Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal, 2027, 21(1), 578–592
https://doi.org/10.12913/22998624/240425
ISSN 2299-8624, License CC-BY 4.0

Received: 2026.08.31
Accepted: 2026.11.16
Published: 2026.12.01

Use of machine learning methods for anomaly detection in water distribution systems

Kamil Grodzki[1], Ariel Antonowicz[1*]

[1] Institute of Computing Science, Poznan University of Technology, ul. Piotrowo 2, 60-965 Poznan, Poland
* Corresponding author's e-mail: ariel.antonowicz@put.poznan.pl

ABSTRACT

The primary objective of this study is to evaluate selected machine learning architectures for spatial localization of leaks in water distribution networks using pressure telemetry data in a multi-topology benchmark spanning 35 networks. Since all simulated scenarios contain at least one leak, scenario-level leak presence detection is trivial in this setting. Accordingly, we do not report leak/no-leak detection metrics and focus on localization selectivity. The comparative analysis focuses on node-level localization performance, spatial concentration of false alarms expressed as topological hop-distance, and inference time. The graph-based architectures exhibited a pronounced over-prediction tendency: although they achieved very high node-level recall, they also generated a large number of false-positive node classifications, limiting their practical applicability in the investigated configuration. Among the evaluated architectures, the LSTM model provided the most favorable trade-off between localization sensitivity, false-positive rate, spatial precision, and computational efficiency. It achieved a node-level sensitivity of 93.44%, a node-level false-positive rate of 21.96%, a mean false-positive hop-distance of 1.26, and an inference time of approximately 2 seconds. The results indicate that LSTM-based architectures are promising candidates for further validation toward decision-support systems for water distribution network monitoring, while graph-based models require additional probability calibration, threshold optimization, and sparsity-oriented regularization. These results should be interpreted as a comparative simulation benchmark under full-node pressure observability and per-network training; validation on real, sparsely instrumented SCADA telemetry remains future work.

Keywords: water distribution system, anomaly detection, machine learning, graph neural networks, hydraulic simulation, WNTR, EPANET, critical infrastructure, decision support system.

INTRODUCTION

Water distribution systems (WDSs) constitute a foundation of modern civilization. Their history dates back to ancient times when the first water distribution systems enabled the dynamic development of cities and human settlements. Today, water infrastructure plays a multidimensional role, ensuring uninterrupted access to drinking water for households, public institutions, industrial facilities, and other areas of daily human activity. The continuity and security of water supply are critical for the functioning of key economic sectors, including the energy and food industries, and serve as a direct determinant of public health. Any disruptions to the operation of water infrastructure entail a severe risk of cascading economic and epidemiological crises. These circumstances make water infrastructure one of the most vital elements of a nation's critical infrastructure. Given the growing scale of this infrastructure [8, 12] and the widespread adoption of SCADA systems, utility operators currently manage massive volumes of telemetry data. However, manual or poorly automated analysis of these data is highly inefficient, necessitating the exploration of novel analytical methods [19]. This problem gains particular prominence in light of new legal frameworks, notably the Directive (EU) 2020/2184 of the European Parliament and of the Council, which imposes an obligation on water utilities to quantitatively

assess water losses using the infrastructure leakage index (ILI) [4]. Historically, anomaly detection in WDSs relied primarily on minimum night flow (MNF) analysis and manual monitoring based on predefined alarm thresholds. These reactive methods can delay the response time to emerging leaks by up to 72 hours [7]. To address these limitations, advanced mathematical models have been developed, for example non-linear NLARX models [16] or hydraulic models like WaterGems coupled with SCADA. Nevertheless, they still exhibit a high false-positive rate due to their reliance on fixed decision thresholds.

Breakthrough results were achieved by applying machine learning (ML) algorithms to district metered area (DMA) data, which reduced the average leak detection time to under 11 hours [7]. However, identifying anomalies in poorly metered networks (e.g., in rural areas) [6] and incorporating physico-chemical parameters [9] remain significant challenges. Machine learning approaches applied in this domain are generally divided into time-series methods and spatial-graph methods [18]. Among unsupervised algorithms, researchers analyzing LSTM autoencoders have highlighted the so-called generalization paradox, wherein models with the lowest reconstruction error learn to reconstruct anomalous states as well [15]. Conversely, in the realm of supervised methods, a review of anomaly detection techniques for SCADA systems by Biström [3] demonstrated that ensemble classifiers, such as Random Forest, can significantly outperform classical deep learning architectures (including autoencoders and GANs).

Furthermore, in water demand forecasting studies [1], a multilayer perceptron (MLP) was successfully employed, reducing the relative root mean square error (RRMSE) to below 8.5%. Concurrently, the digital twin concept is gaining traction. The deployment of an IoT-SCADA-ML system on a small-scale campus recycled water network achieved 100% accuracy in the simultaneous detection and localization of controlled leaks [14].

Nonetheless, an analysis of the existing literature reveals three key research gaps. Firstly, existing comparative studies typically evaluate one or a few networks with predefined topologies, which precludes assessing how network structure impacts algorithmic performance [18]. Secondly, graph neural networks (GNNs) and spatio-temporal graph approaches (DGT, STGT) are rarely included in systematic comparative studies alongside classical convolutional and recurrent architectures [3, 18]. Thirdly, the precise spatial localization of detected anomalies in complex, real-world conditions remains an open challenge.

In response to the aforementioned challenges, the primary objective of this study is to evaluate the effectiveness of selected machine learning architectures for spatial leak localization under a controlled multi-leak simulation regime, and analyze the resulting false-alarm behavior (hop-distance) and inference time. The main contributions of this paper are as follows:

1. A multi-topology benchmark covering 35 diverse WDNs (up to 12,527 junctions) with synthetic multi-leak pressure scenarios generated in EPANET/WNTR under a controlled leak-density stress test (10–50% junctions).
2. An operationally oriented evaluation of localization selectivity that goes beyond node recall/FPR by quantifying the spatial concentration of false alarms via the mean false-positive hop-distance to the nearest true leak.
3. A systematic per-network comparison of five architecture families (CNN, LSTM, GNN, DGT, STGT), including scalability limits and inference-time analysis across topologies.

RESEARCH METHODOLOGY

The water distribution network is represented as an undirected graph $G = (V, E)$, where V is the set of nodes (junctions V_J, tanks V_T, and reservoirs V_R) and E is the set of pipe connections. The network topology is represented by a binary adjacency matrix $A \in \{0,1\}^{(N\times N)}$. Self-loops and symmetric normalization are defined as:

$$\tilde{A} = A + I, \qquad \tilde{D}_{ii} = \sum_j \tilde{A}_{ij},$$
$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\,\tilde{A}\,\tilde{D}^{-\frac{1}{2}} \tag{1}$$

Here, I is the identity matrix, $\tilde{D}$ is the degree matrix of $\tilde{A}$, and $\hat{A}$ is the normalized adjacency matrix used by the graph-based architectures. The input data for a single scenario consists of a matrix of hydraulic pressure time series across all network nodes:

$$X \in \mathbb{R}^{T\times N}, \quad X_{t,i} = max\big(0, h_i(t\Delta t)\big) \tag{2}$$

where: $T = 96$ is the number of time steps, $\Delta t =$ 15 min corresponds to a 24-hour horizon,

and $h_i(\cdot)$ denotes the pressure head at node v_i obtained from pressure-dependent demand (PDD) simulation.

