Leak diagnosis in pipelines using a combined artificial neural network approach*

E.J. Pérez-Pérez[a], F.R. López-Estrada[a,*], G. Valencia-Palomo[c], L. Torres[b], V. Puig[e], J.D. Mina-Antonio[d]

[a] *Tecnológico Nacional de México / Instituto Tecnológico de Tuxtla Gutiérrez, TURIX-Dynamics Diagnosis and Control Group, Carretera Panam. km 1080,CP 29050, Tuxtla Gutierrez, Chiapas, Mexico.*
[b] *Universidad Nacional Autónoma de México (UNAM) Instituto de Ingeniería, Circuito Exterior, Cd. Universitaria, 04510 Coyoacan, Ciudad de México, Mexico.*
[c] *Tecnológico Nacional de México / Instituto Tecnológico de Hermosillo, Av. Tecnológico y Periférico Poniente, S/N, 83170, Hermosillo, Sonora, Mexico.*
[d] *Tecnológico Nacional de México / CENIDET, Interior Internado Palmira S/N, Col. Palmira, 62490, Cuernavaca, Morelos, Mexico.*
[e] *Universitat Politècnica de Catalunya / Institut de Robòtica i Informàtica Industrial, CSIC-UPC./ Parc Tecnològic de Barcelona. C Llorens i Artigas 4-6, 08028, Barcelona, Spain.*

Abstract

This paper presents a methodology to detect and locate water leaks in pipelines by using artificial neural networks (ANN) techniques and online measurements of pressure and flow rate. Contrary to reported works in the literature, the proposed method estimates the friction factor of the pipe and uses this information as an input to compute the leak position. A combination of experimental and numerical data was used to enrich the data-training set for the ANN. Various leak scenarios were considered to characterize pressure losses and their differentials in different sections of the pipeline. Finally, the algorithm was tested experimentally in a pilot plant, and the results demonstrate good performance and the applicability of the proposed method.

Keywords: Artificial neural network, Water distribution systems, Pipelines leak detection, Pipeline diagnosis.

* This work has been supported by the National (Mexican) Council of Science and Technology (CONACYT) under the program National Problems, grant number PN-2016/3595. The support is gratefully acknowledged.

*Corresponding author
Email address: frlopez@ittg.edu.mx (F.R. López-Estrada)

1. Introduction

Pipeline networks are used to transport different fluids like water, hydrocarbons, oil, natural gas, among others. Like any other liquid distribution system, pipelines are subject to factors that may cause leaks, e.g. corrosion, aging, installation failures, natural events, anthropogenic activities or environmental factors. The application of corrective plus preventive maintenance would allow the reduction of leakage repair and costs assigned to repairs (Carnero & Gómez, 2018). However, other factors such as fluid theft, external blows to the pipeline, hydraulic shocks, among others, cannot be prevented with maintenance, and online monitoring systems are required to detect and locate leaks. Leaks cause damage in the infrastructure, economic losses, environmental contamination, and in some cases (as in hydrocarbons leaks), it represents health risks and may cause human losses (Olivera-Villaseñor & Rodríguez-Castellanos, 2012). Therefore, the studies dedicated to leak detection justify their relevance from a scientific, technological and social perspective.

The aim of an automated leak detection system is to locate, as quickly as possible, the presence and location of leaks with a minimum of instrumentation and cost. The main idea in computer-based methods based on analytical redundancy through the use of models to calculate mass balance, estimate parameters of the pipeline, such as pressure, flow rates, roughness and friction factor, in order to recognize anomalies that indicate a fluid loss (Datta & Sarkar, 2016).

In general, computer-based methods for leak location in pipelines, assume that the quasi-static friction is constant, however, this is not true (Rojas et al., 2018) as this parameter changes with time. An accurate value of the friction factor would improve the performance of leak detection methodologies. In this sense, some authors have proposed leak localization schemes based on the complementary use of quasi-static friction along with other variables or the estimation of parameters; or even explicitly based on the nearly static friction values before and after the occurrence of the leak. For example, in Reddy et al. (2011), the methodology implemented uses an efficient state estimation technique based on a transfer function model, from which the friction factor is calculated recursively. Another proposed solution is to use filters for the estimation of the state, e.g. in Arifin et al. (2015) a particle filter was used, it considers that the friction factor changes with respect to the load loss profile after a leak appears. In addition, an Extended Kalman Filter (EKF) was also used to detect and locate leaks, incorporating differential equations based on the dynamic model of the pipeline. Here, the friction factor is a function of the flow rate and is determined iteratively through the use of the Swamee-Jain equation (Delgado-Aguiñaga et al., 2016; Verde & Rojas, 2017; Santos-Ruiz et al., 2018a; Rojas & Verde, 2020).

An aspect to be highlighted on the existing approaches, that take into account the quasi-static friction, is that they are effective. However, they require the implementation of sophisticated methodologies for its estimation. Moreover, they are often based on highly accurate analytical models and design, e.g. sophisticated observers which are useful only for a limited number of leak scenarios (Datta et al., 2018; Torres et al., 2020). Then, model-based methods require precise adjustment in the estimation and measurement of the model parameters. On the other hand, data-driven methods have proven to present higher accuracy and performance in detecting leaks due to the fact that these methods are based on dynamic behavior instead of mathematical equations. In this sense, data-based approaches as artificial neural networks become attractive.

Artificial neural networks (ANNs) are data-driven techniques used for fault diagnosis. ANNs have recently been used for modeling and prediction of thermophysical properties in nanofluids such as viscosity using experimental data (Hemmat Esfe & Sadati Tilebon, 2018a,b, 2020). Some authors have worked with ANNs for monitoring and diagnose pipes and hydraulic installations, e.g. a time series neural network was presented by Mounce et al. (2002) for leak detection using sensor measurement data to build an empirical model. In Salvatore et al. (2004), a leak detection system was proposed to determine the size and position of the leaks of hazardous materials in approximately 100 seconds. A combination of grouping and classification tools for leak detection was proposed in da Silva et al. (2005) and Santos et al. (2013), where n neuro-fuzzy systems were used in gas pipelines.

More recently, in Leu & Bui (2016), a Bayesian learning algorithm to maximize the accuracy of leak predictions was proposed, this method also indicates the most critical factors that affect water leaks. The authors in Zadkarami et al. (2016) proposed a technique based on a multilayer perceptron neural network classifier (MLPNN) which was improved in Zadkarami et al. (2017) by adding a Dempster-Shafer fusion technique. In both cases, numerical simulations were performed to test the algorithms to the first 20 [km] of the Golkhari pipeline in the south of Iran. A similar MLPNN was presented in Gómez-Camperos et al. (2019), the authors ran experiments in a pipeline for the detection of leaks, however, leak location was not performed. In Jia et al. (2018) advanced self-developed pressure-drop sensors in combination with a backpropagation ANN were developed for leak location. However, this method requires the installation of a minimum number of sensors along the pipe (not only at the ends). In Pulido et al. (2019) a state-space neural network and model-decomposition was presented, it was capable to detect small leaks, even the ones that were undetected by the plant operators; the main drawback is that the proposed method demonstrated to be very sensitive causing false alarms. More complex ANNs approaches can be found in the literature for leak diagnosis using convolutional neural networks in combination with a support vector machine (Kang et al., 2018), spline-local mean decomposition (Zhou et al., 2019), or Bayesian reasoning (Javadiha et al., 2019) but they are intended for pipe networks rather than pipelines.

