International Journal of Robotics and Control Systems
Vol. 4, No. 1, 2024, pp. 312-333
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Oil Pipeline Leak Detection in Iraqi Oil Fields based on 1DCNN

Mustafa Raad Al-Khalidi [a,1,*], Ahmad Taha Abdulsadda [a,2], Mudafeer Sadaq Al Zuhryi [b,3]

[a] Department of Communication Engineering, Al-Furat Al-Awsat Technical University, Iraq
[b] Chancellor of Al-Furat Al-Awsat Technical University, Iraq
[1] mustafa.ms.etcn36@student.atu.edu.iq; [2] coj.abdulsad@atu.edu.iq
* Corresponding Author

ARTICLE INFO

Article history
Received January 11, 2024
Revised March 09, 2024
Accepted March 18, 2024

Keywords
Oil Pipelines;
Leak Detection;
Leak Localization;
Mathematical Model;
1DCNN

ABSTRACT

The oil industry plays a crucial role in Iraq's economy. There's a growing need for technologies that can quickly detect leaks in oil pipelines because leaks can have serious ramifications, including monetary losses, endangerment to public safety, environmental degradation, and resource waste. Advances in technology and software have made it possible to detect leaks. Current approaches often require manual extraction of features, which can be slow and inefficient. This paper presents a new method that proposes using convolutional neural networks (CNNs) for automatic feature extraction. The Iraqi Ministry of Oil, specifically the Basra Oil Company, provided the dataset, such as total distance (km), pressure (bar), and flow rate (STB/d). We split the data into training (70%) and testing (30%) sets. then we calculate metrics such as confusion matrices, accuracy, precision, recall, and F-score to evaluate performance and calculate errors from the regression analysis (root mean square error, root mean absolute error, and relative error). Our contribution to this work is to use 1DCNN to identify leaks, pinpoint their location, and even predict the amount of spilled oil, unlike other research that only uses it to evaluate the presence or absence of a leak only. Additionally, we've created a user-friendly interface for the system. Finally, compare the proposed approach with conventional and alternative methods to show its efficiency. In the future, we plan to expand the system to assess pipeline corrosion and predict its remaining lifespan.

1. Introduction

Pipelines are a significant method for transporting petroleum products, including fossil fuels, gases, chemicals, and other important hydrocarbon fluids that contribute to the nation's economy [1]. Research has shown that oil and gas pipeline networks are the most cost-effective and secure method for moving crude oils, meeting a significant need for effectiveness and dependability [2], [3]. For example, the projected fatalities from accidents per ton-mile of transported petroleum products are 87%, 4%, and 2.7% higher when using trucks, ships, and trains, respectively, compared to pipelines [4]. Transporting hazardous chemicals over extensive pipes has been a common practice worldwide in recent decades, leading to a higher risk of severe incidents caused by pipeline breakdowns [5]. Deliberate actions (such as vandalism) or accidental incidents (such as device or material failure and corrosion) can cause failures [6], [7], leading to pipeline failure and irreversible consequences like financial losses and severe environmental pollution, especially if the leakage is not promptly detected [8], [9].

The economic impact of pipeline leakages is substantial [10]. Over the last 30 years, pipeline incidents in the USA have resulted in about $7 billion in property damage, more than 500 fatalities, and thousands of injuries [11]. An example is the pipeline explosion in San Bruno, California, USA, on September 6, 2010, which resulted in the deaths of eight people and injuries to over fifty more [11]. On July 26, 2010, a pipeline fault in Michigan, USA, resulted in about 840,000 gallons of crude oil leaking into the Kalamazoo River, causing an estimated $800 million in damages [11]. There are several reasons for pipeline degradation. Fig. 1 displays a pie chart depicting the primary reasons for pipeline failures, such as pipeline corrosion, human carelessness, installation and construction problems, manufacturing issues, and external influences [12].

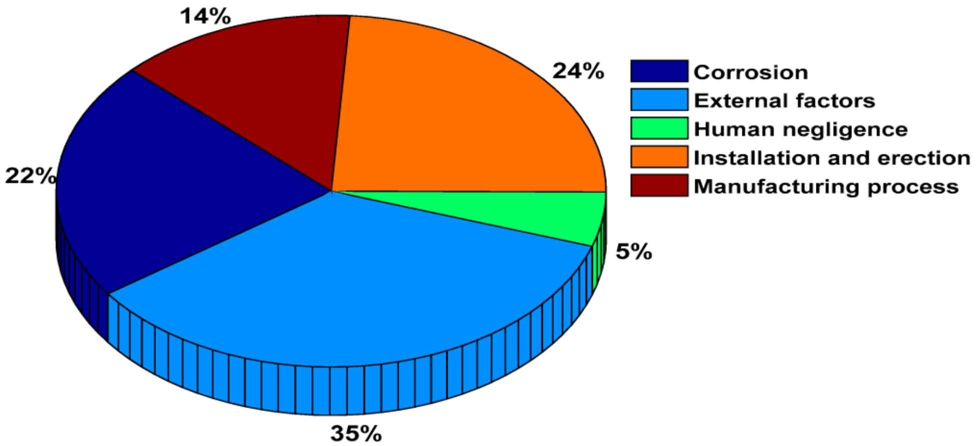

Fig. 1. displays a pie chart showing the statistics of pipeline failure origins. Information is acquired in reference [12]

Given these facts, it is difficult to completely prevent the occurrence of pipeline leaks due to the wide range of potential failure causes. It is essential to monitor pipelines to promptly discover and forecast leaks, enabling swift actions to prevent oil spills and ensure adequate pipeline upkeep, thereby minimizing the societal repercussions of oil spillage. Therefore, it is feasible to decrease the rate of loss, injuries, and other significant societal and environmental impacts caused by pipeline failures.

Various pipeline leak detection technologies have been suggested during the last few decades, using diverse functioning principles and methodologies. Current methods for detecting leaks include acoustic emission [13]-[15], fiber optic sensors [16]-[18], ground penetration radar [19], [20], negative pressure wave [21]-[23], pressure point analysis [24]-[26], dynamic modeling [27], [28], vapor sampling, infrared thermal imaging, digital signal processing, and mass-volume balance [29]-[33]. These approaches have been categorized using different frameworks. The authors have categorized them into two groups: hardware and software-based techniques [34], [35]. Efforts have been made to categorize these approaches based on technical aspects [36]-[39], resulting in the categorization of leakage detection systems into three main groups: internal, non-technical or non-continuous, and external methods. Fig. 2 displays a comprehensive categorization of various strategies. Man-made sensing technologies are used in the external approach to identify issues outside pipes. The biological technique uses trained animals or experienced workers to identify leaks using visual, aural, and olfactory sensors. The inside approach utilizes software-based technologies that use advanced computational algorithms together with sensors to monitor the internal pipeline environment for detection purposes. Remote monitoring may be accomplished by deploying cameras or sensing equipment to specific places using methods such as smart pigging, helicopters, autonomous underwater vehicles (AUVs), drones, or sensor networks.

One of the software-based technologies is the convolutional neural network (CNN), an extensively used technique in pattern recognition that will be used to identify various aperture leaks. CNN has strong generalization and accuracy in multi-classification tasks. One-dimensional Convolutional Neural Networks (1DCNN) provide clear benefits for processing one-dimensional data signals [40]-[42]. CNN accepts datasets as input, eliminating the need for subjective and difficult data feature extraction and minimizing human involvement in feature extraction and learning. CNN is used for feature extraction in pipeline leaks due to its proven strong performance in machine learning for object identification, fault diagnosis, and similar activities.

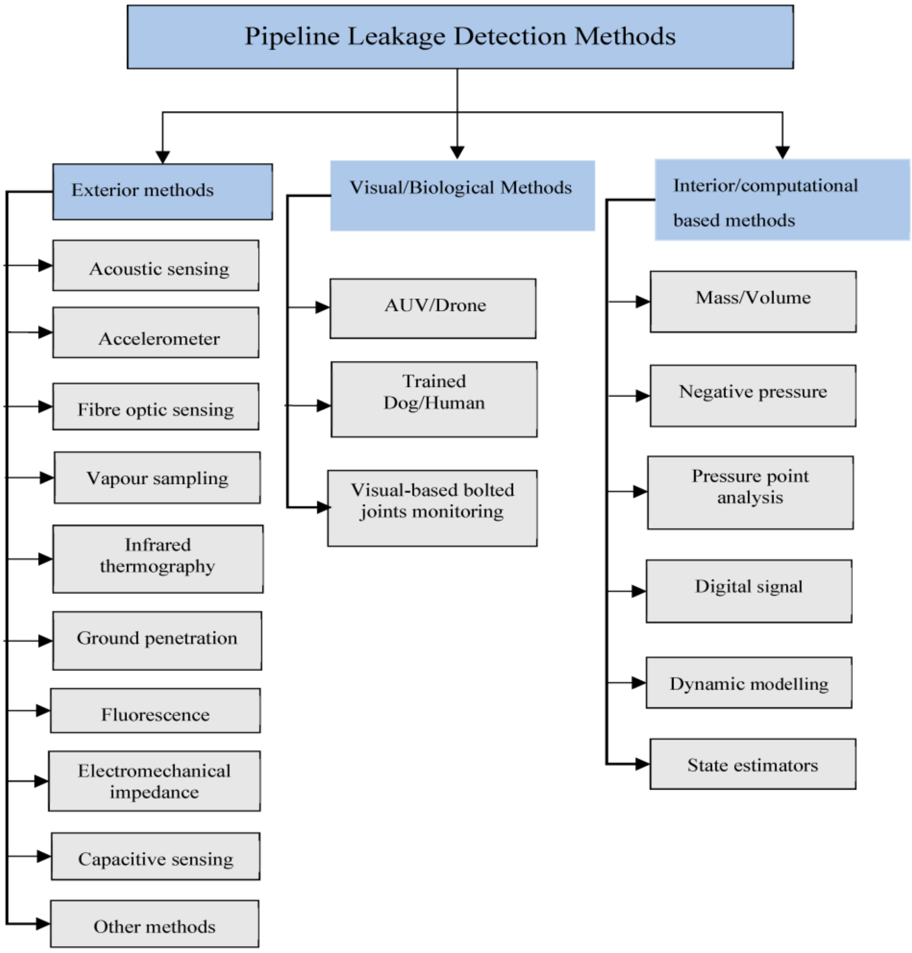

Fig. 2. Flowchart depicting several methods for detecting pipeline leaks

CNNs is a powerful tool for automatically extracting features from an input dataset. This study, however, aims to detect leaks by monitoring pressure drop and flow rate in oil-bearing pipelines, with a special emphasis on accurate leak detection and location, effectively developing and training a one-dimensional convolutional neural network (1DCNN) for this purpose. The data needed for this algorithm comes from real sources, like the Iraqi Ministry of Oil. The dataset was improved by using PIPESIM, a program that simulates fluids inside oil pipelines. Several leakage scenarios were created on purpose, and then all of the system's evaluation criteria were worked out. Regression predicted the leak location, estimated the fluid passing through the leak orifice, and calculated all error rates. After all the classification and regression of our model, we finally designed a graphical user interface that is user-friendly.