Negative pressure-head values are clipped to zero for consistency with the emitter formulation, which is defined for non-negative pressure head.

For a leak-scenario index $s \in \{1, \ldots, S_{leak}\}$, where $S_{leak} = 1000$, the leak-node fraction is determined by linear interpolation:

$$r_s = r_{min} + \frac{s-1}{S_{leak}-1}(r_{max} - r_{min}), \quad r_{min} = 0.10, r_{max} = 0.50 \tag{3}$$

The number of leaking junctions is

$$K_s = max\left(1, round\left(N_J r_s\right)\right) \tag{4}$$

For each scenario, the leaking subset $S_s \subset V_J$ with $|S_s| = K_s$. Each node $v_k \in S_s$ is assigned an independent random orifice area $A_k \sim U(A_{min}, A_{max})$, where $A_{min} = 5\times10^{-5}$ m^2 and $A_{max} = 5\times10^{-4}$ m^2. The leak discharge is modeled using the orifice equation:

$$Q_k(t) = C_d A_k \sqrt{2g\, h_k(t)}, C_d = 0.75 \tag{5}$$

where: $h_k(t)$ is the non-negative pressure head at node v_k and g is gravitational acceleration. The simulated leak remains active throughout the 24-hour hydraulic simulation.

For each node $v_i \in V$, we define a binary localization label. Let S_s denote the set of leaking junction nodes in scenario s:

$$y_{loc,i} = \mathbb{1}[v_i \in S_s] \in \{0,1\} \tag{6}$$

This yields $y_{loc} \in \{0,1\}^N$.

$$y_{det} = \mathbb{1}[|S_s| > 0] \in \{0,1\} \tag{7}$$

In the dataset used in this study, each scenario contains at least one leak; therefore, $y_{det} \equiv 1$ and scenario-level leak/no-leak classification is not evaluated. The detection head output p_{det} is treated as an auxiliary confidence score, while the main supervised objective and reported results focus on node-level localization.

The model M_θ consists of a shared feature extractor and two prediction heads returning $p_{det} = \sigma(f_\theta(X)) \in [0,1]$ and $p_{loc} = \sigma(g_\theta(X)) \in [0,1]^N$, respectively, where $\sigma(\cdot)$ is the sigmoid function. The combined training objective is:

$$\mathcal{L}(\theta) = w_{det}\mathcal{L}_{det} + w_{loc}\mathcal{L}_{loc} \tag{8}$$

In the leak-only setting, the detection loss does not correspond to a leak/no-leak decision problem; thus, all reported comparisons are based on the localization head. Although the architecture contains both heads, only node-level localization results are reported in this paper. Scenario-level detection metrics are omitted because the evaluation dataset does not contain leak-free scenarios. Both components are binary cross-entropy losses. In the leak-only setting, the detection head is retained as an auxiliary confidence output and is not interpreted as a supervised leak/no-leak classifier. The reported comparisons are based on localization. The detection weight is $w_{det} = 1$, whereas wloc depends on the architecture, as specified in Table 4.

$$\mathcal{L}_{det} = BCE(y_{det}, p_{det})$$
$$\mathcal{L}_{loc} = -\frac{1}{N}\sum_{i=1}^{N}\left[\begin{matrix} w^+ y_{loc,i}\, log(p_{loc,i}) + \\ + (1 - y_{loc,i})\, log(1 - p_{loc,i}) \end{matrix}\right]$$

For graph-based models, the positive node-class weight $w+ = 3.0$ compensates for the imbalance between healthy and leaking nodes in the localization task.

Each instance of the model $M_{\theta,G}$ is trained independently for a specific network G, as the input dimensionality N and the topology A are constant for a given network.

DATA

This study utilized a collection of water distribution network models in the EPANET format (.inp), sourced from two repositories: (i) the EPANET Input Files for 7 Networks dataset [2], and (ii) the EPANET Example Networks repository maintained by the Open Water Analytics community [13]. The analyzed networks differ in their hydraulic parameters (e.g., demands, elevations, pipe roughness, and installed elements) as provided in the source EPANET networks. Only networks for which WNTR could successfully load the EPANET file and complete the full 24-h simulation were retained (35 networks; Table 1). Graph-based models were further evaluated on a subset of 31 networks due to the node-count limits reported in Table 4. To illustrate the diversity of network topologies (small, medium, and large systems), a set of maps was automatically generated based on the [COORDINATES] section in the.inp files. Table 1 summarizes all the water

Table 1. Water distribution networks used in the study, presented in a compact two-panel layout.

Network	Junc.	Source	Network	Junc.	Source
1_Hanoi	32	[2]	NET1emit	11	[13]
2_Extended_Hanoi	32	[2]	NET1negemit	11	[13]
4_ZJ	114	[2]	NET1np	10	[13]
7_KLmod	936	[2]	NET1para	11	[13]
2PRVs	5	[13]	Net1_EPANET_EXAMPLE_1s	11	[13]
2PRVs-a	6	[13]	Net3	97	[13]
6848-68nodes	70	[13]	Net3_EPANET_EXAMPLE_No_Demand_Change	97	[13]
anytown-3	25	[13]	Net3_BWSN-2__Morph_Error_Free_1s-WQ*	12527	[13]
cheung	5	[13]	NW_Model*	8569	[13]
Elhay-ZeroFlow	8	[13]	NW_Model1*	8569	[13]
example1	11	[13]	sampletown	8	[13]
example1_2comp	11	[13]	simpson_test	6	[13]
example1_LIFO	11	[13]	simpson_test_2-12	6	[13]
GES4-9	12	[13]	st-net3	11	[13]
mod_anytown-3	25	[13]	st-net3-3	11	[13]
NET1	11	[13]	todinitest	5	[13]
NET1-3	11	[13]	wolf-3*	1786	[13]
NET1cv	11	[13]			

Note: *networks that were included in the study but excluded from the evaluation of the graph-based models (GNN, DGT, STGT) due to the node-count scalability limits reported in Table 4 ($N \leq 2000$ / 1500 / 1000, respectively). In addition to the final 35 networks listed in Table 1, we downloaded further EPANET.inp files from the referenced repositories (e.g., 2Tanks, 5_Balerma, 6_Rural, exnet-3H, exnet-3Q, ky2, test101), but excluded them because WNTR/EPANET parsing or the 24-h simulation did not complete successfully.

distribution networks utilized in this study, along with their node counts and sources

For each network, a total of 2000 leak scenarios were generated using the WNTR library. Scenario-level leak/no-leak classification is not evaluated in this paper. The study focuses on spatial localization of leaks. Every simulation covered a 24-hour horizon with a 15-minute hydraulic time step (96 time steps). In each scenario, a subset of junctions was sampled and assigned pressure-dependent emitters; the leak-node fraction r_s varied linearly from 10% to 50%, and each leak remained active throughout the simulation. Scenario parameters (leak-node sets and leak areas) were generated deterministically using NumPy's $\text{default}_{\text{rng}}$ with a fixed seed ($\text{dev}_{\text{seed}} = 42$). The independent evaluation set was generated with the same script using a different fixed seed ($\text{eval}_{\text{seed}} = 123$). Parameters were precomputed for all scenarios prior to multiprocessing, ensuring reproducibility independent of the number of worker processes. The broad leak-node range was adopted as a controlled stress test across anomaly densities and should not be interpreted as a direct model of the typical number of simultaneous failures in an operational utility. Because the leak-node fraction was intentionally high (10–50%) as a controlled stress test, the hydraulic response may reflect the superposition of many simultaneous anomalies. This can reduce the uniqueness of node-specific pressure signatures and make the localization problem less separable, potentially biasing some models toward more spatially diffuse leak predictions.

