energies

MDPI

Article

Gas Pipeline Leak Detection by Integrating Dynamic Modeling and Machine Learning Under the Transient State

Juhyun Kim [1], Sunlee Han [1], Daehee Kim [2] and Youngsoo Lee [1,3,*]

[1] Department of Mineral Resource and Energy Engineering, Jeonbuk National University, Jeonju 54896, Republic of Korea; slhan@jbnu.ac.kr (S.H.)
[2] Korea CCUS Association, Sejong 30103, Republic of Korea; dhkim@kccus.kr
[3] Department of Environment and Energy, Jeonbuk National University, Jeonju 54896, Republic of Korea
* Correspondence: youngsoo.lee@jbnu.ac.kr; Tel.: +82-63-270-2392

Abstract: This study focused on developing machine learning models to detect leak size and location in transient state conditions. The model was designed for an onshore methane–hydrogen blending gas pipeline in Canada. Base case simulations revealed significant effects on mass flow and pressure due to leaks, with the system taking approximately 6 h to reach a steady state from transient conditions. This made it essential to analyze the flow characteristics during the transient state. Trend data from the pipeline's inlet and outlet were examined, considering the leak size and location. To better represent the data over time, a method was used to create two-dimensional images, which were then fed into a CNN (convolutional neural network) for training. The model's accuracy was assessed using classification accuracy and a confusion matrix. By refining the data acquisition process and implementing targeted screening procedures, the model's classification accuracy increased to over 80%. In conclusion, this study demonstrates that machine learning can enable rapid and accurate leak detection in transient state conditions. The findings are expected to complement existing leak detection methods and support operators in making faster, more informed decisions.

Keywords: pipeline flow simulation; convolution neural network; continuous wavelet transform; leak size detection; leak location detection

check for updates

Citation: Kim, J.; Han, S.; Kim, D.; Lee, Y. Gas Pipeline Leak Detection by Integrating Dynamic Modeling and Machine Learning Under the Transient State. *Energies* **2024**, *17*, 5517. https://doi.org/10.3390/en17215517

Academic Editor: Marcin Kaminski

Received: 8 October 2024
Revised: 29 October 2024
Accepted: 1 November 2024
Published: 4 November 2024

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 remain the most efficient and economical means of transporting gas across vast distances, making their integrity critical to energy infrastructure. In the case of hydrogen, transport technology is a critical issue in the industry, with hydrogen–natural gas blending emerging as a stable and efficient method. This approach mixes natural gas and hydrogen at specific ratios through existing natural gas pipelines. Several studies, including those by Capelle et al. [1] and Messaoudani et al. [2], have suggested blending as a low-cost, highly efficient transport solution. Chae et al. [3] further investigated global trends in hydrogen blending research. Currently, pilot projects for hydrogen–natural gas blending are underway in Europe and North America. Several countries have begun supplying natural gas blended with 10% hydrogen, with future plans to increase this ratio to 20% while maintaining pipeline integrity [4]. As a result, pipelines will inevitably continue to be used for gas transport.

However, various issues such as material defects, corrosion, abrasion, and environmental factors can compromise pipeline integrity, leading to leaks over time. Hydrogen, in particular, has a higher risk of leaks due to its corrosive and embrittling properties. If a leak occurs, it can cause significant human, material, and environmental damage. Moreover, hydrogen–air mixtures are extremely flammable, requiring only 0.017 mJ of ignition energy compared to 0.25 mJ for hydrocarbons [5]. Even small amounts of hydrogen (10–20%) added to natural gas demand extra caution, as hydrogen is highly reactive and supplied at high pressure [6,7].

Pipeline leaks are among the most frequent accidents in the gas industry and happen more often than expected [8]. Despite various efforts, detecting leaks remains a significant challenge. Numerous studies have explored methods for quickly and accurately detecting leak size and location. Adegboye et al. [9] compared various detection methods based on system accuracy, leak size, location detection, usability, and operation mode. Only fiber optic sensing, pressure point analysis, and dynamic modeling can simultaneously detect both leak size and location. While existing technologies like fiber optics and negative pressure wave methods have demonstrated high accuracy in locating single leaks, detecting multiple leaks simultaneously remains a significant challenge.

Pipeline data are typically classified into steady state and transient state data, each requiring different detection approaches. Steady state data involve stabilized fluid properties such as pressure, flow rate, and velocity, allowing for a single unified value to represent specific operating conditions [10]. Machine learning models such as MLP (Multilayer Perceptron), SVM (support vector machines), and auto-encoder are suitable for steady state data, as these models are effective at regression tasks involving stable input values. In the steady state leak detection case, an MLP model with three hidden layers was selected due to its strong performance in regression analysis and its ability to detect leak size and location. In contrast, transient state data involve continuously changing properties over time, necessitating a different approach [11,12]. Although LSTM (long short-term memory), a time-series learning model, is widely used for predicting future data, its performance may experience reduced performance when faced with abrupt changes. Moreover, LSTM is designed for prediction rather than detection, making it unsuitable for this study's objective of identifying leak size and location. To address this, we applied the CWT (continuous wavelet transform) technique to convert transient data into 2D images representing time-frequency information, which were then fed into a CNN (convolutional neural network) model for classification.

The decision to use CWT over traditional methods like the FT (Fourier transform) stems from CWT's superior ability to handle transient state data. While FT performs well with stationary signals, it lacks time localization, making it ineffective for capturing the short-term fluctuations that are typical of pipeline leaks. In contrast, CWT provides time-frequency localization, allowing the model to detect leaks by analyzing signal variations across multiple scales [11,12]. This multi-scale, time-sensitive analysis ensures more accurate detection in transient conditions, which is critical for real-time pipeline monitoring.

Dynamic modeling excels at detecting leak size and location and can analyze both steady and transient states. It not only characterizes leaks effectively but also provides synthetic data for various leak scenarios. Dynamic modeling, though commonly applied in the field, presents certain limitations. For instance, leak size and location need to be predefined prior to simulation (Figure 1). After constructing the model, flow rate, pressure, and temperature are analyzed through complex correlations. As a result, the simulator can only provide calculated results based on preset leak environments. To accurately detect leak size and location, multiple leak models need to be generated, and the model that best matches real data must be selected. This process requires considerable time and resources, meaning dynamic modeling may present challenges for real-time detection, where rapid response is essential.

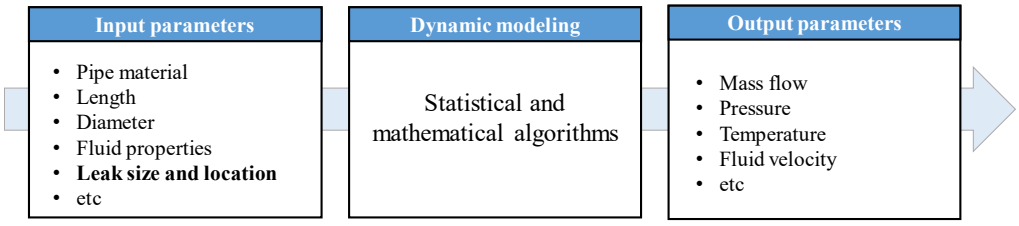


Figure 1. Input/output parameters of dynamic modeling.

To address this drawback, several leak detection and localization methods based on machine learning algorithms have been developed. Machine learning allows computers to automatically improve parameter relationships and build mathematical models based on training and validation data, enabling predictions or decisions without explicit programming [13]. When properly trained, machine learning can quickly and automatically make accurate predictions. This approach has been widely applied in detecting leaks in gas, oil, and water pipelines using acoustic, pressure, or flow data [14–17].

Previous studies have shown that machine learning models can effectively detect leaks, though field applications are limited and the simultaneous detection of leak size and location is rare. Even with dynamic modeling, detecting leaks during transient conditions remains uncommon.

To bridge this gap, this study developed a leak detection model for real-world pipelines that uses machine learning to detect both leak size and location in transient state conditions. First, fluid flow and leak characteristics were analyzed using dynamic modeling. After constructing the dynamic model, the steady state arrival time was evaluated to understand the model's behavior. Most field data, including pressure, flow rate, and temperature, stabilize during the steady state, showing little variation over time. However, these values change continuously in the transient state, complicating fluid flow analysis. Identifying the steady state arrival time is a crucial first step before analyzing fluid flow phenomena. The results of this analysis were used to select sensitive variables, and various leak scenarios were generated for machine learning.

The machine learning model utilized a CNN tailored to the pipeline structure and data type. The CNN offers several advantages that make it highly effective for pipeline leak detection [18]. One significant benefit is its ability to automatically extract and learn features from complex datasets, such as CWT-generated images, without the need for manual feature engineering [19]. This enables the model to detect subtle patterns in the data that may be overlooked by traditional methods. Additionally, CNN is highly scalable and capable of handling large datasets, making it suitable for real-time monitoring applications [20]. Its ability to provide fast and accurate predictions further supports its application in pipeline systems where timely decision-making is critical. A CWT was employed to visualize single vector values over time, and the resulting images were used as training data for the CNN. Additionally, various methods were proposed to improve model accuracy, along with a leak detection flowchart.

In summary, this study developed a model that integrates dynamic modeling and machine learning for rapid and accurate leak detection, specifically designed for transient state conditions.

2. Methodology

2.1. Leak Calculation Model

If a leak occurs, the inlet flow rate will exceed the outlet flow rate and the difference will be equal to the leak flow rate. Thus, the inlet flow rate (Q_{in}) will be the sum of the outlet flow rate (Q_{out}) and the leak flow rate (Q_{leak}):

$$Q_{in} = Q_{out} + Q_{leak} \tag{1}$$

Figure 2 illustrates the change in pipeline pressure with and without leaks. When a leak occurs, the pressure decreases linearly, with the slope varying depending on the location of the leak. Since the flow rate beyond the leak point is lower than the initial flow rate, the pressure gradient becomes smaller.

