Mechanical Systems and Signal Processing 197 (2023) 110360

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

# A probabilistic framework for source localization in anisotropic composite using transfer learning based multi-fidelity physics informed neural network (mfPINN)

Nur M.M. Kalimullah [a], Amit Shelke [a,*], Anowarul Habib [b]

[a] Department of Civil Engineering, Indian Institute of Technology Guwahati, Assam 781039, India
[b] Department of Physics and Technology, UiT The Arctic University of Norway, 9037 Tromsø, Norway

ARTICLE INFO

Communicated by Wei-Xin Ren

*Keywords:*
Impact detection
Multi-fidelity
Neural network
Physics-informed
Transfer learning

ABSTRACT

The practical application of data-driven frameworks like deep neural network in acoustic emission (AE) source localization is impeded due to the collection of significant clean data from the field. The utility of the such framework is governed by data collected from the site and/or laboratory experiment. The noise, experimental cost and time consuming in the collection of data further worsen the scenario. To address the issue, this work proposes to use a novel multi-fidelity physics-informed neural network (mfPINN). The proposed framework is best suited for the problems like AE source detection, where the governing physics is known in an approximate sense (low-fidelity model), and one has access to only sparse data measured from the experiment (high-fidelity data). This work further extends the governing equation of AE source detection to the probabilistic framework to account for the uncertainty that lies in the sensor measurement. The mfPINN fuses the data-driven and physics-informed deep learning architectures using transfer learning. The results obtained from the data-driven artificial neural network (ANN) and physics-informed neural network (PINN) are also presented to illustrate the requirement of a multi-fidelity framework using transfer learning. In the presence of measurement uncertainties, the proposed method is verified with an experimental procedure that contains the carbon-fiber-reinforced polymer (CFRP) composite panel instrumented with a sparse array of piezoelectric transducers. The results conclude that the proposed technique based on a probabilistic framework can provide a reliable estimation of AE source location with confidence intervals by taking measurement uncertainties into account.

## 1. Introduction

Anisotropic plates and shells have become the prominent choice in the aero-industry, auto-mobile industry and wind turbines due to their exceptional stiffness and low weight. Despite their tremendous applications, several issues have been highlighted about the composites' structural integrity, such as debonding, delamination, and matrix cracking, to mention a few. Such complications are caused primarily due to shock, impact and repeated cyclic stresses. The multiple occurrences of these failure modes can lead to a catastrophic failure or subsequent degradation of the structure's performance [1,2]. Focusing towards predictive maintenance, it is imperative to detect the location of the impact and scrutinise the damage in the vicinity of the impact. The current research presents a

* Corresponding author.
*E-mail address:* amitsh@iitg.ac.in (A. Shelke).

https://doi.org/10.1016/j.ymssp.2023.110360
Received 14 October 2022; Received in revised form 4 April 2023; Accepted 8 April 2023
Available online 17 April 2023

novel framework based on deep learning to identify the impact location in the composite structure.

Structural health monitoring (SHM) has emerged on diagnosis, etiognosis, and prognosis to reduce life cycle costs [3,4]. In the SHM, the sparse array of ultrasonic transducers utilized guided ultrasonic waves (GUWs) both in active and passive modes for defect detection and its localisation at the microscale [5]. The acoustic actuators are the source of ultrasonic energy in active monitoring [6] while in passive monitoring, the impact due to foreign object acts as acoustic emission [7,8]. The permanently installed transducers in the structure can be used to perform real-time health monitoring as well as condition-based monitoring.

Further, the capability of active and passive sensing of the ultrasonic transducers can be leveraged to detect active damage and pre-existing cracks. This study utilises the passive mode of acoustic emission (AE) data to localise the impact and/or active damages. For isotropic plates, a minimum number of three sensors are sufficient to localise the AE source using the triangulation technique. However, the triangulation technique does not work in the case of an anisotropic plate, as the wave velocity varies in different directions [9,10]. For composite plates, the stacking sequence of the composite layers governs the spatial velocity profile. In the past decades, several works related to localising AE sources have been found in the literature [11–14]. Based on the above discussions, two salient steps are traditionally followed to localise the AE source.

- First, AE data acquisition, i.e., using the ultrasonic transducers, the time of flight (TOF) measurements are acquired from distinct locations.
- Second, using an inversion algorithm, i.e., the AE data are incorporated to solve a set of non-linear equations deterministically [15] or well-defined loss functions are minimised by an optimisation algorithm [9,10,12,13,16] with some exceptions [17,18].

One must note that in these approaches, the computational costs increase rapidly with the increase in the number of sensors. Further, the accuracy of the AE source estimation depends upon the precision of the TOF measurements. Even a small perturbation of uncertainty in TOF measurement may lead to a considerable error in estimating the AE source. Usually, the common roots of uncertainties are ambient conditions, mode conversion, dispersion phenomenon, systemic error, and noise, to mention a few.

The emerging engineering systems are undergoing a renaissance, the first principle models guiding the way towards a data-driven approach. The confluence of big data, artificial intelligence and machine learning has gained lots of traction across the science and engineering domains and is driving a paradigm shift [19,20]. Even in the domains where the physics-based governing equations prevail, researchers are gradually inclining toward data-driven analysis. Artificial neural networks (ANN) [21,22], and the Gaussian process [23] have been found in the literature for AE source localisation in complex structures. However, in engineering applications, these approaches suffer from three significant challenges. First, data acquisition is often expensive for complex engineering systems; thereby, ANN has access to very sparse data to train the network in such a scenario. Secondly and perhaps more importantly, the data-driven algorithms have inadequate predictive capability outside the training domain as opposed to the physics-based model. For instance, if the training data are generated within a certain physical domain of a plate in the AE location detection problem, these models are capable of predicting the AE source location within the training domain, and it is extremely difficult to predict outside the domain. Last but not least, ANN trained from training data may not satisfy the essential physical laws associated with an engineering problem.

The governing equations used in the domain of science and engineering are often based on some assumptions and approximations [24]. It is obvious that the solution of such approximate governing equations can loosely depict the actual system behaviour (i.e., low-fidelity results). In the current context, the set of non-linear equations or the cost functions to localise the AE source are formulated based on some governing physical laws. On the contrary, it is conceivable to carry out a few sophisticated experiments by employing state-of-the-art instruments and sensors to get accurate results (i.e., high-fidelity results). However, such experiments are costly and time-consuming, and one can carry out only a small number of experiments (typically in the order of tens). Often such a small number of high-fidelity data are not adequate to get a proper insight of the system, specifically in the case of AE location estimation from the uncertainty/noisy data prevalent in real-life scenarios. Moreover, merging the two data sets of different fidelity is not preferable.

One conceivable alternative to make use of datasets of different fidelity is employing the multi-fidelity framework [25]. Perhaps the multi-level Monte-Carlo (MLMC) methods are among the well-known multi-fidelity schemes where high-fidelity and low-fidelity data are used [26]. Gaussian process and auto-regressive like information fusion schemes are known for dealing with multi-fidelity data [27,28]. Researchers have reported several other techniques in the literature where the Gaussian process is substituted by different machine learning (ML) algorithms [29–31]. The ML techniques also influenced the multi-fidelity approach [32]. Deep neural networks are employed to develop multi-fidelity approaches. The multi-fidelity data are used to train the composite neural network [33].

For addressing the challenges related to the purely data-driven algorithms stated above, we propose a novel probabilistic framework to estimate the AE source location. One of the key components of the proposed framework is to develop a probabilistic approach to account for the uncertainty present in the TOF measurements. The proposed approach utilises a multi-fidelity physics-informed neural network (mfPINN) to achieve the goal. The underlying idea here is initially to train the neural network with the low-fidelity model and then update it with sparse high-fidelity data. The low-fidelity model is the approximate governing physics-based equations, and the high-fidelity data are collected by either performing laboratory experiments or direct field measurements. This is accomplished by the newly evolved physics-informed deep learning algorithm [25,34,35].

The advantages of the proposed framework over other deterministic data-driven approaches are three-fold.

- First, this study proposes a novel cost function extended from the physics-based governing equations of AE source localisation using a probabilistic approach to incorporate the uncertainty/noise in the TOF data, which is prevalent in real-life scenarios.

- Second, as the proposed framework is probabilistic in nature, hence it provides a confidence interval of the estimated AE location, which gives an advantage in decision-making.
- Thirdly, the proposed method is a data-driven hybrid technique that takes advantage of the few measurements from a field experiment in conjunction with the physics-informed mathematical model to train the network as compared to the other data-hungry algorithm. This makes the algorithm ease of use in situations of data sparsity.

The usage of transfer learning is one attribute of the proposed novel framework. The idea behind transfer learning is to retain the gained knowledge while solving one problem and applying it to solve different but related problems. In the current context, transfer learning is leveraged to transfer the knowledge gained while solving a low-fidelity problem to a high-fidelity one. This is done by freezing the parameters, i.e., weights and biases of a few layers of a neural network. With this frame-up, the training phase is remarkably accelerated. The efficacy of the proposed framework is evaluated by estimating the AE source location from the uncertain TOF measurements.

To the best of our knowledge, it is the first implementation of PINN utilising transfer learning based multi-fidelity framework to localize AE source in engineering structures, especially for anisotropic structures. The proposed framework shows excellent capabilities to identify the location of the AE source accurately. Moreover, the framework can consider the uncertainty in the sensor measurement so that the developed model can be used for the localisation of AE sources in real-world scenarios. This is confirmed through the experiment on carbon fibre-reinforced polymer (CFRP) composite panel.

## 2. Mathematical background of acoustic emission source location

Consider a sparse array of $n$ piezoelectric transducers instrumented on anisotropic plate-like structures. The coordinates of the unknown AE source $(x_s, y_s)$ and the fixed sensors are located at the coordinates $(x_i, y_i)$, as shown in Fig. 4. The time of flight (TOF) from the unknown source to the $i^{\text{th}}$ sensor is $t_i$. Conventionally, the unknown coordinates of the AE source can be expressed with the TOF and velocity as:

$$(x_s - x_i)^2 + (y_s - y_i)^2 = (t_i v(\theta_i))^2 \tag{1}$$

where, $i$ is the sequence number of sensors. In the anisotropic plate, $v(\theta_i)$ is the functional relationship between the wave propagation velocity and propagation direction $\theta_i$ defined as [16]:

$$\theta_i = \arctan\left(\frac{y_s - y_i}{x_s - x_i}\right) \tag{2}$$

The TOF from the AE source to the first sensor $(t_1)$ is considered as a reference because the velocity of the stress wave cannot be exactly determined. The following sets of equations can be expressed with the time difference of arrival (TDOA):

$$(x_s - x_i)^2 + (y_s - y_i)^2 = [(t_1 + \Delta t_{1i}) v(\theta_i)]^2 \tag{3}$$

where, $\Delta t_{1i} = t_i - t_1$ is defined as TDOA between the $i$-th and the first sensors. By combining Eqs (1) and (3), the following sets of nonlinear equations can be obtained [16]:

$$\Delta t_{1i} = t_i - t_1 = \frac{\sqrt{(x_s - x_i)^2 + (y_s - y_i)^2}}{v(\theta_i)} - \frac{\sqrt{(x_s - x_1)^2 + (y_s - y_1)^2}}{v(\theta_1)} \tag{4}$$

Where $\theta_i$ and $\theta_1$ are the angle of propagation from the AE source to the $i$th and the first sensors, respectively. This set of nonlinear equations is not sufficient to solve the AE source coordinates for the anisotropic structures as the number of equations available $(n-1)$ is less than the number of unknowns $(n+2)$, i.e., AE source coordinates $(x_s, y_s)$ and $n$-wave velocities.

