Webology (ISSN: 1735-188X)
Volume 19, Number 6, 2022

Localization Of Multiple Leaks In Pipelines Using Decision Trees And Support Vector Machines

Msc. July Andrea Gomez Camperos[1], Msc. Haidee Yulady Jaramillo[2], Phd. Sir Alexci Suárez Castrillón [3]

[1] Mechanical Engineering Department, Faculty of Engineering, University Francisco de Paula Santander Sectional Ocaña, Research Group on New Technologies, Sustainability and Innovation (GINSTI), Ocaña, Norte de Santander

[2] Department of Civil Engineering, Faculty of Engineering, University Francisco de Paula Santander Sectional Ocaña, Research Group on New Technologies, Sustainability and Innovation (GINSTI), Ocaña, Norte de Santander

[3] Department of Systems Engineering, Faculty of Engineering, Universidad Francisco de Paula Santander Seccional Ocaña, Grupo de Investigación en Ciencias y Tecnología (GRUCITE), Ocaña, Norte de Santander.

*Corresponding Author: Msc. July Andrea Gomez Camperos

Abstract:

Pipeline transport is widely used in industrial production and daily life. To reduce the waste of resources and economic losses caused by pipeline leaks, leak detection, localization and estimation systems are implemented in liquid pipelines. To minimize leak interpretation errors, leak detection algorithms based on artificial intelligence (AI) and data analysis have been developed. This study proposes a scheme for the detection and localization of multiple sequential leaks, based on the combination of two techniques such as decision trees and support vector machines. The results show that the proposed models have high accuracy, precision, recall and F1 score of 99.9%, 99.7%, respectively, which are better than the traditional classification model.

Keywords: Leak detection, Leak localization, decision tres, support vector machines.

1. Introduction

Pipelines are the most efficient and safest means of transportation(Li et al. 2019), however due to phenomena such as corrosion, vibration and pipe wear, fluid leaks appear, there are several works that focus attention on detection and location of a single leak (Xianming et al. 2020).

During the last decades, a variety of non-invasive methods have been introduced for leak detection in piping systems. One group of these are hardware-based methods such as acoustic techniques that rely on the measurement of the sound of water leaks in the pipeline (Ozevin and Harding 2012).

Other methods are non-invasive methods which are model-based techniques that take advantage of hydraulic simulation to model the leakage behavior of piping systems. One group of model-based methods is based on a constant and extended period analysis, also known as non-transient methods (Houshang Ayati, Haghighi, and Jose Lee 2019).

Recently, advances in machine learning offer exciting opportunities for automatic analysis of big data for pipeline assessment. For example, machine learning models were developed and applied to automatically detect leaks by analyzing pressure data and locate leak positions by using sound data(Zhou et al. 2021). In addition to the use of numerical data, images and videos were used to quantitatively evaluate pipeline defects based on computer vision approaches. Recent research and development showed that machine learning approaches were promising for automatic detection, classification, localization and quantification of pipeline anomalies (Liu and Bao 2022).

2. Materials and methods

2.1 Support Vector Machine (SVM)

SVM is a machine learning algorithm presented by Vapnik in 1995, which is based on statistical learning theory and the principle of structural risk minimization. The input space is transformed into a high-dimensional linearly differentiable space using SVM by defining an appropriate kernel function. Then a nonlinear transformation is implemented to find the optimal linear hyperplane of the high-dimensional space (Yang et al. 2022).

Support vector machines are widely used for both classification and regression. The binary SVM is a linear classifier on the input space to find an optimal boundary, separating one class from the other. Specifically, given the training data. An example of support vector machine used in classification is shown in Figure 1.