The contributions of this paper are summarized as follows: Oil and gas pipelines (OGPs) are a secure and cost-effective means of transporting petroleum products globally, although they encounter obstacles due to risk considerations. The hazards involved include safety, security, design, construction, and operational risks caused by third-party disruption (TPD) and acts of terrorism, especially in developing and unstable nations such as Iraq. The lack of understanding of how to handle these risks, as well as the limited availability of historical data on pipeline failures, are impeding Open Government Partnership (OGP) risk management solutions.

And as a result of developments in the field of artificial intelligence in all activities. Outlined below are the contributions of this study:

1. Reducing maintenance costs in the industry and minimizing the impact of oil spills on the environment requires an intelligent method to locate pipeline leaks and accurately estimate their extent.
2. Non-destructive techniques, such as using pressure and flow sensor data in conjunction with a deep learning technique, can be used to determine if there is a leak in an oil pipeline and locate it.
3. The proposed approach uses the input sensor data to accurately detect and pinpoint the leak location.
4. The deep learning model's two-fold decline curve was verified using the data collected from the breach.

Here is the organization of the remaining sections: Section 2 present the theoretical background, and this section will provide clarification on three topics: Deep Learning, CNN, and 1DCNN, methods for detecting leaks in pipelines, and the most critical parameters to be calculated. Section 3 displays all mathematical models and governing equations associated with pipeline leak detection. Section 4 present the method that will be used to detect and locate pipeline leaks, including the data collection process as well as its processing within the convolutional neural network and system evaluations based on the parameter outputs. Section 5 we present all operations were carried out for both classification and regression to classify the presence and absence of a leak, as well as determining the location of the leak and the amount of leaked fluid and displaying all the mathematical equations on the basis of which these operations are evaluated. Section 6 In this part, the results for both classification and regression were presented, as well as the model graphical interface for the designed application. Section 7 In this section, all the limitations of the work were presented, then compared with other works, and all the challenges that were faced during the implementation were discussed, as well as special suggestions for future work.

2. Theoretical Background

This section briefly discusses deep learning, the convolution neural network (CNN) layers, and one-dimensional convolutional neural networks (1D CNN) with all evaluation parameters that will be collected.

2.1. Deep Learning

Artificial intelligence (AI) has gained significant popularity in the past ten years, with numerous papers in technology and non-technology magazines discussing machine learning (ML), deep learning (DL), and AI. [43]-[48] Confusion persists over AI, ML, and DL. One can closely connect the terms but cannot use them interchangeably. In 1956, a group of computer scientists suggested that computers could imitate human thinking and reasoning by accurately characterizing all aspects of intelligence to create a mimicking system. [49] They labeled this principle “artificial intelligence.” Artificial intelligence aims to automate cognitive processes traditionally performed by humans, utilizing machine learning and deep learning as specialized techniques to accomplish this goal. They fall under the domain of AI, as depicted in Fig. 3.

AI encompasses methodologies that do not require any type of “learning.” Symbolic AI, specifically, involves the explicit creation of rules for all potential situations inside a specific domain, known as hard coding. Humans create these rules using their prior knowledge of the particular subject and task. If someone were to develop an algorithm to control an office’s room temperature, they would likely be aware of the comfortable temperature range for human work and would program the system to cool the room if temperatures exceeded a certain threshold and heat it if they fell below another threshold. Symbolic AI excels at solving well-defined logical issues but struggles with situations that demand advanced pattern recognition, like speech recognition or image categorization. ML and DL algorithms excel at handling more intricate jobs.

Deep learning makes it easier for computational models to collect and visualize data in a way that mimics how the human brain processes and interprets multimodal inputs [50]. The CNN method, discussed below, is a type of deep learning method.

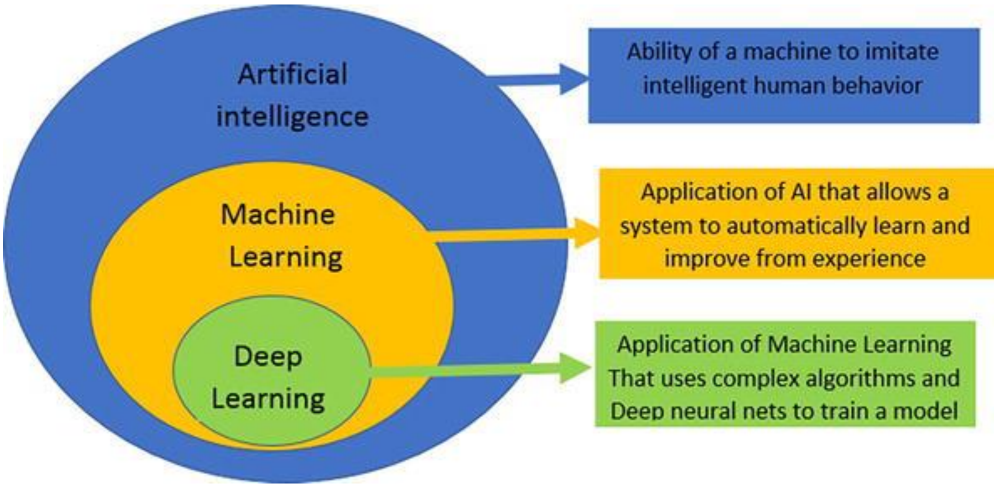

Fig. 3. Artificial intelligent, machine & deep learning

2.2. Convolution Neural Network (CNN)

CNN operates on a similar premise to the organic visual system [51]. Neurons in the brain of the biological visual system primarily focus on local data in the picture because they only receive stimulation in specific regions. The brain of vision creates the overall sense of the picture by processing the local information from each cell. CNN's distinctive architecture helps to reduce the computational cost of the neural network's overfitting and classification. The final output layer, an input layer, a fully connected layer (FCL), a pooling layer (PL), and a convolutional layer (CL) make up the CNN architecture, as shown in Fig. 4.

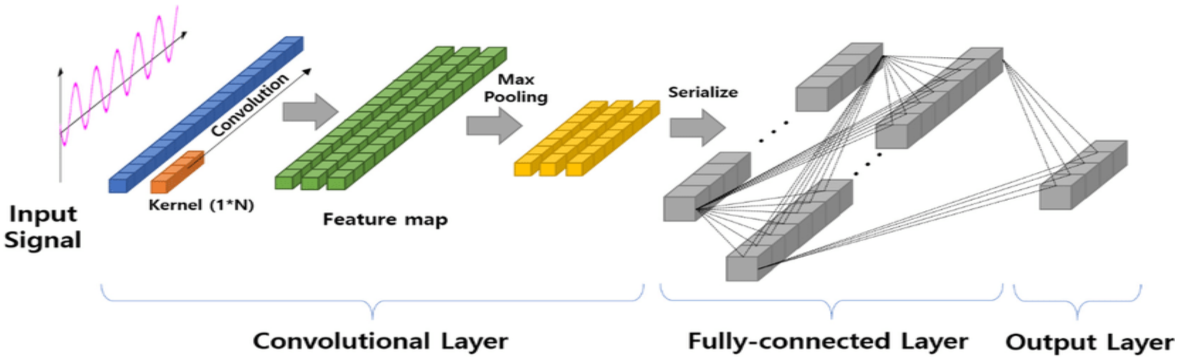

Fig. 4. CNN with four layer

The convolutional layer of the CNN retrieves high-level abstract features from the input data. An activation function with different weights and biases further enhances the abstract characteristics of the convolutional layer. CNN's convolutional process is represented in Equation (1).

$$C_l^j = f\left(\sum_{i \in Pn} C_i^{j-1} * w_{il}^j + b_l^j\right) \quad (1)$$

Where l: the number of layers; j represent component; w: represent weight; b: represent basis. Adding a pooling layer to the CNN makes the features it gets from the convolutional layer even better, since discriminant features are very important for a classifier to correctly classify something. In this study, we employ max-pooling to eliminate unnecessary data. We can use the following formula to mathematically express the maximum pooling procedure, which is represented in Equation (2).

$$C_l^j = f\left(w_l^j * \max\left(C_i^{j-1}\right) + b_l^j\right) \tag{2}$$

Where C_l^j represents the maximum pooling operation. This work employs the modified linear unit as an activation function to enhance the linear discriminant of the features. CNN employs numerous convolutional and pooling layers to improve its ability to take local data and extract it from the input. After extracting the latent information from the convolutional and pooling layers, CNN classifies the data into the appropriate classifications. For this, CNN makes use of a completely connected layer. A fully connected layer classifies the latent features derived from the convolution kernel [52] and is represented in Equation (3).

$$fC^{j+1} = \left(\sum_{i=1}^{n} w_{il}^j a^{j(i)} + b_l^j\right) \tag{3}$$

We used one dimension of CNN in this study to identify pipeline leaks. CNN is used for extracting features in the leak detection method. This is due to CNN's demonstrated proficiency in machine vision, object recognition, fault identification, and other related tasks. In contrast to 2DCNNs, which typically handle two-dimensional images, the 1DCNN model in this work is used to extract features from one-dimensional time series data.