In the past few decades, several number of TOF-based strategies have been developed to overcome this limitation. Kundu, et al. (2007) used a known velocity profile to reduce the number of unknowns in Eq. (4) [16]. In such a case, a minimum number of three sensors $(n = 3)$ are required to compute the AE source coordinates $(x_s, y_s)$. Several techniques are reported in the literature on the localization of AE sources in anisotropic material without knowing its material properties [36–39]. Ciampa and Meo (2010) also proposed a technique in which the information on the velocity profile is not required, where the clusters of sensors are used [12]. In this approach, the $n$ sensors are placed in $n_c$ clusters in such a way that $n - 1 \geqslant n_c + 2$ in order to solve the system of equations. It is assumed that the wave velocity from the source to each sensor in a cluster is the same. Further, in TOF-based strategies, the accuracy of TOF measurement is crucial as the erroneous measurement will lead to a considerable error in estimating the AE source location.

This work proposes a novel probabilistic approach comprised of a deep neural network to estimate the AE source location from the noisy measurement. The proposed network takes the TDOA of all the sensors along with the TOF uncertainties of each sensor as input and gives the AE source coordinates and corresponding uncertainties. It is considered that the TOF $t_i$ and AE source coordinates $(x_s, y_s)$ are correlated random variables since they are related by Eq. (4); the unknowns $x_s$ and $y_s$ are mutually independent Gaussian random variables. Further, the TOF $t_i$ is considered as a mutually independent Gaussian random variable under the condition that the probable AE event occurs far enough away from the sensor locations (i.e., $\sqrt{\left(\mu_{x_s} - x_i\right)^2 + \left(\mu_{y_s} - y_i\right)^2} \gg 0$). Based on the above assumptions, the

probability density function of TDOA ($\Delta t_{1i} = t_i - t_1$) is also a Gaussian random variable ($\mathcal{N}(\mu, \sigma)$) parameterized by mean ($\mu$) and standard deviation ($\sigma$) [9,11,40].

$$\begin{aligned} t_i &\sim \mathcal{N}\left(\mu_{t_i}, \sigma_{t_i}\right) \\ x_s &\sim \mathcal{N}\left(\mu_{x_s}, \sigma_{x_s}\right) \\ y_s &\sim \mathcal{N}\left(\mu_{y_s}, \sigma_{y_s}\right) \\ \Delta t_{1i} &\sim \mathcal{N}\left(\mu_{\Delta t_{1i}}, \sigma_{\Delta t_{1i}}\right) \end{aligned} \tag{5}$$

The mean of TDOA $\mu_{\Delta t_{1i}}$, and variance of TDOA $\sigma^2_{\Delta t_{1i}}$ are defined as:

$$\mu_{\Delta t_{1i}} = \mu_{t_i} - \mu_{t_1}, \;\; \sigma^2_{\Delta t_{1i}} = \sigma^2_{t_i} + \sigma^2_{t_1} \tag{6}$$

One can write the Eq. (1) as follows:

$$t_i^2 = \frac{(x_s - x_i)^2 + (y_s - y_i)^2}{\{v(\theta_i)\}^2} \tag{7}$$

In Eq. (7), $t_i \sim \mathcal{N}\left(\mu_{t_i}, \sigma_{t_i}\right)$, $x_s \sim \mathcal{N}\left(\mu_{x_s}, \sigma_{x_s}\right)$ and $y_s \sim \mathcal{N}\left(\mu_{y_s}, \sigma_{y_s}\right)$ are the random variables. However, it is to be noted that, $t_i^2$ in Eq. (7) is actually a (scaled) generalized chi-squared random variable. In particular, if $\sigma_{x_s} = \sigma_{y_s}$ then $t_i^2$ has a (scaled) non-central chi-squared distribution with two degrees of freedom. Now applying the expectation operator $\mathrm{E}[\cdot]$ on both sides of the Eq. (7) we get:

$$\begin{aligned} \mathrm{E}\left[t_i^2\right] &= \mathrm{E}\left[\frac{(x_s - x_i)^2 + (y_s - y_i)^2}{\{v(\theta_i)\}^2}\right] \\ &= \frac{\mathrm{E}\left[(x_s - x_i)^2\right] + \mathrm{E}\left[(y_s - y_i)^2\right]}{\{v(\theta_i)\}^2} \\ &= \frac{\mathrm{E}\left[x_s^2 - 2x_s x_i + x_i^2\right] + \mathrm{E}\left[y_s^2 - 2y_s y_i + y_i^2\right]}{\{v(\theta_i)\}^2} \\ &= \frac{\mathrm{E}\left[x_s^2\right] - 2\mathrm{E}[x_s]x_i + x_i^2 + \mathrm{E}\left[y_s^2\right] - 2\mathrm{E}[y_s]y_i + y_i^2}{\{v(\theta_i)\}^2} \end{aligned} \tag{8}$$

The statistical moments of $t_i$ are known. Considering $var[t_i] = \sigma^2_{t_i}$ then one can write as:

$$\mathrm{E}\left[t_i^2\right] = \mu_{t_i}^2 + \sigma_{t_i}^2 \tag{9}$$

Similarly, $\mathrm{E}\left[x_s^2\right] = \mu_{x_s}^2 + \sigma_{x_s}^2$ and $\mathrm{E}\left[y_s^2\right] = \mu_{y_s}^2 + \sigma_{y_s}^2$. Now, substituting the value of $\mathrm{E}\left[t_i^2\right]$, $\mathrm{E}\left[x_s^2\right]$ and $\mathrm{E}\left[y_s^2\right]$ in Eq. (8) one gets:

$$\mu_{t_i}^2 + \sigma_{t_i}^2 = \frac{\mu_{x_s}^2 + \sigma_{x_s}^2 - 2\mu_{x_s}x_i + x_i^2 + \mu_{y_s}^2 + \sigma_{y_s}^2 - 2\mu_{y_s}y_i + y_i^2}{\{v(\theta_i)\}^2} \tag{10}$$

$$\Rightarrow \mu_{t_i} = \sqrt{\frac{\mu_{x_s}^2 + \sigma_{x_s}^2 - 2\mu_{x_s}x_i + x_i^2 + \mu_{y_s}^2 + \sigma_{y_s}^2 - 2\mu_{y_s}y_i + y_i^2}{\{v(\theta_i)\}^2} - \sigma_{t_i}^2} \tag{11}$$

Similarly,

$$\mu_{t_1} = \sqrt{\frac{\mu_{x_s}^2 + \sigma_{x_s}^2 - 2\mu_{x_s}x_1 + x_1^2 + \mu_{y_s}^2 + \sigma_{y_s}^2 - 2\mu_{y_s}y_1 + y_1^2}{\{v(\theta_1)\}^2} - \sigma_{t_1}^2} \tag{12}$$

Substituting the value of $\mu_{t_i}$ and $\mu_{t_1}$ in the expression of $\mu_{\Delta t_{1i}}$ given in Eq. (6) one can get:

$$\mu_{\Delta t_{1i}} = \sqrt{\frac{\mu_{x_s}^2 + \sigma_{x_s}^2 - 2\mu_{x_s}x_i + x_i^2 + \mu_{y_s}^2 + \sigma_{y_s}^2 - 2\mu_{y_s}y_i + y_i^2}{\{v(\theta_i)\}^2} - \sigma_{t_i}^2} - \sqrt{\frac{\mu_{x_s}^2 + \sigma_{x_s}^2 - 2\mu_{x_s}x_1 + x_1^2 + \mu_{y_s}^2 + \sigma_{y_s}^2 - 2\mu_{y_s}y_1 + y_1^2}{\{v(\theta_1)\}^2} - \sigma_{t_1}^2} \tag{13}$$

It should be noted that the direction dependent velocity $v(\theta)$ is a deterministic quantity as the velocity profile is known *a priori* to make the Eq. (13) solvable. The low-fidelity physics-based model given in Eq. (13) is employed to train the network initially.

## 3. Time of flight measurements

The accuracy of the AE source estimation depends upon the precision of the TOF measurements. It is contentious to measure the exact arrival time of the signal from the time-history plot. Because the time domain signals are perturbed by the low-level and high-frequency noise that creates an uncertainty in the signal, first time of arrival and this noise obscures the actual arrival time. Consequently, the actual arrival time may deviate from the measured time of arrival. However, the cost function used in this study given in

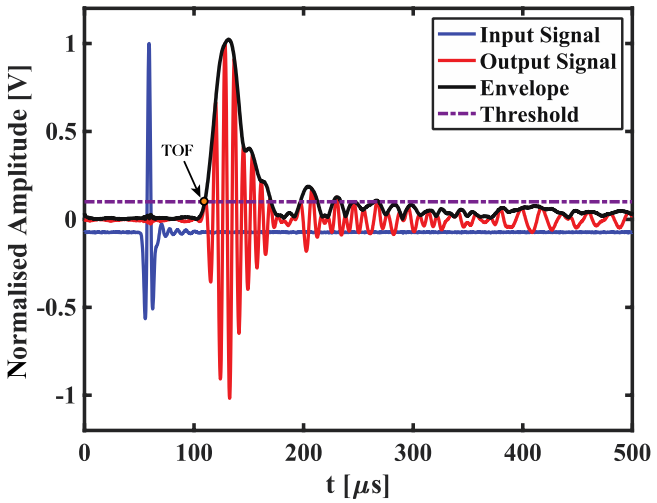



**Fig. 1.** Illustration of the arrival time measurement by threshold crossing of the time history signal envelope.

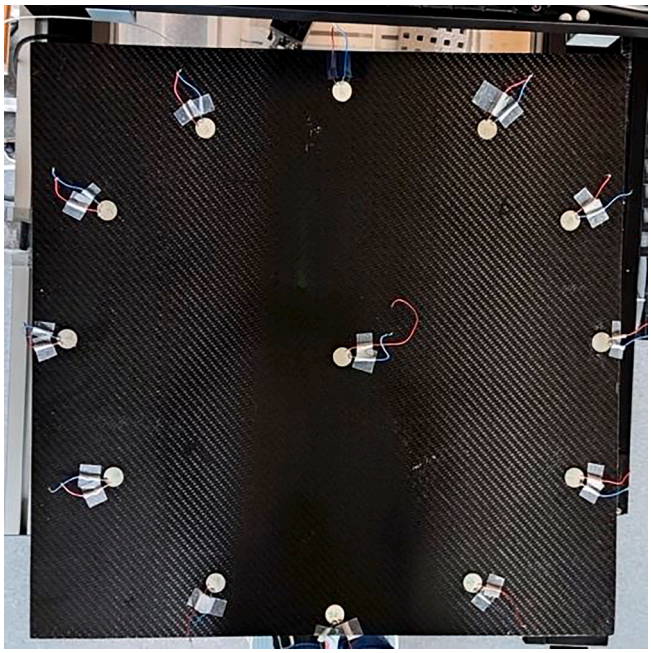


**Fig. 2.** Experimental setup of wave velocity profile measurement.

Eq. (13) is based on TDOA; hence, these minor errors get mutually nullified, and the final estimation should not be affected considerably. Several techniques comprising threshold crossing by fitting the envelope, wavelet transform, cross-correlation, and curve fitting, to mention a few are available in the literature to measure the time of arrival from the time domain acoustic signal [41–43]. This study employs a principled way to measure the arrival time by threshold crossing of the signal envelope. The envelope can be computed by using the definition of an analytic signal $x_A(t)$ of an actual signal $x(t)$:

$$x_A(t) = x(t) + ix_H(t) \tag{14}$$

where, $i$ is the imaginary unit and $x_H(t)$ is the Hilbert transform of $x(t)$ defined as:

$$x_H(t) = \frac{1}{\pi}\int_{-\infty}^{\infty} x(\tau)\frac{1}{t-\tau}d\tau \tag{15}$$

