

Hybrid Wav2Vec2 / Multi-Head Deep Feedforward System for Health Monitoring of Oil and Gas Pipelines

Mohammad Al Qaderi

`mohammadk_al@hu.edu.jo`

Hashemite University

Ahmad J Al-Mahasneh

Philadelphia University

Mohammed Abu Mallouh

Hashemite University

Alaa Al Ghazo

Hashemite University

Mohamed Almahakeri

University of Arkansas at Fayetteville

Research Article

Keywords: Pipeline Health Monitoring, Crack Recognition, FEM (Finite Element Method), Wav2vec2 Neural Network, Machine Learning Algorithms

Posted Date: September 29th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10950002/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Hybrid Wav2Vec2 / Multi-Head Deep Feedforward System for Health Monitoring of Oil and Gas Pipelines

Mohammad K. Al Qaderi ^{1*}, Ahmad J. Al-Mahasneh ², Mohammed A. Abu Mallouh ¹, Alaa T. Al Ghazo ¹, and Mohamed Almahakeri ³

Abstract – Oil and gas pipelines are essential infrastructure systems necessitating constant surveillance to avert safety risks and financial losses from undetected flaws. This study introduces an innovative Structural Health Monitoring (SHM) framework that combines the Wav2Vec2 neural network with a multi-head deep feedforward (MHDF) neural network to identify and analyze pipeline fractures utilizing ultrasonic guided wave (UGW) signals. The methodology utilizes transfer learning and automated feature extraction to overcome the constraints of conventional hand-crafted features. Data augmentation methods are employed to improve model resilience. We test the suggested system on a dataset that has never been seen before and compare it to well-known machine learning models, such as Deep Neural Networks (DNN) and Bi-directional Long Short-Term Memory (Bi-LSTM) networks. The Wav2Vec2/MHDF model demonstrably surpasses these baselines in prediction accuracy and error minimization, establishing it as a viable solution for real-time pipeline health monitoring applications.

Keywords: *Pipeline Health Monitoring; Crack Recognition; FEM (Finite Element Method); Wav2vec2 Neural Network; Machine Learning Algorithms*

I. Introduction

Oil and natural gas, essential energy sources for modern society, heavily depend on pipelines for their transmission, attracting significant attention. However, these pipes provide substantial hazards due to their combustible and explosive characteristics [1]. Pipelines with high pressure can experience defects that cause rapid leakage and diffusion. This can lead to explosions or fires when the leaked substance reaches its lower explosive limit [2], [3]. It is crucial to identify and diagnose these problems at an early stage. The global pipeline network spans tens of millions of miles, with over 5 million kilometers located in the United States alone [4]. In addition to the challenges posed by aging infrastructure, external factors such as earthquakes increase the dangers [5]. Between 2001 and 2020, the United States saw more than 12,500 pipeline accidents, which led to financial damages exceeding \$10 billion [6]. The severity of environmental and public health concerns is evident, as exemplified by incidents such as the Kalamazoo River leak in 2010 [7]. Ongoing surveillance and assessment are crucial to prevent such disasters, even in recently constructed pipelines such as the Keystone pipeline in North Dakota in 2019 [8].

To detect incidents such as leaks and spillage and ensure the safe operation of pipeline systems, Structural Health Monitoring (SHM) techniques have been developed and

implemented. SHM is composed of two primary categories: passive and active SHM [9]. Passive SHM is a methodology that utilizes operational metrics to determine the condition of a structure by analyzing gathered parameters. An example of passive SHM implies the deployment of trained personnel to different positions along the conduit; however, this methodology is progressively being replaced by aerial, land, and underwater robotic devices [10]. Hydrostatic Pressure Testing, also known as Hydrotest, is another conventional passive SHM method that is both costly and difficult to implement. Each of these passive inspection-based approaches possesses unique pros and cons. One notable drawback is the absence of continuity, as inspections can occur at extended intervals, thereby exposing the pipe to the risk of undetected accumulation of damage.

In contrast, active SHM aims to develop a system that relies on transducers that are permanently attached to deliver structural health data in real-time. SHM frequently employs optical fibers [11], ultrasonic acoustic flow meters [12]; and Ultrasound Guided Waves (UGWs) technology [13] as transducer technologies.

Diverse active SHM strategies for continuous monitoring of leakage and damage in oil and gas pipelines have been developed. Anomaly localization was facilitated in [14] by the combination of a microwave signal emitted from the pipe surface and a Fast Fourier Transform-based technique. A loop-integrated Mach Zehnder Interferometer optical fiber sensor was utilized in [15] to monitor and detect leaks in pipelines. Deformation monitoring for pipelines was first proposed by [16] using an inverse finite element method based on the inverse beam element. The system demonstrated effective operation in both typical and freeze-thaw conditions. In [17], the deformation of a subterranean pipe was estimated using spatial deformation curve fitting. In contrast, a Bayesian fusion-based method was utilized in [18] to calculate the longitudinal and axial thermal stresses in a buried pipeline.

UGW stands out as one of the most utilized and promising technologies within active SHM. In a UGW-based SHM system, ultrasound waves are initially generated by an array of Piezoelectric transducers, typically Lead Zirconate Titanate (PZT) transducers. Subsequently, the waves propagate in the pipeline, guided by its geometric boundaries, thereby substantially magnifying the impacts of any geometric imperfections in the boundaries, cracks, breaks, holes, or corroded regions on the waves that reflect such defects. This change facilitates the identification and categorization of defects occurring within the pipeline. Appropriate signal processing schemes are utilized in the analysis of the characteristics of the reflected waves on the pipeline to determine the position and magnitude of structural damage. Multiple methodologies have been suggested by various studies for the purpose of damage identification. As an illustration, [19] utilized sparse estimation to accurately localize defects in UGWs via torsional and flexural propagation modes. UGWs were employed in another study [20] to algebraically reconstruct a pipe image with the intention of subsequently localizing defects. The utilization of shear horizontal guided waves to monitor crack growth in steel pipelines was described in [21]. Damage detection was achieved through the application of quasi-statistical approximation. In [22], it was demonstrated that the time-reversal method could accurately detect and localize damage in the axial and circumferential dimensions of a steel pipe. [23] employed the compressed guided wave inspection technique in conjunction with sparse vector to precisely localize anomalies within pipelines. In addition, the severity, position, and magnitude of corrosion were assessed in [24] by employing the Algebraic Reconstruction Technique in conjunction with a finite-difference method to solve the Eikonal equation.

Although the active SHM methods mentioned above have demonstrated efficacy, they frequently depend on fundamental techniques like regression models or model-driven approaches, which may be insufficient when faced with datasets that have a high dimension or when structural knowledge is limited. As an alternative approach to address these challenges, models based on Deep learning; these models operate without requiring prior knowledge of the structure. Deep learning has transformed SHM by introducing advanced methods for analyzing complex datasets generated from structural vibrations. Techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been widely applied, demonstrating a strong capacity to identify meaningful features directly from raw vibration data [25]. Recent reviews indicate a substantial increase in research focused on vibration-based SHM, with deep learning methods achieving performance levels comparable to traditional machine learning approaches [26]. The authors of [27] employed an enhanced CNN that included a visual transformation layer for the purpose of estimating defect width, length, and depth from MFL signals. This CNN outperformed alternative AI algorithms in terms of accuracy. A fast region CNN was suggested in a study by [28] as a means of identifying and detecting defects in sewer pipelines using CCTV camera images. Although it made a few misclassifications, the CNN achieved an overall excellent performance. Additionally, the authors [29] suggested a deep learning methodology that incorporates CCTV input in order to improve the identification and detection of defects in sewer pipes. The authors of [30] suggested a CNN model that was trained using scalogram images obtained from vibration data to detect leakage in underground PVC pipes.

The integration of transfer learning within deep learning frameworks has further enhanced model performance, particularly in cases where training data is limited [31]. Transfer learning has gained significant traction in the field of SHM, particularly in scenarios where labeled data is limited. This approach allows models trained on one dataset to be adapted for another, enhancing performance without extensive retraining. For instance, Hesser et al. demonstrated that CNNs combined with transfer learning can effectively identify acoustic emission sources in SHM applications, showcasing the potential for knowledge transfer across different structural conditions [32]. Furthermore, transfer learning CNN was implemented in [33] to efficiently identify and pinpoint pipeline leakage scenarios. The Wav2Vec2 model, originally designed for speech recognition, has shown promise in SHM by learning robust feature representations from raw time-series data, which can enhance the detection of anomalies in vibration signals [34]. The integration of Wav2Vec2 with transfer learning not only addresses data scarcity challenges but also provides a framework for developing adaptive SHM methodologies that can generalize across various structural conditions [35]. Furthermore, domain adaptation techniques, such as maximum mean discrepancy, have been effectively employed to align feature distributions between source and target domains, thereby improving model performance in SHM tasks [36]. Recent studies have also highlighted the effectiveness of transfer learning in improving the accuracy of damage detection models by utilizing knowledge from related tasks, thus demonstrating its applicability in SHM [37]. Additionally, the use of transfer learning in conjunction with deep learning architectures has been shown to yield promising results, particularly in the context of limited data environments [38]. Overall, the combination of transfer learning and Wav2Vec2 represents a significant advancement in the field, enabling more effective monitoring of structural integrity [39].

113 Data augmentation is a crucial strategy in developing robust SHM models, particularly for
114 time-series data where obtaining labeled samples can be challenging. Various
115 augmentation techniques have been explored to synthetically expand training datasets,
116 thereby improving model generalization. Traditional methods, such as time warping and
117 jittering, have been complemented by advanced techniques like Variational Autoencoders
118 (VAE) and Generative Adversarial Networks (GAN), which can generate realistic synthetic
119 data that mimics the underlying distribution of the original dataset [40]. The review of data
120 augmentation methods emphasizes the importance of these techniques in enhancing the
121 diversity of training samples, which is essential for training deep learning models effectively
122 [41]. Moreover, wavelet transformations have been utilized to convert 1D vibration signals
123 into 2D representations, enriching the feature space and providing deeper insights into
124 structural conditions [42], [43]. The evaluation of augmented data is also critical, as it
125 ensures that synthetic samples contribute positively to model training and do not introduce
126 noise or bias [44]. Recent studies have shown that data augmentation can significantly
127 improve the performance of deep learning models in SHM, particularly in scenarios with
128 limited data availability [45]. Additionally, the use of data augmentation techniques has been
129 linked to improved robustness and accuracy in damage detection tasks, further underscoring
130 their importance in SHM [40]. Overall, the literature underscores the significance of data
131 augmentation in SHM, as it enables the development of more accurate and reliable
132 predictive models [40]. Overall, the application of deep learning in SHM represents a
133 transformative shift, enabling more effective monitoring and maintenance of structural
134 integrity.

