Journal of Al-Azhar University Engineering Sector

Vol. 20, No. 77, October 2025, 1410 - 1423

AI TRAINING MODEL FOR WATER LEAKS PREDICTION IN UNDERGROUND NETWORK

Osama M. E. A. Wahba[1, *], Aeizaal A. Abdul Wahab[1], Ayman T. El-Faramawy[2]

[1]Department of Electrical Engineering, University Sains Malaysia (USM), Malaysia

[2]Earth and Space Science and Engineering, York University, Ontario, Canada

* Correspondence: o.wahba@gmail.com

Citation:

O. M. E. A. Wahba, A. A. Abdul Wahab, A. T. El-Faramawy" AI Training Model for Water Leaks Prediction in Underground Network", Journal of Al-Azhar University Engineering Sector, vol. 20 (77), pp. 1410-1423, 2025.

Received: 09 April 2025

Revised: 04 September 2025

Accepted: 06 October 2025

Doi: 10.21608/auej.2025.370367.1801

ABSTRACT

Water supply shortages pose a significant global challenge, necessitating either an increase in water production capacity or the optimization of water utility systems to minimize losses. Efficient management of water resources is critical to meeting the escalating demand for high-quality water. Meeting the growing demands for quality water resources requires strategic interventions by governments and private sector entities to transform water management into essential practices for preserving and controlling water storage and distribution facilities. This research provides a "Water Leaks Detection and Prediction model" created to enhance the efficiency of water distribution systems by identifying and mitigating leaks. The proposed model integrates advanced analytical techniques to achieve high accuracy while maintaining cost-effectiveness, offering a practical solution for sustainable water resource management.

KEYWORDS: Water Leak detection, Prediction model, Sustainable water management, Smart water networks, Data-driven water management.

نموذج تدريب الذكاء الاصطناعي للتنبؤ بتسربات المياه في الشبكة تحت الأرض

أسامه محمد عبد الرحمن وهبه [١,*], أزيل عزام عبد الوهاب[١], أيمن طه الفرماوى[٢]

[١] قسم الهندسة الكهربائية، جامعة سينس، ماليزيا

[٢] قسم علوم الارض والفضاء والهندسة، جامعة يورك، اونتاريو، كندا

*** البريد الالكتروني للباحث الرئيسي**: o.wahba@gmail.com

الملخص

يشكل نقص إمدادات المياه تحديا عالميا كبيرا، يستلزم إما زيادة القدرة الإنتاجية للمياه أو تحسين أنظمة مرافق المياه لتقليل الخسائر. تعد الإدارة الفعالة للموارد المائية أمرا بالغ الأهمية لتلبية الطلب المتزايد على المياه عالية الجودة. تتطلب تلبية الطلب المتزايد على الموارد المائية الجيدة للناس تدخلات استراتيجية وإجراءات من قبل الحكومات وكيانات القطاع الخاص لتحويل إدارة المياه إلى ممارسات أساسية للحفاظ على مرافق تخزين المياه وتوزيعها والتحكم فيها. يوفر هذا البحث "نموذجا

للكشف عن تسربات المياه والتنبؤ بها" تم إنشاؤه لتعزيز كفاءة أنظمة توزيع المياه من خلال تحديد التسريبات والتخفيف من حدتها. يدمج النموذج المقترح تقنيات تحليلية متقدمة لتحقيق دقة عالية مع الحفاظ على فعالية التكلفة ، مما يوفر حلا عمليا للإدارة المستدامة للموارد المائية.

الكلمات المفتاحية: الكشف عن تسرب الماء، نموذج التنبؤ، الإدارة المستدامة للمياه، شبكات المياه الذكية، إدارة المياه القائمة على البيانات.

1. INTRODUCTION

This research successfully developed an improved and more efficient model for predicting and detecting water leakage in underground networks. By applying specific leakage factors to an Artificial Neural Network (ANN), the model enhances predictive capabilities and leakage probability assessments. The paper details the process, including data acquisition, training, and validation, demonstrating the model's effectiveness for monitoring and managing water systems.

2. ANN TRAINING MODEL FOR WATER LEAK PREDICTION

The prediction model's Artificial Neural Network (ANN) is mathematically represented using a neuron model with four input classes (x_1 to x_4), where n = 4 corresponds to the number of input parameters. Each class comprises numerous inputs utilized during training. Associated weights (w_1 to w_4) are assigned to each input, and the weighted sum of these inputs is computed to determine the activation function, as formally expressed in **Equation 1**.

The neural network architecture comprises highly interconnected processing elements (neurons), with interconnections characterized by their synaptic weights [1]. A fundamental neuron model incorporates multiple weighted inputs. The magnitude of neuronal output is regulated by an activation function, for which a logistic sigmoid function is employed:

$$\varphi (v) = [1 + \exp (-av)]\text{-}1 \quad (1)$$

The variable “a” stands for the slope of the sigmoid function, and v is the actual input.

As per the basic neuron model, “bk,” is an external bias that is added to raise or reduce the input of the activation function. During neural network training, optimal values for both biases and weights are identified to enhance the model's performance.