Now, the envelope $A(t)$ can be computed as:

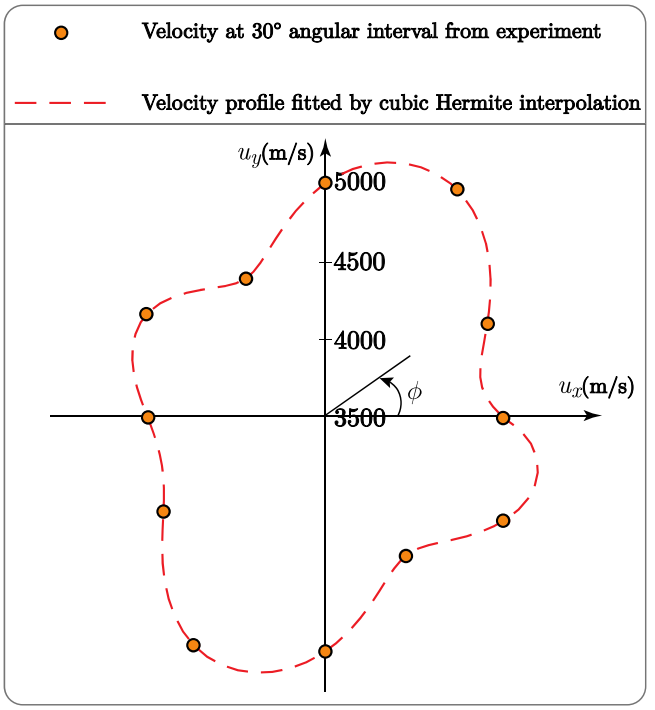



**Fig. 3.** The variation of wave velocity in different directions for the composite plate.

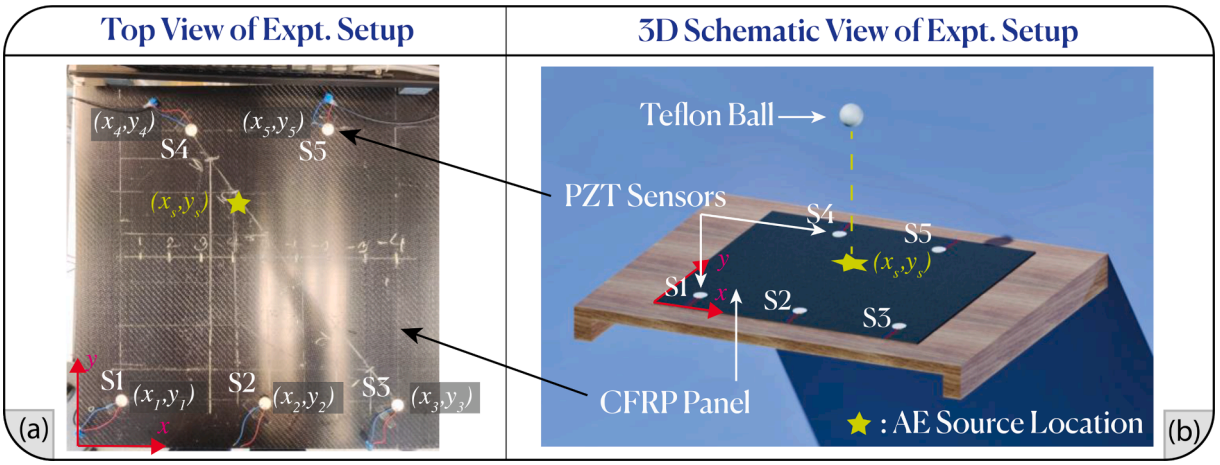



**Fig. 4.** (a) Top view of the experimental setup, and (b) a 3D schematic view of the experimental setup.

$$A(t) = \sqrt{x(t)^2 + x_H(t)^2} \tag{16}$$

The time of flight is measured when the envelope of a signal crosses the defined threshold $(A_t)$. In this study, a threshold $A_t$ was defined as 0.1 V [9,41]. Fig. 1 illustrates the arrival time measurement from the threshold crossing of the envelope of a signal.

## 4. Experimental setup

Experimental tests were conducted using a 60 cm × 60 cm CFRP composite panel to collect high-fidelity data and to evaluate the efficacy of the proposed framework. The anisotropic panel had a nominal thickness of 2 mm. In order to simulate the impact on a plate, a Teflon ball of diameter 13.4 mm was dropped on the composite plate. The ball weighs approximately 10 g and dropped from a height of 0.3 m. The velocities in different directions are calculated by placing twelve PZT sensors in a circle with a radius of 27.5 cm at 30° angular spacing varying from 0° to 360° as shown in Fig. 2.

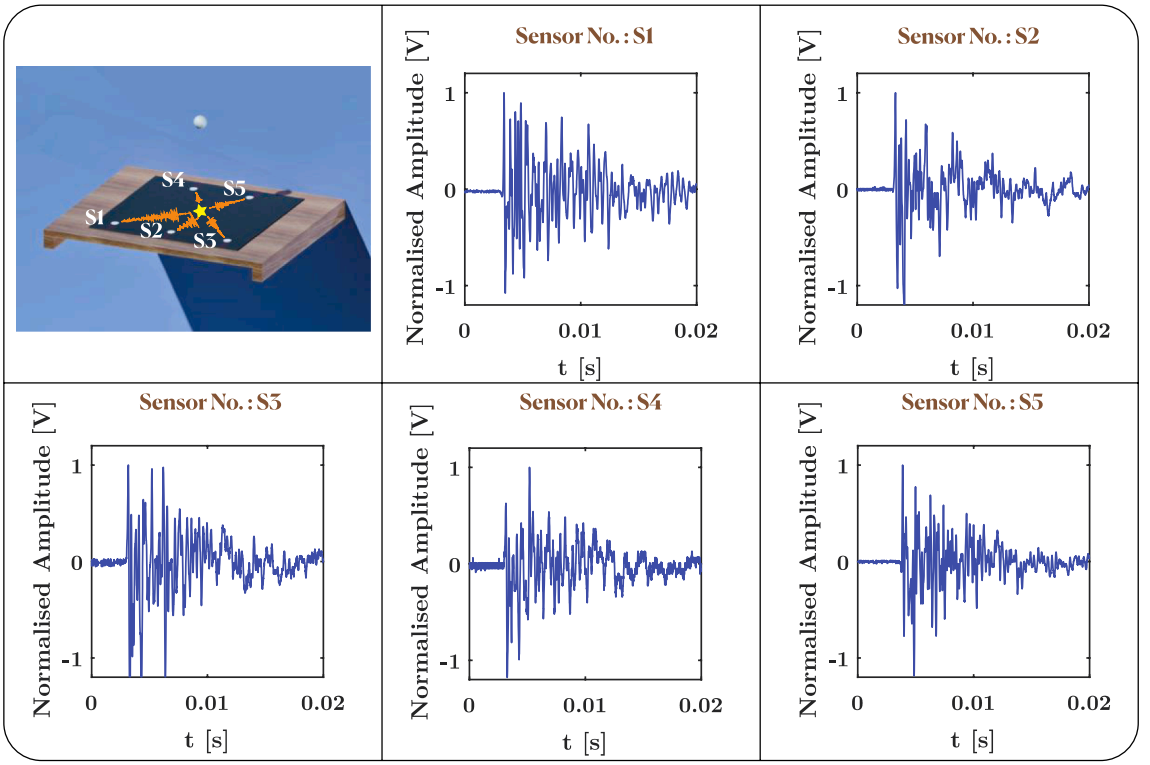



**Fig. 5.** Plots of time domain signals were received at different sensors due to the drop of the Teflon ball.

**Table 1**
Location of sensors placed to detect the AE source in the experiment.

| Sensor No. | Sensor Location | |
|---|---|---|
| | $x_s$ (cm) | $y_s$ (cm) |
| 1 | 7.5 | 7.5 |
| 2 | 30.5 | 7.5 |
| 3 | 53.5 | 7.5 |
| 4 | 19 | 53.5 |
| 5 | 42 | 53.5 |

A guided ultrasonic wave was excited at the centre of the circle through central PZT. The signal is received simultaneously by all the sensors placed at the circumference of the circle. In Fig. 1, the blue and red signals are the input signal excited at the centre and the signal received by the sensor at the circumference, respectively. The time difference between the input signal and TOF of the received signal is the time taken to propagate the wave from the centre to the circumference. By measuring the time of arrival and dividing the radius of the circle by the time of arrival, the wave velocities in different directions are computed. After experimentally measuring the wave velocity in discrete angular directions, the wave velocity at any angle $\phi$ can be obtained by performing the cubic Hermite interpolation. The resulting velocity profile is shown in Fig. 3.

After obtaining the velocity profile, the sparse array of five sensors were installed on the plate, as shown in Fig. 4. A point on the plate was selected randomly, and the Teflon ball was dropped at that point to generate high-fidelity data. The signals received by the five sensors were recorded to estimate the AE source. Fig. 5 illustrates the plots of time domain signals received at different sensors due to the Teflon ball being dropped at a location of $x = 30$ cm and $y = 35$ cm. Table 1 summarises the location of sensors placed on the plate to detect the AE source.

## 5. Mathematical background of multi-fidelity physics informed neural network

The data-driven ANN and PINN are the cornerstone of the proposed mfPINN. In the following sections, a brief background on the ANN and PINN is presented.