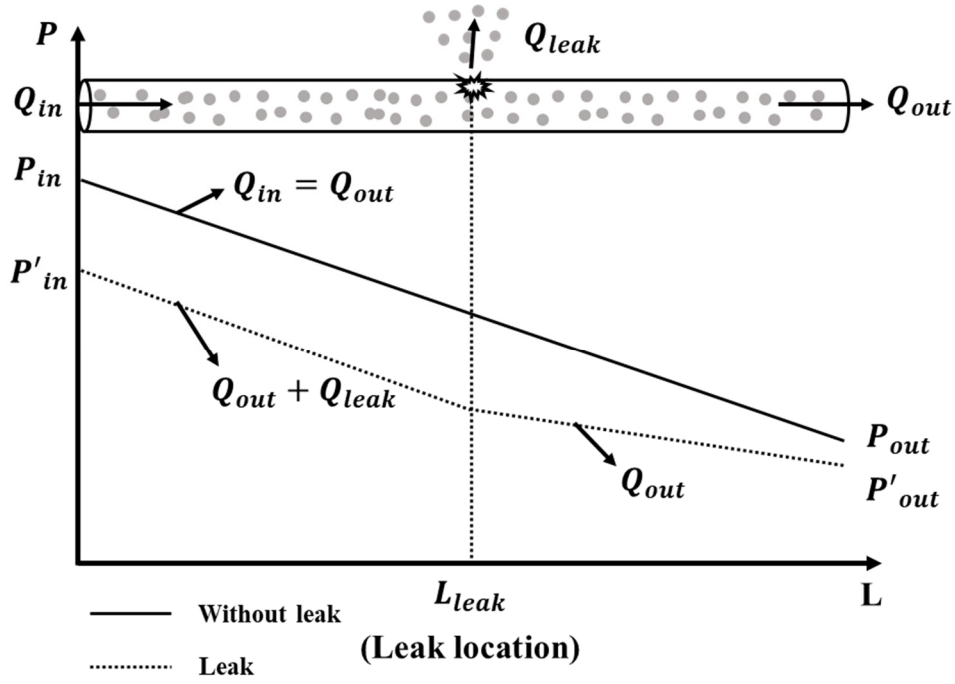


Figure 2. Pipeline pressure variation with and without leaks (modified Dinis et al. [21]).

According to Dinis, Wojtanowicz, and Scott [21], pressure drop is a key factor when constructing a pipeline leak model. The first assumption is that the pressure at the leak point remains unaffected by the leak itself. This is because leak detection relies solely on data measured at the end of the pipeline, reducing the impact of the leak point.

The second assumption is that the orifice equation can be used to describe the pressure drop across the length of the leak. Generally, the mass flow rate Q_m, measured in kg/s across the orifice, is given by the following equation [22]:

$$Q_m = \frac{C_d}{\sqrt{1-\beta^4}} \epsilon \frac{\pi}{4} d^2 \sqrt{2\Delta P \times \rho} \tag{2}$$

where C_d is the dimensionless discharge coefficient, typically between 0.6 and 0.85, β is the ratio of the orifice diameter d to the pipeline diameter D, and ϵ is the expansibility factor (1 for incompressible gasses and most liquids, and lower for gasses under pressure). Additionally, d represents the internal orifice diameter, ρ is the fluid density, and ΔP is the differential pressure across the orifice.

The following expression can be obtained based on the above.

$$Q_{leak} = C_d A \sqrt{\frac{2\Delta P}{S_g \rho}} \tag{3}$$

where A is the area of the leak, ΔP is a pressure drop between the inside pipe and atmosphere and S_g is a specific gravity of a gas.

According to Scott et al., the equation for simulating a single-phase gas leak in a gas pipeline is

$$Q = CF_{leak}\left({P_{in}}^2 - {P_{out}}^2\right)^n \tag{4}$$

where Q is the gas flow rate at the outlet of the pipeline, C is a pipeline constant, F_{leak} is the reduction in efficiency due to a leak (if there is no leak, 1 is normally used), and n is normally 0.5.

F_{leak} can be expressed as

$$F_{leak} = \left[1 + L_{LD}\left({Q_{LD}}^2 + 2Q_{LD}\right)\right]^{-n} \tag{5}$$

In this equation, L_{LD} (dimensionless leak location) and Q_{LD} (dimensionless leak rate) can be expressed as [17]

$$L_{LD} = \frac{L_{leak}}{L_P} \tag{6}$$

$$Q_{LD} = \frac{Q_{leak}}{Q} \tag{7}$$

where L_P is the length of the pipeline and L_{leak} is the distance to the leak.

2.2. CWT (Continuous Wavelet Transform)

Wavelet transforms are powerful tools with various applications in signal processing [23]. They transform a signal, system, or process model into a set of special signals known as wavelets. These transforms use a small wavelet as a pattern, which is then scaled up or down to represent an arbitrary waveform. The CWT is particularly well suited for signal analysis [24–26].

CWT decomposes an input signal into wavelet coefficients through convolution operations at different scales. The CWT of a function $f(t)$ involves the wavelet function $\psi(t)$, also called the "mother wavelet". $\psi(t)$ can be either a real or complex continuous function and must satisfy the following conditions [27]:

(1) The total area under the curve of the function is zero:

$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \tag{8}$$

(2) The total area of $|\psi(t)|^2$ is finite:

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty \tag{9}$$

The group of time scale waveforms can be obtained by shifting and scaling the mother wavelet shown as follows:

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}} \psi(\frac{t-b}{a}) \tag{10}$$

where a is the scale parameter that controls the dilation or contraction of the wavelet and b is the shifting parameter that moves the wavelet along the time axis. For a given signal $x(t)$, the wavelet coefficient can be represented as follows:

$$wt(a,b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} x(t)\psi^*(\frac{t-b}{a})dt \tag{11}$$

where ψ^* denotes the complex conjugate of ψ. The details of the implementation of CWT can be found in Rioul and Duhamel [23].

As seen in the previous equation, scaling and shifting are key concepts in wavelet transforms. Scaling increases or decreases a signal over time, while shifting moves the wavelet along the time axis. A larger scale corresponds to a lower frequency, while a smaller scale corresponds to a higher frequency.

Since CWT operates with continuous scaling and shifting factors, it can generate an infinite number of wavelets. This allows infinite adjustments to the signal (Figure 3). Consequently, CWT is commonly used to produce spectrograms, providing both time and frequency information about a signal. This information is typically represented as a two-dimensional vector array, much like an image.

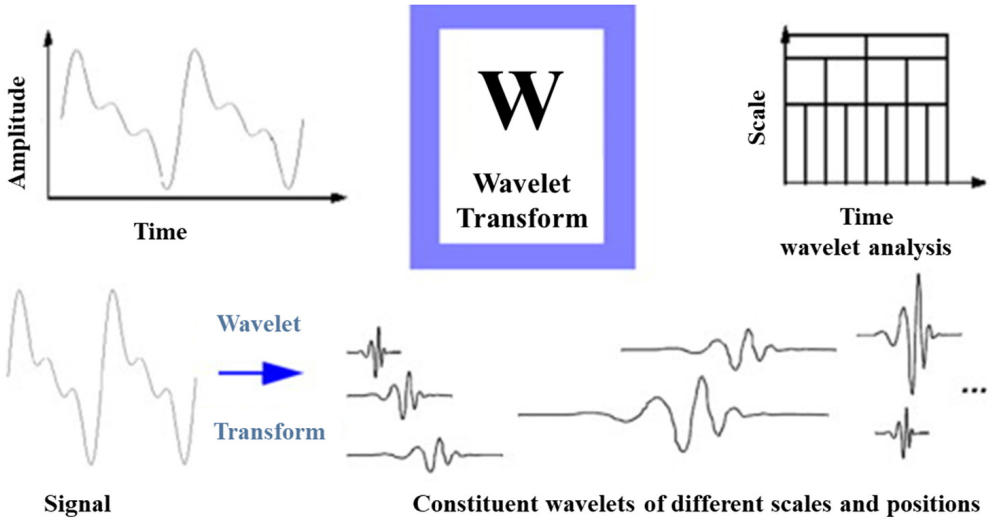


Figure 3. Wavelet transformation and decomposition of a wavelet signal [28].

2.3. Machine Learning Model and Evaluation Methods

2.3.1. GoogLeNet

GoogLeNet [29], which won the ImageNet competition, is highly effective for classification and detection tasks. This algorithm is designed to increase the depth and width of a neural network while maintaining the constant computational demands and maximizing resource utilization. It has four main characteristics as follows:

(1) 1×1 Convolution: GoogLeNet uses 1×1 convolutions to reduce the number of feature maps, which in turn decreases the computational load, allowing the network to become deeper.
(2) Inception Module: This structure introduces a neural network within the main neural network, enabling the extraction of a wider variety of features.
(3) Average Pooling: For the fully connected layers, GoogLeNet uses average pooling, which creates a one-dimensional vector by averaging each feature map. This significantly reduces the number of learnable parameters.
(4) Auxiliary Classifier: To address the vanishing gradient problem, which can occur in deeper networks, GoogLeNet includes two auxiliary classifiers to help with gradient flow during backpropagation.

2.3.2. SqueezeNet

SqueezeNet [30] achieves similar accuracy to AlexNet but with 50 times fewer parameters. It can also be compressed to less than 0.5 MB (510 times smaller than AlexNet) using model compression techniques. This smaller computation requirement allows faster learning and enables more frequent updates, making it ideal for real-time tasks. Additionally, SqueezeNet is well suited for applications with limited memory.

2.3.3. Evaluation Methods

The accuracy was evaluated by how which percentages of the total number of images were correctly classified (Equations (12) and (13)).

$$\text{Accuracy}(\%) = \frac{Number\ of\ correctly\ classified\ images}{Number\ of\ images} \quad (12)$$

Accuracy is commonly used in classification CNN models. However, this metric alone cannot evaluate the accuracy of individual classes. To address this limitation, this study also employed a confusion matrix, which provides a more detailed evaluation for each class. The confusion matrix measures two key metrics: precision and recall (Figure 4).

