Advances in Technology Innovation, vol. 7, no. 3, 2022, pp. 169-180

Developing and Implementing an AI-Based Leak Detection System in a Long-Distance Gas Pipeline

Te-Kwei Wang[1], Yu-Hsun Lin[2,*], Jian-Yuan Shen[1]

[1]Department of Electrical Engineering, Ming Chi University of Technology, New Taipei City, Taiwan

[2]Department of Business and Management, Ming Chi University of Technology, New Taipei City, Taiwan

Received 12 November 2021; received in revised form 13 April 2022; accepted 14 April 2022

DOI: https://doi.org/10.46604/aiti.2022.8904

Abstract

This research proposes an artificial intelligence (AI) detection model using convolutional neural networks (CNN) to automatically detect gas leaks in a long-distance pipeline. The change of gap pressure is collected when leakage occurs in the pipeline, and thereby the feature of gas leakage is extracted for building the CNN model. The gas leak patterns in the long-distance pipeline are analyzed. A pipeline detection model based on AI technology for automatically monitoring the leaks is proposed by extracting the feature of gas leakage. This model is tested by collecting gas pressure data from an existing natural gas pipeline system starting from Mailiao to Taoyuan in Taiwan. The testing result shows that the reduced model of leak detection can be used to detect the leaks from the upstream and downstream pipelines, and the AI-based pipeline leak detection system can obtain a satisfactory result.

Keywords: artificial intelligence, convolutional neural network, pipeline leak, leak detection

1. Introduction

Pipeline transportation systems are generally used to transport oil, natural gas, domestic water, etc. The long-distance transportation system may encounter leak problems due to old pipelines. Also, thieves may destroy pipelines and steal the resources from the pipelines. Thus, leak detection is a critical issue for effective pipeline management. Leak detection techniques based on acoustics and infrared (IR) are the most widely used techniques for the detection of liquid and gas leaks, respectively. However, these two techniques do not work sometimes [1]. For instance, IR approaches cannot accurately detect leaks in wet weather, while acoustic sensors may not accurately detect gas leaks in the target area because of noise [1]. Due to the need to purchase and install new sensors, the additional cost of buying hardware equipment may be incurred.

In recent years, many methods have been proposed based on artificial intelligence (AI) techniques for developing leak detection systems [2-3]. For example, Yang and Zhao [2] used an optimally pruned extreme learning machine (OPELM) to improve the accuracy of the pressure point analysis (PPA), and used bidirectional long-short term memory (BiLSTM) to construct a leak detection system [2]. Zhou et al. [3] used improved spline-local mean decomposition (ISLMD) to analyze the internal pressure value of pipelines. They converted the data into an image, employed AlexNet to build a model to determine the occurrence of leakage, and used a cross-correlation function to calculate the time difference between upstream and downstream sensor values to find the location of the leakage point [3].

A growing number of AI techniques for developing leak detection systems are based on convolutional neural networks (CNNs) [3-5]. CNNs are based on neurons arranged in layers and can therefore learn hierarchical representations; moreover, weights and biases link neurons from one layer to the next [6]. The first layer acts as the input layer for receiving the

* Corresponding author. E-mail address: yslin@mail.mcut.edu.tw

Tel.: +886-2-29089899#3168; Fax: +886-2-29084533

transformed vector from image-based data, e.g., remote leak data. The last layer is the output, e.g., predicting whether a pipeline leaks or not. The feature derived from the previous layer serves as an input for the following layer. The final layer predicts an outcome according to the detected features. Layers between the first and the last layer are hidden layers that transform the feature values from the input to the output. Traditionally, CNNs consist of at least one convolutional layer as a hidden layer for exploiting spatial patterns [6].

Melo [4] used gradient-weighted class activation mapping algorithm (Grad-CAM) and CNN models to judge whether the images recorded by closed-circuit television (CCTV) have a leak response; the experimental results reached an accuracy of 99.78%. Li et al. [5] proposed a method for small-scale natural gas pipeline leak detection. The model converts the value of the acoustic sensor into the frequency domain and then uses a one-dimensional CNN to train the model. The final trained model has higher accuracy than the leakage detection performance of the traditional two-dimensional CNN model [5]. Kang et al. [7] used Ensemble CNN-SVM for leak detection in a water distribution system, and used a graph-based algorithm to determine the leak location according to the time difference of sensor value changes. The experimental results can reach 99.3% of the leak detection accuracy rate and control the leakage point position error within 3 meters [7].

As shown by the literature review above, employing AI techniques to develop leak detection has received growing attention. However, few studies have investigated AI-based leak detection systems based on CNN for automatically monitoring the leaks in a real context. To fill the gap, the present study aims to develop an AI-based model to detect the leaks of a long-distance pipeline and test the model's accuracy in a real context.

In this research, pressure sensors (the inherent hardware equipment of a pipeline) are used to read the gas pressure in the pipeline, and the collected gas pressure data are used for feature extraction and model training. Additionally, the present study employs the pipeline detection data from supervisory control and data acquisition (SCADA) as the training data for deep learning. By doing so, the smart detection of pipeline leaks can be realized, and the hardware cost of the sensor can be saved. Specifically, this research uses the pressure sensor data recorded in SCADA to analyze and extract suitable data features, and then labels the data features separately and conducts model training through CNN. The model trained in this study can realize a rapid and accurate function of leak detection. The developed system is beneficial for managers to detect leaks in the long-distance pipeline.

2. System Structure

This research obtains the dataset from the pressure sensors in the SCADA system provided by a petrochemical manufacturer in Taiwan. Fig. 1 depicts the schematic diagram of this study by showing the pipeline transportation system. PT is a pressure gauge. The pipeline transportation starts from Plant A to Plant C via Plant B. Plant C is the end of the entire pipeline transportation system. The distance between Plant A and Plant B is about 63 kilometers. The distance between Plant B and Plant C is about 136 kilometers. The pipeline in Plant B will be pressurized by pumps before sending resources to Plant C. In this study, the pressure sensor, PT-M621, in Plant B is used to collect data for subsequent data analysis and model training, and the pressure sensor, PT-M622, is used to test and validate the model.