### 5.1. Data-driven artificial neural network

An artificial neural network is a smart data processing system that imitates the biological neural network of the animal brain [44]. It is a robust technique that is extensively used in the domain of machine learning (ML) [45]. An ANN is comprised of input, output and hidden layers in between. These layers are interconnected by the number of neurons. The connectivity between the neurons depicts the functionality of the ANN [46]. ANNs have effectively been utilised to understand complex engineering systems and to extract important and meaningful features from the data.

The nodes of a layer are parameterised by weights ($\mathbf{w}$) and biases ($\mathbf{b}$). Each input parameter ($\mathbf{x}$) is multiplied by its weights prior to reaching the hidden layer node. The weighted sum ($\mathbf{z}$) of the weighted parameters collected at the nodes is computed. The weighted sum can be defined as

$$\mathbf{z} = \mathbf{w}\mathbf{x} + \mathbf{b} \tag{17}$$

Subsequently, the weighted sum is transmitted to the next layer using the activation function and finally to the output layer as the target data. The activation function $f(\cdot)$ is defined as [47]:

$$\mathbf{u} = f(\mathbf{z}) \tag{18}$$

In a feedforward neural network, the input data is propagated through the input, hidden and output layers. However, in the backpropagation algorithm, the output loss is propagated back in the opposite direction of the output, hidden and input layers. During the training process in a data-driven setting, the network parameters, i.e., weights and biases, are updated by minimising the loss function. The loss function ($\mathscr{L}_{\mathbf{u}}$) for this is given by

$$\mathscr{L}_{\mathbf{u}} = \sum_{k=1}^{N_H} \|\mathbf{u} - \widetilde{\mathbf{u}}\| \tag{19}$$

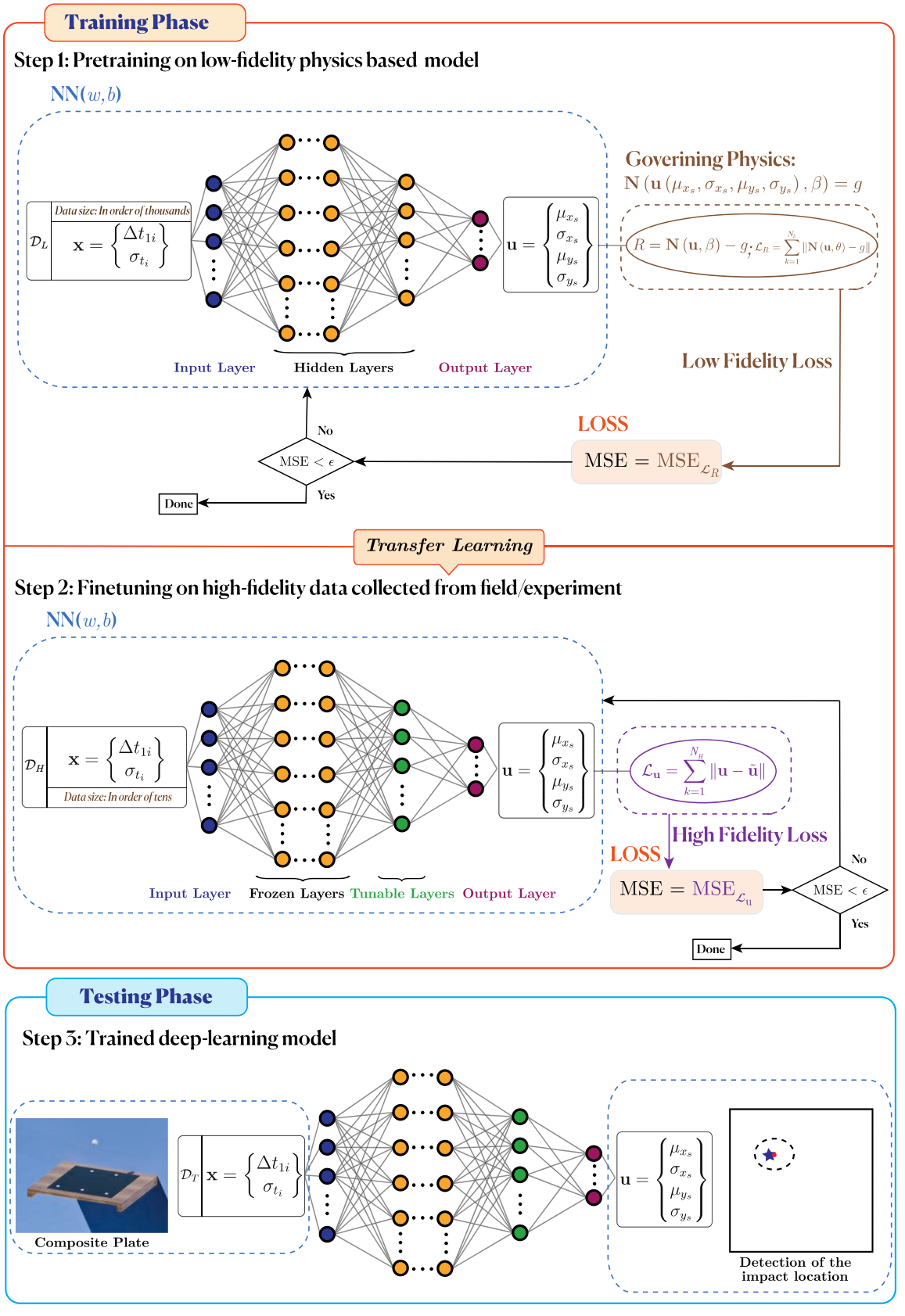
where, $N_H$ is the number of high-fidelity data points available, $\mathbf{u}$ is the observed response and $\widetilde{\mathbf{u}}$ is the neural network predicted response. Despite the fact that ANN architectures are employed to learn physics, there exist crucial issues related to interpretability, generalizability, overfitting and demands for significant data. These factors limit their utility and computational tractability for data-driven models. It is to be noted that for the current work, the high-fidelity data collected from the experiment is very sparse. Thus, the direct application of data-driven ANN is inconceivable to get adequate results.

### 5.2. Physics-informed neural network

One of the key components of the proposed framework is building physics-informed learning machines. In this section, we first delineate how to embed physics in an ML model and how various physics can steer in developing novel NN architectures. Currently, there are three approaches that can be pursued individually or in tandem to expedite training and enhance the generalization of ML models by embedding physics in them [48]. (a) Training ML systems on data that embody the underlying physics allow them to learn functions, vector fields or operators that contemplate the physical structure of the data by introducing observational biases. (b) Another pathway of embedding physics is by introducing inductive biases to incorporate the prior assumptions in tailored interventions to an ML architecture in such a way that the predictions of the ML models are ensured to implicitly satisfy the governing physical laws, often expressed in the form of certain mathematical constraints. The implementation of this approach is complex and difficult to scale. (c) Last but not least approach of embedding physics is by proper selection of loss functions, constraints and inference algorithms. The learning biases can be introduced that can regulate the training phase of the ML system to explicitly favour convergence towards solutions that comply with the governing physics. This enables a very adaptable platform of embedding physics that can be expressed in the form of integral, differential or even fractional equations. These various methods of embedding physics in a learning algorithm towards physically coherent solutions are not mutually exclusive and can be adequately merged together to yield a very broad class of hybrid approaches for developing physics-informed learning machines. From these available pathways of embedding physics, the current work followed the learning biases mode to impose the underlying physics in a soft manner by properly defining the loss function of conventional NN. This variation of NN is also known as a physics-informed neural network (PINN).

Recently, PINNs have become an excellent solution technique for solving various physics-based models pioneered by Raissi, et al. (2019) [34]. Several works are reported in the literature where the PINNs are leveraged to solve various engineering and scientific problems when the underlying physics is in the form of differential equations [25,48,49]. Despite this fact, one can extend the capability of PINN to solve the problem when the underlying physics is in the form of fractional equations. However, in such a scenario, the application of PINN is sparse in the literature. Recently, Pun, et al. (2019) proposed to employ PINN to predict interatomic potential energy surface in a large atomistic model [50]. They combine the physics-based model of bond order potential, which is in the form of a fractional equation with a neural network. Further, they compared the results of PINN with a conventional neural network and concluded that the accuracy of the PINN is better and performs well in the extrapolation regime. Several studies have made similar claims that PINN performs better than the conventional NN trained in a data-driven setting [25,48–50]. However, one can generate data based on the underlying physics and train a conventional neural network on such data by using the observational biases mode of embedding physics on a learning model. This type of model is not generalizable and does not offer any flexibility.

In the current impact detection problem, the governing physics contains parameters like the properties of the material in the form




*(caption on next page)*

**Fig. 6.** Schematic representation of proposed mfPINN framework for estimating AE source location and corresponding uncertainty.

of the velocity of the wave ($v(\theta)$), the location of sensors $(x_i, y_i)$ and the uncertainty of the sensors $(\sigma_t)$. For certain parameters one can generate data that embody the underlying physics and train a conventional neural network on such data to get the intended output from the neural network. However, any changes in the parameter, for instance, the material is changed, and the positions of sensors are changed; in such a scenario, one has to generate the data and train the network again. This data generation task could be time-consuming and computationally expensive in many engineering applications. We proposed to use PINN over a conventional neural network to avoid the data generation step, as PINN can be directly trained by using the underlying physics as expressed in Eq. (13) without generating data from it, as the governing physics is directly embedded in the PINN. The leading motivation for employing PINN in the proposed framework is that the embedding underlying physics can yield more interpretable deep learning architecture that remain robust in the presence of imperfect data and can provide accurate and physically consistent prediction even for extrapolatory / generalization task.

Most real engineering systems suffer from limited high-fidelity or experimental data. In such cases, the conventional NN does not perform well because of its data intensiveness nature and the huge training data requirement. The advantage of PINN over the conventional NN is that the PINN performs very well in the sparse data scenario as it takes advantage of the underlying physics-based mathematical model.

