Webology (ISSN: 1735-188X)
Volume 19, Number 4, 2022

Application Of Machine Learning Techniques For Leak Detection In Horizontal Pipelines

July Andrea. Gomez Camperos

Department of Mechanical Engineering, Faculty of Engineering, Universidad Francisco de Paula Santander Ocaña, Ocaña, Colombia.

Abstract— Pipelines are considered as the safest way to transport oil and gas, critical pipeline failures such as leaks affect the reliability of fluid transport systems causing environmental damage, economic losses and pressure reduction in the pipeline, so this paper presents a methodology to detect leaks in pipelines by using machine learning techniques that, by introducing process data from the experimental pipeline, systematically determine whether or not there are leaks. For this research, two machine learning classification techniques, support vector machines and decision trees, were evaluated and four sensors, two for flow and two for pressure, were used in a ½ inch diameter horizontal experimental pipeline installed in the automation and control laboratory of the Universidad Francisco de Paula Santander Ocaña. With the experimental data, a complete database was created and used for training and validation of each of the machine learning techniques used. As a result, a leak detection method was obtained, using a data set for training, validation and testing and with accuracy levels higher than 97% in leak detection.

I. INTRODUCTION

Pipelines are used to transport fluids over long distances and are typically found in water distribution systems and in the petrochemical industry. However, pipelines are susceptible to leaks due to pipeline defects such as corrosion, fatigue cracks and dents, A fluid leak can harm the environment, living beings; create hazardous conditions; cause financial losses; and/or damage the pipeline itself and the instruments installed for monitoring the pipeline.

Pipeline leak detection methods can be divided into hardware-based methods and software-based methods. The hardware methods can be divided into pipeline inspection methods that include the use of ground infrared thermography[1], acoustic methods [2], methods based on pressure sensors[3], ground penetrating radar based methods[4], [5],[6]. The disadvantage of these methods is that they have higher capital and operating costs compared to hardware-based methods, as they require more instrumentation/equipment and more person-hours to implement and maintain the Leak Detection System. In addition, most hardware methods do not provide continuous monitoring due to the high costs of sampling and analysis.

Computational methods, also known as software methods and pipe internal leak detection methods, work on the basis of a computer algorithm. As they generally require fewer sensors, they are less expensive to implement and maintain.

Software methods can be divided into signal processing methods[7], real-time modeling methods

[8], artificial neural networks[9], harmonic analysis methods[10], pattern recognition methods, among others.[11].

Recently with the development of Industry 4.0, several machine learning (ML) methods have been proposed for leak detection systems based on pressure monitoring. In[12],[13] , the application of artificial neural networks (ANN) in leak detection and localization is experimented and discussed. As one of the commonly applied ML methods, support vector machine (SVM) is also implemented in leak detection, as shown in [14] and [15] . In [16], the k-nearest neighbor (KNN) algorithm is adopted for pipeline leak detection.

This paper uses two machine learning techniques to detect leakage in water pipes, which are:
the support vector machine (SVM), and the decision tree algorithm (DT), the model detects the leak using two operating parameters such as flow rate and flow, both inlet and outlet of the pipe, for this research an experimental pipe was used, which was installed in the Automation laboratory of the Universidad Francisco de Paula Santander Ocaña and is described in section 3.

the article is organized as follows: Section 2 describes the algorithms used, Section 3 explains the materials and methods used, Section 4 shows Results and Discussion and finally Section 5 presents the conclusions.

II. DESCRIPTION OF ALGORITHMS USED

A. Vector Support Machine

Support vector machines (SVMs) are considered a great tool for linear or non-linear classification. An SVM classifier is basically an algorithm that maximises the distance between two classes while minimising 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, as in Figure 1, or it can be composed of many decision boundaries that form a more complex hyperplane.

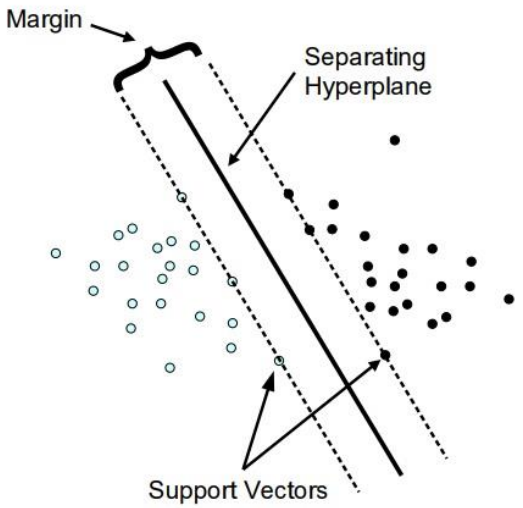

Figure 1. Classification (linear separable case)[17].