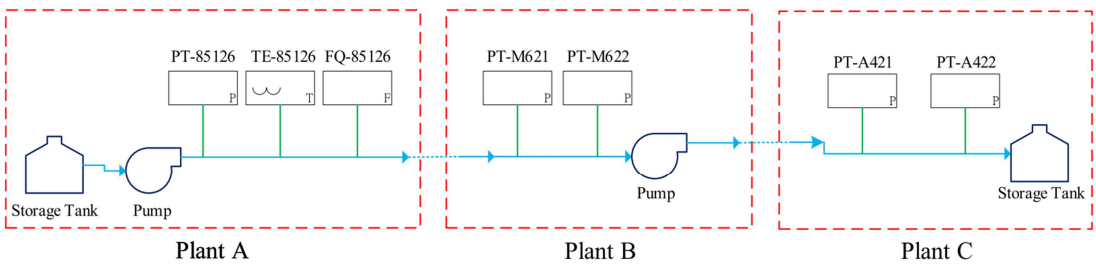


Fig. 1 Schematic diagram of this study

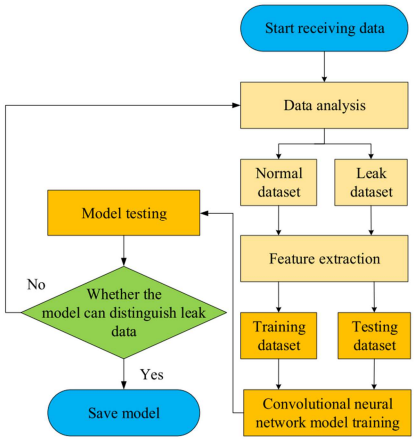


Fig. 2 System architecture

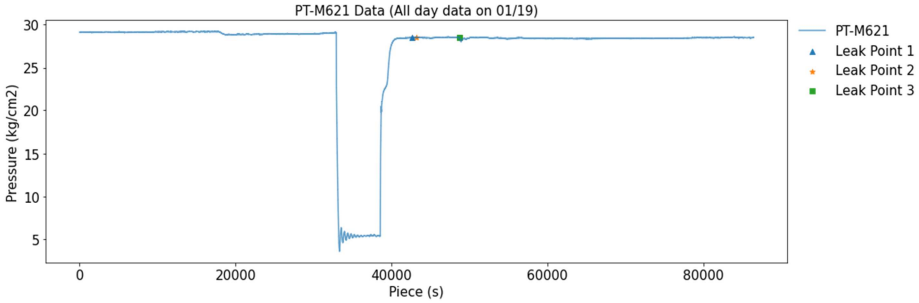


Fig. 3 Data from the real pipeline leak test

The research system architecture is shown in Fig. 2. First, the data received from the SCADA system is classified into the normal data group and the leak data group. Next, the characteristics of the two data groups are extracted through feature engineering. Therefore, the two groups' data are merged and divided into the training data set and the test data set for model training. Finally, this study tests whether the model can distinguish the leak data. Overall, this study employs the CNN architecture to construct the model. If the model cannot accurately classify the two types of data, the feature engineering needs to be adjusted until the model can effectively classify the two kinds of data.

Fig. 3 shows the full-day data on the day of the actual leak test. The time points of the three leak tests for data analysis and feature engineering are marked in Fig. 3. The X-axis presents the time point of measuring the pressure. The time interval between pieces is one second.

2.1. Data analysis and feature extraction

Fig. 4 illustrates the data sampling for the leak test. For observing the data variance in detail, the data for analysis is captured after the start of the leak test. In Fig. 4, the positions pointed to by the red arrow are the data points during the leak test; through data visualization, there are obvious pressure fluctuations when leakage occurs. This study uses these three leakage points as reference points to find more suitable data for feature engineering.

Although the leakage appears at these points, the data of the leak pattern for extracting the feature cannot be reflected by these points. Therefore, according to the rule of thumb for capturing the leak feature found in this study, 1000 pieces of data are taken before and after the leakage points to capture the graphical pattern of leak occurrence, as illustrated in Fig. 4. The time interval for the pressure data sampling between pieces is one second. In doing so, this study does observe and find the data suitable for presenting the characteristics of the leakage data.

In practice, the operators test the functionality of the pressure sensor by opening and then closing the valve after 30 seconds and four minutes to simulate the gas leaks. Fig. 5(a) and Fig. 6(a) show the data collected from the interval between opening the leak valve instantly and waiting for 30 seconds to close the leak valve instantly, respectively. The curves marked

with red squares in the two figures are the pressure fluctuations during the leak test. As shown in Fig. 5(a) and Fig. 6(a), the width of the red region is the period for the opening and closing of the valve and is enlarged in Fig. 5(b) and Fig. 6(b). The pressure variance of the first leak test and the second leak test are shown in Fig. 5(b) and Fig. 6(b), respectively. As depicted in Fig. 6(a), the internal pressure of the pipeline does not return to a stable state during the second leak test. Therefore, the level of pressure fluctuation in the second test is smaller than in the first test.

Fig. 7(a) shows the data collected from the pressure sensor during the third leak test. In the third leak test, the data collection process begins when the leaking valve is opened at a slower speed and ends when the valve is fully opened. The time interval for the opening and closing of the valve is four minutes. After four minutes, the leakage valve is closed at a slow speed. The width of the red region in Fig. 7(a) is the period for the opening and closing of the valve and is enlarged in Fig. 7(b). Thus, the pressure variance of the third leak test can be clearly seen in Fig. 7(b).