To be mathematically precise, we again generalized the representation of the Eq. (13) as

$$\mathbf{N}\left(\mathbf{u}\left(\mu_{x_s}, \sigma_{x_s}, \mu_{y_s}, \sigma_{y_s}\right), \beta\right) = g \tag{20}$$

where,

$$\mathbf{N}\left(\mathbf{u}\left(\mu_{x_s}, \sigma_{x_s}, \mu_{y_s}, \sigma_{y_s}\right), \beta\right) = \sqrt{\frac{\mu_{x_s}^2 + \sigma_{x_s}^2 - 2\mu_{x_s} x_i + x_i^2 + \mu_{y_s}^2 + \sigma_{y_s}^2 - 2\mu_{y_s} y_i + y_i^2}{\{v(\theta_i)\}^2} - \sigma_{t_i}^2} - \sqrt{\frac{\mu_{x_s}^2 + \sigma_{x_s}^2 - 2\mu_{x_s} x_1 + x_1^2 + \mu_{y_s}^2 + \sigma_{y_s}^2 - 2\mu_{y_s} y_1 + y_1^2}{\{v(\theta_1)\}^2} - \sigma_{t_1}^2};$$
$$g = \mu_{\Delta t_{1i}} \tag{21}$$

$\mathbf{N}(\cdot)$ prescribe the nonlinear function as defined in Eq. (21) and $\beta$ represents the other parameters of Eq. (21). The goal is to find a neural network approximation $\widetilde{\mathbf{u}}$ to the observed data $\mathbf{u}$. The approximation should satisfy the governing physics defined in Eq. (13) and in turn Eq. (20). The neural network is then trained to minimize the physics-informed loss function given by

$$\mathscr{L}_R = \sum_{k=1}^{N_L} \|\mathbf{N}(\mathbf{u}, \beta) - \mathrm{g}\,\| \tag{22}$$

where, $N_L$ is the number of points for the input and $R$ is the residual of the governing equation corresponding to the collocation points and is defined as

$$R = \mathbf{N}(\mathbf{u}, \beta) - g \tag{23}$$

The advantage of PINN is the training is fast as compared to the data-driven ANN because the PINN needs no simulation data. The governing equation of the system is satisfied during the training process; thus, the other physical properties of the system are also satisfied. One must note that the governing equation under consideration is available to apply the PINN successfully. However, numerous cases exist in science and engineering where the underlying equation is unknown [24]. Albeit, the equation is known, it is usually derived based on certain assumptions and approximations. The governing equation merely depicts the actual case in an approximate manner. It is expected that the results obtained from PINN would necessarily be erroneous.

### 5.3. Proposed approach

The overarching goal of this work is to leverage the capabilities of PINN and ANN to estimate the AE source location under the condition of sparse and noisy data. The proposed approach is driven by the requirement for an efficient parsimonious model that evades overfitting and exploits the underlying physics of the problem. Unlike previous deep learning approaches, the proposed network architecture is designed particularly to handle a ubiquitous class of sparse data perturbed by noise and uncertainty.

The proposed network architecture is shown in Fig. 6. The objective of this network is to estimate the location $\left(\mu_{x_s}, \mu_{y_s}\right)$ of the AE source and the corresponding uncertainty $(\sigma_{x_s}, \sigma_{y_s})$ estimated from the TDOA information $(\mu_{\Delta t_{1i}})$ perturbed with noise/uncertainty $(\sigma_{t_i})$. Neither data-driven ANN nor the PINN is competent in estimating the AE source from the sparse, noisy data collected from field/real experiments. The ANN fails as the data available is sparse. Moreover, neural networks have considerable potential for overfitting data. Simultaneously, PINN fails because the governing physics in Eq. (13) merely portrays the real scenario in an approximate manner.

To overcome the problem, a novel framework based on mfPINN is presented in this study. The presented approach employs both ANN and PINN. Towards this goal, a single deep neural network is first trained for the low-fidelity model and then updated based on

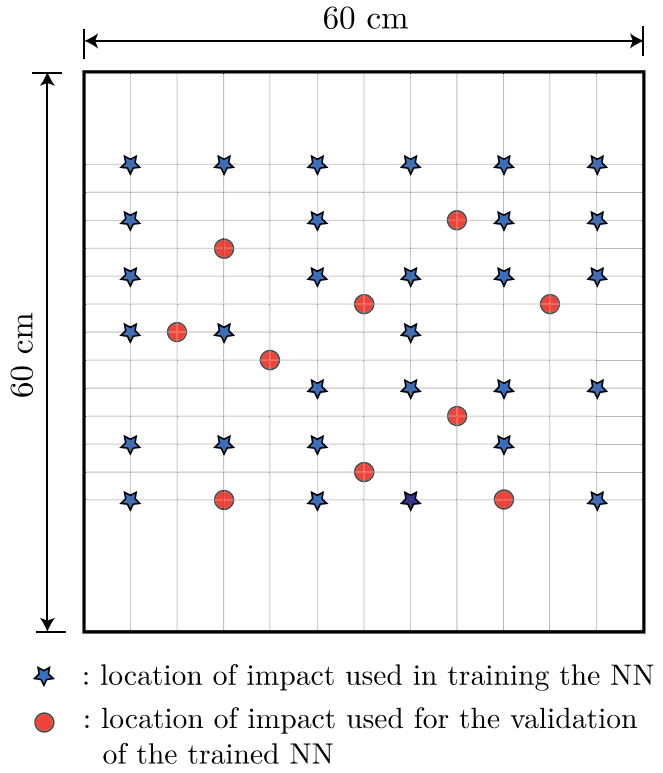



**Fig. 7.** Illustration of the impact location to generate high-fidelity data by performing the experiment.

the high-fidelity data. The idea of transfer learning is employed to tune the network on high-fidelity data. The usage of transfer learning in the multi-fidelity framework is found in the literature [32]. The concept of data-driven ANN is used to update the low-fidelity PINN on the high-fidelity data. While updating the network, two crucial factors are considered. First, the training starts on the high-fidelity data by retaining the initial value of the network parameters obtained from the training on low-fidelity PINN. Second, using the concept of transfer learning [49] the parameters corresponding to the last one or two layers are updated instead of all the network layers. The transfer learning accelerates the training process of the network as the number of parameters to be updated is decreased. Freezing the initial layers ensures the knowledge gained from the low-fidelity model is retained in the deep learning model. Further, transfer learning avoids the overfitting of the network on high-fidelity data.

Closing this section, it is worth mentioning that mfPINN involves a significant number of hyperparameters, that is, the number of layers, the types of activation functions, and the neurons per layer, to mention a few, need to be tuned carefully in order to get optimal results. The universal approximation theorem promises that a neural network comprised of at least one hidden layer and a large enough number of neurons, and the differentiable activation function is competent in depicting any arbitrary function or dataset [51–53]. Hanin (2019) presented expressions for the optimal neural network parameters required to depict a function within a desired accuracy when ReLU activation functions are used [54]. There are several methods, such as grid search or random search, Bayesian hyperparameter optimisation procedure [55] etc., are often used to tune the neural network hyperparameters. Despite these theoretical guarantees, the training algorithm might not be able to find optimal values of the network parameters to achieve the desired accuracy. In practice, with few numbers of hidden layers and a significant number of neurons per layer might be needed. Further, often more than one layer with a small number of neurons per layer is used. For the activation function $f(\cdot)$ in Eq. (18), there are various alternatives available. One common option is the rectified linear unit (ReLU). For an input $z$, the output from ReLU is given in Eq. (24).

$$f_{\text{ReLU}}(z) = \max(0, z) \tag{24}$$

It is worth noting that the derivative of the output from ReLU vanishes for $z < 0$. This conceives a difficulty in training using gradient descent for negative inputs and is called a *dying ReLU* problem. In order to avoid this problem, the applicability of another activation function is explored, namely, hyperbolic tangent or tanh. For this activation function, the output is given in Eq. (25).

$$f_{\tanh}(z) = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{25}$$

For the last layer, we employed the linear activation function, where the output is given in Eq. (26).

$$f_{\text{Linear}}(z) = z \tag{26}$$

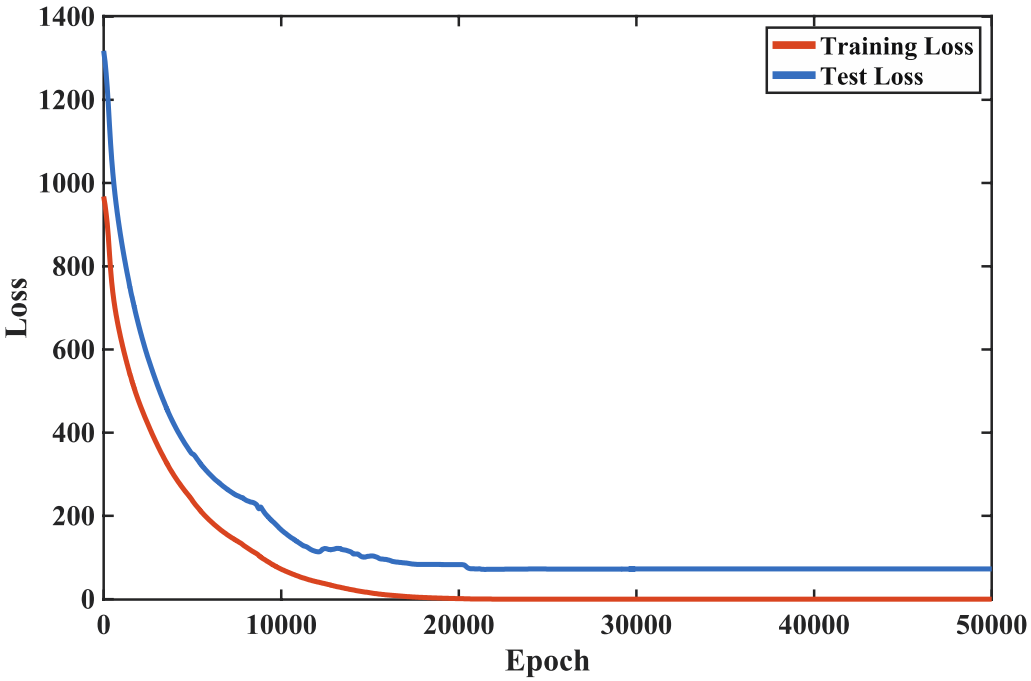



**Fig. 8.** The plot of the performance summary (red line) of the optimization algorithm during the training and the performance summary of ANN (blue line) in prediction during the process. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

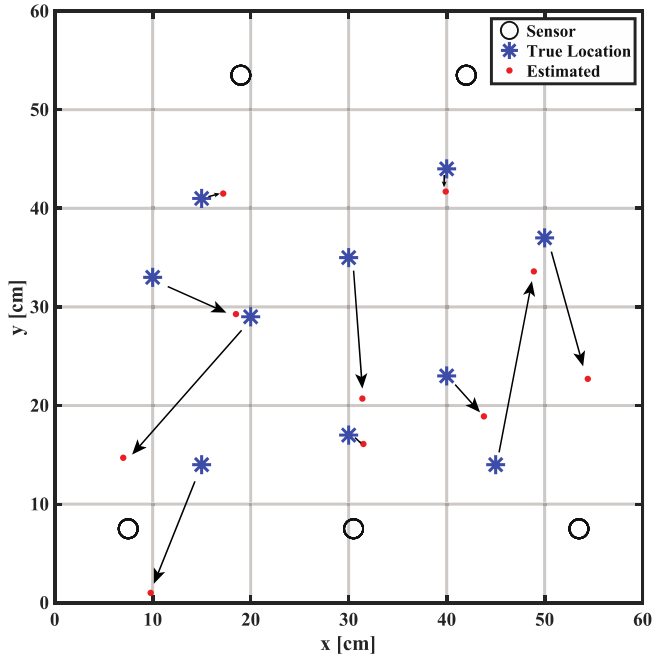



**Fig. 9.** Impact location results showing true and estimated locations using data-driven ANN.

One should note that there exist different alternatives for activation functions, e.g., logistic sigmoid, leaky ReLU, and exponential linear unit (ELU), to mention a few. However, we found the combination of tanh and linear activation provides an excellent result for the AE detection problem. Further, in this work, we use 50 neurons per layer based on a preliminary investigation, where the performance is monitored with different number of neurons per layer. A new hidden layer is included to the network if the accuracy is increased with the included new layer.