2.3. 1D Convolutional Neural Networks (1D CNNs)

The conventional deep CNNs presented in the previous section are designed to operate exclusively on 2D data, such as images and videos. This is why they are often referred to as ‘‘2D CNNs.” As an alternative, a modified version of 2D CNNs called 1D Convolutional Neural Networks (1D CNNs) has recently been developed [53]-[62]. These studies have shown that for certain applications, 1D CNNs are advantageous and thus preferable to their 2D counterparts in dealing with 1D signals due to the following reasons. This preference is based on the following reasons:

1. There is a notable contrast in computational complexity between 1D and 2D convolutions. Specifically, convolving an image with N*N dimensions using a K*K kernel results in a computational complexity of approximately O~(N^2K^2) for 2D convolution and approximately O~(NK) for 1D convolution with the same dimensions. Under identical conditions, the computational complexity of a 1D CNN is notably lower than that of a 2D CNN.
2. Recent studies show that 1D CNN applications typically use compact configurations with 1-2 hidden CNN layers and networks with less than 10,000 parameters. In contrast, 2D CNN applications commonly use deep architectures with over 1 million parameters, often exceeding 10 million. Shallow networks are easier to train and implement compared to deep networks.
3. Training deep 2D CNNs typically necessitates specialized hardware configurations, such as cloud computing or GPU farms. CPU implementation on a regular computer is possible and quite quick for training small 1D CNNs with few hidden layers (e.g., 2 or less) and neurons (e.g., less than 50).
4. Compact 1D CNNs are ideal for real-time and cost-effective applications, particularly on mobile or hand-held devices, because of their minimal processing demands [53]-[64].

2.4. Methods for Detecting Leaks in Pipelines

From the points, the procedure for implementing 1DCNN will be used on the data set that was provided by the client and the simulation application: (a) a dataset was built with instances of all cases of the original dataset that was provided split between a training set and a validation set; (b) the training of the dataset was augmented by 1DCNN using Python and MATLAB for classification and regression; (c) then, after training, we calculated the mean squared error, root mean absolute error, mean absolute error, and relative error; I after that, we transformed this code into a user-friendly platform using App MATLAB Designer (a graphical user interface), enhancing accessibility and usability.

3. Mathematical Model

Previous research stated that the lack of pipeline datasets is one of the main problems facing pipeline leak detection research. Researchers are unable to finance the acquisition of data from actual pipelines, and operators are unwilling to provide the datasets. Researchers used fictitious data for the transmission pipeline model due to financial constraints and operators' unwillingness to provide datasets [65]. Pipeline modeling and simulation are based on the transient pipeline flow model. The continuity, momentum, state, and energy equations are the basic formulas that govern this model [66]. The concept of mass conservation is at the heart of the continuity equation. It requires that the mass change rate inside a pipeline segment correspond to the mass flow variation entering and exiting that section.

Governing equations. To model the pressure behavior induced by leaks in a pipeline system, the one-dimensional mechanistic homogeneous model proposed by Cazarez-Candia and Vásquez-Cruz (2005), was modified by adding a leak term on each conservation equation, similarly to other works (Emara-Shabaik et al., 2002, Kam, 2010). The suggested model assumes a pipe with a constant cross-sectional area, adiabatic liquid flow, a straight pipe, and approximates the leak hole as a circle (Kam, 2010). These assumptions are reflected in Equations (4) and (5).

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho \partial P}{\partial P \partial t} = \frac{1}{C^2}\frac{\partial P}{\partial t} \tag{4}$$

And

$$\frac{\partial \rho}{\partial x} = \frac{\partial \rho \partial P}{\partial P \partial t} = \frac{1}{C^2}\frac{\partial P}{\partial x} \tag{5}$$

Where (ρ) is the liquid density, (P) is the static pressure, (C) is the speed of sound in the liquid phase, (t) is the time coordinate, and (x) is the spatial coordinate. The mathematical model is formed by mass and momentum conservation equations (6) and (7).

$$\frac{1}{C^2}\frac{\partial P}{\partial t} + \frac{Q}{AC^2}\frac{\partial P}{\partial x} + \frac{\rho}{A}\frac{\partial Q}{\partial x} = -\Gamma_L \tag{6}$$

$$\frac{Q}{AC^2}\frac{\partial P}{\partial t} + \frac{\rho}{A}\frac{\partial Q}{\partial t} + \left(\frac{Q^2}{A\,C^2} + 1\right)\frac{\partial P}{\partial x} + (\Psi)\frac{2\rho Q}{A^2}\frac{\partial Q}{\partial x} = -\frac{\tau_W S}{A} - (\Psi)V_L\Gamma_L \tag{7}$$

Where (A) is the pipe cross-section area, $(\Gamma_L) = M_L/Adx$ is the leak term, (M_L) is the leak mass flow rate, (Q) is the one-dimensional flow rate, (S) is the wet perimeter, (V_L) is the velocity in the leak stream, and (τ_W) is the wall friction shear stress, which is defined as shown in Equation (8).

$$\tau_W = \frac{1}{2}F\rho V^2 \tag{8}$$

Where (F) is the Fanning friction factor. On the other hand, The Bernoulli equation calculates the pressure changes in a pipe caused by fluid flow. This equation can also predict potential locations of leakage in a pipe. Can be describe in Equation (9).

$$P_1 + \frac{1}{2}\rho v_1^2 + \rho g h_1 + W_1 = P_2 + \frac{1}{2}\rho v_2^2 + \rho g h_2 + W_2 \tag{9}$$

Where: (P) represents the pressure, (ρ) is the fluid density, (v) is the fluid velocity, (g) is the acceleration due to gravity, (h) is the height of the fluid above a reference point, and (W) represents the work done by external forces (e.g., pumps). In a pipeline system with a steady flow and a leak, this equation can be modified to account for the different factors affecting pressure, as shown in Equation (10).

$$P_1 + \frac{1}{2}\rho v_1^2 + \rho g h_1 + h_{f,1} + h_L = P_2 + \frac{1}{2}\rho v_2^2 + \rho g h_2 + h_{f,2} \tag{10}$$

Where: (h_f) represents the frictional head loss, accounting for energy dissipated due to pipe wall friction; (h_L) represents the local head loss induced by the leak, accounting for the energy lost due to the leak.

Also, insert a branch pipe of a certain diameter into the main pipeline to simulate leakage. The branch pipe can have a variable leakage rate, and the variable leakage rate allows checking for different types of leaks on the main pipeline. Fig. 5 provides a schematic representation of this type of pipe. This figure shows the distances D1 and D2 between the leak location and the upstream and downstream pressure sensors, respectively. The pressure sensors, S1 and S2, are I/P and O/P, respectively. The arrows' direction, which indicates the path from the pin to the outlet, represents fluid flow [67]. Fig. 6 shows the flow rate and pressure for leaking and non-leaking pipelines. And Equation (11) shows that.

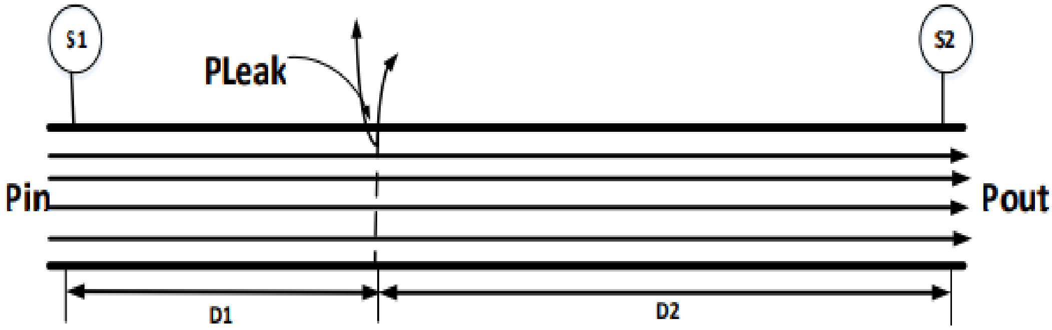

Fig. 5. Leak at distance (X) in the pipeline

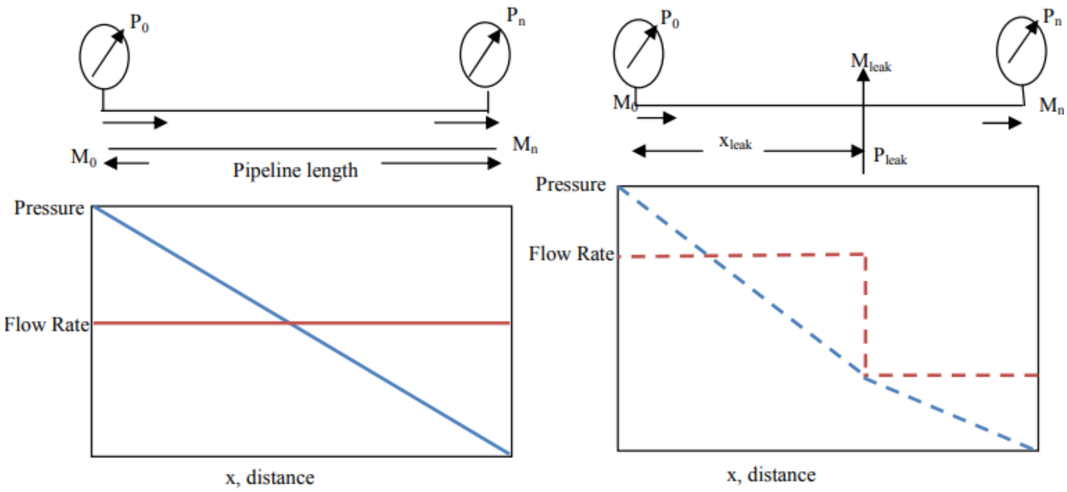

Fig. 6. Pipeline leak and no-leak

$$Pin = Pout + Pleak + Ploss \tag{11}$$

Where $Ploss$: Wax accumulation in the pipeline causes pressure loss (0 for new pipes). $Pleak$: Pressure loss because of leak. D_1: Leak point to pressure sensor distance upstream. D_2: Leak point to pressure sensor distance downstream

4. Method Pipeline Leakage Detection and Localization

This paper used the 1DCNN model, which consists of two convolutional layers and three fully linked layers, excluding pooling layers. This model utilizes variations in leak pressure and flow rate changes to identify various leak apertures. We trained CNN using data from both leak and no-leak scenarios to create and evaluate the pipeline leak identification model. Fig. 7 displays the suggested methodology flowchart for 1DCNN leak classification.