To figure axis labels, use words rather than symbols. Do not label axes only with units. Do not label axes with a ratio of quantities and units. Figure labels should be legible, about 9-point type.

Color figures will be appearing only in online publication. All figures will be black and white graphs in print publication.

B. Decision trees

Decision Trees are statistical algorithms or machine learning techniques that allow us to build predictive data analytics models for Big Data based on their classification according to certain characteristics or properties, or on regression through the relationship between different variables to predict the value of another. On the other hand, decision trees provide high-level efficiency and easy interpretation. These two benefits make this simple algorithm popular in the machine learning space.

The decision tree is a structure that is made up of branches and nodes of different types, read from the top down. See figure 2.

- The internal nodes represent each of the features or properties to be considered in making a decision.
- The branches represent the decision based on a certain condition (e.g. probability of occurrence).
- The end nodes represent the outcome of the decision, i.e. the prediction of the class and have no conditions or nodes below them. They are also called "child nodes".

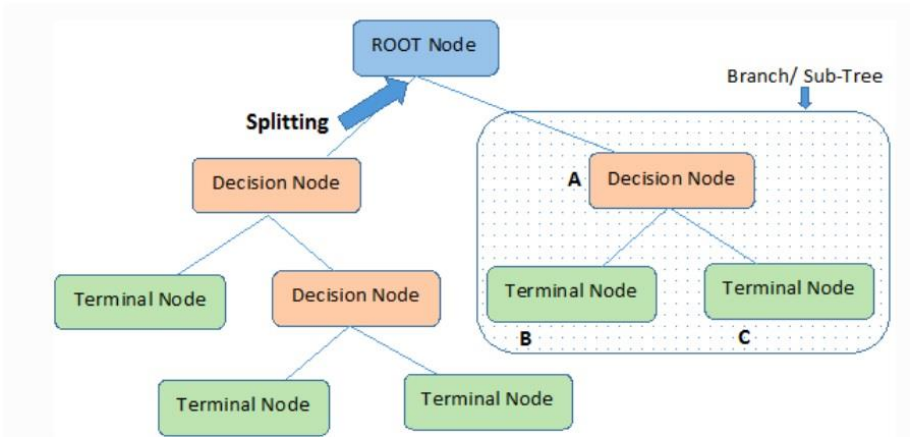

Figure 2. DT classifier [18]

III. MATERIALS AND METHODS

A. Methodology

The methodology followed during this study includes important steps to build a Machine Learning model. The first step consists of collecting the necessary data set and a preprocessing phase. The second step consists of training the proposed model and evaluating its performance. Figure 3 summarizes the methodology of this study.

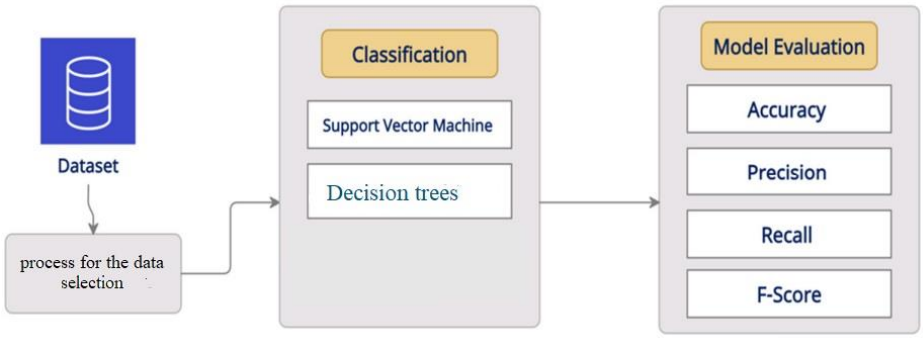

Figure 3. Methodology

B. Research model

The experimental bench was carried out in the automation and control laboratory of the University Francisco de Paula Santander Ocaña, was used for the assembly two square glass containers of 0.4 m long, 0.3 m wide and 0.3 m high with a thickness of 5 mm, two pumps to transfer the fluid from one tank to another, 4 bases or supports to maintain a $\Delta z = 0$ and ½ inch pipes and fittings as described in the following section [19].