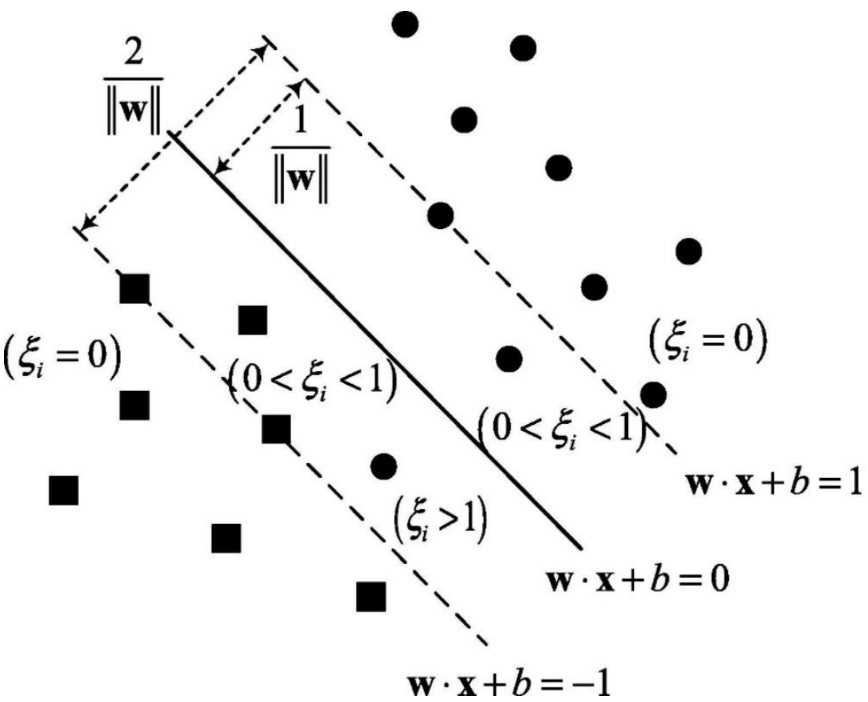

Figure 1. Support vector machine for data classification.

Fuente:(Xiao, Hu, and Li 2019)

2.2 Decision trees

A decision tree (DT) is a white box that represents your decisions through a tree-like structure composed of a set of nodes containing test conditions (internal nodes) and class labels (leaf nodes). The nodes are linked by arcs, which symbolize the possible outcomes of each test condition. DTs are notable for their simplicity and high level of interpretability. Since the DTI process determines attribute importance when creating test conditions, it provides an integrated feature selection mechanism. These features, together with its predictive power, make the DT one of the most widely used classifiers(Rivera-Lopez et al. 2022).

A normal tree includes root, branches and leaves. The same structure is followed in the Decision Tree. It contains root node branches and leaf nodes. The test of an attribute is performed on each internal node, the result of the test is in the branch and the class label as a result is in the leaf node.

Decision Tree is similar to the human decision-making process and so that it is easy to understand. It can solve in both situations whether one has discrete or continuous data as input(Charbuty and Abdulazeez 2021). figure 2 shows the structure of a decision tree.

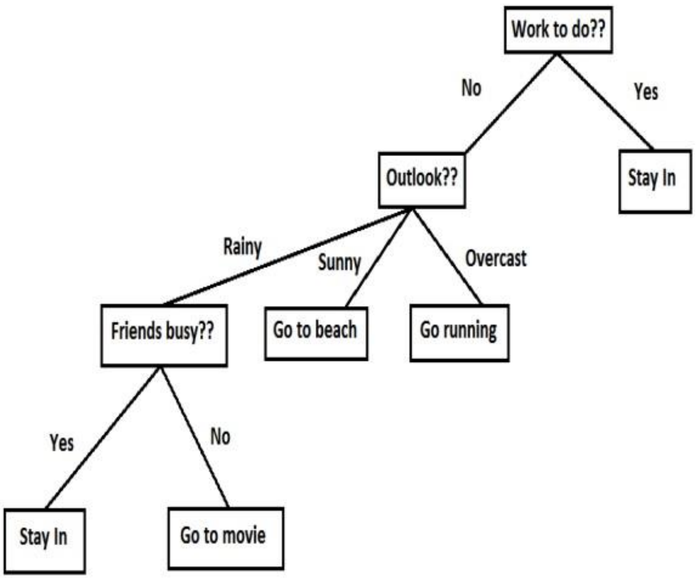

Figure 2. Example of a decision tree
Fuente:(Patel and Prajapati 2018)

2.3 Data set

In order to obtain the data of multiple leaks in pipes, an experimental pipeline developed in the automation and control laboratory of the Francisco de Paula Santander Ocaña University was used, together with a program developed in labview for data collection, as shown in Figure 3,

Figure 3. experimental piping with data acquisition system

The experimental piping includes PVC pipe, two pressure sensors, two flow sensors and two water pumps. There are four possible leak locations, the size of which can be changed using the manual valves that were placed across the bench At first we assumed only one leak at a time. and obtained inlet pressure and outlet pressure data, same as the inlet flow and outlet flow and leak location data as shown in the LabVIEW software simulation in Figure 4 that we set up to acquire and test the proposed algorithms.