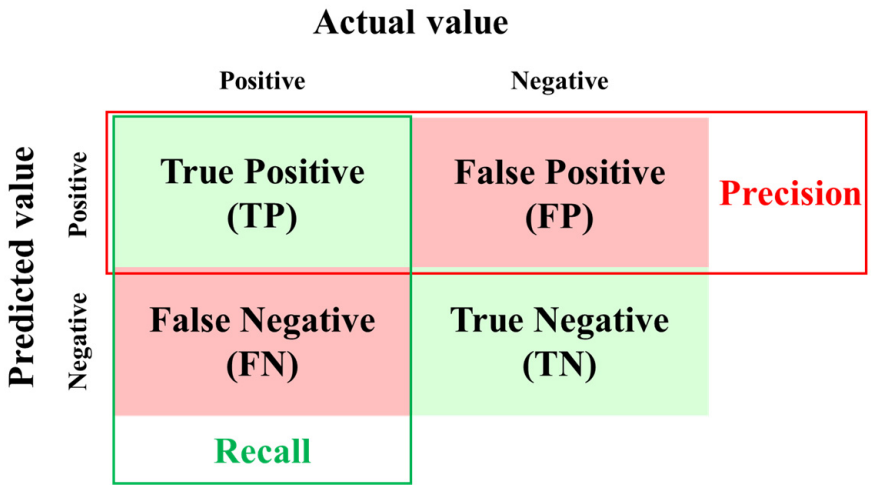


Figure 4. Calculation of precision and recall in the confusion matrix.

(1) Precision is the ratio of true positives to the total number of predicted positives (Equation (14)).
(2) Recall is the ratio of true positives to the total number of actual positives (Equation (15)).

$$Accuracy\ (\%) = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

2.4. Flowchart of Transient Leak Detection System Development Process

As a summary of the methodology discussed in this chapter, Figure 5 provides a flowchart that outlines the development process of the transient leak detection system. This flowchart visualizes the key steps, starting from the initial dynamic modeling and data acquisition to the machine learning model implementation for leak size and location detection. The process begins by modeling the pipeline's behavior under various conditions using dynamic simulations. Following this, sensitivity analysis is conducted to identify the most influential parameters, such as mass flow and pressure, which significantly change during a leak. Using these data, machine learning models, particularly CNN, are trained to detect leaks under transient conditions. By reviewing this flowchart, readers can better understand the structured approach used to develop the detection system, setting the stage for the results and discussions in the following chapters.

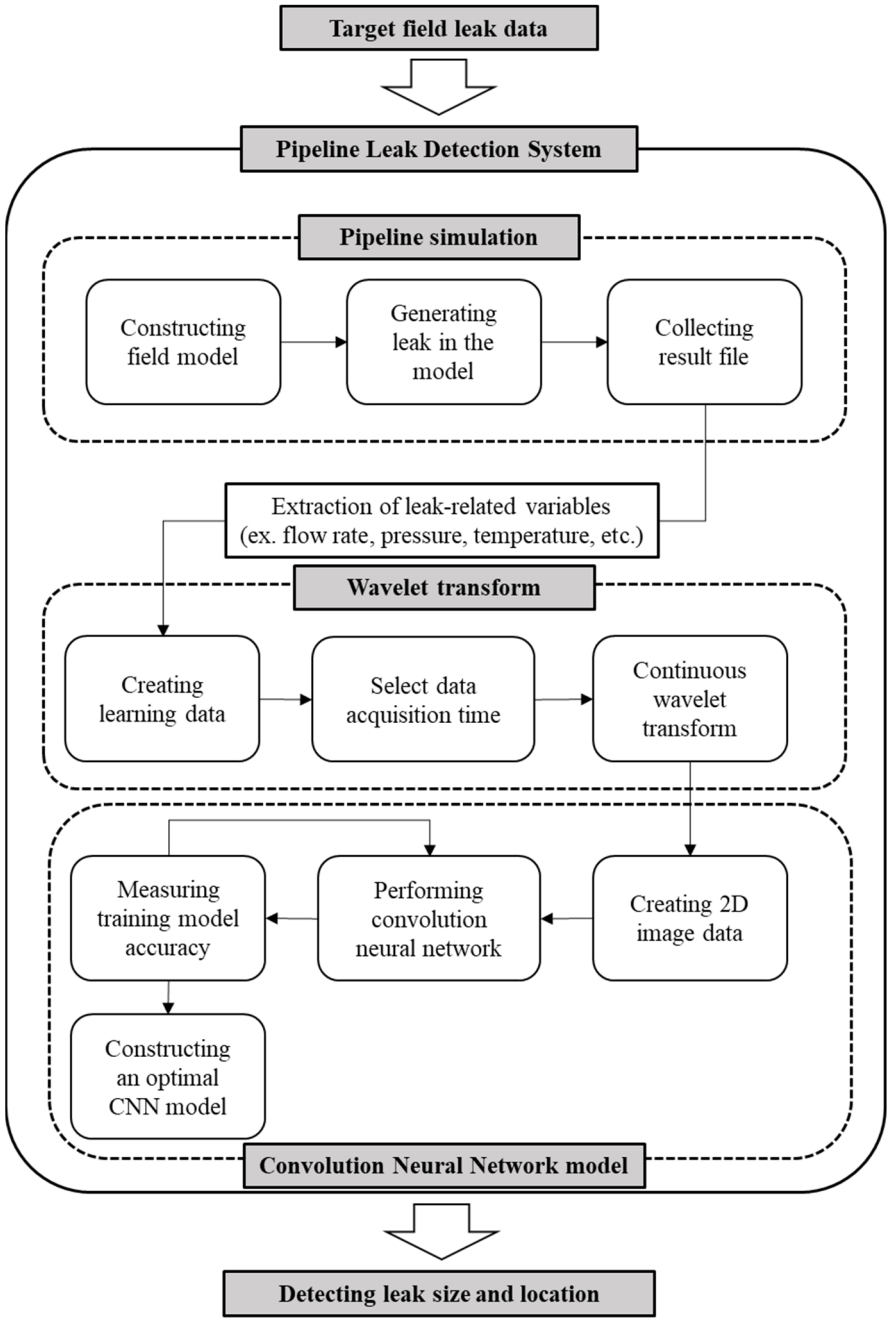


Figure 5. Flowchart for the development of a leak detection system in a transient state.

3. Results and Discussion

3.1. Model Construction and Leak Simulation

The targeted pipeline, located in Alberta, Canada, is approximately 188 km long with a diameter of 91.44 cm. Field data, such as flow rate, pressure, and temperature, are collected through sensors installed in the system.

This study used commercial software (SLB's OLGA, Available online: https://www.software.slb.com/software-news/support-news/olga/olga_2017_2 (accessed on 29 October 2024)) to model the pipeline. OLGA is widely used in the industry and can simulate both steady and transient states. The target fluid is a blend of 90% methane and 10% hydrogen, and PVT modeling was performed using the Multiflash module in OLGA.

Table 1 and Figure 6 provide details about the pipeline's properties and the model used for the base case. The pipeline was divided into 194 sections to ensure numerical

convergence and optimize computing time. Figure 7 shows the mass flow and pressure results from the base model, which assumed average flow and pressure conditions in Canada (flow rate: 286.57 kg/s, outlet pressure: 55 bara). As a result, the flow rate remained constant across all sections, with the inlet pressure calculated to be around 96.5 bara.

Table 1. The properties of the onshore pipeline and a model description.

Components	Values
Fluid	CH_4 + H_2 (blending)
Pipe material	steel, coating
Pipe inner diameter (cm)	91.44
Pipe length (km)	188
Number of sections (ea)	194
Size of sections (m)	≤1000
Inlet pressure (bara)	96.5
Outlet pressure (bara)	55
Inlet and outlet temperature (°C)	20
Flow rate (kg/s)	286.57

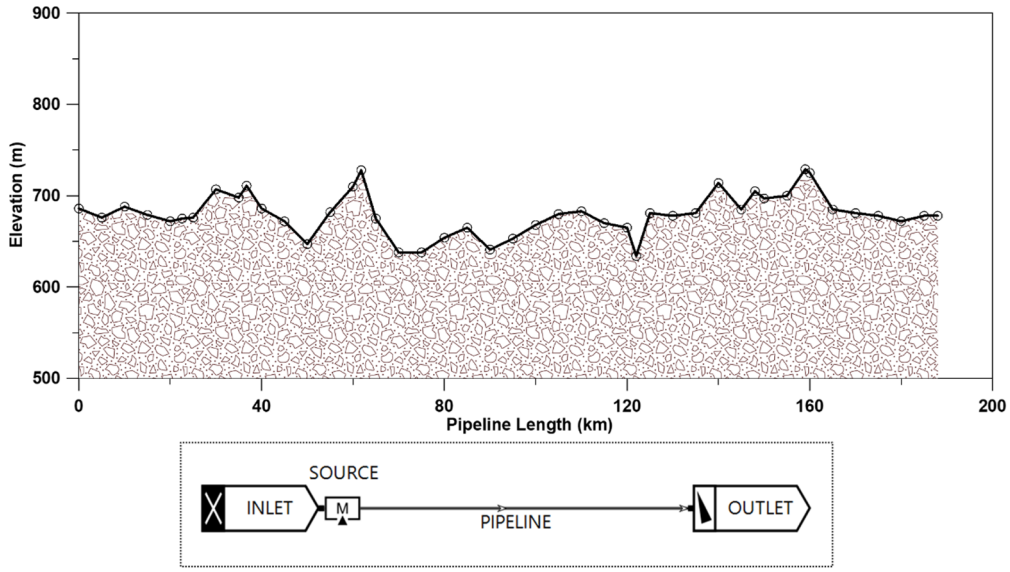


Figure 6. Geometry of onshore target pipeline and OLGA base model.

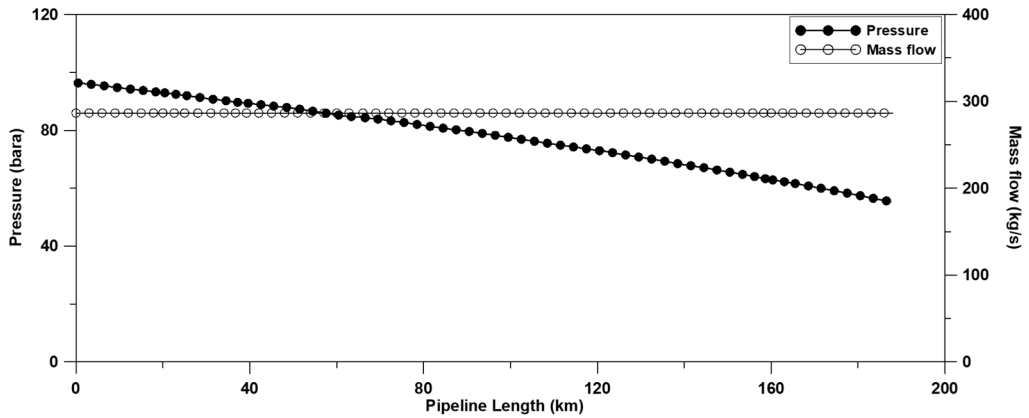


Figure 7. Mass flow and pressure results from the base model.