The methods discussed here are concentrated in Table 1. Detection efficiency presented in terms of higher accuracy and lesser error is calculated with the root mean square, while the efficiency of classification methods are calculated by correct classification rate (CCR).

Table 1: Recent studies on leak diagnosis using neural networks

Reference	Data	System	ANN approach	Efficiency
Leu & Bui (2016)	Pressure	Water network model	Bayesian network learning	84% accuracy
Zadkarami et al. (2016)	Pressure and flow	Oil pipeline (simulated)	Multi-layer perceptron neural network	CCR 92%
Zadkarami et al. (2017)	Pressure and flow	Oil pipeline (simulated)	Multi-layer perceptron neural network classifiers fused by Dempster–Shafer	CCR 95%
Gómez-Camperos et al. (2019)	Flow	Water pipeline (experimental)	Multi-layer perceptron neural network	Leak detection alert
Jia et al. (2018)	Pressure	Water pipeline (experimental)	Backpropagation neural network	1.01% error
Pulido et al. (2019)	Pressure	Steam plant (experimental)	State-space neural networks and model-decomposition	CCR 85.7 %
Kang et al. (2018)	Pressure	Water network (experimental)	Convolutional neural network and a support vector machine	99.3% accuracy
Zhou et al. (2019)	Pressure	Water network (simulated)	Convolutional neural network and spline-local mean decomposition	92.5% accuracy
Javadiha et al. (2019)	Pressure	Water network (simulated)	Convolutional neural network and Bayesian reasoning	87.8% accuracy

Concerning all of these proposals, one can find that most of the implementations are evaluated in numerical simulation scenarios; and when applied to physical pipelines, the evaluation scenarios to test an ANN are limited. One of the drawbacks about ANNs for fault diagnosis purposes is the lack of enough data for its adequate training. For example, in order to train an ANN for leak location in pipelines it is desirable to have enough data for every possible scenario: many leak locations, several input-output pressure conditions, among others. Nevertheless, in a physical pipeline, this may not be possible to obtain due to the impact in damages and costs associated with all those experiments. Then, it is desirable to account with a numerical simulator, which allows the generation of as many as possible scenarios to get the required data for training the ANN used for leak location. Nevertheless, a set of experimental data are also desirable in order to validate the performance of this ANN; without this stage, a non-convincing conclusion can be given with regard to any leak localization scheme based on ANNs. Zaman et al. (2020) highlights that combined methods are more efficient in terms of higher accuracy and lesser error.

This work proposes the use of an ANN for leak localization (ANN-4LL) in pipelines. Given that the ANN-4LL makes use of friction factors, these must be continuously calculated. It is known that the friction factor is defined implicitly employing the well-known Colebrook equation, whose solution is often based on iterative techniques which utilize an essential amount of time and computational burden. But in the context of this work, a highly precise, fast, and straightforward solution method is needed. Then, the proposal made by Brkić & Ćojbašić (2016) was used entirely, where the authors design a feed-forward back-propagation network allowing the estimation of this important parameter quickly and efficiently, avoiding its recalculation. Then, a combination of ANNs is used: two for the friction factor estimation (ANN-4FFE), one for the inlet friction factor estimation and one for outlet; and one ANN for leak location (ANN-4LL). As a result, a robust diagnosis method was obtained capable to detect leaks at the time of appearance and estimate the leak locations with an average percentage error of 0.629 %. The method's performance was evaluated experimentally in a pipeline pilot-plant instrumented with industrial pressure and flow sensors. The main contributions can be summarized as follows:

- An approach for leak location using a combination of ANNs with a cascade-forward back-propagation structure; where the first ANNs allows for estimating the friction factor fast and efficiently, and then, in combination with the second ANN, the leak can be located accurately using measures of pressures and flows at the ends of the pipe.
- The ANN was completely trained with data generated from an exhaustive set of numerical experiments carried out in a previously validated simulator of the pipeline case of study.
- The proposed method was evaluated on the real pipeline pilot plant, using data generated from experiments for different leak locations and operating conditions, outperforming the methods presented in Table 1.

This paper is organized as follows: Section 2 presents the pipeline pilot-plant and its mathematical model; Section 3 describes the methodology for leak detection and location and the design of the ANN for leak location; Section 4 presents the validation of the ANN with experimental data. Finally, Section 5 presents the conclusions.

2. Pipeline pilot-plant

Experimental set-up

The experimental pipeline considered to evaluate the proposed algorithm is located at the Hydroinformatic Laboratory of the Technological Institute of Tuxtla Gutierrez. The pilot-plant and its schematic diagram are shown in Figure 1. In summary, the experimental pipeline is serpentine-shaped, the water is supplied from a tank of 5000 [l], the pressurized flow is driven by a centrifugal pump of 5 [HP] controlled by a Micromaster 420 frequency inverter.

The water flow runs through a 2 inches PVC pipe with an equivalent length of 64.5 [m]. It has four manual gate valves distributed along the pipe to simulate different leak scenarios. These valves are distributed at positions: $z_1 = 0.91$ [m], $z_2 = 12.91$ [m], $z_3 = 26.84$ [m], $z_4 = 45.71$ [m]. At the ends of the pipeline, industrial pressure and flow-rate transmitters have been mounted as shown in Figure 1. These are sensors typically used in industrial water and wastewater monitoring systems and will provide the online measurements for the leak detection and localization algorithm. Flow and pressure measurements are transmitted in current signals to a data acquisition board with USB interface. Details of the instrumentation installed in the pipeline can be found in Table 2.

Figure 1: Experimental pilot-plant (top) and its schematic diagram (bottom).

Table 2: Instrumentation installed in the pipeline

Quantity	Instrument	Description
1	NI 9203	C Series DAQ module with 8 analog current input channels. Input ranges of 0 mA to 20 mA; 16-bit ADC resolution
2	EJA530E	Yokogawa Pressure Transmitter. 4 to 20 mA DC signal; 200 kPa maximum pressure limit. ±0.055% Accuracy. 90 msec response time.
2	GF 2551	Signet Magmeter Flow Transmitter: 4 to 20 mA output. Bi-directional flow, PVDF material. Recommended for water and wastewater monitoring.

Mathematical model and pilot-plant simulator

The ANN-4LL require a rich training dataset to improve their performance and efficiency. Particularly, for the leak detection algorithm, this dataset must contain information about leaks in as many points of the pipe as possible. Nevertheless, in practice, this is not possible. For example, in the experimental pilot-plant presented earlier, only four independent leaks and their combination can be created through the opening of the manual valves. In this case, it is necessary to generate this dataset using a validated model-based simulator, that will complement the experimental dataset by considering different leak scenarios that are not possible to create in the real set-up.

The simulator employed in this work, was designed in OpenModelica by the TURIX Diagnosis and Control Group (Santos-Ruiz et al., 2018b). Internally, the simulator contains the dynamic mathemathical model of the pilot-plant proposed by Chaudhry (1979). This model was obtained from physical principles of momentum and mass conservation (continuity equation), in terms of time t and a spatial variable z defined conveniently in the axial direction of the pipeline. Therefore the mathematical model adopted is:

$$\frac{\partial H(z,t)}{\partial t} + \frac{b^2}{gA_r}\frac{\partial Q(z,t)}{\partial z} = 0, \tag{1}$$

$$\frac{1}{A_r}\frac{\partial Q(z,t)}{\partial t} + g\frac{\partial H(z,t)}{\partial z} - \frac{f(Q(z,t))|Q(z,t)|}{2DA_r^2} = 0. \tag{2}$$

where H refers to the pressure load [mwc], Q indicates the flow $[m^3/s]$, b is the speed of the pressure wave $[m/s]$, g is the gravitational acceleration constant $[m/s^2]$, A_r represents the cross sectional area of the pipeline $[m^2]$ and D is the internal diameter of the pipeline $[m]$. The dynamic model was validated in Santos-Ruiz et al. (2018a).