135 The literature review indicates that the integration of advanced architectures, such as pre-
136 trained deep neural networks that leverage transfer learning, remains underexplored in the
137 context of pipeline crack detection using UGWs, despite the increasing application of ML
138 techniques in SHM. The majority of current research relies on handcrafted features or
139 image-based inputs, which frequently necessitate temporary pipeline shutdowns for data
140 collection—an approach that is not always feasible in real-world scenarios. In contrast, this
141 study suggests a new SHM framework that employs a hybrid Wav2Vec2/Multi-Head Deep
142 Feedforward (MHDF) neural network architecture to facilitate real-time, non-invasive
143 pipeline integrity monitoring. The MHDF network has been constructed to accurately
144 predict three critical crack parameters: position, width, and depth, while the Wav2Vec2
145 component automatically extracts high-level features from UGW signals without the need
146 for manual feature engineering.

147 The proposed approach is compared to three widely used ML-based methods in order to
148 assess the system's effectiveness: a traditional Deep Neural Network (DNN) that employs
149 handcrafted features, a Bidirectional Long Short-Term Memory (Bi-LSTM) network, and a
150 PCA-enhanced Bi-LSTM model. These comparisons underscore the benefits of integrating
151 deep multi-output regression and transfer learning for SHM tasks in data-scarce conditions.

152 The subsequent sections of this paper are structured as follows. The pipeline and cracks
153 modeling, as well as the generation of UGW data with finite element simulations for 285
154 pipeline crack scenarios, are presented in Section 2. The architecture of the
155 Wav2Vec2/MHDF model and the data processing are both introduced in Section 3. The
156 data augmentation strategies, model training procedures, and evaluation setup are all
157 detailed in Section 4. The proposed method is compared to three benchmark ML models in
158 terms of prediction accuracy in Section 5. The study is finally concluded in Section 6 with a
159 summary of the primary findings.

II. Modeling and Data Collection with Finite Element Simulation of Wave Propagation

The study presents a model that aims to simulate the propagation of a single longitudinal burst wave through a steel pipe with a 1.0 m length, an outer diameter (D_o) of 34 mm, and a wall thickness (t) of 4 mm. In order to produce UGWs, a circular arrangement of PZT transducers is placed around the circumference of the pipe's leading edge, as shown in Fig. 1.

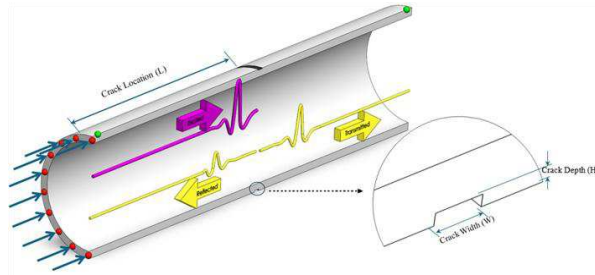


Fig. 1. Schematic of a pipe with an idealized crack, depicting the excited, reflected, and transmitted signals.

The transducers are activated by a function generator. The resulting waves intersect with a hypothetical circular fracture positioned at different positions along the pipe segment. Each crack is defined by specific geometric properties, such as its depth and width, as shown in Fig. 1. The wave patterns that bounce back are detected by two PZT sensors: one positioned 10 mm from the front end (referred to as the "front sensor" in Fig. 1), and another placed 10 mm from the rear end (referred to as the "rear sensor"). Crack characterization entails the assessment of three parameters: the position of the crack (L), the width of the crack (W), and the depth of the crack (D). Within FEA simulations, these parameters are methodically modified within specified ranges. The crack position (L) varies from 50 mm to 950 mm in increments of 50 mm, resulting in 19 different cases of L . The crack width (W) ranges from 1 mm to 5 mm, increasing by 1 mm increments, resulting in 5 different cases. Similarly, the crack depth (D) ranges from 1 mm to 3 mm, increasing by 1 mm increments, resulting in 3 different situations. Therefore, it is necessary to conduct a total of 285 simulations in order to examine every potential combination of fracture characteristics. Table 1 presents the simulation test matrix.

FEA is frequently used to model the propagation of waves in different structures, including plates, pipelines, and train tracks, for the aim of health monitoring. This work employed FEA simulations to recreate virtual sets of failure scenarios in a pipeline. The simulations with various fracture scenarios were conducted using ABAQUS/Explicit, a commercially available FEA software tool that is well-known for its ability to simulate several physical phenomena. The organized characteristics of the crack shape necessitated the utilization of 8-noded linear solid brick elements (C3D8R) for simulating the pipe with the flaw. In order to replicate the stimulation caused by a circular arrangement of PZT transducers located at the front end of the pipe, a predetermined displacement was delivered in the longitudinal direction to all the terminal nodes of the pipe. The displacement occurred in response to a triangular impulsive wave with a frequency of 100 kHz for a length of 0.0005 seconds. This wave was defined using the tabular definition approach in ABAQUS/Explicit, as shown in Table 2. The elastic properties of the pipe are: Young's modulus (193 GPa), Poisson's ratio (0.29), mass density (8027 kg/m³), and longitudinal wave speed (4903 m/s). These

parameters were considered due to the small strain levels applied to the pipe. The dimensions of the element were specified as 1 mm x 1 mm in the longitudinal and radial directions, while 23 elements were added in the circumferential direction.

Table 1
FEA simulation test cases: crack position, depth, and width.

Position, L (mm)	Depth, D (mm)	Width, W (mm)
50	1	1
100		2
⋮		3
900		4
950		5
50	2	1
100		2
⋮		3
900		4
950		5
50	3	1
100		2
⋮		3
900		4
950		5

Table 2
Load amplitude ratio settings used in the FEA model.

Time (s)	0	2.5	5	7.5	10
Amplitude					
Ratio	0	1	0	-1	0

Explicit solvers, such as ABAQUS, perform analyses by dividing the time domain into many increments to ensure precise representation of the test. The stable time increment (Δt) for problems involving solving undamped general equations of motion was determined based on the greatest frequency of interest (f_{max}), as indicated in Eq.(1) [46], where f_{max} is the highest frequency of interest.

$$\Delta t \leq \frac{1}{20 f_{max}} \quad (1)$$

The steady time increment was determined to be 5×10^{-7} seconds when the excitation signal frequency was adjusted to 100 kHz. For this work, a time increment of 1×10^{-7} seconds was used to guarantee the stability of the numerical simulation. In addition, the size of the element was calculated by assuring a minimum of 10 nodes per wavelength, as recommended by [47]. Hence, Eq. (2) establishes the value of the element length (L_e) based on the minimum wavelength of interest (λ_{min}). Assuming that the speed of a compression wave in the pipe is 5778 m/s and the maximum frequency is 100 kHz, the length of the

224 element, L_e , is 2.9 mm. A pipe was discretized using an element size of 1 mm.

$$L_e \leq \frac{\lambda_{min}}{10} \quad (2)$$

225
226 The FEA model layout conforms to established guidelines for modeling guided wave
227 propagation in SHM systems, as per recommendations from experts and researchers [48].
228 Additionally, leveraging the double-symmetry of the structure allowed for modeling only one-
229 quarter of the pipe, resulting in significant reduction of computational costs. Utilizing the
230 explicit FEA solver employed in this study, analysis of the full model required four times the
231 time needed for the quarter model. Despite the decreased computation time achieved
232 through symmetry conditions, and considering the substantial number of parametric
233 simulations along with the high memory and computing speed demands inherent to this type
234 of problem, all 285 analysis runs were executed utilizing high-performance computing
235 resources.

236 First, the simulation test was performed on a reference pipe without any defects. Fig. 2
237 illustrates the resulting time history of the longitudinal amplitude as detected by the front
238 sensor. From the figure, it can be observed that the excitation wave passed the sensor at
239 approximately 0.1×10^{-4} seconds. Furthermore, since the reference pipe had no cracks, the
240 sensor did not detect any significant reflected waves.

241 Examples of the results from the 285 cases of simulation runs are displayed in Figs. 3-5.
242 These figures illustrate the effects of the crack's position, width, and depth on the waves
243 detected by the front sensor. Fig. 3 presents results for three crack scenarios with identical
244 width and depth ($W=3$ mm and $D=2$ mm) but at different positions: case 1 at $P=200$ mm,
245 case 2 at $P=500$ mm, and case 3 at $P=800$ mm. It can be observed that the closer the crack
246 is to the sensor, the faster the reflected wave is detected by the front sensor. Additionally,
247 there is no clear relationship between wave amplitude and crack position.
248

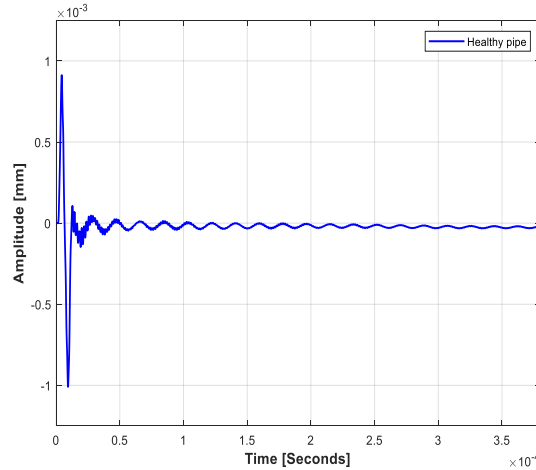


Fig. 2. Longitudinal Displacement at Front Sensor for Intact Pipe.

249
250
251

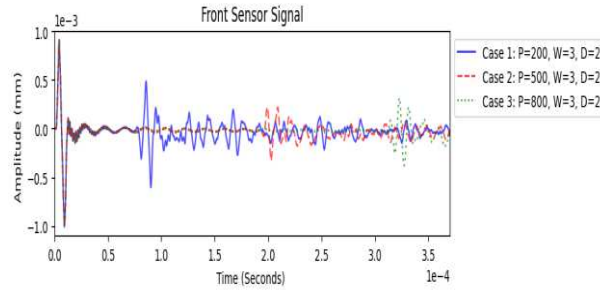


Fig. 3. Effect of Crack position on Longitudinal Displacement at Front Sensor.

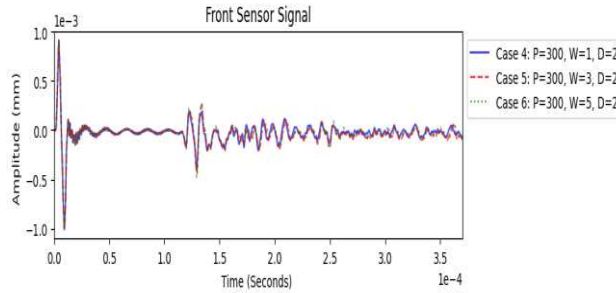


Fig. 4. Effect of Crack Width on Longitudinal Displacement at Front Sensor.

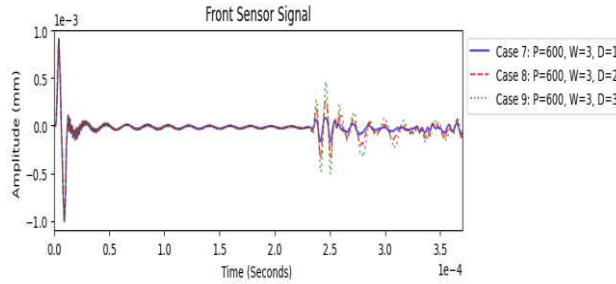


Fig. 5. Effect of Crack Depth on Longitudinal Displacement at Front Sensor.

