Received: 15 October 2023 | Revised: 24 November 2023 | Accepted: 22 December 2023 | Published online: 25 December 2023

Journal of Data Science and Intelligent Systems
2025, Vol. 3(2) 149–164
DOI: 10.47852/bonviewJDSIS32021771

BON VIEW PUBLISHING

RESEARCH ARTICLE

Signal Process of Ultrasonic Guided Wave for Damage Detection of Localized Defects in Plates: From Shallow Learning to Deep Learning

Li Shang[1], Zi Zhang[1,2], Fujian Tang[3], Qi Cao[3], Hong Pan[1,*] and Zhibin Lin[1,*]

[1]*Department of Civil and Environmental Engineering, North Dakota State University, USA*

[2]*School of Civil Engineering, Jilin Jianzhu University, China*

[3]*School of Civil Engineering, Dalian University of Technology, China*

Abstract: Ultrasonic guided waves are one of the non-destructive inspection techniques used in structural health monitoring for localized damage detection. Even though propagation, scattering, and dispersion of the Lamb waves have great progress in the monitoring technology of the Lamb wave, the detection, identification, and quantification of structure damage still face challenges because of the complexity of the process in the propagation, scattering, and dispersion of the Lamb wave. Machine learning methods, including shallow learning (support vector machine (SVM)) and deep learning (DL) methods (convolutional neural networks (CNNs) and long short-term memory (LSTM)), in recent years have brought revolutionary opportunities for decoding the information of the Lamb wave. Therefore, the methodology structure was proposed from dataset collection, data preprocessing (including feature extraction, feature combination, and feature transformation), data training, and classification. Two different cases of damage types and damage sizes were designed in a COMSOL environment. The shallow learning method of the SVM model and the DL method of the CNN-LSTM model were compared with the defined time series features and transformed images. The results showed that both shallow learning methods and DL methods can be used in the application of signal classification, while the DL method of CNN-LSTM exhibited higher accuracy in image classification, as compared to the SVM. The robustness of the proposed models has also been verified under noise interference. The results demonstrated that the DL architecture of CNN-LSTM has the potential to attain greater precision because of better feature extraction and processing ability than the shallow learning model of SVM. In addition, the performance of signal classification and image classification of SVM and CNN-LSTM models dramatically decreased as the noise levels increased.

Keywords: shallow learning models, deep learning models, Lamb wave, data processing, damage detection, non-destructive testing

1. Introduction

Civil engineering structures form the cornerstone of societal economic advancement. To prevent building failure and enhance social stability, structural health monitoring (SHM) is essential for understanding structural working performance [1, 2], evaluating structural working status, and facilitating immediate and effective decision-making [3]. Guided wave-based approaches are frequently able to recognize more minor damage and a little abnormality in a real-time manner [4], while traditional vision-oriented methods [5] or vibration-centric approaches [6, 7] are usually responsive solely to substantial damage.

Apart from signal processing based on physics principles, data-centric methods have gained recognition in the last 10 years, including the use of deep learning (DL) and artificial neural networks (ANNs) [8–10]. In many signal classification systems, waveform-based deep neural networks are now required [11–15]. These advanced neural networks directly utilize unprocessed signals as input in various tasks [16], such as identifying stress levels of structures, assessing the condition of infrastructure, identifying structural damage, and detecting structural damage [17], diagnosing structural damage [18, 19], and monitoring structural health [20, 21].

For example, Yang et al. [22] used the Bayesian technique to predict the size of the crack. Chen et al. [23] developed an algorithm for locating structure damage, which increases the model's accuracy and reduces the training time. Das et al. [24] developed an OC-support vector machine (SVM) model for classifying the structure damage and demonstrated the robustness of the model with more complex experimental cases. Legendre et al. [25] developed a neural network automated model for Lamb wave to recognize the metallic weldments. Su and Ye [26] quantitatively identified structure defects based on ANNs, whose inputs are also Lamb waves. Veiga et al. [27] used ANNs to decode and classify ultrasonic waves to recognize different fault types.

Few studies compare the effectiveness of the features from time-series signals and images transformed from signals. This study fills the knowledge gap and compares the training performance of the shallow learning model of SVM and the DL model of convolutional neural

*__Corresponding authors:__ Hong Pan, Department of Civil and Environmental Engineering, North Dakota State University, USA. Email: hong.pan@ndsu.edu and Zhibin Lin, Department of Civil and Environmental Engineering, North Dakota State University, USA. Email: zhibin.lin@ndsu.edu

© The Author(s) 2023. Published by BON VIEW PUBLISHING PTE. LTD. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

network (CNN)-long short-term memory (LSTM) with the same training datasets. To demonstrate the robustness of these models, different COMSOL scenarios were designed, and different noise levels were considered during model training. Our study innovatively applies ultrasonic guided waves transformation to image in SHM, addressing unique challenges. While prior research transformed data types into images, our focus on Lamb waves, machine learning methods, and noise interference distinguishes our contribution, advancing signal and image classification techniques in SHM. We contribute by demonstrating the effectiveness of these models in classifying Lamb wave signals, showcasing their robustness and potential for precision in real-world scenarios.

This paper is structured in the following manner. Section 2 presents the methodology of SVM and CNN-LSTM models, feature calculation and selection, and models' evaluation. In Section 3, different cases of the COMSOL model were built and examples of signals and images were presented. In Section 4, the classification results of training accuracy, area under the receiver operating characteristic (ROC) curve (AUC) values, and ROC curves of SVM and CNN-LSTM models were presented, compared, and discussed. Further discussion about the effectiveness and robustness of the proposed methods was discussed in Section 5. Section 6 concludes with its conclusions at the end.

2. Machine Learning-Infused Approach for Damage Identification Framework

The structure of methodologies is depicted in Figure 1. In this study, the time and frequency features of signals were calculated and trained by two different neural network models, including the shallow learning model of SVM and the DL model of CNN-LSTM. To evaluate the superiority of DL models, the training effectiveness of the CNN-LSTM model was contrasted with that of the SVM model. Images transformed from signals were input to SVM and CNN-LSTM models for the purpose of testifying the effectiveness of feature extraction from images and the results were compared with signals. Different cases based on the COMSOL model were designed to evaluate the robustness of the developed methods. To further discuss the robustness of the CNN-LSTM DL model, different levels of noise interference were considered, and more complex cases based on COMSOL models were designed.

2.1. SVM model

SVM is a renowned machine learning technique in shallow learning, first introduced by Vapnik [28]. The fundamental concept underpinning SVM's categorization is to construct a hyperplane that separates the training samples into two distinct categories [17]. By using a kernel function, it maps the input series into a characteristic space with a higher dimension [17]. In this study, radius basis function was built as the function of kernel in the structure of SVM. The separation is then made by establishing an ideal hyperplane in that feature space and maximizing the distance between it and the nearest sample points within each class [17].

A collection of training samples $u_1, y_1), \ldots, (u_i, y_i), \in R^N$ that correspond to two categories $(y_i = \{-1, 1\})$ has been taken into consideration. The goal is to locate the hyperplane to divide the samples [17, 29]:

$$v \cdot u + c = 0, \; v \in R^N, \; c \in R^N \tag{1}$$

where v is the weight vector and c is the bias term. The feature domain for the linear classification model is then displayed [17].

$$f(u) = sgn\,(v \cdot u + c) \tag{2}$$

Multiple linear classifiers could classify the samples in a two-dimensional (2D) space [17]. The best hyperplane to find is the one with the highest margin and this is known as the ideal hyperplane [17]. Consequently, all training data meet the following requirements [17, 30].

$$v \cdot u_i + c \;\geq\; +1 \; for \; y_i = \; +1 \tag{3}$$

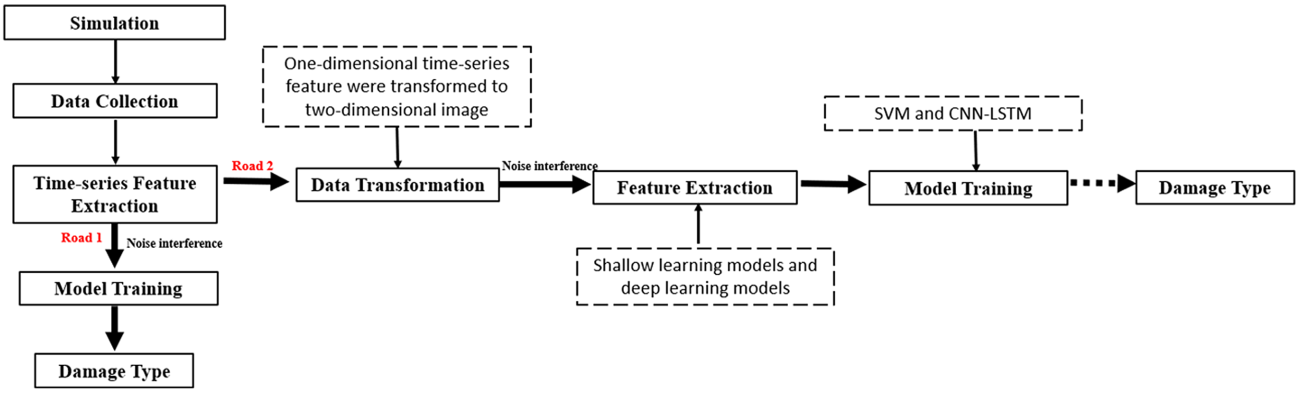
$$v \cdot u_i + c \;\leq\; -1 \; for \; y_i = \; -1 \tag{4}$$

The spatial distance between samples and the ideal hyperplane $(v, \; c)$ is depicted as follows [17].

$$L((v, \; c), u_i) = \frac{y_i(v \cdot u_i + c)}{||v||} \geq \frac{+1}{||v||} \tag{5}$$

The maximum distance to the nearest samples should be calculated to find the ideal hyperplane. Finding the least value of $||v||$ is equivalent to

Figure 1
Illustrative representation of the research approach



finding the maximum distance from Equation (5). As a result, finding the ideal hyperplane could also be transformed into the problem of convex quadratic programming [17, 31].

$$\min(\varnothing(\mathrm{v})) = \frac{1}{2}||v||^2 \tag{6}$$

The main approach for identifying the local peaks and valleys of a function with equality restrictions is the Lagrange multiplier. The goal can be changed into [17, 31]

$$\varphi(v, c, \mathrm{P}) = \frac{1}{2}||v||^2 - \sum_{i=1}^{n} \mathrm{y}_i[y_i(v^T \cdot u_i + c) - 1] \tag{7}$$

where $P = (\Upsilon_1 \cdots \Upsilon_n)^T$ represents the Lagrange multiplier. The $\varphi(v, B, \mathrm{P})$ must be maximized with regard to φ, whereas v and c must be minimized.