In order to compute the flow rates and pressure values at the ends of the pipeline, the numerical simulator, requires the physical parameters of the pipe as given in Table 3, the running time, the leak position (z in meters) and the activation function for the leak. The operation of the numerical simulator is depicted in Figure 2, where the obtained results are stored in a CSV (comma-separated values) file for logging purposes.

Table 3: Pipeline pilot-plant parameters.

Parameter	Description	Value	Units
L	Equivalent length	64.48	$[m]$
D	Internal diameter of the pipe	0.0486	$[m]$
A_r	Cross section area of the pipe	0.0023	$[m^2]$
ε	Relative roughness	4.84×10^{-4}	–
v	Kinematic viscosity	8.03×10^{-7}	$[m^2/s]$
b	Wave speed	422.754	$[m/s]$
g	Gravitational acceleration	9.782757	$[m/s^2]$

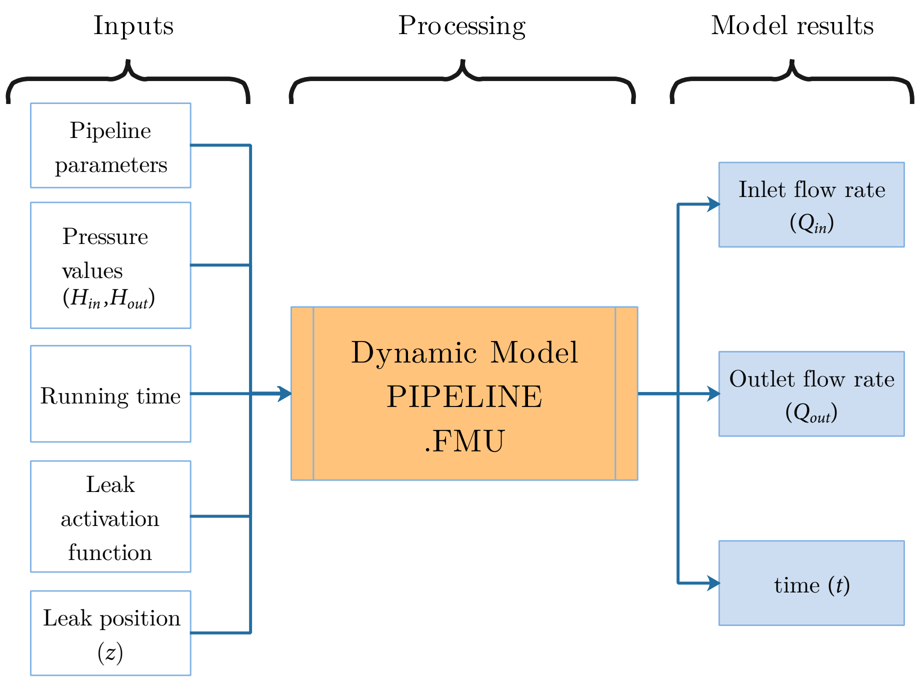

Figure 2: Inputs and outputs of the numerical simulator.

3. Leak detection and localization

This section provides a detailed design procedure for localizing leaks, which can be used for the diagnosis of pipelines. The proposed procedure is depicted in Figure 3, and it comprises different stages. First, measurements of input and output pressures (H_{in}, H_{out}) and flow rates (Q_{in}, Q_{out}) are taken from the pipeline test bench. Then, the inlet (Re_{in}) and outlet (Re_{out}) Reynolds numbers are computed using Q_{in}, Q_{out}. Note that, the Reynolds numbers have a higher magnitude than the relative roughness, which is also needed in the ANN-4FFE blocks. Therefore, logarithmic values of the Reynolds numbers and the relative roughness are considered in order to normalize them. Subsequently, two independent ANN-4FEE blocks compute the friction factors for the input (f_{in}) and output (f_{out}), respectively. Finally, using $f_{in}, f_{out}, H_{in}, H_{out}$, the ANN-4ALL block, yields the leak location (z) estimate. In the following sections, a detailed description of each one of these stages will be provided.

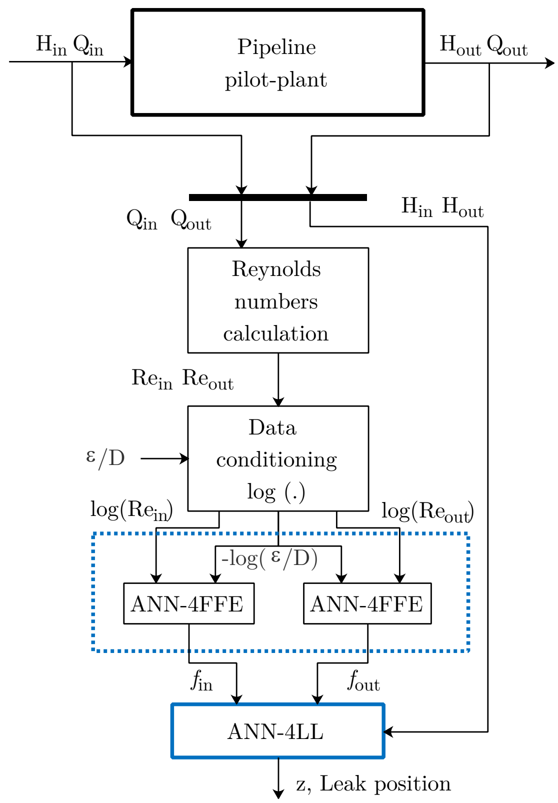

Figure 3: General methodology schema.

3.1. Estimation of the friction factor

In a leakage scenario, the flow rate changes drastically. This causes that the Reynolds number upstream and downstream the leak are different. As a consequence, the quasi-static friction is also different before and after the leak (Verde et al., 2014). In that sense, it is important to have a mechanism for calculating these friction factors. The common approach in the hydraulic community is to calculate the quasi-static friction in turbulent regime using the Colebrook equation (Colebrook & White, 1937), defined as:

$$\frac{1}{\sqrt{f}} = -2 \cdot log_{10}\left(\frac{2.51}{Re \cdot \sqrt{f}} + \frac{\varepsilon}{3.7 \cdot D}\right), \tag{3}$$

where Re is the Reynolds number, ε/D is the relative roughness and f is the friction factor. Note that the Colebrook equation has to be solved implicitly, i.e. in order to find a solution for f, it is necessary to employ an iterative numerical method e.g. Newton-Raphson algorithm (Yildirim, 2009). However, the use of a numerical method increases the computational complexity of the leak diagnosis algorithm. To avoid the computational complexity of the numerical methods, different explicit approximations of the Colebrook equation have been also proposed in the literature. For example, to mention some of the most important we have: the Moody equation (Haaland, 1983), the Churchill approximation (Churchill, 1973), the Swamee-Jain equation (Swanee & Jain, 1976), the Serghides equation (Serghides, 1984) and the Brkic equation (Brkić, 2011). Nevertheless, they present an error in the calculation of the friction factor because these equations are approximations. For instance, in Brkić (2011), a comparison of the numerical errors of several methods was made and it was shown that the approximation with the minimum error was of 0.14% and the one with the highest relative error was close to 8.0%. Recently, the use of artificial neural networks (ANN) has been proposed as an alternative to reduce this approximation error. Particularly, the authors in Brkić & Ćojbašić (2016) proposed an ANN architecture that computes the friction with a maximum error of 0.07%.