Fig. 4 presents the results for three crack cases with a constant position and depth ($P = 300$ mm and $D = 2$ mm) but varying widths: case 4 with $W = 1$ mm, case 5 with $W = 3$ mm, and case 6 with $W = 5$ mm. The figure shows that the front sensor captured the reflected waves from the cracks simultaneously; however, the peak values differed. The results indicate that the peak of the reflected wave rises as the crack width increases

Fig. 5 illustrates the results for three crack scenarios in which the cracks have identical position and widths but vary in depth: case 7 ($P = 600$ mm, $W = 3$ mm, $D = 1$ mm), case 8 ($P = 600$ mm, $W = 3$ mm, $D = 2$ mm), and case 9 ($P = 600$ mm, $W = 1$ mm, $D = 3$ mm). The figure shows that the front sensor recorded the reflected waves from these cracks simultaneously, though with different peak values. The results indicate that the peak of the reflected wave increases with crack depth. Additionally, comparing the peaks in Figs. 4 and 5 reveals that changes in crack depth have a greater impact on peak values than changes in crack width.

The simulation results described above represent the waves detected by the front sensor. On the rear side, it is notable that the first wave reaching the rear sensor is the excitation wave after it has crossed the crack, referred to as the 'propagating wave'. Figs. 6-8 display the simulation waves detected by the rear sensor for the same nine cases previously shown

for the front sensor, illustrating the effects of crack position, width, and depth, respectively.

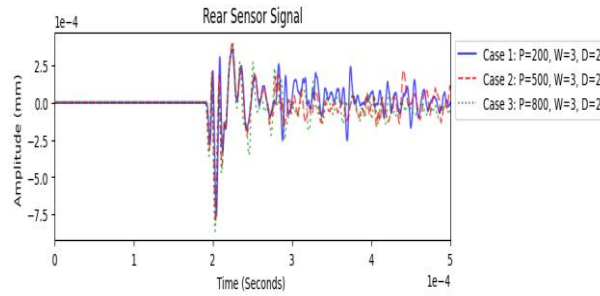


Fig. 6. Effect of Crack position on Longitudinal Displacement at Rear Sensor.

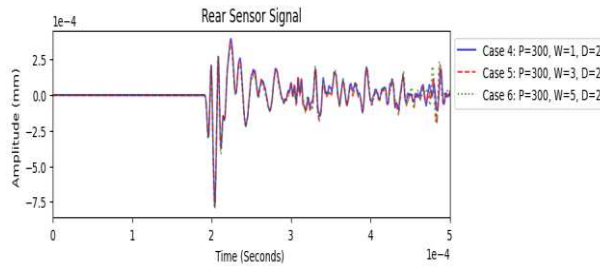


Fig. 7. Effect of Crack Width on Longitudinal Displacement at Rear Sensor.

Fig. 6 shows the wave responses recorded by the rear sensor for cases 1, 2, and 3, previously depicted in Fig. 3 for the front sensor. Similarly, Fig. 7 presents the rear sensor wave responses for cases 4, 5, and 6, initially shown in Fig. 4 for the front sensor. Finally, Fig. 8 displays the rear sensor responses for cases 7, 8, and 9, corresponding to Fig. 5 for the front sensor.

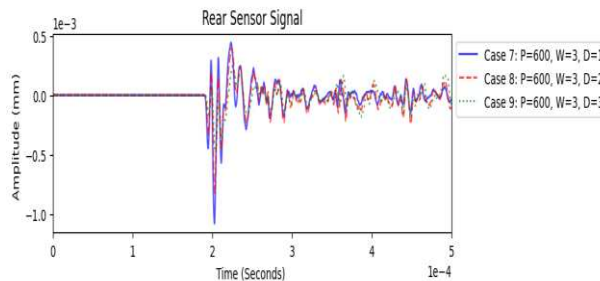


Fig. 8. Effect of Crack Depth on Longitudinal Displacement at Rear Sensor

A comparison between Figs. 6-8 for the rear sensor and Figs. 3-5 for the front sensor suggests that the effects of crack position, width, and depth on the reflected waves captured by the rear sensor are less distinct than those captured by the front sensor. This finding indicates that, while there is a relationship between the crack geometry parameters (position, width, and depth) and the resulting waves, establishing a consistent predictive relationship between excitation wave measurements and crack characteristics is complex. Therefore, this problem is well-suited for machine learning regression techniques. In the next section, the Wav2Vec2 neural network will be applied to capture the relationship between crack position, width, depth, and their reflected waves.

III. Proposed Hybrid Wav2Vec2 / Multi-Head Deep FeedForward Neural Networks

The proposed method for pipeline health monitoring leverages the capabilities of deep learning and transfer learning to enhance the detection and characterization of structural defects. Specifically, this study introduces a hybrid Wav2Vec2 /multi-head deep feedforward (MHDF) neural networks approach for predicting crack position, length, and width in pipelines. The architecture, illustrated in Fig. 9 begins with the captured UGW signals, which are input into a fine-tuned Wav2Vec2 model. This model automatically extracts high-level features from the raw acoustic data. The extracted features are then passed to a multi-head feedforward neural network, which outputs the three target crack parameters. The proposed hybrid Wav2Vec2 / MHDF approach combines the powerful feature extraction capabilities of transfer learning with the precision of supervised deep learning, enabling accurate and robust prediction of crack characteristics even with limited labeled data. This approach does not need any hand-crafting feature vector by extracting rich and useful information that distilled in CNN and transformer layers of pre-trained Wav2Vec 2.0 model.

Training of the proposed hybrid Wav2Vec2 / MHDF neural network is carried out in two stages. In the first stage, a pre-trained Wav2Vec2 model is fine-tuned using the collected and augmented UGW data, as illustrated in Fig. 9. Once fine-tuning is complete, the Wav2Vec2 model is connected to a MHDF neural network, as shown in Fig. 10, and the second training stage is performed. During this second stage, the parameters of the Wav2Vec2 model are kept fixed, while the parameters of the MHFF neural network are trained to optimize the prediction of crack position, length, and width.

III.1. Wav2vec2 model

Wav2vec2 model is a well-known for its state-of-art performance in sound signal encoding and processing. The model architecture integrates the deep convolutional neural network (DCNN) and transformer with self-supervised learning framework. as depicted in Fig. 11. The model encodes the raw waveforms X signal into latent waveform Representations $Z_1, \dots, Z_T$. The model utilizes the latent representation of DCNN as input to transformer-masked network $g: Z \rightarrow C$. The function of transformer network is to quantizes the continuous representation to form a discrete set of outputs $q_1, \dots, q_T$, as shown in Fig. 11. These discrete outputs are utilized in self-supervised learning objectives as targets. Then, the attention blocks in the transformer module contextualized those quantized representations to form a set of discrete contextual representations $C_1, \dots, C_T$. The encoder in the Wav2vec2 consists of seven layers with 512 channels. The stride-kernel for the seven convolutional layers are configured as: $S/K: 5/10, 2/3, 2/3, 2/3, 2/3, 2/2, 2/2$ the transformer network has 24 blocks with 1024 and 4096 inner dimensions numbering, and a total of 16 attention head.

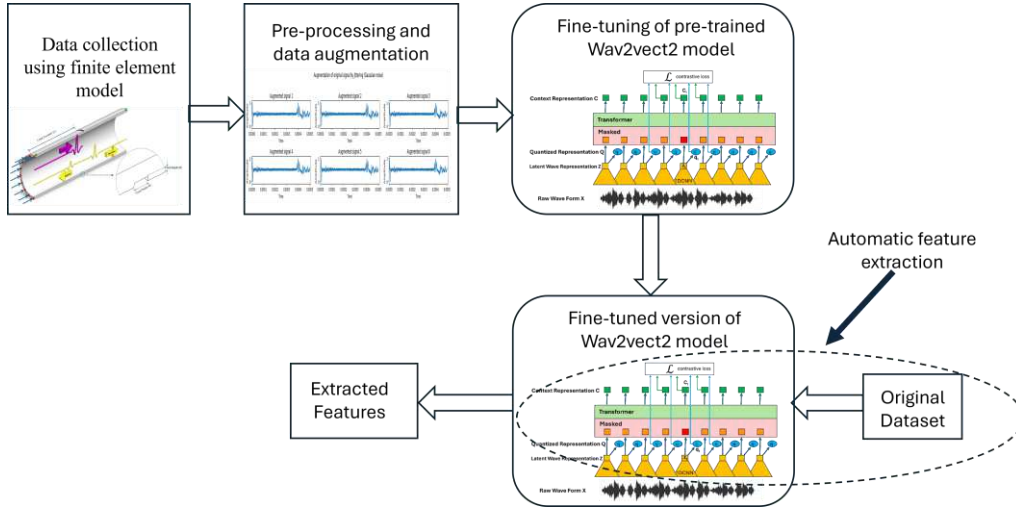


Fig. 9. Flow chart for the proposed hybrid Wav2Vec2 / MHDFN NNs

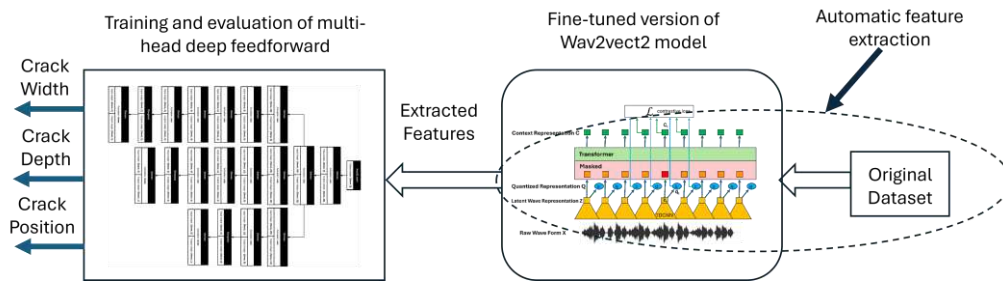


Fig. 10. Flow chart for the fine tuning stage for the Wav2vec2 NN

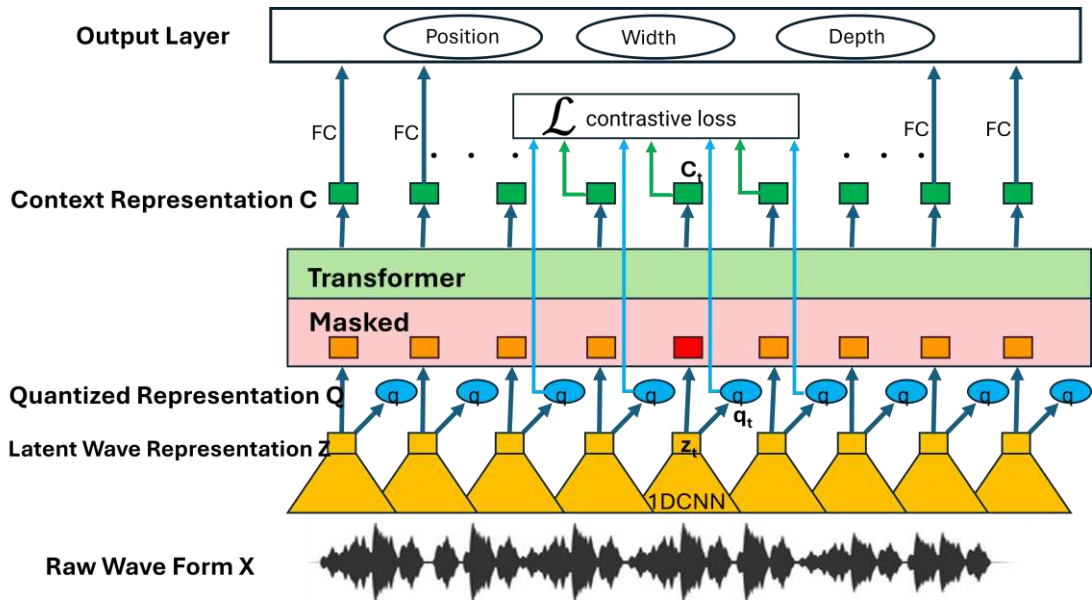


Fig. 11. Wav2vec2 model structure

The automatic feature extraction in the Wav2Vec2 module exploits the power of its DCNN

which transforms raw wave signal into useful representation that can be used in our target application. Since the Wav2vec2 model has been pre-trained on a huge dataset of wave signal encompassing voiced and unvoiced signals, its CNN latent space extracts useful features from wave-like signals.