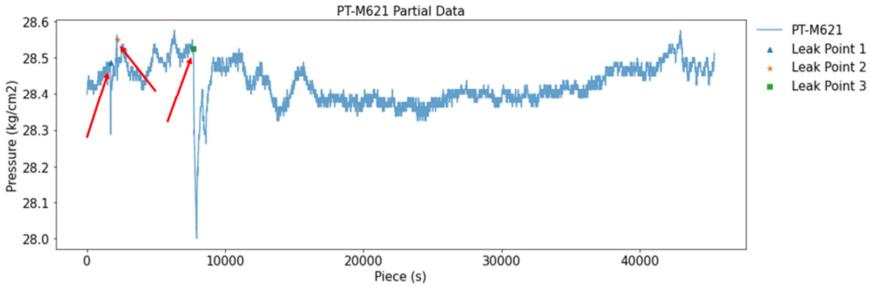
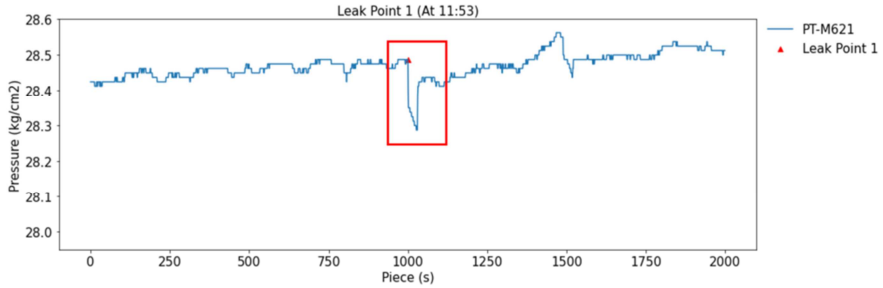
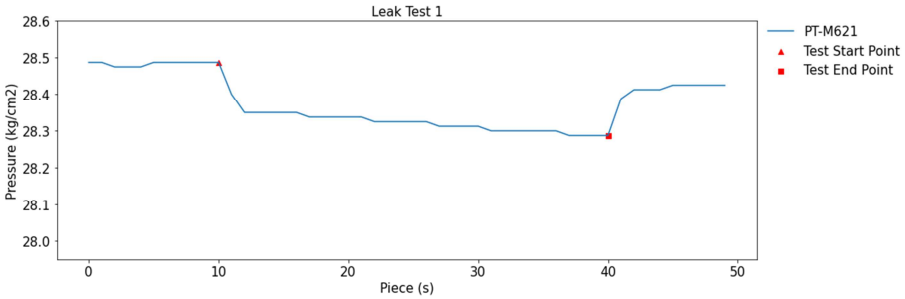


Fig. 4 Illustration of the pressure data for detecting real pipeline leaks

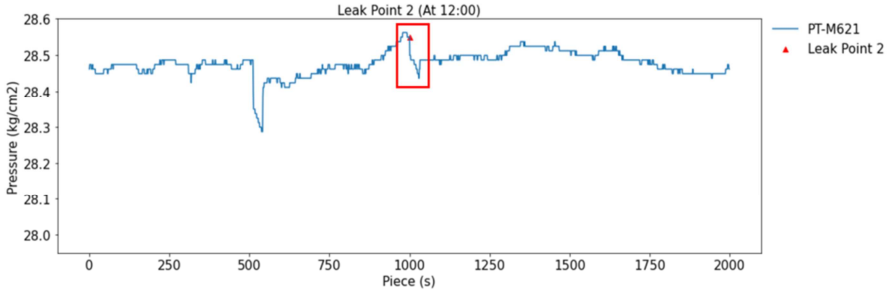


(a) Result of the first leak test showing leak point 1



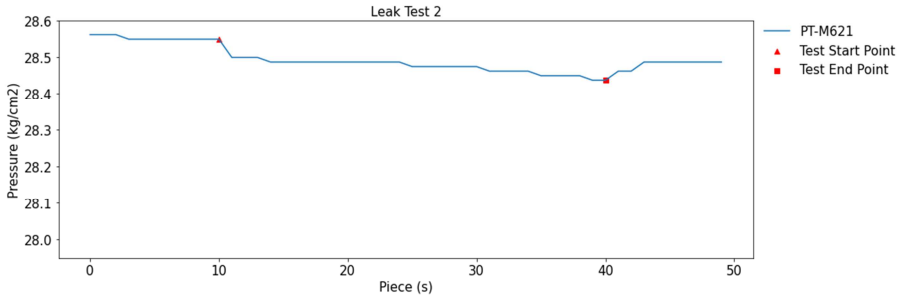
(b) Close-up view of the red region in Fig. 5(a) (showing the test interval)

Fig. 5 The first leak test



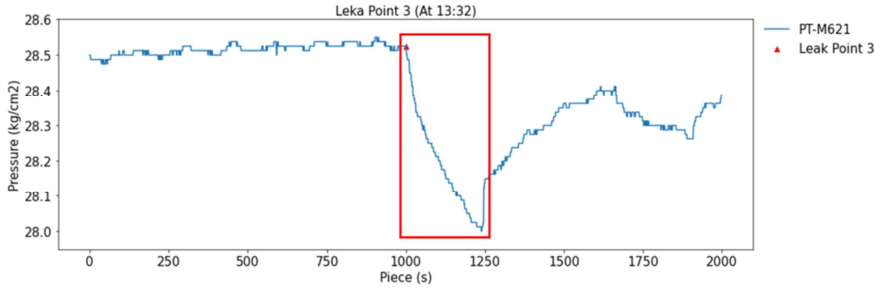
(a) Result of the second leak test showing leak point 2

