applied sciences

MDPI

Article

Real-Time Pipeline Leak Detection: A Hybrid Deep Learning Approach Using Acoustic Emission Signals

Faisal Saleem [1], Zahoor Ahmad [1] and Jong-Myon Kim [1,2,*]

[1] Department of Electrical, Electronics and Computer Engineering, University of Ulsan, Ulsan 44610, Republic of Korea; faisal1999@mail.ulsan.ac.kr (F.S.); zahooruou@mail.ulsan.ac.kr (Z.A.)
[2] PD Technology Co., Ltd., Ulsan 44610, Republic of Korea
* Correspondence: jmkim07@ulsan.ac.kr; Tel.: +82-52-259-2217

Abstract: This study introduces an advanced deep-learning framework for the real-time detection of pipeline leaks in smart city infrastructure. The methodology transforms acoustic emission (AE) signals from the time domain into scalogram images using continuous wavelet transform (CWT) to enhance leak-related features. A Gaussian filter minimizes background noise and clarifies these features further. The core of the framework combines convolutional neural networks (CNNs) with long short-term memory (LSTM), ensuring a comprehensive examination of both spatial and temporal features of AE signals. A genetic algorithm (GA) optimizes the neural network by isolating the most important features for leak detection. The final classification stage uses a fully connected neural network to categorize pipeline health conditions as either 'leak' or 'non-leak'. Experimental validation on real-world pipeline data demonstrated the framework's efficacy, achieving accuracy rates of 99.69%. This approach significantly advances smart city capabilities in pipeline monitoring and maintenance, offering a durable and scalable solution for proactive infrastructure management.

Keywords: acoustic emission; continuous wavelet transform; pipeline leakage; convolutional neural networks; deep learning

check for updates

Academic Editors: Mohsen Besharat and Michal Kubrak

Received: 7 November 2024
Revised: 21 December 2024
Accepted: 22 December 2024
Published: 28 December 2024

Citation: Saleem, F.; Ahmad, Z.; Kim, J.-M. Real-Time Pipeline Leak Detection: A Hybrid Deep Learning Approach Using Acoustic Emission Signals. *Appl. Sci.* **2025**, *15*, 185. https://doi.org/10.3390/app15010185

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Pipelines are one of the most efficient and economical methods of transporting gases and liquids nowadays. While alternatives such as road tankers, rail cars, LNG ships, and compressed gas cylinders exist, they are often constrained by limited capacity, higher operational costs, and increased safety risks. Pipelines, in contrast, offer unmatched efficiency, cost-effectiveness, and safety. They are capable of handling large-scale, continuous transportation over long distances with minimal environmental impact, making them the preferred choice in modern infrastructure for oil and gas transportation [1]. Despite their advantages, pipelines are subjected to leaks caused by corrosion, earthquakes, mechanical cracks, environmental factors and material defects [2]. These leaks can have severe consequences, such as pollution, and public safety hazards, including economic losses and resource wastage [3,4]. It is alarming to note that 46% of pipeline leak incidents globally result in casualties. For instance, a diesel pipeline leak in Guizhou, China, in 2020 resulted in a 1.5 million RMB loss and widespread environmental pollution [5]. Similarly, a petroleum pipeline explosion in Hidalgo, Mexico, caused by a leak, led to over 120 deaths and numerous injuries [6]. These examples underscore the urgent need for effective leak-detection systems [7,8].

Leak detection strategies have evolved over the decades. Traditional methods such as visual inspection, pressure monitoring, and acoustic-based methods were initially employed. However, these techniques were limited by delayed detection times, operator dependency, and the inability to pinpoint leaks in real time. These challenges led to the adoption of advanced approaches utilizing AE signals and machine learning, which now focus on real-time leak detection and precise localization. Real-time leak detection systems continuously monitor pipelines for signals such as pressure fluctuations, acoustic anomalies, or vibration changes [9]. These systems enable immediate detection and corrective actions, significantly reducing the environmental, economic, and safety impacts of leaks. By eliminating delays and ensuring continuous surveillance, real-time systems enhance pipeline reliability and operational efficiency, making them indispensable in modern infrastructure. The pipeline industry now focuses on cost-effective methods for repairing minor leaks, replacing encapsulating collars and clamps rather than replacing entire sections [10,11]. Thus, intelligent leak detection methods are necessary for minimizing maintenance costs [12,13]. Recent advancements in machine learning and artificial intelligence have greatly improved the efficiency and accuracy of pipeline leak detection [14,15]. Techniques such as vibration-based methods, pressure wave techniques, time-domain reflection methods, and AE technology have been developed to ensure pipeline leak detection reliability [16,17]. AE technology is particularly notable for its high sensitivity, easy installation, and real-time leak-detection capabilities [18]. Various studies have demonstrated the potential of AE technology in this domain, and developed the use of AE technology for detecting the beginning of cracks in pipelines [19,20].

Related Research Work

A pipeline leak produces elastic energy, leading to AE events. These occurrences are recorded as AE hits (AEHs) by AE sensors on the pipeline's surface. The variations in the AE signal caused by these AEHs are essential for detecting leaks [21]. Researchers have concentrated on pattern recognition and feature extraction techniques to make effective use of these signal variations [22]. Techniques for detecting pipeline leaks using intelligent pattern recognition can be classified into three main types: frequency domain (FD), temporal domain (TD), and time-frequency domain (TFD). Wang et al. [23] used principal component analysis and pre-processed AE signals to obtain low-dimensional discriminant features before using TD statistical features for leak identification. An artificial neural network (ANN) was applied in conjunction with the amplitude of the TD AE signal to detect leaks [24]. AE signals are complicated and non-stationary; stationary signals are better for FD analysis. Thus, to extract useful features from non-stationary AE data, TFD techniques like wavelet transform and empirical mode decomposition are used [25]. With the use of pattern recognition methods like SVDD, ANN, and fuzzy-SVDD, these features assist in the identification of pipeline health issues. However, TFD preprocessing involves significant computational costs, and choosing the appropriate base wavelet for wavelet transforms requires experimental validation [26].

AEH characteristics have demonstrated good results compared to traditional extracted characteristics from AE signals in FD, TD, and TFD for various approaches [27]. However, AEH features have limitations due to multiple AEH sources, such as fluid pressure, background noise, increased vibrations, and leaks, reducing their sensitivity to leak identification [28]. Extracting leak-related characteristics from the AE signal is the first step in the pipeline leak diagnosis system. DL techniques have an advantage over typical machine learning techniques in their ability to analyze complex data [29]. For pattern recognition tasks, DL techniques independently extract discriminative, meaningful information from complex data. CNN, neural auto-encoders, recurrent neural networks, and deep belief

networks are the most popular DL techniques for defect identification. CNNs use local representative fields, share weights within the network, and use special domain subsampling to minimize the danger of overfitting and enable low computing complexity. Moreover, CNNs have shown promising pattern recognition in pipelines, centrifugal pumps, and bearing failure diagnosis [30,31].

Various studies have demonstrated the potential of AE technology in leak detection and have developed methods for detecting cracks and other anomalies in pipelines. Techniques such as principal component analysis, neural networks, and wavelet transforms have been explored for feature extraction and pattern recognition. Deep learning has gained significant attention for its ability to solve complex problems in various domains. For example, CNN-based transfer learning models have been successfully applied to classify microseismic event waveforms [32,33]. This study highlights the effectiveness of transfer learning in handling data scarcity, which can be adapted to pipeline leak detection scenarios with limited leak data by utilizing pre-trained models. A hybrid deep learning and transfer learning approach for aerosol retrieval [34] showcased the potential of integrating domain-specific preprocessing with deep learning frameworks. Similarly, our work incorporates domain-specific preprocessing CWT scalograms and Gaussian filtering to enhance signal clarity and feature extraction. Attention U-Net architectures for self-potential inversion tasks [35] demonstrated how attention mechanisms can enhance model focus on critical data regions. This inspires potential future extensions of our pipeline monitoring framework to include attention layers for focusing on key time-frequency regions of AE scalograms.

CNNs excel at extracting spatial features from data, but they do not capture temporal dependencies effectively. This limitation affects the ability to fully recognize patterns in uninterrupted data, such as AE signals in pipeline diagnostics. To solve this issue, the CNN network is incorporated with LSTM. LSTMs are specifically designed to capture long-range dependencies and temporal sequences in data, making them optimal for analyzing time-series information. By combining CNN with LSTM, the feature extraction process is enhanced, using CNN for spatial pattern recognition and LSTM for temporal pattern recognition. This hybrid approach enhances the precision and efficiency of leak detection in pipelines, providing a more comprehensive analysis compared to using CNN alone. The latent spaces derived from the combined CNN and LSTM models are further optimized using a GA to accurately assess the health state of the pipeline.