III.2. Multi-head Deep Feedforward neural network

The Multi-Head Deep Feedforward (MHDF) neural network architecture is a specialized extension of standard feedforward networks, designed to handle multi-output learning tasks more effectively, particularly when the outputs exhibit heterogeneous characteristics. Structurally, the MHDF model consists of two main components: a shared feature extraction trunk and multiple output-specific branches, or "heads." The shared layers learn a common representation from the input features, capturing general patterns that benefit all output tasks. These layers typically employ activation functions such as Rectified Linear Unit (ReLU), which are well-suited for deep architectures due to their non-saturating gradient behavior. After the shared representation is formed, the network diverges into multiple independent branches, each dedicated to a specific output variable. This architectural separation allows each head to fine-tune its parameters based on its own loss signal, thereby avoiding the problem of negative transfer—a common issue in standard fully connected networks where a single output layer must accommodate conflicting learning objectives. By isolating the gradient flows and allowing each task to evolve independently from a shared base, MHDF networks enable more accurate, stable, and interpretable predictions for complex, multi-target regression problems. Additionally, the MHDF design supports flexible model complexity through tunable hyperparameters such as the number of shared layers, the number and depth of branches, and neuron decay strategies. When combined with proper regularization techniques and optimization schemes such as cross-validation, MHDF networks offer a powerful and efficient framework for solving high-dimensional, multi-task learning problems across a wide range of applications.

IV. Training the Hybrid Wav2Vec2/ MHDF

One can note that the proposed approach does not need any hand-crafting feature vector by extracting rich and useful information that distilled in DCNN and transformer layers of pre-trained Wav2Vec 2.0 model. Also, this approach address data scarcity problem by exploiting the power of transfer learning and data augmentation methods. Data scarcity significantly impacts the model performance, when data is scarce, the possibility of overfitting increases, where the model memorizes the training data instead of learning generalizable feature. This will lead to poor performance on unseen data, as the model cannot adapt well to new scenarios. As mentioned previously the training of the proposed approach will be conducted in two stage. But initially data need to be preprocessed and augmented.

IV.1. Data preprocessing and augmentation

In this section, collected data will be prepared for feature extraction module including checking for outliers, normalization, and increasing the number of samples in the dataset.

One challenge in developing deep learning systems is the need for large amounts of data. Our case study represents a case of limited data, with only one sample per crack scenario. To mitigate data scarcity, the dataset has been augmented by various methods including; adding noise (jittering), magnitude wrapping, and homogenous scaling to increase the number of samples in the dataset. The signals in the dataset are normalized to intervals of $[0,1]$ to reduce the effects caused by signal amplitudes using the following equation:

$$z_k = \frac{x_k - \min(x)}{\max(x) - \min(x)} \quad (3)$$

Where x_k is the k th element of sample, $\max$ and $\min$ are the maximum and minimum of the sample respectively, z_k is the k th element of the normalized sample x . Fig. 12 shows an example of a normalized input signal. Moreover, in order to ensure the model does not prioritize the output with a larger scale during training, it is crucial to address this problem in our dataset where one output (position) significantly outweighs the others in scale as shown in Table 1. The Output Normalization method has been applied to the three outputs in the dataset. Below equation is used to Normalize each output to a similar scale before training. This helps maintain balanced treatment across outputs by the loss function.

$$y_{norm} = 0.1 + \frac{y - \min(y) * 0.9}{\max(y) - \min(y)} \quad (4)$$

As previously discussed, the proposed system incorporates a data augmentation approach to mitigate data scarcity. The dataset size is expanded using three techniques: adding Gaussian white noise, applying homogeneous scaling, and warping the magnitude of the original signals. It is crucial to emphasize that the adopted augmentation methods preserve the critical crack-related features: time of flight (ToF), normalized amplitude of the incident wave, and normalized amplitude of the reflected wave. These features are essential for characterizing cracks in terms of position, width, and depth. On the other hand, any augmentation method that distorts data's temporal characteristics (i.e. ToF) such as time shifting, time warping, and time slicing could not be used. It is important to mention that the data augmentation methods were applied to the training dataset only. To strictly prevent any data leakage, a split-then-augment pipeline was enforced. First, the original 285 FEA simulation samples were randomly partitioned into training and testing subsets based on the 10-fold cross-validation scheme. The test data was immediately isolated. Only after this isolation were the data augmentation methods (Gaussian white noise, homogeneous scaling, and magnitude warping) applied to the training folds. This ensures that the test set consists solely of 100% un-augmented, previously unseen original samples, thereby providing a true evaluation of the model's generalization capabilities. The training data is used to train the pre-trained Wav2vec2 model to adapt the weights of deep convolutional neural network (FCNN) to our target domain.

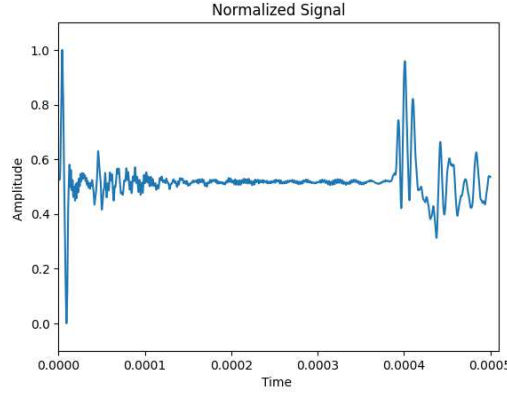


Fig. 12. Example of one of the normalized input signals

Our collected data are augmented by adding gaussian noise to the original signal, the augmented signal can be expressed as :

$$y_{aug} = y_{orig} + \mathcal{N}(0, \sigma^2) \quad (5)$$

Where y_{orig} is the original signal, $\mathcal{N}(0, \sigma^2)$ is Gaussian noise with mean 0 and variance σ^2 , and σ is the standard deviation that controls the noise level. Also, the original signals are augmented by homogenous scaling method, the augmented signal can be expressed as:

$$y_{aug} = \alpha * y_{orig} \quad (6)$$

Where $\alpha > 0$ defines the scale of the change. This value can be defined by a Gaussian distribution with mean of 1 and with σ as a hyperparameter. Figs. 13-15 show an example of augmentation of one data set using jittering, scaling, and warping, respectively.

IV.2. Stage 1 Training: Fine Tuning of Wav2Vec2 Model

In stage 1 of training, the Wav2Vec2 will be fine-tuned. We utilized a pre-trained Wav2Vec 2.0 acoustic model based on the Wav2Vec2-XLS-R-300M architecture, accessible through Hugging Face [49]. This model was initially trained in a self-supervised manner using 436K hours of unlabeled speech data from 128 languages, sourced from diverse corpora such as VoxPopuli, Multilingual LibriSpeech, Common Voice, BABEL, and VoxLingua107. This pre-trained model was subsequently fine-tuned on the training set of our dataset including augmented data. Fine-tuning a pre-trained Wav2Vec 2.0 model for a pipeline health monitoring task, where the input signal is ultrasound wave signal, and the outputs are crack parameters in the pipeline (position, width, and depth). The problem can be considered as multi-output regression; hence the Wav2Vec 2.0 model architecture and training strategy was adapted to handle new input and output types. An output layer configured as fully-connected linear regressor layer with three-output has been added at the end of the model as shown in Fig. 11. The three outputs represent crack position, crack width, and crack depth. The model was trained using the Root Mean Squared Error (RMSE) loss function and the Adam optimizer. The weighted loss function and regularization function have been adopted in constructing and training deep learning neural network to ensure that the loss contributed by the larger-scale and most variance output does not dominate the training process. weighted loss function is given by:

$$L = w_{pos} \cdot Loss(y_{pos}, \widehat{y_{pos}}) + w_{width} \cdot Loss(y_{width}, \widehat{y_{width}}) + w_{depth} \cdot Loss(y_{depth}, \widehat{y_{depth}}) \quad (7)$$

The training loop iterated for 50 epochs, updating the model's regression head to minimize the loss. The outputs from the last layer of the DCNN and transformer layers of the fine-tuned Wav2Vec 2.0 model were collected for each sample in the dataset. the size of the vector that represent the collected outputs from DCNN and transformer layer were 512 and 768, respectively. Since, the extracted feature vector is created by concatenating both DCNN and transformer outputs, Principal Component Analysis (PCA) is used to overcome the curse of dimensionality and avoid overfitting of the MHDFE neural network. This is a critical preprocessing step before feeding the extracted feature vector to MHDFE neural network, where it helps reduce the number of input features while retaining significant information, in other words it transforms the data into a new coordinate system where the greatest variances are captured by the first few principal components. the size of the resultant feature vector after dimensionality reduction method is only 10. this feature vector will be used as input to train MHDFE neural network which designed to capture the nonlinear relationship between inputs and outputs in our research problem.

It can be noted that the automatic feature extraction of the developed Wav2Vec2 model exploits the power of its DCNN which transforms raw wave signal into useful representation that can be used in our target application. Since the Wav2vec2 model has been pre-trained on a huge dataset of wave signal encompassing voiced and unvoiced signals, its DCNN latent space extracts useful features from wave-like signals. To extract useful features from training data in our dataset, the weights of DCNN in the tuned wav2vec2 model are extracted. Then, the principal component analysis method is applied to reduce the number of features in input vectors before feeding them to the deep multilayer neural network configured as regressor that predicts the three crack parameters (position, width, and depth). Since number of samples in our dataset is limited, reducing the length of feature vector is crucial to avoid overfitting memorizing the training data and losing the capability of generalization. System evaluation module is used to check the performance of the trained systems on the training data and unseen testing data. The mean squared error of training data and testing data are plotted and compared to assist the training process for the proposed system.

