AGU ADVANCING EARTH AND SPACE SCIENCES

Water Resources Research

RESEARCH ARTICLE
10.1029/2025WR041195

Topology-Aware Neural Networks for Abnormal Consumption Detection and Location in Water Distribution Networks

João Caetano[1], Nelson Carriço[1,2], Bruno Brentan[3], Andrea Menapace[4], and Dídia Covas[1]

[1]CERIS, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal, [2]Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, Lisboa, Portugal, [3]Hydraulic Engineering and Water Resources Department, School of Engineering, Federal University of Minas Gerais, Belo Horizonte, Brazil, [4]Institute for Renewable Energy, Eurac Research, Bozen-Bolzano, Italy

Key Points:

- Topology-aware neural networks use network topology and data to detect and locate abnormal water consumption
- The relationship between nodal consumption and pressure head drives the algorithm to identify unmeasured consumption
- Metamodels predict the location and approximate magnitude of sudden abnormal use based on user consumption and pressure head data

Correspondence to:

J. Caetano,
joao.caetano@hidrospetiva.pt

Citation:

Caetano, J., Carriço, N., Brentan, B., Menapace, A., & Covas, D. (2026). Topology-aware neural networks for abnormal consumption detection and location in water distribution networks. *Water Resources Research*, *62*, e2025WR041195. https://doi.org/10.1029/2025WR041195

Received 6 JUN 2025
Accepted 20 FEB 2026

Author Contributions:

Conceptualization: João Caetano, Nelson Carriço, Bruno Brentan, Dídia Covas
Data curation: João Caetano
Formal analysis: João Caetano
Funding acquisition: João Caetano, Dídia Covas
Investigation: João Caetano
Methodology: João Caetano, Nelson Carriço, Dídia Covas
Software: João Caetano
Supervision: Nelson Carriço, Bruno Brentan, Andrea Menapace, Dídia Covas
Validation: Nelson Carriço, Bruno Brentan, Andrea Menapace, Dídia Covas
Visualization: João Caetano
Writing – original draft: João Caetano

Abstract This paper presents a topology-aware neural network approach for the detection, location, and quantification of abnormal consumptions in water distribution networks. The approach includes two main steps: the optimization of pressure sensor locations to maximize measurement sensitivity and the development of metamodels based on near real-time data. The metamodel is designed and trained to predict the consumptions at all nodes based on pressure measurements and users' consumption collected by smart meters. These nodal consumptions deduced from the actual measured consumption allow the location of potential abnormal uses in the network. The proposed methodology enables the development of two metamodels, each tailored to specific applications based on the training data. The Static Metamodel relies on pressure head measurements under the assumption of constant nodal consumption, whereas the Dynamic Metamodel accounts for daily consumption variations, enabling the detection and location of abnormal consumption in real-world scenarios. Both metamodels can detect the location of abnormal consumptions with reasonable accuracy, although this accuracy strongly depends on the number and spatial distribution of sensors, as well as the magnitude and location of the abnormal consumption. As water utilities implement advanced metering systems, the application of the proposed approach becomes more viable, enabling more effective and faster abnormal consumption detection.

1. Introduction

The inevitable degradation of water distribution networks (WDN) leads to increased water loss (Barton et al., 2019). This increase can result from existing leaks or new burst events of varying magnitudes. Consumers typically report bursts when they become visible after pipe failures. However, small bursts can remain undetected for a long time and ultimately have consequences larger than those of large bursts (Giustolisi et al., 2024). Finding bursts at an early stage can save water and prevent small bursts from evolving into larger ones. However, the identification of bursts has proven challenging because even substantial events may not show any signs that allow their detection and location (Sophocleous et al., 2019). The paradigm is shifting, and the approach of repairing a burst only when it becomes visible is no longer a suitable option for water utilities, as water-loss management is now driven by increasingly stringent criteria. The widespread implementation of Supervisory Control and Data Acquisition (SCADA) systems, advanced metering infrastructures that automatically record and transmit water consumption at consumers, and distributed sensor networks that monitor pressure and flow rate within networks in near real-time constitute complementary and valuable sources of information that should be integrated into a single tool to allow for early burst detection. However, because monitoring is typically sparse, sensor data alone are insufficient to provide a complete and timely assessment of the current state of the systems (Bartos et al., 2024).

Physics-based models are widely used in the literature for burst detection and location. These models typically use hydraulic models calibrated for non-burst scenarios, where the burst location is determined by calibrating network consumption and comparing simulated results with field measurements; this problem is well known as the inverse problem (Romero-Ben et al., 2022). The rationale is that, in a well-monitored system, a burst creates a unique signature on the pressure head and flow rate data, which can be used to automatically determine the burst magnitude and location. The efficiency and effectiveness of the approach rely on minimizing the differences between the collected data and the model results; however, this is an expensive computational process due to the

Writing – review & editing: Nelson Carriço, Bruno Brentan, Andrea Menapace, Dídia Covas

large number of possible burst locations that must be evaluated. In addition, the calibration of deterministic physics-based models is challenging because it requires accounting for several uncertain parameters (e.g., nodal demands, pipe roughness, and length), and their integration with sparse measurement data is often complex and time-consuming, limiting their applicability for real-time burst detection (Zanfei et al., 2023). Despite their easy interpretability and strong physical consistency, these models remain computationally demanding and difficult to scale to real-time monitoring applications. Another common issue that is often ill-posed in the inverse problem is the non-uniqueness of the solution for a certain attained error. Even in well-monitored systems, multiple combinations of decision variables can produce equally fit solutions but inaccurate burst locations (Ferreira et al., 2025).

Alternatively to physics-based models, the literature presents various data-driven approaches for detecting bursts, ranging from statistical methods to advanced machine learning (ML) models (e.g., Mazzoni et al., 2024; Mounce et al., 2010; Wu & Liu, 2017). In the ML category, surrogate models are highly notable, allowing fast and automated processing of data from multiple sensors across the network. The near real-time assimilation of these data is essential for delivering critical insights into system performance with minimal latency (Bartos et al., 2024). Surrogate models, also known as metamodels, use data-driven function approximation techniques to empirically estimate the responses of high-fidelity models (e.g., EPANET Solver). These models essentially act as emulators of high-fidelity models, for estimating the hydraulic state of WDN with low computational costs while maintaining a reasonable level of accuracy (Razavi et al., 2012). Garzón et al. (2024) refer to the fact that metamodels can be applied in optimization frameworks, real-time applications, and uncertainty analysis. Broad et al. (2005) conducted trailblazing work in the field of WDN that used metamodels to integrate water quality concerns, which are typically computationally expensive, into an optimization framework to design and improve the operation of networks. Broad et al. (2010) used a metamodel to estimate the pressure head at critical nodes of the network, thereby optimizing pump operation. Meirelles et al. (2017) propose a calibration model for pipe roughness that uses a metamodel to estimate pressure at all nodes of the network. These early metamodels were developed using fully connected neural networks.

Garzón et al. (2022) introduced inductive biases in ML-based metamodels for urban water systems, enabling the models to learn more efficiently with less data. These biases simplify the architecture, improve data efficiency, and promote adaptability by leveraging the underlying structures and facilitating transfer learning across different domains. Three relevant inductive biases have been identified: temporal, physical, and relational. Temporal bias uses inputs and outputs from previous steps, whereas the physical bias adheres to physical laws. Relational bias establishes relationships between entities using graphs. Graph Neural Networks (GNN) process graph data, demonstrating stability to changes in topology and functioning as selectors that focus on significant relationships. This promotes adaptability and robustness across diverse input configurations and enables transfer learning to leverage knowledge across domains and case studies.

Several researchers have used inductive bias to enhance metamodels accuracy. Xing and Sela (2022) developed a physics-informed GNN that simultaneously assimilates data gathered from sensors and considers the violation of physical laws in unmonitored nodes of the network. An additional term is incorporated into the loss function to account for the flow rate imbalance at unmonitored nodes and pressure head imbalance at the sources (i.e., reservoirs and tanks). Kerimov et al. (2024) shifted the focus to edge-based GNN to pipe flow rate prediction, afterward deriving nodal pressures from the flow rates. The authors compared the potential for generalization and transferability of their model with that of Xing and Sela (2022). Truong et al. (2024) used a *Graph Attention Network* (GAT) architecture with residual connections for pressure estimation at network nodes. This research proposes training the model by randomly masking the pressure data of the network nodes, claiming that this improves the model's estimation.

Li et al. (2024) propose a *Gated Graph Neural Network* model, with temporal inductive bias, for real-time water quality prediction and, like Truong et al. (2024), employ a masking operation during training to enhance the model's predictions. Garzón et al. (2024) proposed a graph neural network metamodel based on the *Graph Isomorphism Network* architecture, applied to urban stormwater drainage. The results of this study explore the strength of the topological inductive bias in the development of metamodels compared to fully connected neural networks.

In the literature, the application of advanced ML for the detection and location of bursts assumes that only pressure data from a limited number of installed sensors are available to develop the metamodel, as they consider

consumption data at the network nodes as an uncertainty. The use of pressure sensors instead of other types of sensors (e.g., flow rate, smart meters at user points, and chlorine concentration) is due to their lower cost and ease of installation. Örn Garðarsson et al. (2022) trained two *ChebNet* models (Defferrard et al., 2016): one to predict the pressure at network nodes based on previous measurements based on a historical window and another to reconstruct pressure heads based on current measurements gathered at a limited number of pressure sensors within the network. The differences between the outputs of the two models represent a residual signal, which is statistically analyzed to identify changes and detect leaks. It is important to note that because topological information is incorporated into the model, burst locations can be determined. However, this method yields many false positives. Zhang and Fink (2024) use the same rationale to formulate the problem but enhance the accuracy of the models by applying a pre-training strategy. Because WDN operate as flow networks, the model was pretrained to emulate the Ford-Fulkerson algorithm, which is designed to solve max-flow problems. This algorithmic knowledge was then applied to improve the pressure estimation. The experimental results demonstrate that the proposed algorithm-informed approach achieves more successful results with better generalization ability than previous studies, such as Örn Garðarsson et al. (2022).

Daniel and Cominola (2023) employed a physical inductive bias and proposed a physics-informed ML algorithm that analyzes pressure data for burst detection. The algorithm allows the segregation of abnormal consumption caused by unmeasured consumptions (e.g., unauthorized use) from bursts. They highlighted that any abnormal consumption can affect pressure measurements, making it crucial to identify the specific type of abnormal consumption. However, regardless of the type of abnormal consumption, it is essential to detect and locate it, as this represents commercial loss.

Zhou et al. (2019) proposed a model based on the DenseNet architecture, which increases the number of skip connections and reuses the preceding feature maps. Truong et al. (2024) applied this approach in a simpler manner by adding a residual connection to the architecture. These approaches enable the implementation of deeper neural networks compared with traditional Multilayer Perceptron (MLP), facilitating gradient flow and mitigating the vanishing gradient problem. The model can accurately locate bursts in pipes within a small network based on real-time pressure heads gathered from a limited number of sensors.

However, deep learning-based models also face challenges that need to be addressed, such as high data requirements (Wu et al., 2024). For instance, leakage data and the hydraulic behavior response of the network to these events are often limited. Conversely, hydraulic simulators that integrate the conservation of mass and energy (Todini & Rossman, 2013) can be used to yield large amounts of data, creating a labeled and balanced data set. Combining hydraulic models with Graph Neural Network (GNN) based metamodels in a hybrid framework helps overcome the identified limitations. Menapace et al. (2020) proposed a stochastic hydraulic time-series generator to produce simulated data, including leakage, that more closely resemble real-world conditions, with data sets generated by randomizing uncertain parameters, such as roughness and demand, which increases their realism and provides reliable input for the metamodels.