This structure for neuron k can be interpreted as follows:

$$v_k = \sum_{j=0}^{m} w_{kj} x_j \quad (2)$$

$$Y_k = \varphi(v_k) \quad (3)$$

where Y_k is the output signal and $x_0, x_1, x_2, \ldots, x_m$ are the input signals, $wk_0, wk_1, wk_2, \ldots, wk_m$ are the weights of neuron k, and v_k is the activation potential of neuron k.

Table 1 presents the initialization of the neural network prediction model, with weights assigned arbitrarily on a scale of 1 to 10 for each input neuron. These weights correspond to the key input parameters: Pressure, Flow Rate, Temperature, and Noise. Weight initialization constitutes a critical step in the backpropagation error process employed in this study, informed by experiential data obtained from field simulations.

Table 1: The weights of input Neurons in the prediction model represent each data type: Pressure, Flow rate, Temperature, and Noise.

#	Input Neuron	Voting criteria	Initial Weight
1.	Pressure	The continuous drop in pressure below the minimum setpoint	8
2.	Flow rate	Continuous increase in inflow to exceed the maximum setpoint	4
3.	Temperature	Continuous increase in temperature exceeding the maximum setpoint	3
4.	Noise	Consistent noise (no cut-offs) in at least two loggers reaching a level of more than ten decibels	6

Table 2 shows the weights of each data input value, and the other parameters related to the thresholds and required output based on the field's experiments during the leak simulation testing. For the prediction model, the input neurons' data were then scaled from "1" to "10" (**Table 1**), where:

x = the scaled neuron input

s = the actual neuron input

h = 10 (the scale)

z = the actual maximum input

y = the output

These weights were used to design the neuron layout used in calculating the activation function in the prediction model.

Table 2: The activation function calculation for the data types: Pressure, Flow rate, Temperature, and Noise.

	Pressure input neuron	Temperature input neuron	Flow-Rate input neuron	Noise input neuron
Min.	0.5	4.5	1	0
Max. (z)	45.5	142.4	156	63
Input	$x = \left(s \times \frac{h}{z}\right)$	$x = \left(s \times \frac{h}{z}\right)$	$x = \left(s \times \frac{h}{z}\right)$	$x = \left(s \times \frac{h}{z}\right)$
Labeling Rule (for Leak)	< 35	>12	>150	>10
Scaled Threshold	< 7.69	> 0.84	> 9.62	> 1.59
Initial Weight	8	4	3	6
Desired Output	1 (Leak) if ALL rules are met			
	0 (No Leak) if ANY rule is not met			

The values in the "Labeling Rule" row of **Table 2** represent the pre-defined expert rules based on field experience used to label the training data. These rules (Pressure < 35, Flowrate > 150, etc.) were applied to the raw sensor data to automatically generate the true binary label ('Leak' or 'No Leak') for each of the historical cases in the dataset. This labeled data is what allows the ANN to perform supervised learning. The actual desired output of the ANN model itself is a binary classification:

y = 1 (Leak Predicted) or y = 0 (No Leak Predicted)

The model's performance is judged by how well its predictions match these labels.

The sensor's real-time data collection source for pressure, flow rate, temperature, and noise was obtained under a research collaboration agreement with the Metropolitan Water Board. The specific

pipeline segments are part of a classified municipal network, and their exact location is protected by information due to critical infrastructure security protocols.

2.1 ANN Training using NLM

Artificial neural networks (ANNs) are structured variably based on neuronal organization and learning algorithms using the Neural Language Model (NLM). Architectures are broadly categorized as feedforward, single-layer, multilayer feedforward, or recurrent networks [2]. A single-layer network comprises an input layer connected directly to an output layer, where computations occur. In contrast, multilayer feedforward networks incorporate hidden layers that enable the extraction of higher-order statistics [3].

The model of a three-layer feedforward network that is fully linked, referred to as 4-6-1 (four input neurons, six hidden neurons, and one output neuron). The four input neurons are labeled from "1" to "4", where n = 4 represents the actual information in the Water Leaks Detection and Prediction model. The corresponding weights for these inputs are represented by w1, w2, w3, and w4.

The input neurons in the input layer (i.e., the initial layer) deliver the input vector to the second “hidden” layer. The output vector of the hidden layer is then used as input to the third layer (i.e., the output layer), which delivers the final solution of the network.

According to the feedforward neural network, the output signal at a neuron *j* (either an output node or a hidden neuron) is interpreted as follows:

$$Y_j(n) = \varphi(v_j(n)) \tag{4}$$

where:

$v_j(n)$ is the activation potential of neuron *j*, which is calculated as follows:

$$v_j(n) = \sum_{j=0}^{m} \left(w_{ji}(n) y_i(n)\right) \tag{5}$$

where:

m is the total number of inputs (without the bias) applied to neuron *j.*

$w_{ji}(n)$ represents the weight connecting the output of neuron *i* to the input of neuron *j* at iteration *n* (n^{th} training example).

$y_i(n)$ is the output signal of neuron *i* (i.e., represents the input signal of neuron *j*).