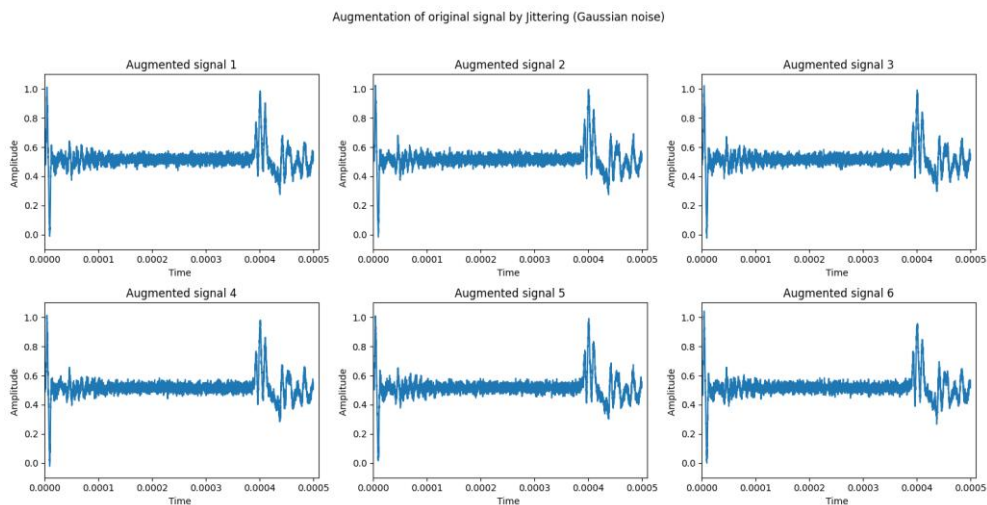


Fig. 13. Example of one of the augmented signals using jittering (Gaussian noise)

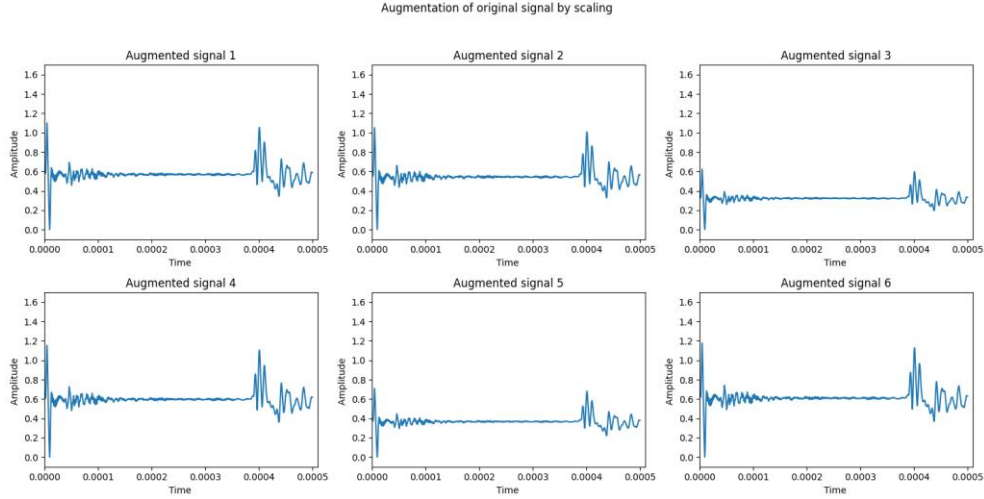


Fig. 14. Example of one of the augmented signals using scaling

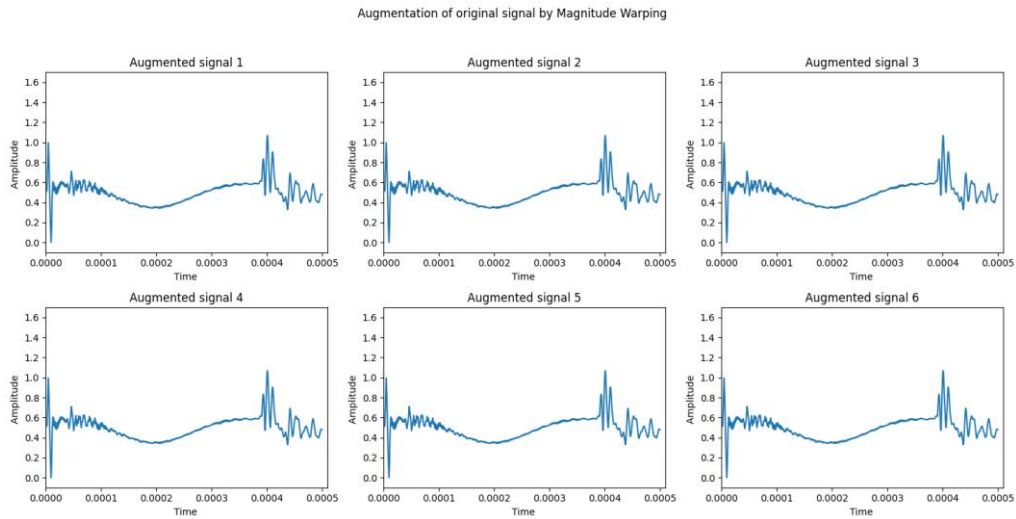


Fig. 15. Example of one of the augmented signals using magnitude wrapping

IV.3. Stage 2: Training of the Multi-Head Deep Feedforward Neural Network

The main idea. In this section, we discuss the training and evaluation process of the MHDFF neural network which is performed using keras framework. The extracted feature vector from the fined-tuned wav2vec2 system are used to train MHDFF

neural network. The architecture of the MHDFF network is designed to address our multi-output regression problem, where the outputs are position, width, and depth. Each output is trained to minimize the Root Mean Squared Error (RMSE), utilizing a custom loss function to appropriately weigh the contributions of each output. The MHDFF model is constructed using a sequential approach. The input layer accepts features from the training data, followed by several dense layers with ReLU activation functions. The proposed system adopted a multi-head (also known as multi-task) NN architecture as shown in Fig. 16. This architecture was specifically chosen over a standard Fully Connected Neural Network (FCNN) to address the challenge of predicting multiple outputs with heterogeneous

characteristics. A standard FCNN would utilize a single stack of hidden layers culminating in a final output layer with three neurons. In such a design, all outputs share the entire network's representational capacity, forcing the model to find a single set of high-level features that is suboptimal for all tasks. This leads to a phenomenon known as negative transfer or task interference, where the large error gradients from the high-variance 'position' output can overwhelm the smaller, subtler gradients required to fine-tune the 'width' and 'depth' predictions. The multi-head architecture directly mitigates this issue by separating shared learning from specialized learning. The network architecture consists of shared layers that extract useful information from input feature vector, in our case we have utilized Gridserach algorithm to optimize the number of shared layers and number of layers in each independent branch. Both shared hidden layers utilize ReLU activation function which is superior in training a deep neural network. The shared layers serve as a common feature extractor, learning low-level representations from the input features that are beneficial for all three prediction tasks. This shared learning is efficient and acts as a form of regularization, as it forces the model to find a compressed, general-purpose representation from the limited dataset which is one challenge in our case. Following the shared trunk, the network splits into three independent branches (Heads), one for each output (position, width, and depth). Each branch is a specialized sub-network that takes the shared features from the trunk and learns the final, specific mapping required for its target variable. This architectural separation is the key advantage; it allows the learning process for the 'depth' output, for example, to proceed without interference from the learning process for the 'position' output. Each head can fine-tune its branch weights based on its own error signals, leading to more accurate and stable training for the lower-variance outputs. Variable branch depth and variable shared layers depth have been utilized to investigate optimal model complexity. The total number of shared layers in the model was iterated from 1 to 5, while each independent branch was iterated from 1 to 7. The total number of hidden layers (shared layers and independent branch layers) was optimized based on performance measure of RMSE on validation set using 10-Fold Cross-Validation scheme. Cross validation approach provides a more robust and less biased estimate of model performance than a single train-test split by training and validating the model 10 times on different partitions of the data. The data was shuffled before splitting to ensure random distribution across folds.

Another important hyperparameter that needs to be optimized is the number of neurons in the hidden layers (both shared and independent branch layers). We have applied Neuron Decay as design restriction to the whole neural network architecture such that the neuron count for each subsequent layer in hidden layer is decreasing as the depth of layer increases. Three separate output layers are defined with one neuron in each layer, each corresponding to one of the outputs (position, width, and depth). Each output uses linear activation function to predict a continuous value. The model is compiled with an Adam optimizer and a custom RMSE loss function that computes the total loss by weighted loss function as expressed in equation (7), the total loss function computes the root mean squared error for each output and aggregates them into a total loss, which is then used during the training phase.

To predict the multi-dimensional characteristics of cracks (position, width, and depth) from a set of input features which is extracted automatically from Wav2vect architecture, a systematic computational experiment was designed and executed. The primary objective

was to develop a robust predictive model while addressing the inherent challenges of a small dataset, the heterogeneous variance of the output variables, and the high risk of model overfitting. The standard pipeline for training and evaluation of MHDFE model, from data handling to model evaluation is depicted in Fig. 16. The pipeline starts with data preprocessing and normalization modules. The function of this module is to prepare input and output variables to ensure stable and effective model training and evaluation. Given the disparate scales of the input and output variables, data normalization strategy was employed on both input and output variables. This critical step ensures that the contribution of each output to the total loss function is proportional, preventing the high-variance 'position' output from dominating the learning process and allowing the model to learn effectively for all three target variables simultaneously. Due to limited training samples in our dataset, overfitting problem may occur where the model becomes overly specialized in mapping the specific input-output relationships present in the training set, including any random fluctuations or noise, and thus fails to accurately predict the outputs for new input. This leads to excellent performance on the training data but poor performance on new unseen data. Therefore, a multi-faceted regularization strategy was implemented including L2 Regularization (Weight Decay), Dropout, and Early Stopping, in order to mitigate and combat the overfitting challenge. L2 regularization was employed on the kernels of all dense layers, with a regularization coefficient $\lambda=0.01$.

