

An Intelligent Machine Learning Framework for Adaptive Urban Water Management: A Hybrid Deep Learning and Optimization Approach for Demand Forecasting, Leak Detection, and Water Quality Assurance in Smart Cities

Rahul Dwivedi¹, Dr.Binod Kumar^{2*} Dr.Deepak kadve³

¹SVIMS Research Scholar,

Email: rahul10jan@gmail.com

²Department of Computer Applications, JSPM's Rajarshi Shahu College of Engineering, Pune – 411033, Maharashtra, India.

Email: binod.istar.1970@gmail.com

³Department of Computer Applications, JSPM's Rajarshi Shahu College of Engineering, Pune – 411033, Maharashtra, India.

Email: deepkadv55@gmail.com

Corresponding author :- **Dr.Binod Kumar**

Abstract

A combination of urban population growth, degradation of distribution infrastructure, and climatic variability is putting tremendous pressure on urban water supply systems, since the traditional rule-based and human-operated systems cannot forecast changes in consumption patterns, detect losses and ensure water safety. The paper proposes a novel solution that combines machine learning and optimization algorithms and allows using sensors, edge and cloud computing and multiple learning models in one decision-making system. This solution addresses three interrelated problems usually considered separately by previous studies that applied single-technique approach to solving the problem: short-term water demand forecasting, detection of the initial phase of leaks and anomalies in distribution network, and risk assessment for water quality continuously. Instead of choosing one algorithm as it is done in previous studies, this research will compare tree ensemble learners (Random Forest, XGBoost, LightGBM, CatBoost), sequential learners and attention-based deep learners (LSTM, GRU, Bi-LSTM, Transformer), as well as graph neural networks, which can take into account topology of the distribution network, and will provide a procedure for selecting a model for solving a certain problem instead of supposing the superiority of one of those approaches. The methodology section outlines the process of acquiring, preprocessing, feature engineering, training, validation, and optimization of the models using publicly available hydrological, meteorological and utility data. Formulas of the forecasting, classification, and optimization models as well as the metrics used for their evaluation (MAE, RMSE, MAPE, precision, recall, F1-score) are provided. As this paper is not about conducting empirical research, this paper does not include any assumptions about the performance of the framework, yet the paper provides an explicit protocol for the falsifiable validation of the framework. This paper discusses the practical implementation, scalability in smart cities, and the possibility of integration of digital twins, federated learning, and explainable AI within the proposed framework. The contribution of this paper is the introduction of an architectural and methodological framework based on literature review which can be built upon by researchers and utilities in order to create integrated water management systems in cities.

Keywords: Urban Water Management · Machine Learning · Deep Learning · Smart Cities · Water Demand Forecasting · Leak Detection · Water Quality Prediction · Optimization

How to cite this article: Rahul Dwivedi¹ Dr.Binod Kumar^{2*} Dr.Deepak kadve³. An Intelligent Machine Learning Framework for Adaptive Urban Water Management: A Hybrid Deep Learning and Optimization Approach for Demand Forecasting, Leak Detection, and Water Quality Assurance in Smart Cities. Int J Drug Deliv Technol. 2026;16(84s):1669-1684. DOI: 10.25258/ijddt.16.84s.169.

1. Introduction

Water is the single infrastructural resource whose failure most immediately threatens public health, economic activity, and social stability in a city. Yet the systems that capture, treat, distribute, and account for urban water remain, in most of the world, substantially unchanged in operating logic since the twentieth century: fixed maintenance schedules, manually read meters, and reactive rather than predictive fault response. The scale of the resulting gap is stark. The United Nations World Water Development Report 2026 finds that 2.1 billion people still lack safely managed drinking water, and that women and girls collectively spend an estimated 250 million hours every day collecting water where piped supply is unreliable or absent. The UN's Sustainable Development Goal 6 monitoring shows that the global

urban population exposed to water scarcity is projected to roughly double, from 930 million in 2016 to between 1.7 and 2.4 billion by 2050, with global water stress holding at 18 percent since 2015 and one in ten people now living under high or critical water stress. Even in well-resourced utilities, non-revenue water that is produced but lost to leakage, metering error, or theft before it reaches a paying customer commonly reaches 1530 percent of total supply in developing-country distribution networks, a loss equivalent to the annual consumption of roughly ten million people.