Before conducting fluid flow analysis for the leak simulation, the feasibility of the composition simulation was assessed. Since methane and hydrogen are blended, the

proportion of leaked gas can vary depending on factors such as gas velocity, density, and molecular size. Many researchers have predicted that smaller molecules like hydrogen would leak much faster than larger ones. However, Mejia et al. [31] conducted experiments with methane–hydrogen mixtures at both low (0.032 bar) and high (4.17 bar) pressures and found no significant difference in leak rates between the gasses.

Additionally, when comparing the remaining gas content after the leak test, the mixing ratio showed less than a 1% change from the initial composition. Although this was a small-scale laboratory test, it was considered a successful comparison of leak rates for mixed gasses.

In this study, a simulation test was conducted to determine if there was a significant difference in the compositional simulation. A mixture of 90% methane and 10% hydrogen was applied to the target pipeline, which had a 10 cm diameter leak (Figure 8). The results showed no significant changes in mass flow or the mole fraction of methane and hydrogen due to the leak. This was attributed to the high fluid velocity, which ensured that the gasses flowed in a fully mixed state. As a result, a single-component simulation was used in this study instead of a compositional simulation to reduce both time consumption and complexity.

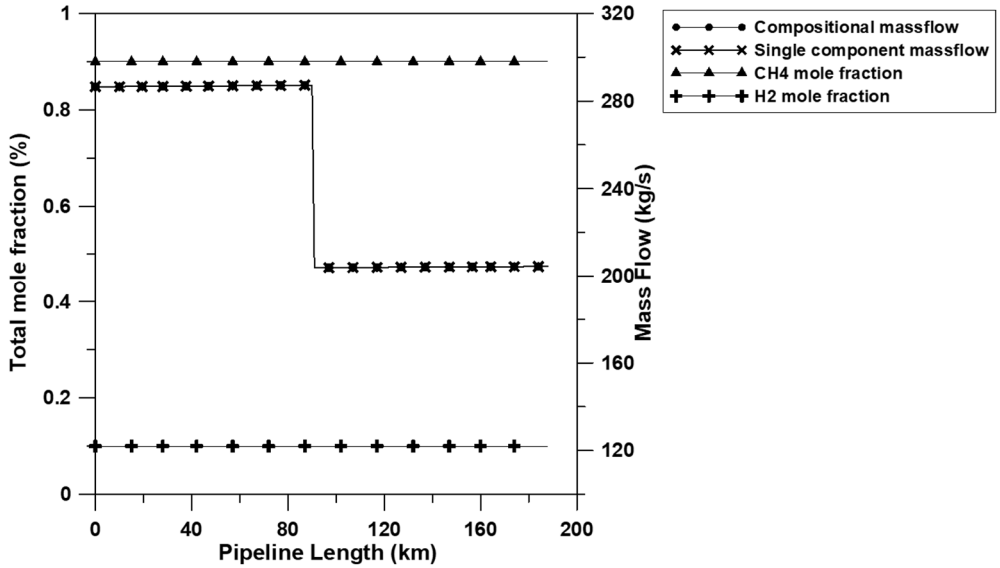


Figure 8. Comparison between the single component simulation and compositional simulation.

To determine the detectable leak size, leak sizes ranging from 0 to 25 cm were applied at the same location. As shown in Figure 9, when the leak size exceeded 20 cm, the mass flow rate became negative. The leak mass flow rate is typically calculated based on factors such as leak size, fluid density, and the pressure difference between the inside and outside of the pipeline. As the leak size increases, the leak mass flow rate rises. When the leak mass flow rate exceeds the system's mass flow rate, it results in a negative value, which is a mathematical outcome. In this model, the maximum detectable leak size was set at 20 cm.

A detailed analysis of mass flow over time demonstrated that stabilization occurred within 6 h across various leak sizes and locations (Figure 10). Most researchers rely on steady state data to detect leak size and location because detecting leaks during transient states is more difficult due to continuous changes over time. For short gas pipelines, the transient state lasts only a few minutes, so complex transient data are unnecessary for leak detection. However, for long pipelines like the one in this study, it takes longer to reach a steady state after a leak.

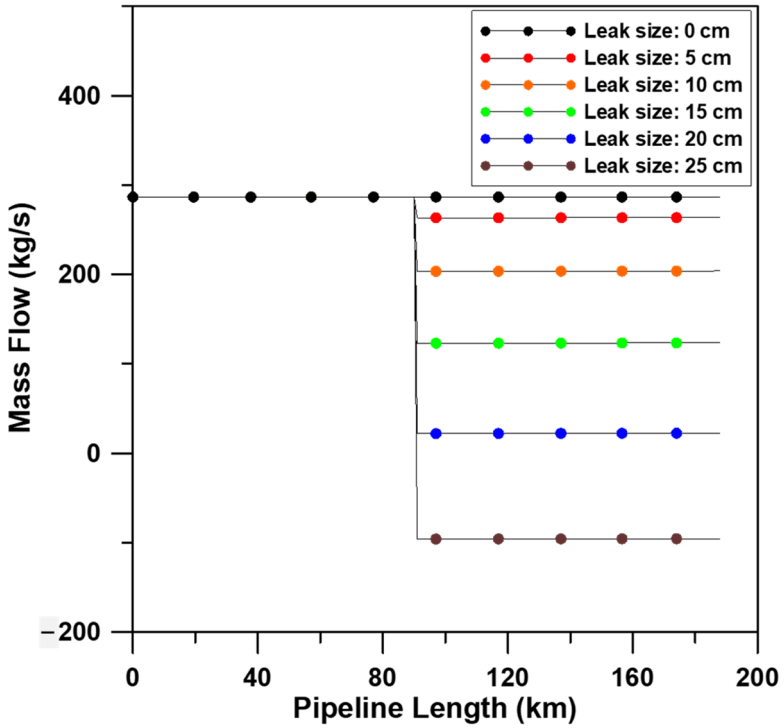


Figure 9. Mass flow profile for various leak sizes (0–2 cm).

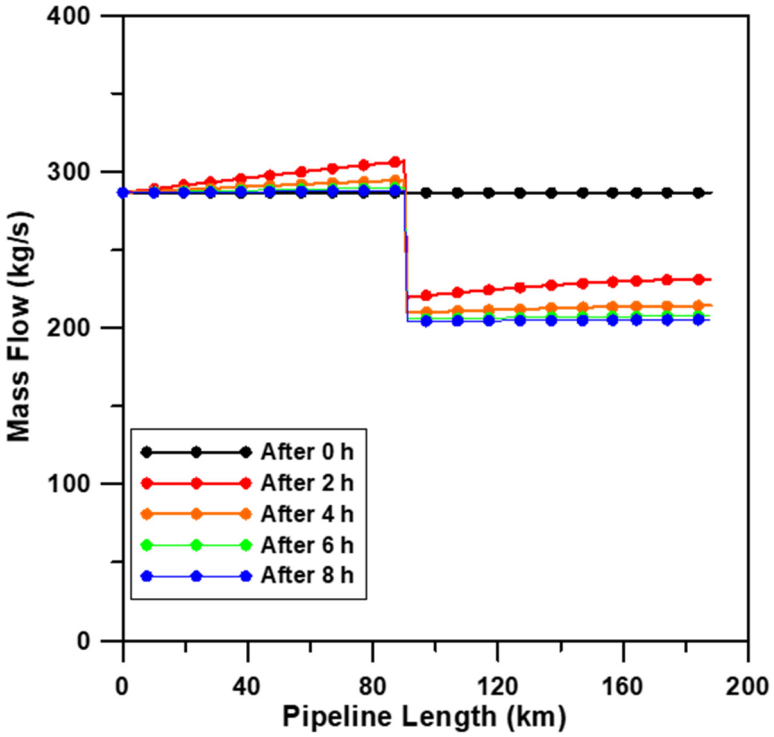


Figure 10. Variation in mass flow over time for the 10cm leak size.

Figure 11 also shows how the time to reach a steady state varies with leak size. Whether the leak size was 5, 10, 15, or 20 cm, the time to stabilize was consistently within 6 h, indicating that leak size does not significantly affect the steady state arrival time.

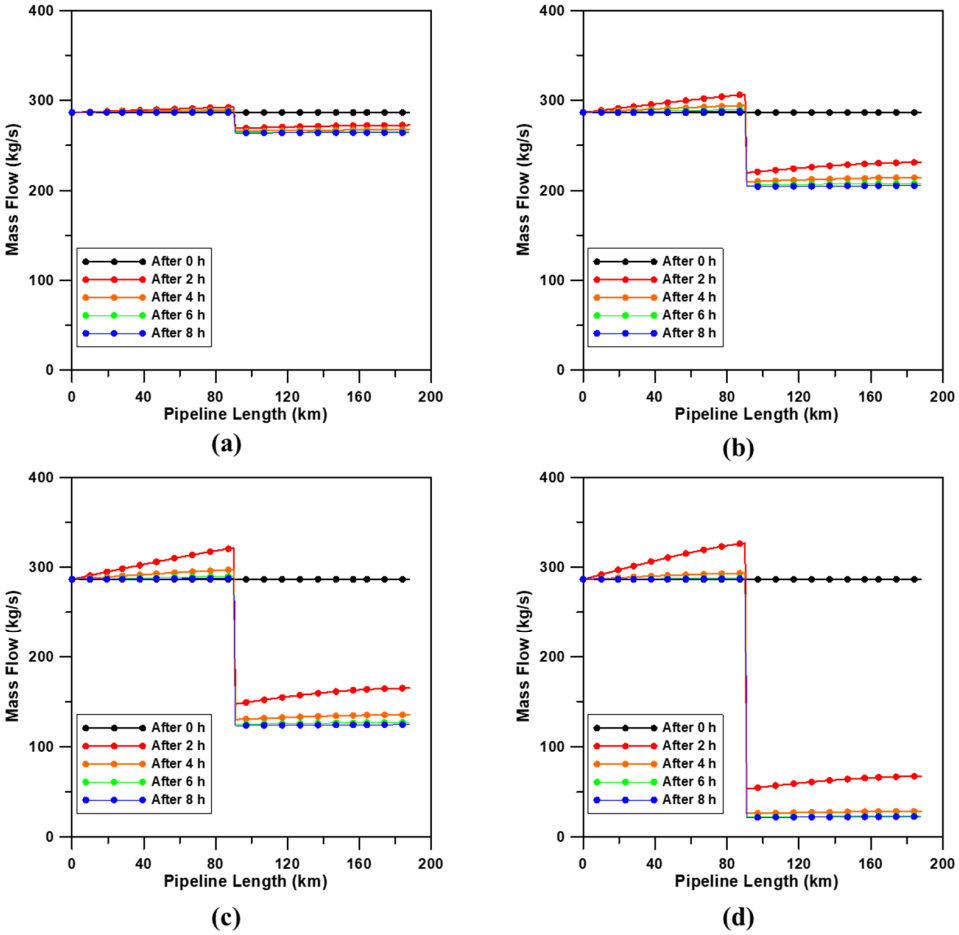


Figure 11. Comparison of steady state arrival time depending on leak size ((**a**): 5 cm, (**b**): 10 cm, (**c**): 15 cm, (**d**): 20 cm).

