International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I11P128

Volume 72 Issue 11, 279-293, November 2024

Original Article

Implementation of a Multiple Sensory System for the Detection of Fluid Losses in Ducts through Adaptive Neuro-Fuzzy Inference

Marco Carbajal-López[1], Anthony Valdivia-Diaz[2], Jose Briones-Zuñiga[3], Carlos Sotomayor-Beltran[4]

[1,2,3,4]*Universidad Tecnológica del Perú, Lima, Perú.*

[4]*Corresponding Author : c22137@utp.edu.pe*

Received: 23 August 2024 Revised: 11 November 2024 Accepted: 13 November 2024 Published: 29 November 2024

Abstract *- This work presents a sensor-based fluid pipeline leak detection system using an Adaptive Neuro-Fuzzy Inference System (ANFIS). The implemented model consists of multiple flow sensors and pressure differentials, processed through a hybrid Neural Networks and Fuzzy Logic hybrid system. The network comprises five layers, with layer 1 being in charge of performing the membership function and layer 5 being part of the calculation of the exit rule. The system aims to detect and prevent fluid leakage in fluid passages by identifying changes in fluid flow and pressure differential. The results demonstrate that the ANFIS system can accurately detect leaks in the ducts, reaching 93.24 % of accuracy, indicating the percentage of correct predictions in the training set. Additionally, a validation set not part of the training was used for the model's generalisation ability. This data set allowed us to measure patterns and characteristics of the model with new and previously unseen data.*

Keywords *- Leak detection, Pressure, ANFIS, Neural network, Python, TensorFlow.*

1. Introduction

Fluid systems are crucial in different industrial environments, mobile machinery, and liquid supplies, driven by compressed air or general fluids in different conduits. However, detecting leaks in these systems represents a critical challenge since these leaks can cause anything from premature wear to very important resource losses, triggering complete system stoppage, which is problematic in large environments [1] [2]. The need to detect a leak more accurately becomes more relevant in Industry 4.0, where energy efficiency and the ability to diagnose faults are essential. In this sense, the combination of advanced techniques can offer a new approach to the early identification of leaks, safeguarding fluid systems [3]. For this reason, a database with different sensors is needed to help detect possible leaks in the ducts. These sensors have the potential to evaluate and analyze the state of the ducts, providing valuable information to create an adaptive neuro-fuzzy inference network, which allows optimization in leak detection. Early detection of leaks in fluid system conduits is critical to conserving resources, reducing costs, and maintaining system efficiency. Combining detection methods with technology such as artificial intelligence reduces the time needed to locate leaks, thus mitigating the harmful effects of the system [4]. The present research focuses on the study and implementation of a multiple sensory system for the preventive diagnosis of leaks in fluid transport systems. Various methodologies have been investigated to increase the effectiveness and accuracy of leak detection, overcoming the limitations of conventional techniques. Piezoelectric and hydrophone sensors effectively detect leaks in fluid system conduits [5] [6]. These sensors use different methods to capture and analyze signals, each with specific conditions to improve the accuracy and speed of detecting fluid leaks. Techniques such as time inversion using negative pressure waves have been explored, where a monitoring system analyzes data for pressure variations and expansions associated with leaks in duct walls [5]. In addition, digital band-pass filters have been used to widen the signal-noise ratio and enable cross-correlation to be carried out in the frequency range, improving the detection of leaks in plastic ducts [6]. The pressure in the ducts is evaluated through different tests on the sensors that consider different parameters to acquire and process data. Negative pressure waves, associated with phenomena such as explosions or implosions, have been investigated to detect leaks when the pressure within the system falls below the atmospheric or reference pressure of a conduit [5]. On the other hand, when transient pressure waves of fluids generate sudden changes in pressure, they cause the flow to have wave-shaped variations, reaching peaks of high or low pressure [7]. One of the most current methods to process the data obtained is using non-biological methods, also called artificial neural networks (ANN), which stands out as a promising tool for improving leak detection. These networks can learn and adapt to the acquired

information, making them especially useful for detecting leaks in fluid systems [7]. A large database is needed to learn an ANN. For this, software such as the Environmental Protection Agency Network (EPANET) allows simulations of water flow in networks with supply under pressure [8]. Autoencoder integration of unsupervised machine learning models, such as autoencoder (AE) neural networks, learns from an imbalanced data set to classify conditions in the presence and absence of leaks in a supply network. This has improved the accuracy of detecting leaks in ducts in the monitored area range, considering multiple variables and data, including fluid pressure and signal noise [9]. Pneumatic systems automate production, but excessive energy consumption can indicate leaks. Energy efficiency is modeled by analyzing historical data and developing a predictive model to improve efficiency and detect possible leaks. Research on pneumatic strain energy accumulators also contributes to leak detection. An analytical model uses two-cylinder systems, recycling the primary cylinder to the secondary cylinder, which allows anomalies in air consumption to be detected. The applied model experimentally tests the system's efficiency with the accumulator, considering the volume relationship between cylinders and the behavior of the ideal gas. This methodology helps to identify efficiency losses related to leaks [10] [11].

Pneumatic parameters are analyzed to identify failures and leaks in fluid systems. The company Emerson Aventics develops monitoring systems that diagnose losses close to the source of waste through an automated monitoring process, continuous analysis, and data collection in valve cycles, air consumption and pressure stored in a Smart Pneumatics Monitor (SPM). These modern monitoring systems identify critical parameters through sensors, such as airflow, pressure, and temperature, saving time and resources. Being an improvement in cost reduction and better efficiency of flow systems [12]. Fluids such as compressed air in mobile machinery are complicated to detect due to their difficult location. Therefore, techniques are used in these types of systems, such as the unsupervised anomaly detection technique with the wavelet transform, which analyzes pressure fluctuations to identify possible leaks in the system, and a supervised machine learning algorithm is used to observe performance. The AE technique extracts and selects important features from unlabeled data before classifying it. Both techniques use machine learning, differing in their approach: Anomaly detection focuses on unlabeled data, while Autoencoder extracts relevant features. The choice between them depends on whether the data is labeled [3] [13].

The importance of identifying relevant parameters and optimizing the architecture of the ANNs is highlighted to improve the effectiveness of leak detection in different types of ducts. Furthermore, applications with the Venturi duct and Arduino stand out for their accessibility and economic reduction in continuous teaching methods according to Bernoulli's laws. The Venturino device is simple to build and configure with the POE (Predict, Observe, and Explain) methodology, demonstrating effectiveness for experiments at a reduced cost [14]. With today's technological advancement, variables and data can be processed quickly to solve problems. Thus, a hybrid algorithm is proposed that uses a fuzzy inference model CMAC (Cerebellum Model Articulation Controller), with fluid adaptation to evolving and real-time data sets. This algorithm requires simulated and real data, which is a more complete and complex method with artificial intelligence for better precision in predicting and detecting problems in non-linear systems, such as fluid movements in conduits or systems with dynamic movements [15]. In view of the above, this work will analyze the behavior of sensors in conjunction with the adaptive neuro-fuzzy inference applied in a duct with fluids. This is to find the optimal way to detect duct leaks with fluids.

2. Methodology

An air compressor will develop the system, with a pressure gauge coupled to its outlet and a PVC conduit. Pressure and flow sensors will be installed in this conduit, as well as a temperature and relative humidity sensor for their respective analysis.

2.1. Electronic Components

The detection and monitoring leaks in fluid conduits represent crucial aspects in numerous industries, from manufacturing to water resources management. In this context, selecting appropriate electronic components is important to ensure the detection system's effectiveness and accuracy. This article explicitly addresses the selection of pressure, flow, temperature, and relative humidity sensors for implementation in fluid conduit leak detection. This analysis seeks to identify the most suitable components for this application and understand their operation and capabilities to contribute to early and accurate leak detection.

2.1.1. HK3022 0.5MPa Pressure Sensor

The electronic devices that perform pressure measurements, as seen in Figure 1, known as pressure transducers or pressure sensors, are widely used in various processes within the industry. Its main function is to convert a physical measurement into an electrical signal. In this specific context, these devices convert the force applied per unit area (pressure) into a voltage value directly proportional to the magnitude of the pressure exerted. Changes in pressure could indicate the presence of a leak within the system.