By Zhang et al. [17] as well as Burbidge and Buxton [31], the decision function is provided

$$F(u) = sgn\left(\sum_{i=1}^{n} \Upsilon_i{}^* y_i K(u,\ u_i) + c\right) \tag{8}$$

where $K(u,\ u_i)$ stands for the function of kernel [17, 31].

$$K(u,\ u_i) = \exp(-\lambda||u_i - u_i||^2),\ \ \lambda > 0 \tag{9}$$

Generally, the kernel function can help samples data to project into a higher hyperplane, which improves the ability of linear classification [17]. And with the use of the kernel function, SVM may be made significantly more appropriate for more complex samples, for example, non-linear classification [17].

2.2. Fusion model of CNN and LSTM (CNN-LSTM)

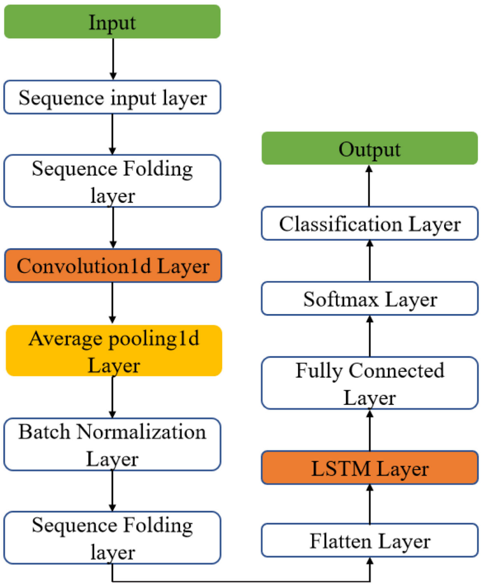
A CNN-LSTM hybrid network was presented in this study and Figure 2 shows its structure. The goal of the CNN layer is to capture the features from temporal and spectral domains from the monitoring signal data and condense the data series. But the only disadvantage of the CNN network is that it is hard to express the temporal information of signals, which is significant to time-series signals. In order to further analyze the time series features, the collected features were then placed into the LSTM layer. The next sections illustrate how CNN and LSTM extract features and process those features. Batch normalization layers were built into CNN-LSTM models to standardize the outputs from every layer, which reduced the danger of overfitting and improved the training process's robustness. Xu et al. [32] and Andhale et al. [33] have also shown how the batch normalization layers work.

The CNN is a widely recognized DL method that can process data in low and high dimensions [34]. The two primary layers in the CNN structure are the convolutional layer and the average pooling layer [35].

The role of convolutional layer is to create characteristic maps by performing convolution and activation operations on the input vector [36]. Presented in Rani and Devarakonda [37], the mathematical formula of convolution layer is as follows:

$$\phi_j = F\left(\sum\nolimits_{mi\ E\ M} X_i \times K_j + C_j\right) \tag{10}$$

Figure 2
The architecture of the hybrid CNN-LSTM model



where $F(\cdot)$ is the activation function, X_i stands for the input vector, K_j represents the kernel function, and C_j represents the bias vectors.

The average pooling layer, which comes after the convolutional layer, is to reduce the spatial dimensions (width and height) of the feature maps while retaining their depth (number of channels) and speed up model calculation. The mathematical representation of the pooling procedure is shown as in Rani and Devarakonda [37]:

$$\varphi_j = \beta_j\ down\left(c_j\right) + B_j K_j \tag{11}$$

where the average pooling method is represented by $down(\cdot)$.

For CNN network, the time series data were put into the input layer, the convolutional layer was to capture features of input vectors, and the purpose of the average pooling layer was to decrease the volume of training samples originating from the convolutional layer, therefor avoiding overfitting [38]. At last, the trained samples from CNN structure were flattened into the LSTM layer.

LSTM serves as a powerful tool for handling sequential and temporal data, particularly in tasks related to categorization and predictive modeling [39]. One of the important layers of the LSTM structure is the sequence input layer, which helps inputting the temporal data. Another critical layer is the LSTM layer, which has the memorization function during the data calculation process. The last layer is the output layer. The specific algorithms in LSTM model are as follows [40].

Input gate (in_t), forget gate (for_t), cell candidate (g_t), and output gate (out_t) are four key parts that has been used during the data processing procedure, the specific algorithms of ever part are as follows [40]:

$$in_t = \sigma_g\left(W_{in}x_t + R_{in}h_{t-1} + C_{in}\right) \tag{12}$$

$$for_t = \sigma_g\left(W_{for}x_t + R_{for}h_{t-1} + C_{for}\right) \tag{13}$$

$$g_t = \sigma_c\left(W_g x_t + R_g h_{t-1} + C_g\right) \tag{14}$$

$$out_t = \sigma_g\left(W_{out}x_t + R_{out}h_{t-1} + C_{out}\right) \tag{15}$$

where in_t stands for input gate, for_t stands for forget gate, g_t stands for cell candidate, and out_t stands for output gate; they have been used during the data processing procedure. t stands for the step of time, and σ_g represents the gate activation function. W represents the matrices of input weights, R represents the matrices of recurrent weights, and C represents the matrices of bias vector.

The signals collected from COMSOL models were highly time dependent, and LSTM structure is developed and is good at capturing time information from time-series signals from CNN layers.

2.3. Features extraction

2.3.1. Definition of features

Time and frequency domain feature parameters were chosen to better signify the information for the signals, as shown in Table 1. And y_1 stands for the frequency vibration energy, $y_2 - y_4$, y_6 and $y_{10} - y_{13}$ stand for spectrum dispersion or concentration; and y_5 and $y_7 - y_9$ represent the position difference of the main frequency [41]; they have been selected based on Chen [41], Shang et al. [16], and Shang et al. [35].

2.4. Data preprocessing

Grayscale maps, also known as grayscale images (GSIs), are a common form of input data in various applications, including image processing and DL. Grayscale maps are essentially 2D representations of data, where each pixel in the image encodes a specific value, typically representing intensity or brightness.

Two-dimensional GSIs are 2D data matrices, which can have much more information than one-dimensional (1D) signals. In addition, DL models are much good at image processing tasks than machine learning models based on previous studies. This study transformed the 1D signal series after feature extraction into the 2D image on the basis of the phase space reconstruction theory [42] and the images were used as the inputs of the training models. The pixels of the time-series signals are filled in order according to the prearranged series [43]. The time-series signals are used to arrange the pixel order in the GSI [43].

$$g(i,j) = \frac{x[(i-1)\times N + j] - \min[x(k)]}{\max[x(k)] - \min[x(k)]} \tag{16}$$

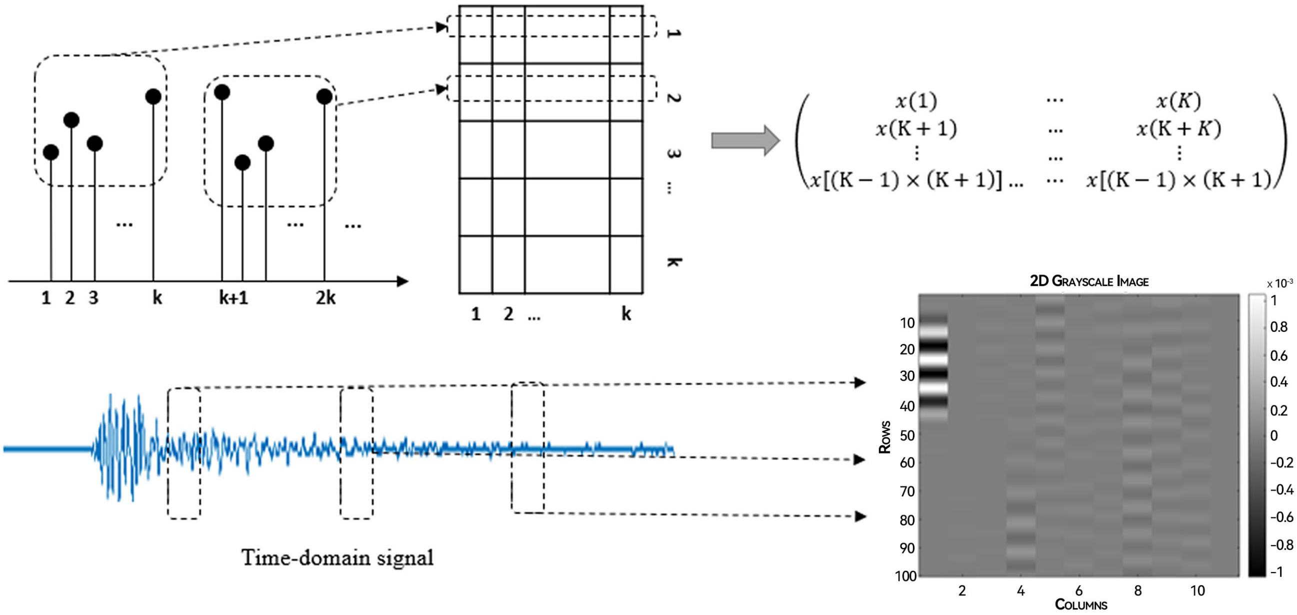
where $g(i,j)$ represents the pixel intensity at the point (i,j) in the image, i ranges from 1 to N, and j ranges from 1 to N. N^2 is assumed to be the length of the time-series signals, and $x(k)$ denotes the value at the position indexed by k [43].

Figure 3 shows the process of transforming signals to images based on Sun et al. [43].