A sensitivity analysis was conducted to identify which parameters were most affected by the leak. Leak locations were distributed along the pipeline at distances of 37 km, 95 km, and 150 km from the inlet. The leak size ranged from 0 to 20 cm at each location. Consistent with previous studies [32], significant changes were observed in both mass flow and pressure, with pressure being the most affected by leak size and location. However, since this result was based on steady state profile data after a leak, it is useful for reviewing flow patterns in the pipeline but not for understanding the flow during the transient state. As mentioned earlier, profile data reflect values along the pipeline's length by determining values for each section. Even if the pipeline length increased, the flow pattern would remain largely the same, with only the magnitude of changes differing. Therefore, profile data are best used for checking overall changes in specific values along the pipeline.

In contrast, trend data capture values such as pressure, flow, and velocity over time in a specific section of the pipeline. These data are particularly helpful for observing how values change over time at a particular point. Figure 12 provides an example of both profile and trend data.

The sensitivity analysis confirmed that mass flow changes primarily occurred at the pipeline outlet, while pressure changes were most notable at the pipeline inlet. As a result, the flow pattern during the transient period was further analyzed using trend data.

Figure 13 shows a graph analyzing the change in mass flow over time at the pipeline's outlet. The mass flow at the outlet gradually decreased as the leak size increased. The pattern of the decrease varied based on the leak location. When the leak occurred near the pipeline's inlet, the mass flow decreased slowly. However, when the leak occurred near the outlet, the

flow rate dropped more rapidly. Because mass flow responds differently depending on leak size, it serves as a valuable parameter for identifying leak size during detection.

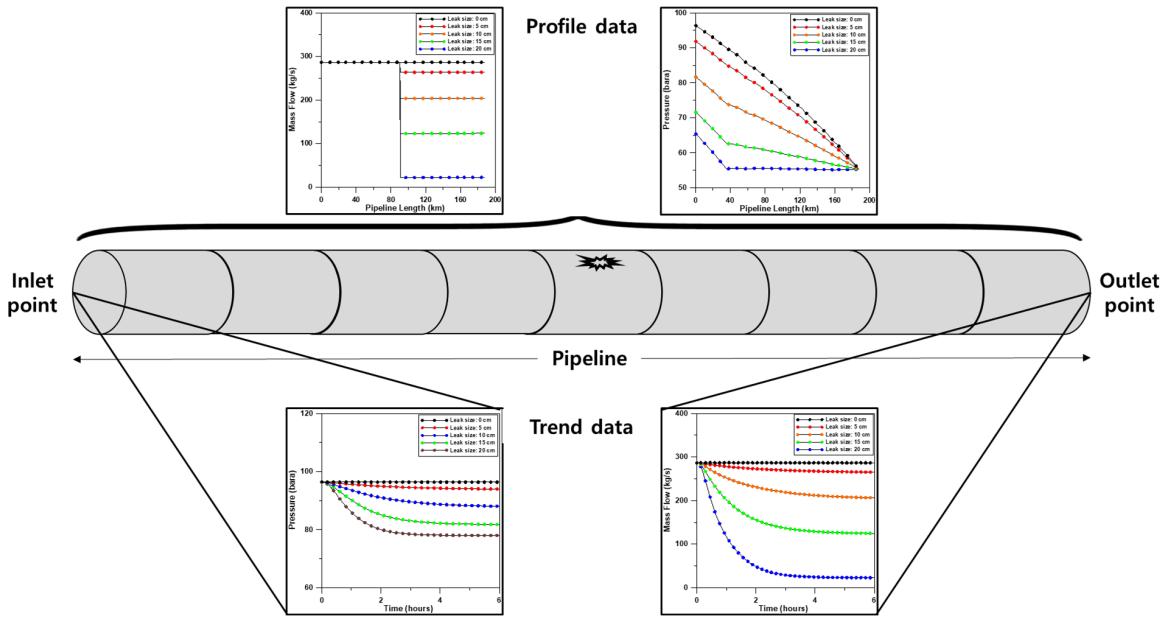


Figure 12. Comparison between profile data and trend data.

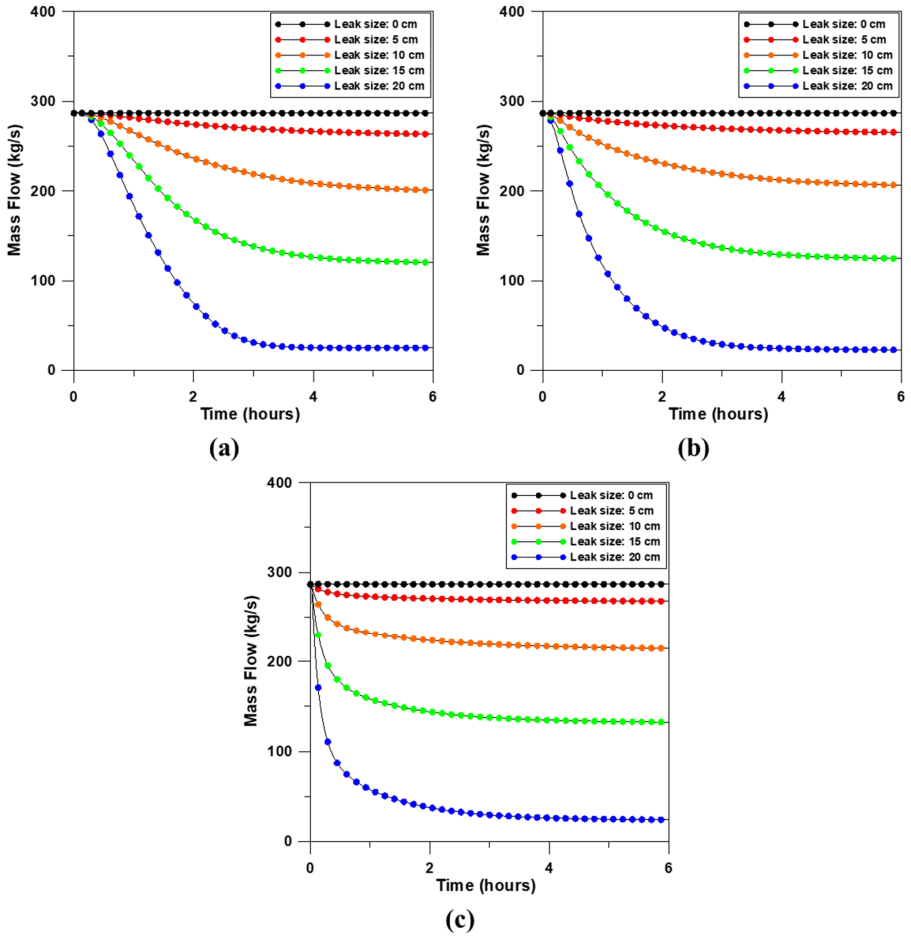


Figure 13. Mass flow trend for various leak sizes (0–20 cm) at (**a**) 37 km, (**b**) 95 km, and (**c**) 150 km at outlet point.

Figure 14 presents a graph of pressure changes over time at the pipeline's inlet. As the leak size increased, the pressure measured at the inlet decreased gradually, with the maximum pressure reduction depending on the leak location. When the leak was near the inlet, the pressure dropped by up to 65 bara for a 20 cm leak. In contrast, when the leak was closer to the outlet, the pressure only decreased to 90 bara, indicating that the pressure is highly sensitive to leak location.