2.1.2. 1/2" Flow Sensor YF-S201

As seen in Figure 2, this sensor operates on the principle of a rotating wheel stimulated by fluid flow inside a pipe. This wheel, equipped with magnets, induces a magnetic field that activates a Hall effect sensor. The wheel's rotation modifies this magnetic field, generating an electrical signal proportional to the rotation speed. This electrical signal is interpreted to quantify real-time fluid flow in the conduit. This

analog signal must be converted to digital by a microcontroller for its respective processing.

2.1.3. SHT31 Temperature and Relative Humidity Sensor

The SHT31 sensor, as shown in Figure 3, is based on capacitive technology and a platinum thermistor for accurately measuring both environmental parameters. For temperature measurement, the platinum thermistor experiences changes in its electrical resistance in response to temperature variations or fluctuations, resulting in accurate temperature readings through an internal electronic circuit. On the other hand, to measure relative humidity, the sensor uses a humidity-sensitive polymer layer that expands or contracts depending on the ambient humidity, thus modifying the capacitance of an internal capacitor. This variation in capacitance is converted into accurate relative humidity readings. This will be very useful for calibrating the sensors.

Fig. 1 Pressure sensor HK3022 0.5MPa

Fig. 2 1/2" Flow Sensor YF-S201

Fig. 3 SHT31 Temperature and relative humidity sensor

2.2. Mechanical Components

For the selection of components, it is essential to ensure the system's durability. These components have been chosen for their easy installation access and the ability to handle operating conditions, guaranteeing a constant and controlled fluid flow necessary for simulation under operating conditions and system evaluation.

2.2.1. Einhell TC-AC 190/24/8 Air Compressor Installation

In the developed system, an air compressor is installed, as seen in Figure 4, which was selected to align with the flow and pressure requirements of the system. The ability to provide a constant flow perfectly matches the specifications of the PVC conduit and sensors.

2.2.2. PVC Conduit Connection

The PVC conduit shown in Figure 5, connected to the outlet of the air compressor, serves as the main path for fluid flow. We have chosen a duct of appropriate size and thickness. Specifically, the tube has a measurement of ½" to withstand the air pressure without deformation. Also, the duct is the appropriate size for integrating the sensors, thus ensuring the system's integrity.

Fig. 4 Einhell TC-AC 190/24/8 air compressor

Fig. 5 ½" PVC conduit

Fig. 6 Arduino UNO board

2.3. Mounting the Pressure and Flow Sensors

2.3.1. HK3022 0.5MPa Pressure Sensor

This sensor is installed at a critical point in the PVC conduit to measure real-time fluid pressure. Thus, ensuring the joints are well-sealed prevents leaks and provides accurate readings.

2.3.2. 1/2" Flow Sensor YF-S201

Located in line with the duct, this sensor measures the flow of fluid passing through the system. Its location lets the sensor be immersed entirely in the flow, guaranteeing accurate measurements.

2.3.3. Integration of SHT31 Temperature and Relative Humidity Sensor

This sensor is mounted near the PVC conduit, where it can accurately measure the temperature and relative humidity of the environment. The location allows environmental conditions to affect the sensor without external interference.

2.3.4. Arduino UNO Board

The Arduino Uno, equipped with the ATmega328P microcontroller, reads data from the flow and pressure sensors thanks to its analog and digital input capabilities (Figure 6).

2.3.5. For 1/2" flow sensor YF-S201

The Arduino Uno uses a digital input that detects pulses generated by this sensor. These pulses are counted using an interruption routine, allowing the fluid flow to be calculated according to the frequency of the pulses.

2.3.6. For the HK3022 0.5MPa Pressure Sensor

An analog input of the Arduino Uno is used for this sensor. The pressure sensor generates a voltage signal proportional to the measured pressure, which the Arduino's integrated ADC converts to a digital value. This digital value represents the applied pressure and is processed for analysis.

2.4. Sensor Conversion and Calibration Process

The sensors are connected to a microcontroller for data acquisition that will allow the conversion of the analog values that the sensors will provide to digital to use as training data for the ANFIS network.

2.4.1. Measurement of Fluid Volume or Consumption

Flow sensors are essential for obtaining accurate and reliable measurements of fluid flow. This involves establishing a precise relationship between the sensor output and the amount of fluid passing through the system. The flow rate Q is directly related to the ΔP, measured by a manometer, during a Δt (Equation 1). This pressure measurement helps in the results of systems that use compressed air or another fluid.

$$Q = \frac{\Delta P}{\Delta t} \ldots "Flow" \tag{1}$$

2.4.2. Flow Sensor Calibration

To have a more accurate pulse measurement, the pulses generated by the sensor are captured using technology such as Arduino. Next, the pressure gauge will be used to measure the pressure variation as the fluid flows through the system, providing a reading of ΔP. To calculate the conversion factor, K is calculated as seen in Equation 2 between the ΔP measurement and the number of pulses $n°P$ recorded by the sensor.

$$K = \frac{\Delta P}{n°P} \tag{2}$$

This K factor converts sensor pulse readings into fluid flow units, ensuring consistent and accurate measurements.

2.5. ANFIS Model

The Adaptive Neuro-Fuzzy Inference System (ANFIS) is a hybrid system combining two soft computing methods, such as fuzzy logic and neural networks, that apply evolutionary algorithms, effectively regulating the uncertainty and imprecision of problems. who need continuous learning regarding their environment.

This hybrid system works with membership functions, which allows ANFIS to be adaptable, resistant to failures and capable of generalizing its fuzzy rules to approximate non-linear functions, improving the flexibility and precision of the detection system [16] [17].

2.5.1. Architecture of the ANFIS Model

The ANFIS model is an adaptive network that combines fixed and adjustable nodes. The architecture of this network is shown in Figure 7, which shows a multilayer network where each node performs a specific function on the input signals. The neuro-fuzzy system divides the structuring of prior knowledge into subsets that facilitate the reduction of the search space, using the backpropagation algorithm to adjust the fuzzy parameters.

These subsets are called Membership Functions. The resulting system is similar to a first-order Takagi-Sugeno inference system, where the input-output relationship is linear and maintains its set of fuzzy If-Then rules. The ANFIS model consists of an adaptive neural network with five layers, each playing a specific role in the fuzzy inference process [18].

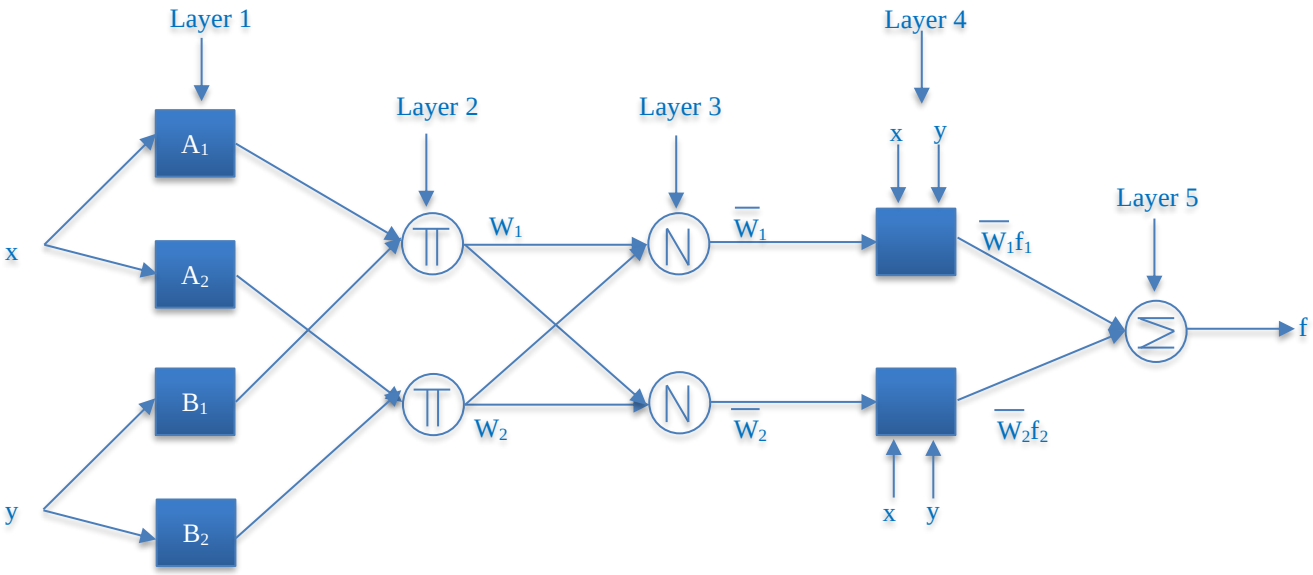

Fig. 7 Adaptive ANFIS configuration for first order TS inference

Layer 1: Membership Functions

- The inputs to this layer are "xy" pairs of the system.
- The individual output of each node is the degree of membership, which produces a value that indicates the extent to which the input satisfies the associated linguistic term (Equation 3).