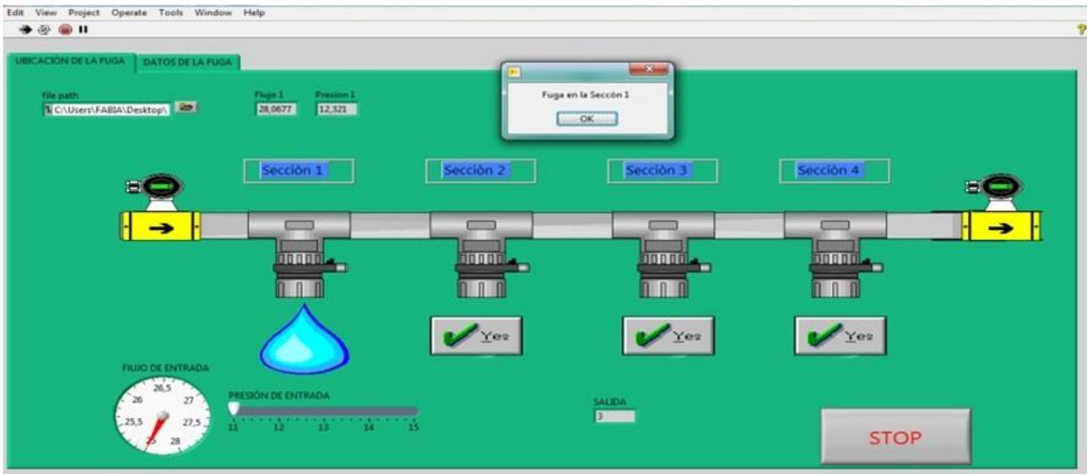

Figure 4. multiple leakage location algorithm check screen

3. Results

3.1 Support Vector Machine (SVM)

The methodology shown in Figure 5 was used for this case, which is divided into 5 steps.

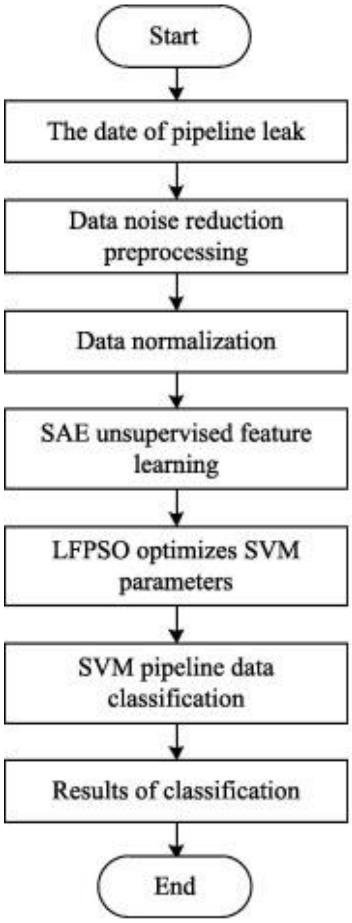

Figure 5. Example of a decision tree
Fuente:(Wang et al. 2020)

- Step 1. Data collection

For this case 25021 data were collected containing values of inlet and outlet pressure of the pipe, as well as the inlet and outlet flow of the pipe with respect to whether there is a leak or not, and in which section the leak originated, for this was classified with 0 if there is no leak, 1 if the leak is in section 1, 2 if the leak is in section 2 and 3 if the leak is in section 3 and finally 4 if the leak is in section 4.

- Step 2. Data preprocessing

For data processing, we used the Google colab program with phyton programming language, as shown in Figure 6.

```
Leak_df = pd.read_csv("datos fugas1.csv",delimiter=";")
Leak_df [0:100000]
```

	P1	P2	FLUJO 1	FLUJO 2	Leak
0	12555740	4723002	28135817	19159066	1
1	12561809	4703281	28172222	19150703	1
2	12550683	4716934	28161399	19144635	1
3	12580775	4717692	28155495	19147505	1
4	12592154	4721181	28174583	19151982	1
...	...	...	...	...	...
25016	14559995	6118623	26144205	24691007	0
25017	14553319	6121657	26127872	24691598	0
25018	14556657	6124084	26077103	24655782	0
25019	14559995	6121050	26125117	24685891	0
25020	14562270	6130759	26104062	24683529	0

25021 rows × 5 columns

Figure 6. data processing

- Step 3. Feature extraction

Figure 7 shows the behavior of the data with reference to the input and output variables. The data are then divided into training data and test data as shown in the figure 8 below.