The key contributions of this study are outlined below.

1. A Gaussian filter is used to enhance variations in color intensity due to energy change across different scales and frequencies in AE scalograms.
2. A deep learning framework is introduced, combining enhanced AE scalograms with CNN-LSTM models and a genetic algorithm (GA) for feature optimization, with the goal of improving the identification of pipeline operating conditions.
3. The proposed approach is validated using real pipeline data, showcasing its effectiveness across varying leak scenarios, fluid types, and pressure conditions.

The composition of this work is organized as follows: Section 2 outlines the proposed model and the sequential DL models for leak identification. The experimental setup is discussed in Section 3. Section 4 comprises the results, while Section 5 consists of the discussion. Finally, the conclusion and future directions of this study are described in Section 6.

2. Proposed Methodology

The proposed approach begins with the collection of AE signals from the pipeline, followed by a series of preprocessing, feature extraction, and classification steps, ultimately

leading to an accurate assessment of the pipeline's health condition. This comprehensive workflow ensures that critical signal characteristics are preserved and analyzed effectively. Figure 1 visually represents the step-by-step flow of the proposed methodology, providing a clear overview of the processes involved. The detailed steps are outlined below:

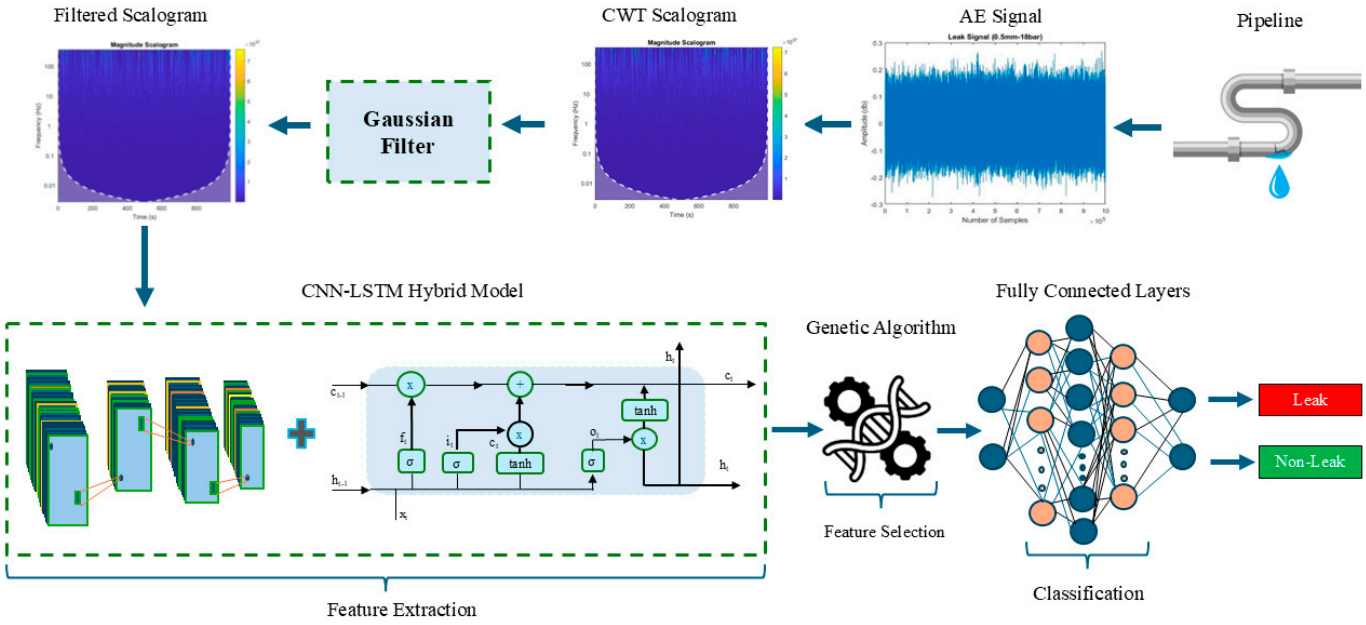


Figure 1. Proposed methodology architecture.

Step I: From the pipeline, the AE signals are collected during both leak and non-leak conditions.

Step II: TD AE signals are converted into images utilizing the CWT. These images use various colors to represent different energy intensities, illustrating how energy levels vary across different time and frequency ranges.

Step III: CWT images are pre-processed using a Gaussian filter to smooth and reduce noise in the scalogram images. This filtering enhances the clarity of the key features, making it easier to identify potential leaks.

Step IV: To extract detailed spatial and temporal features from the enhanced scalograms, a hybrid model combining CNN and LSTM was used. The CNN component effectively captures spatial features, such as changes in energy at specific levels, including variations in AE amplitude. Meanwhile, the LSTM component is adept at capturing temporal dependencies, providing information about the sequence of events in the AE signal, such as AE intensity over time and frequency distribution. These extracted characteristics are then used for further analysis to determine the distinct characteristics of pipeline leaks.

Step V: A feature vector is generated by integrating spatial and temporal features extracted from enhanced scalogram images using a hybrid CNN-LSTM model. These extracted features are collected by a genetic algorithm, which then selects the most important ones. Based on these refined features, pipeline leaks are detected, and the health state of the pipeline is determined using a fully connected layer.

2.1. Continuous Wavelet Transform

CWT is a mathematical tool utilized to examine non-stationary signals by decomposing them into time-frequency components. It provides a representation of the signal at multiple scales and resolutions, making it particularly useful for detecting transient features. A source wavelet function is used by the CWT to transform a TD signal into the time-frequency domain. This source, often known as the "mother" wavelet, is typically a brief time-based

signal that vibrates in cycles. The decomposition process starts with the mother wavelet and divides the complex signal into coefficients that are scaled and translated according to certain parameters [36]. The CWT is collected by summing scaled and shifted copies of the wavelet function across the entire time domain. The CWT of a X (t) signal is defined mathematically as follows:

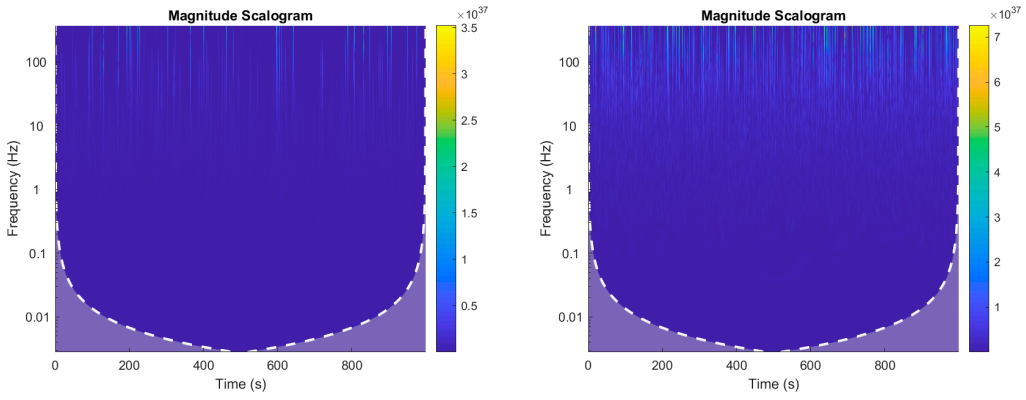
$$CWT_x\,(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} X\,(\mathrm{t}) \cdot \psi^* \cdot \left(\frac{t-\tau}{s}\right) dt \quad (1)$$

where the wavelet's center frequency and window length are determined by s, and its position in the time domain is indicated by τ. Greater detail is revealed by smaller scales, which correspond to higher frequencies, while larger scales provide general information about the signal by matching to lower frequencies.

The Morlet wavelet as "amor" was selected as the basis function for CWT due to its ability to balance time and frequency resolution effectively [37]. Its Gaussian envelope minimizes spectral leakage, which is essential for analyzing transient AE signals. This choice ensures precise localization of leak-related features in the time-frequency domain. The frequency range of interest was set to [1 Hz, 500 kHz] to enclose all relevant spectral components of AE signals. By defining this range, the CWT algorithm dynamically determines the scales, ignoring the need to explicitly set decomposition levels and ensuring optimal representation of the AE signal features.

CWT generates a 2D transformation matrix when it is applied to real AE data from a pipeline. The scale of the AE signal is represented by each row in this matrix, while the translation or size of the pipeline AE signal is represented by each column. These two-dimensional transformation matrices can be seen as what are known as AE scalograms, which are pictures where the color intensities represent the different wavelet energy levels across time and frequency, representing variations in the pipeline's operating conditions.

Different energy regions that correlate to variations in pipeline conditions are clearly seen in the AE images. Figure 2a,b show scalogram images of pipelines working under normal and defective conditions, respectively.