Table 1
Time and frequency domain feature descriptors

Time domain feature descriptors			
Feature identifier	Expressions	Features identifier	Expressions
Average value	$\overline{U} = \frac{1}{K}\sum_{j=1}^{K} u_j$	Kurtosis	$\beta = \frac{1}{K}\sum_{j=1}^{K} u_j^4$
Root mean square (RMS) magnitude	$U_{rms} = \sqrt{\frac{1}{K}\sum_{j=1}^{K} u_j^2}$	Variance	$\sigma_x^2 = \frac{1}{K-1}\sum_{j=1}^{K}\left(u_j - \overline{U}\right)^2$
Amplitude square root	$U_r = \left[\frac{1}{K}\sum_{j=1}^{K}\sqrt{\lvert u_j\rvert}\right]^2$	Peak value	$U_{max} = max\{\lvert u_j\rvert\}$
Mean absolute amplitude	$\lvert\overline{U}\rvert = \frac{1}{K}\sum_{j=1}^{K}\lvert u_j\rvert$	Minimum point	$U_{min} = min\{\lvert u_j\rvert\}$
Skewness	$\propto = \frac{1}{K}\sum_{j=1}^{K} u_j^3$	Peak-to-peak magnitude	$U_{p-p} = U_{max} - U_{min}$
Waveform indicator	$W_f = \frac{U_{rms}}{\lvert\overline{U}\rvert}$	Peak indicator	$C_p = \frac{U_{max}}{U_{rms}}$
Pulse indicator	$P_f = \frac{U_{max}}{\lvert\overline{U}\rvert}$	Margin indicator	$CL_{mar} = \frac{U_{max}}{U_r}$
Kurtosis indicator	$K_v = \frac{\beta}{u_{rms}^4}$	Skewness indicator	$S_s = \frac{\propto}{u_{rms}^3}$
Frequency domain feature descriptors			
Number	Expressions	Number	Expressions
1	$y_1 = \frac{\sum_{n=1}^{N} f(n)}{N}$	8	$y_8 = \sqrt{\frac{\sum_{n=1}^{N} l_n{}^4 f(n)}{\sum_{n=1}^{N} l_n{}^2 f(n)}}$
2	$y_2 = \frac{\sum_{n=1}^{N}(f(n)-y_1)^2}{N}$	9	$y_9 = \frac{\sum_{n=1}^{N} l_n{}^2 f(n)}{\sqrt{\sum_{n=1}^{N} f(n)\sum_{n=1}^{N} l_n{}^4 f(n)}}$
3	$y_3 = \frac{\sum_{n=1}^{N}(f(n)-y_1)^3}{N\left(\sqrt{y_2}\right)^3}$	10	$y_{10} = \frac{y_6}{y_5}$
4	$y_4 = \frac{\sum_{n=1}^{N}(f(n)-y_1)^4}{Ny_2^2}$	11	$y_{11} = \frac{\sum_{n=1}^{N}(l_n-y_5)^3 f(n)}{Ny_6^2}$
5	$y_5 = \frac{\sum_{n=1}^{N} l_n f(n)}{\sum_{n=1}^{N} f(n)}$	12	$y_{12} = \frac{\sum_{n=1}^{N}(l_n-y_5)^4 f(n)}{Ny_6^4}$
6	$y_6 = \sqrt{\frac{\sum_{n=1}^{N}(l_n-y_5)^2 s(n)}{N}}$	13	$y_{13} = \frac{\sum_{n=1}^{N}(l_n-y_5)^{0.5} f(n)}{Ny_6}$
7	$y_7 = \sqrt{\frac{\sum_{n=1}^{N} l_n{}^2 s(n)}{\sum_{n=1}^{N} s(n)}}$		

Figure 3
Process of transforming 1D time domain signals to 2D grayscale images



The dimensions of a GSI are typically defined by its width (number of columns) and height (number of rows); 256 × 256 is 256 pixels wide and 256 pixels tall.

Grayscale maps are used to represent data in a visual format, where variations in intensity or brightness correspond to variations in the underlying data. Each pixel's intensity in the GSI represents a specific value or measurement. The darker pixels typically represent lower values, while lighter pixels represent higher values. Grayscale maps are versatile and can be used to represent various types of data, including but not limited to images, time-series data, and sensor readings. They provide a convenient way to visualize and analyze data, making it easier for DL models to extract patterns and features.

GSIs are commonly used as inputs to neural networks, especially in tasks where color information is not necessary. When used as inputs, GSIs are often treated as multi-channel images with only one channel, where each channel corresponds to the grayscale intensity. DL models, like CNNs, are particularly well-suited for processing GSIs. In CNNs, the dimensions of the input layer are determined by the width, height, and number of channels in the image. DL models can learn to extract meaningful features from GSIs, making them suitable for a wide range of tasks, including image classification, object detection, and segmentation.

In the study, the process begins with a raw signal series, a 1D representation obtained from simulation. Signal processing techniques are then applied to extract time and frequency features, encompassing statistical measures and transformations like Fourier transforms. These features are amalgamated to form a new time series, retaining essential information from the raw signal. GSIs are subsequently derived from this time series using methodologies such as phase space reconstruction. Each pixel in these images corresponds to a value normalized through a specific equation, ensuring consistent representation. The dimensions of the GSIs, commonly 256 × 256 pixels, are defined, making them ready for input in tasks like image classification using DL models. This process enables effective analysis and pattern extraction from the original signal series.

2.5. Evaluation of model performances

The confusion matrix is a widely used method for handling classification problems. It provides a means to assess classification accuracy through the following formula:

$$Accuracy = \frac{A + D}{A + B + C + D} \tag{17}$$

In this formula, A represents the proportion of correct negative predictions, B represents the proportion of incorrect positive predictions, C represents the proportion of precise negative predictions, and D represents the proportion of precise positive predictions.

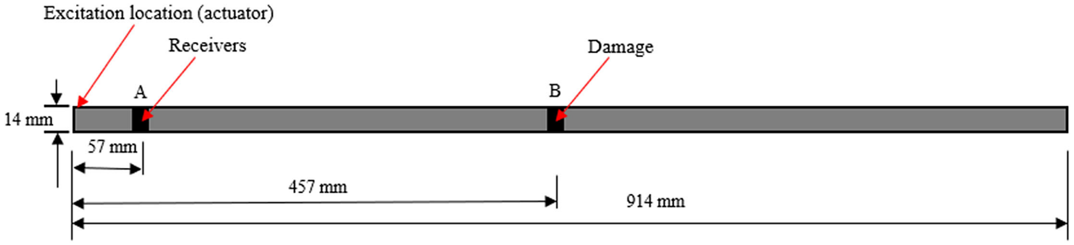
The ROC curves were used as the performance evaluation tool in machine learning and DL [16]. ROC curves are generated by contrasting the true positive rate with the false positive rate at various threshold levels [17]. The degree or measure of separability was described by the AUC [16, 17]. A higher AUC is associated with superior model performance. The AUC reaches 1 when a model achieves 100% accuracy [16].

3. Datasets Generated from Lamb Wave Approaches

3.1. Model construction from COMSOL

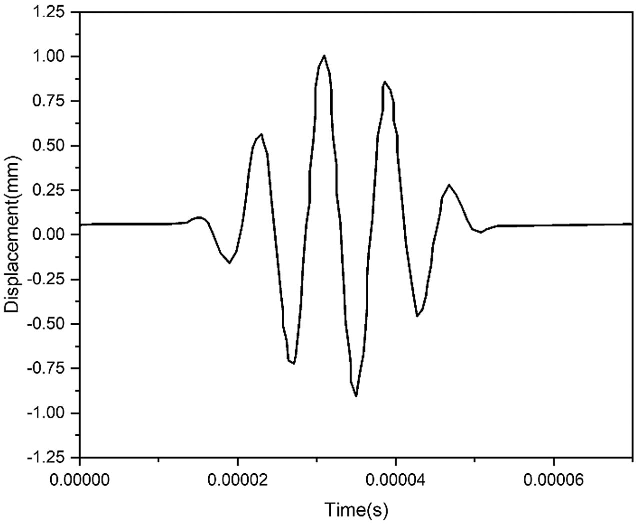
The COMSOL model was established based on Zhang et al. [17]. Figure 4 depicts the COMSOL model of aluminum beam, the thickness of the plate was 1.6 mm, and its measurements were 914 mm by 14 mm. The piezo actuator was positioned at the leading edge of the beam, and the receiver was installed at point A, which was 57 mm away from the left side. The damage was situated at point C, which was at a distance of 457 mm from the left side. To replicate the damage's shape, a notch with an 8 mm thickness passing completely through was created in COMSOL. The specific model design information can be referenced

Figure 4
COMSOL model of aluminum beam



in Zhang et al. [17]. A five-cycle sine function with a Hanning window of 100 kHz frequency was used to define the excited signal, as shown in Figure 5.

Figure 5
Excited guided wave



3.2. Design of scenarios

Two cases were designed in this study. In Case 1, there were five different types of damage types with the length of 6 mm, including notch-shaped damage, circular-shaped damage, square-shaped damage, diamond-shaped damage, and oval-shaped damage. And these damages were placed at point C, as shown in Figure 4. In Case 2, the influence of damage size was studied and different sizes of the oval-shaped damage with 90 degrees were designed, including 2-mm long, 4-mm long, 6-mm long, 8-mm long, 10-mm long, and 12-mm long. The detailed information about the damage design is shown in Table 2. The study utilized a dataset that was split into training, validation, and test sets. Specifically, 70% of the data was designated for training, with 15% each allocated for validation and testing. Evaluation involved using data not previously seen by the models during their training phase, ensuring that both the validation and test sets included unseen data. This methodology aimed to assess the models' capacity to generalize to new and undisclosed data, offering insights into their overall performance on unfamiliar datasets. Figure 6 shows the waveform information of the designed cases.

To study the influence of different forms of training inputs, the time-series signals were transformed into images. Taking the oval-shaped damage with different sizes (Case 2) as an example, Figure 7 shows the information of transformed images.