$$O_i^1 = A_i(x) \tag{3}$$

Layer 2: Rule Nodes

- Each node evaluates the degree of activation of the specific rule.
- The nodes use a T-norm to simulate the "and" logic operation (Equation 4).

$$O_i^2 = W_i = A_i(x).B_i(y), i = 1, 2 \ldots \tag{4}$$

Layer 3: Normalization

- These nodes adjust the degree of activation to a common scale.
- The output demonstrates the activation level normalized relative to the total number of activations (Equation 5).

$$O_i^3 = \hat{W}_i = \frac{W_i}{W_1+W_2}, i = 1, 2 \ldots \tag{5}$$

Layer 4: Parameters of the Consequent

- The output of each node is calculated by multiplying the normalized activation degree by the specific output of each rule.
- This layer's parameters, p, q and r, are adjustable and represent the coefficients of the linear functions in the consequent rules (Equation 6).

$$O_i^4 = \hat{W}_i f_i = \hat{W}_i(p_i x + q_i y + r_i) \tag{6}$$

Layer 5: Aggregation

- A single node in this layer computes the final output of the system as well as the sum of all the individual outputs of the previous nodes (Equation 7).

$$O_i^5 = \sum_i \hat{W}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{7}$$

2.5.2. Logical Description of the ANFIS Model

The ANFIS model has the same function as a first-order Takagi-Sugeno (TS) inference system, which is based on the following fuzzy rules:

$$Rule\ 1: If\ "x"\ is\ A_1\ and\ "y"\ is\ B_1, then$$
$$f_1 = p_1 x + q_1 y + r_1$$

$$Rule\ 2: If\ "x"\ is\ A_2\ and\ "y"\ is\ B_2, then$$
$$f_2 = p_2 x + q_2 y + r_2$$

As can be seen in Figure 8, $A_{1,2}$ y $B_{1,2}$ are the membership functions (fuzzy sets), p_i, q_i and r_i, are adjustable parameters.

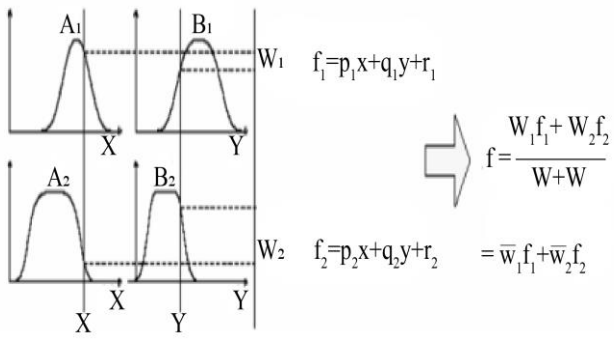

Fig. 8 Adaptive ANFIS configuration for first order TS inference

The activation levels of each of the rules are calculated as $W_i = A_i(x).B_i(y), i = 1, 2$ ..., where the logical operator can be modeled by a t-norm. The control output of the model is obtained by normalizing these degrees of activation of the fuzzy rules and combining them linearly between the antecedent parameters of each rule:

$$f_i = p_i x + q_i y + r_i, \mathrm{i} = 1, 2 \ldots \quad (8)$$

$\hat{W}_1$ y $\hat{W}_2$ are the normalized values from W_1 y W_2 [19].

2.5.3. Learning the ANFIS Model

Learning in the ANFIS model is carried out through a hybrid approach that combines the following:

Gradient Descent Algorithm

To optimally adjust the antecedent parameters, which characterize the membership functions.

Least Squares Algorithm

To determine the linear parameters of the consequent, during each training cycle, a forward pass is carried out where the parameters of the membership functions are initialized, and the outputs of the nodes are calculated.

Then, the parameters of the consequent are fitted using least squares. Subsequently, backward pass to adjust the antecedent parameters using the gradient descent algorithm [18].

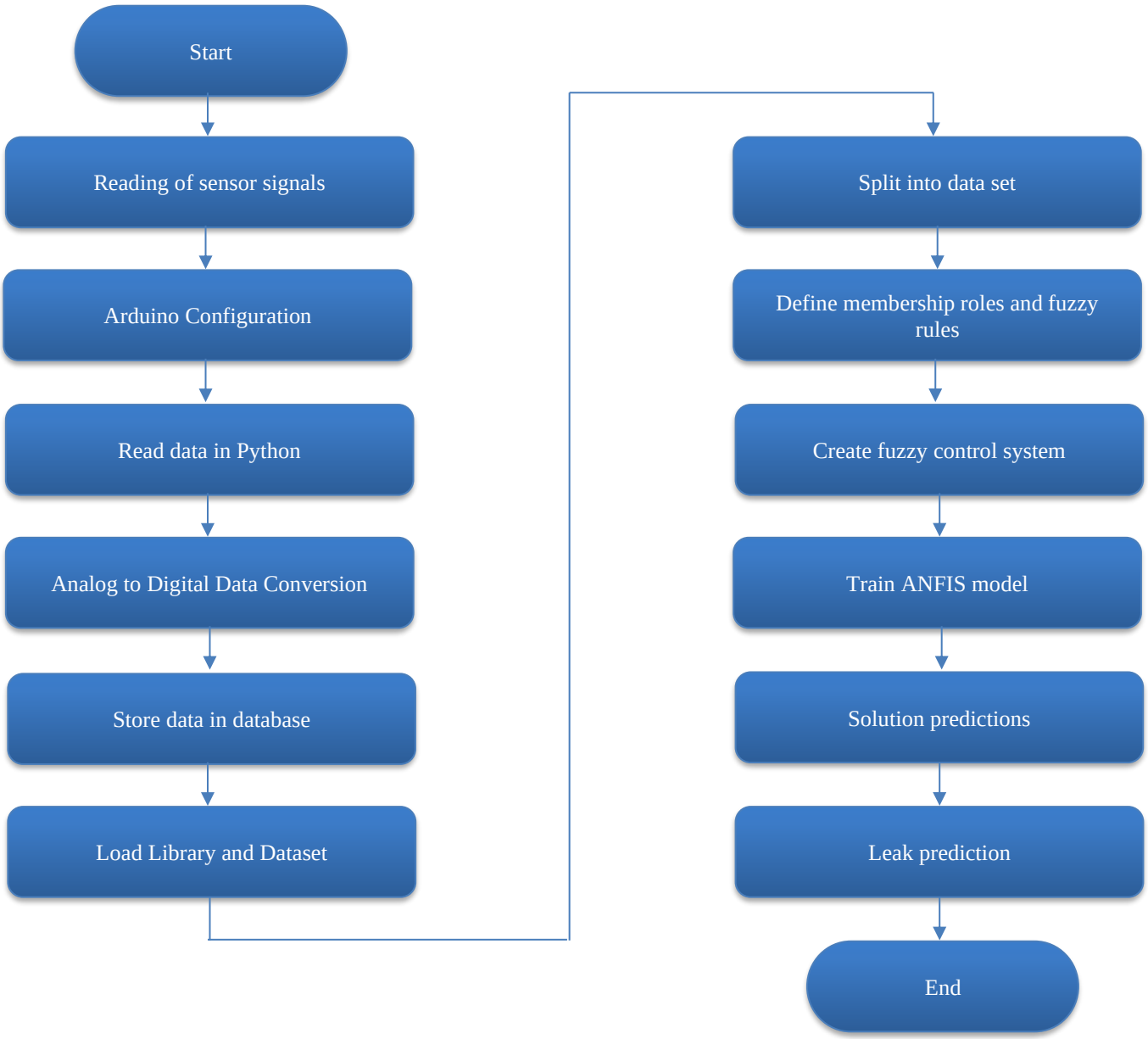

Fig. 9 Block diagram of general system processing

2.5.4. ANFIS Model Training

If-Then Rules

In an ANFIS system with flow and pressure sensor inputs, "If-Then" rules relate these readings to the system outputs. Each rule follows the format "If (condition) Then (consequence)", such as "If the flow is high and the pressure is low, Then the output is (value)." These rules are generated and optimized from sensor data using machine learning, allowing uncertainty in the data to be managed and providing flexible and robust decision making.