Fig. 6 The second leak test

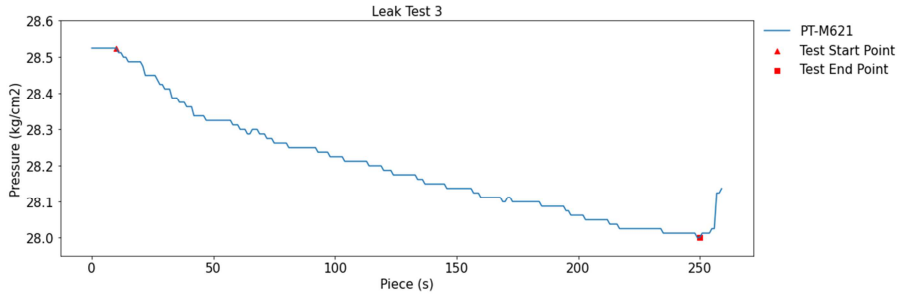


(b) Close-up view of the red region in Fig. 6(a) (showing the test interval)

Fig. 6 The second leak test (continued)



(a) Result of the third leak test showing leak point 3



(b) Close-up view of the red region in Fig. 7(a) (showing the test interval)

Fig. 7 The third leak test

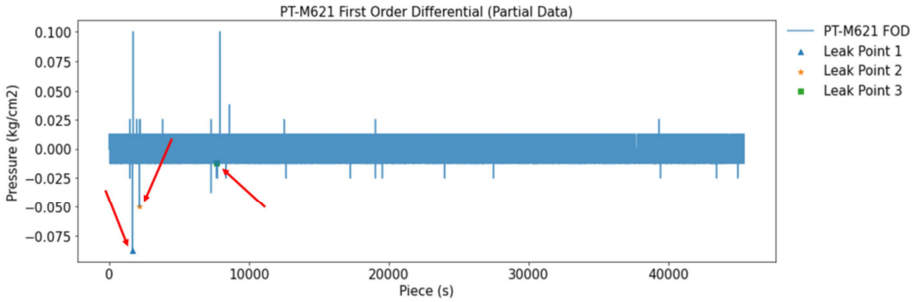
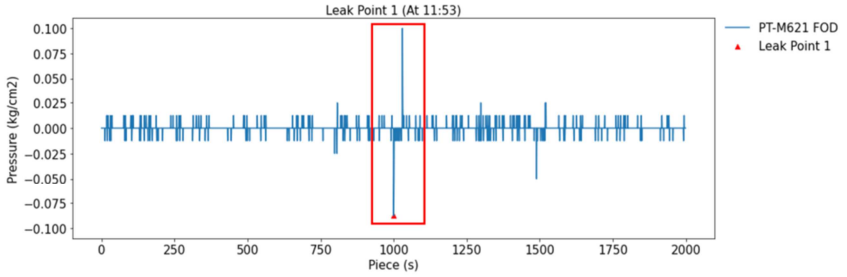


Fig. 8 FOD processing result for the data of real pipeline leaks

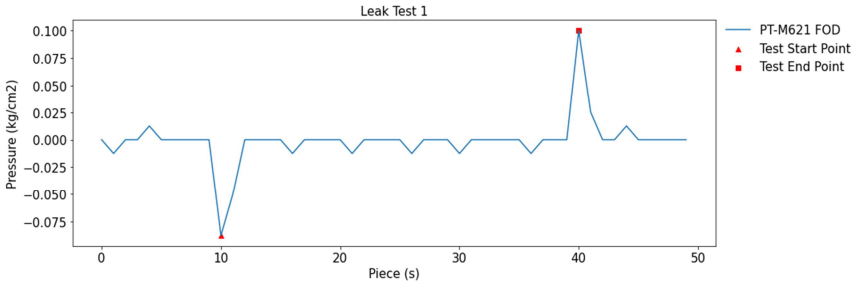
After training the model many times, even if the number of the convolutional layers, the number of the filters, and the size of the convolution kernel are increased, the resisting noise interference does not reach the requirement. The result presents that a single feature is less effective in resisting noise interference. Thus, the data is processed by the first-order differential (FOD) processing and its characteristics are analyzed. As shown in Fig. 8, the red arrows point to the data points of the real leak test. When observing the curve of the data analysis after the FOD processing, it can be understood that relatively large pressure fluctuations will be generated in the interval of the leak test. Thus, these features are adopted as the second feature of model training.

As shown in Fig. 9, Fig. 10, and Fig 11, 1000 points of data extracted from the interval of the first, second, and third leak tests are used to analyze the features. Fig. 9(a), Fig. 10(a), and Fig. 11(a) are the FOD data from the first leak test, the second leak test, and the third leak test, respectively. The width of the red region in Fig. 9(a), Fig. 10(a), and Fig 11(a) is the period for

the opening and closing of the valve and is enlarged in Fig. 9(b), Fig. 10(b), and Fig 11(b), respectively. Taken together, after the data is transformed by the FOD processing, the pressure fluctuation produced by the first leak test is more obvious than the other two. Therefore, the FOD data of the first leak test is selected as the second feature of the training model of this research.