These pressures are compounding rather than independent. Urbanization continues to concentrate demand spatially and temporally: the global urban population grew from 0.8 billion (29.6 percent of the

world total) in 1950 to 4.4 billion (56.2 percent) in 2020 and is projected to reach 6.7 billion (68.4 percent) by 2050, with industrial and domestic urban water demand expected to rise 5080 percent over the next three decades. Climate change is simultaneously altering the timing and volume of precipitation, intensifying both drought and flash-flood risk in the same urban catchments, while aging pipe networks in many mid- and high-income cities were engineered for twentieth-century demand profiles and are now operating well past their design life. Traditional management solutions employed for conventional water utilities, consisting of regular inspections, static hydraulic modeling done from time to time, and demand forecast based on historical average numbers were not meant to handle this peculiar combination of increased rate of demand growth, weather unpredictability, and old infrastructure.

From the technological perspective, the smart city solution could be considered a potential way to solve the problem. With the help of cheap IoT sensors networks, smart meters, control systems capable of integrating with SCADA, and weather information provided by either satellites or ground stations, it is now technically feasible to monitor urban water systems constantly rather than periodically. The issue of dealing with this vast amount of data that is both noisy and informative at the same time, when analyzed as predictions, anomalies, and decisions, became a problem to which ML and DL could be applied. During the last decade, there was a lot of research work done on applying ML and DL to different aspects of urban water system problem, namely demand forecasting, leaks and burst detection, water quality classification, flood risk prediction, and the results seem to be quite encouraging (see Section 2). The last missing element in this picture is an architecture integrating all these proven approaches in one system.

This paper attempts to bridge this gap. In this paper, an approach to urban water management through machine learning, in which demand forecasting, leak detection and classification, and water quality risk classification are considered as separate components of one system, rather than separate research areas, and the place of this system within the architecture of a smart city, comprising IoT sensors, edge pre-processing, cloud-based model serving, GIS integration, and decision-making dashboard for utilities' management and, when necessary, for citizens is discussed. This paper is built on the previous research, in which the application of SVM for demand forecasting and water quality classification was proposed, as well as the use of LSTM networks for anomaly detection in a single-city network. However, current research goes much further from the previous research, not restricting itself to only two models and offering the approach for comparing nine different model families (Section 7), offering mathematical objective function for each of the problems (Section 8), and adding digital twins, federated learning, and explainable AI into the system. Structure of the rest of the paper. A brief overview of the literature concerning the problem is provided in Section 2, and Section 3 emphasizes the knowledge

gap within this area. The measurable objectives of this research are stated once again in Section 4. The multi-layer architecture to be designed is outlined in Section 5, and the overall process of collecting data and deploying models is described in Section 6. Potential models to be utilized to learn machine-learning and deep-learning algorithms for each element of the architecture are provided in Section 7. Mathematical model design of the proposed methods is outlined in Section 8. Appropriate datasets for the application of the approach are provided in Section 9, and the experiment design for the implementer of the system is provided in Section 10. Simulation results are not given here; rather, the performance evaluation criteria and metrics are stated in Section 11. Sections 13-14 discuss expected behaviour, practical applications, and limitations, and Section 15 concludes with directions for future work, including digital twins, federated learning, and generative AI.

The contribution of this paper is therefore best understood as architectural and methodological rather than purely empirical: it specifies, with mathematical and procedural precision, a framework that unifies previously siloed machine-learning techniques for urban water management into a single deployable system, and it lays out exactly how that system should be evaluated once implemented a prerequisite for the credible empirical studies that should follow.

2. Literature Review

The application of machine learning to urban water management has grown substantially since the early smart-water-grid studies of the late 2010s. This section organizes the literature by functional problem demand forecasting, leak and anomaly detection, water-quality prediction, and integrative technologies (digital twins, federated learning, explainable AI, flood prediction) before synthesizing the comparative findings in Table 1.

2.1 Water Demand Forecasting

Early smart-water-grid research (Li et al., 2018) established the conceptual case for integrating advanced analytics, including machine learning, into real-time monitoring and control of urban water distribution. Machine learning applications have been recommended in relation to water management using machine learning algorithms to increase efficiency in water distribution and detect leakage as suggested by Chen and Zhang (2019). The application of machine learning has been proposed in predicting water quality, detecting anomalies, and predicting water demand as noted by Jia & Zeng (2020), where the challenges of data availability and interpretability were identified. There have been advancements in machine-learning comparative benchmarking since the work carried out by Zhang et al. (2022) where several machine learning algorithms were compared in their use for predicting water demand in smart cities. In the more recent data-centric study conducted by Liu, Savic & Fu (2023), systematic improvements in training data rather than adjustments to the model would be more beneficial. The application of machine learning is continually growing into new case study evaluations in varied geographical and climatic environments; for example,

Vieira Filho et al. (2024) compare the application of four machine-learning models for predicting daily water demand in a Brazilian coastal tourist city, while Namdari, Ashrafi & Haghighi (2024) develop a hybrid deep neural network with multichannel input for predicting hourly water demand.