Hybrid Learning Method

Learning in ANFIS involves using algorithms based on gradient descent to optimize the parameters of the antecedent and an algorithm based on least squares to determine the linear parameters of the consequent [18].

Forward or FeedForward step

During this step, the parameters of the membership functions are initialized, the parameters of the membership functions are adjusted, and the input-output vectors are displayed. For each layer of the network, the outputs of the nodes are calculated, and the consequent parameters are determined using the least squares method. [19].

Backward Step

The model parameters are adjusted based on the calculated error, which is the difference between the network output and the desired output. The backpropagation algorithm minimises this error by adjusting the parameters in the direction that reduces it. This adjustment is crucial for the model to learn and improve its accuracy in future predictions [19].

2.6. Computer System Implementation

Two software options were considered for implementing the fluid leak detection system in ducts using ANFIS: MATLAB and Python. Both options offer significant advantages for system development and deployment.

MATLAB: It is a programming and software development environment widely used in engineering and science. Provides an intuitive interface and powerful tools for data analysis, modeling, and simulation. The Fuzzy Toolbox Logic and the Neural Network Toolbox are particularly useful for ANFIS implementation.

Python: It is a high-level programming language that has gained wide popularity in the scientific and engineering fields. Additionally, it offers a wide range of library and machine learning tools, including TensorFlow, Scikit-Fuzzy, and ANFIS, making it easy to implement and analyze fuzzy logic systems.

2.6.1. Computational Development of the ANFIS Model

The overall system (seen as a block diagram in Figure 9) will be implemented using Python, taking advantage of the advanced capabilities and tools available on the Google Colab platform. Data from the sensors will be collected using an Arduino board, which will serve as the data acquisition system.

Software

Python has been chosen due to its versatility and the availability of specialized libraries such as TensorFlow, Scikit-Fuzzy and a specific library for ANFIS. TensorFlow will provide the infrastructure necessary to build and train efficient neural networks. Scikit-Fuzzy will facilitate the design and manipulation of fuzzy logic systems, which are essential for implementing the ANFIS model, while the ANFIS library will integrate these components into a coherent and functional system.

Development Platform

Google Colab will be used as the development platform because it can work with large volumes of data and execute intensive computational processes in a collaborative environment. Colab offers access to accelerated hardware, such as TPUs, which accelerates the training of complex models and improves performance. Additionally, Colab facilitates integration with Google Drive, allowing efficient data management and real-time collaboration between researchers.

Model Development

The use of TensorFlow, which is an open-source library for machine learning across a range of tasks, also serves for the construction and training of neural networks, Scikit-Fuzzy for the management of fuzzy logic and the ANFIS library for the integration of these components into a coherent model.

Training and Validation

The data collected by the sensors will be used to train the ANFIS model. The model will be adjusted and validated using statistical techniques and data visualization tools.

Data Analysis and Validation

The data collected in the experimental tests will be analyzed to evaluate the accuracy and effectiveness of the ANFIS model in detecting leaks. Cross-validation techniques will be used to ensure the reliability of the model. The infrastructure provided by TensorFlow will allow the model parameters to be optimized and adjusted, improving its accuracy and efficiency. Scikit-Fuzzy will provide robust tools for manipulating and configuring fuzzy logic systems.

3. Results and Discussions

Different studies worldwide have shown that leak detection has substantially accelerated growth after the year 2000 due to the concern of preserving the resources of the different fluids, evidencing the diversity in the approaches and techniques to develop new technologies to avoid losses of

these [2]. Among the countries with a wide distribution of these studies are China, the United States, England, and Canada, which demonstrate collaboration between them to mitigate the problem of duct leaks. Here, we present our results.

3.1. Experimental Tests

Once the ANFIS model was developed on the Google Colab platform, experimental tests were conducted using the selected equipment and electronic components.

These tests included the following stages:

3.1.1. Sensor Calibration

Pressure and flow sensors were calibrated to ensure accuracy and reliability of readings. Calibration was performed using standard methods and reference equipment to ensure the accuracy of the data collected.

3.1.2. Simulation of Operating Conditions

Different operating conditions of the fluid system were simulated, including variations in flow and pressure, as well as the introduction of controlled leaks at the end of the conduit.

3.1.3. Data Register

Data from the sensors was recorded in real-time using a microcontroller that collected and transmitted the readings to a central database. This data included detailed information on the operating conditions and simulated leak events, in which approximately 300 samples were obtained between leaks and non-leaks.

3.1.4. Evaluation of the ANFIS Model

The recorded data was used to evaluate the ANFIS model's ability to detect and locate leaks in the system. The model's performance was analyzed in terms of accuracy, sensitivity, and specificity, comparing it with the results of traditional leak detection methods. In Figure 10, the duct system can be seen with the pressure sensor installed in such a way that it can detect the pressure inside the duct.

Fig. 10 HK3022 sensor assembly

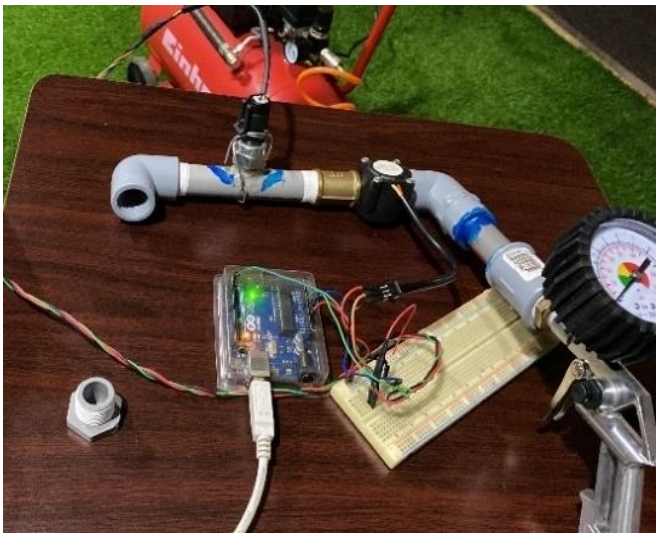

Fig. 11 System assembly

Next, in Figure 11, the system is installed with the YF-S201 flow sensor and the HK 3022 0.5MPa pressure sensor. Both sensors will provide information in real-time in conjunction with the Arduino, and these values will be reflected in the serial monitor of the same Arduino Idle programming platform. These data are based on each sensor, respectively, in which flow values are provided, expressed in Liters/seconds (L/s) and pressure values expressed in pounds per square inch or its acronym in English (PSI).

3.2. Data Analysis

In this study, the data collected during the experimental tests were analysed using advanced statistical techniques and data visualization tools in Python. Data were obtained from pressure and fluid flow sensors installed in a conduit and recorded in an Excel spreadsheet. These data served as inputs for training the ANFIS network. The Pandas library was Used for data manipulation and cleaning. Pandas' functions enabled efficient preprocessing, handling missing data, normalizing values, and preparing data for further analysis. The Matplotlib library was used to visualise detailed plots illustrating the distribution of the data, the relationships between the input variables (pressure and flow) and the system outputs, and to visualize the performance of the ANFIS model. The Scikit-fuzzy library was used to define and graphically represent fuzzy membership functions, crucial in fuzzy inference systems such as ANFIS (Adaptive Neuro-Fuzzy Inference System), where they transform sharp inputs into fuzzy values used in the inference process. These functions allow the input space to be divided into regions characterized by fuzzy labels such as "low", "medium", and "high", applicable to flow and pressure. This is essential in ANFIS, modeling the uncertainty and vagueness inherent in real data. Membership functions were used to create input variables such as flow and pressure. Additionally, membership functions were graphically represented to understand better how entries are categorized in fuzzy terms.

The analysis process was carried out in several stages. First, data preprocessing was carried out, which included data cleaning, normalization, and segmentation of training and test sets. Then, using the preprocessed data, the ANFIS model was trained to identify pressure and flow reading patterns that indicate leaks. The accuracy and effectiveness of the ANFIS model were evaluated using the test data. These data underline that combining ANFIS with advanced data analysis techniques in Python is a powerful and efficient tool for accurate leak detection in fluid systems. This approach not only improves the speed and accuracy of the detection process but also offers significant advantages over conventional methods, thus promoting more effective and sustainable resource management. In Figure 12, the system is shown in operation. The pressure gauge is connected directly to the compressor's air outlet. It is connected to the duct with the sensors within the system, indicating that the duct and the sensors are already pressurized. The air compressor pressurizes the system at 20 psi, and this, being a hermetic system, presents no leak. In Figure 13, we can see how the values of the HK3022 pressure sensor are shown in real-time.