The generated dataset for each network (2000 leak scenarios; $\text{dev}_{\text{seed}} = 42$) was divided into three disjoint subsets: training (70%), validation (15%), and internal test (15%). The split was performed at the scenario level by a random permutation of scenario indices and fixed ratios. In the current implementation, the split seed is not fixed/recorded; therefore, the exact membership of train/validation/test may vary across runs, although the split ratios are constant. The validation subset was used for model selection, learning-rate scheduling, and early stopping. In the reported benchmark, the localization decision threshold was not tuned on the validation set; a single fixed threshold was

used to keep a strictly comparable operating point across all models and networks. In the reported benchmark, the localization decision threshold was not tuned on the validation set; a single fixed threshold was used to keep a strictly comparable operating point across all models and networks:

$$n_{val} = \lfloor 0.15n \rfloor, n_{test} = \\ = \lfloor 0.15n \rfloor, n_{train} = \\ = n - n_{val} - n_{test} \quad (9)$$

Training progress and model selection were based exclusively on the training and validation subsets. The internal test subset was retained for preliminary verification after model selection and was not used for gradient updates. After training and model selection were completed, an independent evaluation set of 2000 newly generated leak scenarios per network was created with a separate generation seed ($eval_{seed} = 123$). This evaluation set was not used for training, validation, hyperparameter tuning, threshold selection, or model selection. Unless stated otherwise, the final results in this paper refer to this independent evaluation set.

For reproducibility, Table 2 summarizes the key configuration used for scenario generation, dataset splitting, and aggregation.

RESEARCH ENVIRONMENT

The experiments were conducted in an Ubuntu 25.10 environment on a workstation equipped with an AMD Ryzen 7 5700X3D processor, 32 GB of RAM, and an NVIDIA GeForce RTX 4060 Ti graphic accelerator (8 GB VRAM). The models were implemented in Python (v3.13) using the PyTorch library (v2.6) with CUDA acceleration support, the WNTR simulation environment, and the Scikit-learn package.

All models were trained in a per-network scheme: a separate model was fitted for each analyzed network. This design evaluates architectural behavior across diverse topologies but does not test cross-topology generalization. The input to each model was a nodal pressure-head time series represented as $X \in \mathbb{R}^{(T \times N)}$.

Here, $T = 96$ denotes the number of time steps and N is the number of nodes in a given network. The models jointly performed scenario-level detection and node-level localization using the objective:

$$\mathcal{L} = w_{det}\mathcal{L}_{det} + w_{loc}\mathcal{L}_{loc} \quad (10)$$

The weights were $w_{det} = 1$ and $w_{loc} \in \{2.0, 8.0\}$, depending on the architecture (Table 4). For graph-based models, the localization loss additionally used $pos_{weight} = 3$ to mitigate node-level class imbalance.

The training process utilized the AdamW optimizer and the ReduceLROnPlateau learning rate scheduler. Early stopping based on validation loss and gradient norm clipping were also employed. GPU-accelerated training was performed using Automatic Mixed Precision (AMP). Due to memory constraints, dynamic batch size reduction was applied whenever a CUDA out-of-memory error occurred.

Table 2. Configuration summary for reproducing the reported aggregates

Item	Value
Scenario generator	WNTR + deterministic parameter sampling (NumPy default_rng)
Scenario generation seeds	dev_seed = 42 (train/val/test generation), eval_seed = 123 (independent evaluation)
Scenarios per network	2000 (development set) + 2000 (independent evaluation set)
Split protocol (development set)	Scenario-level split into 70%/15%/15% after random permutation
Split seed	Not fixed/recorded in the current implementation
Simulation horizon	24 h, Δt = 15 min (96 time steps)
Leak-node fraction	10–50% of junctions (linear schedule across scenarios)
Localization threshold	Fixed τ_loc = 0.2 (no tuning on evaluation set)
Networks used	35 networks listed in Table 1 (successful WNTR load + simulation)
Networks per model	CNN/LSTM: 35; GNN/DGT/STGT: 31 (Table 1 without "*")
Graph-model node limits	GNN: N ≤ 2000; DGT: N ≤ 1500; STGT: N ≤ 1000

Note: Scenario parameters are precomputed before multiprocessing, ensuring reproducibility independent of the number of worker processes.

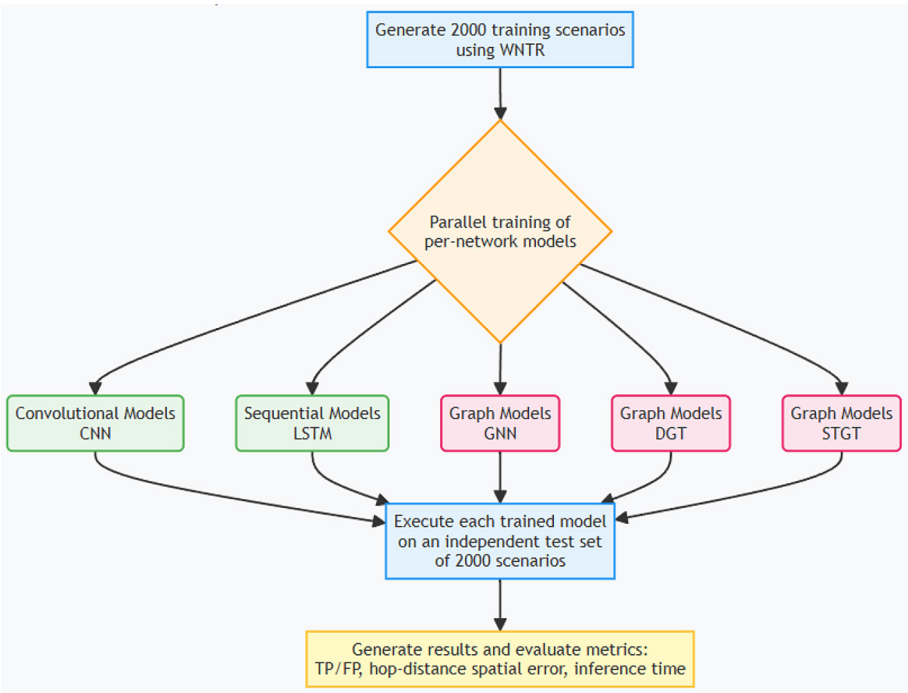


Figure 1. A flowchart representing the operation of the algorithm that creates anomaly detection models in a loop for each individual water distribution network out of 35 available

Table 3. Common training settings for all models

Element	Value
Input representation	Pressure matrix $X \in R^{T \times N}$
Number of time steps	T = 96 (interpolation to a fixed length)
Tasks	Localization (per-node); detection head used as auxiliary confidence (not evaluated as leak/no-leak classification)
Optimizer	AdamW
LR Scheduler	ReduceLROnPlateau(monitor:val_loss,factor=0.5,patience=7,min_lr=10^{-6})
Early stopping	patience: 15 (CNN/LSTM) or 25 (GNN/DGT/STGT)
Gradient clipping	clip_grad_norm: 1.0 (CNN/LSTM) or 0.5 (GNN/DGT/STGT)
Mixed precision	AMP enabled (use_amp=True)
DataLoader	num_workers=0, pin_memory=False

EVALUATION PROTOCOL

Evaluation was performed on the independent evaluation set. The set was not used during training, model tuning, or threshold selection. Inputs were standardized using the scaler saved for the corresponding network, ensuring consistent transformation between training and evaluation. Because all scenarios contain leaks, detection is not evaluated as a binary leak/no-leak task in this study.