Deep sequential and ensemble hybrid architectures have become the dominant direction for short-term forecasting. Nguyen et al. (2021) demonstrated deep learning for real-time water-consumption prediction in smart cities, and Meng et al. (2021) proposed a hybrid machine-learning approach specifically for real-time urban demand forecasting. Qiu et al. (2020) proposed an ensemble model for demand prediction in smart cities, while Rafique et al. (2021) surveyed the comparative strengths and limitations of the algorithm families used across this literature. Xie et al. (2019) demonstrated real-time forecasting with a Beijing case study, illustrating that model performance is sensitive to local consumption patterns and cannot be assumed to transfer across cities without re-calibration a finding with direct implications for the framework's transfer and retraining strategy (Section 6).

2.2 Leak Detection and Distribution-Network Anomaly Detection

Leak and burst detection is the most extensively benchmarked sub-problem in this literature, in part because pressure and flow sensor data are comparatively cheap to simulate via hydraulic models such as EPANET. Hu et al. (2018) and Yang et al. (2018) established early machine-learning approaches to leakage detection and real-time anomaly detection respectively, and Chou et al. (2019) extended this line of work to pressure optimization as a complementary lever for reducing both energy consumption and pipe failures. Li et al. (2020) applied machine learning specifically to pipe-burst prediction, underscoring the value of early-warning models for infrastructure planning. More recent work has moved toward deep and hybrid architectures: Yu et al. (2024) combined deep learning with kriging spatial interpolation for leak detection and localization, McMillan, Fayaz, and Varga (2024) proposed a domain-informed variational neural network and SVM-based framework to support self-healing distribution networks, and Jun and Jung (2025) evaluated deep-learning leak detection across multiple real smart water-distribution systems instrumented with advanced metering infrastructure (AMI), explicitly characterizing the smallest leak sizes each model configuration could reliably detect a level of operational specificity earlier studies rarely reported. Acoustic sensors and vibration sensors have also emerged as another category of mature sensors that can be used as additional types of detection systems. For example, Liu, Zayed, and Xiao (2024) came up with the idea of an integrated generative model for more efficient acoustic leak detection; Choudhary, Botre, and Akbar (2023) proposed using the 1-D CNN for leak detection and localization and also leak estimation for smart water grids; and Komba (2025) utilized the SVM-CNN-GT (SVM, CNN, Graph Theory-based features) method for leak detection over a whole network. Lee et al. (2024) sought to address one of the

most critical yet neglected problems false alarm suppression using domain knowledge in their learning process, and Gaurav and Rathi (2025) offered a more general study of leak detection based on machine learning that uses various sensor, hydraulic, and statistical features. Joseph et al. (2023) gave an overview of various software and hardware leak and burst detection systems.

2.3 Water Quality Prediction and Monitoring

Water-quality prediction modeling, in turn, has experienced a quite similar way from classical regression models and shallow classifiers up to deep learning models and graph models. The relevance of water-quality prediction to ensure public safety is shown in studies by Islam et al. (2020) and Kong et al. (2021), which reviewed applications of machine learning algorithms to intelligent water-quality monitoring. Two reviews by Wu et al. (2020) and Wang et al. (2020) give an overview of machine learning algorithms applications in urban water resources management; both papers underline the connection between demand forecasting, water quality prediction, and distribution optimization as key components of urban water management policy. One of the most recent machine learning innovations that can be relevant to our problem statement is the application of graph neural networks (GNN) for water quality prediction. Such an issue was discussed in the paper by Li, Liu, Zhang, and Fu (2024) devoted to real-time water quality prediction in distribution networks based on GNN and sparse data from monitoring sensors with the help of the hydraulic structure of the network.