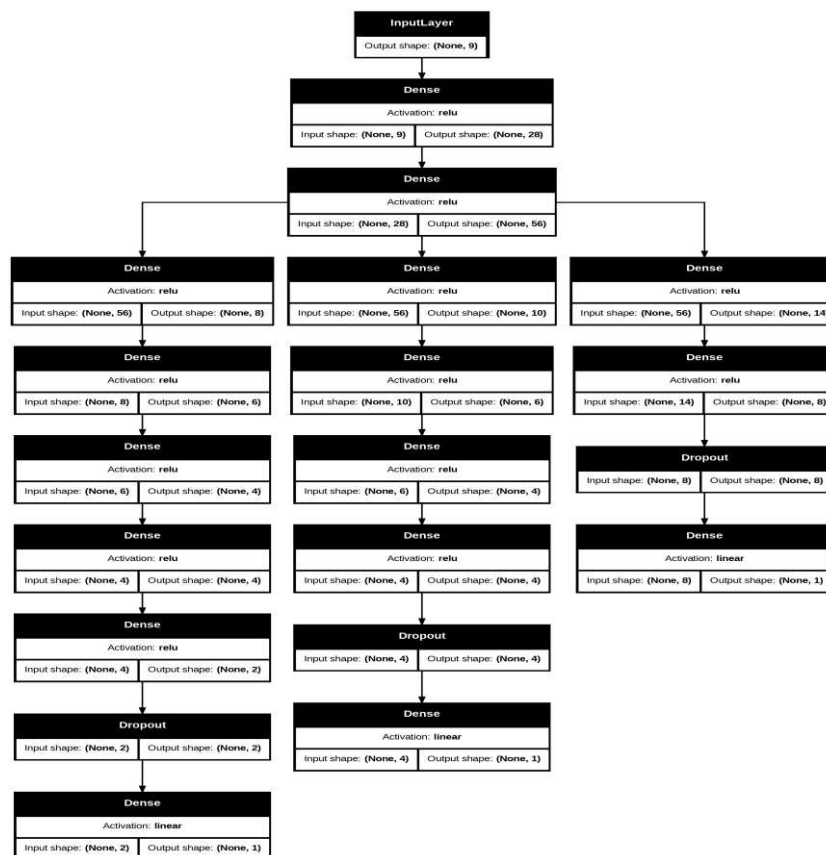


Fig. 16. The architecture of MHDFE neural network.

. This approach penalizes large weight magnitudes, thereby encouraging the development

of a simpler model less susceptible to fitting noise in the training data. Furthermore, Dropout regularization was incorporated by introducing a dropout layer with a rate that varies from 0.1 to 0.3 following

each hidden layer. During training, this mechanism stochastically disables 10% to 30% of the neurons at each iteration, reducing reliance on specific activations and promoting the learning of distributed, robust feature representations. Finally, to prevent overtraining, an Early Stopping strategy was adopted by monitoring the validation loss. If no improvement was observed over 15 consecutive epochs, the training process was halted, and the model parameters corresponding to the lowest recorded validation loss were restored to maximize generalization performance. Dynamic learning rate is also implemented such that to adjust the learning rate dynamically based on the epoch. In this way, the model is trained for a specified number of epochs and captures the loss associated with various learning rates to identify the optimal rate. The outcomes of Grid search algorithm that have been utilized to find the best hyperparameters for Multi-head MLP neural network are depicted in Table 3.

Table 3
Outcomes of Grid search algorithm for MHDFF neural network hyperparameters.

Layer Type	# of layers	Number of neurons
Shared Layer	2	First-shared hidden layers =56 second-shared hidden layers =28
Independent Branch for position output	3	First hidden layer for position= 14 Second hidden layer for position= 14 Third hidden layer for position= 14
Independent Branch for width Output	5	First hidden layer for width= 8 Second hidden layer for width= 6 Third hidden layer for width= 4 Fourth hidden layer for width= 4 Fifth hidden layer for width= 2
Independent Branch for Depth Output	4	First hidden layer for depth= 10 Second hidden layer for depth= 6 Third hidden layer for depth= 4 Fourth hidden layer for depth= 2

IV.4. Training and Validation Results for the Hybrid Wav2Vec2/MHDFF

The main word. After training, a rigorous evaluation protocol was established to obtain a reliable estimate of the performance for the best model configuration. 10-fold cross-validation approach is utilized to address biased model evaluation on limited datasets by systematically reducing the impact of arbitrary data splits. This method provides a more robust and reliable estimate of a model's true generalization performance than a single train-test split. A single train-test split on a limited dataset can lead to a biased evaluation because

the specific random partition of data can significantly influence the observed performance. If the test set happens to contain "easy" samples, the model's performance might appear inflated; conversely, if it contains "hard" or unrepresentative samples, the performance might be underestimated. This introduces sampling bias, where the evaluation metric is highly sensitive to the particular subset chosen for testing. However, the 10-fold cross-validation scheme mitigates this bias through a comprehensive, iterative evaluation process. First, the entire dataset is randomly divided into ten equally sized, non-overlapping subsets (folds). Then, the model is subjected to ten training and evaluation cycles. In each cycle: One-fold serves as the validation (test) set and the remaining nine-folds are aggregated to form the training set. The model is trained on this combined training set and its performance is evaluated on the designated validation set. This process ensures that every data point in the dataset is used for validation exactly once across the ten iterations, and for training in the other nine iterations. The performance metrics obtained from each of the ten iterations are then averaged to yield a single, more stable, and statistically reliable estimate of the model's generalization capability. In our system, the performance of the best model is evaluated on using 10-fold cross validation to assess its performance. The evaluation metrics used include Root Mean Squared Error (RMSE) for each output. This evaluation phase provides insights into how well the model predicts each of the outputs on training and test sets.

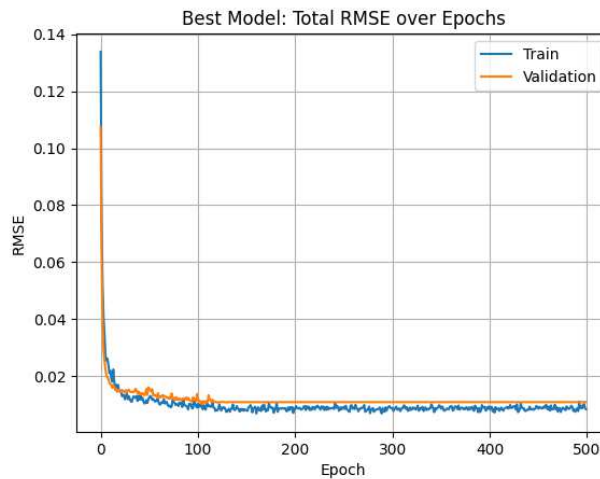


Fig. 17. Total RMSE of the hybrid Wav2Vec2/MHDFF network with for training and validation data

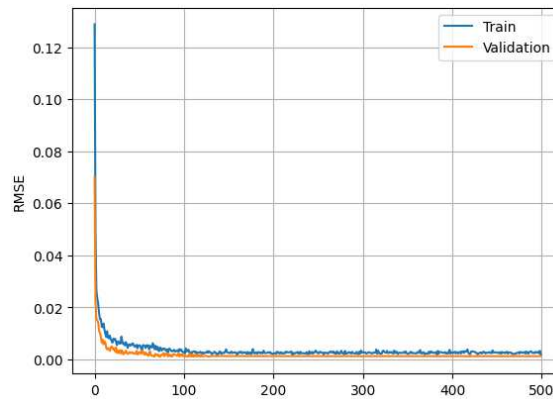


Fig. 18. Position RMSE of the hybrid Wav2Vec2/MHDFF network with for training and validation data

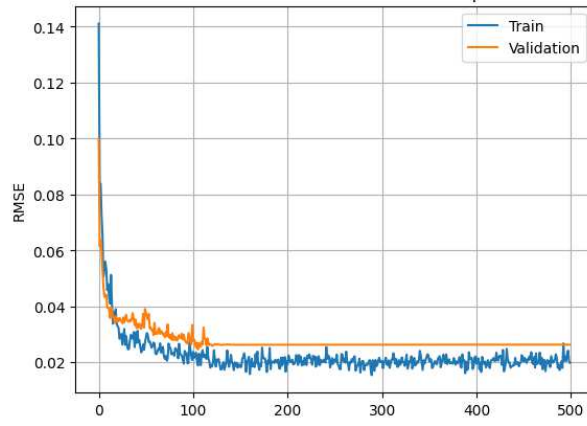


Fig. 19. Width RMSE of the hybrid Wav2Vec2/MHDFF network with for training and validation data

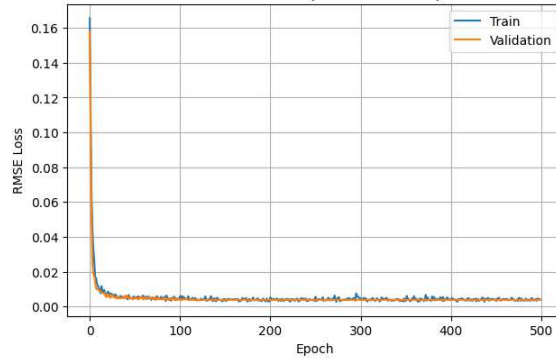


Fig. 20. Depth RMSE of the hybrid Wav2Vec2/MHDFF network with for training and validation data

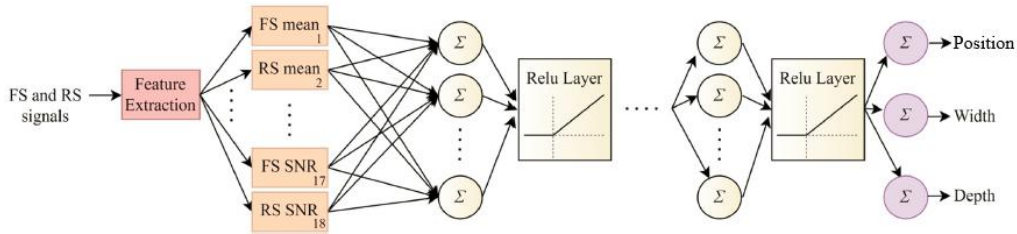


Fig. 21. Deep neural networks structure, FS is the front sensor, and RS is the rear sensor.

By performing multiple training and evaluation cycles across different partitions of the data, 10-fold cross-validation reduces the dependency on a single, potentially unrepresentative split. This comprehensive approach provides a less biased and more stable assessment of a model's predictive power on unseen data, which is particularly crucial when working with limited datasets. Also, the full training history was recorded and averaged across all 10 folds. The overall loss and individual output metrics are displayed for training and testing data as shown in Figs. 17-20, allowing for an assessment of whole model performance and its performance for each individual output. Moreover, these figures allowed for a detailed diagnosis of the training dynamics and overfitting behavior of each output branch. Fig. 17 shows that the overall RMSE of 0.0140 was achieved after 100 epochs. While Fig. 18 shows that the position RMSE of 0.0009 was achieved around epoch 150. Width RMSE of 0.0241 was achieved around epoch 150 as depicted in Fig. 19, while depth RMSE of 0.0036 was achieved around epoch 100 as depicted in Fig. 20.

V. Performance Comparison

In this section, the proposed hybrid Wav2Vec2/MHDFE approach will be compared against three common ML based prediction models: i) Deep Neural Network (DNN) with hand-crafted features as introduced in [50] , ii) Bidirectional Long Short-term Memory (Bi-LSTM), and iii) Bi-LSTM with Principle Component Analysis (PCA-Bi-LSTM).