Node-level localization. The localization head returned $p_{loc} \in [0,1]^N$. Node j was classified as leaking when $p_{loc,j} \geq \tau_{loc}$, with $\tau_{loc} = 0.2$ in the reported experiment. A single fixed threshold was used in the main benchmark to ensure strict comparability across architectures and networks. As an additional analysis, we report threshold sensitivity for the graph-based models on selected networks (Table 7). Systematic per-network threshold selection on the validation set remains a deployment step and is left for future work.The fixed threshold was chosen to keep a strictly comparable operating point across architectures and networks. Threshold selection per network/model based on the validation split is considered a deployment step and is therefore left for future work. Node-level TP, TN, FP, and FN counts were calculated against the ground-truth leak-node vector and aggregated into

Table 4. Hyperparameters and architecture components for analyzed models

Parameter	CNN	LSTM	GNN	DGT	STGT
Epochs	100	100	150	150	150
Batch size	32	32	16	4	4
Learning rate	10^{-3}	10^{-3}	$3x10^{-4}$	$3x10^{-4}$	$3x10^{-4}$
Weight decay	10^{-5}	10^{-5}	10^{-4}	10^{-4}	10^{-4}
Dropout	0.3	0.3	0.3	0.2	0.2
w_{det}	1	1	1	1	1
w_{loc}	2	2	8	8	8
pos_weight	-	-	3	3	3
Main components	ResNet1D+SE	BiLSTM+attn	SE-GCN+BiGRU+attn	SE-GCN+Transformer+DDPM	SE-GCN+spatial/ temporal attn+fusion
N limits	-	-	N ≤ 2000	N ≤ 1500	N ≤ 1000

Note: For GNN, DGT, and STGT models, the N limits were applied due to the memory cost of attention mechanisms (scaling with B x N and $O(N^2)$ respectively), which in practice limits the maximum network size processable on the utilized GPU. The impact of network size on leak localization effectiveness for individual architectures is analyzed further in this paper.

node recall (TPR), node false-positive rate (FPR), and node false-negative rate (FNR). The mean FP hop-distance was calculated from each falsely flagged node to the nearest true leak using breadth-first search. It measures the spatial concentration of false alarms and is not equivalent to the distance between a single predicted leak location and the The mean FP hop-distance is computed only for scenarios with at least one true leak node ($|S_s|>0$). In our dataset all scenarios are leak scenarios; therefore the metric is well-defined for all evaluated cases. Because the metric is computed over the set of false-positive nodes, models with vastly different FP counts may yield hop-distance values that reflect different sampling regimes. Hop-distance should therefore be interpreted jointly with FPR/FP volume.true leak.

Inference time was defined as the computation time required to produce both detection and localization outputs for one scenario. It was measured using time.perf_counter(); torch.cuda.synchronize() was applied for GPU calculations to account for asynchronous execution.

Global values were calculated as network-wise averages weighted by the number of evaluation scenarios. Due to node-count limits, not all graph-based models were available for every network; their aggregates therefore cover only the networks on which the corresponding architecture could be evaluated.

To ensure research reproducibility, mechanisms for saving checkpoints and logging learning metrics were implemented. For graph-based architectures, adjacency matrices were constructed and normalized directly from the source.inp files. Due to the significant memory footprint of attention mechanisms in STGT models, evaluation on the largest network topologies (exceeding the predefined node limit) was restricted to highly scalable sequential and convolutional architectures.

Source code is available at https://github.com/SoBISKUP/wds_anomaly_detection

RESULTS

This section presents the performance of five machine learning architectures (CNN, LSTM, GNN, DGT, and STGT). Table 5 reports node-level localization measures only: node recall, node FPR, node FNR, mean FP hop-distance, and inference time. These values must not be interpreted as scenario-level leak-detection rates. Scenario-level detection metrics are not reported because all evaluation scenarios contain leaks. In preliminary tests (not included here), the detection head saturated near ceiling across models; therefore, we focus on localization selectivity.

Because the per-network standard deviations in Table 5 are comparable to the means, the average values alone can be misleading. Therefore, we additionally report medians and interquartile ranges (IQR) of the per-network metrics in Table 6. This summary highlights the

Table 5. Node-level localization performance on the independent evaluation set (aggregated across networks)

Model	# networks	Node recall [%]	Node FPR [%]	Node FNR [%]	Mean FP hop-distance	Inference time [s]
CNN	35	95.82 ± 4.72	25.87 ± 28.18	4.18 ± 4.72	1.43 ± 0.34	5.02 ± 0.56
LSTM	35	93.44 ± 10.96	21.96 ± 26.22	6.56 ± 10.96	1.26 ± 0.27	2.05 ± 0.28
GNN	31	100.00 ± 0.00	79.82 ± 11.06	0.00 ± 0.00	1.55 ± 0.18	2.58 ± 1.75
DGT	31	100.00 ± 0.00	100.00 ± 0.00	0.00 ± 0.00	1.69 ± 0.15	4.68 ± 3.83
STGT	31	100.00 ± 0.00	100.00 ± 0.00	0.00 ± 0.00	1.69 ± 0.15	10.57 ± 29.62
Top-K heuristic (mean–min, oracle K)	35	42.56 ± 12.97	20.29 ± 5.30	57.44 ± 12.97	—	—

Note: Values are mean ± standard deviation across networks. The independent evaluation set contained 2000 scenarios per network. Graph-based models were unavailable for some networks because of scalability constraints. At such an over-predictive extreme (FPR ≈ 100%), derived spatial metrics (e.g., mean FP hop-distance) are dominated by network geometry and leak-node density rather than by model selectivity. Hence, hop-distance values for DGT/STGT at this operating point should be interpreted with caution. Top-K is a non-deep baseline that ranks nodes by a simple pressure-drop score (mean minus minimum pressure over the 24-h series) and predicts the K highest-ranked nodes, where K equals the ground-truth number of leaking nodes in the scenario (oracle K available in simulation). Mean FP hop-distance and inference time are not reported for this baseline. For DGT/STGT, the reported 100.00 ± 0.00% recall/FPR reflects a degenerate all-positive prediction regime (effectively labeling all nodes as leaking), hence the zero variance is a ceiling effect rather than exceptional stability.Top-K is a non-deep baseline that ranks nodes by a simple pressure-drop score (mean minus minimum pressure over the 24-h series) and predicts the K highest-ranked nodes, where K equals the ground-truth number of leaking nodes in the scenario (oracle K available in simulation). Mean FP hop-distance and inference time are not reported for this baseline.

strong cross-topology heterogeneity, particularly for node-level FPR. Compared with the Top-K baseline (oracle K; available only in simulation), CNN/LSTM achieve substantially higher node--level recall while maintaining a similar order of node-level FPR, indicating that the learned models capture informative pressure patterns beyond simple pressure-drop ranking.

The GNN, DGT, and STGT models achieved a node-level recall of 100%, meaning that all genuinely leaking nodes were identified. However, this was accompanied by very high node-level false-positive rates—from 79.82% for GNN to 100% for DGT and STGT indicating severe over-prediction of leak locations. This substantially limits the practical utility of these architectures as precise node-level localization tools in the investigated configuration. For DGT and STGT, the combination of 100% node-level recall and 100% node-level FPR indicates a degenerate operating point: under the reported configuration these models effectively label all nodes as leaking. Therefore, DGT/STGT did not achieve a usable localization regime in this benchmark, and their results should not be interpreted as evidence of accurate localization.