The leak localization procedure proposed here requires an estimation mechanism for f_{in} and f_{out} that should be fast, simple, and with high precision. Then, the artificial neuronal network (ANN-4FFE) used to estimate the friction factor is a feed-forward back-propagation network whose architecture to compute the friction was proposed in Brkić & Ćojbašić (2016) and its scheme is displayed in Figure 4.

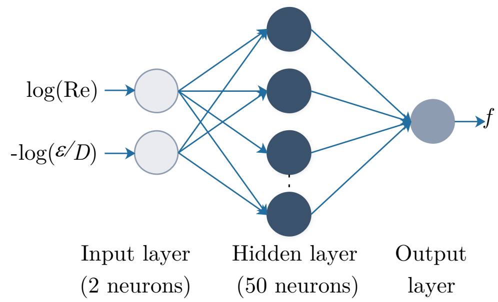

Figure 4: Structure of the proposed ANN-4FFE (Brkić & Ćojbašić, 2016).

3.2. Proposed ANN for Leak Localization (ANN-4LL)

The proposed architecture of the ANN-4LL to estimate the leak position is a cascade forward backpropagation network (CFBPN) (Lashkarbolooki et al., 2013a). During training, calculations are carried out from the input to the output layer, and the error values returned to the previous layer. This network uses continuously valued functions and supervised learning (Jain et al., 1996). Thus, supervised learning proceeds as a closed-loop feedback system where error is the feedback signal. The CFBPN model includes a weight connection from the input layer to each hidden layer and from each hidden layer to the successive layers as shown in Figure 5. By including more layers, the CFBPN may learn multifarious relationships and additional connections can improve the speed at which the network learns the desired co-relationship (Lashkarbolooki et al., 2013b). The main particularity of the CFBPN is that each layer of neurons is related to all previous layers of neurons (Hedayat et al., 2009). A connection multiplies its value by a connection weight, as a synaptic connection does. Each input x_i is weighted by a factor w_i. The neuron has a bias (b) and the overall sum of the inputs is calculated as $\sum w_i x_i + b = net_j$. Then, an activation function (φ) is applied to the result net_j and the neural output is taken as $\varphi(net_j)$.

The ANN-4LL used was fully trained and validated in an off-line supervised manner prior to the application of the network. There are no strict rules for selecting the transfer function. The sigmoid (logsig) and hyperbolic tangent (tansig) functions are the most used in the hidden layer (Hemmat Esfe & Afrand, 2020). In this work, the best performance was obtained using tansig, while the linear function (purelin) was used for the output layer (Samanta, 2004). The complete structure can be seen in Figure 5.

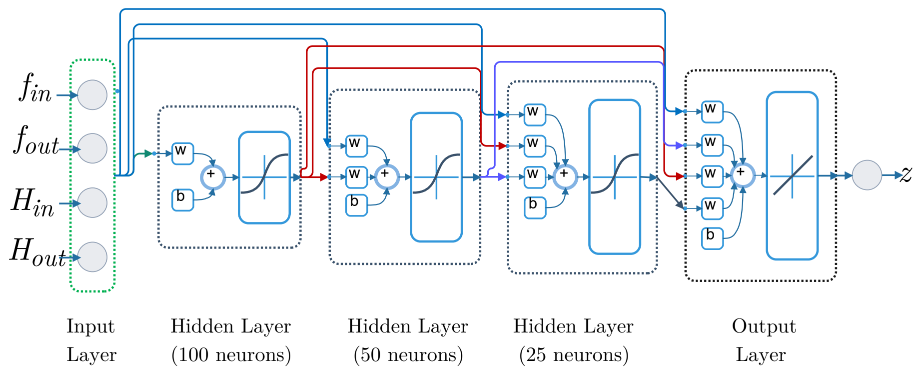

Figure 5: Schematics of the proposed cascade forward backpropagation neural network (ANN-4LL).

The architecture for ANN-4LL (layers, neurons, activation functions) was selected heuristically on a trial and error basis by testing different configurations. Regarding the selection of number of layers and their sizes, it is well known that there are not given rules. Then, for every specific problem or application, the optimal number of hidden layers and neurons is determined through experimentation. In the case of the ANN-4LL, it consists of five layers. The input layer has four neurons considering the friction factors (f_{in}, f_{out}) and pressures (H_{in}, H_{out}), the next three hidden layers contain, one hundred, fifty and twenty-five neurons, respectively. Finally, a neuron in the output layer is considered the target value of the network, i.e. the leak position z.

3.3. Preparation of the dataset for ANN-4LL training

For ANN-4LL, two datasets were considered: (i) a numerical dataset generated by the simulator, exclusively for training purposes, and (ii) a dataset generated from experiments, exclusively for the final evaluation. For both data **R1-5** sets, the input data for the neural network was obtained from flow and pressure sensors installed at the ends of the pipeline as these measures are the most used for leak detection and location (Wong et al., 2018; Van der Walt et al., 2018; Raei et al., 2019). Furthermore, to guarantee the data quality of the experimental set-up, the location of the sensors was done carefully according to the hydraulic handbook (Livelli, 2010). To capture the dynamic effect of the leaks, these were induced by the opening four leak-valves located along the pipeline. Nonetheless, as explained earlier, the number of leaks that can be induced experimentally is limited only to these four valves. Then, in order to enrich the data-set, an experimental-validated pipeline simulator designed in OpenModelica (Santos-Ruiz et al., 2018b) was used. Furthermore, the data generated by the simulator considers noise as in the real pipeline, which is important to guarantee the quality of the data-set. Different operating conditions are considered, so that the neural network learns to locate leaks when they appear.

3.3.1. Numerical dataset

The numerical experiments were carried-out by considering different pressures at the ends of the pipeline. The input pressure (H_{in}) was changed from 21 [mwc] to 2 [mwc] and the output pressure (H_{out}) from 20 [mwc] to 1 [mwc] with respect to each leak. A leak was simulated each 0.5 [m] along the 64 [m] of the pipeline. As a result, 26,040 leaks were generated. The procedure to simulate leaks by considering the pipeline simulator is shown in Figure 6.

An example of a real leak and a simulated leak is shown in Figure 7. It is important to note that simulated leaks contain noise, which was characterized according to the real measurements given by the physical sensors with respect to the real experiments. As expected, at the time the leak appears (at $t = 150$ [s]), the flow rate changes drastically. This information, together with the pressure drops, is collected in the dataset and is used for the ANN-4LL training.

Finally, the Reynolds number and its logarithm ($-log(\varepsilon/D)$) were computed for the 26,040 values of pressure and flow drops that are contained in the dataset as shown in Figure 8. These values will be used to compute the friction factor.