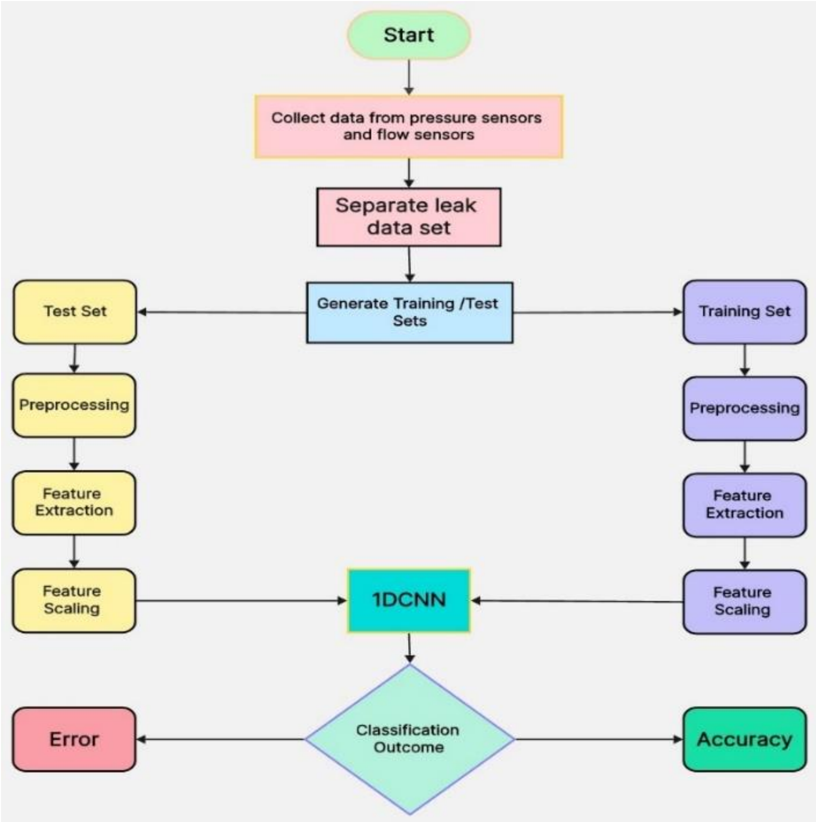

Fig. 7. Flow chart of classification

4.1. CNN Architecture Made

We designed the modal CNN to be suitable for learning the relationship between parameter input and pipeline leak detection and localization.

1. Input layer: In this layer, define the size of the data input. We have four columns and five rows, which represent one labeled set.
2. Convolution layer: We use 16 filters of size 3x3 for the kernel to extract significant features. Values (weights) in the kernel of the convolutional layer convolve with the input data to generate feature maps. Empirical research has demonstrated that networks utilizing 3x3 filters typically perform various tasks. The widespread use of 3x3 filters in successful structures confirms their efficacy. Choosing a 3x3 filter size in convolutional layers provides a balance between obtaining local features, parameter efficiency, and getting optimal performance in various tasks. Padding the output dimensions to match the input dimensions.
3. Relu layer: applies the activation function to the output of the convolution layer to make it non-linear.
4. The fully connected layer contains weights, biases, and neurons to establish connections between neurons in different layers. These layers are usually placed before the output layer and form the final layers of a CNN architecture. • This layer has a single neuron that generates a value indicating the predicted leak.
5. Classification or Regression Layer: Calculates the error between the predicted and actual outputs. • Primarily used for regression assignments, such as predicting continuous values like pipeline leak location and the amount of leakage.

After the CNN architecture has been planned, we do the other requirements, as choosing the right location to conduct the study and collect real-time data is critical to the success of the research. Here are some points to consider when choosing the right location:

1. Availability of historical data: Search for sites where historical data relevant to your study is available. This data can include historical records about environmental conditions, oil flow rates, pressure passing through pipes, etc. Access to this data provides a basis for analysis and comparison with results based on neural networks.
2. Existing sensor infrastructure: sites that already have sensor networks (pressure sensors and flow sensors) should be identified. These sensors collect data related to variables of interest in the research, allowing you to collect the inputs needed to train and test your neural network models.
3. Presence of main pipes or infrastructure: A location must be chosen that contains main pipes or important infrastructure related to the research application.

4.2. Data Collection

During the data collection phase, we faced the challenge that the client's data was insufficient, as most of it was for cases of no leakage in the conveyor pipeline. To address this limitation, we choose to use simulation software for oil pipelines, specifically PIPESIM 2022.2, a versatile multi-phase flow simulator designed for analyzing wells, flow lines, and pipeline systems, as well as surface equipment such as chokes, separators, pumps, and compressors. The fluid characteristics may be modeled using black-oil correlations or completely compositional methods. We have made sure to retain all parameters related to pipe and crude oil specifications that were also provided to us by the company's laboratories. Table 1. shows all the parameters used in the simulation.

Three points along the 10-km oil pipeline were selected to conduct simulations of oil spills at each of 2.5, 5, and 7.5 km. These simulations were conducted with consideration of all parameters related to the crude oil and piping characteristics specified in the chosen oil field. The simulation aimed to model and analyze various potential leakage scenarios, taking into account factors such as flow rates, pressures, fluid properties, piping conditions, and environmental variables. By simulating oil spills at these specific points along the pipe, we can evaluate response mechanisms, evaluate the effectiveness of leak detection systems, and improve emergency procedures to deal with such incidents.

Table 1. The parameters using in simulation

Variables			
Parameters of pipeline	**Value**	**Parameters of crude oil**	**Value**
Inside diameter	15.25 in	Watercut	0.25 %
Wall thickness	0.375 in	Gas specific gravity	0.6636 SCF/STB
Roughness	0.0018	Water specific gravity	1.02
Horizontal distance	8.202.1 ft	API	28 dAPI
U Value type	Insulated	Temperature (1^{st},2^{nd})	(200,60) degF
Heat transfer coefficient	0.2 Btu/(h.degF.ft2)	Viscosity(1^{st},2^{nd})	(3.1342,273.7911) cP
Inside film coefficient	Include	Oil specific heat capacity	0.45 Btu/(Ibm.degF)

The flow gauge is placed between the valve and the pipe to calculate the rate of leakage. The PIPESIM program (version 2022.2, 64-bit) gathers pressure and flow rate data from the pipeline by adjusting parameters such as crude oil pressure, leak size, leak location, and the oil leakage flow rate. It produces precise data without any interference. Modifying the leak flow can classify the gathered data as either a significant or minor leak. Nevertheless, in real-world scenarios, there will always be some interference in the gathered data, including measurement noise and ambient noise.

Furthermore, we selected different leak locations and compared them with real-life scenarios to strengthen the validity of our findings. We placed the data coming from the client, as well as those

Vol. 4, No. 1, 2024, pp. 312-333

obtained from the simulation program, in a csv file after rearranging them as shown below. Table 2, where the first column represents the distance, the second column represents the pressure, the third column represents the oil flow at the source, and the last column represents the oil flow at the point of leakage, with every five rows representing one leak or non-leak condition. The data are arranged in this way to facilitate analysis and understanding of the results, as this organization allows each leak or non-leak condition to be better visualized and evaluated effectively.

Table 2. The sample of dataset

Total Distance (Km)	Pressure (barge)	Flowrate (STB/h)	Leak Flow Rate (STB/h)
0	3.499989	253.0	0
2.5	3.490852	0	63
5	3.482406	0	0
7.5	3.465624	0	0
10	3.44756	0	0

5. Experiments

5.1. Classification

The target data set is split into 70% for the training set and 30% for the validation and test sets. We used Jupyter Notebook in Anaconda Navigator using Python 3.7, together with TensorFlow 2.1 and Keras packages. The system operated on Windows 10 with an Intel i7 CPU. Equations (12), (13), (14), and (15) are used to evaluate the classification performance of various approaches and other parameters.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F\_score = \frac{2TP}{2TP + FP + FN} \tag{15}$$

Where TP, TN, FP, and FN are the abbreviations of true positive, true negative, false positive, and false negative, respectively. After running the program for a number of epochs (32), we obtained an accuracy of training of about 92.4%, as shown below in Fig. 8.

```
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9129 - loss: 0.1902 - val_accuracy: 0.9324 - val_loss: 0.1475
Epoch 99/100
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9215 - loss: 0.1544 - val_accuracy: 0.9324 - val_loss: 0.1393
Epoch 100/100
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9287 - loss: 0.1531 - val_accuracy: 0.9441 - val_loss: 0.1302
```

Fig. 8. Accuracy of training

And by calculating the parameters using the information from the confusion matrix, we can evaluate the classification level for detecting the presence of a leak or not.

5.2. Regression

Moving on to the next level, we improved our approach to not only model the presence or absence of leakage, but also estimate the location and volume of leaked fluid using MATLAB 2022b, incorporating regression methods, especially 1DCNN (Convolutional Neural Network) regression. This advanced technology allows us to predict continuous numerical values, such as the location and size of fluid leaks in a pipeline. Fig. 9. Describe the detection and location processes.

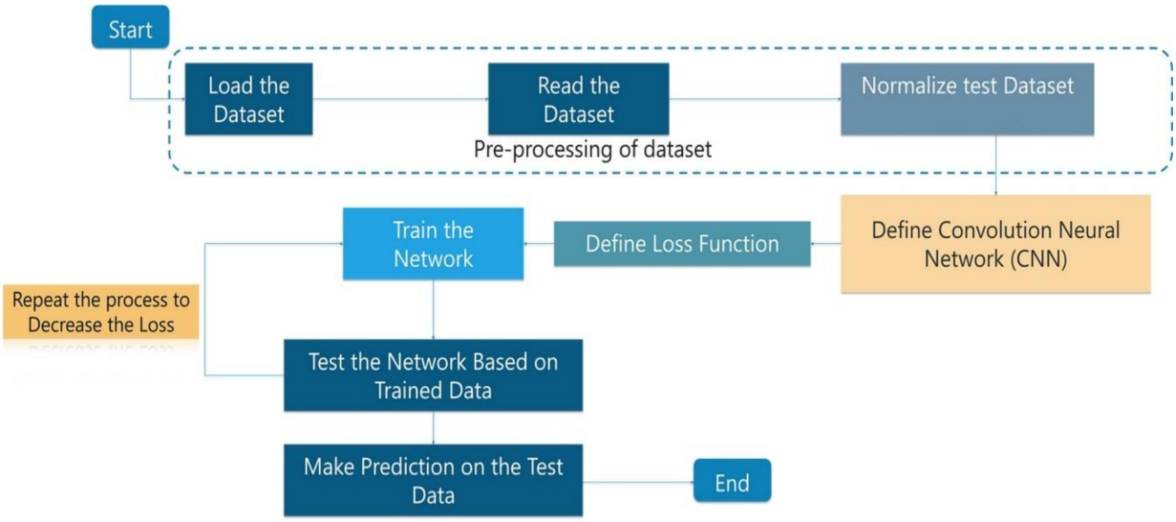

Fig. 9. Process of detection and location

In addition to using CNN regression, we calculated various metrics to evaluate the accuracy of our predictions. These metrics include mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), root mean absolute error (RMAE), and relative error (RAE), as shown in the equation below.