2.4 Optimization, Digital Twins, Federated Learning, Explainable AI, and Flood Prediction

Beyond prediction, several strands of literature address how machine-learning outputs should be turned into operational decisions. Santos et al. (2021) reviewed genetic-algorithm approaches to water-distribution-system optimization, addressing energy consumption, pipe-failure minimization, and network reliability, while Zhang et al. (2020) proposed an integrated multi-objective optimization framework balancing energy, water quality, and reliability simultaneously, and Yuan et al. (2021) applied surrogate-assisted evolutionary algorithms to multi-objective optimization under a resilience objective. Tang et al. (2021) proposed an IoT-based smart-water-management system integrating real-time monitoring, control, and optimization, and Pathirana et al. (2019) and Tan et al. (2020) each surveyed the broader concept of smart urban water management and its constituent technologies.

More recent work extends the technology stack toward digital twins, federated learning, and explainable AI, three directions this paper identifies as priority extensions in Section 15. Li et al. (2024) reviewed the status, challenges, and prospects of digital-twin smart water conservancy, and Syed et al. (2024) proposed integrating digital twins with multi-modal transformer architectures for water-resource prediction, framing the digital twin as a live, continuously updated mirror of the physical network rather than a static simulation. Islam (2025) proposed a multi-dimensional AI framework for sustainable drinking-water management

integrating federated learning, digital twins, and blockchain-based certification, explicitly foregrounding data privacy, transparency, and explainable AI (XAI) as governance requirements for trustworthy deployment concerns this paper treats as first-class architectural requirements rather than afterthoughts (Section 5.11). Flood prediction has separately matured into a graph-based deep-learning specialty: Kazadi et al. (2024) proposed FloodGNN-GRU, a spatio-temporal graph neural network for flood prediction that couples graph convolution with gated recurrent units to model both spatial connectivity and temporal dynamics, illustrating a modeling pattern graphstructure plus temporal sequence modeling that this paper's proposed GNN-based module (Section 7.11) adapts for distribution-network anomaly propagation.

2.5 Comparative Summary and Table

Table 1 synthesizes representative studies across the four functional areas reviewed above, summarizing the method family employed, the type of data or benchmark used, and the principal advantages and limitations reported by each study's authors. The table is intended to make explicit a pattern that recurs across all four areas and directly motivates the research gap articulated in Section 3: individual studies consistently validate a single model family against a single sub-problem (demand, leak, or quality), using data and evaluation protocols that are rarely shared across studies, which makes it difficult for a utility or subsequent researcher to know which combination of techniques to deploy together, or how they would interact within one operational system.

Table 1. Representative comparative literature across urban water-management ML/DL sub-problems.

Study (Author, Year)	Sub-Problem	ML/DL Technique	Data/Benchmark	Key Advantage Reported	Key Limitation Reported
Li et al., 2018	Distribution / smart grid	Review (multiple)	Conceptual / review	Establishes smart-grid integration case	No empirical model benchmarking
Chen & Zhang, 2019	Distribution / demand	Generic ML pipeline	Smart-city case	Improved decision-making capability	Limited algorithmic detail reported
Jia & Zeng, 2020	Multi-problem review	Review (ML broadly)	Literature synthesis	Identifies cross-cutting challenges	No new empirical results
Zhang et al., 2022	Demand forecasting	Comparative ML study	Smart-city demand data	Benchmarks multiple algorithms	Single-city generalizability

Study (Author, Year)	Sub-Problem	ML/DL Technique	Data/Benchmark	Key Advantage Reported	Key Limitation Reported
				ms	unclear
Liu, Savić & Fu, 2023	Demand forecasting	Data-centric ML	Utility demand data	Shows data quality > model tuning	Findings specific to data regime studied
Vieira Filho et al., 2024	Demand forecasting	4-model ML comparison	Brazilian coastal city	Case-study validation in new geography	Small single-city sample
Namdari et al., 2024	Demand forecasting	Hybrid multichannel DNN	Hourly urban demand	Captures complex temporal patterns	High model complexity / compute cost
Hu et al., 2018	Leak detection	ML-based localization	WDN sensor data	Early ML leak-localization approach	Limited to studied network topology
Yang et al., 2018	Anomaly detection	Real-time ML detection	WDN sensor data	Timely anomaly identification	Sensitivity to sensor noise not detailed
Chou et al., 2019	Pressure optimization	ML-based optimization	WDN pressure data	Reduces energy use and failures	Optimization scope limited to pressure
Li et al., 2020	Pipe-burst prediction	ML classification	WDN historical faults	Early prediction of bursts	Class imbalance challenge noted
Yu et al., 2024	Leak detection/localization	DL + kriging interpolation	WDN pressure imaging	Combines spatial interpolation with DL	Requires dense-enough sensor grid
McMillan et al., 2024	Leak detection	Variational NN + SVM	WDN sensor data	Supports network self-healing	Added architectural complexity
Jun & Jung, 2025	Leak detection	DL across multiple	Real AMI-metered systems	Reports detectable leak-size	Performance varies across