This paper proposes a novel GNN-based methodology for the detection, location, and quantification of abnormal consumptions in WDN. This methodology includes two main steps: (a) optimization of pressure sensor locations and (b) development of metamodels to emulate burst scenarios. During the development stage, the metamodel was trained using pressure data from optimally located sensors alongside user consumption data from smart metering systems. In the application stage, these data are used to detect and locate abnormal consumption (i.e., bursts and unauthorized uses) in the application phase. Two metamodels were developed: a Static Metamodel, which relies on pressure head measurements under the assumption of constant nodal consumption (average network consumption), and a Dynamic Metamodel, which accounts for daily consumption variations. The results show that the developed metamodels can effectively detect, in near real-time, the location of the abnormal consumption with reasonable accuracy, although this accuracy strongly depends on the number of sensors, their spatial distribution, and the magnitude and location of the abnormal consumption. The time for the detection and location of abnormal consumption is not dependent on the algorithm time processor (metamodel), which is almost instantaneous, but rather on the time required for the data from users' smart metering systems and from the pressure sensors to be transferred and become available in the central server/computer where the metamodel is run.

2. Methodology

2.1. Water Distribution Network Description

A WDN can be topologically described and modeled as a planar graph G with n nodes and l links, where $N = \{1,\ldots,n\}$ and $L = \{1,\ldots,l\}$ denote the sets of elements (nodes and links) that describe the network. The set of nodes, N, can be divided into subsets according to the type of node and information they provide. In the current research, the following nomenclature is adopted: sources, such as storage tanks, are denoted by N_s; nodes equipped with pressure sensors and/or information gathered from smart metering systems are denoted by N_m; and topological nodes that do not have any associated information are denoted by N_t. The set of all nodes is described as $N = N_s \cup N_m \cup N_t$. The links that connect nodes in a WDN (i.e., pipes or valves) are denoted by $l = (i,j)$, where i and j represent the number of the start and end nodes, respectively.

Although all network components are included in the methodology, different functions are assigned to each type of node and link. Some nodes (i.e., N_t, N_s) are solely used to define the network topology, while others (N_m) serve a dual purpose: they contribute to the definition of the network topology and have attributes associated with consumption ($\mathbf{q}_u$) and/or pressure head ($\mathbf{H}$). In this work, no attributes were considered for the links, as they only serve to define the network topology. This choice is motivated by the fact that the metamodel employs a topology-aware attention mechanism based on GATv2 (Brody et al., 2021), which dynamically adjusts the weights of each link between nodes during the training process. In this mechanism, attention coefficients are computed for each pair of connected nodes using their feature representations as input. The coefficients are obtained through a shared linear transformation followed by a LeakyReLU activation and softmax normalization, allowing the model to learn the relative importance of each link without requiring explicit edge attributes.

This methodology assumes that the total consumption of the users is known, and any other unmeasured consumption is considered anomalous (e.g., bursts). In practice, users' consumption, even with fully implemented smart metering systems, does not account for all nodal outflows in the network. Typically, there are unknown variables that are not considered, which represent the terms associated with abnormal consumption, $\mathbf{q}_b$, including unmeasured consumption and bursts. Thus, the actual nodal outflow consists of the sum of the measured and abnormal consumptions components, as follows: $\mathbf{q} = \mathbf{q}_u + \mathbf{q}_b$.

The proposed methodology combines consumption data from users with pressure head data collected using a limited number of sensors installed within the network to estimate the abnormal consumptions at each node, $\mathbf{q}_b$. This approach involves two main steps. First, the optimal placement of the pressure sensors is performed. Second, a metamodel is developed to serve as a burst scenario emulator, enabling the detection and location of abnormal consumptions in near real-time with minimal computational cost (time and resources) during deployment. This is the main advantage of metamodeling. Once trained, the metamodel provides a response in fractions of a second. It is much faster than the detection and location of bursts using an optimization process for the inverse problem. The metamodel returns the total nodal outflow vector, $\mathbf{q}$, being the abnormal consumption vector, $\mathbf{q}_b$, calculated by deducing the measure consumption, $\mathbf{q}_u$. This process can be considered as a calibration of nodal outflows, where the estimated outflows from the metamodel are used to modify a hydraulic model, similar to the calibration process that adjusts a physics-based model. Although this study does not address the simultaneous occurrence of multiple abnormal consumptions (hereafter referred to as bursts), the underlying methodology for applying the metamodel remains applicable.

2.2. Optimal Location of Pressure Sensors

The optimal sensor configuration is determined using an optimization procedure that maximizes the sensitivity of the nodal pressure to variations in consumption, as shown by Zhou et al. (2019). This sensitivity is represented by the Jacobian matrix, as expressed in Equation 1:

$$\nabla_{\mathbf{q}}\mathbf{H} = \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \tag{1}$$

where $\nabla_{\mathbf{q}}\mathbf{H} \in \mathbb{R}^{|N|\times|N|}$ is the Jacobian matrix, which represents the sensitivities of the nodal pressure, $\mathbf{H}$, to variations in the nodal consumption, $\mathbf{q}$. Here, $\mathbf{H} \in \mathbb{R}^{|N|\times 1}$ is the vector of nodal pressure, and $\mathbf{q} \in \mathbb{R}^{|N|\times 1}$ is the

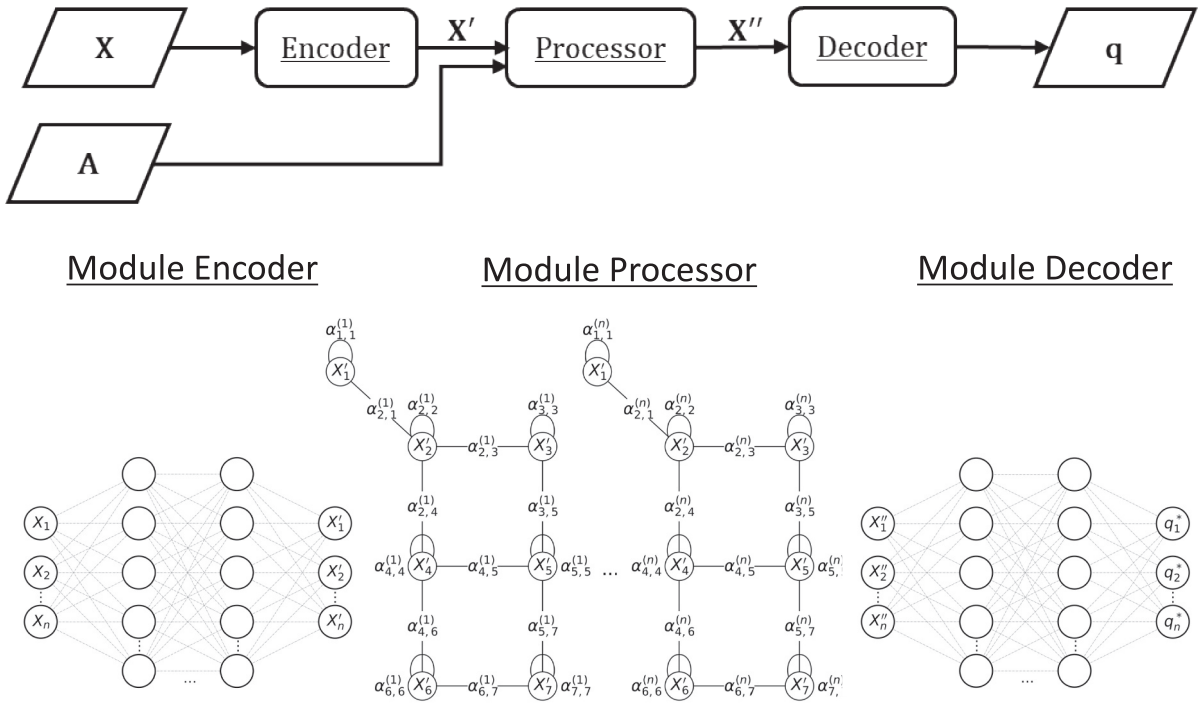

Figure 1. Overview of the metamodel architecture: Inputs, Encoder, Processor, Decoder, and Output.

vector of nodal consumption. The analytical determination of Jacobian matrix is performed using the matrix analysis method proposed by Nian-dong et al. (2017).

The optimal sensor configuration (i.e., set of the best location of sensors N_m) is obtained by fixing the number of sensors to be installed and maximizing the sum of the highest sensitivity of each network node, considering a specific sensor configuration. The optimization process is carried out using Genetic Algorithms (GA) available in the Pymoo library (Blank & Deb, 2020) and is formulated as a single-objective problem, as follows:

$$F(X) = \sum_{j=1}^{N} \max_{i \in N_m} |\nabla_{\mathbf{q}} \mathbf{H}|_{ij}, \ \forall\ j \in [1, N] \tag{2}$$

Several configurations are obtained, one for each fixed number of sensors installed. These configurations are used to develop the metamodel.

2.3. Metamodel Development

The metamodel is composed of three modules, according to the Encoder-Processor-Decoder architecture, as used by Garzón et al. (2024), Kerimov et al. (2024), and Xing and Sela (2022), as depicted in Figure 1. The Encoder receives $\mathbf{X}$, the normalized data from the smart metering systems and pressure sensors, as input. The Encoder transforms the input into a high-dimensional latent space representation, denoted as $\mathbf{X}'$, designed to capture the underlying patterns and structures of the data. The Processor receives the latent space representation, $\mathbf{X}'$, and the adjacency matrix, $\mathbf{A}$, which represents the topology of the WDN, and iteratively aggregates information from neighboring nodes to produce a new latent space representation, $\mathbf{X}''$. Finally, the Decoder receives $\mathbf{X}''$ from the processor's output and transforms this latent space representation into a variable with physical meaning, the total consumption, $\mathbf{q}$, including the users' consumptions, $\mathbf{q}_u$, and the leakage term, $\mathbf{q}_b$.

The modules consist of a multi-layer perceptron, which is used by both the Encoder and Decoder, while the Processor relies on a Graph Attention Network. These components are applied sequentially to enhance the model's capacity. Nodal consumptions and nodal pressure heads have different scales, requiring the normalization of the input data. To address this, the inputs to the metamodel, $\mathbf{q}_u$ and $\mathbf{H}$, are normalized using the z-norm,

resulting in values with a mean of 0 and a standard deviation of 1. This normalization technique is crucial for neural networks, because it ensures stable convergence during training and provides a balanced contribution from all variables. If the inputs have different magnitudes, they can affect gradient propagation during back-propagation, leading to issues such as vanishing and exploding gradients. The user consumption vector, $\mathbf{q}_u$, is normalized using the total consumption, $\mathbf{q}$, to produce $\mathbf{q}_u^*$. The pressure head gathered by sensors installed within a network is organized in the vector, $\mathbf{H}$, with the same length as $\mathbf{q}_u^*$, but with zero values in positions where no sensors are installed. Thus, $\mathbf{H}^*$ represents the normalized pressure head that incorporates the system response to burst events. After normalization, the two vectors are concatenated into a matrix, $\mathbf{X} = \left[\mathbf{q}_u^* \mid \mathbf{H}^*\right]$, which defines the input of the metamodel.

The Encoder transforms the input data into a more useful or convenient form for the next stage, that is, the processor. This encoding step helps to capture complex data relationships that may not be apparent. To address this, an MLP is employed to encode the features of the normalized input matrix $\mathbf{X} \in \mathbb{R}^{N\times 2}$, producing a non-linear representation, known as node embeddings, $\mathbf{X}' \in \mathbb{R}^{N\times \mathrm{d}}$, where d represents the expanded feature dimension. The module encoder can be described as:

$$\mathbf{X}' = \mathrm{MLP}_{\phi}(\mathbf{X}) \tag{3}$$

where $\mathbf{X} = \left[\mathbf{q}_u^* \mid \mathbf{H}^*\right]$ is the vector representing the normalized input data processed by the encoder module, MLP_{ϕ}, with trainable parameters ϕ, which are responsible for encoding the information from all nodes in the network.