(**a**) (**b**)

Figure 2. CWT scalograms (**a**) normal and (**b**) leak conditions.

2.2. Gaussian Filter

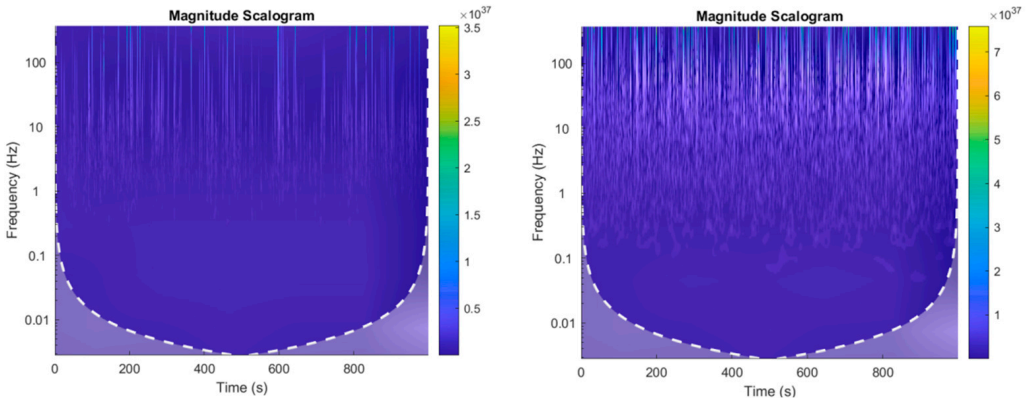
A Gaussian filter uses a Gaussian function, which is defined by its mean and standard deviation and has a bell-like shape. The amount of smoothing applied to the CWT images can be modified by varying the standard deviation [32]. In this study, a Gaussian filter was used to increase the quality of CWT images by minimizing noise and smoothing the images.

The 2D digital Gaussian filter can be mathematically represented as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

where σ represents the variance of the Gaussian filter. The filter kernel size, typically ranging from −1 to 1 for both x and y, is chosen by excluding values less than five percent of the kernel's maximum value [38].

This filter works by blending the CWT image with a Gaussian function, giving more weight to nearby pixel values and less to those farther away. This process effectively minimizes minor, random variations in pixel energy while maintaining the essential features and overall shape of the image. By fine-tuning the standard deviation, the optimal amount of smoothing can be applied, making the images clearer and easier to analyze. Figure 3 shows the CWT scalograms of normal and leak images after applying a Gaussian filter.



(**a**) (**b**)

Figure 3. Filtered CWT scalograms (**a**) normal and (**b**) leak conditions.

This method utilized the Gaussian filter's capability to maintain the key features of CWT images while reducing noise, thereby improving the visual quality and reliability of the images for subsequent processing and analysis. This approach is known for its effectiveness in noise reduction and image enhancement, making it a valuable tool in the initial stages of CWT image analysis.

2.3. CNN and LSTM Hybrid Structure

In this study, a hybrid model that integrates CNN and LSTM networks to extract features from pipeline data was used. This section outlines the individual components of CNN and LSTM and then describes their combined structure, specifically designed for the feature extraction task.

2.3.1. Convolutional Neural Network

CNN is a specialized deep learning model designed primarily for image processing tasks, as illustrated in Figure 4. It efficiently captures spatial hierarchies of features by applying a series of convolutional layers, which use learnable filters to detect local patterns such as edges, textures, and shapes within the input data. This automated feature extraction process allows CNNs to identify intricate relationships in image data, making them highly effective for tasks involving visual pattern recognition and classification. CNN is primarily composed of convolutional, pooling, and activation layers. The central component is the convolutional layer, which performs the convolution operation essentially as an inner product between sections of the input data and a filter matrix. This process is important for extracting feature information from the input data. By utilizing different convolutional

kernels, a variety of features can be extracted, with the size of these kernels playing a significant role in determining the features that are captured. The mathematical representation of the convolution operation is as follows:

$$X_j^{\gamma} = f\left(\sum_{i=0}^{C-1} X_i^{\gamma-1} * K_{ij}^{\gamma} + B_j^{\gamma}\right) \tag{3}$$

where γ denotes the current layer, X_j^{γ} is the j^{th} eigenmatrix of the current layer, f $(\cdot)$ is the activation function, $X_i^{\gamma-1}$ is the data element of the $\gamma - 1$ layer, C is the number of kernels, K_{ij}^{γ} is the weight matrix of the corresponding convolution kernel, and B_j^{γ} is the bias matrix.

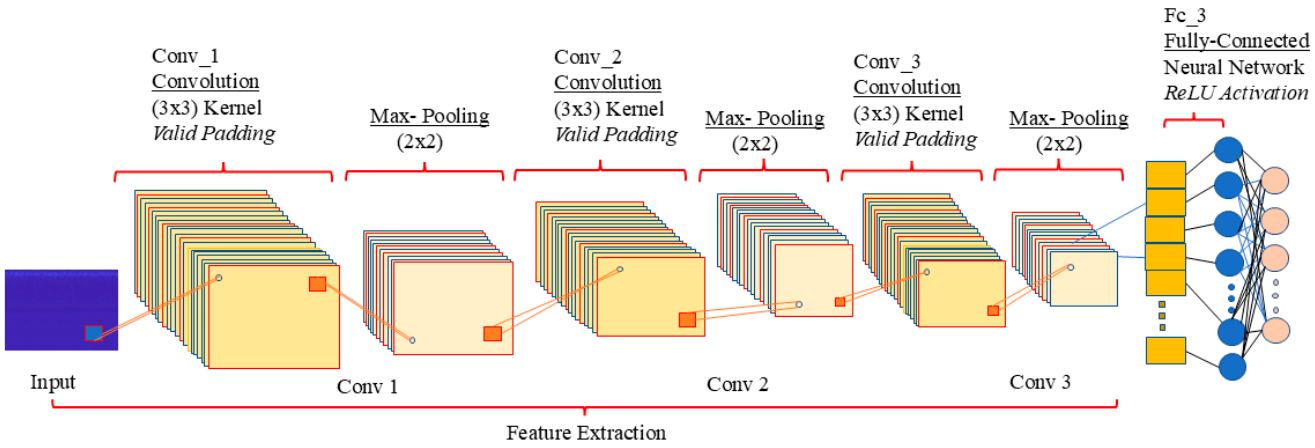


Figure 4. CNN architecture for the proposed model.

The CNN architecture for the proposed model is described below in Table 1. The pooling layer performs downsampling by adjusting the filter on the input data and selecting the maximum or average value within the sliding window. The activation function used in this model is rectified linear unit (ReLU), which is defined as:

$$f(x) = \begin{cases} \max(0, x), & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{4}$$

Table 1. CNN-LSTM hybrid model architecture for pipeline health identification.

Model	Layer Types	No. of Filters	Kernal Size	Output Shape	Activation Function
CNN	input_layer (Input Layer)	-	-	-	-
	Conv2D (block1_conv1)	32	3 × 3	(654, 873, 32)	ReLU
	MaxPooling2D (block1_pool)	-	2 × 2	(327, 436, 32)	-
	Conv2D (block2_conv1)	64	3 × 3	(325, 434, 64)	ReLU
	MaxPooling2D (block2_pool)	-	2 × 2	(162, 217, 64)	-
	Conv2D (block3_conv1)	128	3 × 3	(160, 215, 128)	ReLU
	MaxPooling2D (block3_pool)	-	2 × 2	(80, 107, 128)	-
	Flatten	-	-	(1,095,680)	-
LSTM	Input Layer	-	-	(1, 1,095,680)	-
	lstm	64	-	64	tanh

The ReLU (rectified linear unit) activation function is applied to the output of each convolutional layer in a CNN. It is a non-linear function that allows all positive inputs to pass unchanged while mapping negative inputs to zero. This activation introduces essential non-linearity into the model, enabling it to learn complex patterns and relationships in the data. Additionally, ReLU helps to mitigate the vanishing gradient problem, a common issue with other activation functions like Sigmoid or Tanh, thereby improving the training efficiency and convergence speed of deep neural networks.

Sigmoid:

$$f(x) = \frac{1}{(1 + e^{-x})} \tag{5}$$

Tanh:

$$f(x) = \frac{(e^{x}) - (e^{-x})}{(e^{x}) + (e^{-x})} = \left(\frac{2}{(1 + e^{-2x})}\right) - 1 \tag{6}$$

The CNN architecture was designed with a focus on computational efficiency. Convolutional layers with small kernel sizes (3×3) were used to effectively extract spatial features while minimizing computational costs. Max-pooling layers (2×2) were employed to reduce feature map dimensions progressively, significantly lowering computational overhead. The number of filters (32, 64, and 128) were chosen based on empirical experiments to balance feature extraction capability and computational complexity. Activation functions, ReLU for CNN, were selected for their effectiveness in capturing non-linear relationships and temporal dependencies, respectively. Training was conducted with mini-batch processing, utilizing GPU acceleration and early stopping to optimize training speed.