V.1. Deep Neural Network with Hand-Crafted Features

Fig. 21 illustrates the Deep Neural Network (DNN) model introduced in [50] and employed in this study. The network consists of eight layers, including an input layer, an output layer, and three fully connected hidden layers comprising 200, 100, and 50 neurons, respectively. Each fully connected layer is followed by a ReLU activation layer. The DNN receives as input the extracted features from both front and rear sensors, and it outputs the predicted crack position, width, and depth in the pipe. For each FEA simulation, nine signal features are extracted: mean, standard deviation, root mean square, kurtosis, skewness, peak value, impulse factor, crest factor, and signal-to-noise ratio. As data is collected from two sensors, the total number of input features is 18. These time-domain features are commonly used for signal-based feature extraction and serve as informative indicators for predicting crack characteristics.

V.2. Bidirectional Long Short-term Memory (Bi-LSTM)

LSTM networks are a type of RNN that are well-suited for modeling time-dependent sequences and capturing long-term dependencies in data. A Bidirectional LSTM (Bi-LSTM) extends the traditional LSTM architecture by processing input sequences in both forward and backward directions. This allows the model to capture temporal dependencies from both past and future contexts, which is particularly useful in detecting subtle patterns associated with crack initiation and propagation over time. In this study,

the Bi-LSTM network was used to model the sequential behavior of structural health monitoring data from pipelines. Its ability to learn complex temporal patterns makes it a robust tool for predicting crack development and assessing pipeline integrity in advance. As shown in Fig. 22, each LSTM cell is made of forget, input and output gates. In Bi-LSTM two LSTMs are created: forward and backward ones and their hidden states are concatenated.

V.3. Bi-LSTM with Principal Component Analysis (PCA-Bi-LSTM)

PCA is a widely used statistical technique for reducing the dimensionality of high-dimensional datasets while preserving as much variance as possible. In the context of pipeline crack prediction, sensor data can be high-dimensional, noisy, and often correlated. PCA helps transform the original feature space into a new set of orthogonal components, called principal components, ranked by the amount of variance they capture from the original data. By selecting only the top principal components, it becomes possible to reduce computational complexity, remove redundant features, and improve the generalization ability of machine learning models. In this work, PCA was employed as a pre-processing step to project the high-dimensional sensor input data into a lower-dimensional subspace, thereby enhancing the performance and training efficiency of the subsequent neural network model. The mathematical basis of PCA is described as follows

Considering a dataset with m samples and n features $X \in \mathbb{R}^{m \times n}$. To obtain the PCA of this dataset one should firstly standardize it by subtracting its mean $\bar{X}$ and dividing by its standard deviation $\text{std}(X)$ as:

$$\tilde{X} = \frac{X - \bar{X}}{\text{std}(X)} \quad (8)$$

After that the covariance matrix is calculated as :

$$C = \frac{1}{(m-1)} (\tilde{X}^T \tilde{X}) \quad (9)$$

In the next step the Eigen values and vector of the covariance matrix are calculated as:

$$C v_i = \lambda_i v_i \quad (10)$$

$$V_k = [v_1, \dots, v_k], k < n \quad (11)$$

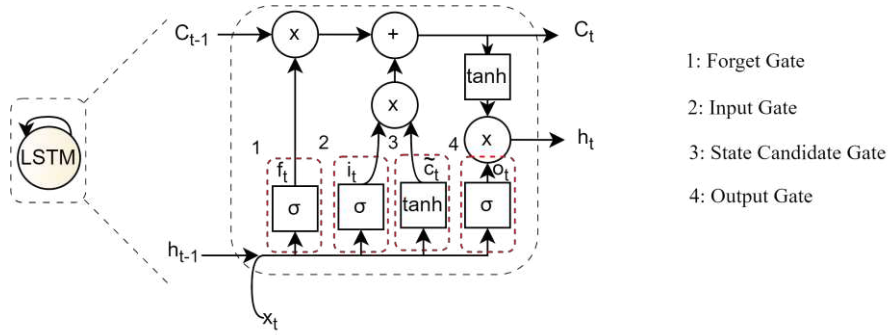


Fig. 22. The structure of one LSTM neuron

Table 4

Prediction performance of the proposed Wav2Vec2/MHDFD vs. other ML approaches using testing data.

Accuracy indicator	Wav2Vec2/MHDFD	DNN [50]	Bi-LSTM	PCA-Bi-LSTM
Total RMSE(mm)	0.0140	0.0288	0.3197	0.1601

Where λ_i is the eigen value and v_i is the eigen vector for each row in the covariance matrix, Finally, the original dataset is projected using the top Eigen vectors V_k as:

$$Z = \tilde{X} V_k \quad (12)$$

V.4. Prediction Performance Comparison

Table 4 presents the prediction performances of the proposed Wav2Vec2/MHDFD model compared to DNN, Bi-LSTM, and PCA-Bi-LSTM, evaluated on the same testing dataset. To ensure a rigorous and direct comparison, all competing baseline models were re-evaluated under a unified evaluation protocol. Despite utilizing different input formats—such as handcrafted time-domain features for the DNN [50] and raw waveforms for the Bi-LSTM variants—all models were subjected to the exact same 10-fold cross-validation scheme using the isolated, un-augmented hold-out test sets. The total RMSE values in Table 4 represent the combined RMSE for the three predicted variables: position, width, and depth. The proposed Wav2Vec2/MHDFD model achieves a total RMSE of 0.0140, which is significantly lower than that of the other ML methods. Specifically, it outperforms DNN, Bi-

LSTM, and PCA-Bi-LSTM by 51.4%, 95.6%, and 91.3%, respectively, in terms of total RMSE. These results clearly demonstrate the superior predictive performance of the proposed Wav2Vec2/MHDFF model

VI. Conclusions

This study illustrates the efficacy of integrating a pre-trained Wav2Vec2 architecture with a multi-head deep feedforward neural network for the structural health monitoring of oil and gas pipelines. The proposed system utilizes transfer learning and automatic feature extraction, facilitating accurate prediction of essential crack parameters—location, width, and depth—independently of manually designed features. The model effectively mitigates challenges related to limited datasets and multi-output regression through data augmentation and a two-stage training strategy. A comparative analysis with conventional ML models, including DNN and Bi-LSTM variants, demonstrates that the Wav2Vec2/MHDFF method attains enhanced performance with markedly reduced prediction errors. The findings highlight the capability of the suggested hybrid architecture to facilitate real-time, precise, and scalable pipeline health monitoring systems in safety-critical contexts. Future work will focus on validating the proposed framework using measured ultrasonic guided-wave data from experimental pipeline specimens to evaluate its robustness against complicated crack morphologies, environmental noise, and variable sensor coupling in real-world operating conditions.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors..

Declarations:

Conflict of interest

The authors declare no competing interests.

Ethical approval

Not Applicable.

Consent to publish

Not Applicable.

Consent to participate

Not Applicable.

Clinical trial

Not Applicable.

VII. References

- [1] X. Liang, W. Liang, L. Zhang, and X. Guo, "Risk assessment for long-distance gas pipelines in coal mine gobs based on structure entropy weight method and multi-step backward cloud transformation algorithm based on sampling with replacement," *J. Clean. Prod.*, vol. 227, pp. 218–228, Aug. 2019, doi: 10.1016/J.JCLEPRO.2019.04.133.
- [2] B. Liu, L. Yao He, H. Zhang, Y. Cao, and H. Fernandes, "The axial crack testing model for long distance oil-gas pipeline based on magnetic flux leakage internal inspection method," *Measurement*, vol. 103, pp. 275–282, Jun. 2017, doi: 10.1016/J.MEASUREMENT.2017.02.051.
- [3] B. Keshtegar and M. el Amine Ben Seghier, "Modified response surface method basis harmony search to predict the burst pressure of corroded pipelines," *Eng. Fail. Anal.*, vol. 89, pp. 177–199, Jul. 2018, doi: 10.1016/J.ENGFAILANAL.2018.02.016.
- [4] Y. Liu and Y. Bao, "Review on automated condition assessment of pipelines with machine learning," *Adv. Eng. Informatics*, vol. 53, p. 101687, Aug. 2022, doi: 10.1016/J.AEI.2022.101687.
- [5] D. L. Naik and R. Kiran, "Data Mining and Equi-Accident Zones for US Pipeline Accidents," *J. Pipeline Syst. Eng. Pract.*, vol. 9, no. 4, p. 04018019, Jul. 2018, doi: 10.1061/(ASCE)PS.1949-1204.0000340.
- [6] "Pipeline Incident 20 Year Trends | PHMSA," 2025. <https://www.phmsa.dot.gov/data-and-statistics/pipeline/pipeline-incident-20-year-trends> (accessed Nov. 03, 2025).
- [7] N. E. Andrzejczyk, V. P. Palace, L. Hrenchuk, L. Peters, and D. Schlenk, "RNA Sequencing of Lake Charr Epidermal Mucus to Assess Molecular Effects of Diluted Bitumen Exposure in a Boreal Lake," *Front. Environ. Sci.*, vol. 10, p. 836640, Feb. 2022, doi: 10.3389/FENV.2022.836640/BIBTEX.
- [8] R. Brinkmann, "Interconnections in Environmental Sustainability: Water and Energy," *Environ. Sustain. a Time Chang.*, pp. 195–216, 2020, doi: 10.1007/978-3-030-28203-5_9.
- [9] H. Ren, X. Chen, and Y. Chen, "Structural Health Monitoring and Influence on Current Maintenance," *Reliab. Based Aircr. Maint. Optim. Appl.*, pp. 173–184, 2017, doi: 10.1016/B978-0-12-812668-4.00009-5.
- [10] V. B. Sharma, S. Tewari, S. Biswas, and A. Sharma, "A comprehensive study of techniques utilized for structural health monitoring of oil and gas pipelines," *Struct. Heal. Monit.*, vol. 23, no. 3, pp. 1816–1841, May 2024, doi: 10.1177/14759217231183715;REQUESTEDJOURNAL:JOURNAL:SHMA;WGROU:STRING:PUBLICATION.
- [11] Y. Mahmutoglu and K. Turk, "A passive acoustic based system to locate leak hole in underwater natural gas pipelines," *Digit. Signal Process.*, vol. 76, pp. 59–65, May 2018, doi: 10.1016/J.DSP.2018.02.007.
- [12] W. H. Png, H. S. Lin, C. H. Pua, and F. A. Rahman, "Pipeline monitoring and leak detection using Loop integrated Mach Zehnder Interferometer optical fiber sensor," *Opt. Fiber Technol.*, vol. 46, pp. 221–225, Dec. 2018, doi: 10.1016/J.YOFTE.2018.10.013.
- [13] S. C. Olisa, M. A. Khan, and A. Starr, "Review of Current Guided Wave Ultrasonic Testing (GWUT) Limitations and Future Directions," *Sensors 2021, Vol. 21, Page 811*, vol. 21, no. 3, p. 811, Jan. 2021, doi: 10.3390/S21030811.
- [14] A. Cataldo, E. De Benedetto, L. Angrisani, G. Cannazza, and E. Piuze, "A Microwave Measuring System for Detecting and Localizing Anomalies in Metallic Pipelines," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, doi: 10.1109/TIM.2020.3038491.
- [15] W. H. Png, H. S. Lin, C. H. Pua, and F. A. Rahman, "Pipeline monitoring and leak detection using Loop integrated Mach Zehnder Interferometer optical fiber sensor," *Opt. Fiber Technol.*, vol. 46, pp. 221–225, Dec. 2018, doi: 10.1016/J.YOFTE.2018.10.013.
- [16] J. Wang, L. Ren, R. You, T. Jiang, Z. Jia, and G. xin Wang, "Experimental study of pipeline deformation monitoring using the inverse finite element method based on the iBeam3 element," *Measurement*, vol. 184, p. 109881, Nov. 2021, doi: 10.1016/J.MEASUREMENT.2021.109881.
- [17] H. Yu *et al.*, "An Online Pipeline Structural Health Monitoring Method Based on the Spatial Deformation Fitting," *IEEE Trans. Ind. Electron.*, vol. 69, no. 7, pp. 7383–7393, Jul. 2022, doi: 10.1109/TIE.2021.3101003.
- [18] M. Li and X. Feng, "Multisensor data fusion-based structural health monitoring for buried metallic pipelines under complicated stress states," *J. Civ. Struct. Heal. Monit.* 2022 126, vol. 12, no. 6, pp. 1509–1521, Aug. 2022, doi: 10.1007/S13349-022-00609-W.
- [19] M. El Mountassir, S. Yaacoubi, G. Mourot, and D. Maquin, "Sparse estimation based monitoring method for damage detection and localization: A case of study," *Mech. Syst. Signal Process.*, vol. 112, pp. 61–76, Nov. 2018, doi: 10.1016/J.YMSSP.2018.04.024.
- [20] S. Livadiotis, A. Ebrahimkhanlou, and S. Salamone, "An algebraic reconstruction imaging approach for corrosion damage monitoring of pipelines," *Smart Mater. Struct.*, vol. 28, no. 5, p. 055036, Apr. 2019, doi: 10.1088/1361-665X/AB1160.
- [21] C. A. Chua and P. Cawley, "Crack growth monitoring using fundamental shear horizontal guided waves," *Struct. Heal. Monit.*, vol. 19, no. 5, pp. 1311–1322, Sep. 2020, doi:

10.1177/1475921719882330;SUBPAGE:STRING:ABSTRACT;JOURNAL:JOURNAL:SHMA;WEBSITE:W
EBSITE:SAGE;ISSUE:ISSUE:DOI.

- [22] Y. Xu, M. Luo, Q. Liu, G. Du, and G. Song, "PZT transducer array enabled pipeline defect locating based on time-reversal method and matching pursuit de-noising," *Smart Mater. Struct.*, vol. 28, no. 7, p. 075019, Jun. 2019, doi: 10.1088/1361-665X/AB1CC9.
- [23] Z. Wang, S. Huang, S. Wang, S. Zhuang, Q. Wang, and W. Zhao, "Compressed Sensing Method for Health Monitoring of Pipelines Based on Guided Wave Inspection," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 4722–4731, Jul. 2020, doi: 10.1109/TIM.2019.2951891.
- [24] S. Livadiotis, A. Ebrahimkhanlou, and S. Salamone, "Monitoring internal corrosion in steel pipelines: a two-step helical guided wave approach for localization and quantification," *Struct. Heal. Monit.*, vol. 20, no. 5, pp. 2694–2707, Sep. 2021, doi: 10.1177/1475921720970139;REQUESTEDJOURNAL:JOURNAL:SHMA;WGROU:STRING:PUBLICATION.
- [25] X. W. Ye, T. Jin, C. B. Yun, X. W. Ye, T. Jin, and C. B. Yun, "A review on deep learning-based structural health monitoring of civil infrastructures," *Smart Struct. Syst.*, vol. 24, no. 5, p. 567, Nov. 2019, doi: 10.12989/SSS.2019.24.5.567.
- [26] A. A. Soomro, A. A. Mokhtar, J. C. Kurnia, N. Lashari, H. Lu, and C. Sambo, "Integrity assessment of corroded oil and gas pipelines using machine learning: A systematic review," *Eng. Fail. Anal.*, vol. 131, p. 105810, Jan. 2022, doi: 10.1016/J.ENGFAILANAL.2021.105810.
- [27] S. Lu, J. Feng, H. Zhang, J. Liu, and Z. Wu, "An Estimation Method of Defect Size from MFL Image Using Visual Transformation Convolutional Neural Network," *IEEE Trans. Ind. Informatics*, vol. 15, no. 1, pp. 213–224, Jan. 2019, doi: 10.1109/TII.2018.2828811.
- [28] J. C. P. Cheng and M. Wang, "Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques," *Autom. Constr.*, vol. 95, pp. 155–171, Nov. 2018, doi: 10.1016/J.AUTCON.2018.08.006.
- [29] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, and L. Kurach, "A deep learning-based framework for an automated defect detection system for sewer pipes," *Autom. Constr.*, vol. 109, p. 102967, Jan. 2020, doi: 10.1016/J.AUTCON.2019.102967.
- [30] H. Shukla and K. Piratla, "Leakage detection in water pipelines using supervised classification of acceleration signals," *Autom. Constr.*, vol. 117, p. 103256, Sep. 2020, doi: 10.1016/J.AUTCON.2020.103256.
- [31] M. H. Rafiei and H. Adeli, "A novel unsupervised deep learning model for global and local health condition assessment of structures," *Eng. Struct.*, vol. 156, pp. 598–607, Feb. 2018, doi: 10.1016/J.ENGSTRUCT.2017.10.070.
- [32] D. F. Hesser, S. Mostafavi, G. K. Kocur, and B. Markert, "Identification of acoustic emission sources for structural health monitoring applications based on convolutional neural networks and deep transfer learning," *Neurocomputing*, vol. 453, pp. 1–12, Sep. 2021, doi: 10.1016/j.neucom.2021.04.108.
- [33] M. Zhou *et al.*, "A Pipeline Leak Detection and Localization Approach Based on Ensemble TL1DCNN," *IEEE Access*, vol. 9, pp. 47565–47578, 2021, doi: 10.1109/ACCESS.2021.3068292.
- [34] V. B. Sharma, S. Tewari, S. Biswas, and A. Sharma, "A comprehensive study of techniques utilized for structural health monitoring of oil and gas pipelines," *Struct. Heal. Monit.*, vol. 23, no. 3, pp. 1816–1841, May 2024, doi: 10.1177/14759217231183715;REQUESTEDJOURNAL:JOURNAL:SHMA;WGROU:STRING:PUBLICATION.
- [35] G. Tsialiamanis, D. J. Wagg, P. A. Gardner, N. Dervilis, and K. Worden, "On partitioning of an SHM problem and parallels with transfer learning," *Conf. Proc. Soc. Exp. Mech. Ser.*, pp. 41–50, Mar. 2022, doi: 10.1007/978-3-030-47717-2_5.
- [36] H. Xiao, H. Ogai, and W. Wang, "Multi-Channel Domain Adaptation Deep Transfer Learning for Bridge Structure Damage Diagnosis," *IEEE Trans. Electr. Electron. Eng.*, vol. 17, no. 11, pp. 1637–1647, Nov. 2022, doi: 10.1002/TEE.23671;WGROU:STRING:PUBLICATION.
- [37] J. Jia and Y. Li, "Deep Learning for Structural Health Monitoring: Data, Algorithms, Applications, Challenges, and Trends," *Sensors (Basel)*, vol. 23, no. 21, p. 8824, Oct. 2023, doi: 10.3390/S23218824/S1.
- [38] M. Azimi, A. D. Eslamlou, and G. Pekcan, "Data-Driven Structural Health Monitoring and Damage Detection through Deep Learning: State-of-the-Art Review," *Sensors 2020, Vol. 20, Page 2778*, vol. 20, no. 10, p. 2778, May 2020, doi: 10.3390/S20102778.
- [39] L. Qian *et al.*, "Damage localization method using ultrasonic lamb waves and Wav2Vec2.0 neural network," *Front. Mater.*, vol. 10, no. August, pp. 1–13, 2023, doi: 10.3389/fmats.2023.1212909.
- [40] G. Iglesias, E. Talavera, Á. González-Prieto, A. Mozo, and S. Gómez-Canaval, "Data Augmentation techniques in time series domain: a survey and taxonomy," *Neural Comput. Appl.*, vol. 35, no. 14, pp. 10123–10145, 2023, doi: 10.1007/s00521-023-08459-3.
- [41] C. R. Farrar and K. Worden, "Structural health monitoring : a machine learning perspective," 2013.

- [42] G. Fan, J. Li, and H. Hao, "Vibration signal denoising for structural health monitoring by residual convolutional neural networks," *Measurement*, vol. 157, p. 107651, Jun. 2020, doi: 10.1016/J.MEASUREMENT.2020.107651.
- [43] G. Azuara, M. Ruiz, and E. Barrera, "Damage localization in composite plates using wavelet transform and 2-D convolutional neural networks," *Sensors*, vol. 21, no. 17, 2021, doi: 10.3390/s21175825.
- [44] A. L. Bowler, M. P. Pound, and N. J. Watson, "A review of ultrasonic sensing and machine learning methods to monitor industrial processes," *Ultrasonics*, vol. 124, no. April, p. 106776, 2022, doi: 10.1016/j.ultras.2022.106776.
- [45] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, no. February, p. 100258, 2022, doi: 10.1016/j.array.2022.100258.
- [46] K.-J. Bathe, "Finite element procedures in engineering analysis," p. 735, 1982.
- [47] R. Y. Vidana Morales, S. Ortega Cisneros, J. R. Camacho Perez, F. Sandoval Ibarra, and R. Casas Carrillo, "3D Simulation-Based Acoustic Wave Resonator Analysis and Validation Using Novel Finite Element Method Software," *Sensors 2021, Vol. 21, Page 2715*, vol. 21, no. 8, p. 2715, Apr. 2021, doi: 10.3390/S21082715.
- [48] M. Gresil, A. Poohsai, and N. Chandarana, "Guided Wave Propagation and Damage Detection in Composite Pipes Using Piezoelectric Sensors," *Procedia Eng.*, vol. 188, pp. 148–155, Jan. 2017, doi: 10.1016/J.PROENG.2017.04.468.
- [49] "facebook/wav2vec2-xls-r-300m · Hugging Face," 2019. <https://huggingface.co/facebook/wav2vec2-xls-r-300m> (accessed Nov. 05, 2025).
- [50] M. Almahakeri, A. J. Al-Mahasneh, M. Abu Mallouh, and B. Jouda, "Deep neural network-based intelligent health monitoring system for oil and gas pipelines," *Appl. Soft Comput.*, vol. 171, p. 112827, Mar. 2025, doi: 10.1016/J.ASOC.2025.112827.

Authors' information

1 Mechatronics Engineering Department, Faculty of Engineering, The Hashemite University, Zarqa 13133, Jordan. * Corresponding author, E-mail address: mohammadk_al@hu.edu.jo

2 Mechatronics Engineering Department, Philadelphia University, Amman, 19392, Jordan

3 Mechanical Engineering Department, University of Arkansas, Fayetteville 72701, Arkansas, USA.