The module processor is based on a Graph Neural Network (GNN), which uses a graph structure and node features to learn a specific node representation. The GNN follows a neighborhood aggregation strategy, where the model iteratively updates a node's representation by aggregating its neighbors' representation (Xu et al., 2018). In the current research, a GATResBlock proposed by Truong et al. (2024) builds upon the GAT (Veličković et al., 2017) is used as a Processor. The GAT incorporates the concept of attention to the GNN by assigning a coefficient to each neighbor, indicating the relevance of its features in updating the target node representation. Self-loops are included to account for the significance of the node. This process describes a single attention head. In multi-head attention, multiple attention heads are used in parallel, each focusing on different aspects of the neighbors' features. This approach enables the model to capture a more diverse and enriched graph representation of the data.

The GATResBlock uses two GAT layers to process information. The inner layer enhances the diversity of the attention heads, whereas the outer layer adjusts the features to give more weight to certain information. A squeezing technique consolidates the features for a consistent output, and a residual connection helps prevent overfitting. The attention coefficients are computed as proposed by Brody et al. (2021) due to the enhancement of expressivity and the dynamic attention approach. The GATResBlock is described as follows.

$$\mathrm{GATResBlock} = x_i + \frac{1}{|\mathcal{N}_1(i)+1|}\sum_{j\in\mathcal{N}(i)\cup\{i\}} \mathrm{GAT}(\mathrm{GAT}(x_j;\alpha,\Theta);\beta,\Psi) \tag{4}$$

The stacking of multiple GATResBlocks enables efficient information propagation across distant nodes. The number of stacking blocks is defined by the diameter, d, of the network under study, which refers to a metric from graph theory. In the context of a WDN, the diameter corresponds to the longest shortest path between any two nodes in the network, and therefore depends on the discretization of the WDN. This design ensures effective message propagation and accurate consumption value recovery in the graph representation. The generic composition of the GATResBlock is as follows:

$$\mathbf{X}'' = (\mathrm{GATResBlock}_{\phi_d} \circ \mathrm{GATResBlock}_{\phi_{d-1}} \circ \cdots \cdot \mathrm{GATResBlock}_{\phi_1})(\mathbf{X}') \tag{5}$$

Although ML models have limited interpretability, it is important to note that assigning different attention coefficients to the connections between network nodes aims to model the effect of water volume flowing from node to node within a water distribution network. Following the pipeline described in Figure 1, the output of the processor is forwarded to the next module, the decoder.

The module decoder is responsible for converting the latent space representation from the processor $\mathbf{X}''$ into a vector with a physical meaning. To address this, an MLP is employed to decode the features of the embeddings $\mathbf{X}'' \in \mathbb{R}^{N\times d}$, producing $\mathbf{q} \in \mathbb{R}^{N\times 1}$. The module decoder can be described as follows.

$$\mathbf{q}^* = \mathrm{MLP}_{\psi}(\mathbf{X}'') \tag{6}$$

The equation indicates that each processed latent space representation $\mathbf{X}''$ is decoded using an MLP with trainable parameters ψ. In brief, this step uses the high-dimensional values of the processor to estimate the value of consumptions at all network nodes, including the burst flow rate. The output of the decoder, $\mathbf{q}^*$, is unnormalized to vector, $\mathbf{q}$.

The Mean Squared Error (MSE) is the loss function used to train and test the metamodel.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(q_i - \hat{q}_i\right)^2 \tag{7}$$

where, q_i represents the ground truth value associated with node i and $\hat{q}_i$ represents the predicted value associated with node i. The MSE penalizes larger errors more severely than smaller errors, encouraging the metamodel to focus on reducing large deviations and improving the overall prediction accuracy.

The recommendation is to develop two metamodels. The first is a single-period simulation metamodel, referred as Static Metamodel, with one set of nodal consumptions (the 24-hr average value at each node) combined with several burst sizes and locations. This stage allows us to establish the problem formulation and develop the metamodel architecture. The second model is the extended-period simulation metamodel, referred as Dynamic Metamodel, with the same architecture as the static. This model is trained and validated on a data set with a finer temporal resolution, including the 24-hr consumption variations, which in turn leads to a larger data set.

2.4. Accuracy Metrics Calculation

During the test phase, the assessment of the accuracy of the metamodel is carried out by comparing the output vector of the metamodel, $\hat{\mathbf{q}}_b$, with the ground truth vector, $\mathbf{q}_b$. Two different metrics were computed: (a) the errors associated with the estimated magnitude of the burst and (b) the error associated with the estimated Euclidean distance between the predicted and ground truth location of the burst.

Before describing the first metric, it is necessary to explain the concept of a hop. In graph theory, a hop is the direct connection between two nodes. In the general case, k-hops two nodes are k-hops apart if a path passes through exactly k-edges (pipes). In other words, k-hop indicates the number of steps required to travel from one node to another, considering the k-edges (pipes) in the path.

This k-hop concept is relevant for assessing the accuracy of the burst magnitude because of how the processor handles information between the nodes of the WDN (i.e., message passing between nodes and updates). In areas with a higher concentration of nodes, there may be a tendency to spread the burst flow rate in nodes close to each other. Thus, the first metric is the error between the sum of the predicted burst magnitudes considering different values of k-hops (e.g., 0-hop, 1-hop, and 2-hop) and the ground truth, $\varepsilon_{q,k_{\mathrm{hops}}}$. For instance, if 0-hop is considered, the comparison is computed between the predicted and ground truth magnitudes of the node where the burst is located. However, if 1-hop is considered, the comparison is computed between the sum of the predicted magnitudes of the node's neighbors where the burst occurs and the ground truth value. The same rationale is applied to another value of k. Accordingly, the error associated with the estimated magnitude of the burst for each k-hop is expressed as follows:

$$\varepsilon_{q,k} = \left(\sum_{k=0}^{k_h}\sum_{j\in\mathcal{N}_k(i)}\hat{q}_{b_{j,k}}\right) - q_{b_i} \tag{8}$$

where k_h is the number of neighborhood depth layers (i.e., number of hops), $j \in \mathcal{N}_k(i)$ describes the neighbors of i considering the neighborhood depth layers k, $\hat{q}_{b_{i,k}}$ and q_{b_i} describes the burst flow rate predicted by the metamodel and the ground truth values, respectively.

The second metric is the Euclidean distance, D, between the centroid of the predicted burst and the true burst location. The centroid coordinates, x_0 and y_0, are given by the summation of the nodes' coordinates weighted by their estimated burst value, $\hat{q}_{b_i}$ concerning the total predicted burst flow rate, $\hat{q}_b = \sum_{i=1}^{N} \hat{q}_{b_i}$.

$$\varepsilon_x = \left[(x_0 - x_b)^2 + (y_0 - y_b)^2 \right]^{\frac{1}{2}} \tag{9}$$

$$x_0 = \frac{\sum_{i=1}^{N_j} x_i \cdot \hat{q}_{b_i}}{\sum_{i=1}^{N_j} \hat{q}_{b_i}} \tag{10}$$

$$y_0 = \frac{\sum_{i=1}^{N_j} y_i \cdot \hat{q}_{b_i}}{\sum_{i=1}^{N_j} \hat{q}_{b_i}} \tag{11}$$

where x_i and y_i, describe the coordinates of node i, and x_b and y_b describe the coordinates of the burst location.

The metamodels allow the detection of abnormal consumptions. The type of abnormal consumption (i.e., unauthorized use or burst) can be identified by analyzing its evolution over time: abnormal consumption related to unauthorized consumption has a short duration (a couple of hours), while a burst does not disappear until it is repaired. An eventual metric to evaluate the occurrence of a burst is the duration of the anomalous event: whenever it occurs over several consecutive timestamps (e.g., 3–4 times) it is identified as a burst.

2.5. Parameters' Sensitivity Analysis

The accuracy of the metamodel depends on the choice of the parameters. These parameters are of different natures: some are related to the architecture of the metamodel, whereas others control the training process and need to be defined during the training stage. A sensitivity analysis was performed for Static Metamodel. Due to the stochastic nature of the metamodel training process, this model was trained for many (ca. 50) combinations of parameters, and the combination with the highest coefficient of determination was identified and adopted for both metamodels. Table 1 summarizes the combinations of parameters used and those that led to a more accurate metamodel (in terms of the coefficient of determination, R^2). The bold values denote the optimal parameters that result in the best performance/accuracy for each model.

3. Case Study

3.1. Description of Case Study

The case study is one district metered area (DMA) of the VilaMoura WDN, located in the southern region of Portugal. This DMA was designed to meet minimum pressure constraints while minimizing costs (for more details, see Caetano et al. (2025)). The network is entirely gravity-fed, supplied by a single storage tank and spans 17 km, comprising 162 pipes and 151 nodes (150 junctions and one tank).

The DMA has a tree-shaped topology, which enhances the network's sensitivity to head losses caused by increased consumption due to bursts. The tree-shaped topology makes the case particularly suitable for demonstrating the proposed approach, which relies on the sensitivity of nodal pressure to variations in consumption. Additionally, the relatively small number of nodes and pipes results in a smaller training data set and shorter training times for the metamodel compared to larger networks, thereby reducing computational complexity.

It should be noted that the selection of this DMA, with its tree-shaped topology and relatively small size, provides ideal conditions for demonstrating the approach, but it also limits the direct generalization of the results to looped or oversized networks.

Water Resources Research

10.1029/2025WR041195

Table 1
Parameters Used for the Metamodel Development of Any Pressure Sensor Configuration

Group	Name	Description	Set of values
Metamodel	Number of hidden layers	Number of hidden layers in the MLPs used for the Encoder and Decoder	{0, 2, **4**, 5}
	Non-linearity	Activation function applied between layers to introduce non-linearity	{**ReLU**, LeakyReLU, PReLU}
	Hidden dimension	Number of units in each hidden layer of the Encoder and Decoder	{16, **32**, 64}
	Num. of GATRes blocks	Number of Graph Attention Network blocks for message passing and attention-based aggregation	{1, 5, 10, 15, **23**}
Hyperparameters	Batch size	Number of training samples processed together in a single forward and backward pass	{16, **32**, 64, 128}
	Learning rate	Scaling factor that determines the step size for updating weights during gradient descent	{0.01, **0.001**}
	Weight decay	Regularization coefficient that discourages large weights and biases to prevent overfitting	{**1e−6**}
	Scheduler	Adjusts the learning rate during training	{**ReduceLROnPlateau**, Cosine Annealing}
	Patience scheduler	Number of epochs to wait before reducing the learning rate if no improvement is observed	{**10**}
	Gradient clipping	Limits the magnitude of gradients to prevent unstable updates	{False, **True**}

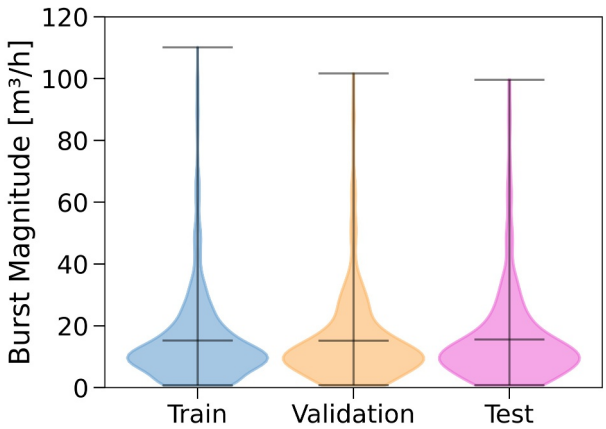

Figure 2. Distribution of burst magnitudes in data set 1 across the training, validation, and test sets.