2.3.2. Long Short-Term Memory

LSTM is a type of RNN, specifically designed to handle and process long-time sequence information. The LSTM architecture for the proposed model is described below in Table 1. Unlike traditional RNNs, LSTM can handle long-term dependencies more effectively by incorporating a gate mechanism and cell state [34]. The network consists of three key gates: the forget gate, input gate, and output gate, which are responsible for controlling the flow of information through the network. These gates work together to manage memory retention, input updates, and output generation, enabling LSTMs to effectively capture long-term dependencies in sequential data. Figure 5 provides a visual representation of this architecture, illustrating how these gates interact to regulate the information flow across different time steps.

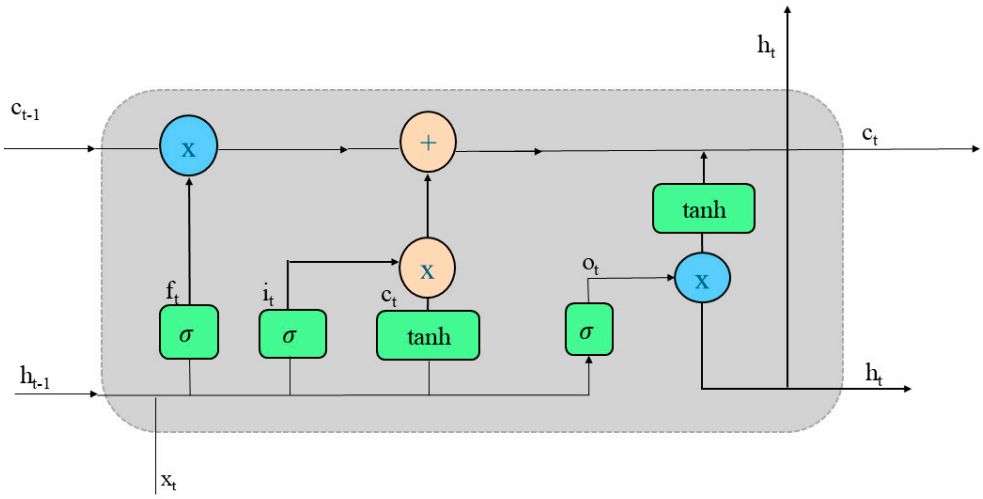


Figure 5. LSTM architecture for the proposed model.

1. The forget gate controls which information from the previous cell state should be discarded or retained. It evaluates the importance of past information using a sigmoid activation function and is mathematically expressed as:

$$f_t = \sigma\left(w_f \cdot [\, h_{t-1}\, ,\, x_t] + b_f\right) \tag{7}$$

where w_f is the weight matrix, b_f is the bias vector, h_{t-1} is the output of the previous unit, and x_t is the current input.

2. The input gate controls which new information is added to the cell state, computed as:

$$i_t = \sigma\left(w_i \cdot [\, h_{t-1}\, ,\, x_t] + b_i\right) \tag{8}$$

$$C_t = tanh\left(w_g \cdot [\, h_{t-1}\, ,\, x_t] + b_g\right) \tag{9}$$

Here, w_i and w_g are weight matrices, b_i and b_i are bias vectors, and C_t represents the candidate cell state.

3. The cell state combines the previous cell state information with the new one. The cell state is updated using:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t \tag{10}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t \tag{11}$$

where C_{t-1} is the previous cell state and $\cdot\cdot$ denotes element-wise multiplication.

4. Output gates determine the next hidden state, and from the cell state, which information should be output, where the output gate is given by:

$$h_t = o_t \cdot tanh\left(C_t\right) \tag{12}$$

where w_0 and b_0 are the weight matrix and bias vector. The hidden state h_t is the output of the LSTM unit.

The CNN-LSTM model processes AE-CWT images through a hybrid architecture that combines spatial feature extraction and temporal pattern recognition. The workflow begins with the CNN component, where the input images are passed through three convolutional layers to extract spatial features. The first convolutional layer applies 32 filters of size 3×3, followed by a 2×2 max-pooling layer, reducing the feature map size from 654×873 to 327×436. In the second convolutional layer, 64 filters of size 3×3 are applied, and another 2×2 max-pooling layer further reduces the feature map to 162×217. The third convolutional layer employs 128 filters of size 3×3, followed by a final 2×2 max-pooling layer, which compresses the output to 80×107. The resulting 3D feature map is then flattened into a 1D vector of size 1,095,680 and passed into an LSTM layer with 64 units. The LSTM layer captures temporal dependencies within the sequential data using a tanh activation function, enabling the model to recognize time-dependent patterns essential for detecting pipeline faults effectively.

This hybrid CNN-LSTM architecture combines the spatial feature extraction capabilities of the CNN with the temporal sequence modeling strengths of the LSTM. This synergy allows the model to capture intricate spatial patterns while maintaining an understanding of time-dependent relationships, making it highly effective for pipeline fault detection and diagnosis.

2.4. Genetic Algorithm

GA is an optimization and search technique inspired by the principles of natural selection and evolution. It mimics biological processes such as reproduction, mutation, crossover, and selection to efficiently explore and optimize complex solution spaces. By iteratively evolving a population of potential solutions, GA identifies optimal or near-optimal outcomes for challenging problems. This evolutionary approach enables GA to handle non-linear, multi-dimensional, and highly constrained optimization tasks, making it a powerful tool for solving complex engineering and computational problems.

The idea of "survival of the fittest", which comes from Charles Darwin's theory of evolution, is the foundation of the GA function. This strategy makes use of the essential processes of crossover, mutation, and selection, all of which are useful for building robustness and accomplishing global optimization [35].

The first step in the GA process is to initialize a population of candidate solutions, with each candidate representing a unique subset of features. The GA assesses each candidate's fitness over several iterative generations by measuring its accuracy against a fully linked model that was trained with those characteristics. In this study, the initial population was set to 20, with generations set to 10. The accuracy of the classification observed on the test dataset acts as the benchmark fitness for evaluating performance throughout the optimization process. The algorithm uses the following steps to evolve the population and improve the solutions continually:

Selection: The GA selects the best-performing candidates from the current population based on their fitness scores. These candidates are more likely to pass on their features to the next generation.

Crossover: Pairs of selected candidates are combined to produce offspring. This crossover operation involves mixing the feature subsets of two parents to create new feature subsets, promoting diversity in the population.

Mutation: To maintain genetic diversity and prevent premature convergence, the GA introduces random changes to some feature subsets. This mutation operation ensures that the search space is thoroughly explored and helps in discovering potentially better feature combinations.

The pipeline health monitoring process is improved by the GA's selection of the most important features, which guarantees that the model is efficient in spotting leaks in the pipeline. The fully connected layers of the model are subsequently trained using the top-performing feature subset from the last generation, improving the precision and dependability of the leak detection procedure.

2.5. Fully Connected Layer

In the fully connected layer, each neuron is directly connected to every neuron in both the preceding and succeeding layers. This layer performs a weighted sum of the input data, followed by the application of an activation function to introduce non-linearity. This non-linear transformation allows the network to learn intricate relationships in the data, making fully connected layers essential for the final classification step.

In the proposed model, the genetic algorithm (GA) selects the most relevant features from the combined output of the CNN-LSTM network, ensuring that only the most informative features are passed into the fully connected layers.

As described in Table 2, the features first pass through a dense layer with ReLU activation. This activation function enables the layer to learn complex and non-linear feature representations, enhancing the model's ability to capture subtle patterns in the data. The output from this layer is then fed into the final fully connected layer, which employs a sigmoid activation function. The sigmoid activation converts the output into probability

scores corresponding to the target classes, enabling the model to make a final classification decision with high accuracy and confidence.

Table 2. Fully connected layer architecture for pipeline health identification.

Layer Types	No. of Filters	Output Shape	Activation Function
Dense	64	64	ReLU
Output Layer	2	2	sigmoid

This architecture ensures an efficient flow of information, combining optimized feature selection from GA with the powerful representation capabilities of dense layers, resulting in robust and reliable fault classification performance.