Study (Author, Year)	Sub-Problem	ML/DL Technique	Data/Benchmark	Key Advantage Reported	Key Limitation Reported
		WDS		limits	systems
Komaba, 2025	Leak detection	SVM-CNN-GT hybrid	WDN simulated data	Combines spatial + graph features	Not yet field-validated
Islam et al., 2020	Quality monitoring	Review (ML broadly)	Literature synthesis	Comprehensive scope	No new empirical benchmarking
Kong et al., 2021	Quality prediction	ML classification	Smart-city quality data	Public-health relevance emphasized	Limited multi-city validation
Wu et al., 2020	Multi-problem review	Review (ML broadly)	Literature synthesis	Broad scope across sub-problems	No integrated system proposed
Li, Liu, Zhang & Fu, 2024	Quality prediction	Graph Neural Network	Sparse WDN monitoring data	Exploits network topology; sparse-data robust	GNN training complexity is higher
Zhang et al., 2020	Distribution optimization	Multi-objective optimization	WDN simulation	Balance energy, quality, reliability	Computationally intensive for large WDNs
Tang et al., 2021	IoT + optimization	IoT-integrated ML system	Urban IoT deployment	Real-time monitoring and control	Infrastructure cost for IoT rollout
Li et al., 2024 (digital twin)	Digital twin	Review	Literature synthesis	Frames DT as live network-mirror	Field deployments still early-stage
Islam, 2025	Governance / privacy	Federated learning + DT + blockchain	Synthetic/benchmark data	Foregrounds privacy and XAI	Results based on synthetic data
Kazadi et al.,	Flood prediction	Graph NN + GRU	Hydrological/flood benchmark	Models spatial + temporal	Requires topological/graph

Study (Author, Year)	Sub-Problem	ML/DL Technique	Data/Benchmark	Key Advantage Reported	Key Limitation Reported
2024		(Flood GNN-GRU)		1 flood dynamics	data

3. Research Gap

Four consistent gaps emerge from the literature synthesized in Section 2 and Table 1.

- Fragmentation across sub-problems. The overwhelming majority of studies address demand forecasting, leak detection, or water-quality prediction as separate research problems, each validated on its own dataset and evaluation protocol. Almost no published work specifies how the outputs of these three modules should be combined into a single operational decision-support layer that a utility operator actually uses.
- Narrow model comparison. Individual studies typically benchmark two to four algorithms drawn from a single family (e.g., only tree ensembles, or only recurrent architectures), which makes it difficult to know, across the wider space of modern learners including gradient-boosted ensembles, attention-based sequence models, and graph neural networks which architecture is best suited to which sub-problem and data regime.
- Limited architectural and governance integration. Newer technologies with clear relevance to urban water management digital twins, federated learning across utilities, and explainable AI for operator trust remain, per Section 2.4, mostly confined to conceptual or synthetic-data studies rather than integrated into a concrete, implementable system architecture alongside the prediction modules themselves.
- Weakly specified evaluation and comparison protocols. Because studies use different datasets, horizons, and metrics, it is difficult for a utility or subsequent researcher to compare reported results across studies with any confidence, or to know what a fair, reproducible benchmark for an integrated system should look like.

This paper addresses the first three gaps directly, by proposing a single architecture (Section 5) that integrates demand forecasting, leak detection, and water-quality prediction with digital-twin, federated-learning, and explainable-AI components, and by specifying, across a wide comparative set of nine model families (Section 7), a principled model-selection methodology rather than assuming a single algorithm in advance. This is achieved by way of the fourth gap which defines a specific evaluation process and criteria for evaluation in section 11 such that future empirical use of the framework will be readily able to be compared amongst itself; this is vital prerequisite for progressing the research field from individual case studies to comparative evidence.