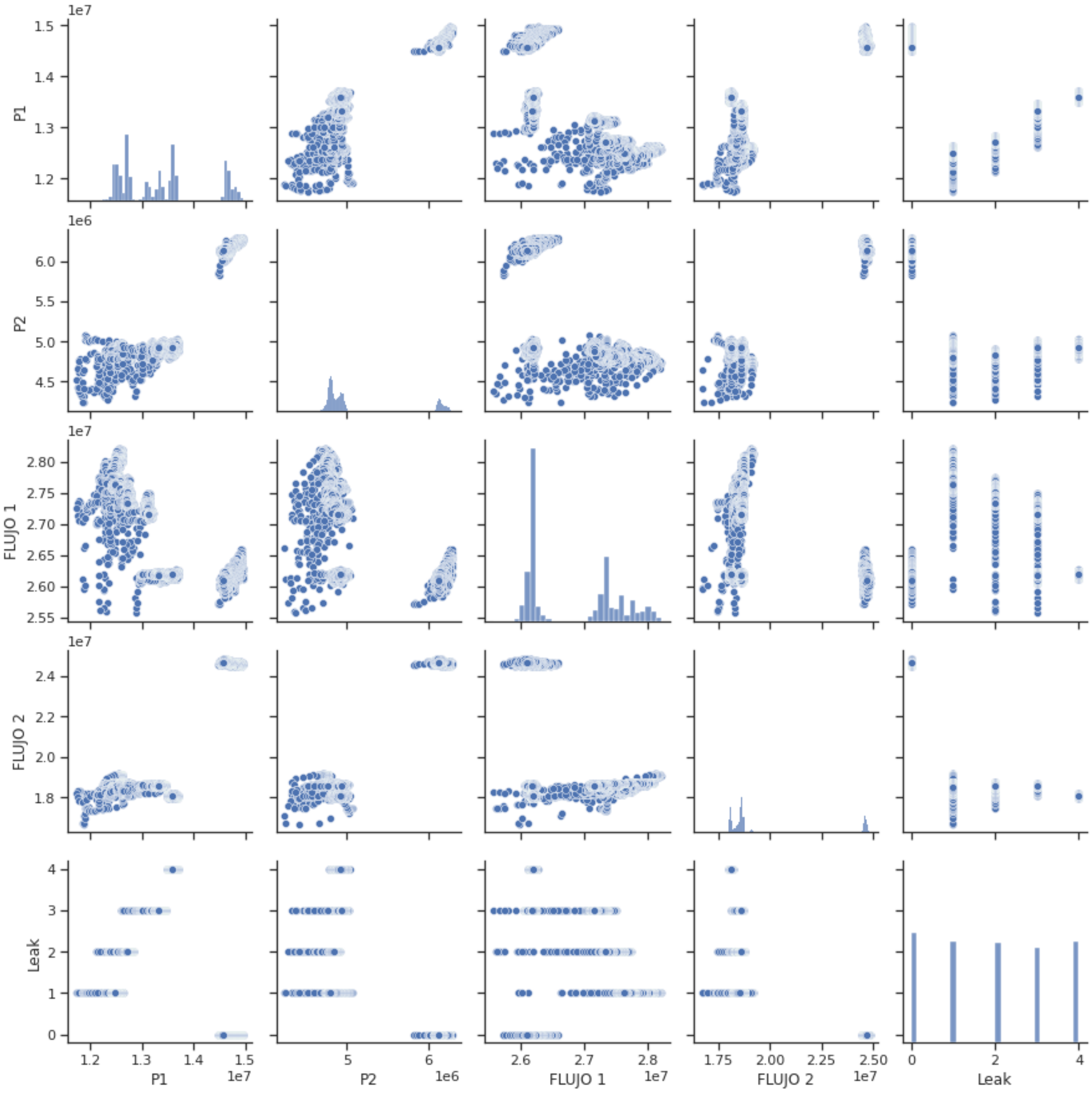

Figure 7. feature extraction

```
[ ]  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4)
     print ('Train set:', X_train.shape, y_train.shape)
     print ('Test set:', X_test.shape, y_test.shape)

     Train set: (20016, 4) (20016,)
     Test set: (5005, 4) (5005,)
```