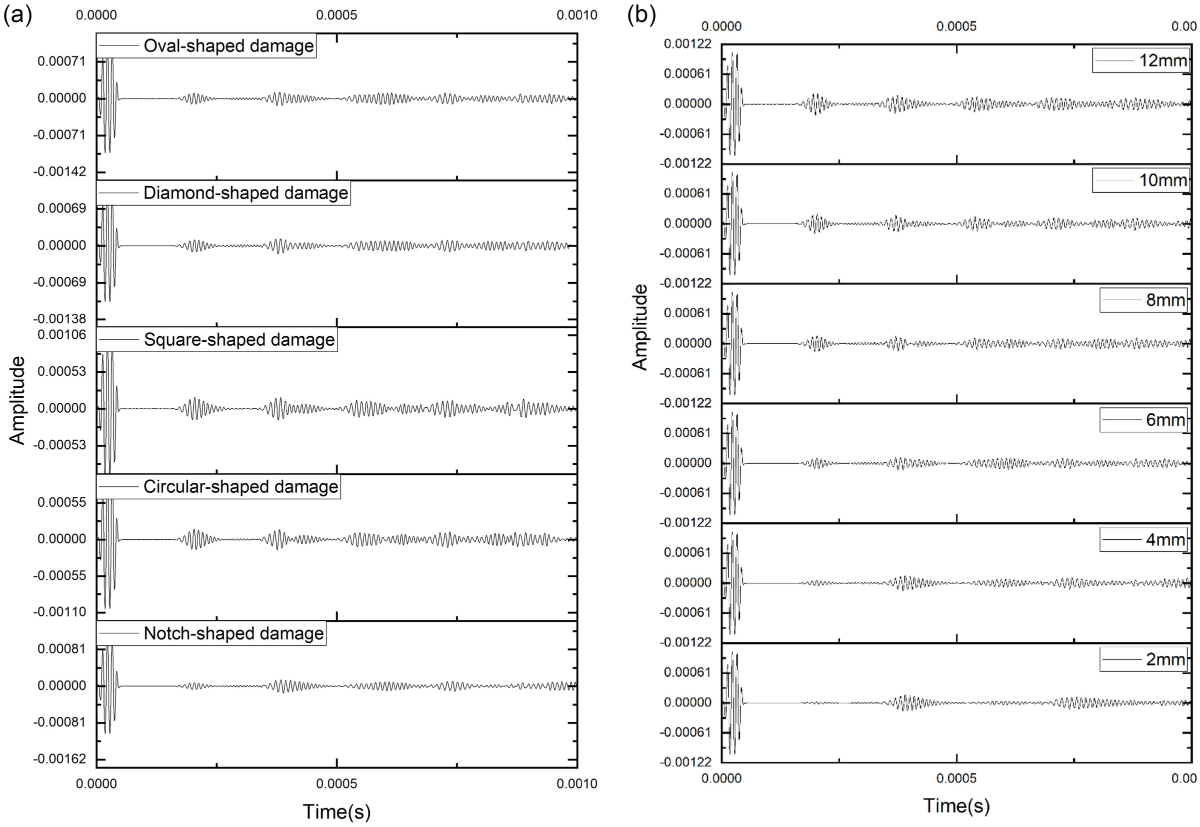
3.3. Signals with noise interference

Contrary to signals gathered from a modeling environment, signals collected from building sites are seriously polluted by noise. An ideal environment is one that does not interfere with signals with noise, such as a simulation environment. Different noise levels would be added to simulation signals to better represent the real-world condition to assess the efficacy of the methodology utilized in this

Table 2
Experimental design for computational modeling

Case	Label	Defects type	Defects size	Defects orientation	Noise interference
Base	State #1	/	/	/	
Case 1: variance due to the defects type	State #2	Notch-shaped damage	6 mm long	90 degrees	
	State #3	Circular-shaped damage	6 mm diameter	/	
	State #4	Square-shaped damage	6 mm long	/	From 3 dB to 15 dB
	State #5	Diamond-shaped damage	6 mm long	/	
	State #6	Oval-shaped damage	6 mm long	/	
Case 2: variance due to defects size	State #7	Oval-shaped damage	2 mm long	/	
	State #8	Oval-shaped damage	4 mm long	90 degrees	
	State #6	Oval-shaped damage	6 mm long	90 degrees	From 3 dB to 15 dB
	State #9	Oval-shaped damage	8 mm long	90 degrees	
	State #10	Oval-shaped damage	10 mm long	90 degrees	
	State #11	Oval-shaped damage	12 mm long	90 degrees	

Figure 6
Plate waveforms with different damages



Signals of five types of defects with 6-mm size

Signals of oval-shaped damage with different sizes

study. Figure 8 depicts the signals with various types of noise interference, using 2-mm oval-shaped damage as an example. The original signal, which is an ideal signal devoid of noise interference, is gathered from a simulated environment. Figure 8 illustrates how difficult it is to distinguish the signal from the noise at 3 dB noise level. In a 3 dB noise environment, the signal and noise have approximately equal power, making it difficult to discern the signal clearly. As the noise levels dropped, the signal improved dramatically in clarity and strength. The tainted signal was nearly identical to the original signal when the noise level is 15 dB.

Figure 9 illustrates the images of 2-mm oval-shaped damage on different noise levels. Compared with the original images, the noised images have much more contaminated information in the images, which confused with the useful information. For instance, when the noise levels increased to 3 dB, the effective information in the images was totally covered by the noised information. While when the noise levels decreased to 15 dB, it is clear to see that the images have the same information as the original images.

4. Results and Discussion

4.1. The classification accuracy of SVM and CNN-LSTM models with signals and images as input without noise interference (Case 1)

To quantitatively evaluate the difference from shallow learning to DL, this study constructed SVM as shallow learning models and the CNN-LSTM model as DL models. There are two different types of input data used to train these models, including signals of time and frequency feature series and images of time and frequency feature series. The classification capabilities of SVM and CNN-LSTM models were compared. Training curve and accuracies were used as evaluation indicators of the classification performance. Table 3 and Figures 10 and 11 show the comparison results of the evaluation indicators of SVM and CNN-LSTM models. When signals of time and frequency feature series were used as input to train the models, the two models (SVM and CNN-LSTM) can achieve the best performance and up to 100% accuracy. But when images of time and frequency feature series were used as input to train the models, only the DL model of CNN-LSTM can achieve 100% accuracy, while the shallow learning model of SVM can only achieve 90% accuracy. The findings showed that the two models of SVM and CNN-LSTM can be successfully used at signal classification tasks and the DL model of CNN-LSTM can be successfully applied in image classification tasks, while the shallow learning model of SVM is much bad at image classification tasks.

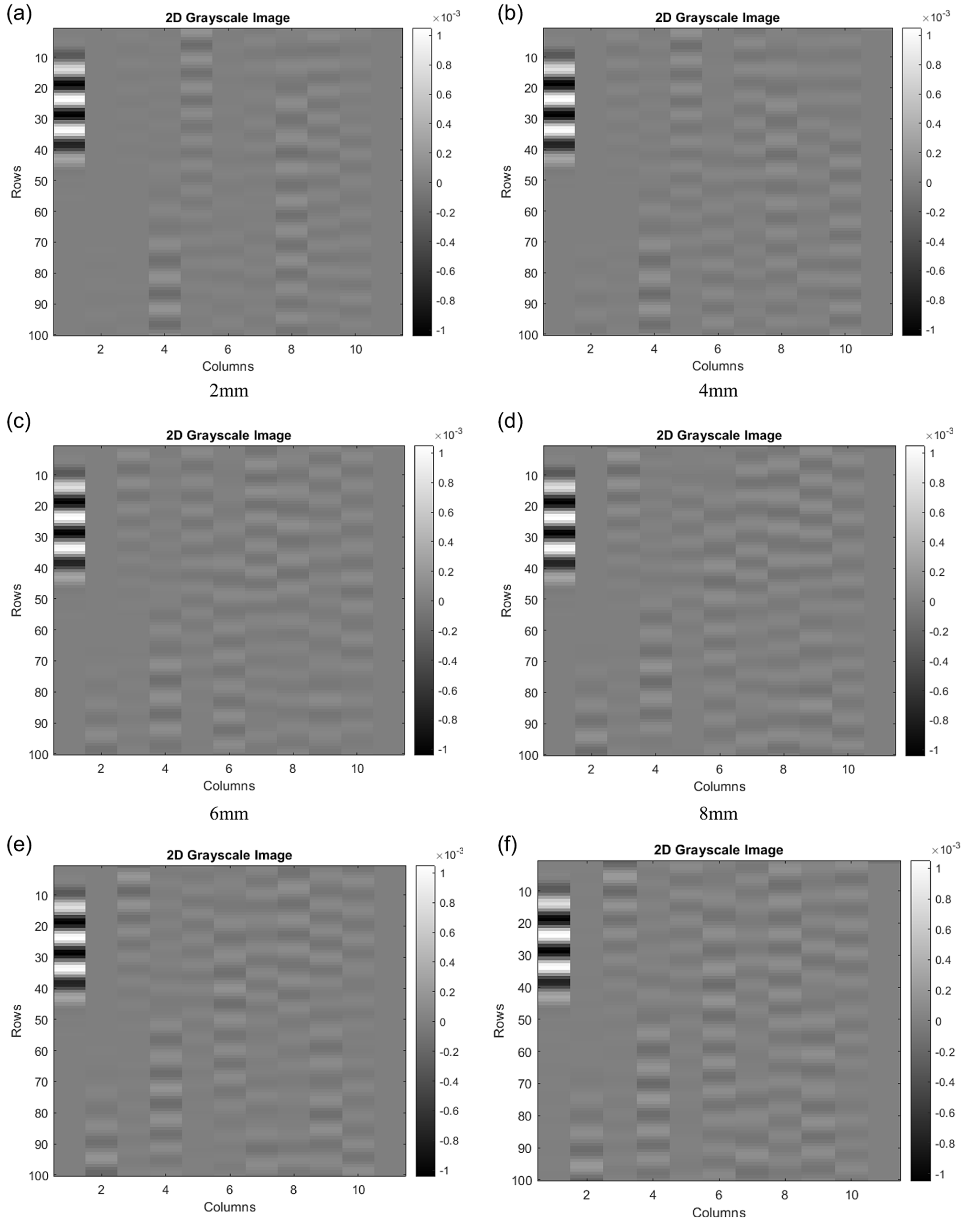
Figures 10 and 11 show the training progress of SVM and CNN-LSTM models for both the training and validation sets over 450 epochs with different kinds of input. The term "Classification accuracy at epoch 0" in Table 3 refers to the accuracy achieved by the machine learning models (SVM and CNN-LSTM) at the initial epoch of training. In the context of DL, an epoch is one complete pass through the entire training dataset during the model training process. The classification accuracy at epoch 0 provides insight into how well the models perform right at the beginning of

Table 3
The classification accuracy for SVM and CNN-LSTM models with signal series as input

Machine learning models	Input	Classification accuracy at epoch 0	Output accuracy
SVM	Signals of time and frequency feature series	60%	100%
CNN-LSTM		70%	100%
SVM	Image of time and frequency feature series	55%	90%
CNN-LSTM		67%	100%

the training phase before any iterations or adjustments. It serves as a baseline measure of the models' initial ability to correctly classify the input data into their respective classes. The values presented in this column indicate the percentage of accurate classifications achieved by each model at the start of the training process, specifically for the given input types (signal series with time and frequency features or images of time and frequency feature series). For the input of signals of time and frequency feature series, the CNN-LSTM model outperforms the SVM model in terms of classification accuracy at the very beginning. For instance, when epoch is at 0, the CNN-LSTM model's accuracy on both the training set and the test set start from 70%, while the accuracy of SVM model is 60%. The result revealed that the CNN-LSTM hybrid model has better