1. Mean square error (MSE).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^{\wedge})^2 \tag{16}$$

2. Root mean square error (RMSE).

$$RMSE = \sqrt{MSE} \tag{17}$$

3. Mean Absolute Error (MAE).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i^{\wedge}| \tag{18}$$

4. Root Mean Absolute Error (RMAE).

$$RMAE = \sqrt{MAE} \tag{19}$$

5. Relative Error (RAE).

$$RAE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - y_i^{\wedge}}{y_i}\right| \times 100\% \tag{20}$$

Where: n is number of samples; $y_i$is the actual target value; $y_i^{\wedge}$ is the predicated value. By calculating these metrics, we can comprehensively evaluate the performance of our CNN regression model in predicting leak's location and estimating fluid quantities. This allows us to evaluate accuracy and improve the Regression Training process.

5.3. MATLAB App Designer

Converting the code into a user-friendly platform using MATLAB App Designer after completing the regression training and calculating the system evaluation parameters is a great idea to enhance accessibility and ease of use. App Designer allows the creation of interactive MATLAB applications using a graphical user interface (GUI) that users can easily navigate. The design was completed as shown in Fig. 10.

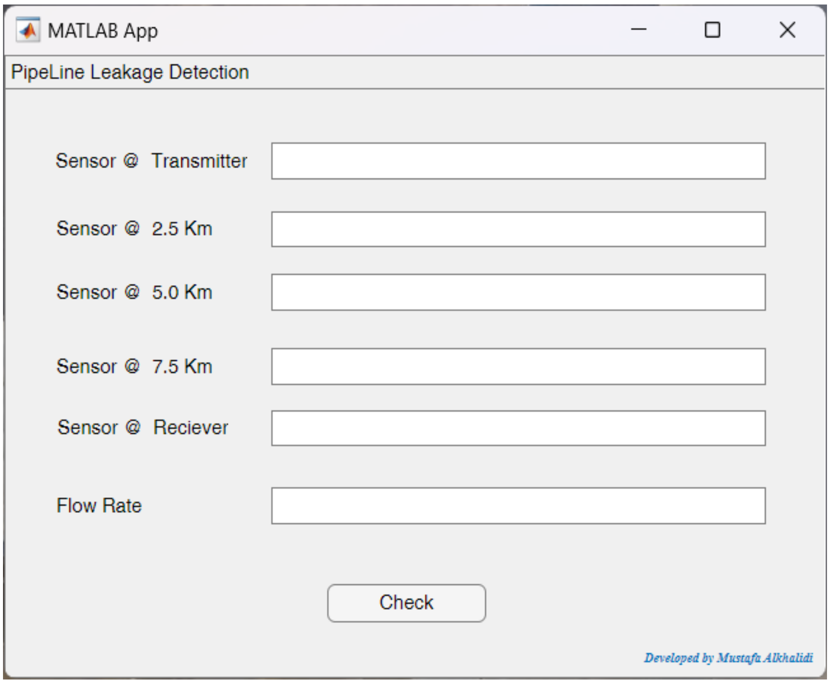

Fig. 10. Pipeline leakage detection and localization (GUI)

Where the first five fields represent the liquid pressure points at the distances marked next to each of them, while the sixth field represents the flow rate of the liquid at the source. When the check button is pressed, a message is displayed explaining the absence or presence of a leak or no-leak.

6. Result

After the classification step, we obtained the results as the blow graphs and confusion matrix. We notice the high performance of Calsifiction. The graphs in Fig. 11. show the training and validation losses, and Fig. 12. shows the training and validation accuracy. We notice the two curves represent the training and validation are close to each other, which means the value for training and validation is near; on the other hand, the accuracy is high.

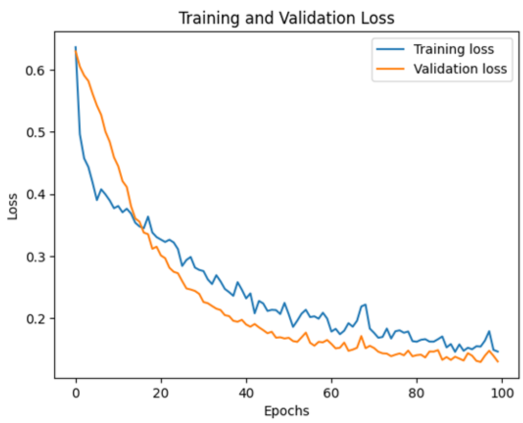

Fig. 11. Training and validation loss

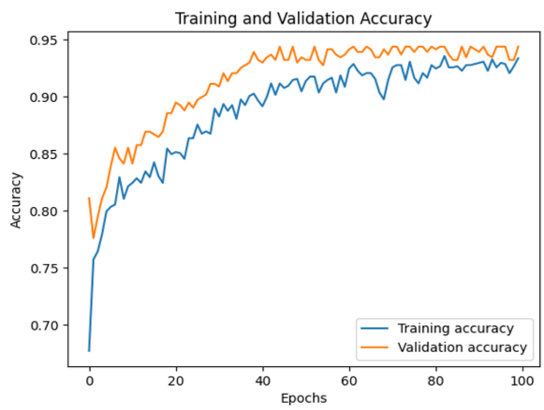

Fig. 12. Training and validation accuracy

Fig. 13 shows the confusion matrix of the true and false predictions of the system. There are four regions. the first one with number (214), which represents true and predicts equally the same status (no leak); the second one with number (191), which represents true and predicts equally the same status (leak); the other regions represent not accuracy in calcification (a small number). The confusion matrix used to determine metrics such as classification accuracy, precision, recall, and F-score.

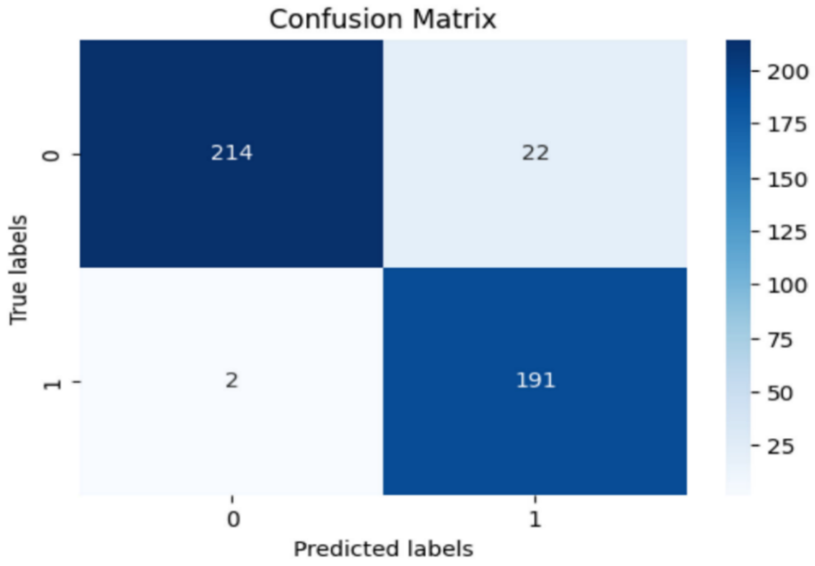

Fig. 13. Confusion matrix

1. Classification Accuracy is defined as the proportion of accurate predictions to the entirety of predictions made.
2. Precision is calculated as the number of real positive predictions divided by the entire number of positive forecasts.
3. Recall It is the ratio of true positive predictions to the total number of actual positives.
4. The F1-Score is the harmonic mean of precision and recall. It measures test accuracy while balancing precision and recall. All these parameters are calculated as shown in Table. 3.

Table 3. Evolution parameters of classification

Parameters	Value
Accuracy	0.944
Precision	0.896
Recall	0.989
F1-Score	0.94

After the classification step is finished with high accuracy. The next stage is approaching. The training process in the regression step to estimate the location and volume of leaked fluid is shown in Fig. 14, and Fig. 15 shows the predicted visualizations (plot the predicted values vs. the true values). After the completion of the regression training process, the overall performance of the system was evaluated, and the parameters for the evaluation were calculated as shown in Table. 4.

During this process, we modified several training parameters, such as the number of layers, learning rate, epoch, and batch size, to achieve the best possible outcomes. Data preprocessing is necessary before constructing the model. Standardizing the data before it is input into the neural network accelerates weight parameter convergence.

After completing the training processes for both classification and regression, we created a GUI model to facilitate users in examining the sensors data. The following figure shows the data examination processes for two different cases.

The Fig. 16 represent the input data from the (pressures and flow) sensors, and when you press the check button, it predicts the location of the leak as well as the amount of liquid coming from the leak. While the Fig. 17 shows entering the data coming from the sensors and in this case show no leak in pipeline for this input data, both cases gave relatively acceptable results that are close to the real situation.