3. Experimental Setup

Test Setup for Pipeline

The experimental setup and its schematics are presented in Figures 6 and 7. A stainless-steel pipeline, with 6 mm of thickness and 144 mm outer diameter, was fitted with sensors R15I-AST from Mistras Group, Inc., West Windsor Township, NJ, USA, to simulate pipeline leaks. The data acquisition system of National Instruments, model NI-9223, was used to collect AE signals at a 1 MHz sample rate; detailed specifications of the test setup are presented in Table 3. An electric drill was used to drill holes, simulating leaks of different sizes. For controlling the fluid flow, a fluid control valve was welded at each hole. Water was selected for the experiment because it is safe and environmentally friendly due to its non-hazardous and non-toxic properties. The parameter settings directly influence the quality of captured AE signals and, consequently, the final analysis results. Deviations in frequency range, resonant frequency, or placement could lead to a reduction in model accuracy by affecting the clarity and reliability of the scalograms used for classification.

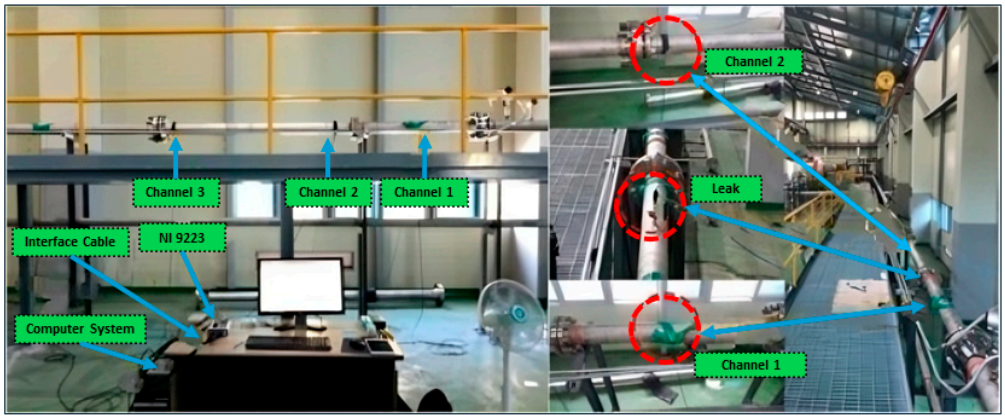


Figure 6. Experimental setup for the pipeline leak detection.

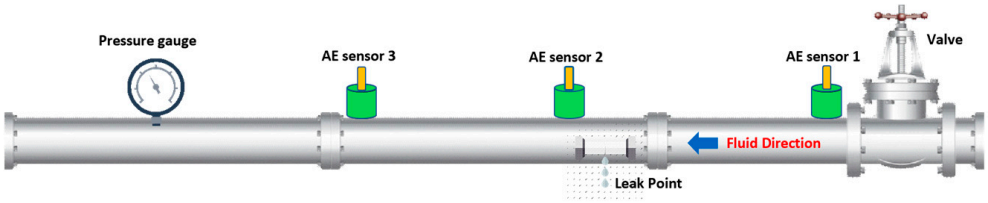


Figure 7. Pipeline architecture for the experiment.

Table 3. R151-AST specifications.

No.	Elements	Value
1	Peak Sensitivity, ref [V/(m/s)]	109 [dB]
2	Peak Sensitivity, ref [V/μbar]	22 [dB]
3	Frequency Operating Range	50–400 [kHz]
4	Resonance Frequency, ref [V/(m/s)]	75 [kHz]
5	Resonance Frequency, ref [V/mbar]	150 [kHz]
6	Signal Directionality	±1.5 [dB]
7	Operational Temperature Range	35 to 75 [°C]
8	Pipeline Wall Thickness	6.02 mm
9	Pipeline Material Composition	304 stainless steels
10	Pipeline Outer Diameter	114.3 mm

The sensitivity and frequency range of the sensors significantly affected the data quality. Sensors with a higher sensitivity, e.g., 75 kHz resonance frequency, extracted AE signals more effectively, improving the scalograms clarity and classification performance. Additionally, sensor placement closer to potential leak sources resulted in better signal strength, contributing to higher model accuracy.

Initial measurements were conducted with the valve closed for one minute to establish baseline conditions. Following this, the valve was opened to simulate a 1 mm hole in the pipeline, and measurements were recorded for an additional two minutes. Further data collection was performed with the valve closed at a pressure of 18 bars for one minute. Subsequently, a 0.5 mm leak was intentionally introduced, and data were collected for two minutes with the valve open. This process was repeated for leak sizes of 0.7 mm and 1 mm at a reduced pressure of 7 bars. For each fluid pressure level, a total of 360 samples were collected: 120 samples during normal operation (no leaks) and 240 samples under leakage conditions, ensuring balanced representation across conditions. This procedure generated 360 samples per pressure condition, capturing AE signal variations corresponding to both normal and faulty states.

The AE signals recorded during normal operation and leakage conditions are illustrated in Figures 8 and 9, offering visual insights into signal differences between these states. A detailed summary of the collected data, including variations across normal and leak states under different pressure levels, is presented in Table 4. This dataset provides a comprehensive foundation for fault detection and classification analysis in pipeline monitoring systems.

Table 4. Data set description.

Data Set	Pressure of Fluid (Bars)	Leak Size (mm)	Time (s)	Number of Samples	
				Non-Leak (Normal)	Leak
Water	13	1.0	360	120	240
Gas	18	1.0	360	120	240
Water	7	0.7	360	120	240
Gas	7	0.5	360	120	240

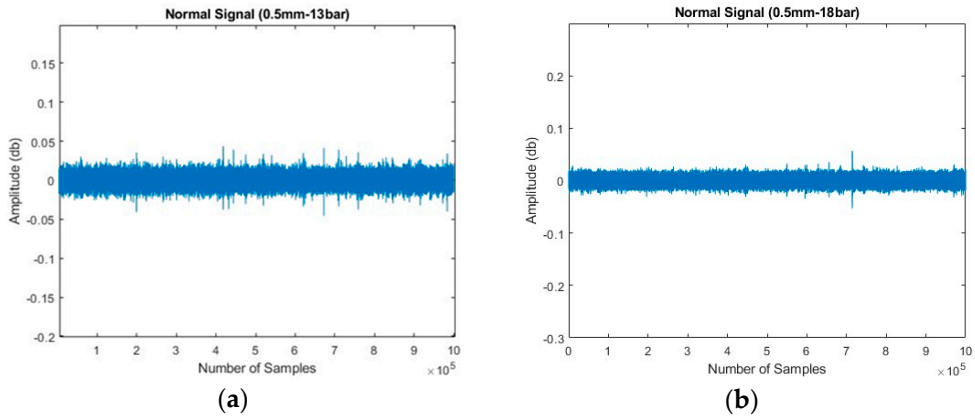


Figure 8. Non-leak AE signal (**a**) 13 bar (**b**) 18 bar.

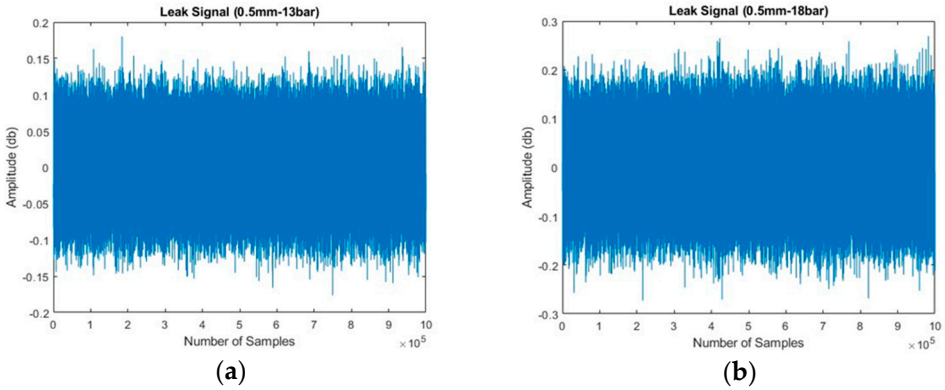


Figure 9. Leak AE signal (**a**) 13 bar (**b**) 18 bar.

4. Results

The arrangement of training and validation data plays a crucial role in evaluating the effectiveness of the proposed method. During the training phase, data corresponding to a leak size of 1 mm under fluid pressures of 13 and 18 bars were utilized, while the evaluation phase employed data from varying pressure levels and leak sizes to ensure a comprehensive assessment. The dataset comprised 1080 samples, evenly divided between non-leak samples and leak samples. To construct and validate the model, 80% of the dataset was randomly allocated for training, while the remaining 20% was reserved for validation. Class imbalance was addressed by adjusting the loss function to assign higher importance to the minority class and using stratified k-fold cross-validation to ensure balanced splits during training and testing phases.