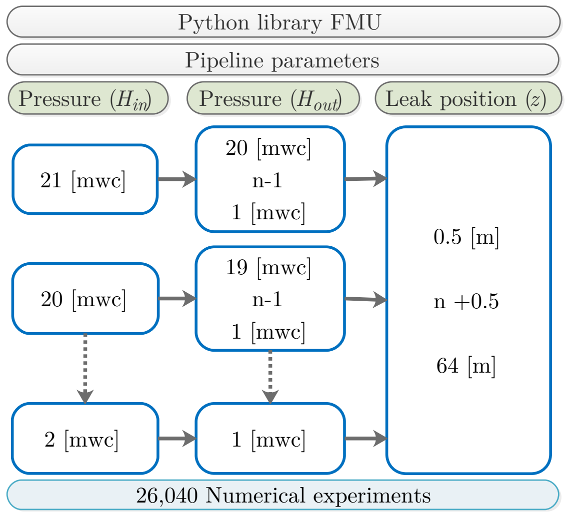

Figure 6: Diagram of the procedure to carry out the numerical simulations

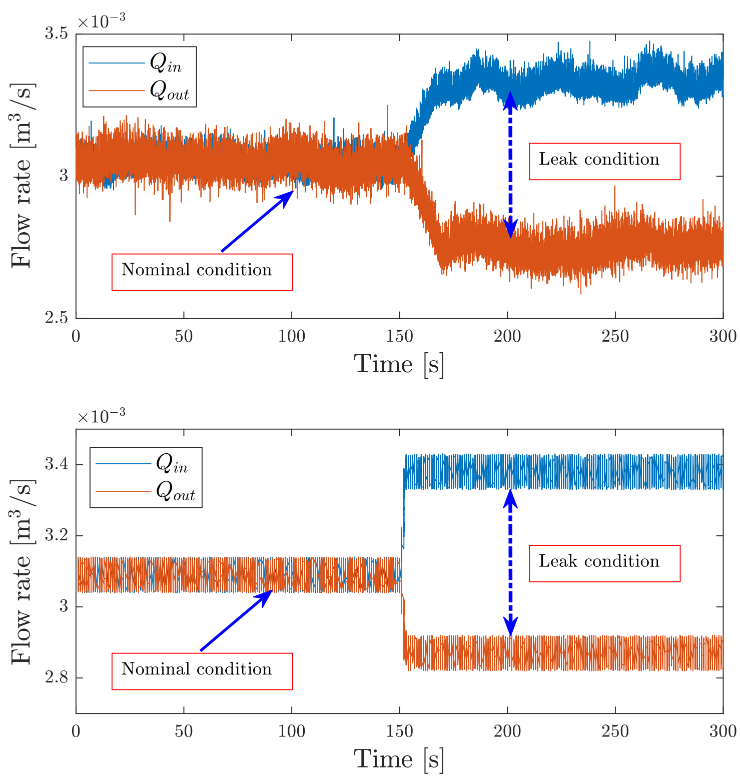

Figure 7: Inlet (blue) and outlet (red) flow rate when a leak occurrence. a) experimental data; b) numerical simulation

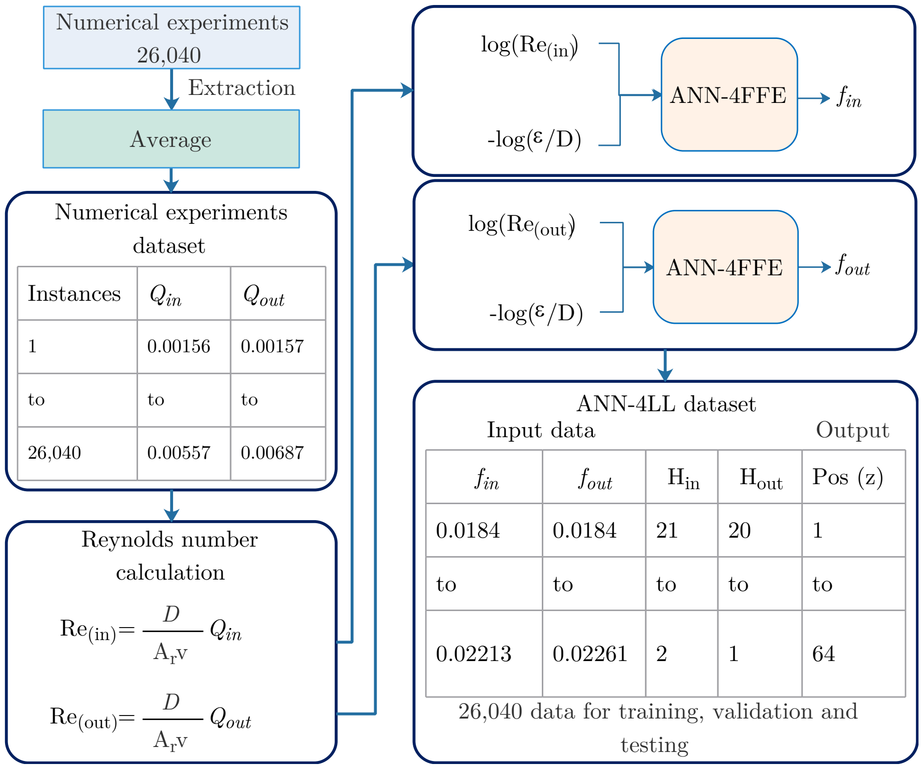

Figure 8: Diagram of the process to obtain the dataset

3.3.2. Training of ANN-4LL

The network is trained with the backpropagation algorithm Scaled Conjugate Gradient (Møller, 1993) that updates the values of weight and bias, being reliable and effective in the training of complex datasets (Offor & Alabi, 2016; Meireles et al., 2003). The numerical dataset was stored in a CVS file and it was divided in three subsets:

- A subset of 18,228 data quintuples (70 %) designed to train the ANN-4LL through an iterative procedure that updates the value of synaptic weights and minimizes an error function (Charalambous, 1992).

- A subset of 3,906 data quintuples (15 %) used in a procedure called cross-validation (Zhang et al., 1999) to prevent the occurrence of overfitting

(Lawrence & Giles, 2000).

- A subset of 3,906 quintuples (15 %) for testing purposes, it has no effect on training and therefore provides an independent measure of ANN-4LL performance during and after training.

The learning of the ANN-4LL stopped when training was stopped, i.e. when the performance on the validation set dropped in relation to that of the training set (Jack & Nandi, 2002). The mean squared error (MSE) was used as a measure of performance for the training phase, i.e. the square of the difference between the network output and the data output is the main criterion for estimating the learning level in supervised mode (Hemmat Esfe & Sadati Tilebon, 2020). The MSE is computed by

$$MSE = \frac{1}{Y}\sum_{k=1}^{Y} e_k^2 = \frac{1}{Y}\sum_{k=1}^{Y} (t_k - y_k)^2 \tag{4}$$

where Y indicates the number of samples in the training dataset, e_k is the neural network error, t_k are the target values, and y_k are network output values. The MSE is determined after each epoch and the learning process is finished when MSE is minimized (Beale et al., 1992). The validation process takes place during training to determine whether to continue iterating in the search for a better model. The learning process was repeated until the mean squared error between the actual output and the desired output was 10^{-4} or less. In this process, different ANN architectures were evaluated, opting for the one with the best performance according to the MSE. The test process is post-training, it is used to measure the prediction capacity of the model obtained. This last process was carried out exclusively with experimental data from the pilot pipeline.

3.4. Training result

The ANN-4LL was developed in Matlab 2017b, however, given the complexity of the ANN-4LL model and the dataset size, it was decided to use GPU with the CUDA platform developed by NVIDIA (Lindholm et al., 2008), which allows the parallel processing of the numerical calculations. The used hardware was a NVIDIA GeForce GTX-750Ti 2048 MB GDDR5 card with 640 CUDA cores. Table 4 describes the attributes of ANN-4LL training.