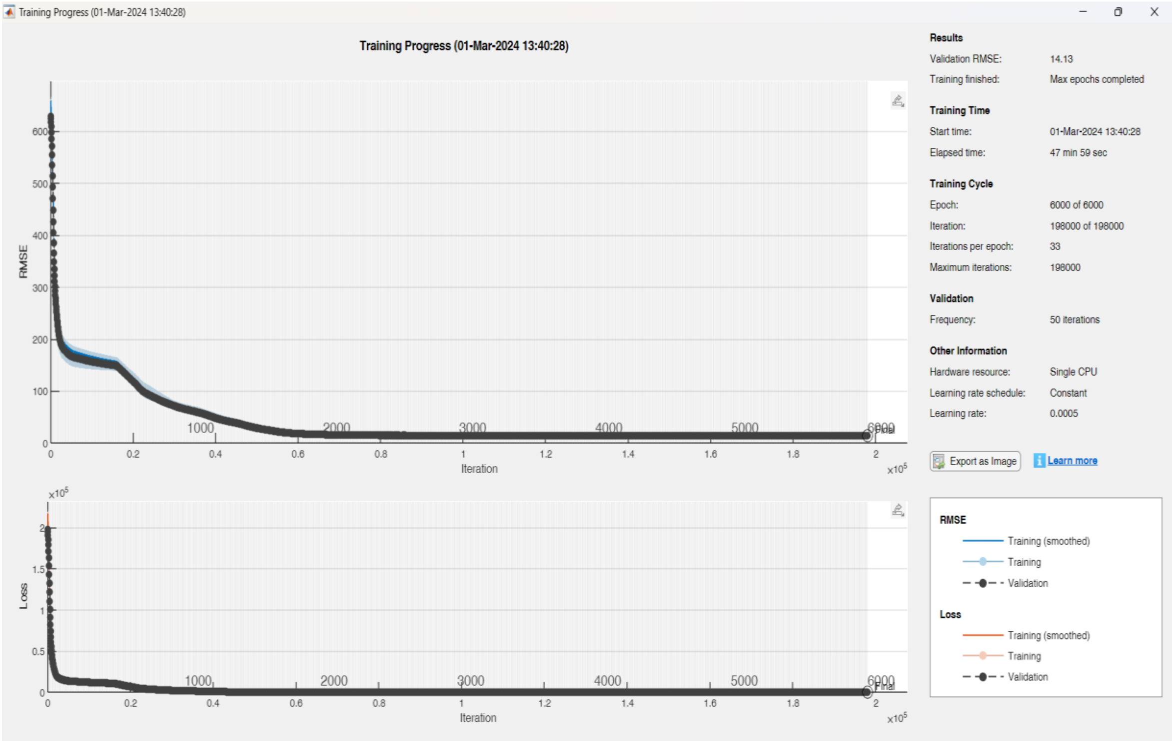

Fig. 14. Regression training process

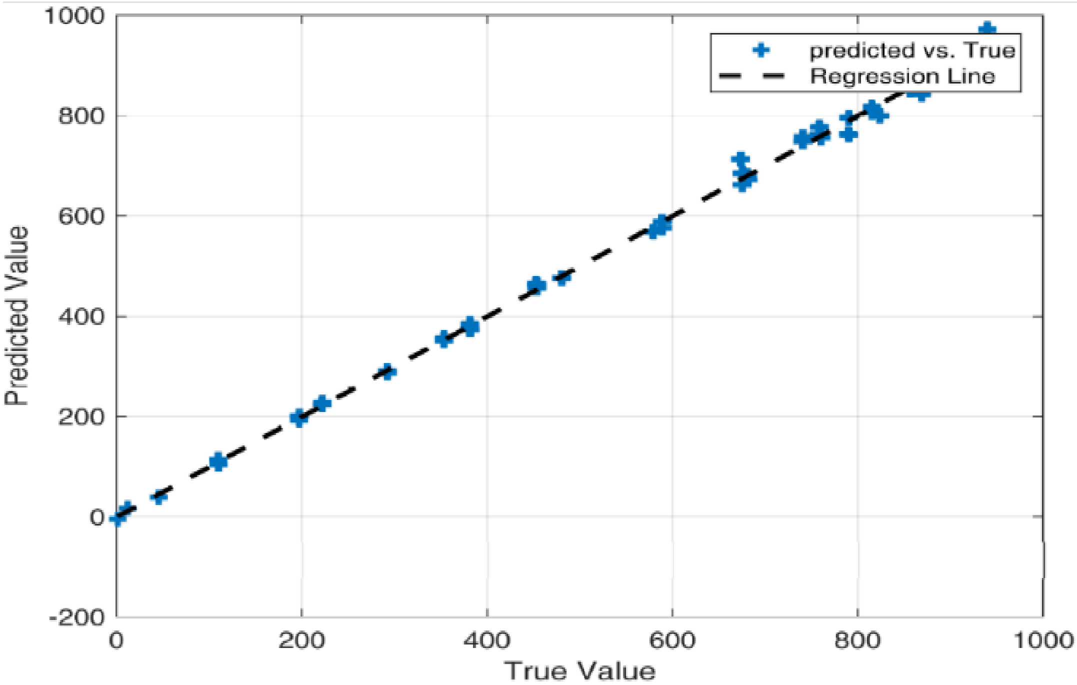

Fig. 15. Visualize the predictions values against the true values

After completing the training processes for both classification and regression, we created a GUI model to facilitate users in examining the sensors data. The following figure shows the data examination processes for two different cases.

Table 4. Evolution parameters of regression

Parameters	Value
Mean Square Error (MSE)	200
Root Mean Square Error (RMSE)	14.13
Mean Absolute Error (MAE)	10.294
Root Mean Absolute Error (RMAE)	3.208
Relative Error (RAE)	0.000388 %

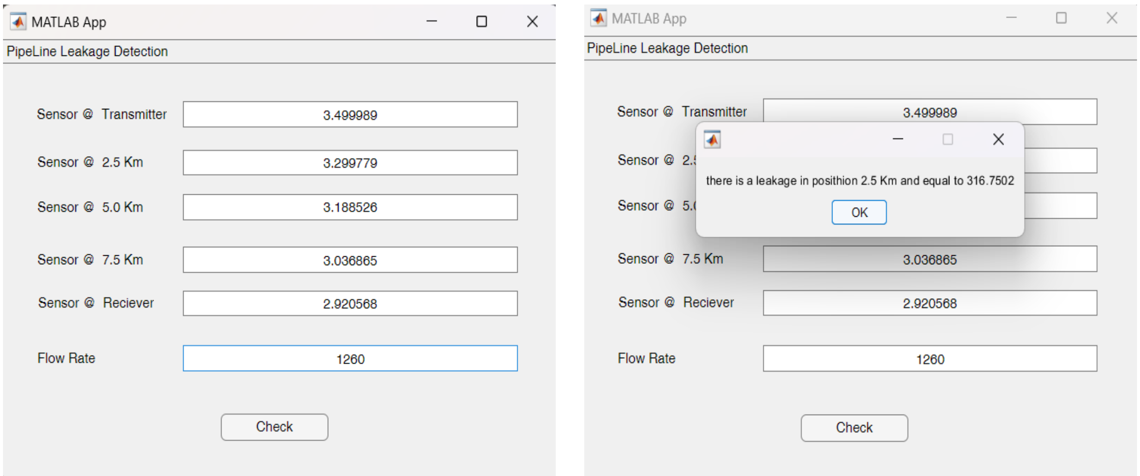

Fig. 16. MATLAB APP (oil pipeline leak state check)

The Fig. 16 represent the input data from the (pressures and flow) sensors, and when you press the check button, it predicts the location of the leak as well as the amount of liquid coming from the leak. While the Fig. 17 shows entering the data coming from the sensors and in this case show no leak in pipeline for this input data, both cases gave relatively acceptable results that are close to the real situation.

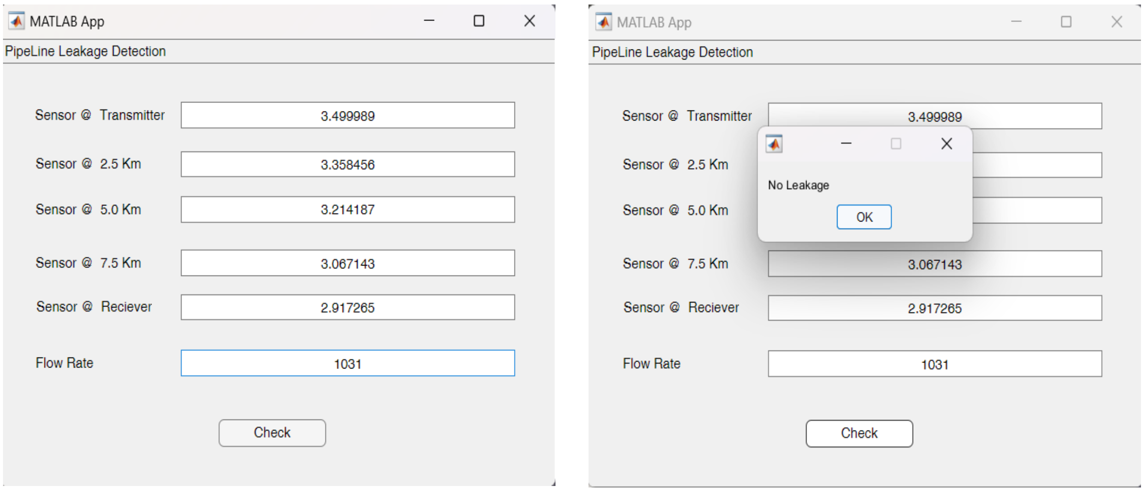

Fig. 17. MATLAB APP (oil pipeline no-leak state check)

7. Conclusion

This work aims to create a one-dimensional convolutional neural network model that can determine the detection, localization, and amount of oil spills using pressure and flow rate sensors for a 10-kilometer pipeline as input data. We subsequently used a numerical data set obtained for distinct sources of leakage at varied distances to train and evaluate the model.

7.1. Step of Research

Three levels divided the work:

a. Classification: After we build our 1DCNN by using Jupyter Notebook Anaconda in Python 3.7 with TensorFlow 2.1 and Keras libraries, the target dataset is divided into 70% for the training set and 30% for the validation and test sets. We acquired the following results for the evaluation parameters: (Accuracy: 0.944, Precision: 0.896, Recall: 0.989, F1 Score: 0.94).

b. Regression: at this level, the approach was improved to not only model the presence or absence of leakage but also estimate the location and volume of leaked fluid using MATLAB 2022b, resulting in the following results: Mean square error (MSE) = 200, root mean square error (RMSE) = 14.13, mean absolute error (MAE) = 10.294, root mean absolute error (RMAE) = 3.208, and relative error (RAE) = 0.000388%.

c. Graphical user interface application: we design a user-friendly platform using MATLAB App Designer to make an easy and accusable end user interface.

7.2. Limitations of Our Work

1. This work requires a large number of sensors and a history of data related to the study topic because the process of collecting data from sensors takes a lot of time.
2. Applying the dataset to another pipeline will render it useless because it cannot be generalized to all pipelines in an oil field.
3. If the data set is insufficient or the number of sensors is insufficient, this will limit the accuracy of spill results (classification, localization, and estimation of oil spills).
4. Any failure in the sensors located along the oil pipeline will lead to error detection, and the location results in this case will be unreliable.
5. External influences on the sensors may lead to incorrect data being included, and thus this may affect giving false alerts.
6. This method cannot identify more than one leak location on the conveyor line at the same time.

7.3. Comparison with Previous Research

This research utilizes an internal calculation-based (CNN) leak detection approach. Monitoring production data from several sensors located throughout the oil pipeline, including inlet and outlet pressure and flow rate, helps achieve this. Operators evaluate any departure from normal behavior and subsequently identify it as a leak.