To maintain consistency and reliability, the experiments were repeated 10 times. The model's convergence was examined by varying the number of training epochs (50, 100, and 150 epochs), and it was observed that optimal accuracy was achieved between 70 and 100 epochs. Early stopping was applied to prevent overfitting by halting training if the validation loss did not improve for 10 consecutive epochs. Additionally, a dropout rate of 0.5 was introduced in the fully connected layers to prevent over-reliance on specific features, and training was stopped once validation loss plateaued. Furthermore, a 5-fold cross-validation strategy was implemented to validate the model's robustness across different data subsets. These combined strategies ensured a balanced, efficient, and reliable training process, resulting in a robust and generalizable fault diagnosis model suitable for real-world applications.

Metrics like precision, accuracy, recall, and F1 score were employed to evaluate the efficacy of the suggested method. These metrics give an accurate measure of the classifi-

cation algorithm's efficiency and data classification accuracy. Equations (13)–(16) are the specific formulas that were used to calculate these measurements.

$$Precision = \frac{(\sum_{\alpha}^{A} n_{\alpha}) * \left(\frac{T.P_{\alpha}}{T.P_{\alpha} + F.P_{\alpha}} \right)}{N} \tag{13}$$

$$Recall = \frac{(\sum_{\alpha}^{A} n_{\alpha}) * \left(\frac{T.P_{\alpha}}{T.P_{\alpha} + F.N_{\alpha}} \right)}{N} \tag{14}$$

$$F1 = \frac{1}{N} \sum_{\alpha}^{A} n_{\alpha} * 2 * \sum_{\alpha}^{A} \left(\frac{Recall_{\alpha} * Prcision_{\alpha}}{Recall_{\alpha} + Precision_{\alpha}} \right) \tag{15}$$

$$Accuracy = \frac{(\sum_{\alpha}^{A} T.P_{\alpha})}{N} \tag{16}$$

where '$\boldsymbol{FP_{\alpha}}$', '$\boldsymbol{FN_{\alpha}}$', and '$\boldsymbol{TP_{\alpha}}$' represent false positive, false negative, and true positive outcomes for class A, respectively. A false positive ($\boldsymbol{FP_{\alpha}}$) occurs when a sample is incorrectly classified as belonging to class A when it does not. A true positive ($\boldsymbol{TP_{\alpha}}$) indicates the correct identification of samples that genuinely belong to class A. On the other hand, a false negative ($\boldsymbol{FN_{\alpha}}$) happens when samples that actually belong to class A are misclassified as belonging to another class.

The total number of samples in class A is represented by the sum of $\boldsymbol{TP_{\alpha}}$ and $\boldsymbol{FN_{\alpha}}$ denoted as $\boldsymbol{n_{\alpha}}$. The total number of samples misclassified as belonging to class A is the sum of $\boldsymbol{FP_{\alpha}}$ and the difference between the total number of data samples (N) and $\boldsymbol{n_{\alpha}}$. Here, N signifies the total number of data samples in the testing dataset.

To evaluate the effectiveness of the proposed approach, a comparison was conducted with three other models designed for similar tasks. The first model (CWT-CNN), developed by Li et al. [39], employs a CNN trained on CWT images extracted from AE signals under a similar experimental setup. The second model (FFT-CNN), proposed by Masoumeh Rahimi et al. [40], utilizes FFT images as input to a CNN for feature extraction and classification. The third model (STFT-CNN) integrates STFT images with a CNN, leveraging time-frequency representations for fault detection. These models represent diverse TFD approaches combined with CNN architectures, providing a benchmark to assess the performance and robustness of the proposed method across multiple preprocessing techniques and feature extraction strategies

5. Discussion

The proposed CNN-LSTM hybrid model, applied to AE data from industrial fluid pipelines, demonstrated outstanding performance, achieving precision, accuracy, F1 score, and recall values of 99.71%, 99.69%, 99.82%, and 99.75%, respectively, as presented in Table 5. These results highlight the model's superiority over reference models, including CWT-CNN, FFT-CNN, and STFT-CNN, in terms of classification accuracy. The model's enhanced performance stems from its ability to integrate spatial and temporal features effectively. The CNN component extracts spatial features from enhanced CWT scalograms, capturing intricate energy variations in the AE signals. Meanwhile, the LSTM component excels in modeling temporal dependencies, identifying meaningful sequential patterns within the data. Additionally, the inclusion of a genetic algorithm (GA) refines the extracted features, ensuring that only the most relevant and discriminative features are selected for classification. This hybrid architecture enables a seamless combination of spatial and temporal feature extraction, significantly enhancing the model's capacity to differentiate

fault conditions accurately. The integration of these advanced techniques contributes to the model's robust performance across various experimental conditions and performance metrics, establishing it as a highly reliable solution for fault detection and classification in industrial pipeline systems.

Table 5. Comparison metrics of the proposed model with the reference models.

Models	Accuracy	Precision	F1 Score	Recall
Proposed	99.69	99.71	99.82	99.75
Li et al. [39]	85.18	90.72	70.08	72.15
Rahimi et al. [40]	92.05	97.41	78.91	88.35
STFT-CNN	93.05	98.01	79.58	89.30

Li et al. [39] collected acoustic signals from a gas pipeline system with small, synthetic leaks to apply deep learning techniques for leak detection. Their approach involved introducing controlled artificial leaks, which exposed the system to diverse acoustic signatures under predefined conditions. This deliberately created input was important for training the model to recognize patterns associated with small leaks, simulating real-world conditions effectively. The system utilized these inputs to iteratively optimize its performance, refining its robustness and adaptability by addressing edge cases through expert feedback. By transforming the acoustic signals into the frequency domain and applying a 1D CNN model, the methodology was able to develop discriminative features for small-leak detection. However, noise in the acoustic signals introduced challenges that impacted performance, resulting in an accuracy of 85.18%. Despite these limitations, the approach demonstrated its effectiveness and was selected for comparison due to its compatibility with our experimental setup. The performance metrics in Table 5 reflect the application of this reference technique to our dataset.

Masoumeh Rahimi et al. [40] employed a DL approach for leak detection by collecting data using a hydrophone from a leaking plastic tank. Their study systematically compared multiple signal preprocessing techniques across the frequency domain, time domain, and time-frequency domain, with each preprocessed signal subsequently analyzed using a convolutional neural network (CNN) for feature extraction and classification. The study revealed that the FFT-CNN approach outperformed other preprocessing methods in effectively detecting leaks. To ensure a fair and consistent comparison, the same methodology was applied to our pipeline dataset, where signals underwent similar preprocessing across the time, frequency, and time-frequency domains before being processed by a CNN model. The results obtained from our dataset were carefully recorded and analysed, allowing for a direct comparison with ABC's findings to evaluate the relative performance of each method in the context of pipeline fault detection.

The proposed model was compared with a TFD method, specifically the STFT-CNN. In this approach, AE signals from the TD are transformed into the TFD using the STFT. The resulting representations are fed into a CNN, which is trained to extract features indicating leaks. This method utilizes the CNN's pattern recognition capabilities combined with STFT for feature extraction. For a fair comparison, the same dataset was used, and the STFT-CNN achieved an accuracy of 93.05%. The lower accuracy is primarily due to information loss caused by the windowing effect in STFT, which reduces its ability to capture transient signal variations accurately. This limitation affects the TFD resolution, resulting in decreased fault classification performance compared to the proposed model.

In comparison to the methods previously stated, the results obtained using the proposed method show a higher classification accuracy. Figures 10 and 11 show the confusion

matrices and t-SNE visualizations to demonstrate this superiority, which is mainly attributable to the method's improved consistency and precision in identifying leak statuses as well as pipeline normal operating conditions.

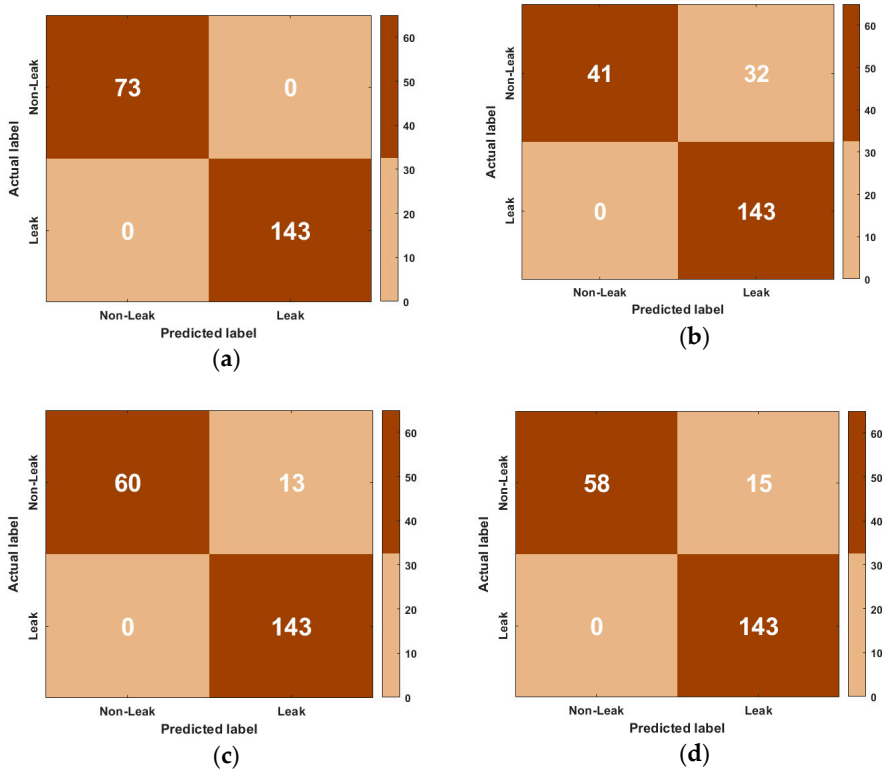


Figure 10. Confusion matrix of (**a**) proposed model, (**b**) CWT-CNN, (**c**) FFT-CNN, (**d**) STFT-CNN.