Fig. 12 Implemented system

```
00:34:47.779 -> Flow: 0.00 L/s, Pressure: 20.66 PSI
00:34:48.776 -> Flow: 0.00 L/s, Pressure: 20.66 PSI
00:34:49.773 -> Flow: 0.00 L/s, Pressure: 20.66 PSI
00:34:50.770 -> Flow: 0.00 L/s, Pressure: 20.54 PSI
00:34:51.767 -> Flow: 0.00 L/s, Pressure: 20.54 PSI
00:34:52.758 -> Flow: 0.00 L/s, Pressure: 20.43 PSI
00:34:53.758 -> Flow: 0.00 L/s, Pressure: 20.43 PSI
00:34:54.759 -> Flow: 0.00 L/s, Pressure: 20.43 PSI
00:34:55.756 -> Flow: 0.00 L/s, Pressure: 20.43 PSI
00:34:56.759 -> Flow: 0.00 L/s, Pressure: 20.43 PSI
00:34:57.758 -> Flow: 0.00 L/s, Pressure: 20.31 PSI
00:34:58.757 -> Flow: 0.00 L/s, Pressure: 20.31 PSI
00:34:59.757 -> Flow: 0.00 L/s, Pressure: 20.20 PSI
00:35:00.758 -> Flow: 0.00 L/s, Pressure: 20.31 PSI
00:35:01.759 -> Flow: 0.00 L/s, Pressure: 20.31 PSI
00:35:02.758 -> Flow: 0.00 L/s, Pressure: 20.20 PSI
00:35:03.759 -> Flow: 0.00 L/s, Pressure: 20.20 PSI
00:35:04.758 -> Flow: 0.00 L/s, Pressure: 20.20 PSI
00:35:05.779 -> Flow: 0.00 L/s, Pressure: 19.95 PSI
```

Fig. 13 Series monitor with values under pressure

It should be noted that these measurements are the product of the sensors calibrated together with the manometer at the entrance of the duct, as shown in Figure 14 since this provides us with a measurement of the pressure at the compressor's air outlet. Since this is a known value, we can correctly calibrate the sensors to obtain more precise measurements. Internally, the program has carried out a unit conversion because the pressure sensor gives us a measurement in MPa. After pressurizing the system, the plug at the end of the duct is progressively opened to simulate a leak within the system, as seen in Figure 15.

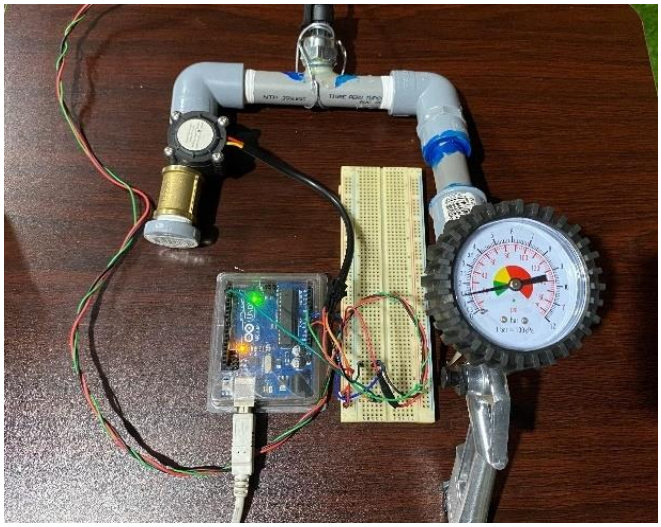

Fig. 14 System with constant pressure

Fig. 15 System with a fluid leak

```
00:24:07.581 -> Flow: 3.33 L/s, Pressure: 6.59 PSI
00:24:08.564 -> Flow: 7.33 L/s, Pressure: 7.42 PSI
00:24:09.536 -> Flow: 10.13 L/s, Pressure: 11.34 PSI
00:24:10.595 -> Flow: 9.20 L/s, Pressure: 11.99 PSI
00:24:11.573 -> Flow: 8.80 L/s, Pressure: 11.99 PSI
00:24:12.596 -> Flow: 7.20 L/s, Pressure: 12.94 PSI
00:24:13.567 -> Flow: 7.20 L/s, Pressure: 13.53 PSI
00:24:14.590 -> Flow: 6.80 L/s, Pressure: 14.25 PSI
00:24:15.571 -> Flow: 6.80 L/s, Pressure: 14.35 PSI
00:24:16.595 -> Flow: 7.60 L/s, Pressure: 12.94 PSI
00:24:17.577 -> Flow: 7.00 L/s, Pressure: 13.55 PSI
00:24:18.571 -> Flow: 6.13 L/s, Pressure: 13.94 PSI
00:24:19.595 -> Flow: 5.73 L/s, Pressure: 13.53 PSI
00:24:20.567 -> Flow: 5.73 L/s, Pressure: 13.14 PSI
00:24:21.570 -> Flow: 5.73 L/s, Pressure: 12.77 PSI
00:24:22.565 -> Flow: 5.60 L/s, Pressure: 13.15 PSI
00:24:23.594 -> Flow: 6.00 L/s, Pressure: 12.77 PSI
00:24:24.571 -> Flow: 6.00 L/s, Pressure: 12.88 PSI
00:24:25.595 -> Flow: 6.13 L/s, Pressure: 11.88 PSI
00:24:26.568 -> Flow: 6.20 L/s, Pressure: 11.85 PSI
00:24:27.572 -> Flow: 6.40 L/s, Pressure: 11.88 PSI
00:24:28.566 -> Flow: 6.40 L/s, Pressure: 11.88 PSI
00:24:29.570 -> Flow: 6.60 L/s, Pressure: 11.88 PSI
00:24:30.595 -> Flow: 6.40 L/s, Pressure: 11.88 PSI
```

Fig. 16 Sensor values when subjected to a leak

Following this, it can be seen in Figure 16 that both sensors present alterations in their values because of the leak caused. These are the values reflected on the serial monitor. These values will be later stored in a database.

3.3. Training with the ANFIS Network

In Figures 17 and 18, part of the database with the values provided by the sensors extracted with the Arduino and the serial monitor can be seen, with and without a leak, respectively. These collected values were saved in an Excel file that was later converted to a file with a CSV extension for better data analysis within the ANFIS training.

	Flow	Pressure	Leak
0	0.00	6.41	1
1	0.00	6.41	1
2	0.00	6.41	1
3	0.00	6.35	1
4	0.00	6.35	1
5	0.40	6.41	1
6	2.13	6.47	1
7	5.20	6.53	1
8	8.13	6.41	1
9	10.67	6.35	1

Show 10 per page 1 2 10 17

Fig. 17 Database when there is an anomaly

	Flow	Pressure	Leak
90	0.00	23.93	0
91	0.00	23.81	0
92	0.00	23.69	0
93	0.00	23.57	0
94	0.00	23.28	0
95	0.00	22.03	0
96	0.00	16.33	0
97	0.00	24.94	0
98	0.00	24.70	0
99	0.00	24.58	0

Show 10 per page 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

Fig. 18 Database when there is no anomaly

Once the data was obtained within the CSV file, the ANFIS network code was executed in the Google Colab environment, as seen in Figure 19, giving us the graphs of the membership functions of each sensor as results; these can be seen in Figures 20 and 21. Figure 20, which belongs to the membership functions of the flow sensor, indicates the ranges and values between work under normal conditions (without a leak) and when a leak occurs.

It can be seen in Figure 20 that the green line indicates that the flow sensor, when subjected to a low incidence of leakage, does not show any alteration until 8 L/s, at which point the internal rotor blades that the sensor has begun to move. On the other hand, the blue line indicates that when starting with a leak in the system, the flow is detected slowly until it passes the 8 L/s threshold. As a result, the orange line gives us the optimal operating range for the flow sensor with respect to its measurements between 4 L/s and 12 L/s.