Figure 7
Images of oval-shaped damage with different damage sizes



2mm

4mm

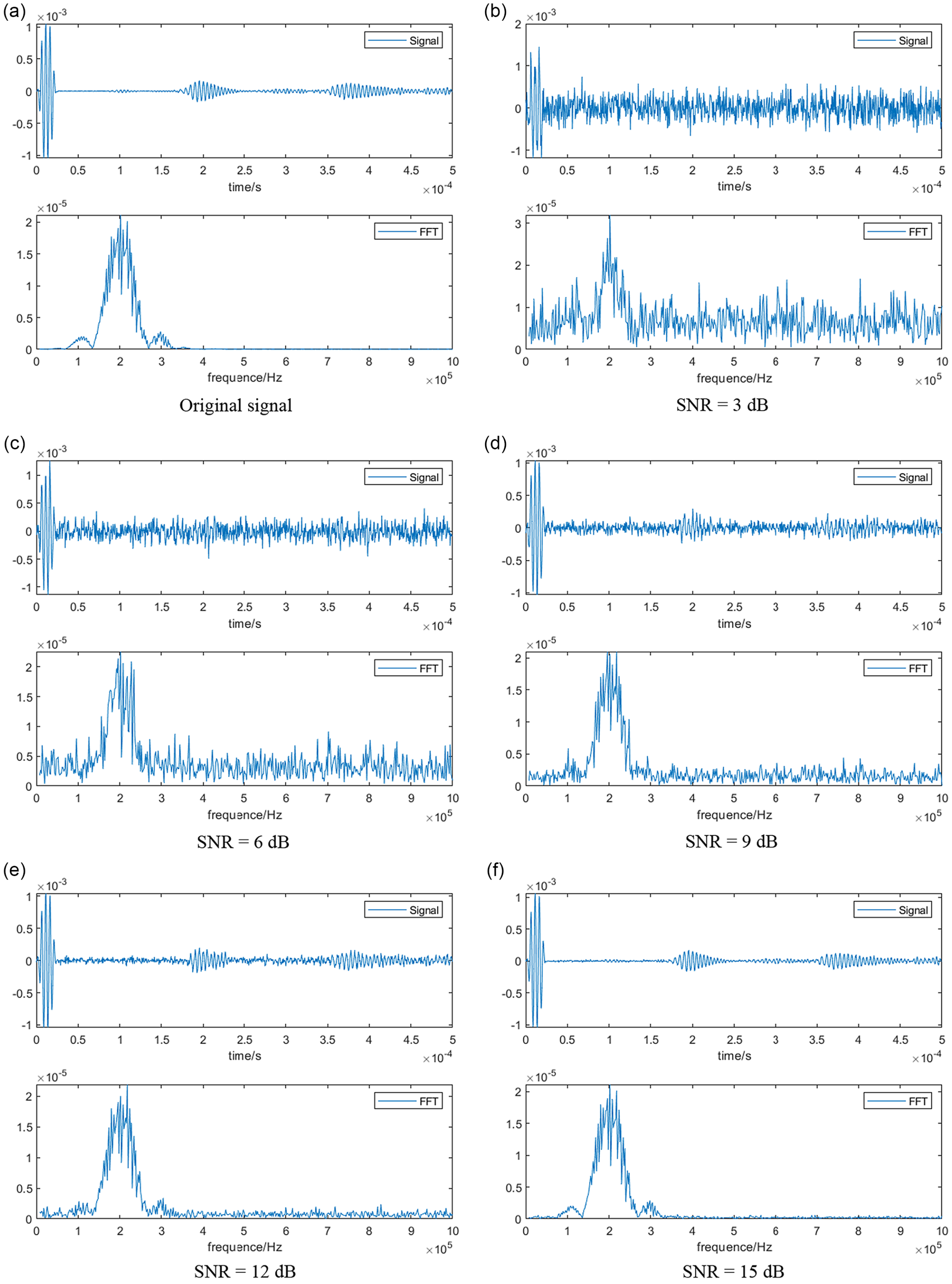
6mm

8mm

10mm

12mm

Figure 8
The signals of 2-mm oval-shaped damage on different noise levels



Original signal

SNR = 3 dB

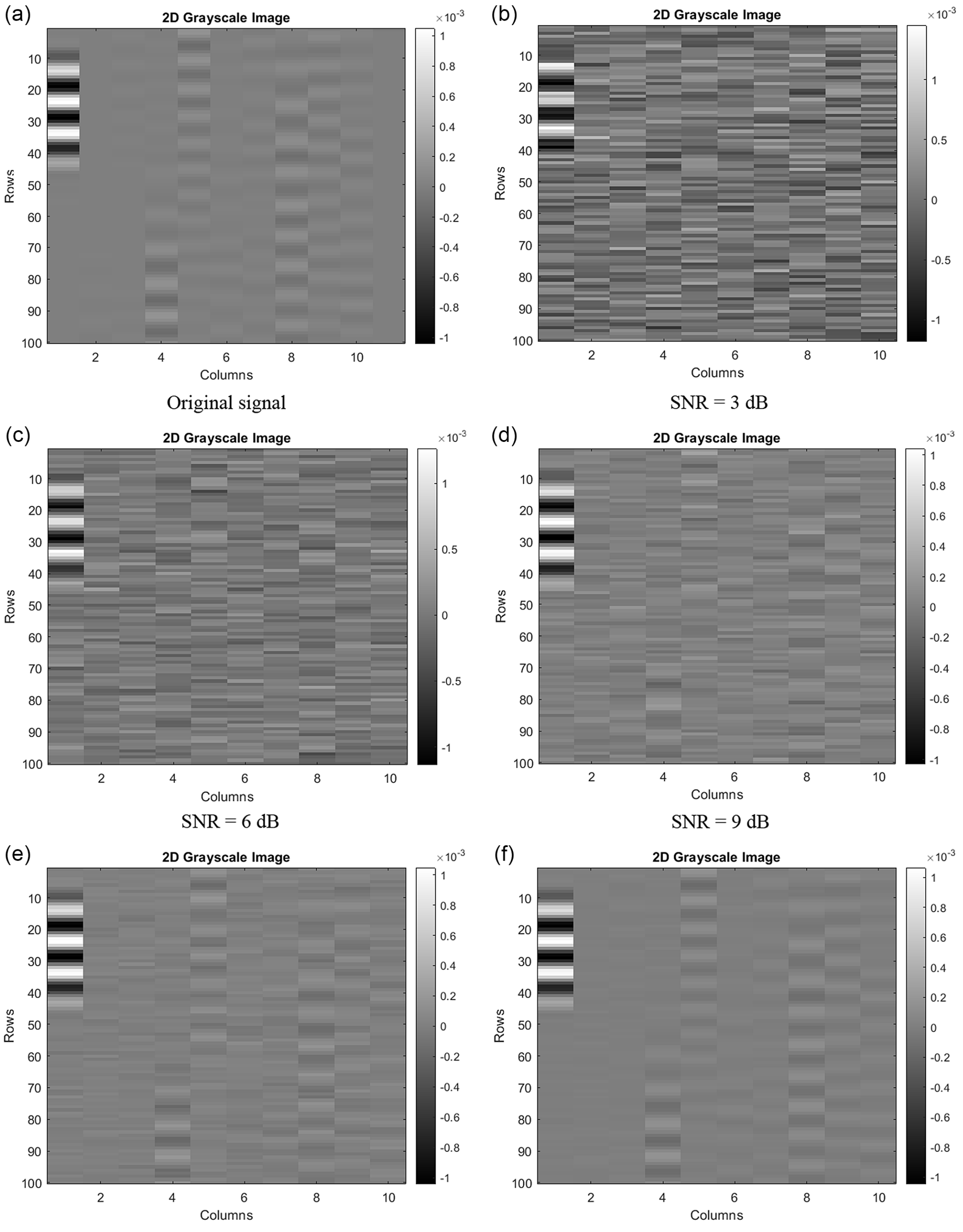
SNR = 6 dB

SNR = 9 dB

SNR = 12 dB

SNR = 15 dB

Figure 9
The images of 2-mm oval-shaped damage on different noise levels



Original signal

SNR = 3 dB

SNR = 6 dB

SNR = 9 dB

SNR = 12 dB

SNR = 15 dB

performance than SVM model in the beginning of signal and image classification tasks.

The accuracies of the training set and test set of SVM and CNN-LSTM models exhibit a growing trend as the number of epochs grows. Furthermore, it is evident that for SVM and CNN-LSTM models, the training accuracies are substantially greater than the validation accuracies. As shown in Figures 10 and 11, the accuracy of model training and validation of SVM and CNN-LSTM models are at their maximum values when the epoch is over 400. The classification accuracies for both the training set and the test set are stable at or near their maximum values at the same time, indicating that the model can adapt to the training set.

The reason why of the difference among SVM and CNN-LSTM models is as follows. SVM is a traditional classification model, and it is invented and is good at dealing with 1D time-series signal for this study, while SVM's linear decision boundaries have limited performance on large-scale data and could not be productively used to capture intricate features in image analysis. As a result, the application of SVM model on signal processing can realize high classification both at the beginning and at the end of model

Figure 10
Accuracy of training and validation for SVM and CNN-LSTM models with signal series as input