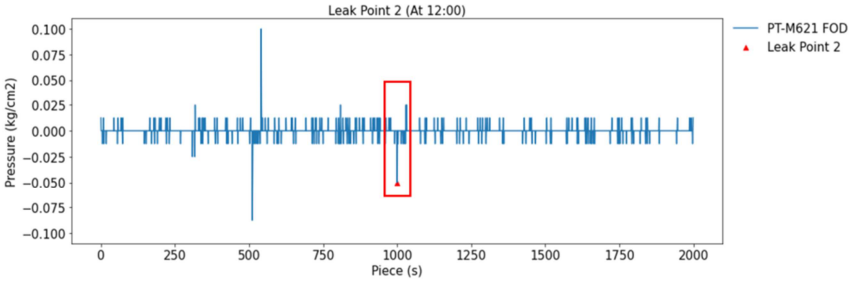


(a) FOD processing result of the first leak test showing leak point 1

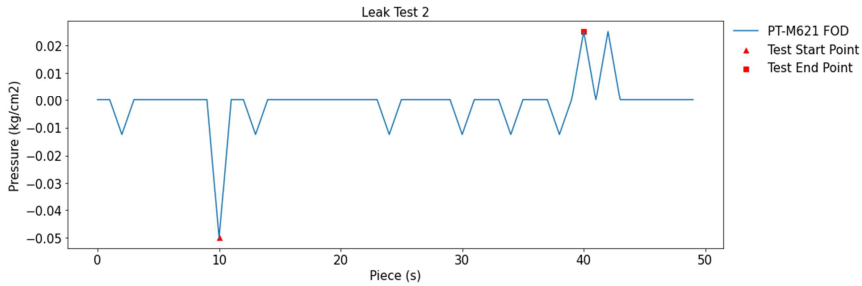


(b) Close-up view of the red region in Fig. 9(a) (showing the test interval)

Fig. 9 First-order data differentiation for the first leak test

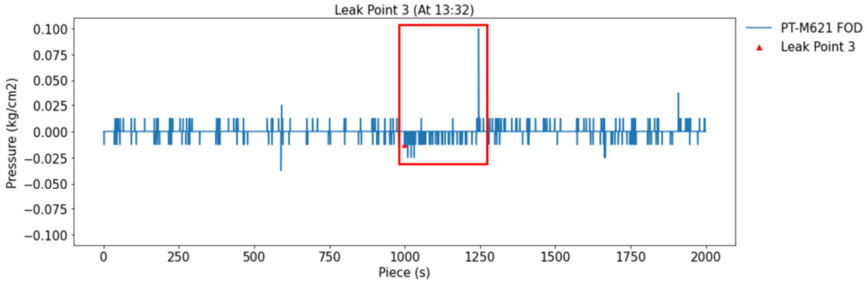


(a) FOD processing result of the second leak test data showing leak point 2



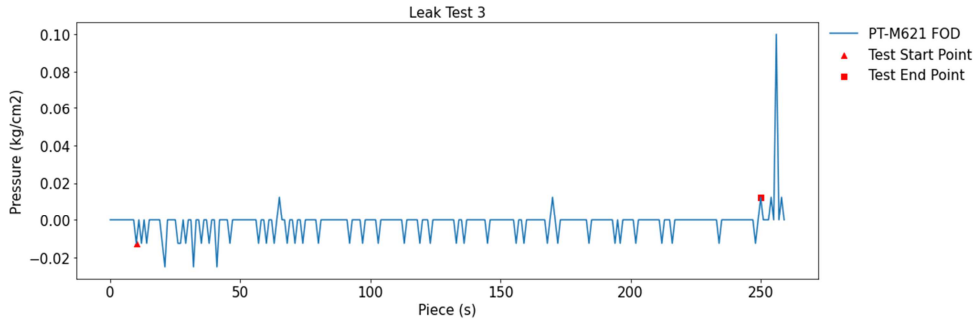
(b) Close-up view of the red region in Fig. 10(a) (showing the test interval)

Fig. 10 First-order data differentiation for the second leak test



(a) FOD processing result of the third leak test data showing leak point 3

Fig. 11 First-order data differentiation for the third leak test



(b) Close-up view of the red region in Fig. 11(a) (showing the test interval)

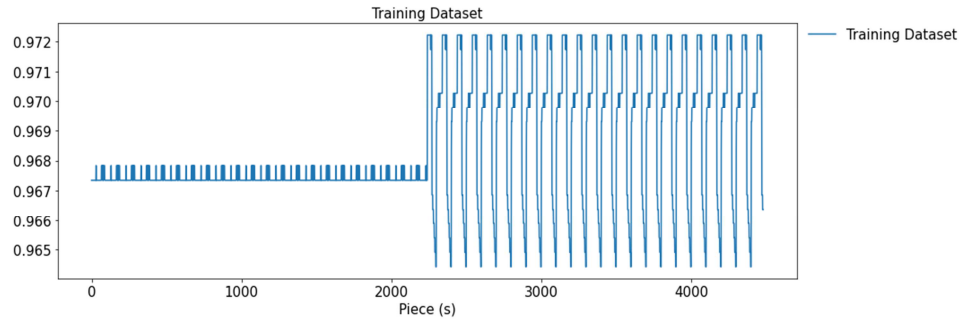
Fig. 11 First-order data differentiation for the third leak test (continued)

2.2. Model training data set and testing data set

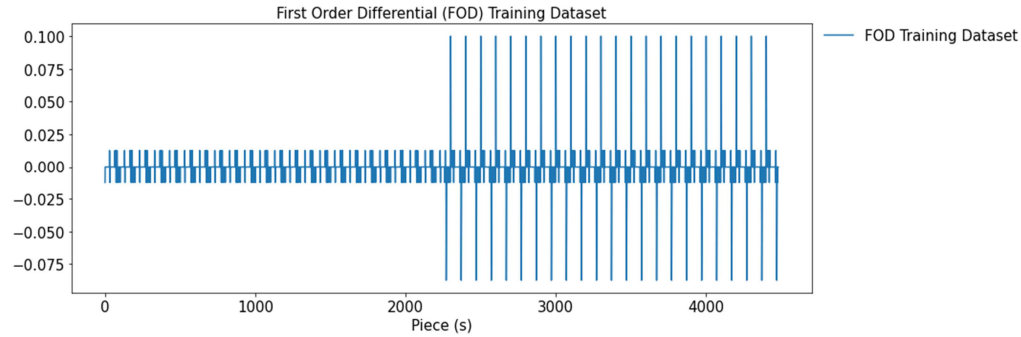
According to the analysis result, a suitable data interval for identifying the features of the model training is found. The normal data and the leak data are given corresponding labels. As presented in Figs. 12(a) and 13(a), the data obtained from the pressure sensors are normalized to make the value with the range from 0 to 1, so the data in the two figures are not marked with the X-axis unit. Because the feature amount of the leak data is less than the feature amount of the normal data, data expansion is needed to achieve better training performance.