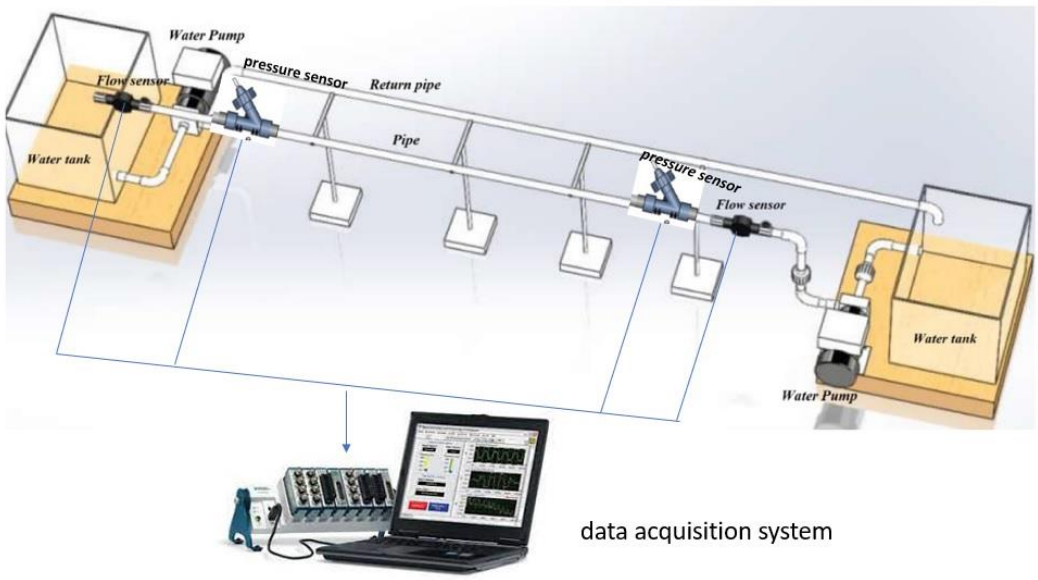

Figure 4. model of the experiment

The experiment had four sensors, two pressure sensors and two flow sensors each located at the end of a horizontal pipe, with analog signal inputs from 4 to 20 mA, connected to a National Instruments data acquisition system, using a NI cDAQ-9178 chassis with a NI 9203 module to capture the values of the variables of inlet flow, outlet flow, inlet pressure and outlet pressure of the pipe and transmit them to a database accessible through a virtual instrument that was developed and described in [19].

The model of the experiment is shown in Figure 4.

C. Data Acquisition

Data were obtained from the experimental bench described in section III.2, from the two flow sensors and the two pressure sensors installed in the experimental bench through a virtual instrument developed in LabVIEW. Data were collected on the pipe network in normal operation without leakage. And then a leak is performed in the middle of the experimental bench for data collection with leakage. Then these data are brought to an Excel document, where they are processed by Google colab (phython) tool.

the process for the data selection of the experimental bench is shown in figure 5.

It should be noted that the pre-processing of the data is done through the calibration of the flow and pressure sensors, obtaining a mathematical formula to convert the current values to pressure and flow units.

after the results of the data collection are obtained, they are exported to an excel file, which is analyzed in google colab for data processing.

Figure 6 illustrates how the inlet pressure (P1) and outlet pressure (P2) data are displayed when there is leakage and when there is no leakage, as well as the inlet flow (F1) and outlet flow (F2) data analyzed by the google colab tool.

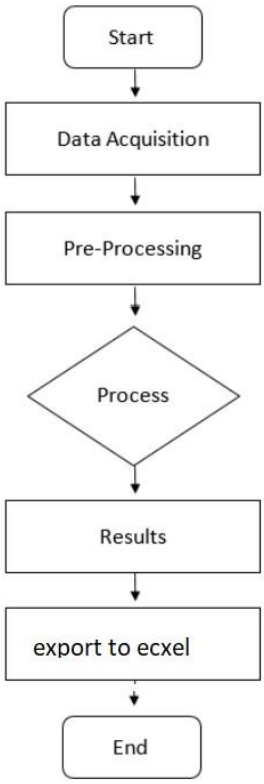

Figure 5. process for the data selection

```
# Importacion del dataset
dataset = pd.read_csv("Datos TOTAL 2.csv",delimiter=";")
X = dataset.iloc[:, [0, 3]].values
y = dataset.iloc[:, 4].values
dataset
```

	P1	P2	F1	F2	Leak
0	1187863	1698144	6972500	6770399	0
1	2230266	2761850	12065172	11718466	0
2	3102495	3820702	17194051	16646855	0
3	3851204	4902005	22293807	21546119	0
4	4495027	5989983	27385691	26491629	0
...	...	...	...	...	...
2580	13075997	5462074	26369518	24161644	1
2581	13078424	5468142	26416942	24156921	1
2582	13080852	5472390	26441343	24149640	1
2583	13083280	5468142	26443311	24143737	1
2584	13081611	5471176	26452756	24180930	1

2585 rows × 5 columns

Figure 6. data visualization code

IV. RESULTS AND DISCUSSION

A. decision tree technique

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 7.