**Table 2**
Source estimation results using data-driven ANN.

| Point | Actual Location | | Estimated Location | | $\varepsilon_r$ (cm) |
|---|---|---|---|---|---|
| | $x_s$ (cm) | $y_s$(cm) | $\mu_{x_s}$ (cm) | $\mu_{y_s}$ (cm) | |
| 1 | 15 | 14 | 9.8 | 1.0 | 13.9993 |
| 2 | 30 | 17 | 31.5 | 16.1 | 1.7251 |
| 3 | 40 | 23 | 43.8 | 18.9 | 5.5748 |
| 4 | 10 | 33 | 18.5 | 29.1 | 9.3269 |
| 5 | 20 | 29 | 7.0 | 14.7 | 19.3140 |
| 6 | 40 | 44 | 39.9 | 41.7 | 2.3125 |
| 7 | 15 | 41 | 17.2 | 41.5 | 2.2244 |
| 8 | 50 | 37 | 51.4 | 22.7 | 14.3986 |
| 9 | 45 | 14 | 48.9 | 33.6 | **19.9423** |
| 10 | 30 | 35 | 31.4 | 20.7 | 14.3986 |
| | | | | | Average (cm) |
| | | | | | 10.3217 |

## 6. Results and discussion

In this section, three sets of results are presented to illustrate the performance of the proposed framework. The results are presented in the first two sets using data-driven ANN and PINN. The intention of these two sets of results is to illustrate the necessity of a multi-fidelity framework using transfer learning.

A Teflon ball is dropped on 40 different locations of the plate, given the wave velocity profile (Fig. 3) is known to generate high-fidelity data $\mathscr{D}_H$, as shown in Fig. 7. These data are randomly divided into two categories. The first category consists of the TDOA and uncertainty of the sensor corresponding to the thirty different locations, as illustrated by the blue star in Fig. 7 are used to train the neural network model. The red circles in Fig. 7 depict rest of the data used to evaluate the efficacy of the proposed model.

### 6.1. Estimation of impact location using high-fidelity data-driven ANN

A data-driven ANN is modelled to solve the problem under consideration. The ANN estimates the location of impact deterministically as in the observed high-fidelity data; the impact location information is deterministic. The network has five inputs and five hidden layers with 50 neurons each. The five inputs are the TDOA information measured from 5 sensors installed on the plate. The network has two outputs corresponding to the *x*- and *y*- coordinates of the estimated locations. Hyperbolic tangent (tanh) activation is used for all except the last layer. Linear activation is considered for the last layer. The Adam optimizer is run for 50,000 iterations for training the ANN with available 30 training data. The training loss is converged satisfactorily, as presented in Fig. 8. However, the network over-fitted the data, as one can observe that the test loss is converged but significant.

The results obtained from data-driven ANN are illustrated in Fig. 9 in terms of true locations and estimated locations. To evaluate the accuracy of the results, an error function is defined as

$$\varepsilon_r = \sqrt{\left(\mu_{x_s} - x_s\right)^2 + \left(\mu_{y_s} - y_s\right)^2} \tag{27}$$

where, $(x_s, y_s)$ are the true impact coordinates and $\left(\mu_{x_s}, \mu_{y_s}\right)$ are the neural network estimated coordinates. Table 2 summarises the impact of localisation results using data-driven ANN. It is observed from Table 2 that the performance of the data-driven ANN in the prediction of AE source location is poor, with an average error of 10.3217 cm and a maximum error of 19.9423 cm.

### 6.2. Estimation of impact location using low-fidelity PINN

To solve the problem under consideration, the unknown location and corresponding uncertainty $\left(\mathbf{u} = \left\{\mu_{x_s}, \sigma_{x_s}, \mu_{y_s}, \sigma_{y_s}\right\}^T\right)$ are first represented by using a neural network. The network comprises ten inputs, five hidden layers with 50 neurons per layer and four outputs. The ten inputs are TDOA information $(\mu_{\Delta t_{1i}})$ and corresponding uncertainty in the TOF measurement $(\sigma_{t_i})$ of the installed five sensors. Hyperbolic tangent (tanh) activation is used in all the layers except the last layers. Linear activation is considered for the last layer. The residual for training the network is given in Eq. (28).

$$\begin{aligned} R_k \quad &= \sqrt{\frac{\left(\mu_{x_s}^2\right)_k + \left(\sigma_{x_s}^2\right)_k - 2\left(\mu_{x_s}\right)_k x_i + x_i^2 + \left(\mu_{y_s}^2\right)_k + \left(\sigma_{y_s}^2\right)_k - 2\left(\mu_{y_s}\right)_k y_i + y_i^2}{\{v(\theta_i)\}^2} - \left(\sigma_{t_i}^2\right)_k} \\ &- \sqrt{\frac{\left(\mu_{x_s}^2\right)_k + \left(\sigma_{x_s}^2\right)_k - 2\left(\mu_{x_s}\right)_k x_1 + x_1^2 + \left(\mu_{y_s}^2\right)_k + \left(\sigma_{y_s}^2\right)_k - 2\left(\mu_{y_s}\right)_k y_1 + y_1^2}{\{v(\theta_1)\}^2} - \left(\sigma_{t_1}^2\right)_k - \left(\mu_{\Delta t_{1i}}\right)_k} \end{aligned} \tag{28}$$

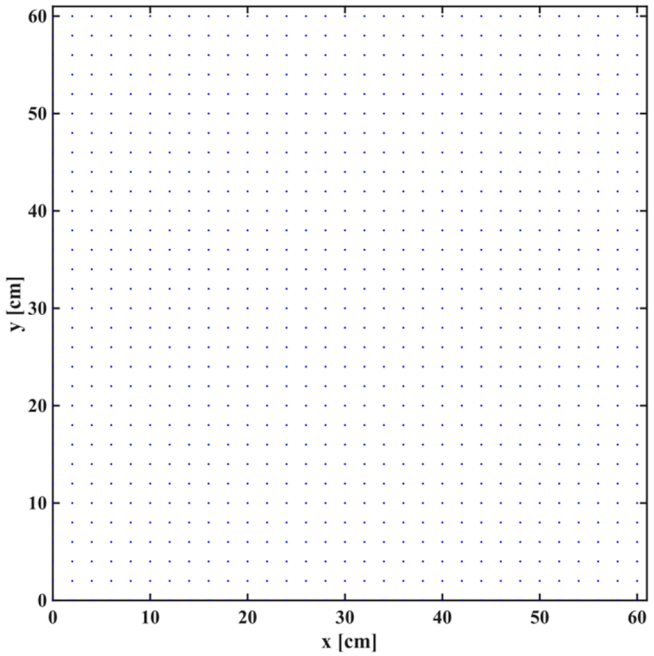



**Fig. 10.** The collocation points are used to train the low-fidelity model of the neural network.

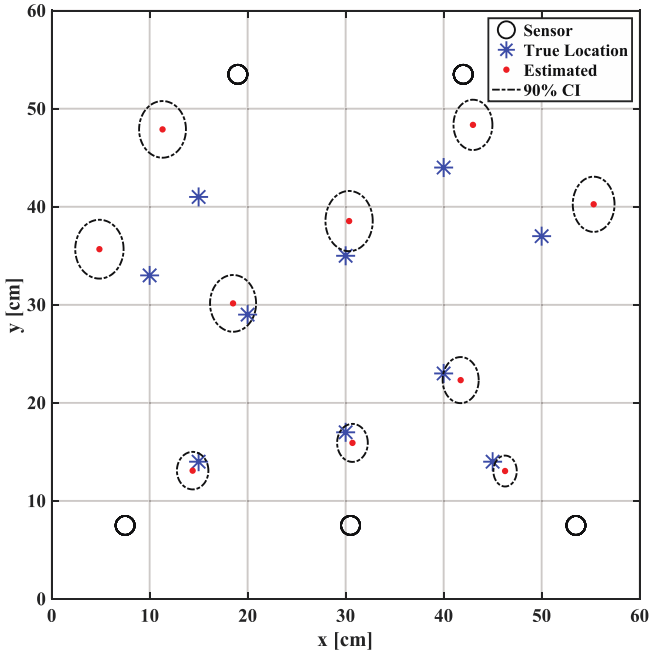



**Fig. 11.** Impact location results showing true and estimated locations using low-fidelity PINN.

Where, $R_k$ is the residual and $\widetilde{\mathbf{u}} = \left\{\mu_{x_s}, \sigma_{x_s}, \mu_{y_s}, \sigma_{y_s}\right\}^T$ is obtained from the neural network output. '$i$' and '$k$' in the suffix indicating the $i$-th sensor and $k$-th collocation point. For training the low-fidelity PINN, the collocation points, as shown in Fig. 10 are generated. The number of collocation points is 961 with an interval of 2 cm in $x$- and $y$- directions.

Naively initiating the input in the neural network will not ensure that the model is trained within the given plate boundary. Alternatively, one can create the TDOA ($\Delta t_{1i}$) information by using Eq. (4) corresponding to these collocation points in a deterministic sense. However, this step can be avoided by approximately estimating the largest possible value of TDOA ($\Delta t_{1i}$), then initiate the

**Table 3**
Source estimation results using low-fidelity PINN.

| Point | Actual Location | | Estimated Location | | $\varepsilon_r$ (cm) |
|---|---|---|---|---|---|
| | $x_s$ (cm) | $y_s$ (cm) | $\mu_{x_s}$ (cm) | $\mu_{y_s}$ (cm) | |
| 1 | 15 | 14 | 14.2 | 12.8 | 1.4408 |
| 2 | 30 | 17 | 31.0 | 16.3 | 1.2253 |
| 3 | 40 | 23 | 41.6 | 22.2 | 1.7591 |
| 4 | 10 | 33 | 5.9 | 35.0 | 4.6197 |
| 5 | 20 | 29 | 18.4 | 30.1 | 1.9649 |
| 6 | 40 | 44 | 43.1 | 48.7 | 5.6773 |
| 7 | 15 | 41 | 12.1 | 48.3 | **7.8926** |
| 8 | 50 | 37 | 55.3 | 40.3 | 6.2253 |
| 9 | 45 | 14 | 46.0 | 13.1 | 1.3266 |
| 10 | 30 | 35 | 30.3 | 38.4 | 3.3970 |
| | | | | | Average (cm) |
| | | | | | 3.5529 |