This approach differs from exterior leak detection methods such as visual human inspection and external systems like acoustic and fiber optic.

1. Visual Manual Inspection: Visual inspection involves manually monitoring the pipeline to detect leaks. Patrolling may be conducted using several methods, such as walking, in a vehicle, or from a helicopter. The operator inspects the area for discoloration or other signs of leakage. The inspection team's skill, inspection frequency, and leak size influence leak detection effectiveness. Restricted to accessible pipelines; lacks real-time monitoring, leading to increased oil and gas loss and environmental damage.
2. External systems rely on local sensors to detect fluids leaking from pipelines. Impedance techniques use wires equipped with fiber optic or electrochemical sensors to detect liquids. Sniffing techniques rely on detecting vapors through tubes. Acoustic techniques rely on detecting leakage sounds. The systems are very sensitive to leakage and can precisely pinpoint them. These systems are only used in sensitive areas or for short sections of pipelines due to their high prices and inability to accurately compute the volume of fluid leaked.

From installing specific devices along the pipelines for hardware-based leak detection makes them more expensive and unable to determine the leakage rate. On the other hand, there are software-based techniques that deal with Software programs, at their core, implement algorithms continuously to monitor the state of pressure and flow rate or other pipeline parameters and can infer, based on the evolution of these quantities, if a leak has occurred. and this method does not have the complexities of most methods in this area, and its accuracy in leak localization is very good.

7.4. Most Important Challenges Faced in this Research

1. Sparse Data Challenges: Pipeline data, particularly in regards to leak detection, often has gaps and inconsistencies. However, convolutional neural networks (CNNs) require ample data for optimal performance. Sparse data can cause models to overfit or generalize poorly.
2. Class Imbalance Limitations: In pipeline data, the distribution of leak and non-leak events is frequently unbalanced. CNNs may struggle with imbalanced datasets, resulting in biased predictions or reduced accuracy for detecting leaks (the minority class).
3. Which data should be used: Because at first, when we worked on this research, we didn't know how much data would represent the normal behavior of the pipelines (for example, one month or more).

7.5. Suggestions for More Study in this Field are as Follows

Implement this research by using sensors to read the thickness of the pipeline at certain points, evaluate the amount of corrosion that occurs in the walls of the pipes, and calculate corrosion rates, estimating the remaining life of the pipe based on the API 670 (pipeline inspection) standard. This research may overcome the shortcomings of the current fracture detection systems, which mostly rely on human input and cut costs and labor while conducting crack inspections.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

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

References

[1] K. Rehman and F. Nawaz, "Remote pipeline monitoring using Wireless Sensor Networks," *2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, pp. 32-37, 2017, https://doi.org/10.1109/C-CODE.2017.7918897.

[2] L. Boaz, S. Kaijage and R. Sinde, "An overview of pipeline leak detection and location systems," *Proceedings of the 2nd Pan African International Conference on Science, Computing and Telecommunications (PACT 2014)*, pp. 133-137, 2014, https://doi.org/10.1109/SCAT.2014.7055147.

[3] Q. Xiao, J. Li, J. Sun, H. Feng, and S. Jin, "Natural-gas pipeline leak location using variational mode decomposition analysis and cross-time–frequency spectrum," *Measurement*, vol. 124, pp. 163-172, 2018, https://doi.org/10.1016/j.measurement.2018.04.030.

[4] R. Cramer, D. Shaw, R. Tulalian, P. Angelo, and M. van Stuijvenberg, "Detecting and correcting pipeline leaks before they become a big problem," *Marine Technology Society Journal*, vol. 49, no. 1, pp. 31-46, 2015, https://doi.org/10.4031/MTSJ.49.1.1.

[5] Z. Jia, Z. Wang, W. Sun, and Z. Li, "Pipeline leakage localization based on distributed FBG hoop strain measurements and support vector machine," *Optik*, vol. 176, pp. 1-13, 2019, https://doi.org/10.1016/j.ijleo.2018.09.048.

[6] A. Lukman, A. EMMANUEL, N. Chinonso, A. Mutiu, A. James, and K. Jonathan, "An anti-theft oil pipeline vandalism detection: embedded system development," *International journal of engineering science and application*, vol. 2, no. 2, pp. 55-64, 2018, https://dergipark.org.tr/en/pub/ijesa/issue/38052/408770#article_cite.

[7] M. Wasim and M. B. Djukic, "External corrosion of oil and gas pipelines: A review of failure mechanisms and predictive preventions," *Journal of Natural Gas Science and Engineering*, vol. 100, p. 104467, 2022, https://doi.org/10.1016/j.jngse.2022.104467.

[8] B. M. S. Arifin, Z. Li, S. L. Shah, G. A. Meyer, and A. Colin, "A novel data-driven leak detection and localization algorithm using the Kantorovich distance," *Computers & Chemical Engineering*, vol. 108, pp. 300-313, 2018, https://doi.org/10.1016/j.compchemeng.2017.09.022.

[9] D. T. Thekkuden, A.-H. I. Mourad, and A.-H. Bouzid, "Failures and leak inspection techniques of tube-to-tubesheet joints: A review," *Engineering Failure Analysis*, vol. 130, p. 105798, 2021, https://doi.org/10.1016/j.engfailanal.2021.105798.

[10] N. Li, J. Wang, R. Liu, and M. Höök, "Methane emission reduction in China's natural gas industry: Construction of technology inventory and selection of optimal technology programs," *Sustainable Production and Consumption*, vol. 44, pp. 39-54, 2024, https://doi.org/10.1016/j.spc.2023.12.002.

[11] K. Siler-Evans, A. Hanson, C. Sunday, N. Leonard, and M. Tumminello, "Analysis of pipeline accidents in the United States from 1968 to 2009," *International journal of critical infrastructure protection*, vol. 7, no. 4, pp. 257-269, 2014, https://doi.org/10.1016/j.ijcip.2014.09.002.

[12] I. Bolotina, V. Borikov, V. Ivanova, K. Mertins, and S. Uchaikin, "Application of phased antenna arrays for pipeline leak detection," *Journal of Petroleum Science and Engineering*, vol. 161, pp. 497-505, 2018, https://doi.org/10.1016/j.petrol.2017.10.059.

[13] L. Meng, L. Yuxing, W. Wuchang, and F. Juntao, "Experimental study on leak detection and location for gas pipeline based on acoustic method," *Journal of Loss Prevention in the Process Industries*, vol. 25, no. 1, pp. 90-102, 2012, https://doi.org/10.1016/j.jlp.2011.07.001.

[14] H. Jin, L. Zhang, W. Liang, and Q. Ding, "Integrated leakage detection and localization model for gas pipelines based on the acoustic wave method," *Journal of Loss Prevention in the Process Industries*, vol. 27, pp. 74-88, 2014, https://doi.org/10.1016/j.jlp.2013.11.006.

[15] Y. Mahmutoglu and K. Turk, "A passive acoustic based system to locate leak hole in underwater natural gas pipelines," *Digital Signal Processing*, vol. 76, pp. 59-65, 2018, https://doi.org/10.1016/j.dsp.2018.02.007.

[16] K. Lim, L. Wong, W. K. Chiu, and J. Kodikara, "Distributed fiber optic sensors for monitoring pressure and stiffness changes in out-of-round pipes," *Structural Control and Health Monitoring*, vol. 23, no. 2, pp. 303-314, 2016, https://doi.org/10.1002/stc.1771.

[17] Z. Jia, L. Ren, H. Li, and W. Sun, "Pipeline leak localization based on FBG hoop strain sensors combined with BP neural network," *Applied Sciences*, vol. 8, no. 2, p. 146, 2018, https://doi.org/10.3390/app8020146.

[18] W. H. Png, H. S. Lin, C. H. Pua, and F. Abd Rahman, "Pipeline monitoring and leak detection using Loop integrated Mach Zehnder Interferometer optical fiber sensor," *Optical Fiber Technology*, vol. 46, pp. 221-225, 2018, https://doi.org/10.1016/j.yofte.2018.10.013.

[19] S.-H. Ni, Y.-H. Huang, K.-F. Lo, and D.-C. Lin, "Buried pipe detection by ground penetrating radar using the discrete wavelet transform," *Computers and Geotechnics*, vol. 37, no. 4, pp. 440-448, 2010, https://doi.org/10.1016/j.compgeo.2010.01.003.

[20] Q. Hoarau, G. Ginolhac, A. M. Atto, and J.-M. Nicolas, "Robust adaptive detection of buried pipes using GPR," *Signal Processing*, vol. 132, pp. 293-305, 2017, https://doi.org/10.1016/j.sigpro.2016.07.001.

[21] H. Li, D. -Y. Xiao and X. Zhao, "Morphological filtering assisted field-pipeline small leakage detection," *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3769-3774, 2009, https://doi.org/10.1109/ICSMC.2009.5346676.

[22] M. R. Delgado and O. B. Mendoza, "A comparison between leak location methods based on the negative pressure wave," *2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1-6, 2017, https://doi.org/10.1109/ICEEE.2017.8108841.

[23] Q. Chen *et al.*, "Effect of rubber washers on leak location for assembled pressurized liquid pipeline based on negative pressure wave method," *Process Safety and Environmental Protection*, vol. 119, pp. 181-190, 2018, https://doi.org/10.1109/ICEEE.2017.8108841.

[24] S. I. Kam, “Mechanistic modeling of pipeline leak detection at fixed inlet rate,” *Journal of Petroleum Science and Engineering*, vol. 70, no. 3-4, pp. 145-156, 2010, https://doi.org/10.1016/j.petrol.2009.09.008.

[25] S. Tian, J. Du, S. Shao, H. Xu, and C. Tian, “A study on a real-time leak detection method for pressurized liquid refrigerant pipeline based on pressure and flow rate,” *Applied Thermal Engineering*, vol. 95, pp. 462-470, 2016, https://doi.org/10.1016/j.applthermaleng.2015.11.067.

[26] G. He *et al.*, “A method for simulating the entire leaking process and calculating the liquid leakage volume of a damaged pressurized pipeline,” *Journal of hazardous materials*, vol. 332, pp. 19-32, 2017, https://doi.org/10.1016/j.jhazmat.2017.02.039.