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

print ( f "Tree depth:  {tree.get_depth()} " )
print ( f "Number of terminal nodes:  {arbol.get_n_leaves()} " )
predTree = arbol.predict(X_test)
print (predTree [0:4])
print (y_test [0:5])
from sklearn import metrics
print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_test, predTree))
```

```
Tree depth: 4
Number of terminal nodes: 5
['YES' 'NO' 'NO' 'NO']
     Leak
1776 YES
1287 NO
59 NO
605 NO
2573 YES
DecisionTrees's Accuracy: 0.9987113402061856
```

Figure 7. training algorithm

Figure 7 shows that the decision tree has a depth of 4 and the number of nodes is 5 with a hit rate of 0.998711340202061856. Figure 8 visualizes the decision tree.

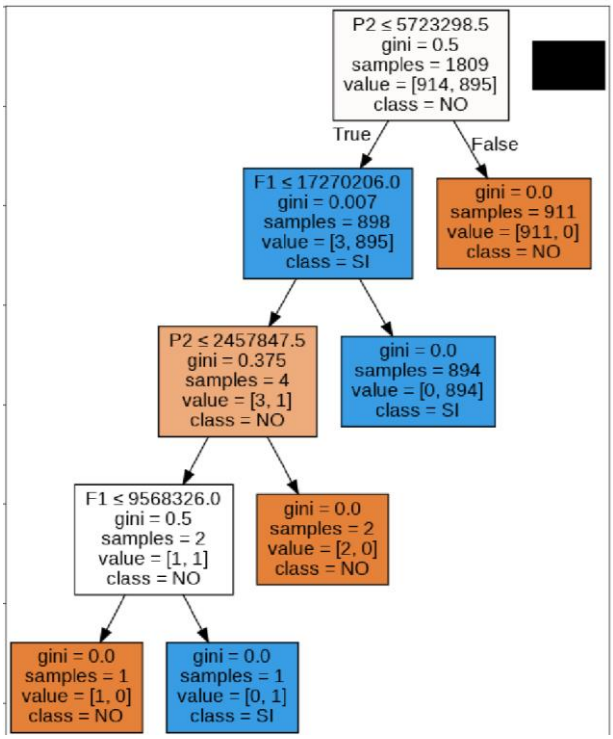

Figure 8 decision tree

The interpretation of the decision tree would be: if the pressure P2 which is the outlet pressure of the system is less than or equal to 5,723 psi:

if this condition is false then there is no leakage. But if the condition is true it is necessary to evaluate if the flow F1 which is the inlet flow is less or equal to 17,270 L/min if the condition is

false there is a leak, but if the condition is true it is necessary to evaluate if the pressure P2 is less or equal to 2,457 Psi, if this condition is false there is no leak in the System, but if the condition is true it is necessary to evaluate if the F1 is less or equal to 9,568 L/min if this condition is false there is a leak in the System but if this condition is true there is no leak in the system.

B. Vector Support Machine

The Support Vector Machines (SVM) technique was used to build and train the model to determine leakage and non-leakage from flow and pressure records. Table 1 shows the results of the evaluation of the models on the water leakage dataset.

TABLE I

Classifier	Precision	Recall	F1-Score
SVM	1	1	0.998

Figure 9 show the confusion matrix for the SVM model.

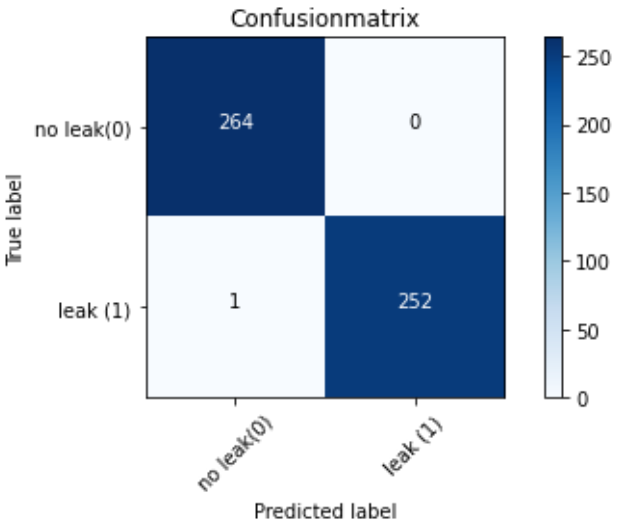

Figure 9. Confusion or error matrix.

The confusion matrix shows that the model misclassified 1 samples in the "no leak" class (0) and 0 samples in the "Leak" class (1). The confusion matrix shows a great improvement in the model's ability to distinguish between the two classes.

The evaluation of the final performance of the algorithm in terms of the number of hits and misses the training set had in making predictions is shown in Figure 10.