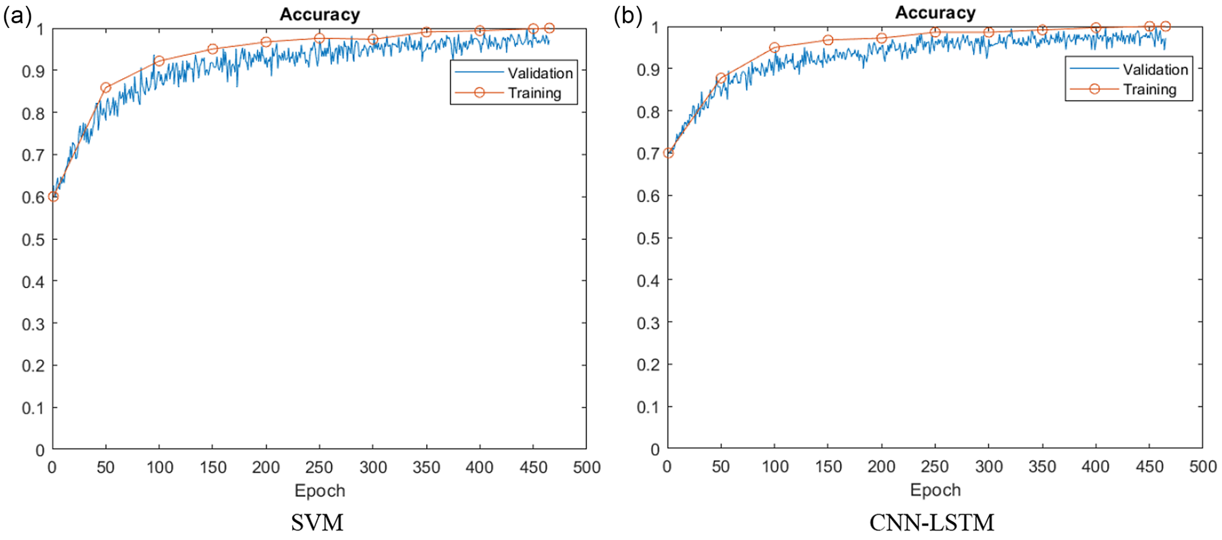
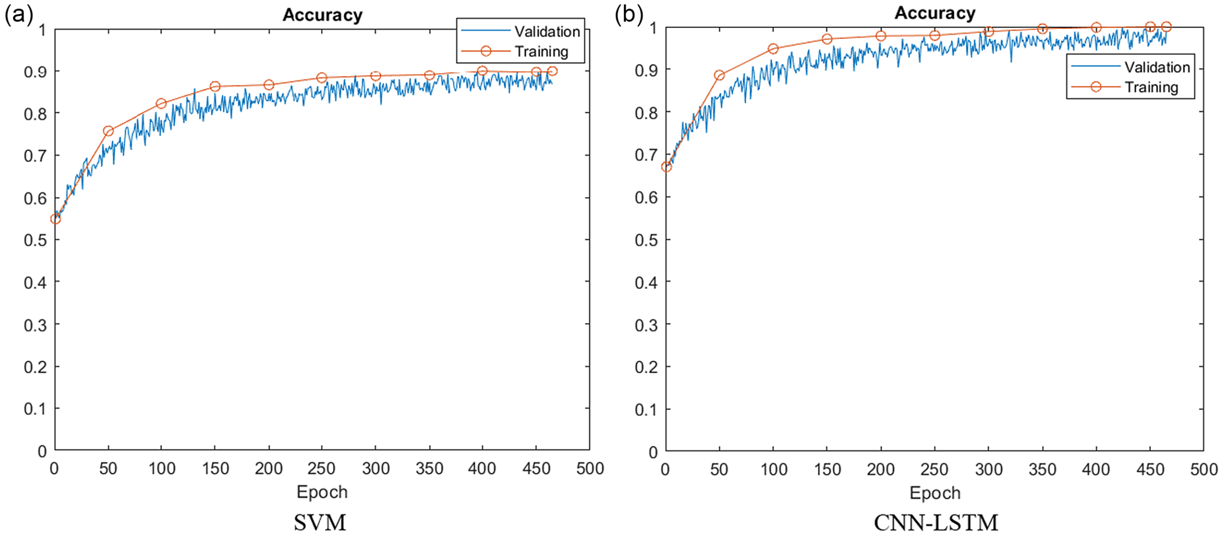


Figure 11
Accuracy of training and validation for SVM and CNN-LSTM models with images as input

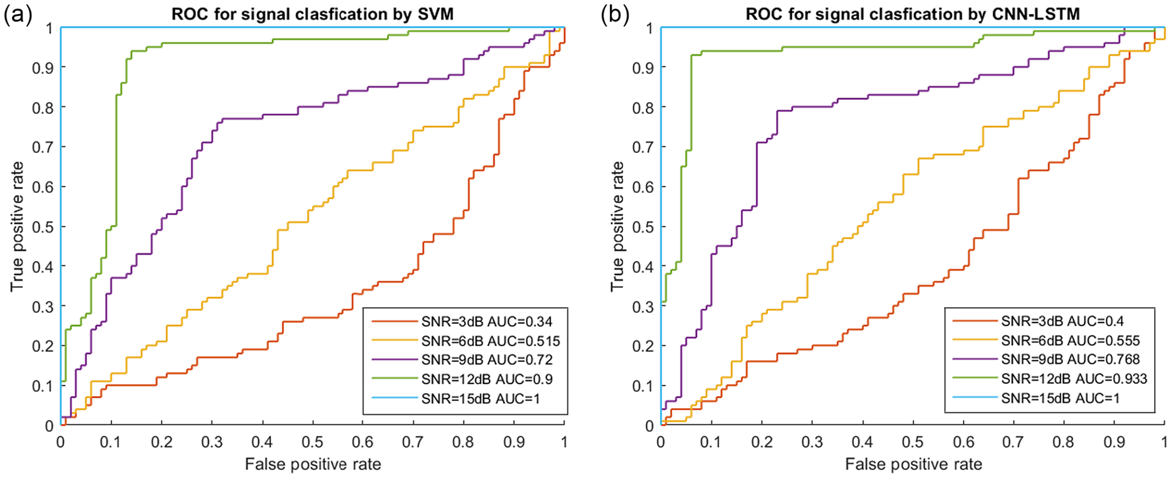


training. For CNN-LSTM hybrid model, it combines three important layers together, including the convolutional layer, the average pooling layer, and the LSTM layer, which decides the high performance of CNN-LSTM model. The convolutional layer has the capability to capture detailed features of images by sliding window methods and the average pooling layer can decrease the data dimensionality and increase the training efficiency of models, while the LSTM layer can capture temporal information and is good at processing time-series signal. As a result, the CNN-LSTM hybrid model outperforms the SVM model in signal and image classification tasks.

4.2. The classification accuracy of SVM and CNN-LSTM models with noised signals as input (Case 1)

To evaluate the robustness of model training under noise interference, two models of SVM and CNN-LSTM were trained with and without noise interference. Time and frequency feature series of original signals and time and frequency feature series of noised signals were used as training inputs. Classification accuracies, AUC values, and ROC curves were used as evaluation indicators of training performance of SVM and CNN-LSTM models. Table 4, Table 5, and Figure 12 show and compare the training results of evaluation indicators. Clearly, the classification accuracies and AUC values of SVM and CNN-LSTM models showed an upward trend as the noise levels of signals decreased. For instance, at a noise level of 15 dB, the classification accuracies of SVM and CNN-LSTM were all 100% and AUC values of two models were all 1.000, which had the same values as the models with the input of original signals. The results have also been demonstrated in Figure 8, where noised signals with 15 dB were almost the same as the original signals. It proved that the classification accuracies and AUC values of noised signals with 15 dB should be the highest and should be the same as the performance of

Figure 12
ROC curve for SVM and CNN-LSTM models on different noise levels



SVM

CNN-LSTM

Table 4
The classification accuracy of SVM and CNN-LSTM models on different SNR

		Accuracy	
Input	SNR (dB)	SVM	CNN-LSTM
Time and frequency feature series (original signal)	NAN	100.0%	100.0%
Time and frequency feature series (noised signals)	3	34.0%	40.2%
	6	50.5%	55.0%
	9	71.8%	76.5%
	12	89.6%	93.0%
	15	100.0%	100.0%

Table 5
The AUC values of SVM and CNN-LSTM models on different noise levels

		AUC	
Input	SNR (dB)	SVM	CNN-LSTM
Time and frequency feature series (original signal)	NAN	1.000	1.000
Time and frequency feature series (noised signals)	3	0.340	0.400
	6	0.515	0.555
	9	0.720	0.768
	12	0.900	0.933
	15	1.000	1.000

models with original signals. While when the noise level was 3 dB, 6 dB, and 9 dB, the classification accuracies of SVM and CNN-LSTM were almost all below 75% and the AUC values of two models were almost all below 0.750. It has been verified in Fan et al. [44] that it is unacceptable when the accuracies were less than 75% and the AUC values were less than 0.750. As a result, it can be concluded that low levels of noise interference could improve the performance of machine learning models.

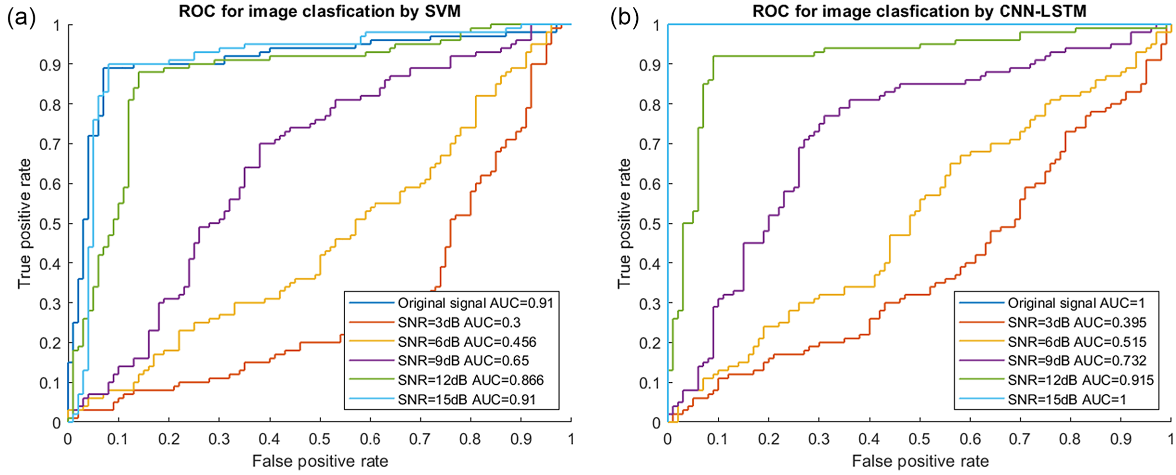
In addition, the DL models of CNN-LSTM model had better performance than the shallow learning model of SVM under high noise interference. For instance, at a noise level of 3 dB, the accuracy and AUC value of CNN-LSTM model increased by 18% than that of SVM model. But the training difference narrowed as the noise levels decreased. For instance, at a noise level of 12 dB, the accuracy and the AUC value of CNN-LSTM model just increased by 4% than that of SVM model. The results revealed that DL models have much more advantages in dealing with noised signals than shallow learning models. It gives us guidance for model choosing for noise signal classification in the future.

4.2.1. The classification accuracy of SVM and CNN-LSTM models with noised images as input (Case 1)

To evaluate the effectiveness of transforming signals to images and noise interference, SVM and CNN-LSTM models were trained with and without noise signals. Tables 6, 7 and Figure 13 show and compare the training results of accuracies, AUC values, and ROC curves. Firstly, the training performance of SVM and CNN-LSTM models also showed an increasing trend as the noise levels reduced. For instance, when the noise level reached 15 dB, the accuracy and AUC value of the CNN-LSTM model improved by nearly 156%, compared to when the noise level was 3 dB. Similarly, the accuracy and AUC value of the SVM model increased by nearly 203% when the noise level was raised to 15 dB, in comparison to when it was at 3 dB. The results revealed that noise in image construction could contaminate important information and features in image processing.