It should be clear that $y_i(n) = x_i(n)$, the i^{th} element in the input vector if neuron ‘*j*’ is in the first hidden layer.

In other words,

- The SUM will use the input layer vector *x* values from the input layer vector to the “hidden layer” vector.
- The SUM will use the hidden layer vector *y* values from the “hidden layer” vector to the output layer vector.

2.2 The learning Process using a nonlinear backpropagation algorithm.

Neural networks solve complex problems through training and subsequent generalization to new inputs. This process involves the iterative adjustment of free parameter weights and biases until optimal values are achieved. Among various learning algorithms, backpropagation is the most prevalent for feedforward networks [4]. This study employs the Levenberg–Marquardt algorithm, which, despite higher memory demands, provides superior training speed and accuracy [5-7].

According to the backpropagation procedure, the output signal of a neuron *j, $y_j(n)$,* is to be compared with a desired (i.e., target) output, $d_j(n)$. The error signal at the output of neuron *j,* $e_j(n)$, can be defined:

$$e_j(n) = d_j(n) - y_j(n) \quad (6)$$

Here, *n* stands for the n^{th} training sample (i.e., the n^{th} pattern). The goal of iterative adjustments is to have $y_j(n)$ as near as feasible to $d_j(n)$, which can be accomplished by reducing the cost function (total instantaneous error energy over all neurons in the output layer) that is specified as follows:

$$E(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n) \quad (7)$$

where *c* represents all neurons of the output layer.

2.3 Stopping Criterion (error function)

The weights of the neural network are continuously adjusted through iterative training, with each new epoch of training data. However, there is no exact measure to determine the appropriate point at which to stop training, or when the backpropagation algorithm has sufficiently converged. If training is not appropriately terminated, there is a potential for overfitting the training data.

Optimal neural network performance requires architectural flexibility and precise control over generalization. Conventionally, early stopping prevents overfitting by halting training at the minimum validation Mean Squared Error (MSE). This study introduces a systematic solution using a dedicated water leaks test dataset to evaluate generalization performance at the vertex of the performance curve, the minimum of the parabolic error trajectory before overfitting begins [8]. This method robustly identifies the critical inflection point where error increases, thereby optimizing the trade-off between model complexity and predictive accuracy.

Employing a stopped minimization procedure (**Fig. 1**), training is halted at an optimal point to prevent the network from learning high-frequency noise, a known contributor to overfitting during iterative gradient-based optimization. The stopping point is determined by continuously monitoring generalization errors, which decrease initially, reach a minimum, and then increase due to overtraining, while training error decreases monotonically [9]. Termination at the generalization error minimum ensures an optimal balance between learning the underlying data distribution and avoiding noise overfitting [10].

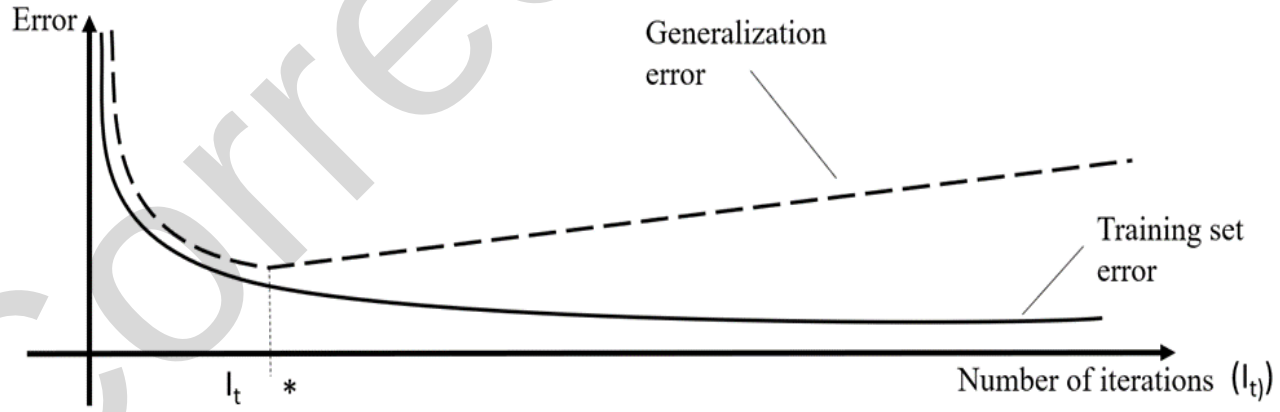

Fig. 1: Typical learning and validation curves [1]

2.4 Performance Evaluation

The generalization (validation) error is usually estimated as the Mean-Squared error (MSE) on a separate validation (test) set of data. This validation error is used to stop the training process at the optimal iteration value. Ideally, this validation set should be independent and uncorrelated with the data used for training to get an unbiased estimate of generalization performance [11]. The MSE error and the Model Correlation Value "R" can be estimated as:

$$MSE = \frac{\sum_{j=1}^{m}(d_j - y_j)^2}{l} \tag{8}$$

$$R = \frac{\sum_{j=1}^{l}(d_j - \overline{d_j}) \times (y_j - \overline{y_j})}{\sqrt{\sum_{j=1}^{l}(d_j - \overline{d_j})^2}\sqrt{\sum_{j=1}^{l}(y_j - \overline{y_j})^2}} \tag{9}$$

where l is the number of samples for the validation data set, d_j is the desired value, $\overline{d_j}$ is the meaning of the desired values in the data set, y_j is the model output value, and $\overline{y_j}$; is the meaning of the model output values in the data set. The Mean Squared Error (MSE) can also be computed for both training and testing datasets, where l represents the number of water network samples in the respective dataset [12].