In order to avoid the model from only learning the characteristics of a single data and causing identification failure, the data set is balanced. Finally, 70% of the normal data and the leak data are used as the training data. The rest of the data is used as the test data set. The result of experimental tests shows that if the data set is randomly arranged, it will reduce the accuracy of the overall model classification. Thus, this study orders the data according to the same characteristics and by the arrangement method.

As shown in Fig. 12(a) and Fig. 12(b), the first half of the training data set is normal data, and the second half is leak data. Fig. 13(a) and Fig. 13(b) are the testing data sets, which are used to evaluate the effect of the training model during model training.

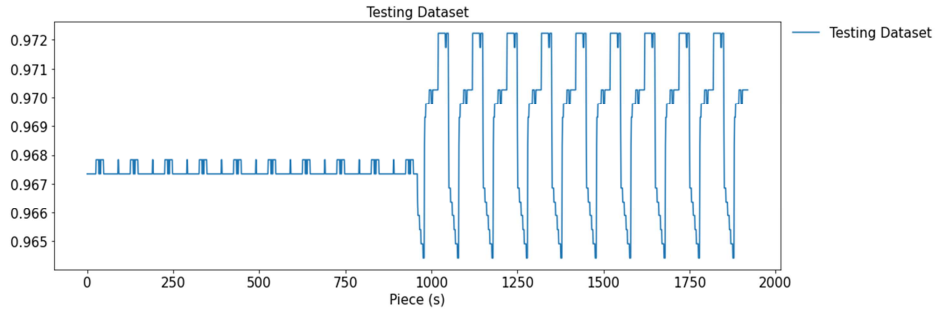


(a) Original training data set

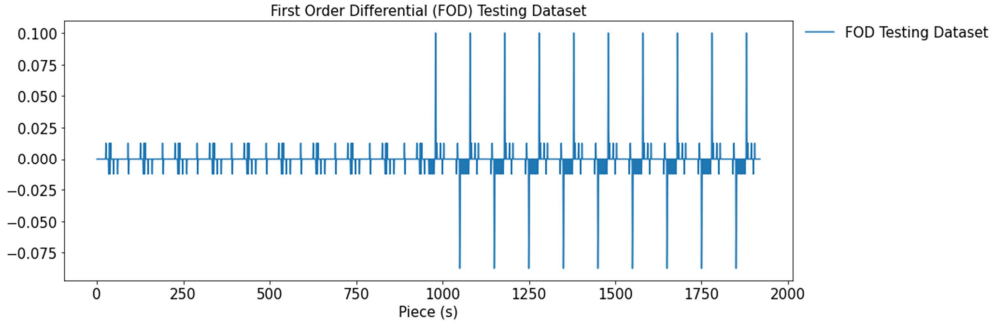


(b) FOD training data set

Fig. 12 Training data set



(a) Original testing data set



(b) FOD testing data set

Fig. 13 Testing data set

2.3. *Model building*

This study uses Keras to construct a CNN model. The values collected from the pressure sensor are time series and belong to a one-dimensional data type. Therefore, a one-dimensional convolutional layer is used to build the model, and ReLU is selected as the activation function. To avoid the over-fitting of the model on the training data and reduce the generalization of the model, this study adds a dropout layer after the convolutional layer. Furthermore, through adding the pooling layer, the dimensions of the model features are reduced and output to the full connection layer. In the full connection layer, Softmax is used as the activation function. The loss function for the evaluation model is categorical cross-entropy, and the function for the optimizer is the Adam function. Fig. 14 depicts the flowchart of model building in this study.

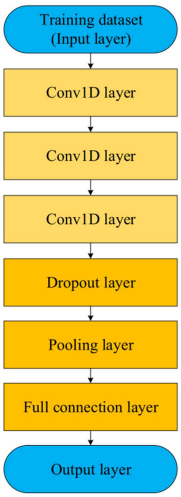


Fig. 14 Model building

3. System Implementation

After confirming that the training model can identify the leak data, the training model is used to perform a leak test in the data interval, thus verifying whether the model can correctly determine the occurrence of leaks and effectively resist noise

interference. As shown in Fig. 15, if the result of the model test is different from the expected result, then the hyperparameters of the model are re-adjusted to meet the condition where the time interval of the leak occurrence is consistent with the time interval of the real leak test.

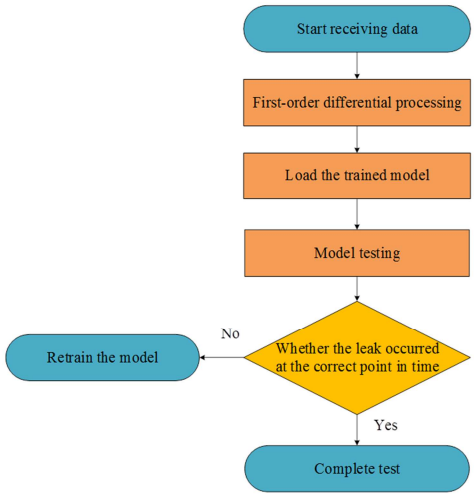


Fig. 15 System implementation

3.1. Model architecture

Fig. 16 shows the model architecture, in which the three layers of the one-dimensional convolutional layer are stacked, and the dropout layer is added to prevent the model from overly relying on the training data set. After repeated testing is conducted, the classification effect of using the average pooling layer is found to be better than the effect of the max pooling layer. Thus, the average pooling layer is used here.