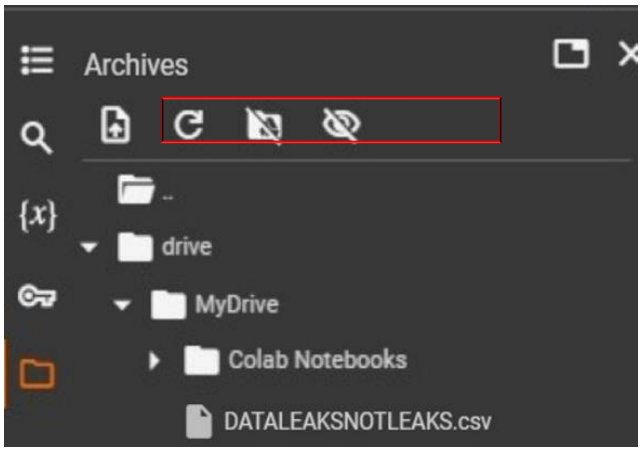

Fig. 19 Database mounted in Google Colab

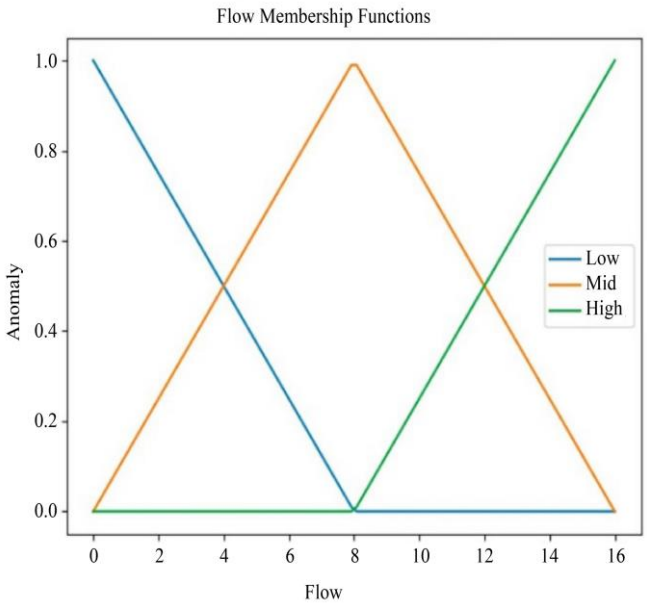

Fig. 20 Membership functions for flow

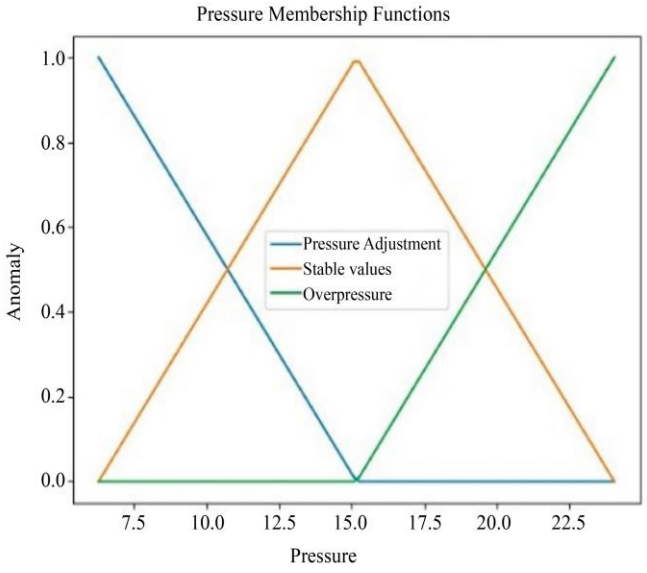

Fig. 21 Membership functions for Pressure

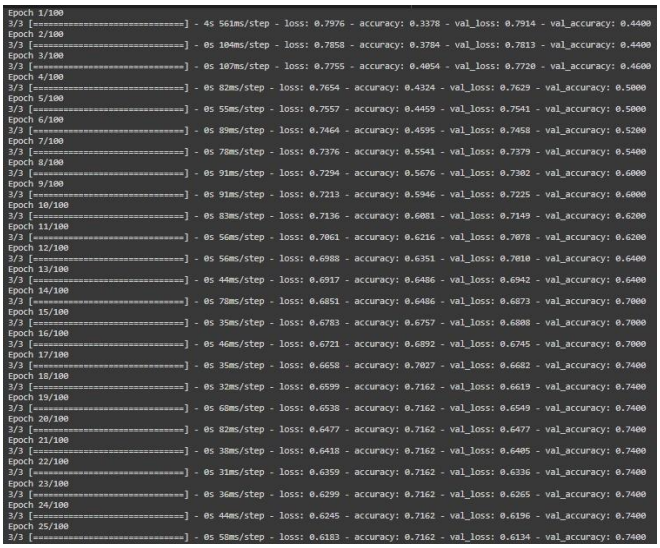

Fig. 22 First 25 epochs of training

Likewise, the same interpretation of the membership functions of the flow sensor is fulfilled for the pressure sensor. In Figure 21, data is observed to interpret operation conditions without a leak (pressurized system) and the system's operation with a leak or alteration of pressure within the duct. Figure 21 shows that when subjected to an initial pressure, the green line remains constant without anomalies up to 15 psi. From this point, it can be inferred that the probability that the system presents an anomaly increases. Likewise, the blue line indicates that the system starts with a leak that is subsequently corrected until it begins to pressurize at 15 psi; the orange line represents this pressure value. These methods were implemented to evaluate the stability of a pressurized system, showing the result of the pressure that remained constant in an optimal range of 10 to 20 psi, indicating an improved level of stability and observing a low percentage of anomalies. The training of the ANFIS network was carried out, considering that the parameter of the number of epochs and iterations is very important so that the effectiveness of the model converges to an optimal solution and that, if the epochs and iterations are low, the network does not learn enough and underfits the data increasing the margin of error. The number of epochs and iterations with which the model was initially tested was 25 epochs and 15 iterations each, being parameters that were too low for training the network and a problem for the accuracy of the training. One of the consequences is that learning converges to a point where there is no good similarity. Since the precision in the validation set after training is 61.2%, which varies with ±3% if it is retrained, this result suggests that the accuracy of recognition will be doubtful. For this reason, the ANFIS network is retrained by increasing the parameters of the number of epochs and the number of iterations, as shown in Figures 22 and 23. During this process, 100 epochs were carried out to adjust the ANFIS modeling parameters better. Each epoch worked on processing 33 iterations in 3 batches in training and thus was able to have a better observation regarding the results obtained. The average time per batch was 65 milliseconds, highlighting the speed of model processing in each training cycle.

In Figure 23, after completing the 100 epochs, the stabilization of the training set loss was at 25.21 %. This indicates that the model learning did not overfit the training data, which is crucial in representing the closeness to the model predictions of the actual results during training. The accuracy achieved was 93.24 %, indicating that the predictions were 93.24 % correct in the training set. A validation set not part of the training was used to evaluate the model's generalization ability. This data set allows you to measure patterns and characteristics of the model with new and previously unseen data. The loss of the validation set stabilized at 21.43 %, indicating that good performance is maintained on unseen data, and the accuracy in the validation set was 94 %, these values being essential to measure the model's performance in generalising and predicting with new data.

In Figure 24, the test set simulates real-world conditions; 6 batches were processed with an average time of 4 milliseconds, making the model lose 20.28 %, which indicates its average model error. This loss is a measure to see if the model predicts the expected result on the test data set. The lower this value is, the better the model's predictive ability on the test set. The model also achieved an accuracy of 96.11 %. It should be noted that the accuracy of the training was lower than that of the tests. However, as the difference was minimal, it did not affect the ability of the model to generalize and make predictions, demonstrating a high capacity of the model to generalize previously unseen data. The results of the ANFIS model show a robust ability for anomaly classification.