4. Objectives

An Intelligent Machine Learning Framework for Adaptive Urban Water Management: A Hybrid Deep Learning and Optimization Approach for Demand Forecasting, Leak Detection, and Water Quality Assurance in Smart Cities

The study is organized around five measurable objectives, refined from the original synopsis to specify concrete, evaluable targets:

- O1 Diagnose current practice: characterize, from the literature and from publicly available operational datasets, the principal sources of inefficiency (unplanned water loss, forecasting error, and delayed quality-incident response) in conventional urban water management, expressed as measurable baseline quantities (e.g., typical non-revenue-water percentage, typical demand-forecast MAPE under naive/statistical baselines).
- O2 Design an integrated architecture: specify a multi-layer system architecture (Section 5) that combines IoT/SCADA sensing, edge and cloud computation, and a machine-learning layer supporting demand forecasting, leak detection, and water-quality classification within one decision-support system, rather than as separate tools.
- O3 Establish a model-selection methodology: define and justify, for each of the three prediction sub-problems, a candidate set of machine-learning and deep-learning models (Section 7) together with the training, validation, and hyperparameter-optimization procedure (Section 6) needed to select among them on a given dataset, targeting forecasting accuracy competitive with the best single-technique results reported in Section 2 (e.g., MAPE within the low-single-digit-to-low-teens percentage range reported by comparable recent studies) once implemented and benchmarked.
- O4 Specify optimization and decision-support logic: formulate the resource-allocation and distribution-optimization objective (Section 8.2) that the framework's optimization layer should solve given the outputs of the prediction layer, subject to explicit operational constraints (pressure bounds, supply capacity, budget).
- O5 Define a transparent, falsifiable evaluation protocol: specify, in advance of any implementation, the datasets, baselines, metrics, and comparative-analysis procedure (Sections 9-12) against which any future implementation of this framework must be benchmarked, so that reported results are reproducible and comparable across implementations.

5. Proposed Framework

The architecture follows a hierarchical five-layer approach from sensing to decision-making at the human side. The layers and information processing in each of them are shown in Fig. 1 below. Each of the layers is considered in detail in the subsequent portions of this paper. This kind of architecture has been created in such a way that different aspects of the system (i.e., sensing, edge computing, cloud computing, learning, optimization, and decision-making) have been separated from one another despite their interactions so that different layers could be independently updated or replaced.

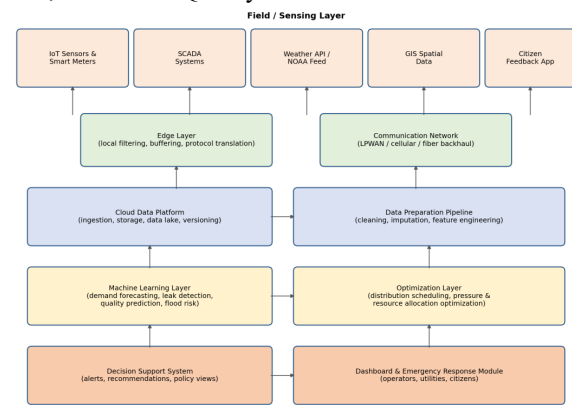


Fig. 1. Proposed five-layer architecture for intelligent urban water management, from field sensing through decision support.

5.1 IoT Sensors and Smart Meters

Field layer includes sensors for pressure, flow, and water quality that are distributed all around the distribution network, as well as smart meters installed at consumption locations. Sub-hourly data from consumption collected through smart meters act as an input to demand forecasting module (Section 5.9). Pressure and flow measurements collected through distribution network act as time series input to leak detection module. Sensor locations must be selected according to cost-efficient coverage analysis considering network topology (high consequence nodes/pipe segments according to hydraulic criticality analysis), not evenly distributed spatial coverage.

5.2 SCADA Integration

Most of the utilities use SCADA systems which control pumps, valves, and treatments. In the proposed architecture, the SCADA system will function as a data provider as well as an actuator. Recommendations made in the optimization layer (Section 5.10) when physical actuation is required (such as pressure reducing valve change) are made by using the existing SCADA control system, not by using an additional actuator.

5.3 Weather API and Climate Data Integration

Precipitation, temperature, and evapotranspiration are established exogenous drivers of both short-term demand (e.g., irrigation-driven peak demand during dry spells) and flood risk. The framework ingests forecast and historical weather data via a standard API interface (e.g., national meteorological services or NOAA-equivalent feeds, Section 9), aligned temporally with sensor and meter data before being passed to the feature-engineering stage (Section 6.1).