Figure 8. algorithm for splitting training and test data

- Step 4. Data classification and recognition

In this step the algorithm is trained, for this research a support vector sorting machine algorithm as shown in the figure 9.

```
[ ] from sklearn import svm
    clf = svm.SVC(kernel='poly')

    clf.fit(X_train, y_train)

    SVC(kernel='poly')
```

```
[ ] C = 1.0  # parametro de regulacion SVM
    svc = svm.SVC(kernel='linear', C=C).fit(X, y)
    rbf_svc = svm.SVC(kernel='rbf', gamma=0.7, C=C).fit(X, y)
    poly_svc = svm.SVC(kernel='poly', degree=3, C=C).fit(X, y)
    lin_svc = svm.LinearSVC(C=C).fit(X, y)
```

Figure 9. support vector machine algorithm for classification

- Step 5. Finalize

For testing the support vector machine algorithm to identify multiple leaks, the confusion matrix is found as shown in the figure 10 below.

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1053
           1       0.99      1.00      1.00      1020
           2       0.99      0.99      0.99       996
           3       1.00      0.99      1.00       923
           4       1.00      1.00      1.00      1013

    accuracy                           1.00      5005
   macro avg       1.00      1.00      1.00      5005
weighted avg       1.00      1.00      1.00      5005

Confusion matrix, without normalization
[[1053    0    0    0    0]
 [   0 1017    3    0    0]
 [   0    6  990    0    0]
 [   0    0    5  918    0]
 [   0    0    0    0 1013]]
```

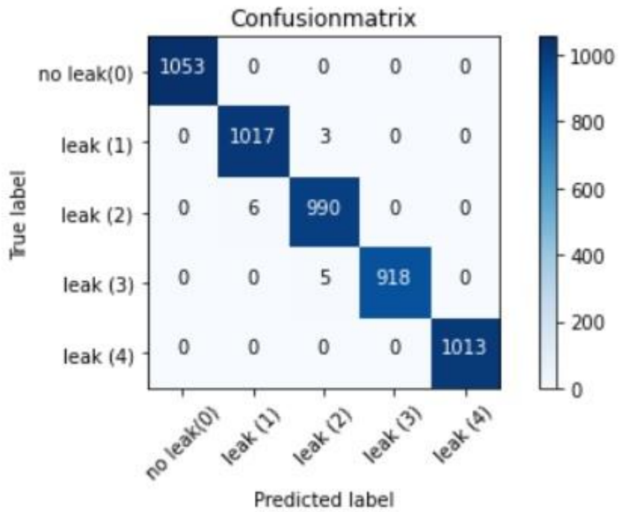

Figure 10. confusion matrix

With the confusion matrix it can be analyzed that the prediction algorithm has an accuracy of 100% in the case of detecting that there is no leakage, that the output is 0, and that 1017 data successfully detected leakage in section 1, and only 6 misclassified data, it can also be observed 990 data detected leakage in section 2, 918 in section 3 and finally 1013 in section 4.

3.2 Decision tres

For the decision tree technique, the data set obtained from the experiment is divided in two: 70% training data and 30% test data, then the Machine Learning algorithm for decision trees is imported into the Google colab tool as shown in figure 11.

```
[ ]  print ('Test set:', X_test.shape, y_test.shape)

     Test set: (7507, 4) (7507, 1)
```

```
[ ]  from sklearn.tree import  DecisionTreeClassifier
     arbol= DecisionTreeClassifier ()
     arbol_1= arbol.fit (X_train,y_train)
```

```
from matplotlib import pyplot as plt
from sklearn import tree

print(f"Profundidad del árbol: {arbol.get_depth()}")
print(f"Número de nodos terminales: {arbol.get_n_leaves()}")
predTree = arbol.predict(X_test)
print (predTree [0:4])
print (y_test [0:4])
from sklearn import metrics
print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_test, predTree))
```