Layer (type)	Output Shape	Param #
conv1d_146 (Conv1D)	(None, 1, 64)	704
conv1d_147 (Conv1D)	(None, 1, 128)	41088
conv1d_148 (Conv1D)	(None, 1, 256)	164096
dropout_42 (Dropout)	(None, 1, 256)	0
global_average_pooling1d_11	(None, 256)	0
flatten_33 (Flatten)	(None, 256)	0
dense_44 (Dense)	(None, 2)	514

Total params: 206,402
Trainable params: 206,402
Non-trainable params: 0

Fig. 16 Model architecture

3.2. Result of leak point identification

After completing the model training, the data of two different pressure sensors are employed to test whether the testing model can correctly distinguish and predict the point of pipeline leak where the leak actually occurs. The pressure sensor data shown in Fig. 16 is the data of the training model (PT-M621). In Fig. 16, if the model finds leakage, the leak point is one, and the normal data is zero. Moreover, in order to avoid the excessive dependence of the model on the training data set, the data from the differential pressure sensor (PT-M622) is used to verify the generalization of the model.

The model architecture used in Figs. 17-18 is the same as the one shown in Fig. 16. In these figures, the red line represents the zero value of the pressure. In the pooling layer, this study chooses the max pooling instead of the average pooling as the mean to gain the results of the model training. Fig. 17 and Fig. 18 use PT-M621 and PT-M622 as the data sources of the model verification, respectively. Fig. 18 shows that Model 1 can accurately detect the occurrence of leakage, while Fig. 17 can only

detect the signal of the leakage generated during the first leak test. Fig. 19 and Fig. 20 use PT-M621 and PT-M622 as the data sources of the test model, respectively. The architecture of this test model uses the average pooling instead of the max pooling in designing the pooling layer, and the size of the convolution kernel is 3. As shown in Fig. 19, Model 2 only identifies the first and third leak occurrences but does not identify the second leak test.

Fig. 20 shows that this model can accurately detect leakage in the interval where the leakage occurs. Fig. 21 and Fig. 22 use PT-M621 and PT-M622 as the data sources of the test model, respectively. The structure of the test model is shown in Fig. 16. The size of the convolution kernel is adjusted to 5, and the test results are shown in Fig. 21 and Fig. 22. As shown in Fig. 21 and Fig. 22, this model architecture can accurately detect leaks in the time interval of the leak test on two different data sets, which confirms that the model is with good generalization and can be applied to different data sets. Table 1 lists the result of testing leak data after adjusting the hyperparameters of the above models.

Finally, the model developed by this study is employed to test the data set from different sensors. The result indicates that this model can still play a role in leak detection when a real leak occurs, thus confirming that this model can well function, receive the accurate leak detection function, and achieve the level of generalization.

The model developed by this study is implemented with simple architecture. The time interval for the pressure sensor to record data is one second. Thus, the data volume is also equal to the sampling time of the data volume. If the real-time leak detection function is to be achieved, the time for calculating the data of the model must be controlled within one second. The calculating time on different data volumes in the model is shown in Table 2. The experimental results show that if the data recorded every hour, 3600 cases, is used for leak detection, the calculating time for the model to predict the leakage can be below one second, thus being able to satisfy the requirement of real-time leak detection.

Table 1 Results of testing the pipeline leakage with different hyperparameters

Model	Parameters	PT-M621	PT-M622
Model 1	Max pooling Kernel size = 3	Fail	Success
Model 2	Average pooling Kernel size = 3	Fail	Success
Model 3	Average pooling Kernel size = 5	Success	Success

Table 2 Calculating time of the model for different data volume

Data volume (cases)	Calculating time (second)
86400	3.339572191238403
43940	1.904508652279663
3600	0.636376142501831
600	0.476074934005737

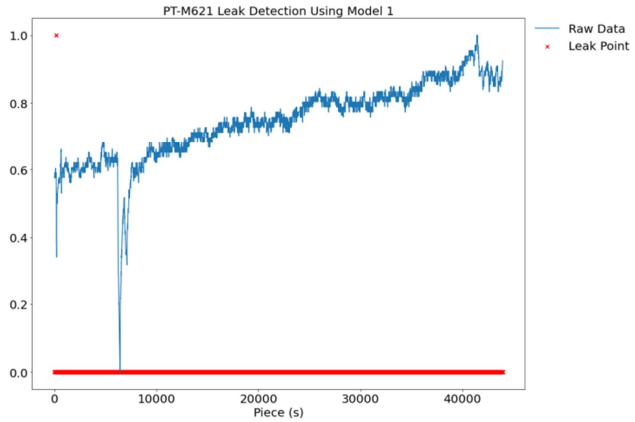


Fig. 17 Results of leak identification for Model 1 (using the PT-M621 pressure sensor)

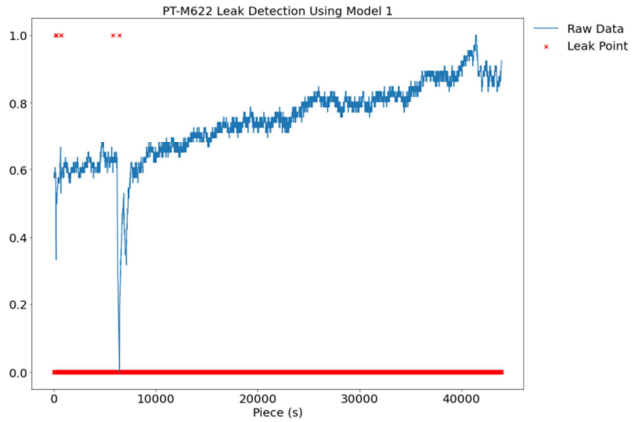


Fig. 18 Results of leak identification for Model 1 (using the PT-M622 pressure sensor)

Fig. 19 Results of leak identification for Model 2 (using the PT-M621 pressure sensor)