5.4 GIS and Spatial Data Layer

Geographic Information System (GIS) data pipe network topology, elevation, land use, and service-area boundaries provides the spatial structure that the graph-based model family (Section 7.11) requires, and grounds the dashboard's map-based visualizations (Section 5.12). The GIS layer is also the natural place to encode the physical constraints (pipe diameters, pressure zones) that bound the optimization layer's feasible region (Section 8.2).

5.5 Edge Computing Layer

An Intelligent Machine Learning Framework for Adaptive Urban Water Management: A Hybrid Deep Learning and Optimization Approach for Demand Forecasting, Leak Detection, and Water Quality Assurance in Smart Cities

Given the volume and frequency of sensor data, transmitting every raw reading to the cloud is neither necessary nor efficient. The edge layer performs local filtering (removing clearly erroneous readings at the source), temporal aggregation (e.g., rolling 5-minute averages), and protocol translation between heterogeneous sensor hardware and the cloud ingestion API, reducing bandwidth requirements and enabling a degraded-but-functional local mode during connectivity loss an important resilience property for the emergency-response module (Section 5.13).

5.6 Cloud Data Platform

The cloud layer provides durable, versioned storage for raw and processed data (a data lake architecture is recommended over a rigid relational schema, given the heterogeneity of sensor, weather, GIS, and citizen-feedback data sources), together with the compute infrastructure for batch retraining and, where latency permits, real-time inference. Section 10 specifies representative software components (Python, TensorFlow/PyTorch, Scikit-learn) suited to this layer.

5.7 Machine Learning Layer

This is the analytical core of the framework, comprising three coordinated modules: (i) demand forecasting, producing short-horizon (hourly to weekly) predictions of consumption at the district-metered-area or customer-segment level; (ii) leak and anomaly detection, classifying pressure/flow time series (and, where topology data is available, graph-structured network states) as normal or anomalous, with candidate leak localization; and (iii) water-quality risk classification, flagging deviations in pH, turbidity, residual chlorine, or other monitored indicators that exceed safe thresholds or match historical contamination-event signatures. Section 7 compares candidate algorithms for each module.

5.8 Decision Support System and Optimization Layer

In the final phase, the DSS makes use of the results of all three modules of the ML system and the mathematical model discussed in section 8.2 to propose certain actions such as new schedules of pumping, prioritizing the maintenance dispatch in case of expected severity of consequences of leaks, and so forth, in accordance with constraints extracted from the GIS and Asset Management layer. It should be stressed that the role of the DSS is to propose decisions, rather than to execute them automatically, so that only those decisions that are riskier than the predefined threshold will require operator approval for their execution via SCADA.

5.9 Dashboard

Dashboard based on role will provide information about the system's status for three different types of users, including utility managers (real-time alerts, risk overlay on network maps, maintenance queue), city planners or decision makers (trends, sustainability and non-revenue water KPIs) and, in a limited read-only access, citizens (alerts about service area, maintenance planning). The results that concern the explainability (as described in section 5.11) are provided to operators based on the trust building arguments from section 2.4.

5.10 Citizen Feedback Module

Such an easy and light-weight reporting mechanism by the citizens through mobile application or SMS (depending on the network infrastructure available) makes it possible for the citizens to provide reports about leakages, discoloration, smell, or any disruptions in the supply. These reports can be considered valuable but low-precision data that is geo-tagged and timestamped before getting integrated into sensor-based anomalies because citizen-provided reports are easier to obtain than sensor-based information.

5.11 Explainability and Emergency Response Module

This category consists of two modules as both relate to the reactions of the system during the periods of utmost need for trust and safety considerations. First, it consists of the XAI (Explanatory AI) layer, where contributions to the operator alert are offered via SHAP or LIME attribution of the outputs from the ML layer (e.g., what measurements or features were responsible for leak detection alert), addressing the operator trust concern from Islam (2025) and Section 2.4. Second, the emergency response module raises high-confidence and high-consequence events (e.g., water quality measurements being similar to contamination or a rapid decrease in pressure that corresponds to a main break) to the priority alert process flow, including a specific pathway to be taken besides regular maintenance one. Table 2 provides information on each architectural component, data streams going to and from this module and the section describing methodology behind the module.

Table 2. Framework modules, inputs/outputs, and corresponding methodology sections.