In addition, the DL model of CNN-LSTM outperformed the shallow learning model of SVM in image classification tasks. For instance, when images without noise interference and images with 15 dB levels of noise interference were input into SVM and CNN-LSTM models, the DL model of CNN-LSTM could achieve 100% accuracy and the AUC values were 1.000, while the shallow learning model of SVM just achieved 90% accuracy and the AUC value was 0.910. The results can be proved from Figure 9 that the original images and noised images with 15 dB noise levels had the same image features. The results also

Figure 13
ROC curve for SVM and CNN-LSTM models on different noise levels



SVM

CNN-LSTM

Table 6
The classification accuracy of SVM and CNN-LSTM models on different SNR

		Accuracy	
Input	SNR (dB)	SVM	CNN-LSTM
Time and frequency feature series (original image)	NAN	90.0%	100.0%
Time and frequency feature series (noised image)	3	30.0%	39.0%
	6	45.5%	52.0%
	9	64.8%	73.4%
	12	85.6%	91.0%
	15	90.0%	100.0%

Table 7
The AUC values of SVM and CNN-LSTM models on different noise levels

		AUC	
Input	SNR (dB)	SVM	CNN-LSTM
Time and frequency feature series (original image)	NAN	0.910	1.000
Time and frequency feature series (noised image)	3	0.300	0.395
	6	0.456	0.515
	9	0.650	0.732
	12	0.866	0.915
	15	0.910	1.000

demonstrated the advantages of the complex structures of DL models in image processing tasks in comparison to shallow learning model with linear algorithm.

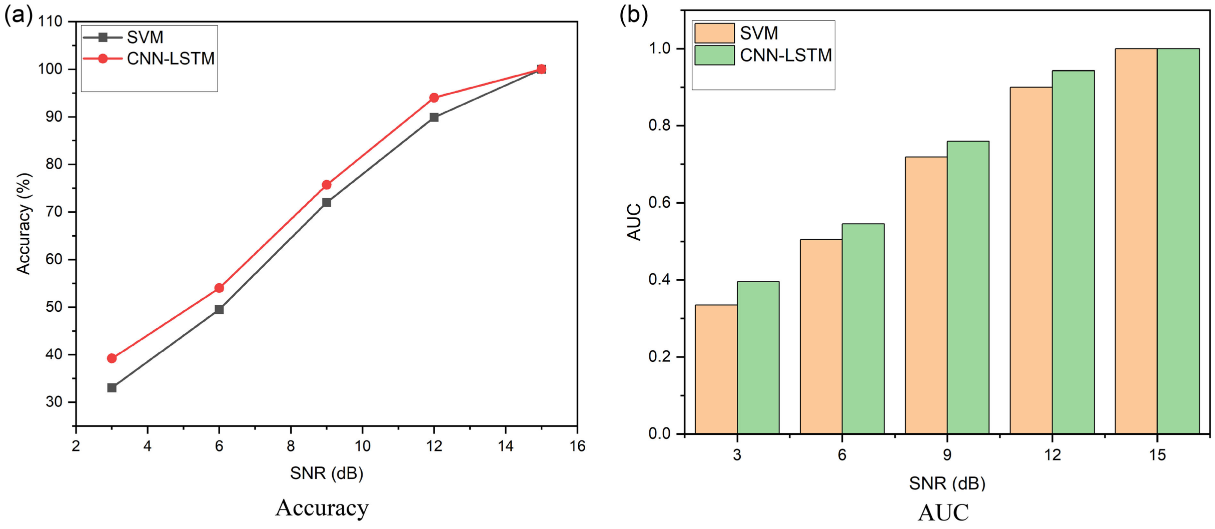
The study in Case 1 for signal classification and image classification tasks revealed that both signal classification and image classification can be effective methods for damage detection, but for different classification tasks, the choosing of appropriate training model is a key step because different models have different network structures, and they are suitable for different classification tasks.

5. Further Discussion of Pipelines with Different Kinds of Defects (Case 2)

5.1. The classification accuracy of SVM and CNN-LSTM models with signals and images as input under noise interference (Case 2)

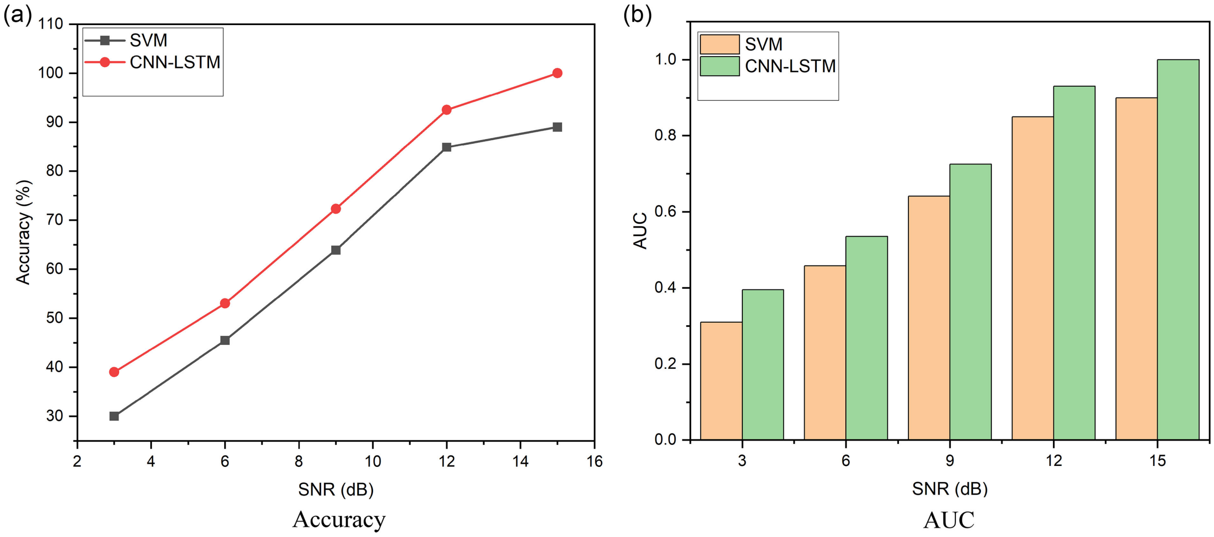
To demonstrate the robustness of the developed models, SVM and CNN-LSTM models were trained with the data from Case 2. Accuracy and AUC values were calculated to evaluate the performance of training models. Figures 14 and 15 show and compare the evaluation indicators. It is obvious to see that the values of accuracy and AUC for both signal classification and image classification manifested a rising tendency. Additionally, it demonstrated a faster growth for both signal and image classification tasks under high noise interference and a slower increase under low noise levels. For example, for signal classification tasks, there had around 51% and 38% increase for both accuracies and AUC value of SVM and CNN-LSTM models respectively from 3 dB to 6 dB noise interference, while there only had around 11% and 6% improve for both accuracies and AUC value of SVM and CNN-LSTM models respectively from 12 dB to 15 dB noise interference. Similarly, for image classification tasks, the accuracy and AUC values of both SVM and CNN-LSTM models increased by approximately 52% and 36%, respectively, from 3 dB to 6 dB of noise interference, whereas from 12 dB to 15 dB of noise interference, these improvements were only about 6% and 8%. Shang et al. [16] conducted a study using the CNN-LSTM model on ultrasonic signals from field collection, achieving a successful application with an accuracy of 93.8% at 15 dB. The results demonstrated that the reduction of noise interference could improve the training performance of machine learning and DL models to some extent and could narrow the different of different network structures of DL models and shallow learning models.

Figure 14
The classification accuracy and AUC of SVM and CNN-LSTM models with signals on different noise levels



Accuracy AUC

Figure 15
The classification accuracy and AUC of SVM and CNN-LSTM models with images on different noise levels



Accuracy AUC

In addition, the results further revealed that SVM linear model is good at the classification of 1D time series, while it shows poor performance in image processing tasks. For instance, when the noise level is reduced to 15 dB, the SVM could achieve 100% accuracy and the AUC value is 1.000 for signal classification, while for image classification tasks, only 90% performance was achieved and the AUC value is 0.900. And the CNN-LSTM model could be used to do signal and image processing tasks under light noise interference. For instance, the CNN-LSTM model could achieve 100% performance at 0 dB and 15 dB noise interference for both signal and image processing tasks. The results revealed that both shallow learning models and DL models are suitable for signal processing under light noise interference, but for heavy noise interference, DL models can better capture signal features than shallow learning models. And DL models are better at image processing than shallow learning models for both light and heavy noise interference. It further revealed that the complex network structure of DL models can better deal with 2D data and even high dimensional data than shallow learning models.

6. Conclusions

This study demonstrated the effectiveness of the developed SVM and CNN-LSTM models on damage detection on metallic plates. Different damage statuses were designed based on COMSOL models to produce different cases of training data. Time and frequency feature series and the images transformed time series were extracted and input into models to SVM and CNN-LSTM models. Additionally, various noise interference levels were employed to gauge how robust the developed models were. It is possible to draw the following conclusions.

1) Feature extraction in time and frequency domain and signal transformation to images are two effective signal preprocessing methods to improve the damage detection deficiency.
2) Both time and frequency feature series and the images transformed time series can be used as effective input to train SVM and CNN-LSTM models.
3) The DL model of CNN-LSTM has better performance in signal and image processing tasks than the shallow learning model of SVM under noise interference.
4) The effectiveness and robustness of SVM and CNN-LSTM models were identified by different cases (Case 1 and Case 2) and noise interference. The results demonstrated that the robustness of the DL model of CNN-LSTM is much higher than that of shallow learning mode of SVM.

Acknowledgments

The authors gratefully acknowledge the financial support provided by USDOTs (DTPH5616HCAP03, 693JK318500010CAAP, 693JK31850009CAAP, 693JK32110003POTA, 693JK32250007CAAP). The results, discussion, and opinions reflected in this paper are those of the authors only and do not necessarily represent those of the sponsors.

Funding Support

This research was funded by USDOTs (DTPH5616HCAP03, 693JK318500010CAAP, 693JK31850009CAAP, 693JK32110003POTA, 693JK32250007CAAP).

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data available on request from the corresponding author upon reasonable request.

Author Contribution Statement

Li Shang: Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft. **Zi Zhang:** Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft. **Fujian Tang:** Conceptualization, Writing – review & editing, Visualization. **Qi Cao:** Conceptualization, Writing – review & editing, Visualization. **Hong Pan:** Conceptualization, Writing – review & editing, Visualization, Supervision, Project administration. **Zhibin Lin:** Conceptualization, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition.