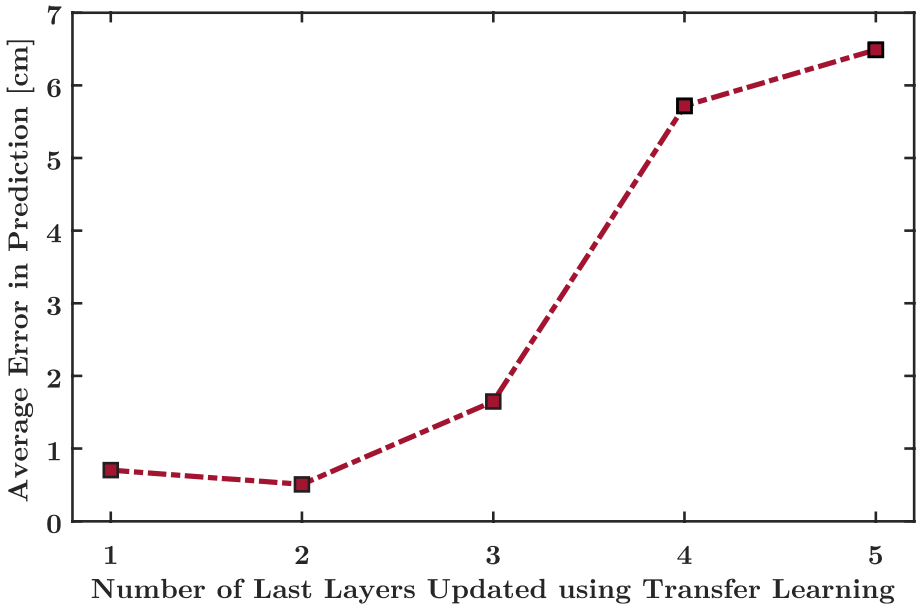



**Fig. 12.** The plot of the performance summary of the proposed mfPINN framework in predicting the AE source location while the last various layers are updated using transfer learning.

training by providing the TDOA ($\Delta t_{1i}$) as an input within the largest possible value. The unconservative estimation of this value can be done by dividing the distance of a furthest point in the plate from a sensor by the minimum value of velocity profile. One must note that the proposed approach is able to take uncertainty into account. Towards this aim, the uncertainty present in the measurement is assumed to be additive zero-mean Gaussian noise of standard deviation $\sigma_t$. In particular, different values of $\sigma_t = \{1, 2, 3, 4, 5\}\mu s$ are considered during the training process. The low-fidelity input data $\mathscr{D}_L = \left\{ \left( \mu_{\Delta t_{1i},k}, \sigma_{t,k} \right) \right\}_{k=1}^{N_L}$ is generated to train the network.

$$\mathscr{D}_L = \mu_{\Delta t_{1i}} \otimes \sigma_{t_i} \tag{29}$$

The operator $\otimes$ in Eq. (29) indicates the Kronecker product. The Adam optimizer is run for 50,000 epochs. A learning rate of 0.001 is used. The loss function used in training the PINN is

$$\mathscr{L}_R = \sum_{k=1}^{N_L} \|R_k\| \tag{30}$$

The variation of training loss of low-fidelity PINN with respect to the number of epochs is illustrated by the red line in Fig. 13. The results obtained from low-fidelity PINN are presented in Fig. 11. Table 3 summarises the estimated impact location and the average error in the estimation.

The results obtained using only data-driven ANN and low-fidelity PINN are discussed in Sections 6.1 and 6.2 respectively. Both data-driven ANN and low-fidelity PINN are found to yield erroneous results with an average error of 10.3217 cm and 3.5529 cm

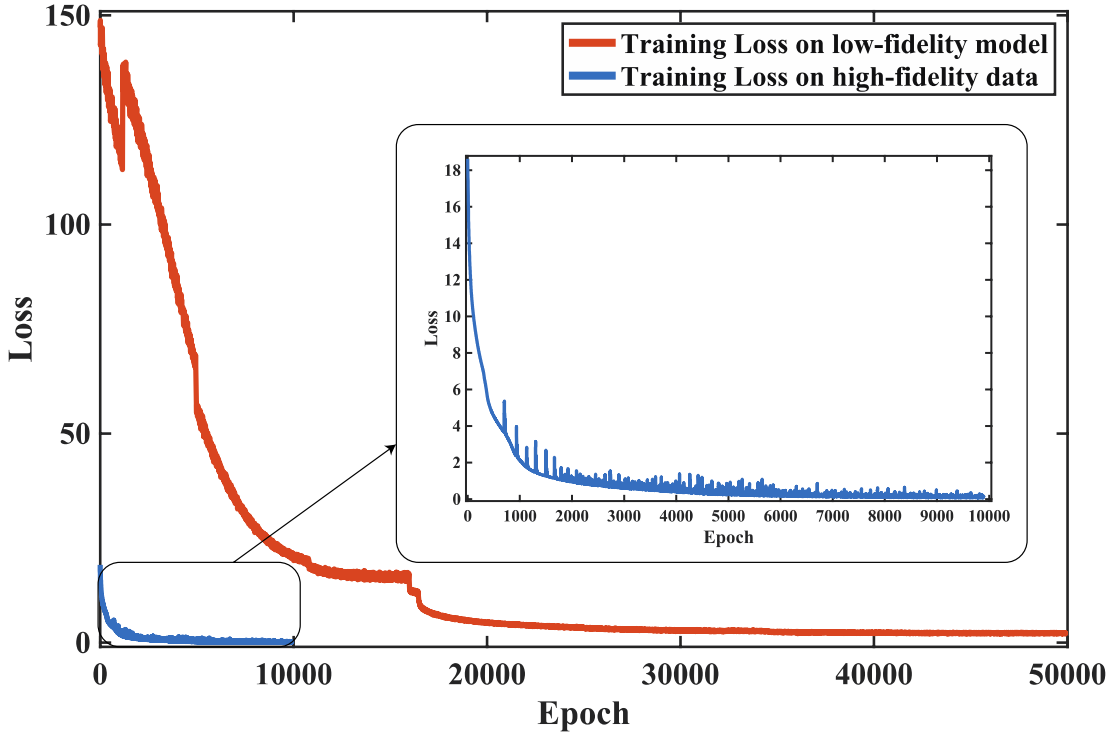



**Fig. 13.** The plot of the performance summary of the optimization algorithm during the low-fidelity and high-fidelity training processes.

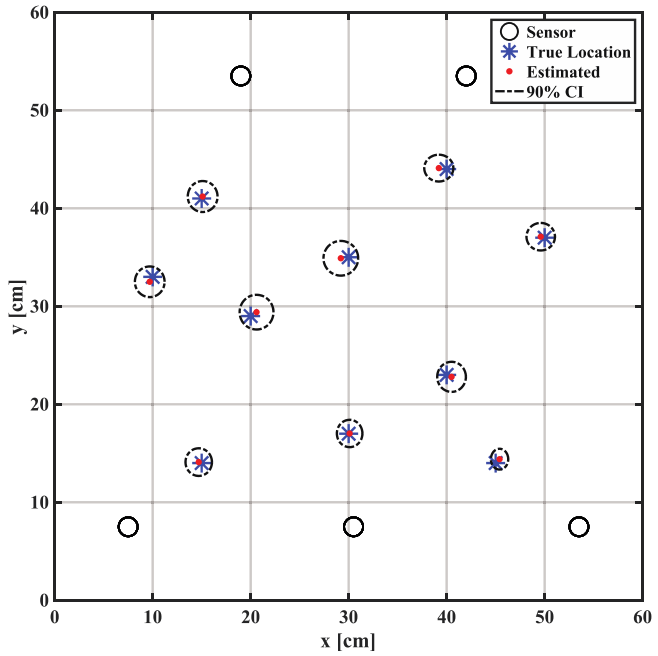



**Fig. 14.** Impact location results showing true and estimated locations using mfPINN.

**Table 4**
Source estimation results using mfPINN.

| Point | Actual Location | | Estimated Location | | $\varepsilon_r$ (cm) |
|---|---|---|---|---|---|
| | $x_s$ (cm) | $y_s$ (cm) | $\mu_{x_s}$ (cm) | $\mu_{y_s}$ (cm) | |
| 1 | 15 | 14 | 14.7 | 14.1 | 0.3162 |
| 2 | 30 | 17 | 30.1 | 17.02 | 0.1020 |
| 3 | 40 | 23 | 40.5 | 22.8 | 0.5385 |
| 4 | 10 | 33 | 9.7 | 32.5 | 0.5831 |
| 5 | 20 | 29 | 20.6 | 29.4 | 0.7211 |
| 6 | 40 | 44 | 39.2 | 44.1 | **0.8062** |
| 7 | 15 | 41 | 15.1 | 41.2 | 0.2236 |
| 8 | 50 | 37 | 49.6 | 37.1 | 0.4123 |
| 9 | 45 | 14 | 45.4 | 14.4 | 0.5657 |
| 10 | 30 | 35 | 29.2 | 34.9 | **0.8062** |
| | | | | | Average (cm) |
| | | | | | 0.5075 |

**Table 5**
Comparison of the performances of data-driven ANN, low-fidelity PINN and mfPINN in the AE source location prediction.

| Sl. No. | Type of Neural Network | Average Error (cm) | Maximum Error (cm) |
|---|---|---|---|
| 1 | data-driven ANN | 10.3217 | 19.9423 |
| 2 | low-fidelity PINN | 3.5529 | 7.8926 |
| 3 | mfPINN | 0.5075 | 0.8062 |

respectively and maximum error of 19.9423 cm and 7.8926 cm respectively. Because in the case of high-fidelity ANN, the access to high-fidelity data $\mathscr{D}_H$ is sparse, and the model overfitted the data. On the other hand, the number of data available in the case of PINN is not the issue; instead, the governing physics is an approximate representation of a real scenario. To rescue the problem, we proposed using a multi-fidelity framework that employs both the ANN and PINN. The utility of mfPINN is discussed in the following section 6.3.

### 6.3. Estimation of impact location using multi-fidelity PINN

In mfPINN, the network trained with the low-fidelity model, as discussed in Section 6.2 is updated with high-fidelity data by transfer learning. The basic premise here is to store knowledge gained by low-fidelity PINN while training on low-fidelity data to improve the efficiency of the final model after training on a few high-fidelity data. This is achieved by freezing the weights and biases of specific layers in a deep neural network. To retain the knowledge gained from the low-fidelity PINN and avoid overfitting, the network parameters corresponding to the last few layers are updated, and for other layers, the parameters are kept fixed. In the current work, only the last two layers are updated by using transfer learning based on the investigation performed as illustrated in Fig. 12. Here, the network's performance is monitored by computing the average error in predicting the AE source locations as we increase the number of last layers to be updated. From Fig. 12, one can observe that when only the last two layers are updated, the average error in predicting AE source location has achieved the minimum value. It should be noted that allowing more than two layers or all the layers to update will lead to possible loss of information gained and will overfit the model while training due to the small size of high-fidelity data. The network architecture of mfPINN considered is the same as before; the only change here is that the Adam optimizer is run for 10,000 epochs with high-fidelity data. Transfer learning is used to update the last two layers of the network. For the Adam optimizer, a learning rate of 0.001 is used. One must note that the training loss with high-fidelity data converged within 10,000 epochs only. On the other hand, it can be observed from Fig. 13 that 50,000 epochs are required to converge the training loss for the low-fidelity PINN.