```
Profundidad del árbol: 14
Número de nodos terminales: 35
[4 2 0 2]
       Leak
15622     4
5622      2
24309     0
7705      2
DecisionTrees's Accuracy:  0.9992007459704276
```

Figure 11. Training algorithm

Figure 11 shows that the decision tree has a depth of 14 and the number of nodes is 35 with a hit rate of 0.999.

Conclusions

An SVM classifier is basically an algorithm that maximizes the distance between two classes, minimizing classification error. The SVM searches, from a set of training data, for a hyperplane that separates the two classes to which they belong. This hyperplane can be as simple as a straight line in the case of linearly separable data, or it can be composed of many decision boundaries that form a more complex hyperplane.

This paper has described a method for locating multiple leaks in a pipe network by processing pressure and flow values obtained at various points in the network using Support Vector Machines (SVM) and decision trees. The work is based on the concept that the required information can be obtained by sufficiently sophisticated processing of pressure and flow data in a network and that SVMs are suitable for this processing task**.**

When applying Decision Trees, we are using a supervised machine learning technique that is very easy to understand, this technique takes a series of decisions in the form of a tree. decisions in the form of a tree. In addition, the color of the nodes is more intense the more certain the classification is.

and each color of the tree represents a class, so it helps to verify which measurement point is being evaluated. The color of the nodes is more intense the more certain the classification is and each color of the tree represents a class, so it helps to verify which measurement point is being evaluated in the case of the dataset data used.

Bibliography

Charbuty, Bahzad, and Adnan Abdulazeez. 2021. "Classification Based on Decision Tree Algorithm for Machine Learning." Journal of Applied Science and Technology Trends 2(01):20–28. doi: 10.38094/jastt20165.

Houshang Ayati, Amir, Ali Haghighi, and Pedro Jose Lee. 2019. "Statistical Review of Major Standpoints in Hydraulic Transient-Based Leak Detection." JOURNAL OF HYDRAULIC STRUCTURES Shahid Chamran University of Ahvaz Journal of Hydraulic Structures J. Hydraul. Struct 5(1):1–26. doi: 10.22055/jhs.2019.27926.1095.

Li, Juan, Qiang Zheng, Zhihong Qian, and Xiaoping Yang. 2019. "A Novel Location Algorithm for Pipeline Leakage Based on the Attenuation of Negative Pressure Wave." Process Safety and Environmental Protection 123:309–16. doi: 10.1016/j.psep.2019.01.010.

Liu, Yiming, and Yi Bao. 2022. "Review on Automated Condition Assessment of Pipelines with Machine Learning." Advanced Engineering Informatics 53(June):101687. doi: 10.1016/j.aei.2022.101687.

Ozevin, Didem, and James Harding. 2012. "Novel Leak Localization in Pressurized Pipeline Networks Using Acoustic Emission and Geometric Connectivity." International Journal of Pressure Vessels and Piping 92:63–69. doi: 10.1016/j.ijpvp.2012.01.001.

Patel, Harsh H., and Purvi Prajapati. 2018. "Study and Analysis of Decision Tree Based Classification Algorithms." International Journal of Computer Sciences and Engineering 6(10):74–78. doi:

10.26438/ijcse/v6i10.7478.

Rivera-Lopez, Rafael, Juana Canul-Reich, Efrén Mezura-Montes, and Marco Antonio Cruz-Chávez. 2022. “Induction of Decision Trees as Classification Models through Metaheuristics.” Swarm and Evolutionary Computation 69(September 2021):101006. doi: 10.1016/j.swevo.2021.101006.

Wang, Chuang, Fei Han, Yong Zhang, and Jingyi Lu. 2020. “An SAE-Based Resampling SVM Ensemble Learning Paradigm for Pipeline Leakage Detection.” Neurocomputing 403:237–46. doi: 10.1016/j.neucom.2020.04.105.

Xianming, Lang, Li Ping, Zhang Baocun, Cao Jiangtao, Guo Ying, Kan Zhe, and Lu Siyu. 2020. “Localization of Multiple Leaks in a Fluid Pipeline Based on Ultrasound Velocity and Improved GWO.” Process Safety and Environmental Protection 137:1–7. doi: 10.1016/j.psep.2020.02.006.

Xiao, Rui, Qunfang Hu, and Jie Li. 2019. “Leak Detection of Gas Pipelines Using Acoustic Signals Based on Wavelet Transform and Support Vector Machine.” Measurement: Journal of the International Measurement Confederation 146:479–89. doi: 10.1016/j.measurement.2019.06.050.

Yang, Dandi, Nan Hou, Jingyi Lu, and Daan Ji. 2022. “Novel Leakage Detection by Ensemble 1DCNN-VAPSO-SVM in Oil and Gas Pipeline Systems.” Applied Soft Computing 115:108212. doi: 10.1016/j.asoc.2021.108212.

Zhou, Mengfei, Yanhui Yang, Yinze Xu, Yinchao Hu, Yijun Cai, Junjie Lin, and Haitian Pan. 2021. “A Pipeline Leak Detection and Localization Approach Based on Ensemble TL1DCNN.” IEEE Access 9:47565–78. doi: 10.1109/ACCESS.2021.3068292.