2.5 Model Evaluation, Assessment, and Analysis

This study employs specific water leakage factors as inputs to an Artificial Neural Network (ANN) methodology to develop an enhanced model for leakage prediction and detection. The resulting ANN technique improves leakage probability prediction in underground water networks and augments the system's predictive capabilities.

The prediction model employs a neural network initialized with arbitrary weights and trained on inputs from a detection database. For each data class, the system evaluates the output and adjusts the corresponding weights. Inputs within predefined class thresholds yield an output of "one"; otherwise, "zero" is produced. This supervised learning approach uses a training set of seventeen leak occurrences from a multi-vote detection model. The backpropagation algorithm evaluates network status and adjusts weights by processing these inputs and outputs.

As explained in Section 2, the activation function for the specifically selected data types-Pressure, Flow rate, Temperature, and Noise — was used to calculate the scaled neuron inputs for the prediction model's initial data. The initial values for weights and the desired output $x_1, x_2, x_3,$ and x_4 were calculated for these four data types, as was explained in **Table 2**, as follows:

For pressure:

$$x_1 = s \times \frac{h}{z} = s \times \frac{10}{45.5}$$

For temperature:

$$x_2 = s \times \frac{h}{z} = s \times \frac{10}{142.4}$$

For flow rate:

$$x_3 = s \times \frac{h}{z} = s \times \frac{10}{156}$$

For noise:

$$x_4 = s \times \frac{h}{z} = s \times \frac{10}{63}$$

$w_1 = 8$ $w_2 = 4$ $w_3 = 3$ $w_4 = 6$

$y \geq 100$ (assumed desired output)

Accordingly, the scaled measured input dataset was created for each of the seventeen instances for all four input neurons, as shown in **Fig. 2**.

	A	B	C	D	E	F	G	H	I	K	L	M	N	O
1											The measured input layer required for the nonlinear regression model			
2	ReadingLocationCode	ID		ReadingDate	ReadingTime	P	T	Q	N		P	T	Q	N
3	RL1		3068	5/12/2018	12:21:04 PM	10	13	154	63		2.20	0.89	9.87	10.00
4	RL1		3069	5/12/2018	12:26:04 PM	10	13	151	63		2.20	0.89	9.68	10.00
5	RL1		3071	5/12/2018	12:36:04 PM	10	13	156	63		2.20	0.89	10.00	10.00
6	RL1		3073	5/12/2018	12:46:04 PM	8	13	152	63		1.76	0.89	9.74	10.00
7	RL1		3074	5/12/2018	12:51:04 PM	8	13	151	63		1.76	0.89	9.68	10.00
8	RL1		3075	5/12/2018	12:56:04 PM	8	13	151	63		1.76	0.89	9.68	10.00
9	RL1		3082	5/12/2018	1:31:04 PM	8	13	153	63		1.76	0.89	9.81	10.00
10	RL1		3105	5/12/2018	3:26:04 PM	3	13	155	63		0.66	0.89	9.94	10.00
11	RL1		3208	5/13/2018	12:01:04 AM	20	13	151	30.5		4.40	0.89	9.68	4.84
12	RL1		3209	5/13/2018	12:06:04 AM	20	13	151	30.5		4.40	0.89	9.68	4.84
13	RL1		3210	5/13/2018	12:11:04 AM	25.5	13	151	30.5		5.60	0.89	9.68	4.84
14	RL1		3211	5/13/2018	12:16:04 AM	25.5	13	151	30.5		5.60	0.89	9.68	4.84
15	RL1		3212	5/13/2018	12:21:04 AM	26.7	13	151	30.5		5.87	0.89	9.68	4.84
16	RL1		3213	5/13/2018	12:26:04 AM	30.4	13	151	30.5		6.68	0.89	9.68	4.84
17	RL1		3214	5/13/2018	12:31:04 AM	31.2	13	151	30.5		6.86	0.89	9.68	4.84
18	RL1		3215	5/13/2018	12:36:04 AM	33	13	151	30.5		7.25	0.89	9.68	4.84
19	RL1		3216	5/13/2018	12:41:04 AM	34	13	151	30.5		7.47	0.89	9.68	4.84

Fig. 2: The generation of the measured input layer required for the nonlinear model (P: Pressure, T: Temperature, Q: Flow rate, and N: Noise level)

The model computes the seventeen input dataset readings by incorporating each input neuron's weight via the activation function (**Equation 2**). **Fig. 3** presents the initial Artificial Neural Network training dataset, with the error representing the discrepancy between desired and computed outputs highlighted in red. The circled data point corresponds to an experimentally identified "less leak severity" condition. These three error points (values < 100) resulting from the activation function's execution across seventeen instances of the four input neurons highlight discrepancies between computed and expected outputs. These errors are utilized in the backpropagation algorithm to refine weights and improve model accuracy.