The results obtained from low-fidelity PINN are presented in Fig. 14. Table 4 summarises the estimated impact location and the average error in the estimation. The results obtained using the proposed mfPINN are considerably more accurate as compared to data-driven ANN and low-fidelity PINN with an average error of 0.5075 cm and maximum error of 0.8062 cm.

Table 5 summarises the comparison of the performances of data-driven ANN, low-fidelity PINN and mfPINN in the prediction of AE source location.

## 7. Conclusion

In this study, a probabilistic framework based on mfPINN methods is proposed to estimate the location of AE sources in anisotropic panels. In particular, a deep neural network is trained with a low-fidelity model, followed by updating the network with high-fidelity data using transfer learning. These algorithms are applied for the situation where the wave velocity profile is known *a priori*. Experimental tests were carried out on a carbon-fibre reinforced polymer (CFRP) composite panel instrumented with a sparse array of piezoelectric transducers to collect high-fidelity data and evaluate the efficacy of the proposed framework. The AE sources were simulated by dropping a Teflon ball in order to show that the proposed framework can detect the AE source generated by such a small drop. The results show that the proposed approach can accurately estimate AE source locations with an average error of 0.5075 cm,

whereas 10.3217 cm and 3.5529 cm for data-driven ANN and low-fidelity PINN, respectively. However, an advantage of the framework over the other heuristic algorithm is its ability to provide a confidence interval of the estimated AE source location instead of a single point, despite the fact that the training cost of a deep neural network is generally higher. However, almost instant estimation can be done once the network has been trained successfully. Another advantage of the proposed method is its ability to consider uncertainty in sensor measurements. The accuracy and computational efficiency of the proposed framework demonstrate its potential application in real-time SHM applications.

## Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Amit Shelke reports financial support was provided by Indian Space Research Organisation. Anowarul Habib reports was provided by UiT The Arctic University of Norway. Not applicable.

## Data availability

Data will be made available on request.

## Acknowledgement

The authors would like to acknowledge Indian Space Research Organisation (ISRO) for supporting this work under grant number ISRO/RES/STC/IITG/2021-22. The publication charges for this article have been funded by a grant from the publication fund of UiT The Arctic University of Norway.

## References

[1] W. Staszewski, C. Boller, T. G.R., Health Monitoring of Aerospace Structures: Smart Sensor Technologies and Signal Processing, John Wiley and Sons Ltd., 2004.
[2] P. Coverley, W. Staszewski, Impact damage location in composite structures using optimized sensor triangulation procedure, Smart Mater. Struct. 12 (2003) 795.
[3] C.R. Farrar, N.A. Lieven, Damage prognosis: the future of structural health monitoring, Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 365 (2007) 623–632.
[4] L. Hui, O. Jinping, Structural health monitoring: from sensing technology stepping to health diagnosis, Procedia Eng. 14 (2011) 753–760.
[5] K.V. Jata, T. Kundu, T.A. Parthasarathy, An introduction to failure mechanisms and ultrasonic inspection, in: T. Kundu (Ed.), Advanced ultrasonic methods for material and structure inspection, Wiley-ISTE, London, 2007, pp. 1–42.
[6] V. Giurgiutiu, Lamb wave generation with piezoelectric wafer active sensors for structural health monitoring, Proc. SPIE 5056 (2003) 111–122.
[7] A.K. Mal, F. Shih, S. Banerjee, Acoustic emission waveforms in composite laminates under low velocity impact, Proc. SPIE 5047 (2003) 1–12.
[8] A.K. Mal, F. Ricci, S. Gibson, S. Banerjee, Damage detection in structures from vibration and wave propagation data, Proc. SPIE 5047 (2003) 202–210.
[9] E.D. Niri, A. Farhidzadeh, S. Salamone, Nonlinear Kalman Filtering for acoustic emission source localization in anisotropic panels, Ultrasonics 54 (2014) 486–501.
[10] T. Hajzargerbashi, T. Kundu, S. Bland, An improved algorithm for detecting point of impact in anisotropic inhomogeneous plates, Ultrasonics 51 (2011) 317–324.
[11] E.D. Niri, S. Salamone, A probabilistic framework for acoustic emission source localization in plate-like structures, Smart Mater. Struct. 21 (2012) 035009.
[12] F. Ciampa, M. Meo, A new algorithm for acoustic emission localization and flexural group velocity determination in anisotropic structures, Compos. A Appl. Sci. Manuf. 41 (12) (2010) 1777–1786.
[13] T. Kundu, S. Das, K.V. Jata, Detection of the point of impact on a stiffened plate by the acoustic emission technique, Smart Mater. Struct. 18 (2009) 035006.
[14] B. Park, H. Sohn, S.E. Olson, M.P. DeSimio, K.S. Brown, M.M. Derriso, Impact localization in complex structures using laser-based time reversal, Struct. Health Monit. 11 (2012) 577–588.
[15] L. Gaul, S. Hurlebaus, Identification of the impact location on a plate using wavelets, Mech. Syst. Sig. Process. 12 (1998) 783–795.
[16] T. Kundu, S. Das, K.V. Jata, Point of impact prediction in isotropic and anisotropic plates from the acoustic emission data, J. Acoust. Soc. Am. 122 (2007) 2057–2066.
[17] T. Kundu, H. Nakatani, N. Takeda, Acoustic source localization in anisotropic plates, Ultrasonics 52 (2012) 740–746.
[18] T. Kundu, Acoustic source localization, Ultrasonics 54 (2014) 25–38.
[19] K.P. Murphy, Machine learning: a probabilistic perspective, MIT Press, 2012.
[20] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press (2016).
[21] J. Haywood, P. Coverley, W.J. Staszewski, K. Worden, An automatic impact monitor for a composite panel employing smart sensor technology, Smart Mater. Struct. 14 (2004) 265.
[22] J. LeClerc, K. Worden, W.J. Staszewski, J. Haywood, Impact detection in an aircraft composite panel—a neural-network approach, J. Sound Vib. 299 (2007) 672–682.
[23] J. Hensman, R. Mills, S. Pierce, K. Worden, M. Eaton, Locating acoustic emission sources in complex structures using Gaussian processes, Mech. Syst. Sig. Process. 24 (2010) 211–223.
[24] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 113 (2016) 3932–3937.
[25] S. Chakraborty, Transfer learning based multi-fidelity physics informed deep neural network, J. Comput. Phys. 426 (2021) 109942.
[26] C. Bierig, A. Chernov, Approximation of probability density functions by the multilevel Monte Carlo maximum entropy method, J. Comput. Phys. 314 (2016) 661–681.
[27] I. Kaymaz, Application of kriging method to structural reliability problems, Struct. Saf. 27 (2005) 133–151.
[28] I. Bilionis, N. Zabaras, Multi-output local Gaussian process regression: Applications to uncertainty quantification, J. Comput. Phys. 231 (2012) 5718–5746.
[29] B. Liu, S. Koziel, Q. Zhang, A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems, J. Comput. Sci. 12 (2016) 28–37.
[30] C. Park, R.T. Haftka, N.H. Kim, Remarks on multi-fidelity surrogates, Struct. Multidiscip. Optim. 55 (2017) 1029–1050.
[31] P.S. Palar, T. Tsuchiya, G.T. Parks, Multi-fidelity non-intrusive polynomial chaos based on regression, Comput. Methods Appl. Mech. Eng. 305 (2016) 579–606.
[32] S. De, J. Britton, M. Reynolds, R. Skinner, K. Jansen, A. Doostan, On transfer learning of neural networks using bi-fidelity data for uncertainty propagation, Int. J. Uncertain. Quantif. 10 (6) (2020) 543–573.

[33] X. Meng, G.E. Karniadakis, A composite neural network that learns from multi-fidelity data: application to function approximation and inverse PDE problems, J. Comput. Phys. 401 (2020) 109020.
[34] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[35] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, J. Comput. Phys. 394 (2019) 56–81.
[36] N. Sen, T. Kundu, A new wave front shape-based approach for acoustic source localization in an anisotropic plate without knowing its material properties, Ultrasonics 87 (2018) 20–32.
[37] W.H. Park, P. Packo, T. Kundu, Acoustic source localization in an anisotropic plate without knowing its material properties–a new approach, Ultrasonics 79 (2017) 9–17.
[38] N. Sen, M. Gawroński, P. Packo, T. Uhl, T. Kundu, Square-shaped sensor clusters for acoustic source localization in anisotropic plates by wave front shape-based approach, Mech. Syst. Sig. Process. 153 (2021) 107489.
[39] N. Sen, T. Kundu, A new signal energy-based approach to acoustic source localization in orthotropic plates: a numerical study, Mech. Syst. Sig. Process. 171 (2022) 108843.
[40] E.D. Niri, A. Farhidzadeh, S. Salamone, Determination of the probability zone for acoustic emission source location in cylindrical shell structures, Mech. Syst. Sig. Process. 60 (2015) 971–985.
[41] G. Andria, F. Attivissimo, N. Giaquinto, Digital signal processing techniques for accurate ultrasonic sensor measurement, Measurement 30 (2001) 105–114.
[42] B. Lei, G. Xu, M. Feng, Y. Zou, F. Van der Heijden, D. De Ridder, D.M. Tax, Classification, parameter estimation and state estimation: an engineering approach using MATLAB, John Wiley & Sons, 2017.
[43] M. Parrilla, J. Anaya, C. Fritsch, Digital signal processing techniques for high accuracy ultrasonic range measurements, IEEE Trans. Instrum. Meas. 40 (1991) 759–763.
[44] Y.C. Wu, J.W. Feng, Development and application of artificial neural network, Wirel. Pers. Commun. 102 (2018) 1645–1656.
[45] G. Zhou, Y. Ji, X. Chen, F. Zhang, Artificial neural networks and the mass appraisal of real estate, Int. J. Online Eng. 14 (03) (2018) 180.
[46] R. Benzer, Population dynamics forecasting using artificial neural networks, Fresen. Environ. Bull. 12 (2014) 14–26.
[47] P. Kim, Matlab deep learning with machine learning, neural networks and artificial intelligence, Apress, 2017.
[48] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (2021) 422–440.
[49] S. Goswami, C. Anitescu, S. Chakraborty, T. Rabczuk, Transfer learning enhanced physics informed neural network for phase-field modeling of fracture, Theor. Appl. Fract. Mech. 106 (2020) 102447.
[50] G.P. Pun, R. Batra, R. Ramprasad, Y. Mishin, Physically informed artificial neural networks for atomistic modeling of materials, Nat. Commun. 10 (2019) 2339.
[51] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. 2 (1989) 303–314.
[52] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359–366.
[53] K. Hornik, M. Stinchcombe, H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, Neural Netw. 3 (1990) 551–560.
[54] B. Hanin, Universal function approximation by deep neural nets with bounded width and relu activations, Mathematics 7 (2019) 992.
[55] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, Adv. Neural Inf. Proces. Syst. 25 (2012).