References

[1] Mishra, M., Lourenço, P. B., & Ramana, G. V. (2022). Structural health monitoring of civil engineering structures by using the internet of things: A review. *Journal of Building Engineering*, *48*, 103954. https://doi.org/10.1016/J.JOBE.2021.103954

[2] Sun, M., Staszewski, W. J., & Swamy, R. N. (2010). Smart sensing technologies for structural health monitoring of civil engineering structures. *Advances in Civil Engineering*, *2010*(1), 724962. https://doi.org/10.1155/2010/724962

[3] Giurgiutiu, V. (2008). *Structural health monitoring: With piezoelectric wafer active sensors*. Netherlands: Elsevier.

[4] Mitra, M., & Gopalakrishnan, S. (2016). Guided wave based structural health monitoring: A review. *Smart Materials and Structures*, *25*(5), 053001. https://doi.org/10.1088/0964-1726/25/5/053001

[5] Feng, D., & Feng, M. Q. (2016). Vision-based multipoint displacement measurement for structural health monitoring. *Structural Control and Health Monitoring*, *23*(5), 876–890. https://doi.org/10.1002/STC.1819

[6] Doebling, S. W., Farrar, C. R., & Prime, M. B. (1998). A summary review of vibration-based damage identification methods. *Shock and Vibration Digest*, *30*(2), 91–105.

[7] Kong, X., Cai, C. S., & Hu, J. (2017). The state-of-the-art on framework of vibration-based structural damage identification for decision making. *Applied Sciences*, *7*(5), 497. https://doi.org/10.3390/APP7050497

[8] Ahn, B., Kim, J., & Choi, B. (2019). Artificial intelligence-based machine learning considering flow and temperature of the pipeline for leak early detection using acoustic emission. *Engineering Fracture Mechanics*, *210*, 381–392. https://doi.org/10.1016/J.ENGFRACMECH.2018.03.010

[9] Carvalho, A. A., Rebello, J. M. A., Sagrilo, L. V. S., Camerini, C. S., & Miranda, I. V. J. (2006). MFL signals and artificial neural networks applied to detection and classification of pipe weld defects. *NDT & E International*, *39*(8), 661–667. https://doi.org/10.1016/J.NDTEINT.2006.04.003

[10] Kim, J. W., & Park, S. (2018). Magnetic flux leakage sensing and artificial neural network pattern recognition-based automated damage detection and quantification for wire rope non-destructive evaluation. *Sensors*, *18*(1), 109. https://doi.org/10.3390/S18010109

[11] Abdel-Hamid, O., Mohamed, A. R., Jiang, H., & Penn, G. (2012). Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing,* 4277–4280. https://doi.org/10.1109/ICASSP.2012.6288864

[12] Mohamed, A. R., Dahl, G. E., & Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(1), 14–22. https://doi.org/10.1109/TASL.2011.2109382

[13] Nagrani, A., Chung, J., & Zisserman, A. (2017). VoxCeleb: A large-scale speaker identification dataset. In *Interspeech 2017*, 2616–2620. https://doi.org/10.21437/Interspeech.2017-950

[14] Variani, E., Lei, X., McDermott, E., Moreno, I. L., & Gonzalez-Dominguez, J. (2014). Deep neural networks for small footprint text-dependent speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing,* 4052–4056. https://doi.org/10.1109/ICASSP.2014.6854363

[15] Zhang, C., & Koishida, K. (2017). End-to-end text-independent speaker verification with flexibility in utterance duration. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop,* 584–590. https://doi.org/10.1109/ASRU.2017.8268989

[16] Shang, L., Zhang, Z., Tang, F., Cao, Q., Pan, H., & Lin, Z. (2023). CNN-LSTM hybrid model to promote signal processing of ultrasonic guided lamb waves for damage detection in metallic pipelines. *Sensors*, *23*(16), 7059. https://doi.org/10.3390/S23167059

[17] Zhang, Z., Pan, H., Wang, X., & Lin, Z. (2020). Machine learning-enriched lamb wave approaches for automated damage detection. *Sensors*, *20*(6), 1790. https://doi.org/10.3390/S20061790

[18] Lin, Z., Pan, H., Wang, X., & Li, M. (2018). Data-driven structural diagnosis and conditional assessment: From shallow to deep learning. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, *10598*, 1059814.

[19] Pan, H., Azimi, M., Yan, F., & Lin, Z. (2018). Time-frequency-based data-driven structural diagnosis and damage detection for cable-stayed bridges. *Journal of Bridge Engineering*, *23*(6), 04018033. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001199

[20] Gui, G., Pan, H., Lin, Z., Li, Y., & Yuan, Z. (2017). Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection. *KSCE Journal of Civil Engineering*, *21*(2), 523–534. https://doi.org/10.1007/s12205-017-1518-5

[21] Zhang, Z., Pan, H., Wang, X., & Lin, Z. (2022). Deep learning empowered structural health monitoring and damage diagnostics for structures with weldment via decoding ultrasonic guided wave. *Sensors*, *22*(14), 5390. https://doi.org/10.3390/s22145390

[22] Yang, J., He, J., Guan, X., Wang, D., Chen, H., Zhang, W., & Liu, Y. (2016). A probabilistic crack size quantification method using in-situ Lamb wave test and Bayesian updating. *Mechanical Systems and Signal Processing*, *78*, 118–133. https://doi.org/10.1016/j.ymssp.2015.06.017

[23] Chen, H., Liu, Z., Gong, Y., Wu, B., & He, C. (2021). Evolutionary strategy-based location algorithm for high-resolution Lamb wave defect detection with sparse array. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, *68*(6), 2277–2293. https://doi.org/10.1109/TUFFC.2021.3060094

[24] Das, S., Chattopadhyay, A., & Srivastava, A. N. (2010). Classifying induced damage in composite plates using one-class support vector machines. *AIAA Journal*, *48*(4), 705–718. https://doi.org/10.2514/1.37282

[25] Legendre, S., Massicotte, D., Goyette, J., & Bose, T. K. (2001). Neural classification of Lamb wave ultrasonic weld testing signals using wavelet coefficients. *IEEE Transactions on Instrumentation and Measurement*, *50*(3), 672–678. https://doi.org/10.1109/19.930439

[26] Su, Z., & Ye, L. (2004). Lamb wave-based quantitative identification of delamination in CF/EP composite structures using artificial neural algorithm. *Composite Structures*, *66*(1–4), 627–637. https://doi.org/10.1016/j.compstruct.2004.05.011

[27] Veiga, J. L. B. C., de Carvalho, A. A., da Silva, I. C., & Rebello, J. M. A. (2005). The use of artificial neural network in the classification of pulse-echo and TOFD ultra-sonic signals. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *27*(4), 394–398. https://doi.org/10.1590/S1678-58782005000400007

[28] Vapnik, V. N. (2000). *The nature of statistical learning theory*. USA: Springer Science & Business Media.

[29] Dibike, Y. B., Velickov, S., Solomatine, D., & Abbott, M. B. (2001). Model induction with support vector machines: Introduction and applications. *Journal of Computing in Civil Engineering*, *15*(3), 208–216. https://doi.org/10.1061/(ASCE)0887-3801(2001)15:3(208)

[30] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121–167. https://doi.org/10.1023/A:1009715923555

[31] Burbidge, R., & Buxton, B. (2001). An introduction to support vector machines for data mining. *Keynote Papers, Young OR12*, 2–14.

[32] Xu, J., Sun, X., Zhang, Z., Zhao, G., & Lin, J. (2019). Understanding and improving layer normalization. In *33rd Conference on Neural Information Processing Systems*, 1–11.

[33] Andhale, Y. S., Masurkar, F., & Yelve, N. (2019). Localization of damages in plain and riveted aluminium specimens using lamb waves. *International Journal of Acoustics & Vibration*, *24*(1), 150–165.

[34] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

[35] Shang, L., Zhang, Z., Tang, F., Cao, Q., Yodo, N., Pan, H., & Lin, Z. (2023). Deep learning enriched automation in damage detection for sustainable operation in pipelines with welding defects under varying embedment conditions. *Computation*, *11*(11), 218. https://doi.org/10.3390/computation11110218

[36] Zhang, W., Zhou, H., Bao, X., & Cui, H. (2023). Outlet water temperature prediction of energy pile based on spatial-temporal feature extraction through CNN–LSTM hybrid model. *Energy*, *264*, 126190. https://doi.org/10.1016/j.energy.2022.126190

[37] Rani, C. J., & Devarakonda, N. (2022). An effectual classical dance pose estimation and classification system employing convolution neural network–long shortterm memory (CNN-LSTM) network for video sequences. *Microprocessors and Microsystems*, *95*, 104651. https://doi.org/10.1016/J.MICPRO.2022.104651

[38] Mellit, A., Pavan, A. M., & Lughi, V. (2021). Deep learning neural networks for short-term photovoltaic power forecasting. *Renewable Energy*, *172*, 276–288. https://doi.org/10.1016/j.renene.2021.02.166

[39] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

[40] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[41] Chen, C. (2014). *Reliability assessment method for space rolling bearing based on condition vibration feature*. Chongqing University.

[42] Huang, X., Zhou, L., & Zhang, D. (2022). Jī yú tè zhēng róng hé hé hùn lèi zēng qiáng de shēn dù xué xí gŭn dòng zhóu chéng gù zhàng zhěn duàn [Deep learning for rolling bearing fault diagnosis based on feature fusion and hybrid enhancement]. *Computer Systems & Applications*, *31*(8), 345–353. https://doi.org/10.15888/j.cnki.csa.008630

[43] Sun, Z., Liu, K., Jiang, J., Xu, T., Wang, S., Guo, H., . . . , & Liu, T. (2021). Optical fiber distributed vibration sensing using grayscale image and multi-class deep learning framework for multi-event recognition. *IEEE Sensors Journal*, *21*(17), 19112–19120. https://doi.org/10.1109/JSEN.2021.3089004

[44] Fan, J., Upadhye, S., & Worster, A. (2006). Understanding receiver operating characteristic (ROC) curves. *Canadian Journal of Emergency Medicine*, *8*(1), 19–20. https://doi.org/10.1017/S1481803500013336

How to Cite: Shang, L., Zhang, Z., Tang, F., Cao, Q., Pan, H., & Lin, Z. (2025). Signal Process of Ultrasonic Guided Wave for Damage Detection of Localized Defects in Plates: From Shallow Learning to Deep Learning. *Journal of Data Science and Intelligent Systems*, *3*(2), 149–164. https://doi.org/10.47852/bonviewJDSIS32021771