3.2. Data Set Generation

Two data sets are generated: Data set 1 for the metamodel that uses a single time step to represent the data (Static Metamodel) and Data set 2 for the metamodel that simulates an extended period with multiple time steps (Dynamic Metamodel). For the development of the metamodels, it is assumed that users'consumptions and pressure measurements occur at simultaneously on an hourly basis, despite pressure data having a finer temporal resolution (every 5 min) compared to consumption data (1 hr). Both rely on a pressure-driven hydraulic solver to generate the data set numerically, and uncertainty is not considered (e.g., pipe roughness and node consumptions). The decision to neglect uncertainty is made to evaluate the metamodels performance under controlled and deterministic conditions, allowing to assess the model architecture. Including uncertainty in nodal consumptions or pipe diameter/roughness would hinder these effects and reduce model accuracy.

Data set 1, used to train, validate, and test the Static Metamodel, was generated for different burst scenarios, considering various burst locations and magnitudes. A set of 14,700 burst scenarios was generated, each corresponding to a single time step. This set corresponds to 14 burst sizes (each with seven variation values) simulated at each of the 150 nodes. The burst size is adjusted by changing the emitter coefficient, which is a burst size between 2 and 110 m^3/hr. Each emitter coefficient was multiplied seven times by a factor between 0.8 and 1.2, generated using a uniform distribution, to increase the number of scenarios from 2,100 (14 × 150) to 14,700 (2,100 × 7).

The training, validation, and test data sets were defined considering proportions of 60%, 20%, and 20%, respectively. These data sets also complied with similar proportions of burst magnitudes and locations to have a balanced distribution of bursts in the three data sets. Accordingly, the burst scenarios were divided into three sets according to the magnitude range proposed by Vrachimis et al. (2022): small bursts (0–8 m^3/hr), medium bursts (>8–18 m^3/hr) and large bursts (>18 m^3/hr). Each of these three sets was further divided into the established proportions of 60%, 20%, and 20% to train, validate, and test the metamodel. Finally, the training, validation, and test data sets were obtained by merging the small, medium, and large burst subsets. This approach ensured a balance between the different burst sizes in each data set. Figure 2 depicts the similar distribution of burst magnitudes in the three data sets used.

Another important aspect that contributes to the balance of the data sets is the spatial distribution of the bursts across the network. Thus, the data set must be sufficiently large to ensure a uniform spatial distribution across the network. In other words, each network node should have at least one burst of each size (small, medium, and large) in the training, validation, and test data sets.

Dynamic Metamodel considers an extended evaluation period. The training, validation, and test data sets were defined as 67%, 17%, and 17%, respectively. Given the proximity of the nodes, only 50 node locations were considered. This metamodel was trained for these 50 burst locations, each with 14 fixed magnitudes, resulting in 700 burst scenarios. Each scenario was applied over 24 time steps. Uncertainty is not applied to user consumption, which means that only 24 combinations of constant nodal consumption are considered. Thus, a data set with 16,800 samples was generated to train and validate the model. The test data set (17%) was carried out for 24 hr considering 14 possible burst sizes with 10% variation from those considered in the training combined with 10 locations (3,360 samples).

The burst locations considered in the training and test phases of the metamodel are shown in Figure 3. The nodes established as potential burst locations were selected to be distributed across the network to avoid redundancy. Excessive proximity between nodes can result in similar pressure head variations at other points in the network, making it more difficult for the metamodel to predict accurately.

3.3. Computational Resources

The data set generation and metamodel development were carried out using Python version 3.11.2 and three main libraries, namely, the Water Network Tool for Resilience (WNTR) (Klise et al., 2017) to perform hydraulic simulations for data set generation and PyTorch (Paszke et al., 2019) and PyTorch Geometric (Fey &

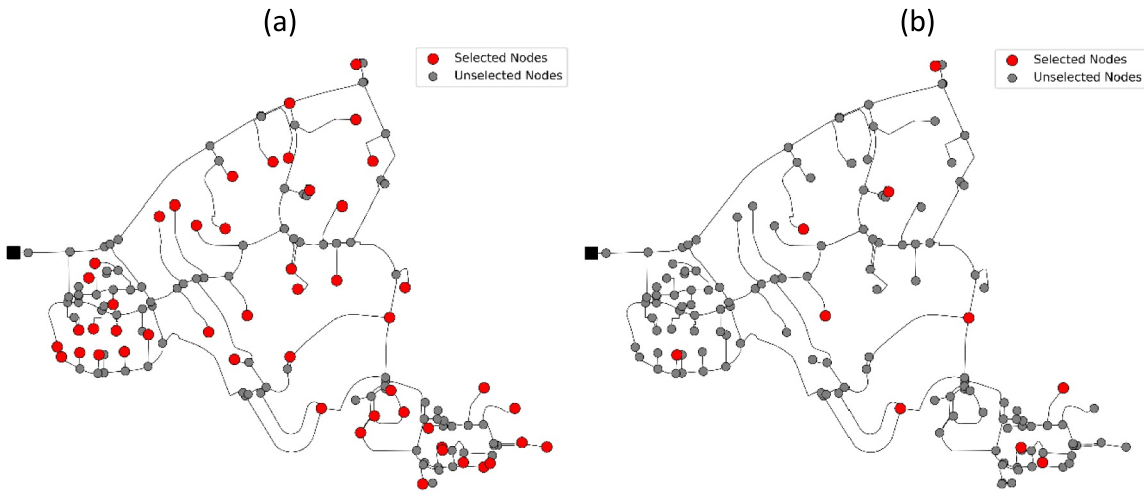

Figure 3. Burst locations for (a) training data set (50 nodes) and (b) test data set (10 nodes).

Lenssen, 2019) for model definition and metamodel training. An NVIDIA GeForce RTX 3070 GPU with 8 GB of memory was used to enhance the training process, whereas testing was conducted on a PC equipped with an Intel® Core™ i7-10700 CPU operating at 2.90 GHz. Ensuring that metamodels can run efficiently on standard hardware is crucial for facilitating the broader adoption and practical application of these techniques. The training times for the static and dynamic metamodels were approximately 6 and 24 hr, respectively.

4. Results

The results are divided into three subsections: optimal sensor configuration, single-period metamodel (Static Metamodel), and extended-period metamodel (Dynamic Metamodel). Metamodels were obtained using the parameters corresponding to the highest coefficient of determination, R^2, achieved after an exhaustive training phase. The parameters are highlighted in bold in Table 1.

4.1. Optimal Sensor Configurations

Before analyzing the optimal sensor configurations, it is important to clarify the experimental workflow adopted in this study. The optimal sensor configurations were obtained in an independent optimization stage (see Section 2.2) and, then, used as fixed sensor layouts for metamodel training and evaluation. Several trial-and-error tests were conducted to obtain the model hyperparameters corresponding to the highest determination coefficient R^2 for all sensor configurations and, thus, allow a fair comparison.

The input parameters were the measured consumption collected from the users' smart metering system at 140 consumption nodes and the measured pressure head for a specific pressure sensor configuration. The optimization algorithm (GA) was run until the stop-criteria are met, namely 10^4 generations or no changes in the objective function (Equation 2) for 100 generations. Nine optimal configurations of pressure sensors were obtained using the procedure described in Section 2.1, namely, 5, 10, 15, 20, 25, 30, 45, 60, and 75 sensors. The locations of the sensors in four of these optimal configurations (5, 20, 45, and 75) are shown in Figure 4. These optimal configurations enable the distribution of pressure sensors across the network, ensuring the maximum piezometric head sensitivity to nodal consumption. Higher sensitivity enhances the metamodel's ability to identify and locate abnormal consumptions in the network. As an increase in nodal consumption leads to a decrease in the pressure head at the nodes, the sensitivity is represented by negative values, indicating that a higher sensitivity corresponds to lower values.

For each number of sensors, the obtained sensor locations were well spread across the WDN instead of concentrating them in a few areas, although they tended to locate sensors closer to larger consumers, ensuring the maximization of the sensor sensitivity. Nodes closer to the storage tank (i.e., the source) exhibited lower sensitivity than those farther away. Network discretization is a key factor in this context, as regions with a higher node density and lower sensitivity tend to have more sensors.

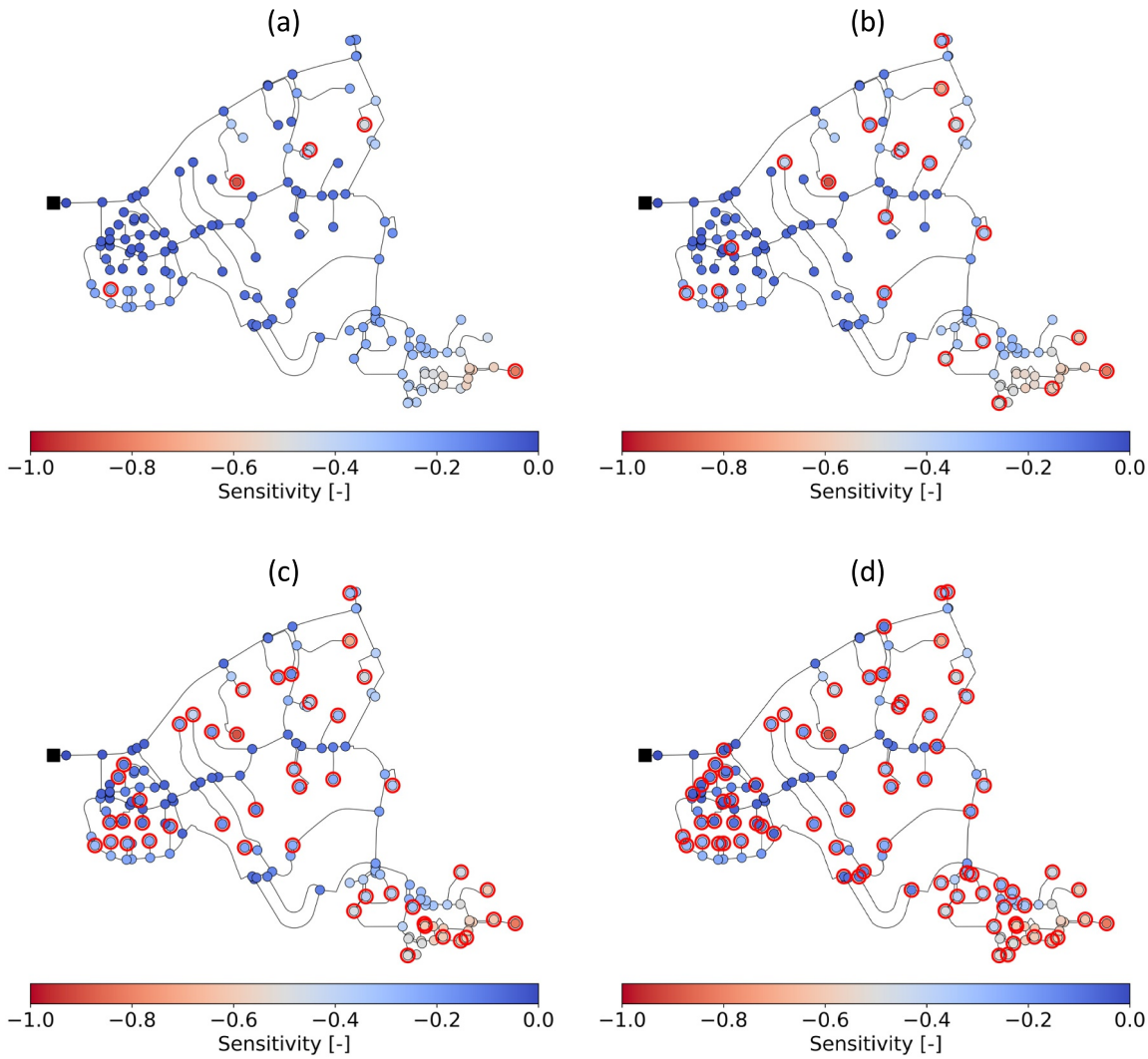

Figure 4. Sets of optimal locations for pressure sensors: (a) 5 sensors, (b) 20 sensors, (c) 45 sensors, (d) 75 sensors.