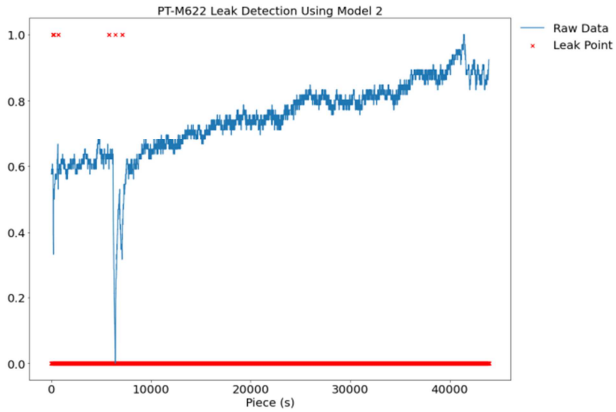


Fig. 20 Results of leak identification for Model 2 (using the PT-M622 pressure sensor)

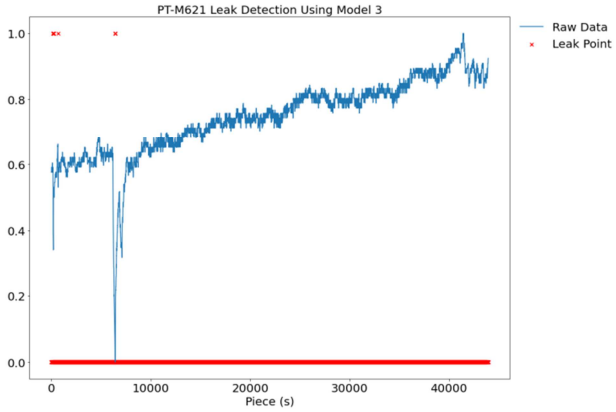


Fig. 21 Results of leak identification for Model 3 (using the PT-M621 pressure sensor)

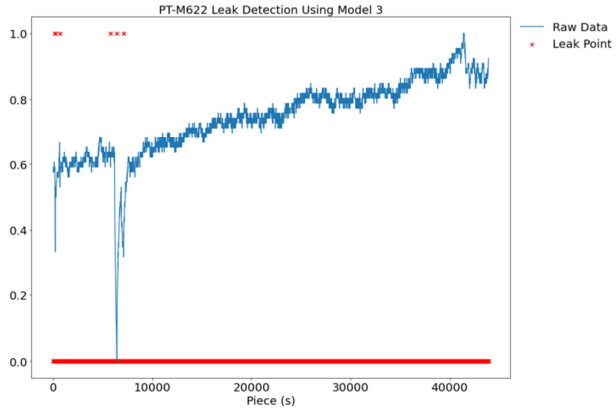


Fig. 22 Results of leak identification for Model 3 (using the PT-M622 pressure sensor)

The result can be summarized as follows. First, based on the physical features of the pipeline, the AI model of pipeline transportation is derived. Second, based on the AI model, the pipeline leak patterns are analyzed. It is found that the reduced model of detecting the pipeline leakage can be used to detect the leakage from the upstream and downstream pipelines. Finally, the model is tested in an existing natural gas pipeline system starting from Mailiao to Taoyuan in Taiwan. The results show that in the proposed AI-based leak detection system, the correctness concerning the detection function is not affected by various operational conditions of the pipelines.

4. Conclusions and Directions for Future Research

The present study contributes to the research and practice of pipeline leak detection in the following ways:

(1) The model developed by this study can effectively detect the leakage in the real long-distance transmission pipeline system. It can achieve the same performance as the traditional leak detection system.

(2) The training data used in this study is collected from the sensor of the original pipeline system. Thus, no additional hardware equipment is required, which can save the cost of hardware equipment.

(3) It can run smoothly without adding the extra graphics card.

The operator can easily load the trained model when performing leak detection. Nevertheless, some limitations remain and need to be resolved in further research:

(1) This study uses the pressure sensor data collected from a single site to train the model. However, the data recorded by the pressure sensor at different sites may be different. It will be better to collect the training data from different sites in the pipeline system in future research.

(2) The present study only uses the basic CNN architecture for model training and collects the data from one site in the pipeline. Future research can collect sensor values from other sites and employ RNN or LSTM algorithms for processing the time-series data.

(3) Only a few pressure data for pipeline leakage can be collected due to the expensive testing cost of the existing detection system. Thus, the issue of overfitting and false negatives may occur in the proposed model. More pressure data should be collected in further research.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] M. Meribout, et al., "Leak Detection Systems in Oil and Gas Fields: Present Trends and Future Prospects," Flow Measurement and Instrumentation, vol. 75, Article no. 101772, October 2020.
[2] L. Yang, et al., "A Novel PPA Method for Fluid Pipeline Leak Detection Based on OPELM and Bidirectional LSTM," IEEE Access, vol. 8, pp. 107185-107199, 2020.
[3] M. Zhou, et al., "Leak Detection and Location Based on ISLMD and CNN in a Pipeline," IEEE Access, vol. 7, pp. 30457-30464, 2019.
[4] R. O. Melo, et al., "Applying Convolutional Neural Networks to Detect Natural Gas Leaks in Wellhead Images," IEEE Access, vol. 8, pp. 191775-191784, 2020.
[5] J. Li, et al., "A Small Leakage Detection Approach for Gas Pipelines Based on CNN," 2019 CAA Symposium on Fault Detection, Supervision, and Safety for Technical Processes, pp. 390-394, July 2019.
[6] T. Kattenborn, et al., "Review on Convolutional Neural Networks (CNN) in Vegetation Remote Sensing," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 173, pp. 24-49, March 2021.
[7] J. Kang, et al., "Novel Leakage Detection by Ensemble CNN-SVM and Graph-Based Localization in Water Distribution Systems," IEEE Transactions on Industrial Electronics, vol. 65, no. 5, pp. 4279-4289, May 2018.