Table 4: Attributes of ANN-4LL

Attribute	Description
Dataset	26,040
Network type	CFBPN
Input variables	$f_{in}, f_{out}, H_{in}, H_{out}$
Output variable	leak position (z)
Training algorithm	Gradient Conjugate Scaled
Act. function (hidden layers)	tansig
Act. function (output layer)	purelin
Performance function	MSE (default)
Iterations	13,340

The performance of training, testing and validation of ANN-4LL, based on their corresponding MSE is shown in Figure 9. The set destined for validation is used to measure the degree of generalization of the network, stopping the training when it no longer improves. This prevents overfitting that implies a poor performance of the model to predict new values.

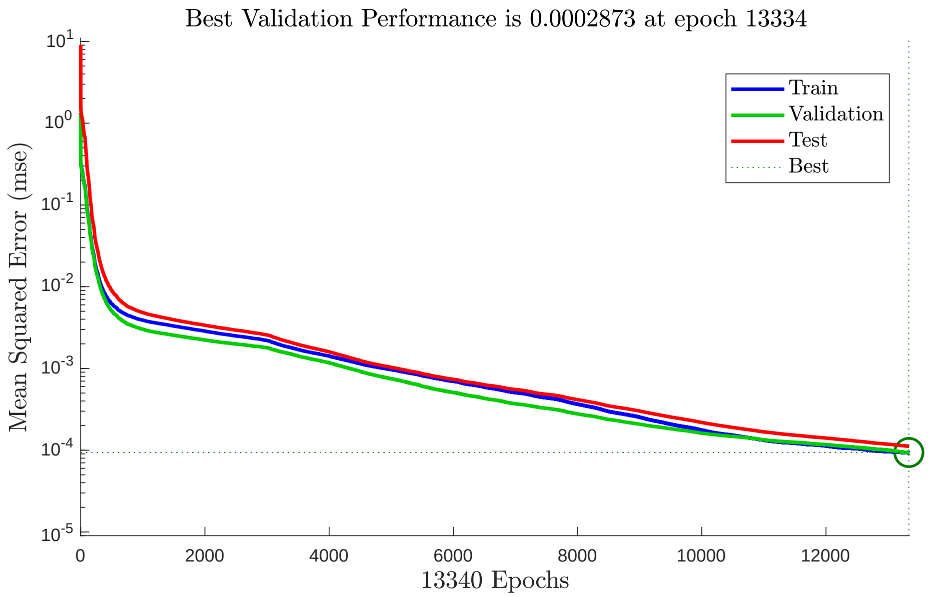

Figure 9: The mean squared error (MSE) during the process of training of the ANN-4LL

The training of the ANN-4LL was made up to 13,340 epochs. The MSE of ANN-4LL was calculated at 10^{-4} (close to zero), after which there was no further tendency to decrease as it can be seen in Figure 9. This means that ANN-4LL with the proposed structure has a great precision capacity for generalization. As mentioned earlier, the architecture selection was made empirically, by testing a different number of layers, neurons, and activation functions. Other configurations involving more than 100 neurons in one hidden layer and two layers with 50 neurons in each of them, achieved similar accuracy results. In contrast, ANN-4LL structures tested with fewer than 100 neurons in a hidden layer resulted in lower precision compared to the proposed structure, even after 15,000 training epochs.

4. Experimental results

Experimental dataset

Similar to the training stage of the ANN-4LL, in the experimental validation, quintuples of values of Q_{in}, Q_{out}, H_{in}, H_{out}, z are generated. The quintuples of experimental values were obtained from the pilot-plant in experiments for each leak position (4 possible positions), which were repeated 4 times, i.e. 16 experiments were generated. It is emphasized that the running time of each experiment was 360 [s] (180 [s] before and 180 [s] after the leak occurrence), with a sample frequency of 1kHz. The nominal condition of the pressure before the leak occurrence was $H_{in} = 5.7087$ [mwc] and $H_{out} = 1.9998$ [mwc]; and for the flow, it was $Q_{in} = 3.04^{-3}[m^3/s]$ and $Q_{out} = 3.04^{-3}[m^3/s]$.

4.1. Validation of ANN-4LL with experimental data

The leaks are located in the following positions: the first leak is at $z_1 = 0.91$ [m], the second at $z_2 = 12.91$ [m], the third at $z_3 = 26.84$ [m] and the fourth leak is located at $z_4 = 45.71$ [m]. So, the main challenge of ANN-4LL is to locate the leak with the minimum position error. The results are shown in Tables 5 to 8. Table 5 shows the estimation of first leak position (z_1), where the average percentage error between the four test was of 2.198%. The estimate was good, considering that the leak was very close to the pump, so it generates a lot of turbulence. The leak location (z_1) with the ANN-4LL, presents the maximum approximation error with respect to the other experiments.

The results of the evaluation of ANN-4LL with the second leak (z_2) can be seen in Table 6. An improvement in the location error can be appreciated, given that average percentage error between the four tests was of 0.155%, which is much better than estimation obtained for z_1.

The results of the evaluation of ANN-4LL with the third leak (z_3) can be seen in Table 7, where the resulting average percentage error between the four tests was of 0.075%. The performance achieved in this test was the best because this leak is located in a zone of the pipeline where the turbulence is lower, i.e. far from the pump.

Table 5: Validation of the ANN-4LL with experimental data for leak in z_1

Test	f_{in}	f_{out}	Estimated position [m]	Percentage error
1	0.0200052	0.0202999	0.9310	2.308 %
2	0.0200053	0.0203000	0.9304	2.242 %
3	0.0200067	0.0202997	0.9289	2.077 %
4	0.0200048	0.0202996	0.9297	2.165 %
avg	0.0200055	0.0202995	0.93	2.198 %

Table 6: Validation of the ANN-4LL with experimental data for leak in z_2

Test	f_{in}	f_{out}	Estimated position [m]	Percentage error
1	0.0200755	0.0203425	12.8952	0.115 %
2	0.0200753	0.0203486	12.8935	0.128 %
3	0.0200739	0.0203501	12.9011	0.069 %
4	0.0200689	0.0203644	12.8702	0.308 %
avg	0.0200734	0.0203514	12.89	0.155 %

Table 7: Validation of the ANN-4LL with experimental data forleak in z_3

Test	f_{in}	f_{out}	Estimated position [m]	Percentage error
1	0.0201468	0.0204002	26.8198	0.075 %
2	0.0201451	0.0204118	26.8206	0.072 %
3	0.0201478	0.0204017	26.8213	0.070 %
4	0.0201311	0.0204122	26.8183	0.081 %
avg	0.0201427	0.0204065	26.82	0.075 %

Table 8: Validation of the ANN-4LL with experimental data for leak in z_4

Test	f_{in}	f_{out}	Estimated position [m]	Percentage error
1	0.0202211	0.0205099	45.7461	0.079 %
2	0.0202165	0.0205161	45.7578	0.105 %
3	0.0202155	0.0204899	45.7615	0.113 %
4	0.0202309	0.0205121	45.7347	0.054 %
avg	0.0202210	0.0205070	45.75	0.088 %

Finally, in Table 8 the results of the ANN-4LL with the fourth leak (z_4) are shown, where the resulting average percentage error between the four tests was of 0.088%. Note here that the performance error increases with respect to the leak position z_3. Nevertheless, this estimation is good, considering that z_4 is in the final part of the pipeline and the outlet flow rate is lower.