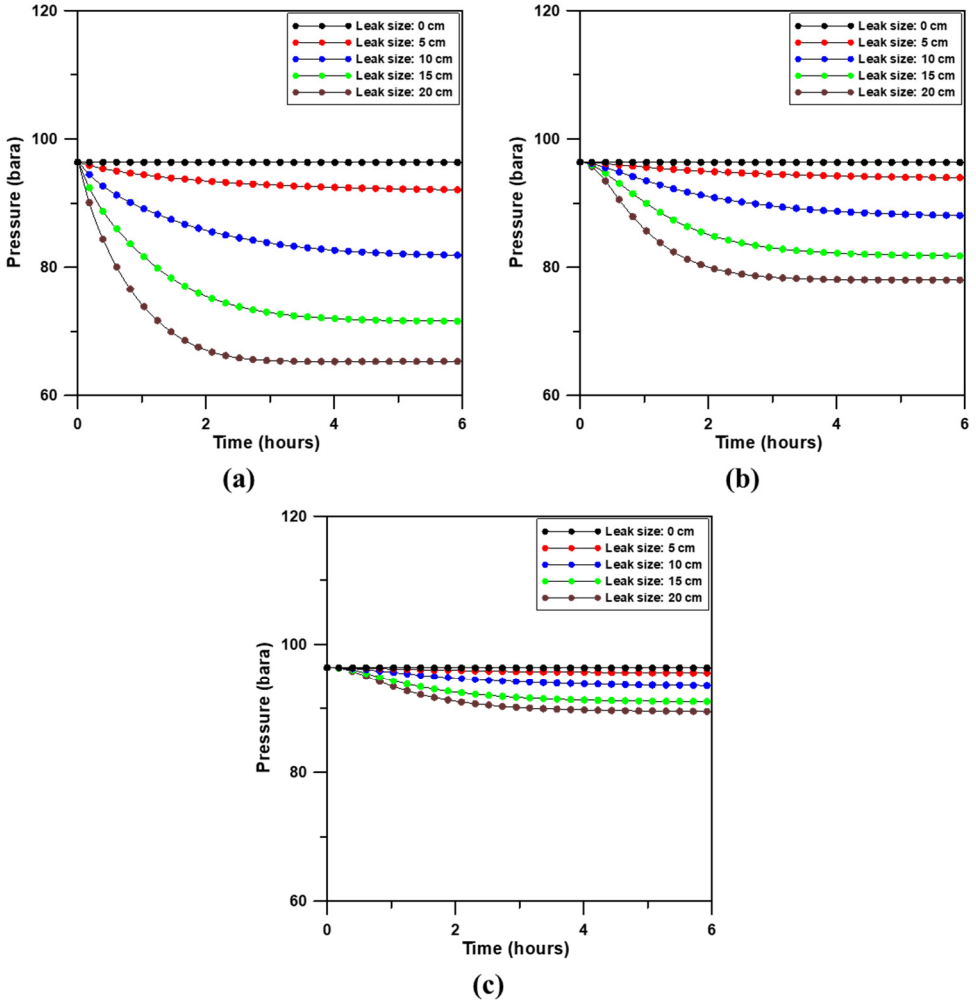


Figure 14. Pressure trend for various leak sizes (0–20 cm) at (**a**) 37 km, (**b**) 95 km, and (**c**) 150 km at inlet point.

In conclusion, the maximum leak size was set at 20 cm, and both mass flow and pressure profiles, along with trend analyses, were performed. However, at 20 cm, most profile values were similar to the boundary conditions, as nearly all gasses had leaked out, leaving only small amounts flowing through the pipeline. The pressure after the leak point was almost equal to the outlet pressure, and the mass flow rate was close to zero. Therefore, the 20 cm leak data were not suitable for learning purposes. The study focused on data from leaks sized 0 to 10 cm, targeting the rapid detection of smaller leaks.

To deepen the physical interpretation of the observed results, we analyzed variations in velocity and temperature in detail and conducted correlation analyses. We also confirmed that the physical changes in mass flow, pressure, velocity, and temperature observed in this study align with our previous research, which extensively examined the physical changes due to leak size and location and established a foundation for understanding these variables' relationships in subsea pipelines [32]. By referring to these prior findings, our study provides a consistent basis for interpreting physical changes in onshore pipeline leak detection under transient conditions. It is important to note that this study does not

consider noise in the sensor data or environmental interference, as it focuses on the primary mechanisms of leak detection.

3.2. Data Acquisition and Preprocessing

The purpose of using machine learning is to accurately detect the leak size and location through the sensitivity analysis of mass flow and pressure. Machine learning requires a large amount of training data, which is collected through multiple simulations involving various leak sizes and locations. Table 2 outlines the range of data used. Leak sizes were classified into 10 categories, ranging from 1 cm to 10 cm, and leak locations were divided into 194 sections, with each section being less than 1 km in length. In total, 1940 leak scenarios were created, with each simulation running for one hour.

Table 2. Parameter range according to leak scenarios.

Target Mass Rate (kg/s)		**Methane Mole Percent (%)**	**Hydrogen Mole Percent (%)**	
286.57		90	10	
	Min	**Max**	**Interval size**	**Number of cases**
Leak size (cm)	1	10	1	10
Leak location (km)	0	188	$\leq$1	194

To optimize machine learning efficiency, data were collected every 10 s, resulting in 360 data points per simulation. The collected data were then split into training, validation, and test sets. First, we divided the data into Training + Validation (80%) and Test (20%) sets, resulting in 1552 datasets for Training + Validation and 388 for Testing. Next, the Training + Validation set was split into Training (64%) and Validation (16%) at an 8:2 ratio, resulting in 1242 datasets for Training and 310 for Validation. Importantly, the test data were not used during training and were only applied to evaluate the model's performance.

3.3. CWT Analysis

CWT was used to create spectrograms based on measured values over time. Spectrograms typically provide both time and frequency information, expressed as a two-dimensional vector array, similar to an image. In recent studies, CWT images have been widely used as input data for training CNNs. This study applied wavelet transforms using mass flow and pressure data, which are the most sensitive parameters in leak events. A CNN model was then developed to detect both leak size and location. The generated images were expressed in Red, Green, and Blue—the three primary colors of light—and had a pixel size of 224 $\times$ 224, resulting in an image array of 224 $\times$ 224 $\times$ 3.

Figure 15 provides an example of the type of image produced when wavelet transformation is applied to leak trend data. Various CWT images were created depending on the leak size or location. In total, 1940 CWT images were generated based on the previously discussed leak scenarios.

In this study, the images generated by the CWT were carefully preprocessed to ensure they were suitable for the CNN. The images were resized to a standard resolution of 224 $\times$ 224 pixels to maintain consistency with the input requirements of the CNN architecture. Data normalization was also applied to the image pixel values, mapping them to a predefined range $[0, 1]$ or $[-1, 1]$, which helps improve the convergence of the model during training. This normalization technique ensures that the image data are scaled appropriately without introducing biases, as the leak events were evenly distributed across the pipeline. These preprocessing steps, including resizing and normalization, ensure that the quality and format of the CWT images are optimal for training the CNN model effectively.

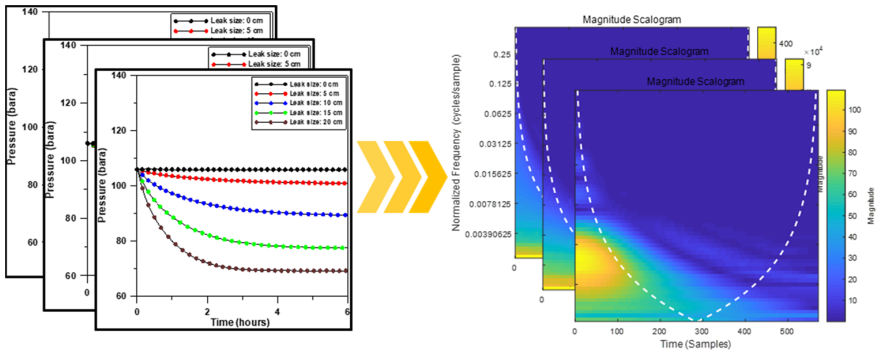


Figure 15. Example of wavelet transformation on leak trend data.

3.4. Convolution Neural Network Training and Evaluation

A CNN was selected as the learning model due to its effectiveness in solving classification problems with image data. This study used two different CNN models: GoogLeNet and SqueezeNet. The CNN for leak detection was trained on two tasks: one for detecting leak size and the other for detecting leak location. The leak size detection model used mass flow trend data, while the leak location detection model used pressure trend data. Both datasets were gathered every 10 s over a one-hour period. Leak size detection data were classified into 10 categories, with 1 cm intervals up to 10 cm. Leak location detection data were divided into 10 sections, each representing approximately 20 km intervals. Figure 16 illustrates the CNN model structure, and Table 3 provides details about the models.

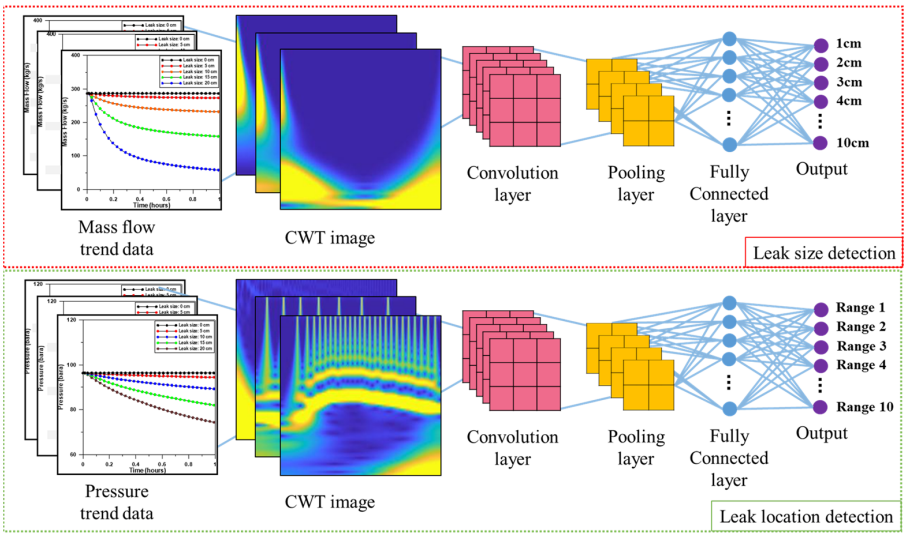


Figure 16. Process and structure of leak size and location detection model.

Table 3. Details of the convolution neural network model.

Parameters	Values
Number of images	1940
Size of input image	224 × 224 × 3
Output layer node	10
Number of hidden layers	144
Pooling layer	Max pooling
Activation function	ReLU
Evaluation	Accuracy (%) Confusion matrix

Variables like learning rate, batch size, and epoch are often referred to as hyperparameters. These are crucial elements in the model, but there is no one-size-fits-all solution, so a trial-and-error process is essential to find the optimal configuration. Table 4 shows

the accuracy evaluation results using different hyperparameters. Training a CNN requires numerous calculations, and using a Central Processing Unit (CPU) increases time consumption. To reduce learning time, this study utilized a Graphics Processing Unit (GPU) (NVIDIA GeForce GTX 1080 Ti 11GB). Comparing training times, the GPU reduced the process by up to 20 times compared to the CPU.

Table 4. Hyperparameter ranges and the accuracy evaluation of the optimal case.