4.2. Static Metamodel

The single-period metamodel only considers the daily average node consumption, without accounting for daily consumption variations. Table 2 summarizes the results obtained from the analysis and highlights the capacity of Static Metamodel to accurately locate bursts of different sizes (small, medium, and large). The results were obtained for different pressure sensor configurations (5, 20, 45, 60, and 75). Although pressure sensor configurations depend on the network topology, the number of pressure sensors per kilometer enables a fair comparison of the results.

The values of the objective function, $F(X)$, of the optimization for the different configurations demonstrate that increasing the number of pressure sensors enhances the monitoring network's sensitivity until 75 sensors reach a plateau when more than 75 nodes are used. This trend was also observed in the coefficient of determination of the metamodel, R^2. In addition to the R^2 for assessing the metamodel performance, two other metrics were used to evaluate the performance relative to both the burst magnitude and its spatial location. The burst magnitude is estimated using the mean absolute error (MAE), considering different neighborhood range levels. However, the metamodel's ability to spatially locate a burst was assessed by the percentage of test scenarios with location errors lower than 200 m, between 200 and 500 m, and higher than 500 m.

Table 2
Summary of Results for the Static Metamodel With Different Sensor Configurations

No. Sensors	Type of burst	No. of sensors per km	$F(X)$	Train R^2	MAE (m^3/hr)			Distribution of spatial accuracy (%)		
					0-hop	1-hop	2-hop	0–200 m	200–500 m	>500 m
5	Small	0.29	30.11	0.67	5.17	4.79	**4.52**[a]	20.10	**66.50**	13.40
	Medium				11.14	9.96	**9.08**[a]	35.07	**64.22**	0.72
	Large				22.04	17.36	**15.15**[a]	**74.22**	25.78	0.0
20	Small	1.18	36.83	0.83	4.18	3.47	**3.19**[a]	28.45	**61.86**	9.69
	Medium				7.61	5.52	**5.07**[a]	**76.95**	23.05	0.0
	Large				11.75	7.39	**6.35**[a]	**97.72**	2.28	0.0
45	Small	2.65	39.42	0.92	2.80	2.30[a]	**2.15**	40.41	**48.87**	10.72
	Medium				3.40	**1.90**[a]	1.93	**92.02**	7.98	0.0
	Large				6.73	**3.09**[a]	3.37	**99.52**	0.48	0.0
60	Small	3.53	39.66	0.93	2.25	1.91	**1.89**	**57.53**	35.05	7.42
	Medium				2.63	**1.97**	2.02	**98.39**	1.61	0.0
	Large				6.16	**4.64**	4.81	**98.92**	1.08	0.0
≥75	Small	4.40	39.7	0.98	1.88	**1.56**[a]	1.57	**69.90**	25.15	4.95
	Medium				1.33	**1.24**[a]	1.47	**100.0**	0.0	0.0
	Large				3.11	**2.08**[a]	2.41	**100.0**	0.0	0.0

Note. Bold = best MAE values or highest percentage of spatial accuracy. [a]*k*-hops presented in Figure 5.

The accuracy of the metamodel in predicting the magnitude of bursts is evaluated by the MAE calculated by Equation 8 for all scenarios considered in each category of bursts (i.e., small, medium, and large) for the test data set:

$$\mathrm{MAE} = \frac{1}{N_c}\sum_{s=1}^{N_c} |\varepsilon_{q,k}|_s \tag{12}$$

in which N_c is the number of scenarios in category c. MAE values increased with the burst magnitude, reaching values close to 2 m^3/hr only in the configuration with 45 sensors (2-hop). For a higher number of sensors, the metamodel demonstrated a high accuracy in predicting the burst magnitude, whereas bursts below 1.5–2 m^3/h were practically undetectable, regardless of the number of sensors. Another aspect is the tendency of the metamodel to spread the predicted burst flow rate across the neighboring nodes. Configuration with 5–30 sensors presents the lowest error when two neighbors (2-hop) levels were considered. However, configurations with more than 45 sensors tended to present the lowest errors in the first level of neighbors (1-hop).

The spatial accuracy of the metamodel is evaluated by percentage of bursts detected within a certain range (0–200, 200–500, >500 m) for all scenarios considered in each category of bursts for the test data set. Each burst location accuracy is described by the Euclidian Distance between the predicted burst centroid and the ground truth location, given by Equation 9.

The distribution of spatial accuracy values shows that the metamodel ability to locate bursts increases with burst magnitude. Small bursts were only detected with 200 m accuracy for configurations with 60 (57.53%) and 75 sensors (69.90%). The percentage of detected medium bursts with 200 m accuracy was significantly higher, reaching 76.95% for 20 and 100% for 75 sensors. Large bursts were detected with a 200 m accuracy with five sensors (74.22%), reaching 100% with 75 sensors. Almost all medium- and large-sized bursts were located up to 500 m. From the configuration with 60 sensors onwards, almost all bursts were located with a 200 m accuracy.

The metamodel performance obtained from the configurations of 5, 20, 45, and 75 sensors associated with the burst magnitude and spatial location is presented in Figures 5 and 6, respectively.

10.1029/2025WR041195

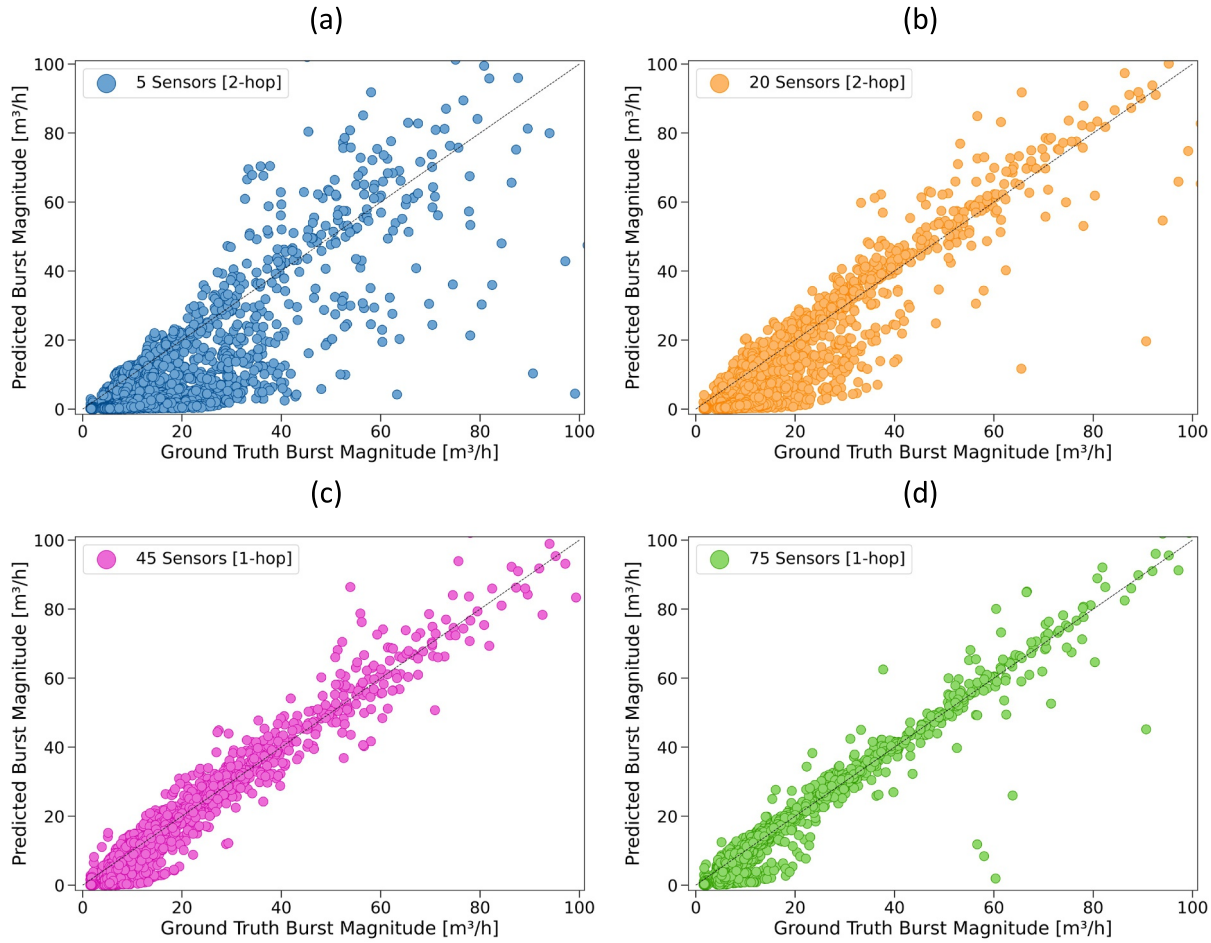

Figure 5. QQ-plot of burst magnitudes between the predicted values and ground truth values for all test samples, considering (a) 5 sensors (ranges from 4.52 to 15.15 m^3/hr), (b) 20 sensors (ranges from 3.19 to 6.35 m^3/hr), (c) 45 sensors (ranges from 2.30 to 3.09 m^3/hr), and (d) 75 sensors (ranges from 1.56 to 2.08 m^3/hr).

Figure 5 depicts the correlation between the predicted and ground truth values of the burst magnitude for the most accurate metamodels in each configuration. These results are consistent with those shown in Table 2. The magnitude error decreases with the increase in the number of pressure sensors. The burst flow rate is spread in the neighboring nodes of the ground truth location, which becomes less evident as the number of sensors increases. Therefore, Figures 5a and 5b represent the burst predicted value as the summation of flow rates in the second level of neighbors (2-hops), whereas Figures 5c and 5d represent this value in the first level of neighbors (1-hop). However, the underestimation of magnitudes for small bursts is evident in 5 and 20 sensor configurations, in which the bursts are predominantly predicted below their ground truth values. This indicates a tendency to underestimate low-magnitude events. Additionally, the MAE values, ranging from 4.52 to 5.17 m^3/hr for the five sensor setup and 3.19–4.18 m^3/hr for the 20 sensor setup, indicate that the average prediction errors represent a substantial part of the small burst magnitude (0–8 m^3/hr). The 75 sensor configuration has more outliers than the 45 sensor one, even though it is more accurate.