IUMS - Data Training for Artificial Neural Network

Inputs					Weights					Output		
Bias	Pressure	Temperature	Flowrate	Noise						Activation	Threshold check	
X0	X1	X1	X3	X4	W0	W1	W2	W3	W4	net	Calculated Output (Y1)	Desired output
1	2.20	0.89	9.87	10.00	1	8	4	3	6	111.77	1	1
1	2.20	0.89	9.68	10.00						111.19	1	1
1	2.20	0.89	10.00	10.00						112.15	1	1
1	1.76	0.89	9.74	10.00						107.86	1	1
1	1.76	0.89	9.68	10.00						107.67	1	1
1	1.76	0.89	9.68	10.00						107.67	1	1
1	1.76	0.89	9.81	10.00						108.06	1	1
1	0.66	0.89	9.94	10.00						99.65	0	1
1	5.87	0.89	9.68	4.84						109.60	1	1
1	7.25	0.89	9.68	4.84						120.68	1	1
1	5.60	0.89	9.68	4.84						107.49	1	1
1	5.60	0.89	9.68	4.84						107.49	1	1
1	4.40	0.89	9.68	4.84						97.82	0	1
1	6.68	0.89	9.68	4.84						116.10	1	1
1	6.86	0.89	9.68	4.84						117.51	1	1
1	4.40	0.89	9.68	4.84						97.82	0	1
1	7.47	0.89	9.68	4.84						122.43	1	1

Fig. 3: The Artificial Neural Network's data initial Training dataset for the prediction model

The same seventeen instances will be used in subsequent nonlinear neural network development. As outlined in Section 2.3, the stopped minimization procedure is employed since iterative gradient-based training causes networks to learn mapping components by frequency while training error decreases. Generalization error, however, reaches a minimum before increasing during overtraining, as previously demonstrated in Section 2.3, **Fig. 1**.

For the nonlinear regression training, the training model needs to be initialized with three sets of data; those are:

a) The Training Data set
b) The Validation Data set, and
c) The Testing Data Set

Table 3 shows the Input and Output data sets. The four sets of data for Pressure, Temperature, Flow rates, and Noise levels will constitute the INPUT to the nonlinear training. The OUTPUT of the non-linear training model was extracted from the experimental scaled data, targeting the less-severe leak data points. These two sets of data (INPUT and OUTPUT) are then rearranged in a Transpose matrix for each of them.

Table 3: The transposed matrices of the two data sets for INPUT and DESIRED OUTPUT (the three highlighted cells are the same three less-severity leak data as extracted from Fig. 3)

INPUT																
2.2	2.2	2.2	1.76	1.76	1.76	1.76	0.66	5.87	7.25	5.6	5.6	4.4	6.68	6.86	4.4	7.47
0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
9.87	9.68	10	9.74	9.68	9.68	9.81	9.94	9.68	9.68	9.68	9.68	9.68	9.68	9.68	9.68	9.68
10	10	10	10	10	10	10	10	4.84	4.84	4.84	4.84	4.84	4.84	4.84	4.84	4.84
OUTPUT																
111.77	111.19	112.15	107.86	107.67	107.67	108.06	99.65	109.6	120.68	107.49	107.49	97.82	116.1	117.51	97.82	122.43

The software application serves as a training tool to simulate the model under multiple scenarios. Seventeen data points for INPUT and OUTPUT were stored in text files to execute the training program. Each scenario incorporates a variable number of hidden layers, with iterative testing conducted for each architectural variation. Training continues until the calculated output minimizes discrepancies in low-severity leak data points relative to the desired output. The desired output is derived through empirical optimization of the four input neurons' weights, ensuring alignment between predictions and target outcomes through iterative refinement.

The data employed for model training is partitioned into three distinct datasets: (1) the training dataset, used to fit the model; (2) the validation dataset, which provides an unbiased evaluation for hyperparameter tuning during training; and (3) the test dataset, utilized for the final unbiased evaluation of the trained model.

The seventeen data samples used for the training model are categorized accordingly as follows into three data sets:

- Eleven samples for the Training data (practically selected as 70% of the dataset)
- Three samples for the Validation data set (practically selected as 15% of the dataset)
- Three samples for the Testing data set (practically selected as 15% of the dataset)

Various neural network architectures were evaluated under different scenarios by adjusting the number of hidden-layer neurons. **Tables 4, 5, 6 and 7** present the Sample Training Structure (STS), for multiple re-training cycles aimed at minimizing the Mean Squared Error (MSE) and maximizing the Model Correlation Value (R). The (4-10-1) set, achieving an MSE value of "3.804939267" and a correlation value (R) of "0.991539', was selected based on predefined criteria and is highlighted in light green.

The training process extended to additional structures (4-8-1), (4-6-1), and (4-4-1), each undergoing similar retraining cycles to identify the lowest MSE and highest R, with optimal results for each structure indicated in green.