Hyperparameter	Value Ranges	
	GoogLeNet	**SqueezeNet**
Learning rate	0.001–0.0001	
	0.0001	0.0001
Leak size batch size	3–100	
	7	3
Leak size epoch	10–100	
	30	20
Leak location batch size	3–100	
	3	3
Leak location epoch	10–100	
	20	20
Leak size accuracy	90.26%	85.13%
Leak location accuracy	78.60%	70.62%

The model performed well in both leak size and location detection. GoogLeNet achieved an accuracy of 90.26% for leak size detection and 85.13% for SqueezeNet, while the leak location detection accuracy was 78.60% for GoogLeNet and 70.62% for SqueezeNet. Figures 17 and 18 display the confusion matrix results for the model. Overall, leak size detection performed well for each class. Leak location detection was most accurate in the front and middle sections of the pipeline, but less so in the rear sections. To further improve detection accuracy, two additional enhancements were applied: (1) adjusting data acquisition time and (2) screening the learning data.

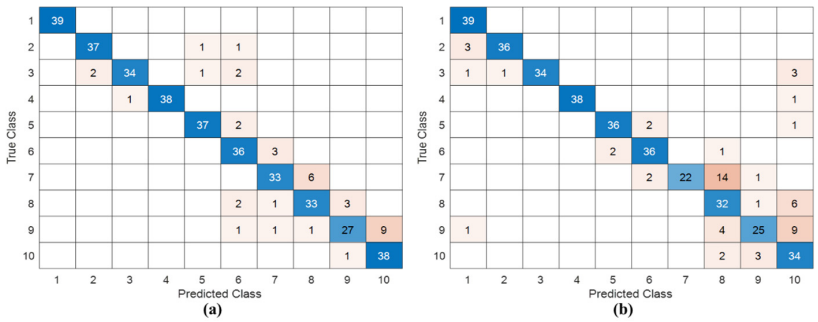


Figure 17. Confusion matrix of leak size (**a**) GoogLeNet, (**b**) SqueezeNet.

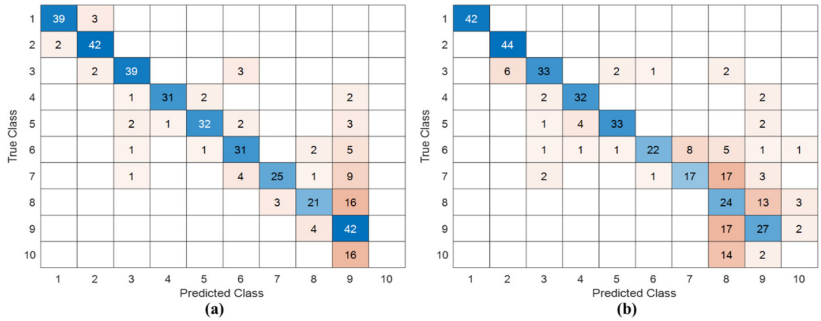


Figure 18. Confusion matrix of leak location (**a**) GoogLeNet, (**b**) SqueezeNet.

3.5. Accuracy Improvement

3.5.1. Data Acquisition Time Adjustment

First, the accuracy of the CNN model was analyzed after adjusting the data acquisition times. The initial model was trained using one hour of data collected at 10 s intervals. However, in this step, additional data were collected at 15 and 30 min intervals for leak size detection and at 30 and 120 min intervals for leak location detection. For leak size detection, the 15 and 30 min data were used to assess whether the high performance could be maintained with less data, as the initial results were already strong. In contrast, for leak location detection, 120 min data were included because the initial results were relatively lower (Figure 19).

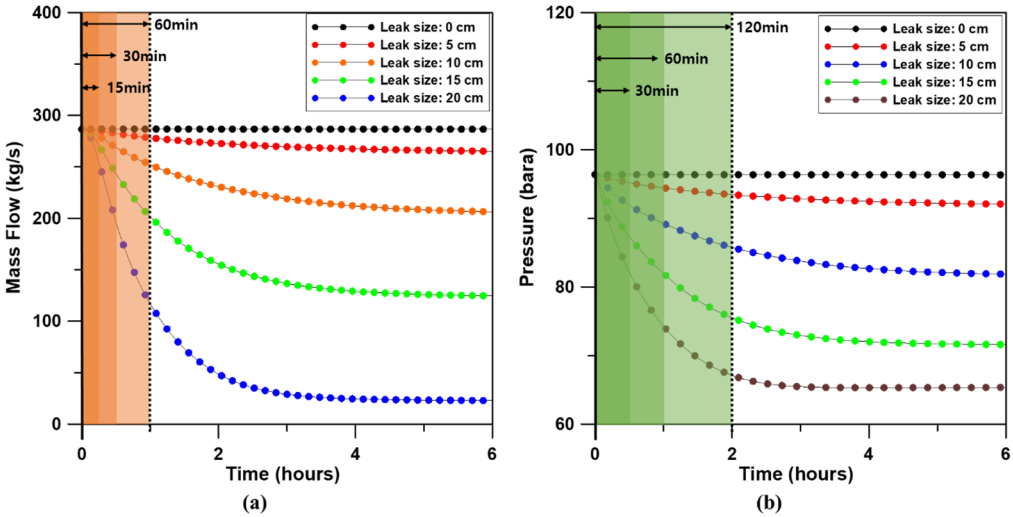


Figure 19. Adjustment of data acquisition time for (**a**) leak size detection and (**b**) leak location detection.

Model training was then performed using the adjusted data acquisition times for both leak size and location, and the model's accuracy was compared (Table 5). As the data acquisition time increased, the accuracy for both leak size and location detection improved. Leak size detection reached nearly 90% accuracy with only 30 min of data. For leak location detection, the accuracy significantly improved when 120 min of data were used. This suggests that additional accuracy improvements are necessary for leak location detection.

Table 5. CNN model accuracy evaluation of the modified model.

Data Acquisition Time (min)	Detection Type	GoogLeNet Accuracy	SqueezeNet Accuracy
15 min	Leak size	64.10%	48.21%
30 min	Leak size	88.21%	87.18%
60 min	Leak size	90.26%	85.13%
Data Acquisition Time (min)	**Detection Type**	**GoogLeNet Accuracy**	**SqueezeNet Accuracy**
30 min	Leak location	68.30%	61.60%
60 min	Leak location	78.60%	70.62%
120 min	Leak location	83.25%	78.61%

3.5.2. Learning Data Screening

Secondly, according to Kim et al. [32], leaks smaller than 1 cm result in only a minimal amount of gas escaping, leading to no significant changes in mass flow or pressure. This lack of data can degrade the model's accuracy. Therefore, the model was retrained and

reevaluated after excluding cases with leaks smaller than 1 cm. Table 6 presents the accuracy of the CNN model based on the leak size range. This adjustment significantly improved the accuracy across all leak location classes, with some cases showing nearly a 14% improvement (Figure 20).

Table 6. CNN model accuracy evaluation of rearranged leak size model.

Data Acquisition Time (min)	CNN Model	Initial Leak Size	Data Acquisition Time (min)
15 min	GoogLeNet	68.30%	75.93%
	SqueezeNet	61.60%	70.77%
30 min	GoogLeNet	78.60%	83.67%
	SqueezeNet	70.62%	83.38%
60 min	GoogLeNet	83.25%	89.68%
	SqueezeNet	78.61%	82.81%

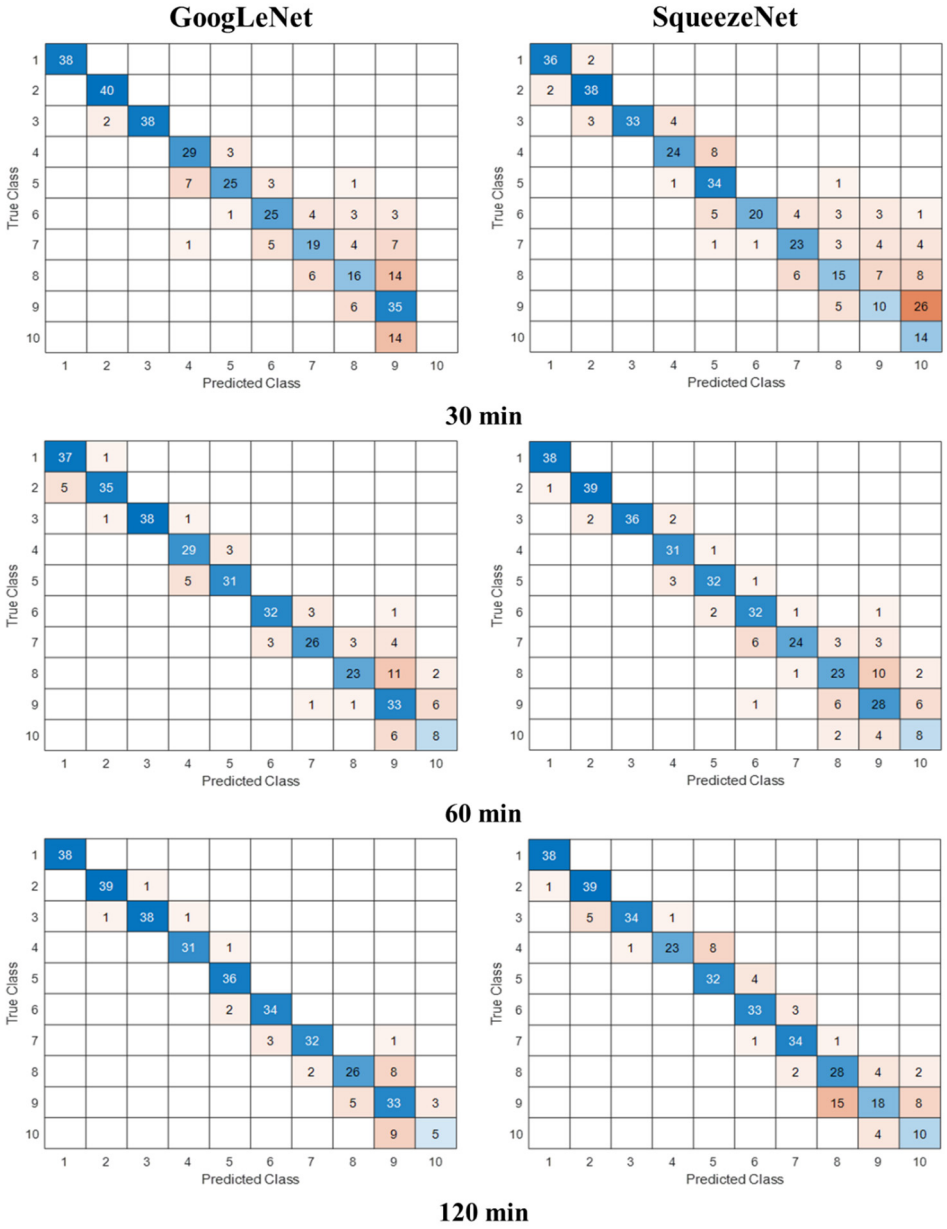


Figure 20. Confusion matrix results of GoogLeNet and SqueezeNet by data acquisition time.