According to the results shown in previous tables for the experimental tests, it is verified that ANN-4LL performs very well for leak location in the pipeline case of study. In this regard, the global average percentage error between the four leaks is of 0.629%. It is clear that the proposed method has a higher precision than most of those presented in Table 1, achieving a 99.3% of effectiveness (0.629% error), similar to the one presented in Kang et al. (2018) but with a far simpler architecture. The results are good taking into account that experimental data was taken from noisy measurements; much more important is the fact that ANN-4LL was trained with data from the numerical simulator of the pilot plant. This means that such results also validate the good performance of the simulator and implicitly, the good architecture selection for ANN-4LL and its corresponding training. As a final comment, the ANN architecture presented in this paper can be applied to similar pipes provided that measures of pressure and flow at the ends of the pipe are available and the ANN has been adequately trained.

5. Conclusions

In this work, encouraging results were obtained, giving a new perspective of leak location in pipelines when using ANNs. Parameters in deterministic mathematical models are considered ideal, however, in real applications, there is parametric uncertainty and some of the parameters vary with time as in the case of the friction factor. Moreover, data obtained via experimental measurements contains noise. An important contribution of this work was the design of a two-stage neural network that detects and estimates the position of a leak in a pipe, where the first stage calculates the friction factor using inlet and outlet flows. Then, in a second stage, it uses this information together with the inlet and outlet pressures to locate the leak position. The training of the proposed neural network was performed using data generated from a validated numerical simulator of the pilot-plant. Finally, the proposed approach was validated experimentally in a pilot-plant with leaks in the near, mid, and far range with respect to the sensors. The designed neural network results in a robust predictor for leak localization with an average percentage error of 0.629%. It is important to mention that the applicability of the proposed method could be limited due to the requirement of pressurized flow. Nonetheless, this is a common assumption in the leak diagnosis literature (Zhou et al., 2018; Li et al., 2019; Zaman et al., 2020). Another important point is that industrial pipes could be considerably longer, and sensors may be separated by a great distance; in this case, communication problems would arise, e.g., long delays, data loss, data corruption, data packages discarded or dropped out, among others. These communication issues can be addressed by considering networked control algorithms. However, it must be taken into account that the diagnosis and location of leaks is a monitoring process and that in case of being affected by these communication problems, they do not represent a safety-critical issue. As future work, the use of ANN will be extended to detect multiple leaks and to estimate frictions factors in hydraulic networks (Bermúdez et al., 2018). Also, future work will be done to consider optimal criteria for the selection of the best architecture (number of layers, neurons, etc.) for the ANN.

References

Arifin, B., Li, Z., & Shah, S. L. (2015). Pipeline leak detection using particle filters. *IFAC-PapersOnLine*, *48*, 76–81.

Beale, M. H., Hagan, M. T., & Demuth, H. B. (1992). Neural network toolbox™ user's guide. *The Mathworks Inc*, .

Bermúdez, J.-R., López-Estrada, F.-R., Besançon, G., Valencia-Palomo, G., Torres, L., & Hernández, H.-R. (2018). Modeling and simulation of a hydraulic network for leak diagnosis. *Mathematical and Computational Applications*, *23*, 70.

Brkić, D. (2011). Review of explicit approximations to the colebrook relation for flow friction. *Journal of Petroleum Science and Engineering*, *77*, 34–48.

Brkić, D., & Ćojbašić, Ž. (2016). Intelligent flow friction estimation. *Computational Intelligence and Neuroscience*, *2016*.

Carnero, M. C., & Gómez, A. (2018). Optimization of maintenance in production and storage systems for domestic water. *Water resources management*, *32*, 359–380.

Charalambous, C. (1992). Conjugate gradient algorithm for efficient training of artificial neural networks. *IEE Proceedings G (Circuits, Devices and Systems)*, *139*, 301–310.

Chaudhry, M. H. (1979). *Applied hydraulic transients*. Springer.

Churchill, S. W. (1973). Empirical expressions for the shear stress in turbulent flow in commercial pipe. *AIChE Journal*, *19*, 375–376.

Colebrook, C., & White, C. (1937). Experiments with fluid friction in roughened pipes. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, *161*, 367–381.

Datta, S., Gautam, N. K., & Sarkar, S. (2018). Pipe network blockage detection by frequency response and genetic algorithm technique. *Journal of Water Supply: Research and Technology-Aqua*, *67*, 543–555.

Datta, S., & Sarkar, S. (2016). A review on different pipeline fault detection methods. *Journal of Loss Prevention in the Process Industries*, *41*, 97 – 106.

Delgado-Aguiñaga, J., Besancon, G., Begovich, O., & Carvajal, J. (2016). Multi-leak diagnosis in pipelines based on extended kalman filter. *Control Engineering Practice*, *49*, 139–148.

Gómez-Camperos, J. A., Espinel-Blanco, E. E., & Regino-Ubarnes, F. J. (2019). Diagnosis of horizontal pipe leaks using neural networks. *Journal of Physics: Conference Series*, *1388*, 012032.

Haaland, S. E. (1983). Simple and explicit formulas for the friction factor in turbulent pipe flow. *Journal of Fluids Engineering*, *105*, 89–90.

Hedayat, A., Davilu, H., Barfrosh, A. A., & Sepanloo, K. (2009). Estimation of research reactor core parameters using cascade feed forward artificial neural networks. *Progress in Nuclear Energy*, *51*, 709 – 718.

Hemmat Esfe, M., & Afrand, M. (2020). Mathematical and artificial brain structure-based modeling of heat conductivity of water based nanofluid enriched by double wall carbon nanotubes. *Physica A: Statistical Mechanics and its Applications*, *540*, 120766.

Hemmat Esfe, M., & Sadati Tilebon, S. M. (2018a). Modeling and prediction of rheological behavior of al2o3-mwcnt/5w50 hybrid nano-lubricant by artificial neural network using experimental data. *Physica A: Statistical Mechanics and its Applications*, *510*, 625 – 634.

Hemmat Esfe, M., & Sadati Tilebon, S. M. (2018b). Using artificial neural network for investigating of concurrent effects of multi-walled carbon nanotubes and alumina nanoparticles on the viscosity of 10w-40 engine oil. *Physica A: Statistical Mechanics and its Applications*, *510*, 610 – 624.

Hemmat Esfe, M., & Sadati Tilebon, S. M. (2020). Statistical and artificial based optimization on thermo-physical properties of an oil based hybrid nanofluid using nsga-ii and rsm. *Physica A: Statistical Mechanics and its Applications*, *537*, 122126.

Jack, L., & Nandi, A. (2002). Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms. *Mechanical systems and signal processing*, *16*, 373–390.

Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: a tutorial. *Computer*, *29*, 31–44.

Javadiha, M., Blesa, J., Soldevila, A., & Puig, V. (2019). Leak localization in water distribution networks using deep learning. In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 1426–1431).

Jia, Z., Ren, L., Li, H., & Sun, W. (2018). Pipeline leak localization based on fbg hoop strain sensors combined with bp neural network. *Applied Sciences*, *8*.

Kang, J., Park, Y.-J., Lee, J., Wang, S.-H., & Eom, D.-S. (2018). Novel leakage detection by ensemble CNN-SVM and graph-based localization in water distribution systems. *IEEE Transactions on Industrial Electronics*, *65*, 4279–4289.

Lashkarbolooki, M., Shafipour, Z. S., & Hezave, A. Z. (2013a). Trainable cascade-forward back-propagation network modeling of spearmint oil extraction in a packed bed using SC-CO2. *The Journal of Supercritical Fluids*, *73*, 108 – 115.