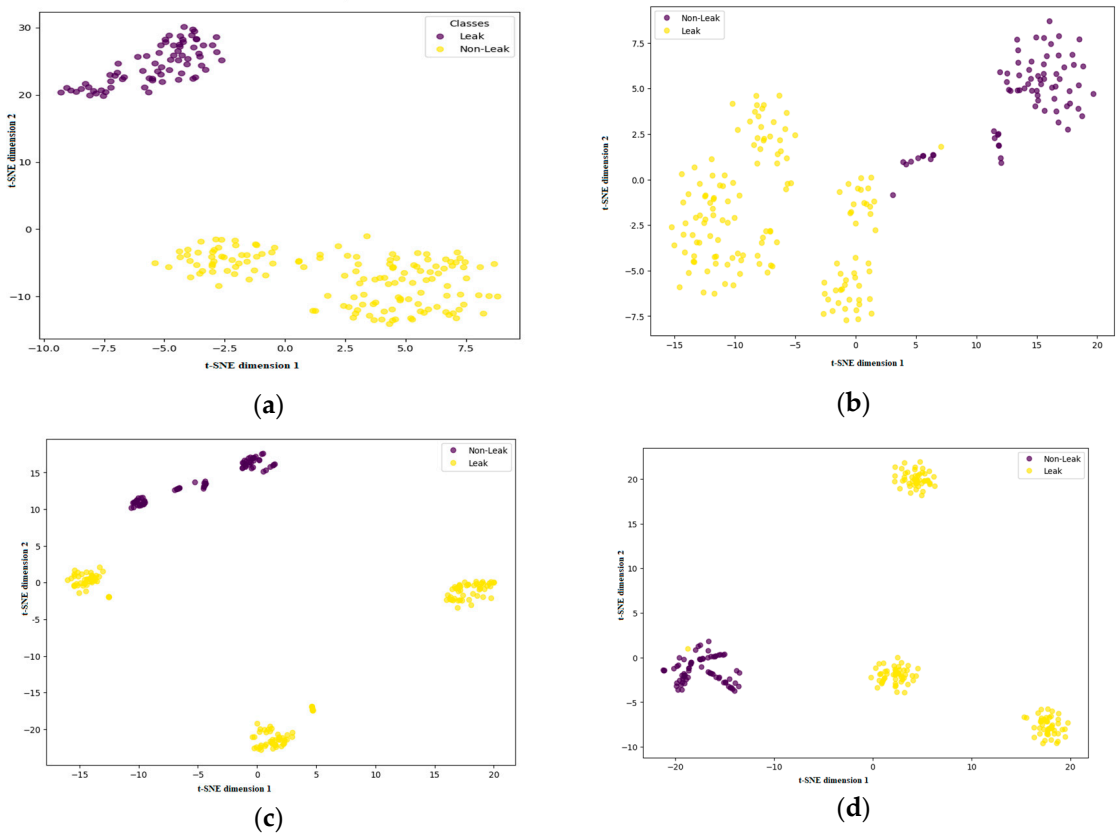


Figure 11. t-SNE plot of (**a**) proposed model, (**b**) CWT-CNN, (**c**) FFT-CNN, (**d**) STFT-CNN.

6. Conclusions

This study introduced an innovative approach for pipeline leak detection using advanced deep learning (DL) techniques. AE signals were collected from a pipeline system and transformed into CWT images to capture essential time-frequency features. A hybrid DL framework, integrating CNN and LSTM models, was developed to extract both spatial and temporal features effectively from these images. To further enhance feature relevance and classification accuracy, a Genetic Algorithm (GA) was employed for feature selection, ensuring that only the most discriminative features were retained. These optimized features were then fed into a fully connected layer for pipeline health classification. The proposed method demonstrated outstanding performance, achieving an impressive accuracy of 99.69% in leak detection. This highlights its robustness, reliability, and superiority over traditional approaches. The scientific significance of this research lies in its seamless integration of AE signal processing, time-frequency analysis, and DL techniques, resulting in a highly accurate and scalable solution for pipeline leak detection. Furthermore, the approach holds significant practical value, particularly for industries relying on pipelines as critical infrastructure, offering a reliable and efficient tool for real-time monitoring and maintenance.

Future work will address precise leak localization by developing methods such as accurately extracting leak-related AE events and implementing time difference of arrival techniques. Building on this, the integration of hydraulic behavior analysis will be explored, incorporating physical models like Bernoulli's principle and pressure-loss equations. These advancements will enable a comprehensive framework that not only detects leaks with high accuracy but also localizes them precisely while considering the hydraulic dynamics of pipeline systems. Together, these developments aim to enhance operational safety, reduce environmental and economic impacts, and contribute to a broader understanding of pipeline health monitoring.

Author Contributions: Conceptualization, F.S., Z.A. and J.-M.K.; methodology, F.S., Z.A. and J.-M.K.; validation, F.S., Z.A. and J.-M.K.; formal analysis, F.S., Z.A. and J.-M.K.; resources, F.S., Z.A. and J.-M.K.; writing—original draft preparation, F.S., Z.A. and J.-M.K.; writing—review and editing, J.-M.K.; visualization, F.S., Z.A. and J.-M.K.; project administration, J.-M.K.; funding acquisition, J.-M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korean government (MOTIE) ('RS-2023-00232515', 'Development of life prediction safety technology and hydrogen embrittlement estimation of LNG pipe mixed hydrogen'). This work was also supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korean government (MOTIE) ('RS-2024-00449107', 'Development of Flexible Pipe and Connector for Hydrogen Gas').

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Author Jong-Myon Kim was employed by the company PD Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Nomenclature

AE	Acoustic emission
CWT	continuous wavelet transform
CNN	Convolutional neural network
LSTM	Long short-term memory
ANN	Artificial neural network

GA	Genetic algorithm
TD	Time domain
FD	Frequency domain
TFD	Time-frequency domain
STFT	Short-time Fourier transform
FFT	Fast Fourier transform
DL	Deep learning
ReLU	Rectified neural network