[27] Z. Yang, S. Fan and T. Xiong, “Simulation and Numerical Calculation on Pipeline Leakage Process,” *2010 2nd International Symposium on Information Engineering and Electronic Commerce*, pp. 1-5, 2010, https://doi.org/10.1109/IEEC.2010.5533189.

[28] X. Li, G. Chen, R. Zhang, H. Zhu, J. Fu, “Simulation and assessment of underwater gas release and dispersion from subsea gas pipelines leak,” *Process Safety and Environmental Protection*, vol. 119, pp. 46-57, 2018, https://doi.org/10.1016/j.psep.2018.07.015.

[29] J. Wan, Y. Yu, Y. Wu, R. Feng, and N. Yu, “Hierarchical leak detection and localization method in natural gas pipeline monitoring sensor networks,” *Sensors*, vol. 12, no. 1, pp. 189-214, 2011, https://doi.org/10.3390/s120100189.

[30] M. H. Manekiya and P. Arulmozhivarman, “Leakage detection and estimation using IR thermography,” *2016 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1516-1519, 2016, https://doi.org/10.1109/ICCSP.2016.7754411.

[31] J. Sun, Z. Peng, and J. Wen, “Leakage aperture recognition based on ensemble local mean decomposition and sparse representation for classification of natural gas pipeline,” *Measurement*, vol. 108, pp. 91-100, 2017, https://doi.org/10.1016/j.measurement.2017.05.029.

[32] Y. Gao, Y. Liu, Y. Ma, X. Cheng, and J. Yang, “Application of the differentiation process into the correlation-based leak detection in urban pipeline networks,” *Mechanical Systems and Signal Processing*, vol. 112, pp. 251-264, 2018, https://doi.org/10.1016/j.ymssp.2018.04.036.

[33] S. Yin *et al.*, “Mass transfer characteristics of pipeline leak-before-break in a nuclear power station,” *Applied Thermal Engineering*, vol. 142, pp. 194-202, 2018, https://doi.org/10.1016/j.applthermaleng.2018.06.077.

[34] N. Turner, “Hardware and software techniques for pipeline integrity and leak detection monitoring,” *SPE Offshore Europe Conference and Exhibition*, pp. SPE-23044, 1991, https://doi.org/10.2118/23044-MS.

[35] P.-S. Murvay and I. Silea, “A survey on gas leak detection and localization techniques,” *Journal of Loss Prevention in the Process Industries*, vol. 25, no. 6, pp. 966-973, 2012, https://doi.org/10.1016/j.jlp.2012.05.010.

[36] M. Ayaz and H. Yüksel, “Design of a new cost-efficient automation system for gas leak detection in industrial buildings,” *Energy and Buildings*, vol. 200, pp. 1-10, 2019, https://doi.org/10.1016/j.enbuild.2019.07.038.

[37] D. Matko, G. Geiger, and T. Werner, “Neural net versus classical models for the detection and localization of leaks in pipelines,” *Mathematical and Computer Modelling of Dynamical Systems*, vol. 12, no. 6, pp. 505-517, 2006, https://doi.org/10.1080/13873950500068526.

[38] D. Chatzigeorgiou, K. Youcef-Toumi, and R. Ben-Mansour, “Design of a novel in-pipe reliable leak detector,” *IEEE/ASmE Transactions on mechatronics*, vol. 20, no. 2, pp. 824-833, 2015, https://doi.org/10.1109/TMECH.2014.2308145.

[39] M. A. Adegboye, W.-K. Fung, and A. Karnik, “Recent advances in pipeline monitoring and oil leakage detection technologies: Principles and approaches,” *Sensors*, vol. 19, no. 11, p. 2548, 2019, https://doi.org/10.3390/s19112548.

[40] S. Sangeethaa and P. Uma Maheswari, “An intelligent model for blood vessel segmentation in diagnosing DR using CNN,” *Journal of medical systems*, vol. 42, no. 10, p. 175, 2018, https://doi.org/10.1007/s10916-018-1030-6.

[41] L. Song *et al.*, “A deep multi-modal CNN for multi-instance multi-label image classification,” *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 6025-6038, 2018, https://doi.org/10.1109/TIP.2018.2864920.

[42] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, “Action recognition in video sequences using deep bi-directional LSTM with CNN features,” *IEEE access*, vol. 6, pp. 1155-1166, 2017, https://doi.org/10.1109/ACCESS.2017.2778011.

[43] J. M. Brown *et al.*, “Automated diagnosis of plus disease in retinopathy of prematurity using deep convolutional neural networks,” *JAMA ophthalmology*, vol. 136, no. 7, pp. 803-810, 2018, https://doi.org/10.1001/jamaophthalmol.2018.1934.

[44] V. Gulshan *et al.*, “Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs,” *jama*, vol. 316, no. 22, pp. 2402-2410, 2016, https://doi.org/10.1001/jama.2016.17216.

[45] A. S. Coyner *et al.*, “Automated fundus image quality assessment in retinopathy of prematurity using deep convolutional neural networks,” *Ophthalmology retina*, vol. 3, no. 5, pp. 444-450, 2019, https://doi.org/10.1016/j.oret.2019.01.015.

[46] P. Rajpurkar *et al.*, “Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning,” *arXiv preprint arXiv:1711.05225*, 2017, https://doi.org/10.48550/arXiv.1711.05225.

[47] L. Jones, D. Golan, S. Hanna, and M. Ramachandran, “Artificial intelligence, machine learning and the evolution of healthcare: A bright future or cause for concern?,” *Bone & joint research*, vol. 7, no. 3, pp. 223-225, 2018, https://doi.org/10.1302/2046-3758.73.BJR-2017-0147.R1.

[48] J. De Fauw *et al.*, “Clinically applicable deep learning for diagnosis and referral in retinal disease,” *Nature medicine*, vol. 24, no. 9, pp. 1342-1350, 2018, https://doi.org/10.1038/s41591-018-0107-6.

[49] J. Moor, “The Dartmouth College artificial intelligence conference: The next fifty years,” *Ai Magazine*, vol. 27, no. 4, pp. 87-91, 2006, https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1911.

[50] P. Ritter, M. Schirner, A. R. McIntosh, and V. K. Jirsa, “The virtual brain integrates computational modeling and multimodal neuroimaging,” *Brain connectivity*, vol. 3, no. 2, pp. 121-145, 2013, https://doi.org/10.1089/brain.2012.0120.

[51] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights into imaging*, vol. 9, pp. 611-629, 2018, https://doi.org/10.1007/s13244-018-0639-9.

[52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929-1958, 2014, https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer.

[53] S. Kiranyaz, T. Ince, R. Hamila and M. Gabbouj, “Convolutional Neural Networks for patient-specific ECG classification,” *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2608-2611, 2015, https://doi.org/10.1109/EMBC.2015.7318926.

[54] S. Kiranyaz, T. Ince, and M. Gabbouj, “Personalized monitoring and advance warning system for cardiac arrhythmias,” *Scientific reports*, vol. 7, no. 1, p. 9270, 2017, https://doi.org/10.1038/s41598-017-09544-z.

[55] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, and D. J. Inman, “Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks,” *Journal of Sound and Vibration*, vol. 424, pp. 158-172, 2018, https://doi.org/10.1016/j.jsv.2018.03.008.

[56] O. Avci, O. Abdeljaber, S. Kiranyaz, and D. Inman, “Structural damage detection in real time: implementation of 1D convolutional neural networks for SHM applications,” *Structural Health Monitoring & Damage Detection*, pp. 49-54, 2017, https://doi.org/10.1007/978-3-319-54109-9_6.

[57] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, “Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks,” *Journal of Sound and Vibration*, vol. 388, pp. 154-170, 2017, https://doi.org/10.1016/j.jsv.2016.10.043.

[58] O. Avci, O. Abdeljaber, S. Kiranyaz, B. Boashash, H. Sodano, and D. J. Inman, "Efficiency validation of one dimensional convolutional neural networks for structural damage detection using a SHM benchmark data," *5th International Congress on Sound and Vibration 2018*, pp. 4600-4607, 2018, https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1362683.

[59] O. Abdeljaber, O. Avci, M. S. Kiranyaz, B. Boashash, H. Sodano, and D. J. Inman, "1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data," *Neurocomputing*, vol. 275, pp. 1308-1317, 2018, https://doi.org/10.1016/j.neucom.2017.09.069.

[60] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067-7075, 2016, https://doi.org/10.1109/TIE.2016.2582729.

[61] S. Kiranyaz, A. Gastli, L. Ben-Brahim, N. Al-Emadi, and M. Gabbouj, "Real-time fault detection and identification for MMC using 1-D convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8760-8771, 2019, https://doi.org/10.1109/TIE.2018.2833045.

[62] O. Abdeljaber, S. Sassi, O. Avci, S. Kiranyaz, A. A. Ibrahim, and M. Gabbouj, "Fault detection and severity identification of ball bearings by online condition monitoring," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 10, pp. 8136-8147, 2019, https://doi.org/10.1109/TIE.2018.2886789.

[63] L. Eren, T. Ince, and S. Kiranyaz, "A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier," *Journal of Signal Processing Systems*, vol. 91, pp. 179-189, 2019, https://doi.org/10.1007/s11265-018-1378-3.

[64] P. A. Opala, "Influence of lime and phosphorus application rates on growth of maize in an acid soil," *Advances in Agriculture*, vol. 2017, 2017, https://doi.org/10.1155/2017/7083206.

[65] P. Sukarno, K. A. Sidarto, A. Trisnobudi, D. I. Setyoadi, N. Rohani, and D. Darmadi, "Leak detection modeling and simulation for oil pipeline with artificial intelligence method," *Journal of Engineering and Technological Sciences*, vol. 39 B, no. 1, pp. 1-19, 2007, https://doi.org/10.5614/itbj.eng.sci.2007.39.1.1.

[66] M. Chaczykowski, A. J. Osiadacz, and F. Uilhoorn, "Exergy-based analysis of gas transmission system with application to Yamal-Europe pipeline," *Applied Energy*, vol. 88, no. 6, pp. 2219-2230, 2011, https://doi.org/10.1016/j.apenergy.2010.12.045.

[67] R. Ben-Mansour, M. Habib, A. Khalifa, K. Youcef-Toumi, and D. Chatzigeorgiou, "Computational fluid dynamic simulation of small leaks in water pipelines for direct leak pressure transduction," *Computers & Fluids*, vol. 57, pp. 110-123, 2012, https://doi.org/10.1016/j.compfluid.2011.12.016.