The LSTM and CNN models achieved node-level recalls of 93.44% and 95.82%, respectively, with false-positive rates of approximately 22–26%. Although substantially lower than those of the graph-based models, these false-positive rates should not be described as operationally acceptable without validation under real deployment conditions.

Table 6. Per-network distribution summary (median and interquartile range) of node-level localization metrics

model	n_ networks	recall_ median	recall_ q1	recall_ q3	fpr_ median	fpr_q1	fpr_q3	hop_ median	hop_ q1	hop_ q3	time_ median	time_ q1	time_ q3
CNN	35.00	97.63	91.33	99.54	12.52	4.15	47.21	1.39	1.22	1.67	5.04	5.01	5.13
LSTM	35.00	97.84	89.35	99.04	10.89	2.05	28.77	1.11	1.00	1.48	2.01	1.99	2.13
GNN	31.00	100.00	100.00	100.00	76.00	76.00	86.12	1.55	1.51	1.61	2.30	2.28	2.36
DGT	31.00	100.00	100.00	100.00	100.00	100.00	100.00	1.70	1.63	1.73	4.08	4.03	4.16
STGT	31.00	100.00	100.00	100.00	100.00	100.00	100.00	1.70	1.63	1.73	5.36	5.32	5.43

Note: Medians and IQR (Q1–Q3) are computed across networks. CNN and LSTM are summarized over 35 networks, whereas graph-based models are summarized over 31 networks due to the node-count limits in Table 4.

To make the degenerate behavior of the graph-based models more explicit, we provide a short threshold-sensitivity analysis on a representative subset of networks (Net3 and 7_KLmod; 300 scenarios per network). Table 7 reports node-level recall and node-level FPR for several localization thresholds. This analysis is illustrative and does not replace the main benchmark in Table 5, which uses a single fixed threshold for strict comparability across all networks. We emphasize that Table 7 is a sensitivity check (multiple thresholds), not a validation-optimized threshold selection procedure.

To visualize the heterogeneity indicated by the large standard deviations in Table 5. Figure 2 shows the per-network distribution of node-level FPR for each architecture.

Each box summarizes the distribution of per-network node-level FPR values (median, IQR, whiskers; points denote outliers). CNN and LSTM include 35 networks, while graph-based models include 31 networks due to the node-count limits in Table 4. The wide spread confirms strong cross-topology heterogeneity, and the near-ceiling FPR for DGT/STGT reflects the degenerate operating regime discussed in the text.

The heatmaps in Figures 3 and 4 show how node recall and node FPR vary across networks. They illustrate localization behavior rather than scenario-level leak detection and should be interpreted in conjunction with network size and topology. The next verified parameter was the localization precision of false predictions, measured as the average topological distance (hop-distance) in the graph from the node falsely classified as damaged to the nearest genuinely leaking node. This parameter determines whether the model's prediction errors are randomly scattered across the entire network or clustered in the immediate vicinity of the actual anomaly.

Table 7. Threshold-sensitivity analysis (additional) for graph-based localization on selected networks (Net3 and 7_KLmod)

network	model	threshold	n_scenarios	n_nodes	node_recall_pct	node_fpr_pct	avg_inference_time_s
7_KLmod	DGT	0.20	300	936.00	100.00	100.00	0.11
7_KLmod	DGT	0.50	300	936.00	100.00	100.00	0.11
7_KLmod	DGT	0.80	300	936.00	0.00	0.00	0.11
7_KLmod	DGT	0.90	300	936.00	0.00	0.00	0.11
Net3	DGT	0.20	300	97.00	100.00	100.00	0.01
Net3	DGT	0.50	300	97.00	100.00	100.00	0.01
Net3	DGT	0.80	300	97.00	0.00	0.00	0.01
Net3	DGT	0.90	300	97.00	0.00	0.00	0.01
7_KLmod	GNN	0.20	300	936.00	100.00	99.85	0.08
7_KLmod	GNN	0.50	300	936.00	0.00	0.00	0.08
7_KLmod	GNN	0.80	300	936.00	0.00	0.00	0.08
7_KLmod	GNN	0.90	300	936.00	0.00	0.00	0.08
Net3	GNN	0.20	300	97.00	100.00	92.82	0.02
Net3	GNN	0.50	300	97.00	0.00	0.00	0.02
Net3	GNN	0.80	300	97.00	0.00	0.00	0.02
Net3	GNN	0.90	300	97.00	0.00	0.00	0.02
7_KLmod	STGT	0.20	300	936.00	100.00	100.00	1.10
7_KLmod	STGT	0.50	300	936.00	100.00	100.00	1.10
7_KLmod	STGT	0.80	300	936.00	0.00	0.00	1.10
7_KLmod	STGT	0.90	300	936.00	0.00	0.00	1.10
Net3	STGT	0.20	300	97.00	100.00	100.00	0.02
Net3	STGT	0.50	300	97.00	100.00	100.00	0.02
Net3	STGT	0.80	300	97.00	0.00	0.00	0.02
Net3	STGT	0.90	300	97.00	0.00	0.00	0.02

Note: When FPR is near 100%, derived spatial quantities such as mean FP hop-distance are dominated mainly by network geometry rather than model selectivity; therefore, hop-distance is not the focus of Table 7.

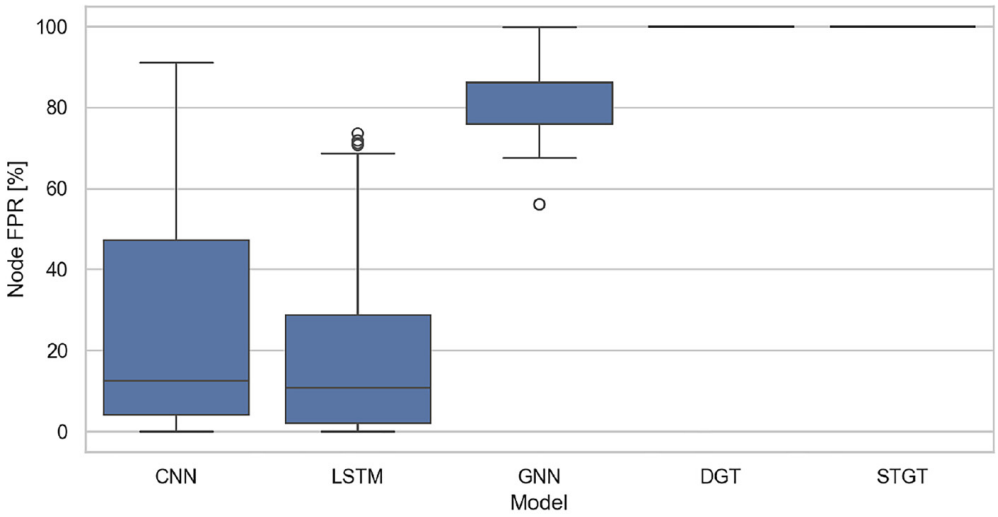


Figure 2. Per-network distribution of node-level false-positive rate (FPR) across models

The LSTM model achieved the lowest mean FP hop-distance (1.26). This indicates that falsely identified nodes tended to cluster near actual leak locations rather than being distributed randomly throughout the network. Such clustering may help reduce the inspection area, but the metric does not mean that the model missed a single true location by an average of 1.26 pipe segments. Graph-based models produced mean FP hop-distances of approximately 1.55–1.69. The applicability of analytical models in proactive early warning systems also necessitates a rigorous approach to computation time. Real-time monitoring systems (e.g., SCADA systems) require the model to process new signals at a frequency corresponding to measurement intervals.