Table 4: Training Structures (4-10-1) show the multiple re-training cycles approach to picking up the Lowest (MSE) and the Highest (R) as highlighted in light green color.

4-10-1	Mean Square Error (MSE)	Model Correlation Value (R)
	Multiple re-training cycles to pick up the Lowest	Multiple re-training cycles to pick up the Highest
Training	0.344125	0.99855
Validation	85.51545	-0.891184
Testing	0.3995	0.996049
Overall	28.753025	0.367805
Training	5.50768E-29	1.00E+00
Validation	0.0956378	0.999821
Testing	11.31918	0.974796
Overall	3.804939267	0.991539
Training	0.385135	0.997741
Validation	1.03004	0.976257
Testing	18.54105	0.835636
Overall	6.652075	0.936544

Table 5: Training Structures (4-8-1) show the multiple re-training cycles approach to picking up the Lowest (MSE) and the Highest (R) as highlighted in light green color.

4-8-1	Mean Square Error (MSE)	Model Correlation Value (R)
	Multiple re-training cycles to pick up the Lowest	Multiple re-training cycles to pick up the Highest
Training	8.04E-05	1.00E+00
Validation	8.28E-03	1.00E+00
Testing	7.11E+01	1.00E+00
Overall	23.71556461	0.999893
Training	5.82E-08	1.00E+00
Validation	1.69E-03	1.00E+00
Testing	2.30E-02	1.00E+00
Overall	0.008219639	0.999996667
Training	2.16E+00	9.94E-01
Validation	3.63E-01	-1.00E+00
Testing	9.62E+00	8.00E-01
Overall	4.047189667	0.264818667

Table 6: Training Structures (4-6-1) show the multiple re-training cycles approach to picking up the Lowest (MSE) and the Highest (R) as highlighted in light green color.

4-6-1	Mean Square Error (MSE)	Model Correlation Value (R)
	Multiple re-training cycles to pick up the Lowest	Multiple re-training cycles to pick up the Highest
Training	1.09E-03	1.00E+00
Validation	3.62E+01	9.68E-01
Testing	1.97E-01	1.00E+00
Overall	12.13211594	0.989332
Training	2.47E-25	1.00E+00
Validation	1.31E-03	1.00E+00
Testing	1.41E+01	9.17E-01
Overall	4.70411492	0.9723373
Training	9.22E-04	1.00E+00
Validation	2.36E+00	9.55E-01
Testing	4.02E+00	9.66E-01
Overall	2.126760661	0.973669

Table 7: Training Structures (4-4-1) show the multiple re-training cycles approach to picking up the Lowest (MSE) and the Highest (R) as highlighted in light green color.

4-4-1	Mean Square Error (MSE)	Model Correlation Value (R)
	Multiple re-training cycles to pick up the Lowest	Multiple re-training cycles to pick up the Highest
Training	3.05E-06	1.00E+00
Validation	2.61E-05	1.00E+00
Testing	7.83E-03	1.00E+00
Overall	0.002619842	0.999999
Training	8.04E+00	9.09E-01
Validation	2.39E+01	9.80E-01
Testing	3.92E-01	9.99E-01
Overall	10.7640293	0.96235
Training	1.84E-19	1.00E+00
Validation	4.28E+00	9.84E-01
Testing	4.41E+00	-7.45E-01
Overall	2.89634333	0.4129636

Based on the results above, the 4-4-1 training structure yielded optimal MSE (0.002619842) and R (0.999999) values, as highlighted in light green. Each structure was trained three times until performance converged.

Table 8 provides a comparative analysis of neural network architectures evaluated by mean squared error (MSE), correlation coefficient (R), and predictive capability for low-severity water leak instances. Among the tested architectures (4-10-1, 4-8-1, 4-6-1, 4-4-1), the 4-4-1 structure demonstrated superior performance, achieving the lowest MSE and highest R value, as shown in **Table 4 (a, b, c, and d)**, confirming its optimal balance of accuracy and predictive efficacy.

Table 8: The comparison between the performance of different neural network structures is based on the estimated mean square error (MSE) and correlation value (R) (Based on the model of eleven for Training, three for Validation, and three for Testing).

Neural Network Structure		**4-10-1**	**4-8-1**	**4-6-1**	**4-4-1**
Mean Square Error (MSE) Lowest	Training	5.50768E-29	5.82E-08	9.22E-04	3.05E-06
	Validation	0.0956378	1.69E-03	2.36E+00	2.61E-05
	Testing	11.31918	2.30E-02	4.02E+00	7.83E-03
	Overall	3.804939267	0.008219639	2.126760661	0.002619842
Model Correlation Value (R) Highest	Training	1.00E+00	1.00E+00	1.00E+00	1.00E+00
	Validation	0.999821	1.00E+00	9.68E-01	1.00E+00
	Testing	0.974796	1.00E+00	1.00E+00	1.00E+00
	Overall	0.991539	0.999996667	0.989332	0.999999
Water Leak Prediction Capability (all Levenberg leak points predicted?)		Yes	Yes	Yes	Yes

The training algorithm is the Levenberg-Marquardt, which requires more memory but less execution time, and higher accuracy was achieved, as mentioned in section 2.2. The initial random weight of "10" for a start was selected, the seventeen-training sample was initialized for the training, validation, and testing data sets as eleven for "Training", three for "Validation", and three for "Testing". The repetition of multiple training cycles for each of the four Neural Network Structures used in the model is highlighted with a total of four hidden neurons.

