U. Isik and A. Krishnaswamy, “Attention Wave-U-Net for Speech Enhancement,” 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp. 249-253, 2019.
[48] F. Topsoe, “Bounds for entropy and divergence for distributions over a two-element set.,” JIPAM. Journal of Inequalities in Pure & Applied Mathematics, vol. 2, no. 2, pp. Paper No. 25, 13 p., 2001.
[49] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, San Diego (2015).
[50] Meta AI, “SGD with Momentum,” 2023. [Online]. Available: https://paperswithcode.com/method/sgd-with-momentum. [Accessed 1 November 2023].
[51] A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recogn. 30 (7) (1997) 1145–1159.
[52] T.G. Dietterich, Ensemble Methods in machine learning, Heidelbeg, Berlin, 2000.
[53] G. Vilone, L. Longo, Notions of explainability and evaluation approaches for explainable artificial intelligence, Information Fusion 76 (2001) 89–106.