References

1. Latif, J.; Shakir, M.Z.; Edwards, N.; Jaszczykowski, M.; Ramzan, N.; Edwards, V. Review on condition monitoring techniques for water pipelines. *Measurement* **2022**, *193*, 110895. [CrossRef]
2. Zhang, Z.; Zhang, L.; Fu, M.; Ozevin, D.; Yuan, H. Study on leak localization for buried gas pipelines based on an acoustic method. *Tunn. Undergr. Space Technol.* **2022**, *120*, 104247. [CrossRef]
3. Rahman, I.U.; Mohammed, H.J.; Siddique, M.F.; Ullah, M.; Bamasag, A.; Alqahtani, T.; Algarni, S. Application of membrane technology in the treatment of waste liquid containing radioactive materials. *J. Radioanal. Nucl. Chem.* **2023**, *332*, 4363–4376. [CrossRef]
4. Che, T.-C.; Duan, H.-F.; Lee, P.J. Transient wave-based methods for anomaly detection in fluid pipes: A review. *Mech. Syst. Signal Process.* **2021**, *160*, 107874. [CrossRef]
5. Rai, A.; Ahmad, Z.; Hasan, M.J.; Kim, J.-M. A Novel Pipeline Leak Detection Technique Based on Acoustic Emission Features and Two-Sample Kolmogorov–Smirnov Test. *Sensors* **2021**, *21*, 8247. [CrossRef] [PubMed]
6. Xing, J.; Meng, H.; Meng, X. An urban pipeline accident model based on system engineering and game theory. *J. Loss Prev. Process Ind.* **2020**, *64*, 104062. [CrossRef]
7. Miao, X.; Zhao, H.; Xiang, Z. Leakage detection in natural gas pipeline based on unsupervised learning and stress perception. *Process Saf. Environ. Prot.* **2023**, *170*, 76–88. [CrossRef]
8. Xu, T.; Zeng, Z.; Huang, X.; Li, J.; Feng, H. Pipeline leak detection based on variational mode decomposition and support vector machine using an interior spherical detector. *Process Saf. Environ. Prot.* **2021**, *153*, 167–177. [CrossRef]
9. Siddique, M.F.; Ahmad, Z.; Ullah, N.; Ullah, S.; Kim, J.-M. Pipeline Leak Detection: A Comprehensive Deep Learning Model Using CWT Image Analysis and an Optimized DBN-GA-LSSVM Framework. *Sensors* **2024**, *24*, 4009. [CrossRef] [PubMed]
10. Xiao, R.; Hu, Q.; Li, J. A model-based health indicator for leak detection in gas pipeline systems. *Measurement* **2021**, *171*, 108843. [CrossRef]
11. Li, S.; Song, Y.; Zhou, G. Leak detection of water distribution pipeline subject to failure of socket joint based on acoustic emission and pattern recognition. *Measurement* **2018**, *115*, 39–44. [CrossRef]
12. Park, S.; Yeo, D.; Bae, J.H. Unsupervised Learning–Based Plant Pipeline Leak Detection Using Frequency Spectrum Feature Extraction and Transfer Learning. *IEEE Access* **2024**, *12*, 88939–88949. [CrossRef]
13. Siddique, M.F.; Ahmad, Z.; Kim, J.-M. Pipeline leak diagnosis based on leak-augmented scalograms and deep learning. *Eng. Appl. Comput. Fluid Mech.* **2023**, *17*, 2225577. [CrossRef]
14. Banjara, N.K.; Sasmal, S.; Voggu, S. Machine learning supported acoustic emission technique for leakage detection in pipelines. *Int. J. Press. Vessel. Pip.* **2020**, *188*, 104243. [CrossRef]
15. Xu, C.; Du, S.; Gong, P.; Li, Z.; Chen, G.; Song, G. An Improved Method for Pipeline Leakage Localization with a Single Sensor Based on Modal Acoustic Emission and Empirical Mode Decomposition with Hilbert Transform. *IEEE Sens. J.* **2020**, *20*, 5480–5491. [CrossRef]
16. Ullah, N.; Siddique, M.F.; Ullah, S.; Ahmad, Z.; Kim, J.-M. Pipeline Leak Detection System for a Smart City: Leveraging Acoustic Emission Sensing and Sequential Deep Learning. *Smart Cities* **2024**, *7*, 2318–2338. [CrossRef]
17. Rojek, I.; Studzinski, J. Detection and Localization of Water Leaks in Water Nets Supported by an ICT System with Artificial Intelligence Methods as a Way Forward for Smart Cities. *Sustainability* **2019**, *11*, 518. [CrossRef]
18. Elforjani, M.; Mba, D. Detecting natural crack initiation and growth in slow speed shafts with the Acoustic Emission technology. *Eng. Fail. Anal.* **2009**, *16*, 2121–2129. [CrossRef]
19. Kim, H.; Lee, J.; Kim, T.; Park, S.J.; Kim, H.; Jung, I.D. Advanced Thermal Fluid Leakage Detection System with Machine Learning Algorithm for Pipe-in-Pipe Structure. *Case Stud. Therm. Eng.* **2023**, *42*, 102747. [CrossRef]
20. Ali, H.; Choi, J. A Review of Underground Pipeline Leakage and Sinkhole Monitoring Methods Based on Wireless Sensor Networking. *Sustainability* **2019**, *11*, 4007. [CrossRef]
21. Wang, W.; Mao, X.; Liang, H.; Yang, D.; Zhang, J.; Liu, S. Experimental research on in-pipe leaks detection of acoustic signature in gas pipelines based on the artificial neural network. *Measurement* **2021**, *183*, 109875. [CrossRef]

22. Lee, S.; Kim, B. Machine Learning Model for Leak Detection Using Water Pipeline Vibration Sensor. *Sensors* **2023**, *23*, 8935. [CrossRef] [PubMed]
23. Wang, F.; Lin, W.; Liu, Z.; Wu, S.; Qiu, X. Pipeline Leak Detection by Using Time-Domain Statistical Features. *IEEE Sens. J.* **2017**, *17*, 6431–6442. [CrossRef]
24. Wang, W.; Sun, H.; Guo, J.; Lao, L.; Wu, S.; Zhang, J. Experimental study on water pipeline leak using In-Pipe acoustic signal analysis and artificial neural network prediction. *Measurement* **2021**, *186*, 110094. [CrossRef]
25. Esu, O.E.; Wang, Y.; Chryssanthopoulos, M.K. A baseline-free method for damage identification in pipes from local vibration mode pair frequencies. *Struct. Health Monit.* **2022**, *21*, 2152–2189. [CrossRef]
26. Shukla, H.; Piratla, K. Leakage detection in water pipelines using supervised classification of acceleration signals. *Autom. Constr.* **2020**, *117*, 103256. [CrossRef]
27. Huang, L.; Hong, X.; Yang, Z.; Liu, Y.; Zhang, B. CNN-LSTM network-based damage detection approach for copper pipeline using laser ultrasonic scanning. *Ultrasonics* **2022**, *121*, 106685. [CrossRef]
28. Shang, L.; Zhang, Z.; Tang, F.; Cao, Q.; Pan, H.; Lin, Z. CNN-LSTM Hybrid Model to Promote Signal Processing of Ultrasonic Guided Lamb Waves for Damage Detection in Metallic Pipelines. *Sensors* **2023**, *23*, 7059. [CrossRef] [PubMed]
29. Spandonidis, C.; Theodoropoulos, P.; Giannopoulos, F. A Combined Semi-Supervised Deep Learning Method for Oil Leak Detection in Pipelines Using IIoT at the Edge. *Sensors* **2022**, *22*, 4105. [CrossRef] [PubMed]
30. Chen, C.-C.; Liu, Z.; Yang, G.; Wu, C.-C.; Ye, Q. An Improved Fault Diagnosis Using 1D-Convolutional Neural Network Model. *Electronics* **2020**, *10*, 59. [CrossRef]
31. Hasan, M.J.; Rai, A.; Ahmad, Z.; Kim, J.-M. A Fault Diagnosis Framework for Centrifugal Pumps by Scalogram-Based Imaging and Deep Learning. *IEEE Access* **2021**, *9*, 58052–58066. [CrossRef]
32. Dong, L.; Shu, H.; Tang, Z.; Yan, X. Microseismic event waveform classification using CNN-based transfer learning models. *Int. J. Min. Sci. Technol.* **2023**, *33*, 1203–1216. [CrossRef]
33. Li, W.; Huang, R.; Li, J.; Liao, Y.; Chen, Z.; He, G.; Yan, R.; Gryllias, K. A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges. *Mech. Syst. Signal Process.* **2022**, *167*, 108487. [CrossRef]
34. Fu, D.; Shi, H.; Gueymard, C.A.; Yang, D.; Zheng, Y.; Che, H.; Fan, X.; Han, X.; Gao, L.; Bian, J.; et al. A Deep-Learning and Transfer-Learning Hybrid Aerosol Retrieval Algorithm for FY4-AGRI: Development and Verification over Asia. *Engineering* **2024**, *38*, 164–174. [CrossRef]
35. Guo, Y.; Cui, Y.; Chen, H.; Xie, J.; Zhang, C.; Liu, J. Self-potential inversion based on Attention U-Net deep learning network. *J. Cent. South Univ.* **2024**, *31*, 3156–3167. [CrossRef]
36. Nitin; Gupta, S.B. A Hybrid Image Denoising Method Based on Discrete Wavelet Transformation with Pre-Gaussian Filtering. *Indian J. Sci. Technol.* **2022**, *15*, 2317–2324. [CrossRef]
37. Büssow, R. An algorithm for the continuous Morlet wavelet transform. *Mech. Syst. Signal Process.* **2007**, *21*, 2970–2979. [CrossRef]
38. Deng, G.; Cahill, L.W. An adaptive Gaussian filter for noise reduction and edge detection. In Proceedings of the 1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference, San Francisco, CA, USA, 31 October–6 November 1993; pp. 1615–1619. [CrossRef]
39. Li, J.; Liu, Y.; Chai, Y.; He, H.; Gao, M. A Small Leakage Detection Approach for Gas Pipelines based on CNN. In Proceedings of the 2019 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS), Xiamen, China, 5–7 July 2019; pp. 390–394. [CrossRef]
40. Rahimi, M.; Alghassi, A.; Ahsan, M.; Haider, J. Deep Learning Model for Industrial Leakage Detection Using Acoustic Emission Signal. *Informatics* **2020**, *7*, 49. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.