As was highlighted in **Fig. 1**, the training automatically stops when the mean square error (MSE) of the validation dataset is achieved, where the MSE error function (generalization curve) shows a parabola curve. **Fig. 4** shows the training regression (Plotregression) validation stop of the performance MSE error for the structure (4-10-1). The extraction of predicted leak points in YData matched the same position in XData.

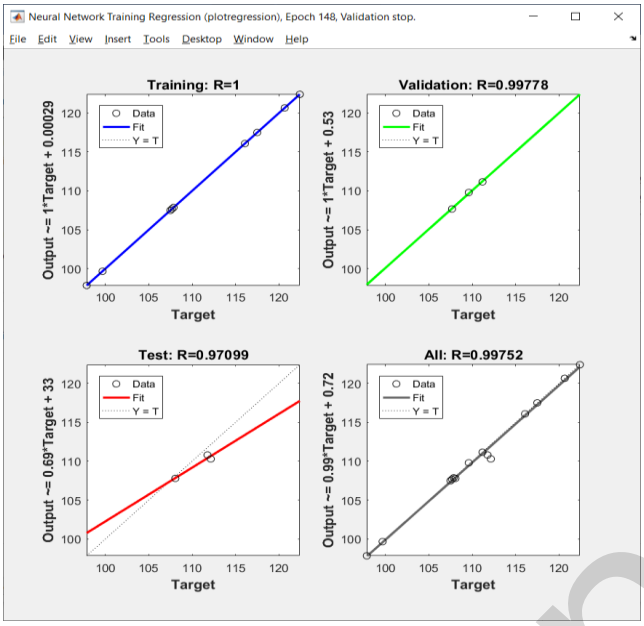

Fig. 4: The training regression (Plotregression) validation stop of the performance MSE error for the structure (4-10-1)

As shown in **Fig. 5**, the epoch (or iteration number) must be monitored, as it is integral to updating the network's weights. In this study, the training process was halted at epoch 236 (iteration 236), where the validation curve exhibited a parabolic trend. The primary objective of the training was achieved by determining the optimal weight values through the backpropagation algorithm. The training was halted at the minimum mean squared error (MSE) of the validation dataset, as indicated by the green line, ensuring optimal generalization performance and preventing overfitting.

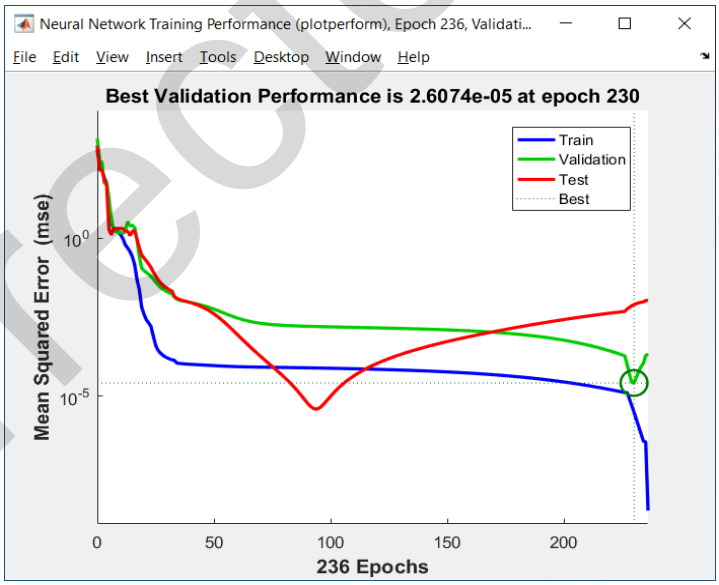

Fig. 5: The backpropagation calculated the correct weights, and the training stopped at the lowest MSE of the validation dataset (green line) at Epochs 236.

3. DISCUSSION

The primary conclusion of this study is that the 4-4-1 ANN model achieved 100% prediction accuracy on the validation dataset. Here are some key points:

- The proposed ANN model demonstrated superior performance (MSE = 0.0026, R = 0.999) compared to traditional methods
- Pressure and noise were identified as the most significant predictors of a leak event.
- The model successfully identified all 17 known leak events in the dataset.
- Our findings indicated that noise is a critical indicator in leak detection.
- The achieved high accuracy is potentially due to our use of the Levenberg-Marquardt algorithm and a focused set of input parameters.

Implications and Limitations:

- Implications: These results suggest that water utilities can implement such ANN-based models for early leak detection, potentially reducing water loss and infrastructure damage.
- Limitations: It is important to note that this study was conducted on a limited dataset (n=17). While the results are promising, further validation on a larger and more diverse dataset is necessary to confirm generalizability.