Figure 6 depicts the Euclidean distance between the centroid of the predicted burst and the ground truth burst location, and represents these values with the magnitude of the bursts. Different threshold lines were plotted to identify distinct regions in the analysis space: the boundary between small, medium, and large bursts on the horizontal axis and the boundary between 0 and 200, 200–500, and >500 m on the vertical axis. Some regions did not present any points. The number of empty regions increases with the number of sensors (e.g. for 75 sensors, medium and large bursts are located with an error of less than 200 m). Even with a higher number of sensors, the small bursts showed poor performance. This is because small bursts do not significantly affect the pressure

10.1029/2025WR041195

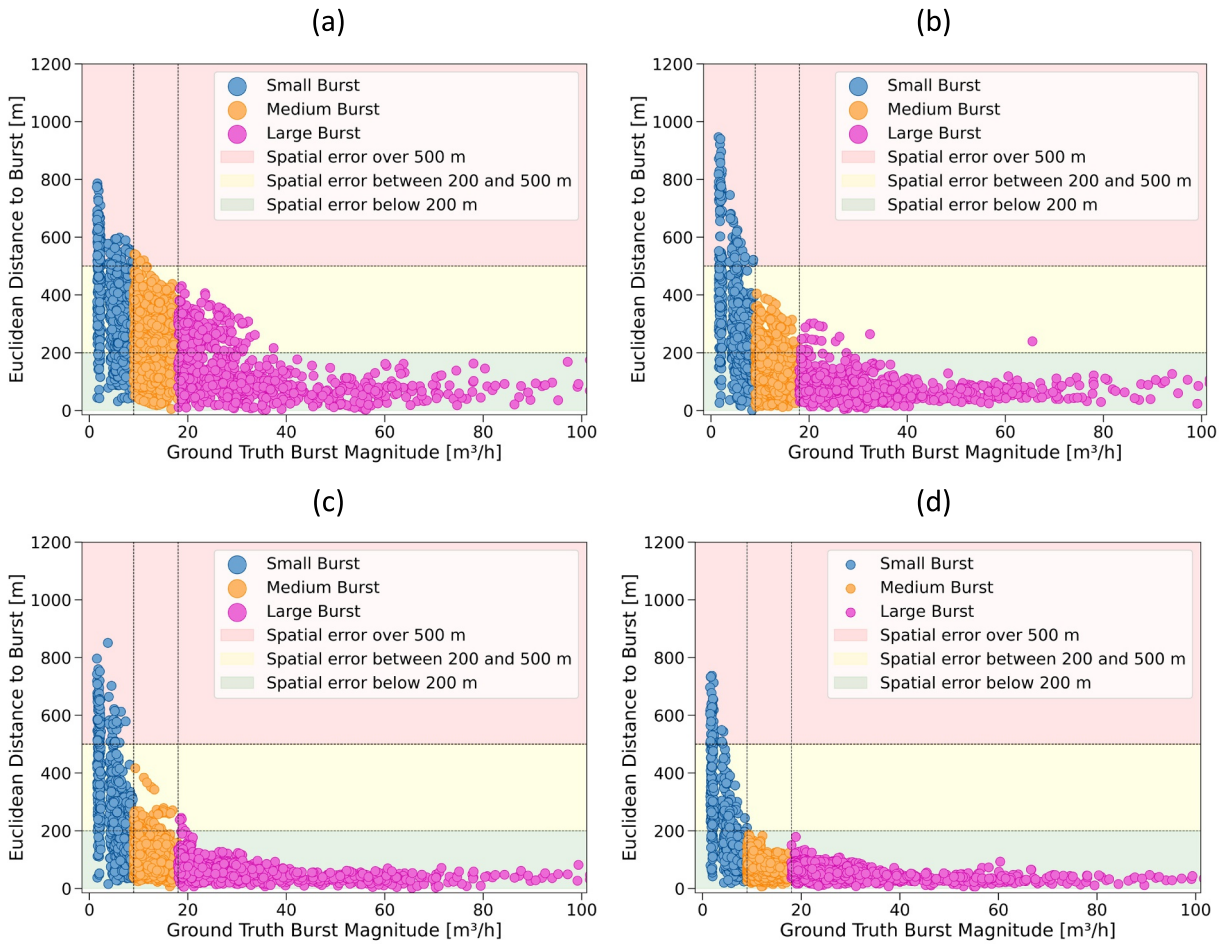

Figure 6. Euclidean distance between the centroid of the predicted burst and ground truth burst location, considering (a) 5 sensors, (b) 20 sensors, (c) 45 sensors, and (d) 75 sensors.

measurements, and the variations are so small that, in real-world cases, they may fall within the accuracy of the piezometer, making it impossible for the metamodel to detect them.

Figure 7 depicts four examples of the predicted burst results in the network map for the configuration with 75 sensors. A color bar represents the predicted burst flow rate of each node. Four vertical lines are plotted in this bar: yellow, green, purple, and black, representing the predicted burst magnitude associated with 0-hop, 1-hop, 2-hop, and ground truth burst magnitudes, respectively. The centroid of the predicted burst is represented by a target symbol, which is used to assess the distance between the predicted and ground truth burst locations (red circle).

Figures 7a and 7b depict two small bursts with 1.78 and 6.04 m^3/hr, respectively. The smallest had a location error of 590.86 m, whereas the other had 115.28 m. Although the smallest burst (1.78 m^3/hr) is at a large distance from the storage tank, it is not possible to accurately identify either its location or magnitude because of its small size. The other small burst (6.04 m^3/hr) achieved good accuracy in both magnitude and location because its size was sufficient to create a pressure variation captured by the sensors, despite being closer to the storage tank.

Figures 7c and 7d depict the results for a medium burst at 12.93 m^3/hr and a large burst at 33.05 m^3/hr, respectively, which are located with errors of 70.24 and 51.71 m, respectively. Even when the burst location accuracy is high, the burst flow rates tend to be spread around the ground truth location. This is particularly evident in the medium burst case (Figure 7c) that is located in an area with a high density of nodes with shorter pipes. The pressure differences between adjacent nodes in these areas are quite small and is hardly captured by this metamodel, or by any other metamodel or hydraulic-based model because this is intrinsic of the physics of the flow.

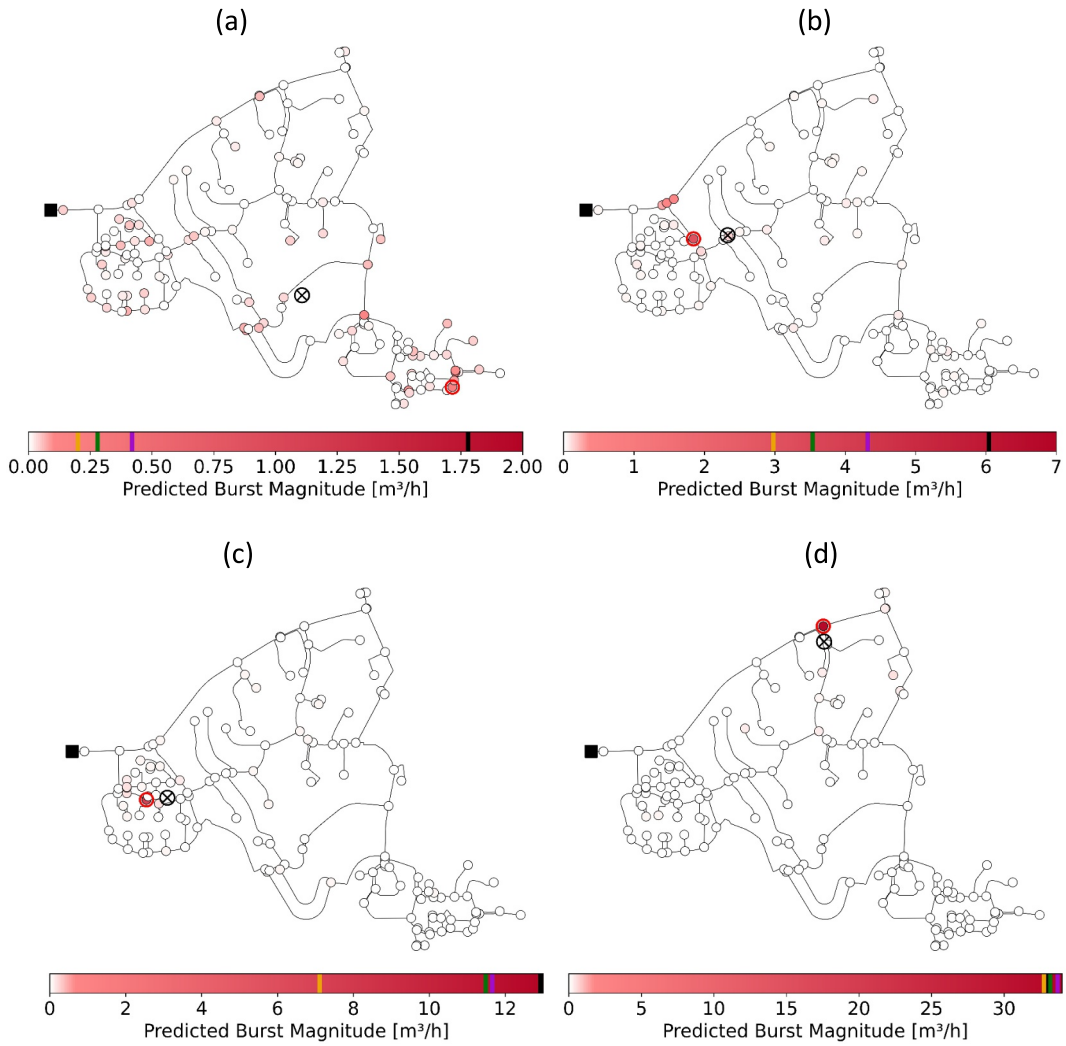

Figure 7. Examples of predicted burst locations using 75 pressure sensors: (a) a small burst located at a distance greater than 500 m; (b) a small burst located within the range of 200–500 m; (c) a medium burst located at less than 200 m; and (d) a large burst located at less than 200 m.

4.3. Dynamic Metamodel

An extended period simulation was performed to account for daily variations in consumption. The input of Dynamic Metamodel consisted of a set of user consumptions (140 nodes) and a set of pressure measurements (20 sensors) for 1 hr. Twenty sensors were selected as a balance between metamodel accuracy and sensor costs. Data from each of the 24 hr were included separately in the training data set. Dynamic Metamodel is trained with 16,800 samples (i.e., 24 hr combined with 14 burst sizes and 50 burst locations) for 20 pressure sensors' configurations (Figure 3b). Dynamic Metamodel was tested for 3,360 samples (i.e., 24 hr, 14 burst sizes ±10% variation, and 10 burst locations).

The test results are presented in Table 3 for the hours of the highest consumption (5 a.m.) and the lowest consumption (3 p.m.). This atypical consumption pattern is caused by the water used for irrigation of green spaces, which occurs at night. It has a greater impact than domestic consumption, which occurs during the day. Even when accounting for daily variations, the metamodel locates 100% of medium and large bursts with an error lower than 200 m, whereas only 85% of the small bursts are located with the same accuracy. The model's ability to predict burst magnitude is higher at the hour of the highest consumption because friction losses increase with flow rate, and pressure measurements are more sensitive to burst flow rate. Although there was a discrepancy in the prediction of burst magnitude, the MAE remained consistently below 1 m^3/hr.

Table 3
Summary of Results for the Dynamic Metamodel With 20 Pressure Sensors

No. sensors	Time	Type of burst	Train R^2	Mean absolute error (m^3/hr)			Distribution of spatial accuracy (%)		
				0-hop	1-hop	2-hop	0–200 m	200–500 m	>500 m
20	5 a.m.	Small	0.99	0.42	0.42	0.42	85.29	14.71	0.0
		Medium		0.13	0.14	0.15	100.0	0.0	0.0
		Large		0.23	0.25	0.29	100.0	0.0	0.0
20	3 p.m.	Small	0.99	0.94	0.92	0.91	85.71	10.71	3.57
		Medium		0.23	0.21	0.20	100.0	0.0	0.0
		Large		0.88	0.86	0.83	100.0	0.0	0.0

Figure 8 presents the Dynamic Metamodel test results for the burst magnitude (a, c) and location (b, d). These results are consistent with those presented in Table 3. Dynamic Metamodel detects the lowest small bursts (i.e., vertically aligned blue points) with low accuracy (200–600 m), as in the case of Static Metamodel. These bursts are almost undetectable not only due to their low magnitude but also because they are located close to the storage tank, with a distance of 300 m measured within the graph that represents the water distribution network. The metamodel was not sensitive to the burst flow rate (see an example of this case in Figure 7b).