```
from sklearn.metrics import accuracy_score
hit_rate = accuracy_score(y_test,yhat)
hit_rate
```

```
0.9980657640232108
```

```
miss_rate =  1  - hit_rate
n_hits =  int ( round (n_examples_test*hits_rate))
n_failures =  int ( round (n_examples_test*failure_rate))
print ( 'Examples of test: %d'  % n_examples_test)
print ( 'Examples correctly classified: %d'  % n_hits)
print ( 'Misclassified examples: %d'  % n_failures)
print ( 'Hit rate: %.3f'  % hit_rate)
print ( 'Failure rate: %.3f'  % failure_rate)
```

```
Test examples: 517
Correctly classified examples: 516 Misclassified
examples: 1 Hit rate
: 0.998 Failure
rate: 0.002
```

Figure 10. The evaluation of the final performance of the algorithm

Figure 10 shows that there were 516 correctly classified examples and only 1 misclassified example, with a rate of 0.998 and a failure rate of 0.002.

V. CONCLUSION

The techniques used for this study are Decision Trees and SVM. Both algorithms are supervised learning algorithms that classify the data to be processed very well. In this research, the two machine learning techniques were implemented on a data set provided by the experiment carried out in an experimental bench of horizontal pipes described in section 3 of materials and methods. Pressure and mass flow measurements taken at the pipe inlet and outlet were sampled.

By applying Decision Trees, a supervised machine learning technique is being used which is very easy to understand, this technique makes a series of decisions in the form of a tree. In addition, the colour of the nodes is more intense the more certain the classification is and each colour of the tree represents a class, so it helps to verify which measurement point is being evaluated in the case of the data used.

By applying SVM to the data set of the experiment in question, it was possible to verify that it is an effective method in the processing of high-dimensional data, in addition to the fact that it uses a subset of training points in the decision function called support vectors, so it is also efficient in memory.

In SVM we trained all the data obtained from the experiment, divided into 20% test data and 80% training data, finally obtaining a hit rate of 0.9987 according to the Jaccard index metric of the sklearn Library and in decision trees we trained with all the complete data divided into 70% training and 30% test obtaining a hit rate of 0.9987 according to the accuracy score metric of sklearn.

REFERENCES

[1] M. H. Manekiya and P. Arulmozhivarman, “Leakage detection and estimation using IR thermography,” Int. Conf. Commun. Signal Process. ICCSP 2016, vol. 632014, pp. 1516–

1519, 2016, doi: 10.1109/ICCSP.2016.7754411.

[2] Y. Mahmutoglu and K. Turk, "A passive acoustic based system to locate leak hole in underwater natural gas pipelines," Digit. Signal Process. A Rev. J., vol. 76, pp. 59–65, 2018, doi: 10.1016/j.dsp.2018.02.007.

[3] L. Liang, K. Feng, G. Xu, Z. Zhu, and X. Zhou, "Pipeline Leakage Test Based on FBG Pressure Sensor," IOP Conf. Ser. Earth Environ. Sci., vol. 170, no. 2, 2018, doi: 10.1088/1755-1315/170/2/022049.

[4] 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, doi: 10.1016/j.sigpro.2016.07.001.

[5] 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," Comput. Geotech., vol. 37, no. 4, pp. 440–448, 2010, doi: 10.1016/j.compgeo.2010.01.003.

[6] S. Demirci, E. Yigit, I. H. Eskidemir, and C. Ozdemir, "Ground penetrating radar imaging of water leaks from buried pipes based on back-projection method," NDT E Int., vol. 47, pp. 35–42, 2012, doi: 10.1016/j.ndteint.2011.12.008.