CONCLUSION

This study developed an artificial neural network (ANN) training methodology using backpropagation to construct a water leakage prediction model. The 4-4-1 ANN architecture achieved optimal predictive accuracy, evidenced by a mean squared error (MSE) of 0.002619842 and a correlation coefficient (R) of 0.999999, indicating 100% prediction success and demonstrating high effectiveness for leakage detection.

The network was initialized with random weights scaled from 1 to 10. Input data were forwarded and compared to desired outputs, with deviations computed as an error metric. This error was backpropagated to adjust neuronal weights proportionally to the error magnitude and type, refining model parameters iteratively until predicted outputs converged with ideal values. The supervised learning approach utilized predefined desired outputs, calculated from threshold bounds associated with the four input neurons. Training concluded upon achieving the minimum validation MSE (indicated by the green line), corresponding to optimal weights determined by backpropagation and a 100% success rate.

FURTHER RESEARCH AND DEVELOPMENT IN THE FUTURE

This paper analyzes current challenges and proposes a novel solution grounded in real-world conditions. While the research offers valuable contributions to the field, it also identifies critical questions and initiatives necessitating further investigation.

Based on the findings, designing an optimized water network is recommended to identify vulnerable locations representing potential leak weak points. These critical points should be determined at an early stage. The ANN model with a 4-4-1 architecture is recommended for water leak prediction, as demonstrated by the training results.

Additionally, the integration of Lean Engineering and Six Sigma methodologies is proposed to complement ANN training. This approach incorporates DMAIC (Define, Measure, Analyze, Improve, Control) capabilities, a data-driven improvement cycle that enables precise performance measurement, waste elimination, problem resolution, process enhancement, and outcome tracking. Another statistical data analysis tool is using ANOVA (Analysis of Variance), which is a statistical method used to compare the means of multiple data groups to determine if there is a statistically significant difference between them. Also, Pareto analysis can be used to focus on the biggest data variations. Data sampling strategies can be utilized to select which data observations to include in any measurement. After the data collection phase comes the analysis phase, in which the data analyst analyzes the collected data and makes a diagnosis. The goal is to determine the nature of the system's performance and the situation at hand.

CONFLICT OF INTEREST

The authors have no financial interest to declare in relation to the content of this article.

REFERENCES

[1] Haykin, S. (2009). Neural networks and learning machines (3rd ed.). Pearson Education. The textbook Neural Networks and Learning Machines by Haykin.

[2] Junaid Khan, Eunkyu Lee, Awatef Salem Balobaid, and Kyungsup Kim (2023). A Comprehensive Review of Conventional, Machine Learning, and Deep Learning Models for Groundwater Level (GWL) Forecasting, Applied Science, pp. 1-19.

[3] M. Aghashahi, L. Sela, and M.K. Banks (2023). Benchmarking dataset for leak detection and localization in water distribution systems, Elsevier Inc., pp. 2352-3409.

[4] Danilo Aparecido Carnevale Castillo and Marco Carminati (2023). A Pipe-Embeddable Impedance Sensor for Monitoring Water Leaks in Distribution Networks: Design and Validation, Sensors, pp. 1-27.

[5] Wei Hua and Qili Chen (2023). A Survey of Small Object Detection Based on Deep Learning in Aerial Images, Research Square, pp. 1-51.

[6] Abdul-Mugis Yussif, Haleh Sadeghi, and Tarek Zayed (2023). Application of Machine Learning for Leak Localization in Water Supply Networks, MDPI journal, pp. 1-21.

[7] Ildeberto Santos-Ruiz, Francisco-Ronay Lopez-Estrada, Vicenc¸ Puig, and Guillermo, and Valencia-Palomo (2023). Leak localization in water distribution networks using machine learning based on cosine features, ResearchGate publication-369381999, pp. 119.148.

[8] Wei Zeng, Nhu Do, Martin Lambert, Jinzhe Gong, Benjamin Cazzolat, and Mark Stephens (2023). Linear phase detector for detecting multiple leaks in water pipes, ELSEVIER, Applied Acoustics, volume 202, pp. 1-22.

[9] Hunsoo Song and Jinha Jung, Member (2023). An unsupervised, open-source workflow for 2D and 3D building mapping from airborne LiDAR data, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, pp. 1-18.

[10] Jiawei Chen, Pingbo Tang, Todd Rakstad, Michael Patrick, Xiran Zhou (2020). Augmenting a deep-learning algorithm with canal inspection knowledge for reliable water leak detection from multispectral satellite images, Elsevier, Advanced Engineering Informatics, pp. 1-14.

[11] Christopher Rausch, Chloe Edwards, Carl Haas (2020). “Benchmarking and Improving Dimensional Quality on Modular Construction Projects – A Case Study”, Vol. 1, No. 1, pp. 1- 20.

[12] Claudia Quintiliani, Ina Vertommen, Karel van Laarhoven, Joey van der Vliet, and Peter van Thienen (2020). Optimal Pressure Sensor Locations for Leak Detection in a Dutch Water Distribution Network, MDPI, Article in Environmental Sciences · September 2020, pp. 1-9.