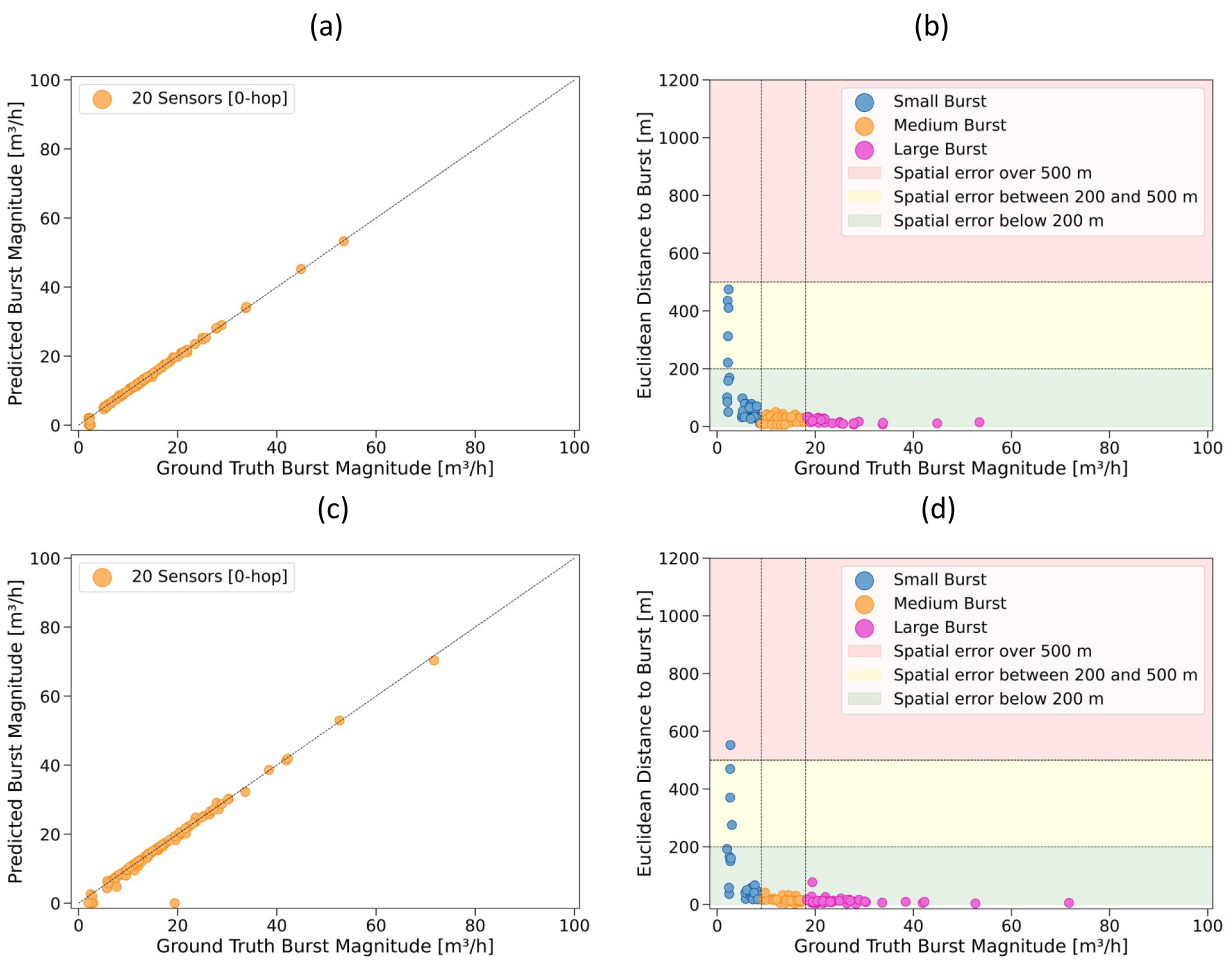

Figure 8. Dynamic metamodel results in terms of burst magnitude and location: (a, b) the hour of highest consumption (5 a.m.); and (c, d) the hour of lowest consumption (3 p.m.).

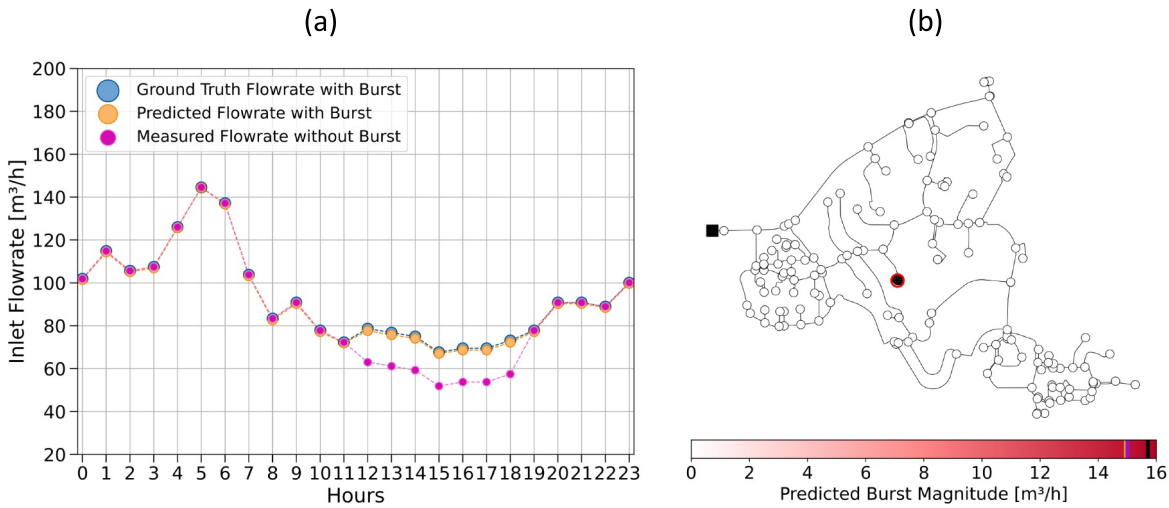

Figure 9. Scenario with a burst occurring between 12 and 6 p.m.: (a) flow rate variation at the inlet of the network; (b) burst location and magnitude (hop-0).

Although Dynamic Metamodel shows high accuracy at both the highest and lowest consumption hours (Figure 8a), it presents a significant error at hop-0 for a burst with a magnitude of approximately 20 m^3/hr at the lowest consumption hour (see point dispersions in Figure 8c). This is because Dynamic Metamodel spreads the burst flow rate in the influence area of the burst node (compare hop-0 and hop-2 in Table 3). This raises concerns regarding its practical application; however, a prediction failure at a specific time step does not compromise the entire approach.

The application of the metamodel in near real-time (with hourly latency) is demonstrated herein with two scenarios: one without a burst and another with a burst occurring at a certain period of the day.

Dynamic Metamodel has been tested and validated and is being applied almost in real-time. Consider that data from the users' smart metering system (i.e., measured nodal consumption) and SCADA system (i.e., pressure sensor data) are collected every hour and immediately processed and used as input for Dynamic Metamodel. This metamodel returns the prediction of the nodal consumption at all network nodes, and the burst magnitude is computed as the difference between the predicted and measured nodal consumption at all nodes. In the absence of a burst, the metamodel only predicts the measured flow at the node's rate (i.e., measured nodal consumption). In the event of a burst, the metamodel should be able to predict the sum of the burst with the measured nodal consumption.

Figure 9 illustrates the application of Dynamic Metamodel for the scenarios without any burst and with a burst of 16 m^3/hr occurring at 12 p.m. Figure 9a presents the flow rate variation at the inlet of the network for both scenarios, in which the difference between the two lines can be used to identify the time of occurrence of the burst. Figure 9b shows that the metamodel detects and locates the burst, triggering the repair team, which completes the repair work by 6 p.m. During the time interval between the burst detection and repair, the metamodel predicts the burst flow rate. After the burst is repaired, the metamodel no longer predicts the existence of a burst. The burst did not have the same magnitude over the 7 hours because the magnitude was pressure-dependent; nevertheless, the metamodel could detect these variations in the burst magnitude. The vertical lines on the colorbar (yellow, green, purple, and black) represent the average burst magnitude predicted by the metamodel over seven predictions for the 0-hop, 1-hop, and 2-hop and actual burst magnitudes.

5. Discussion

Two metamodels were developed in this study. The first considered a single set of nodal consumptions (the 24-hr average value at each node) combined with 98 burst sizes at 150 locations. This stage allowed us to establish the problem formulation and develop the metamodel architecture. The results are promising but far from being applied in a real-life system because the metamodel did not meet the daily nodal consumption variation. Dynamic Metamodel, with the same architecture as Static Metamodel, is trained and validated on a data set with a finer

temporal resolution, including the 24-hr consumption variations. Both metamodels allow the accurate location of medium and large bursts (<200 m), but small bursts, mainly when located close to the storage tank, are challenging to detect because the metamodel tends to spread the burst flow rate over a large area of the network. Areas with a high density of nodes have less sensitivity of pressure-head to flow rate changes and, thus, the metamodel tends to naturally spread bursts around the ground truth location even for medium and large burst. This has also been observed when using hydraulic-based models for burst location (Ferreira et al., 2025), because this is intrinsic of the physics of the flow.

Dynamic Metamodel is close to real-life applications, as it is developed to receive nodal consumption every hour; however, it is trained with a single 24-hr pattern at each node. It does not account for the fact that, although there is a daily nodal consumption pattern in real-life scenarios, it is never deterministic. Therefore, this metamodel should be tested with small variations associated with the 24-hr patterns, and if the results are not satisfactory, it should be trained with data sets that integrate this uncertainty. It is also important to note that the analyzed DMA has a favorable tree-shaped topology and relatively small size, providing ideal conditions for demonstrating the approach; however, these characteristics limit the direct generalization of the results to the case of looped or oversized networks, where pressure variations from abnormal consumptions are not significant. This issue is transversal to all models dependent on pressures sensitivities to consumption changes. Further research should be developed to adapt or improve the metamodels to address this concern, eventually by developing physics-informed metamodels.

The developed metamodels rely heavily on the number and location of the measurement sensors. A larger number of sensors (i.e., 75 sensors, corresponding to 4.4 sensors/km) enables the creation of a more accurate metamodel with a burst location uncertainty of less than 200 m. In contrast, a limited number of sensors (i.e., 5 sensors, corresponding to one sensor every 3 km) may fail to capture the necessary information about burst occurrences, making it challenging to detect, locate, and approximately quantify the event. However, a higher density of sensors may be optimistic and potentially unrealistic for medium- and small-sized water utilities due to cost and maintenance constraints. The authors truly believe that increasing advances in metering technology will allow the development of low cost sensors with dual-measurements (pressure and flow rate) in the short or medium terms.

Another important insight is that the pipe network diameter configuration strongly affects the metamodel performance, and the sensitivity to head loss is crucial for improving accuracy. This was observed indirectly by noting that in Dynamic Metamodel, the lowest consumption hour had lower accuracy than the highest consumption hour. Before developing the metamodels, an analysis was conducted to determine the optimal number and location of the sensors. This analysis should be integrated into the metamodel development process, particularly by incorporating a sensitivity matrix that describes the sensitivity of nodal pressure to variations in consumption into the metamodel training process. This sensitivity is represented by the Jacobian matrix, which can be computed by the neural network during training, providing the metamodel characteristics of a physics-informed neural network. By incorporating these changes into the architecture, the metamodels learn the physical laws of pressurized flow, thereby improving their accuracy and reliability for real-life applications.

In future studies, the detection of simultaneous abnormal consumptions should be considered, despite simultaneous bursts being very rare situations that have hardly been addressed in literature. Indeed, overlapping hydraulic responses from multiple events may challenge the metamodel's ability to accurately locate each occurrence. If the metamodel performance is poor, two main improvements can be considered: using a very large data set with multiple simultaneous bursts, or adding inductive biases for reducing the need for prohibitively large training data sets. These inductive biases can be physical informed graph neural networks that respect the physical laws of pressurized flow, or temporal bias to capture the natural time shifts between multiple abnormal consumption events. Together, these measures improve the model efficiency, generalization, and adaptability.

In addition, the uncertainty associated with real nodal consumptions will require the metamodels to generalize effectively and handle out-of-distribution inputs, which are expected in real operational data. Therefore, the final validation of the proposed approach should be carried out under real operating conditions and will be presented in future work, representing the ultimate test of its robustness, adaptability, and practical applicability.