3.6. Possible Extensions of the Study

Future research could focus on validating the model using actual field data from operational pipelines, which would enhance its real-world performance and applicability. Additionally, the detection of multiple simultaneous leaks, which is a common challenge in complex pipeline networks, could be explored to further improve the model's versatility. Another potential area of study is optimizing the model for various pipeline materials, environmental conditions, and gas compositions, ensuring its broader applicability across different pipeline systems. These extensions would contribute to making the model more robust and adaptable for real-world pipeline monitoring and safety.

4. Conclusions

This study developed a machine learning-based leak detection model for onshore methane–hydrogen blending pipelines, specifically designed to function in transient state conditions. Through the integration of dynamic modeling and CNN, the model achieved high accuracy in detecting both leak size and location. One key limitation of this study is the lack of access to actual leak data from operating pipelines. Due to this limitation, synthetic leak scenarios were generated for simulations, which allowed for the controlled testing and training of the machine learning model. While these simulations provide valuable insights, the results may not fully capture real-world complexities. This constraint highlights the need for future studies incorporating actual leak data to validate and potentially improve the model's real-time accuracy.

The numerical simulations highlighted that mass flow and pressure are the most sensitive parameters for identifying leaks. The model's performance was evaluated using various leak scenarios, confirming that the time to reach a steady state after a leak is approximately 6 h, regardless of leak size. Using CWT to generate spectrograms, the CNN demonstrated over 90% accuracy for leak size detection and over 80% for leak location detection after data acquisition and screening optimizations.

The key advantages of this approach include automatic feature extraction from complex datasets, real-time leak detection, and scalability for large datasets, making it particularly suitable for pipeline monitoring. Future extensions of this study could explore the integration of field data to improve real-time applicability and the development of a more comprehensive model capable of detecting multiple simultaneous leaks or leaks under varied environmental conditions. This work provides a foundation for further advancements in pipeline safety and operational efficiency.

Author Contributions: Conceptualization, J.K. and Y.L.; methodology, Y.L.; software, J.K.; validation, Y.L. and J.K.; formal analysis, S.H.; investigation, D.K. and J.K.; resources, Y.L.; data curation, J.K.; writing—original draft preparation, J.K.; writing—review and editing, S.H.; visualization, J.K.; supervision, S.H. and Y.L.; project administration, Y.L.; funding acquisition, S.H. and Y.L. 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) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (Nos. 20226A10100020, 20172510102150).

Data Availability Statement: Data are contained within the article.

Acknowledgments: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00210272).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Capelle, J.; Gilgert, J.; Dmytrakh, I.; Pluvinage, G. Sensitivity of pipelines with steel API X52 to hydrogen embrittlement. *Int. J. Hydrogen Energy* **2008**, *33*, 7630–7641. [CrossRef]
2. labidine Messaoudani, Z.; Rigas, F.; Hamid, M.D.B.; Hassan, C.R.C. Hazards, safety and knowledge gaps on hydrogen transmission via natural gas grid: A critical review. *Int. J. Hydrogen Energy* **2016**, *41*, 17511–17525. [CrossRef]

3. Chae, M.J.; Kim, J.H.; Moon, B.; Park, S.; Lee, Y.S. The present condition and outlook for hydrogen-natural gas blending technology. *Korean J. Chem. Eng.* **2022**, *39*, 251–262. [CrossRef]
4. Mahajan, D.; Tan, K.; Venkatesh, T.; Kileti, P.; Clayton, C.R. Hydrogen blending in gas pipeline networks—A review. *Energies* **2022**, *15*, 3582. [CrossRef]
5. European Industrial Gases Association, Hydrogen Pipeline Systems. Available online: https://www.eiga.eu/uploads/documents/DOC121.pdf (accessed on 10 October 2024).
6. Veziro, T.; Barbir, F. Hydrogen: The wonder fuel. *Int. J. Hydrogen Energy* **1992**, *17*, 391–404. [CrossRef]
7. Subani, N.; Amin, N.; Agaie, B.G. Hydrogen-natural gas mixture leak detection using reduced order modelling. *Appl. Comput. Math.* **2015**, *4*, 10. [CrossRef]
8. Edelstein, K.; FracTracker Alliance. *Latest Oil and Gas Incidents in the Gulf of Mexico*; FracTracker Alliance: Johnstown, PA, USA, 2014. Available online: https://www.fractracker.org/2014/11/latest-incident-gulf-of-mexico/ (accessed on 12 August 2024).
9. Adegboye, M.A.; Fung, W.-K.; Karnik, A. Recent advances in pipeline monitoring and oil leakage detection technologies: Principles and approaches. *Sensors* **2019**, *19*, 2548. [CrossRef] [PubMed]
10. Jiang, R.; Jiang, Y. Leak Detection and Localization of Gas Pipeline Network Based on a Steady State Model. In Proceedings of the 2014 Seventh International Symposium on Computational Intelligence and Design, Hangzhou, China, 13–14 December 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 10–13.
11. Mallat, S. *A Wavelet Tour of Signal Processing: The Sparse Way*; Academic Press: Cambridge, MA, USA, 2008.
12. Percival, D. *Wavelet Methods for Time Series Analysis*; Cambridge University Press: Cambridge, UK, 2000.
13. Koza, J.R.; Bennett, F.H.; Andre, D.; Keane, M.A. Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In *Artificial Intelligence in Design'96*; Springer: Dordrecht, The Netherlands, 1996; pp. 151–170.
14. Song, Y.; Li, S. Gas leak detection in galvanised steel pipe with internal flow noise using convolutional neural network. *Process Saf. Environ. Prot.* **2021**, *146*, 736–744. [CrossRef]
15. Rashid, S.; Akram, U.; Khan, S.A. WML: Wireless sensor network based machine learning for leakage detection and size estimation. *Procedia Comput. Sci.* **2015**, *63*, 171–176. [CrossRef]
16. Zhou, X.; Tang, Z.; Xu, W.; Meng, F.; Chu, X.; Xin, K.; Fu, G. Deep learning identifies accurate burst locations in water distribution networks. *Water Res.* **2019**, *166*, 115058. [CrossRef] [PubMed]
17. Afebu, K.; Abbas, A.J.; Nasr, G.G.; Kadir, A. Integrated Leak Detection in Gas Pipelines Using OLGA Simulator and Artificial Neural Networks. In Proceedings of the Abu Dhabi International Petroleum Exhibition and Conference, Abu Dhabi, United Arab Emirates, 9–12 November 2015; SPE: Bellingham, DC, USA, 2015; p. D011S016R007.
18. Xu, W.; Huang, J.; Sun, L.; Yao, Y.; Zhu, F.; Xie, Y.; Zhang, M. An Improved Convolutional Neural Network for Pipe Leakage Identification Based on Acoustic Emission. *J. Mar. Sci. Eng.* **2024**, *12*, 1720. [CrossRef]
19. Zha, W.; Liu, Y.; Wan, Y.; Luo, R.; Li, D.; Yang, S.; Xu, Y. Forecasting monthly gas field production based on the CNN-LSTM model. *Energy* **2022**, *260*, 124889. [CrossRef]
20. Chen, J.; Wu, H.; Liu, X.; Xiao, Y.; Wang, M.; Yang, M.; Rao, Y. A real-time distributed deep learning approach for intelligent event recognition in long distance pipeline monitoring with DOFS. In Proceedings of the 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Zhengzhou, China, 18–20 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 290–2906.
21. Dinis, J.; Wojtanowicz, A.; Scott, S. Leak detection in liquid subsea flowlines with no recorded feed rate. *J. Energy Resour. Technol.* **1999**, *121*, 161–166. [CrossRef]
22. Potter, M.C.; Wiggert, D.C.; Ramadan, B.H. *Mechanics of Fluids SI Version*; Cengage Learning: Boston, MA, USA, 2012.
23. Rioul, O.; Duhamel, P. Fast algorithms for discrete and continuous wavelet transforms. *IEEE Trans. Inf. Theory* **1992**, *38*, 569–586. [CrossRef]
24. Bertrand, J.; Bertrand, P.; Ovarlez, J.P. Discrete Mellin transform for signal analysis. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Albuquerque, NM, USA, 3–6 April 1990; IEEE: Piscataway, NJ, USA, 1990; pp. 1603–1606.
25. Flandrin, P. Some aspects of non-stationary signal processing with emphasis on time-frequency and time-scale methods. In *Wavelets: Time-Frequency Methods and Phase Space*; Springer: Berlin/Heidelberg, Germany, 1989; pp. 68–98.
26. Goupillaud, P.; Grossmann, A.; Morlet, J. Cycle-octave and related transforms in seismic signal analysis. *Geoexploration* **1984**, *23*, 85–102. [CrossRef]
27. Salomon, D.; Motta, G. *Handbook of Data Compression*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2010.
28. Guan, L.; Du, Y.; Li, L. Wavelets in petroleum industry: Past, present and future. In Proceedings of the SPE Annual Technical Conference and Exhibition, Houston, TX, USA, 26–29 September 2004; SPE: Bellingham, DC, USA, 2004; p. SPE-89952-MS.
29. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–9.
30. Iandola, F.N. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.

31. Mejia, A.H.; Brouwer, J.; Mac Kinnon, M. Hydrogen leaks at the same rate as natural gas in typical low-pressure gas infrastructure. *Int. J. Hydrogen Energy* **2020**, *45*, 8810–8826. [CrossRef]
32. Kim, J.; Chae, M.; Han, J.; Park, S.; Lee, Y. The development of leak detection model in subsea gas pipeline using machine learning. *J. Nat. Gas Sci. Eng.* **2021**, *94*, 104134. [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.