Lashkarbolooki, M., Vaferi, B., Shariati, A., & Hezave, A. Z. (2013b). Investigating vapor–liquid equilibria of binary mixtures containing supercritical or near-critical carbon dioxide and a cyclic compound using cascade neural network. *Fluid Phase Equilibria*, *343*, 24 – 29.

Lawrence, S., & Giles, C. L. (2000). Overfitting and neural networks: conjugate gradient and backpropagation. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium* (pp. 114–119 vol.1). volume 1.

Leu, S.-S., & Bui, Q.-N. (2016). Leak prediction model for water distribution networks created using a bayesian network learning approach. *Water Resources Management*, *30*, 2719–2733.

Li, J., Zheng, Q., Qian, Z., & Yang, X. (2019). A novel location algorithm for pipeline leakage based on the attenuation of negative pressure wave. *Process Safety and Environmental Protection*, *123*, 309–316.

Lindholm, E., Nickolls, J., Oberman, S., & Montrym, J. (2008). Nvidia tesla: A unified graphics and computing architecture. *IEEE micro*, *28*, 39–55.

Livelli, G. (2010). Flowmeter piping requirements. *Flow Control Network*, *26*.

Meireles, M. R., Almeida, P. E., & Simões, M. G. (2003). A comprehensive review for industrial applicability of artificial neural networks. *IEEE Transactions on Industrial Electronics*, *50*, 585–601.

Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, *6*, 525 – 533.

Mounce, S., Day, A., Wood, A., Khan, A., Widdop, P., & Machell, J. (2002). A neural network approach to burst detection. *Water Science and Technology*, *45*, 237–246.

Offor, U. H., & Alabi, S. B. (2016). Artificial neural network model for friction factor prediction. *Journal of Materials Science and Chemical Engineering*, *4*, 77.

Olivera-Villaseñor, R. E., & Rodríguez-Castellanos, A. (2012). Estudio del riesgo en ductos de transporte de gasolinas y diesel en México. *Científica*, *16*.

Pulido, B., Zamarreño, J. M., Merino, A., & Bregon, A. (2019). State space neural networks and model-decomposition methods for fault diagnosis of complex industrial systems. *Engineering Applications of Artificial Intelligence*, *79*, 67 – 86.

Raei, E., Shafiee, M. E., Nikoo, M. R., & Berglund, E. (2019). Placing an ensemble of pressure sensors for leak detection in water distribution networks under measurement uncertainty. *Journal of Hydroinformatics*, *21*, 223–239.

Reddy, H. P., Narasimhan, S., Bhallamudi, S. M., & Bairagi, S. (2011). Leak detection in gas pipeline networks using an efficient state estimator. part-i: Theory and simulations. *Computers & chemical engineering*, *35*, 651–661.

Rojas, J., & Verde, C. (2020). Adaptive estimation of the hydraulic gradient for the location of multiple leaks in pipelines. *Control Engineering Practice*, *95*, 104226.

Rojas, J., Verde, C., Torres, L., & Pérez, E. (2018). On-line head loss identification for monitoring of pipelines. *IFAC-PapersOnLine*, *51*, 748–754.

Salvatore, B., Paolo, L., Paolo, A., & Sanjoy, B. (2004). Leak detection in liquefied gas pipelines by artificial neural networks. *AIChE Journal*, *44*, 2675–2688.

Samanta, B. (2004). Gear fault detection using artificial neural networks and support vector machines with genetic algorithms. *Mechanical systems and signal processing*, *18*, 625–644.

Santos, R. B., Rupp, M., Bonzi, S., Fileti, A. et al. (2013). Comparison between multilayer feedforward neural networks and a radial basis function network to detect and locate leaks in pipelines transporting gas. *Chem. Eng. Trans*, *32*, e1380.

Santos-Ruiz, I., Bermudez, J., Lopez-Estrada, F., Puig, V., Torres, L., & Delgado-Aguinaga, J. (2018a). Online leak diagnosis in pipelines using an

EKF-based and steady-state mixed approach. *Control Engineering Practice*, *81*, 55–64.

Santos-Ruiz, I., Lopez-Estrada, F.-R., & Puig, V. (2018b). Pipeline leaks simulator. URL: `https://doi.org/10.5281/zenodo.2573811`.

Serghides, T. (1984). Estimate friction factor accurately. *Chemical Engineering*, *91*, 63–64.

da Silva, H. V., Morooka, C. K., Guilherme, I. R., da Fonseca, T. C., & Mendes, J. R. (2005). Leak detection in petroleum pipelines using a fuzzy system. *Journal of Petroleum Science and Engineering*, *49*, 223 – 238.

Swanee, P., & Jain, A. K. (1976). Explicit equations for pipeflow problems. *Journal of the Hydraulics Division*, *102*.

Torres, L., Jiménez-Cabas, J., González, O., Molina, L., & López-Estrada, F.-R. (2020). Kalman filters for leak diagnosis in pipelines: brief history and future research. *Journal of Marine Science and Engineering*, *8*, 173.

Verde, C., Molina, L., & Torres, L. (2014). Parameterized transient model of a pipeline for multiple leaks location. *Journal of Loss Prevention in the Process Industries*, *29*, 177 – 185.

Verde, C., & Rojas, J. (2017). Recursive scheme for sequential leaks' identification. In *Modeling and Monitoring of Pipelines and Networks* (pp. 125–145). Springer.

Van der Walt, J., Heyns, P. S., & Wilke, D. N. (2018). Pipe network leak detection: comparison between statistical and machine learning techniques. *Urban Water Journal*, *15*, 953–960.

Wong, L., Deo, R., Rathnayaka, S., Shannon, B., Zhang, C., Chiu, W. K., Kodikara, J., & Widyastuti, H. (2018). Leak detection in water pipes using submersible optical optic-based pressure sensor. *Sensors*, *18*, 4192.

Yildirim, G. (2009). Computer-based analysis of explicit approximations to the implicit colebrook–white equation in turbulent flow friction factor calculation. *Advances in Engineering Software*, *40*, 1183 – 1190.

Zadkarami, M., Shahbazian, M., & Salahshoor, K. (2016). Pipeline leakage detection and isolation: An integrated approach of statistical and wavelet feature extraction with multi-layer perceptron neural network (MLPNN). *Journal of Loss Prevention in the Process Industries*, *43*, 479–487.

Zadkarami, M., Shahbazian, M., & Salahshoor, K. (2017). Pipeline leak diagnosis based on wavelet and statistical features using dempster–shafer classifier fusion technique. *Process Safety and Environmental Protection*, *105*, 156 – 163.

Zaman, D., Tiwari, M. K., Gupta, A. K., & Sen, D. (2020). A review of leakage detection strategies for pressurised pipeline in steady-state. *Engineering Failure Analysis*, *109*, 104264.

Zhang, G., Hu, M. Y., Patuwo, B. E., & Indro, D. C. (1999). Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis. *European Journal of Operational Research*, *116*, 16 – 32.

Zhou, M., Pan, Z., Liu, Y., Zhang, Q., Cai, Y., & Pan, H. (2019). Leak detection and location based on islmd and cnn in a pipeline. *IEEE Access*, *7*, 30457–30464.

Zhou, S., O'Neill, Z., & O'Neill, C. (2018). A review of leakage detection methods for district heating networks. *Applied Thermal Engineering*, *137*, 567–574.