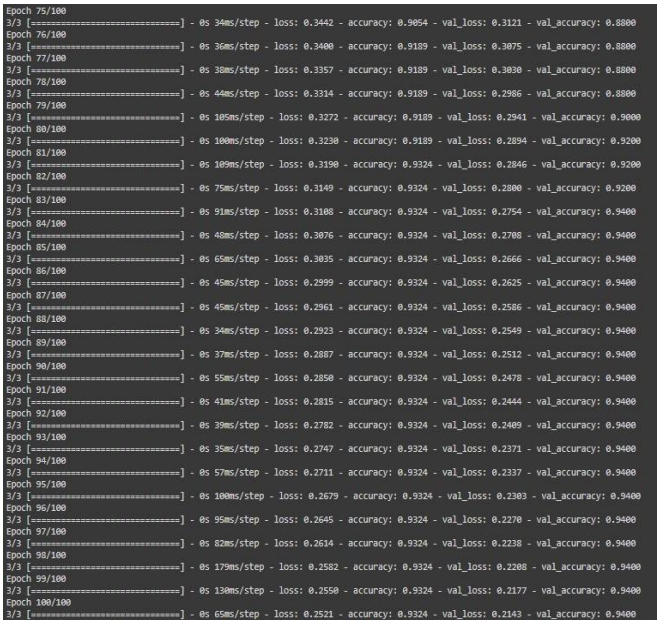

Fig. 23 Last 100 training epochs

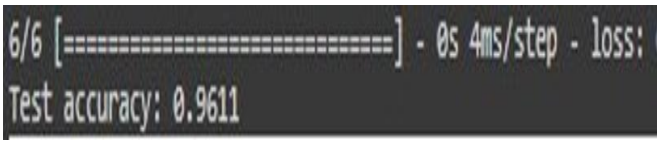

Fig. 24 Training test set

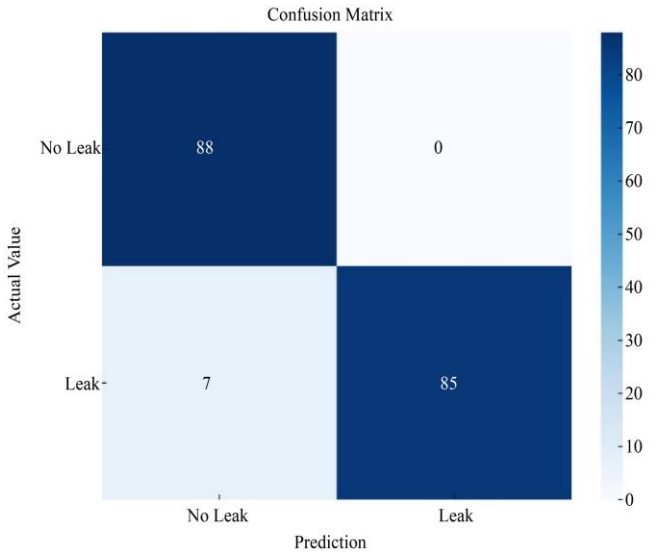

Fig. 25 Confusion or error matrix

As seen in Figure 25, the confusion matrix is essential for evaluating the performance of the classification model. For our model, ANFIS showed 88 instances of class 0 (No Leak) well classified, 85 instances of class 1 (Leak) correctly classified, no instances of class 0 misclassified as 1 and 7 instances of class 1 misclassified as 0. These results indicate a high performance in the classification of both classes, with few errors, especially in the no leak class. For a better observation regarding the confusion matrix, we have the following results:

- True Negatives (TN): Cases correctly classified as No Leak are 88.
- True positives (TP): Cases correctly classified as Leak are 85.
- False Positives (FP): Cases incorrectly classified as Leakage when they are actually Non-Leakage are 0.
- False Negatives (FN): Cases incorrectly classified as No Leak when in reality they are Leak are 7.

In Figure 26, the classification report is shown.

Classification Report:

	precision	recall	f1-score	support
0	0.93	1.00	0.96	88
1	1.00	0.92	0.96	92
accuracy			0.96	180
macro avg	0.96	0.96	0.96	180
weighted avg	0.96	0.96	0.96	180

Fig. 26 Classification report

The precision for class 0 (No leak) was 93 %, and for class 1 (leak) was 100 %, calculated using Equation 9.

$$Precision = \frac{TN}{TN + FN}\ (No\ leak);\ \frac{TP}{TP + FP}\ (Leak)$$

$$Class\ 0 = \frac{88}{88 + 7} = 0.93$$

$$Class\ 1 = \frac{85}{85 + 0} = 1 \qquad (9)$$

For the Recall (sensitivity), class 0 was 100 %, and class 1 was 92 %, and these were calculated using Equation 10.

$$Recall = \frac{TN}{TN + FP}\ (NoFuga);\ \frac{TP}{TP + FN}\ (Fuga)$$

$$Clase\ 0 = \frac{88}{88 + 0} = 1.0$$

$$Clase\ 1 = \frac{85}{85 + 7} = 0.92 \qquad (10)$$

For the F1-Score, both classes reached 96 %. For these, Equation 11 was used.

$$F1 - S = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall}$$

$$Class\ 0 = \frac{0.93\ x\ 1.0}{0.93 + 1.0} = 0.96$$

$$Class\ 1 = \ 2\ x\ \frac{1.0\ x\ 0.924}{1.0 + 0.924} = 0.96 \quad (11)$$

For the overall accuracy or precision, 96 % was reached. For the calculation, Equation 12 was used.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{85 + 88}{85 + 88 + 0 + 7} = 0.96 \quad (12)$$

As seen in Figure 27, the evolution of the loss or error during the training and validation of the model is composed of the X axis, which is the total epochs, while the Y axis is the model loss, considering that the lower the loss, the better the performance. The blue curve refers to the loss of the training set, which decreases as training progresses, indicating that the model learns. It is observed in the orange curve that the loss in the validation set decreases, referring to an improvement in the generalization capacity of new data.

We can separate the graph into 3 phases: initial, middle, and final. In the initial phase (0 – 20 epochs), both curves fall rapidly, indicating that the model captures relevant patterns in the data; in the middle phase (21 – 60 epochs), the decline of the curves is slow, demonstrating continuous improvement. In the final phase (61 – 100 epochs), both curves continue to decrease and remain close, converging at low values, suggesting a correct generalization without overfitting in the training and validation data. Figure 28 shows how the global accuracy evolves during model training and validation over 100 epochs. The number of epochs is observed in the ability to make correct predictions.

On the other hand, the orange curve shows the precision in the validation set, showing that the model also increases its ability to generalize to new data. In the initial phase (0 - 20 epochs), both curves rise rapidly from 0.4 to 0.7, indicating rapid learning. During the middle phase (21 - 60 epochs), the accuracies continue to increase, although they present some fluctuations. Finally, in the final phase (61 - 100 epochs), both accuracies stabilize around 0.9, with minimal differences between the curves, suggesting a more refined fit and good generalization capacity of the model.

When comparing the graphs of Figures 27 and 28, a consistency in the model's performance is noted: the loss decreases while the precision increases. Both graphs show that the model improves steadily and stabilizes towards the end of training. This analysis ensures that the model adequately fits the training data and makes accurate predictions on new data. Figure 29 highlights how the learning rate varies over 100 training epochs. The X-axis represents the number of epochs, while the Y-axis shows the learning rate. This parameter controls the magnitude of the model weight updates. The blue curve illustrates how the learning rate decreases at certain points during training, indicating using a learning rate reduction strategy (learning rate decay).