The inference time analysis presented in Table 5 indicates that the leading model in terms of computational efficiency is, once again, the recurrent architecture. The LSTM model executes the full decision process for a single scenario in an average time of 2.05s. The classical convolutional architecture (CNN) requires roughly twice as much time (5.02s). The highest computational burden on the runtime environment is associated with the STGT model (averaging 10.57s, with a very large standard deviation of 29.62s, resulting from the sizes of the largest networks). The computational burden of STGT is consistent with the quadratic $O(N^2)$ cost of its attention mechanism. Among the investigated configurations, LSTM offered the most favorable balance of node-level localization performance and inference time. This supports further validation—not immediate deployment—in monitoring and decision-support settings.

DISCUSSION

The conducted experiments revealed significant differences in the effectiveness of the individual machine learning architectures, highlighting the complexity of the leak detection and localization problem in water distribution networks. These findings enable drawing important conclusions regarding both the theoretical limitations of the investigated models and their practical applicability in early warning systems.

The most striking localization result is the extreme behavior of the graph-based architectures. Their 100% node-level recall, combined with a near-complete lack of selectivity, indicates massive over-prediction rather than flawless scenario-level leak detection.

Two hypotheses may explain this behavior. First, repeated graph propagation can make neighboring node representations overly similar, hindering discrimination between leaking junctions and their healthy neighbors. Second, probability calibration may be inadequate. Because every architecture was evaluated at the same fixed localization threshold $\tau_{loc} = 0.2$, calibration differences may have strongly influenced the observed FPR. This hypothesis should be tested through validation-set threshold optimization and probability calibration diagnostics. In addition, the anomaly density used in the simulations may have contributed to this over-prediction behavior. When a large fraction of junctions leaks simultaneously (10–50%), nodal pressure signatures overlap and become less distinctive, which reduces spatial specificity. In graph-based

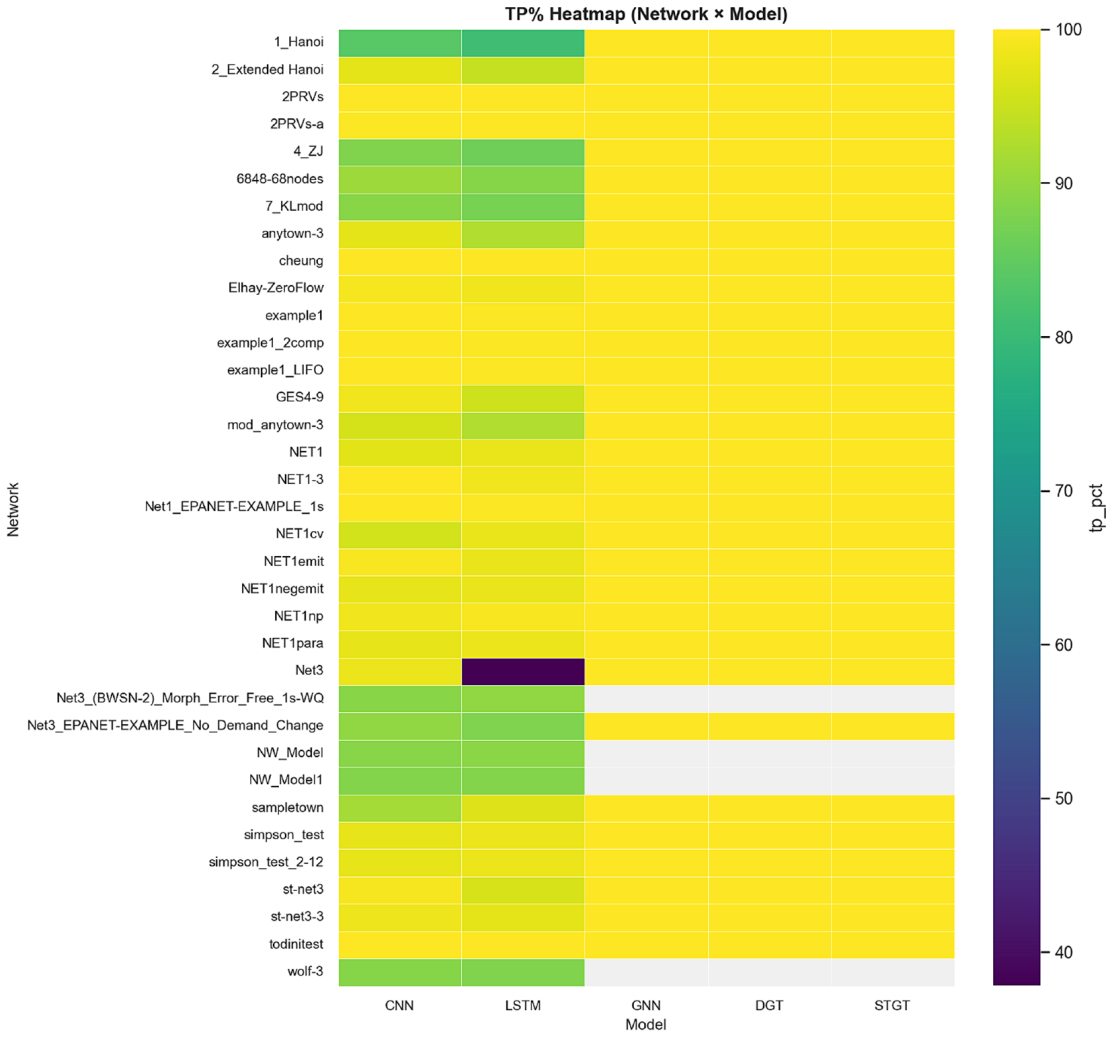


Figure 3. Heatmap of the TP% distribution across networks. Empty cells indicate networks on which the corresponding model was not evaluated due to scalability constraints (node-count limits); these excluded networks are marked in Table 1 (asterisk) and the limits are reported in Table 4

models, neighborhood aggregation can further increase representation similarity (over-smoothing), amplifying the tendency to assign elevated leak probabilities to many neighboring nodes. Therefore, the observed high node-level false-positive rates should be interpreted as a joint effect of the chosen stress-test regime and the investigated model configurations, rather than as an architecture-only limitation. Importantly, this over-prediction should not be attributed to the architectures alone. The deliberately high leak-node prevalence (10–50%) used as a stress test increases the positive-node base rate and reduces the uniqueness of node-specific pressure signatures, which can push models toward spatially diffuse positive predictions.

The CNN and LSTM models learned more selective pressure patterns and achieved node recalls of 93–96%, while their node-level FPRs of approximately 22–26% were substantially lower than those of the graph-based models. These values nevertheless remain high enough to require further reduction before operational use.