[7] N. He, C. Qian, R. Li, and M. Zhang, "An improved pipeline leak detection and localization method based on compressed sensing and event-triggered particle filter," J. Franklin Inst., vol. 358, no. 15, pp. 8085–8108, 2021, doi: 10.1016/j.jfranklin.2021.08.012.

[8] A. Malekpour and Y. She, "Real-time leak detection in oil pipelines using an Inverse Transient Analysis model," J. Loss Prev. Process Ind., vol. 70, no. February, 2021, doi: 10.1016/j.jlp.2021.104411.

[9] A. Z. Selvaggio, F. M. M. Sousa, F. V. da Silva, and S. S. V. Vianna, "Application of long short-term memory recurrent neural networks for localisation of leak source using 3D computational fluid dynamics," Process Saf. Environ. Prot., vol. 159, pp. 757–767, 2022, doi: 10.1016/j.psep.2022.01.021.

[10] Y. Deng, G. Zhao, K. Zhu, T. Zhou, and Z. Xu, "NCAFI : Nuttall convolution window all-phase FFT interpolation-based harmonic detection," Infrared Phys. Technol., p. 104310, 2022, doi: 10.1016/j.infrared.2022.104310.

[11] N. Behari, M. Z. Sheriff, M. A. Rahman, M. Nounou, I. Hassan, and H. Nounou, "Chronic leak detection for single and multiphase flow: A critical review on onshore and offshore subsea and arctic conditions," J. Nat. Gas Sci. Eng., vol. 81, no. July, p. 103460, 2020, doi: 10.1016/j.jngse.2020.103460.

[12] E. J. Pérez-Pérez, F. R. López-Estrada, G. Valencia-Palomo, L. Torres, V. Puig, and J. D. Mina-Antonio, "Leak diagnosis in pipelines using a combined artificial neural network approach," Control Eng. Pract., vol. 107, no. May 2020, p. 104677, 2021, doi: 10.1016/j.conengprac.2020.104677.

[13] Y. Liu, X. Ma, Y. Li, Y. Tie, Y. Zhang, and J. Gao, "Water pipeline leakage detection based on machine learning and wireless sensor networks," Sensors (Switzerland), vol. 19, no. 23, pp. 1–21, 2019, doi: 10.3390/s19235086.

[14] Y. Song and S. Li, "Gas leak detection in galvanised steel pipe with internal flow noise using convolutional neural network," Process Saf. Environ. Prot., vol. 146, pp. 736–744, 2021, doi: 10.1016/j.psep.2020.11.053.

[15] R. Xiao, Q. Hu, and J. Li, "Leak detection of gas pipelines using acoustic signals based on

wavelet transform and Support Vector Machine," Meas. J. Int. Meas. Confed., vol. 146, pp. 479–489, 2019, doi: 10.1016/j.measurement.2019.06.050.

[16] S. Rashid, U. Akram, and S. A. Khan, "WML: Wireless sensor network based machine learning for leakage detection and size estimation," Procedia Comput. Sci., vol. 63, no. Euspn, pp. 171–176, 2015, doi: 10.1016/j.procs.2015.08.329.

[17] J. G. S. De Tejada and J. S. Martìnez-Echevarrìa, "Support vector machines," Comput. Intell. Eng. Manuf., vol. 1, pp. 147–191, 2008, doi: 10.1007/0-387-37452-3_7.

[18] "Decision Tree Algorithm, Explained," 2022. https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

[19] J. A. Gomez Camperos, F. R. Ubarnes, and E. E. Blanco, "Experimental study for detection of leaks in horizontal pipelines," Contemp. Eng. Sci., vol. 11, no. 101, pp. 5017–5025, 2018, doi: 10.12988/ces.2018.810551.

Authors' information

[1]Department of Mechanical Engineering, Faculty of Engineering, Universidad Francisco de Paula Santander, Ocaña, Colombia.

Gomez Camperos July Andrea received the BSc. Eng. in Mechatronic Engineering from the Universidad de Pamplona, Colombia, in 2007. Msc. in Industrial Controls of the same University. Currently, she is full-time teacher in the Department of Mechanical Engineering of the Faculty of Engineering of the Universidad Francisco de Paula Santander, Ocaña, Colombia from 2018. Also, she is the director of the research group in new technologies, sustainability, and innovation (GINSTI). Her research is based on control systems, energy and new technologies.