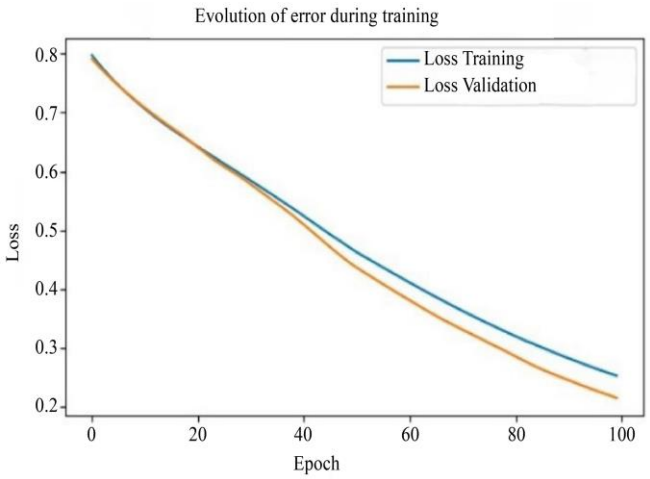

Fig. 27 Evolution of the error during training

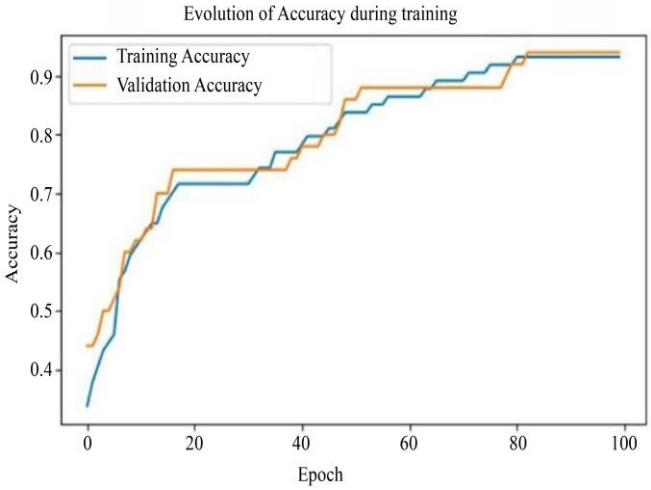

Fig. 28 Evolution of global precision or accuracy during training

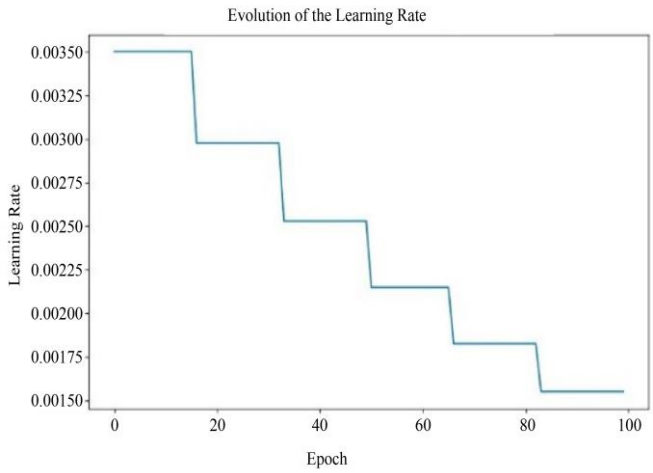

Fig. 29 Evolution of the learning rate

In the initial phase (0 to 20 epochs), the learning rate starts high and remains constant, allowing the model to learn general patterns quickly. In the final phase (80 to 100 epochs), the

learning rate stabilizes at a low value (approximately 0.0015), allowing small fine adjustments and stabilization. Starting with a high learning rate facilitates rapid updates and efficient initial learning. Step reduction allows for fine adjustments and improved precision. Keeping a low learning rate at the end prevents large overfitting and allows for more precise adjustments. There are different models for machine learning, among which is the SVM model, which is useful for leak detection. The training and validation loss curves converge to zero, suggesting that the model does not overfit and effectively reconstructs original data [13]. In both the aforementioned work and ours cases, the loss curves decrease steadily during training, indicating continuous improvement in the model. Comparing the accuracy, both models show an accuracy greater than 90 % during training and validation, with minimal differences between the accuracy curves. This indicates that our model has no overfitting and excellent generalization ability. Our study highlights the effectiveness of combining multiple signals to improve model accuracy and demonstrate that the model can be trained, maintaining high accuracy in detecting anomalies. The use of pressure and flow sensors is effective, and the accuracy is comparable to that reported in previous studies using advanced machine learning models.

4. Conclusion

This study demonstrated that the implemented system achieved 96 % overall accuracy in classifying fluid losses in ducts. In addition, the error's evolution during the model's training and validation is presented, which demonstrated continuous improvement and suggested a correct generalization without overfitting in the training and validation data. In conclusion, the proposed system presents a promising and efficient approach for detecting fluid losses in ducts using Adaptive Neuro-Fuzzy Inference. This system could be implemented in pipeline monitoring systems, which would allow the early identification of fluid losses and the implementation of the corresponding repairs, thus reducing resource waste and environmental impact. In view of the study presented, the analysis showed that the system was able to adapt to groups of data in real-time, so it would be interesting to implement the system on a real-time platform for use in the real world, considering the use of components of higher quality and a faster microcontroller for their respective analysis. In addition, it would be important to analyze the economic viability of implementing the proposed system in different scenarios to determine its impact on the efficiency and costs of pipeline operation.

References

[1] K.G. Pugin, "Improving the Reliability of Hydraulic Systems of Technological Machines," *IOP Conference Series: Materials Science and Engineering*, vol. 971, pp. 1-5, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Samer El-Zahab, and Tarek Zayed, "Leak Detection in Water Distribution Networks: An Introductory Overview," *Smart Water*, vol. 4, no. 5, pp. 1-23, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] Antoine Desmet, and Matthew Delore, "Leak Detection in Compressed Air Systems Using Unsupervised Anomaly Detection Techniques," *Annual Conference of the PHM Society*, vol. 9, no. 1, pp. 1-10, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sahil Adsul, Ashok Kumar Sharma, and R.G. Mevekari, "Development of Leakage Detection System," *International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, Pune, India, pp. 673-677, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[5] Guangmin Zhang et al., "A Time Reversal Based Pipeline Leakage Localization Method with the Adjustable Resolution," *IEEE Access*, vol. 6, pp. 26993-27000, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Jorge Ramírez-Beltrán et al., "Detection and Location of Background Leakage in Plastic Water Pipes Under a Noisy Environment," *Electronic, Automatic and Communications Engineering*, vol. 40, no. 3, pp. 1-15, 2019. [Publisher Link]

[7] Jessica Bohorquez et al., "Leak Detection and Topology Identification in Pipelines Using Transient Fluids and Artificial Neural Networks," *Journal of Water Resources Planning and Management*, vol. 146, no. 6, pp. 1-11, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Edgar-Orlando Ladino-Moreno, César-Augusto García-Ubaque, and María-Camila García-Vaca, "Estimation of Leaks in Pressure Pipes for Drinking Water Systems Using Artificial Neural Networks and Epanet," *Revista Científica*, vol. 43, no. 1, pp. 2-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Xudong Fan, Xijin Zhang, and Xiong Yu, "Machine Learning Model and Strategy for Fast and Accurate Detection of Leaks in The Water Supply Network," *Journal of Infrastructure Preservation and Resilience*, vol. 2, no. 10, pp. 1-21, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Anh Hoang, Phuc Do, and Benoît Iung, "Energy Efficiency Performance-Based Prognostics for Aided Maintenance Decision-Making: Application to A Manufacturing Platform," *Journal of Cleaner Production*, vol. 142, pp. 2838-2857, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] Joshua J. Cummins et al., "Energy Conservation in Industrial Pneumatics: A State Model for Predicting Energy Savings Using a Novel Pneumatic Strain Energy Accumulator," *Applied Energy*, vol. 198, pp. 239-249, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[12] Kyle Abela, Paul Refalo, and Emmanuel Francalanza, "Analysis of Pneumatic Parameters to Identify Leakages and Faults on the Demand Side of a Compressed Air System," *Cleaner Engineering and Technology*, vol. 6, pp. 1-15, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Zhiwen Wang et al., "Facilitating Energy Monitoring and Fault Diagnosis of Pneumatic Cylinders with Exergy and Machine Learning," *International Journal of Fluid Power*, vol . 24, no. 4, pp. 643-682, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] A.S. Cid, and T. Correa, "Venturino: Analysis of Pressure Variation in A Venturi Tube Using Arduino and Pressure Sensor," *Brazilian Journal of Physics Teaching*, vol. 41, no. 3, pp. 1-7, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[15] Rui Gabriel Modesto de Souza, Bruno Melo Brentan, and Gustavo Meirelles Lima, "Optimal Architecture for Artificial Neural Networks as Pressure Estimator," *Brazilian Magazine of Water Resources*, vol. 26, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Lamine Thiaw et al., "Designing ANFIS with Self-Extraction of Rules," *2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation*, Madrid, Spain, pp. 44-50, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[17] Diana Carolina Bastos Guerrero, Mario Joaquin Illera Bustos, and Sergio Basilio Sepúlveda Mora, "Adaptive Neuro-Fuzzy Inference System (ANFIS) for Estimating Global Solar Radiation," *Research and Innovation in Engineering*, vol. 9, no. 1, pp. 34-49, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Mirko Stojčić, "Application of ANFIS Model in Road Traffic and Transportation: A Literature Review from 1993 to 2018," *Operational Research in Engineering Sciences: Theory and Applications*, vol. 1, no. 1, pp. 40-61, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] Chaouki Ghenai et al., "Short-Term Building Electrical Load Forecasting Using Adaptive Neuro-Fuzzy Inference System (ANFIS)," *Journal of Building Engineering*, vol. 52, 2022. [CrossRef] [Google Scholar] [Publisher Link]