From a utility perspective, predictive value depends on the balance between leak sensitivity and the cost of false-alarm verification. Consequently, the graph-based architectures in their current configuration are not yet suitable for direct operational deployment as precise node-level leak-localization tools. Among the investigated architectures, LSTM provided the most favorable balance between localization sensitivity, false-positive rate, spatial concentration of false alarms, and inference time. Its mean FP hop-distance of 1.26 indicates that false alarms often occurred near actual leaks; it does not by

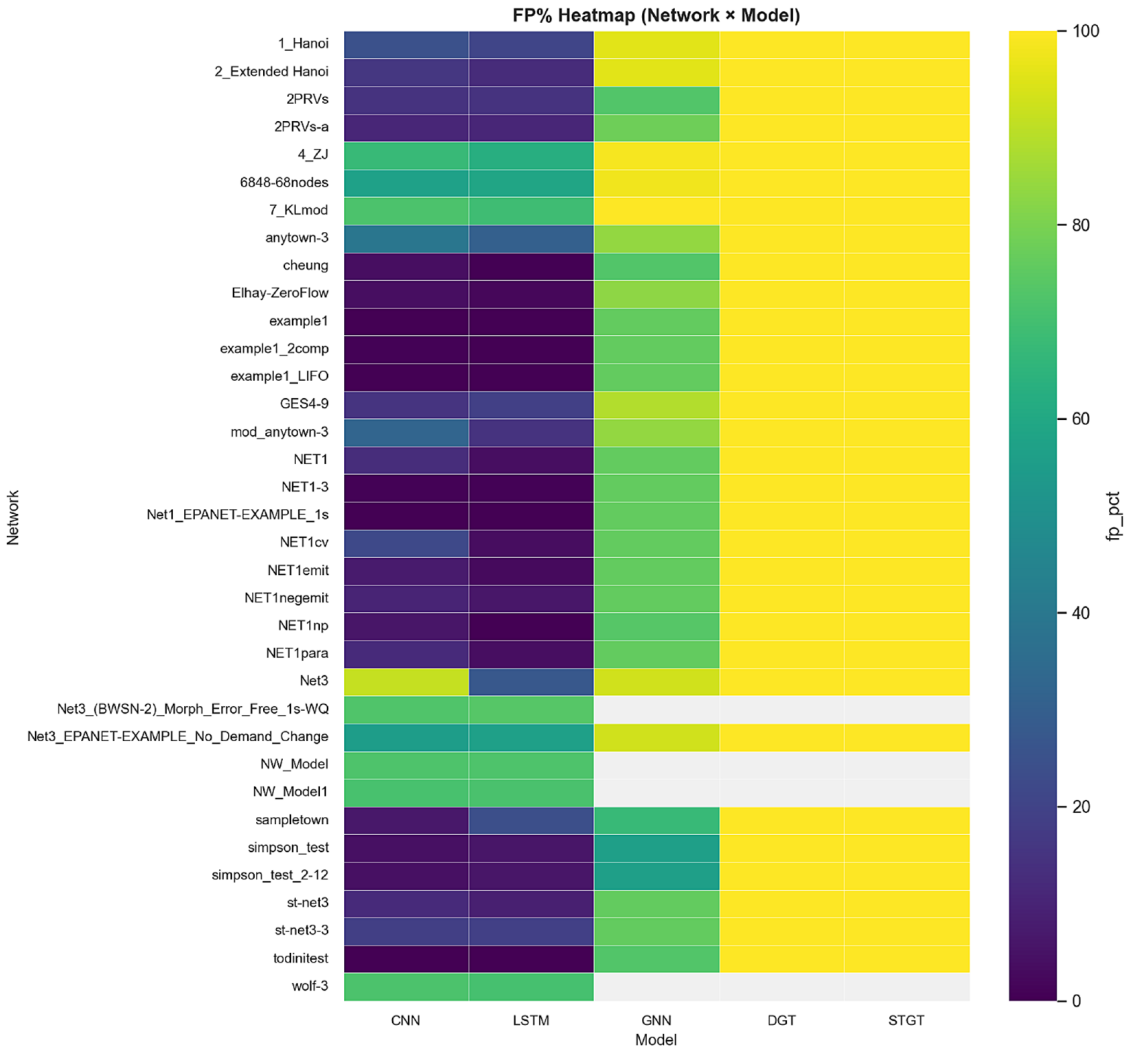


Figure 4. Heatmap of the FP% distribution across networks. Empty cells indicate networks on which the corresponding model was not evaluated due to scalability constraints (node-count limits); these excluded networks are marked in Table 1 (asterisk) and the limits are reported in Table 4

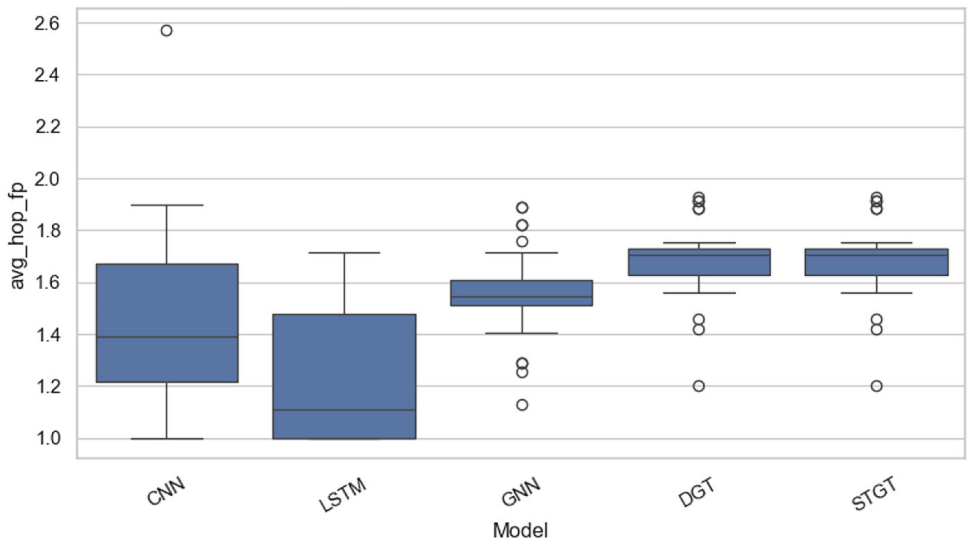


Figure 5. Box plot presenting the spatial localization error (average graph distance to the nearest leak) for false predictions

itself establish operationally sufficient localization accuracy. The approximately 2-second inference time is compatible with common telemetry intervals, making LSTM the strongest candidate for further validation in SCADA-supported leak--monitoring systems.

Several limitations must be acknowledged. The experiments used hydraulic simulations that do not represent the full range of measurement noise, communication failures, sensor faults, and irregular demands found in operational systems. Pressure was available at all modeled nodes, whereas real SCADA deployments typically provide sparse sensor coverage. A key practical limitation of this study is the assumption of full-node pressure observability, whereas real SCADA deployments typically provide measurements at only a small subset of nodes (e.g., 5–10%). Therefore, the reported results should be interpreted as an upper-bound benchmark for pattern learning under complete observability rather than as a ready-to-deploy solution. To adapt the approach to realistic telemetry, future work will evaluate models under sensor subsampling and incorporate strategies for handling missing nodes, including maskin-g-based training, learned spatial imputation, and hybrid SCADA–hydraulic state estimation. Furthermore, models were trained separately for each network, so the study evaluates consistency across topologies rather than generalization to unseen networks. Changes in network topology may therefore require retraining.