Finally, this paper evaluates model performance mainly using MAE for burst magnitude and Euclidean distance for burst location. While these metrics are appropriate for quantifying numerical prediction accuracy, they do not

capture the detection performance (e.g., whether a burst is detected at all, or the rate of false positives). In anomaly detection, complementary metrics, such as precision, recall, F1-score, or time-to-detection are often used to provide a better understanding of detection reliability. Future studies will consider equal emphasis on the three main aspects of the problem—detection, localization, and quantification—incorporating additional metrics to assess detection reliability. However, only Dynamic Metamodel, which was trained using an extended period analysis, could be used to evaluate temporal detection performance, particularly the time-to-detection of an abnormal event.

6. Conclusions

This paper proposes a methodology for locating and identifying abnormal consumptions in WDN. The methodology has two main steps: the optimization of the location of pressure sensors and the development of metamodels as emulators of burst scenarios. Two metamodels with the same architecture have been developed: the first simulates a single time step representing average network consumption (Static Metamodel) and is used for model architecture definition, and the second simulates the 24-hr consumption (Dynamic Metamodel). Both models receive data collected from all users and pressure data sensors distributed throughout the network as input parameters. The metamodels return the estimate of the total consumption in all network nodes, and the abnormal consumption is calculated by deducing the measured consumption. The best model is the Dynamic Metamodel as it was trained with the daily variation of consumptions, allowing bursts' detection in near real-time independently of the time of the day, whereas the static model is much more limited, since it was only trained with the average daily consumption for all users.

The results obtained by considering different pressure sensor configurations highlighted several key findings regarding the sensitivity and accuracy of the metamodel in locating and identifying bursts. For Static Metamodel, even with a minimal number of sensors (5–15), most small- and medium-sized bursts are located within 200–500 m, whereas large bursts are predominantly detected within 200 m, regardless of the number of sensors. Increasing the number of sensors enhances the success of the metamodel in locating medium and large bursts within 200 m, nearly reaching 100%. However, the metamodel did not show good accuracy for small bursts due to the reduced sensitivity of the network to pressure head loss caused by lower-magnitude bursts. For Dynamic Metamodel, the results are similar: the metamodel locates 100% of medium and large bursts with a small error (<200 m), whereas only 85% of the small bursts are located with the same accuracy. The metamodel can more easily predict the burst magnitude at the peak hour, despite the MAE remaining consistently below 1 m^3/hr even at the minimum consumption hour.

As water utilities adopt near-real-time monitoring using smart meters and pressure sensors, the application of the proposed approach becomes more viable. The main advantage of metamodels over classic hydraulic simulators is the computational time, allowing the metamodel to detect bursts in near-real-time as the models receive the collected data. The bottleneck of this approach lies in data transmission, gathering, and processing within the central server. Once this is overcome, along with more robust metamodels integrating consumption uncertainty, burst detection will be almost instantaneous and part of the daily operation of the WDN. This methodology supports the ongoing evolution of smart water networks, enabling more effective and faster abnormal consumption detection and location.

Conflict of Interest

The authors declare no conflicts of interest relevant to this study.

Data Availability Statement

All data and the code supporting the conceptual model used in this study have been deposited in the Zenodo public repository and can be accessed at https://doi.org/10.5281/zenodo.17802822.

Acknowledgments
The authors would like to thank the Fundação para a Ciência e a Tecnologia (FCT) for supporting this work through the StreamWater project (reference no. 2024.07282.IACDC), funded by the Recovery and Resilience Plan (RRP) within the framework of the financing agreement with the "Recuperar Portugal" Task Force. Additional support was provided by FCT through project UID/6438/2025 of the research unit CERIS, and by FEDER and FCT for the AQUALEARN project—Machine learning-based digital twins for real time anomaly detection in water supply systems (reference numbers LISBOA2030-FEDER-00816100 and FCT 16878, https://doi.org/10.54499/2023.18249.ICDT). FCT also supported João Caetano's PhD grant (reference no. 2022.13214.BD).

References

Barton, N. A., Farewell, T. S., Hallett, S. H., & Acland, T. F. (2019). Improving pipe failure predictions: Factors affecting pipe failure in drinking water networks. *Water Research*, *164*, 114926. https://doi.org/10.1016/j.watres.2019.114926

Bartos, M., Thomas, M., Kim, M.-G., Frankel, M., & Sela, L. (2024). Online state estimation in water distribution systems via extended Kalman filtering. *Water Research*, *264*, 122201. https://doi.org/10.1016/j.watres.2024.122201

Blank, J., & Deb, K. (2020). Pymoo: Multi-objective optimization in python. *IEEE Access*, *8*, 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567

Broad, D. R., Dandy, G. C., & Maier, H. R. (2005). Water distribution system optimization using metamodels. *Journal of Water Resources Planning and Management*, *131*(3), 172–180. https://doi.org/10.1061/(ASCE)0733-9496(2005)131:3(172)

Broad, D. R., Maier, H. R., & Dandy, G. C. (2010). Optimal operation of complex water distribution systems using metamodels. *Journal of Water Resources Planning and Management*, *136*(4), 433–443. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000052

Brody, S., Alon, U., & Yahav, E. (2021). How attentive are graph attention networks? Retrieved from http://arxiv.org/abs/2105.14491

Caetano, J., Ferreira, B., Carriço, N., & Covas, D. (2025). Redundancy-driven resilience in optimal design of water distribution networks. https://doi.org/10.1061/JWRMD5.WRENG-6613

Daniel, I., & Cominola, A. (2023). Estimating irregular water demands with physics-informed machine learning to inform leakage detection. Retrieved from http://arxiv.org/abs/2309.02935

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. Retrieved from http://arxiv.org/abs/1606.09375

Ferreira, B., Carriço, N., & Covas, D. (2025). Near real-time leak location by inverse analysis integrating measurement uncertainty. *Water Resources Management*, *39*(1), 503–521. https://doi.org/10.1007/s11269-024-03983-w

Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch geometric. Retrieved from http://arxiv.org/abs/1903.02428

Garzón, A., Kapelan, Z., Langeveld, J., & Taormina, R. (2022). Machine learning-based surrogate modeling for urban water networks: Review and future research directions. *Water Resources Research*, *58*(5), e2021WR031808. https://doi.org/10.1029/2021WR031808

Garzón, A., Kapelan, Z., Langeveld, J., & Taormina, R. (2024). Transferable and data efficient metamodeling of storm water system nodal depths using auto-regressive graph neural networks. *Water Research*, *266*, 122396. https://doi.org/10.1016/j.watres.2024.122396

Giustolisi, O., Mazzolani, G., Berardi, L., & Laucelli, D. B. (2024). From advanced hydraulic modelling to performance indicator for the efficiency of investments in leakage management of pressurized water systems. *Water Research*, *258*, 121765. https://doi.org/10.1016/j.watres.2024.121765

Kerimov, B., Taormina, R., & Tscheikner-Gratl, F. (2024). Towards transferable metamodels for water distribution systems with edge-based graph neural networks. *Water Research*, *261*, 121933. https://doi.org/10.1016/j.watres.2024.121933

Klise, K. A., Bynum, M., Moriarty, D., & Murray, R. (2017). A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study. *Environmental Modelling & Software*, *95*, 420–431. https://doi.org/10.1016/j.envsoft.2017.06.022

Li, Z., Liu, H., Zhang, C., & Fu, G. (2024). Real-time water quality prediction in water distribution networks using graph neural networks with sparse monitoring data. *Water Research*, *250*, 121018. https://doi.org/10.1016/j.watres.2023.121018

Mazzoni, F., Marsili, V., Alvisi, S., & Franchini, M. (2024). Detection and pre-localization of anomalous consumption events in water distribution networks through automated, pressure-based methodology. *Water Resources and Industry*, *31*, 100255. https://doi.org/10.1016/j.wri.2024.100255

Meirelles, G., Manzi, D., Brentan, B., Goulart, T., & Luvizotto, E. (2017). Calibration model for water distribution network using pressures estimated by artificial neural networks. *Water Resources Management*, *31*(13), 4339–4351. https://doi.org/10.1007/s11269-017-1750-2

Menapace, A., Zanfei, A., Felicetti, M., Avesani, D., Righetti, M., & Gargano, R. (2020). Burst detection in water distribution systems: The issue of dataset collection. *Applied Sciences*, *10*(22), 8219. https://doi.org/10.3390/app10228219

Mounce, S. R., Boxall, J. B., & Machell, J. (2010). Development and verification of an online artificial intelligence system for detection of bursts and other abnormal flows. *Journal of Water Resources Planning and Management*, *136*(3), 309–318. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000030

Nian-dong, L., Kun, D., Jia-peng, T., & Wei-Xin, D. (2017). Analytical solution of Jacobian matrices of WDS models. *Procedia Engineering*, *186*, 388–396. https://doi.org/10.1016/j.proeng.2017.03.236

Örn Garðarsson, G., Boem, F., & Toni, L. (2022). Graph-based learning for leak detection and localisation in water distribution networks. *IFAC-PapersOnLine*, *55*(6), 661–666. https://doi.org/10.1016/j.ifacol.2022.07.203

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. Retrieved from http://arxiv.org/abs/1912.01703

Razavi, S., Tolson, B. A., & Burn, D. H. (2012). Review of surrogate modeling in water resources. *Water Resources Research*, *48*(7), W07401. https://doi.org/10.1029/2011WR011527

Romero-Ben, L., Alves, D., Blesa, J., Cembrano, G., Puig, V., & Duviella, E. (2022). Leak localization in water distribution networks using data-driven and model-based approaches. *Journal of Water Resources Planning and Management*, *148*(5), 04022016. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001542

Sophocleous, S., Savić, D., & Kapelan, Z. (2019). Leak localization in a real water distribution network based on search-space reduction. *Journal of Water Resources Planning and Management*, *145*(7), 04019024. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001079

Todini, E., & Rossman, L. A. (2013). Unified framework for deriving simultaneous equation algorithms for water distribution networks. *Journal of Hydraulic Engineering*, *139*(5), 511–526. https://doi.org/10.1061/(ASCE)HY.1943-7900.0000703

Truong, H., Tello, A., Lazovik, A., & Degeler, V. (2024). Graph neural networks for pressure estimation in water distribution systems. *Water Resources Research*, *60*(7), e2023WR036741. https://doi.org/10.1029/2023WR036741

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph attention networks. Retrieved from http://arxiv.org/abs/1710.10903

Vrachimis, S. G., Eliades, D. G., Taormina, R., Kapelan, Z., Ostfeld, A., Liu, S., et al. (2022). Battle of the leakage detection and isolation methods. *Journal of Water Resources Planning and Management*, *148*(12), 04022068. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001601

Wu, Y., & Liu, S. (2017). A review of data-driven approaches for burst detection in water distribution systems. *Urban Water Journal*, *14*(9), 972–983. https://doi.org/10.1080/1573062X.2017.1279191

Wu, Y., Liu, S., & Kapelan, Z. (2024). Addressing data limitations in leakage detection of water distribution systems: Data creation, data requirement reduction, and knowledge transfer. *Water Research*, *267*, 122471. https://doi.org/10.1016/j.watres.2024.122471

Xing, L., & Sela, L. (2022). Graph neural networks for state estimation in water distribution systems: Application of supervised and semi-supervised learning. *Journal of Water Resources Planning and Management*, *148*(5), 04022018. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001550

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks? Retrieved from http://arxiv.org/abs/1810.00826

Zanfei, A., Menapace, A., Brentan, B. M., Sitzenfrei, R., & Herrera, M. (2023). Shall we always use hydraulic models? A graph neural network metamodel for water system calibration and uncertainty assessment. *Water Research*, *242*, 120264. https://doi.org/10.1016/j.watres.2023.120264

Zhang, Z., & Fink, O. (2024). Algorithm-informed graph neural networks for leakage detection and localization in water distribution networks. Retrieved from http://arxiv.org/abs/2408.02797

Zhou, X., Tang, Z., Xu, W., Meng, F., Chu, X., Xin, K., & Fu, G. (2019). Deep learning identifies accurate burst locations in water distribution networks. *Water Research*, *166*, 115058. https://doi.org/10.1016/j.watres.2019.115058