In the context of the aforementioned issue with false alarms, the primary limitation of the evaluation was the application of a fixed classification threshold. For future research, the following improvements are recommended:

- Replacing the rigid decision threshold with a dynamic approach by tuning the threshold individually for each network on the validation set based on an application-specific objective (e.g., minimizing FPR at a required recall) and calibration diagnostics.
- Shifting the localization paradigm from threshold-based classification to an approach that returns the top-K most probable leak locations, which completely eliminates the issue of massive false alarms.
- Introducing additional sparsity-enforcing regularization, such as an L_1 penalty on the localization head, to discourage graph-based models from designating an excessive number of nodes as damaged.
- Improving the usability of graph-based models requires moving to an interpretable operating point via validation-based threshold selection and calibration and/or adopting a top-K localization formulation to enforce sparse outputs.

We also consider three practical adaptation paths:

- Sensor masking during training and evaluation, where only a fixed fraction of nodes is observed and the model receives an explicit binary observation mask.
- Two-stage pipelines combining a graph-based imputer/state estimator with an LSTM detector-localizer operating on reconstructed pressure fields.
- Sensor placement and robustness studies to identify which 5–10% of nodes provide sufficient observability for reliable detection and coarse localization.
- Tune τ_loc per network on the validation set using an application-specific operating point, e.g., selecting the smallest τ_loc such that node-level recall $\geq R^*$ and then minimizing FPR. This would allow graph-based models to trade off recall for selectivity in a controlled manner.

CONCLUSIONS

The primary objective of this study was to evaluate machine learning architectures for node-level leak localization in water distribution networks. The experiments covered 35 networks of varying topology and complexity and compared CNN, LSTM, GNN, DGT, and STGT models under a per-network training scheme. Unlike prior comparisons focused mainly on classification rates, hop-distance characterizes how far false alarms spread in the network graph, which is directly linked to operational inspection burden. The conducted analysis of the results allows for the formulation of the following main conclusions:

1. Sequential-model performance: LSTM provided the most favorable overall balance, achieving a node-level localization sensitivity of 93.44%, a node-level false-positive rate of 21.96%, and the lowest mean FP hop-distance (1.26). The latter shows that false alarms tended to occur near true leak nodes. It is not a direct point-localization error.

2. Computational efficiency: LSTM processed one scenario in an average of 2.05 s, compared with 5.02 s for CNN and 10.57 s for STGT. STGT also exhibited high variability across network sizes, consistent with the quadratic cost of attention.
3. Graph-model over-prediction: GNN, DGT, and STGT achieved 100% node-level recall but also produced node-level FPRs of approximately 80–100%. This severe over-prediction demonstrates the need for probability calibration, validation-based threshold optimization, and stronger sparsity constraints before operational application.

At the current stage, the results do not provide a basis for direct operational deployment because the study relies on simulated data, full--node pressure availability, and network-specific training. Within this simulation-only benchmark (full-node observability and per-network training), LSTM-based models provided the most favorable trade-off among the evaluated architectures and are the strongest candidate for further validation. The presented results should be treated as an upper-bound comparison and do not by themselves justify operational deployment without validation on real, sparse SCADA data.

Acknowledgements

I would like to express my sincere gratitude to Dr. Ariel Antonowicz for pointing out inspiring research directions and providing invaluable methodological support throughout the course of this study. This work was supported by the Poznań University of Technology under Grant 0311/SBAD/0778.

REFERENCES

1. Bełcik M., Nowakowska M. Application of artificial neural networks to predict water demand for the purpose of optimizing network operation. *INSTAL*. 2026; (7): 27–31. https://doi.org/10.17512/INSTAL.2026.07.02. (in Polish).
2. Bi W., Dandy G., Maier H. EPA NET Input Files for 7 Networks. figshare; 2015.
3. Biström C. Anomaly detection methods in water infrastructure. Bachelor's thesis; 2025.
4. Bylka J., Ramm K. New requirements for the quality of water supply services regarding water losses. *Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie.* 2021; 4(994): 47–65. (in Polish).
5. Dzimińska P., Stańczyk J., Pałczyński K., Licznar P., Andrysiak T. The potential of computational intelligence algorithms in the interpretation of distributed water meter data. *Instal.* 2024; (1): 36–43. https://doi.org/10.36119/15.2024.1.5. (in Polish).
6. Flia A., Gęca W., Iwanek M., Kowalski D. Analysis of the volume of water losses in two rural waterworks. 2023. https://gazwoda.pl/wp-content/uploads/Openaccess/2023/101519917202312.pdf. (in Polish).
7. Głomb P., Romaszewski M., Cholewa M., Koral W., Madej A., Skrabski M., Kołodziej K. Detection of emergent leaks using machine learning approaches. *Progress in Polish Artificial Intelligence Research 4.* 2023; 193–194. https://doi.org/10.34658/9788366741928.30.
8. Statistics Poland. Municipal infrastructure – water supply and sewage systems in 2024. 2024. (in Polish).
9. Hanifa R., Cha M., Kang W., Yu J., Kim K.-J., Yun Y.-M., Kim S. Development of water quality analysis for anomaly detection and correlation with case studies in water supply systems. *Electronics.* 2025; 14(10): 1933. https://doi.org/10.3390/electronics14101933.
10. Kowal B., Organiściak P., Kuraś P., Masłoń A., Wadiak B. Machine learning in forecasting the amount of wastewater for the Rzeszów city wastewater treatment plant based on weather conditions. *Pomiary Automatyka Robotyka.* 2025; 29(1): 81–90. https://doi.org/10.14313/PAR_255/81. (in Polish).
11. Kowalski D., Kowalska B., Suchorab P. Proposal of a diagnostic method for the water supply network. *Instal.* 2021; (7–8): 56–60. https://doi.org/10.36119/15.2021.7-8.8. (in Polish).
12. Kwietniewski M. Failure rate of water supply and sewage infrastructure in Poland in the light of operational research. Warsaw University of Technology; 2010. https://repo.pw.edu.pl/docstore/download/WUT395376/Kwietniewski_M_Awaryjnosc_infrastruktury_wodociagowej_i_kanalizacyjnej_w_Polsce_w_swietle_badan_eksploatacyjnych.pdf. (in Polish).
13. OpenWaterAnalytics. EPANET Example Networks. GitHub; 2026. https://github.com/OpenWaterAnalytics/epanet-example-networks. Repository accessed on 2026-07-23 (commit: 4f0ceb2641c35adece59145881f10232c40c26b7).
14. Pandey P., Mücke N.T., Jain S., Ramachandran P., Bohté S.M., Oosterlee C.W. Machine learning-based digital twin for water distribution network anomaly detection and localization. *Engineering Proceedings.* 2024; 69(1): 201. https://doi.org/10.3390/engproc2024069201.

15. Pataca da Costa A.M. Unsupervised anomaly detection in water system networks using recurrent neural networks. Master's thesis; 2020.
16. Przystałka P. A method for leak detection in water supply networks using NLARX models. *Modelowanie Inżynierskie.* 2013; 15(46): 94–102. (in Polish).
17. Sebestyen G., Hangan A., Czako Z. Anomaly detection in water supply infrastructure systems. 23rd International Conference on Control Systems and Computer Science (CSCS). 2021; 349–355. https://doi.org/10.1109/CSCS52396.2021.00064.
18. Wen H., Gu X., Han J., Huo X., Ye Z., Lu W. A review of anomaly detection methods for smart water distribution networks. Proceedings of the International Conference on Artificial Intelligence and Advanced Autonomous Systems. 2026; 69-75. https://doi.org/10.1145/3813723.3813732.
19. Zdebik D., Wyrwik S. Management of the water distribution system using IT tools. 2020; 5: 38–41. (in Polish).