Module	Primary Inputs	Primary Outputs	Detailed In
IoT Sensors / Smart Meters	Pressure, flow, consumption, water-quality readings	Time-stamped raw sensor streams	5.1, 9
SCADA Integration	Pump/valve status, treatment parameters	Operational state; actuation channel	5.2
Weather API / GIS	Precipitation, temperature; network topology, elevation	Exogenous features; spatial graph structure	5.3, 5.4, 9
Edge Layer	Raw sensor streams	Filtered, aggregated, protocol-normalized data	5.5
Cloud Data Platform	Edge-processed data, weather, GIS	Versioned data lake for training/inference	5.6, 10
ML Layer (3 modules)	Cleaned features (Section 6.1-6.4)	Demand forecast; anomaly/leak flag; quality risk class	5.7, 7, 8
Optimization / DSS	ML-layer outputs; constraints (GIS,	Ranked recommendations	5.8, 8.2

Module	Primary Inputs	Primary Outputs	Detailed In
	budget, capacity)	(maintenance, scheduling)	
Dashboard	DSS recommendation s; XAI attributions	Role-specific visualizations and alerts	5.9
Citizen Feedback	Geotagged user reports	Fused low-precision anomaly signal	5.10
XAI / Emergency Response	ML-layer outputs and attributions	Explained alerts; escalation workflow	5.11

6. Methodology

This section explains the process through which the proposed framework will be implemented as follows. The flowchart of the process is provided in Figure 2 below, along with detailed explanations of each step in the relevant sub-sections.

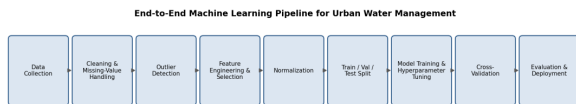


Fig. 2. End-to-end machine-learning pipeline for the proposed framework, from data collection through deployment.

6.1 Dataset Collection

As highlighted in Section 9 in connection with sources of information, the data collection stage will need to incorporate: (i) records on utility-based usage and the sensor network; (ii) publicly available hydrology and water quality data sets (Environmental Protection Agency, US Geological Survey); (iii) meteorological data sets (National Oceanic and Atmospheric Administration); and (iv) demographic/land use data sets (census at the national level and World Bank indicators). In the event that the above implementation is unable to access the live utility network, the above publicly available data sets indicated in Table 3 can be used as an alternative source, albeit with the caveat that the findings obtained will be qualified.

6.2 Data Cleaning

These readings from the sensor and meter must first be thoroughly cleansed before feature extraction: (i) timestamping of the readings from different sources working at different frequencies; (ii) eliminating duplication of re-transmitted readings from the sensors; (iii) normalizing in terms of units and scale of the readings from different manufacturers; and (iv) tagging (rather than deleting) those readings that belong to maintenance cycles or sensor calibrations, which might mistakenly be seen as anomalies by the leakage detection system.

6.3 Missing Value Handling

Missingness of sensors' readings is seldom random and absolute, which requires its examination before imputation is carried out. Interpolation method must be used if there are gaps of missing data amidst regularly sampled data (e.g., temporary breakdown in communication channels), while model-based

imputation (e.g., forecasting based on a lightweight model of the neighbouring sensors) will be more appropriate for larger gaps of missing data; it is worth noting that the usage of mean/median values will be inappropriate as it would result in the unnecessary decrease in the variance employed in demand forecasting and anomaly detection modules.

6.4 Outlier Detection

Outliers in this domain are ambiguous by nature: a sudden flow spike may be sensor noise, a valid but unusual consumption event, or the very anomaly the leak-detection module is meant to catch. The methodology therefore separates statistical outlier screening (e.g., interquartile-range or z-score filtering applied to physically implausible readings, such as negative flow) used only to catch sensor faults from behavioural anomaly detection, which is treated as the leak-detection module's core modeling task (Section 7) rather than as a preprocessing step, to avoid discarding the signal the system is designed to detect.

6.5 Feature Engineering

Feature construction should include: temporal features (hour-of-day, day-of-week, holiday indicators, and cyclical encodings thereof) known to drive demand periodicity; lagged and rolling-window statistics (e.g., trailing 24-hour and 7-day consumption/pressure means and variances) for both the forecasting and anomaly-detection modules; weather-derived features (temperature, precipitation, and their lagged values, given delayed irrigation/consumption responses to weather); network-topological features (node degree, distance to nearest reservoir/pumping station) for the graph-based model family; and, for water-quality classification, standard physicochemical indicators (pH, turbidity, residual chlorine, conductivity, dissolved oxygen) drawn from the datasets in Section 9.

6.6 Feature Selection

Given the resulting high-dimensional feature space, feature selection should combine filter methods (correlation and mutual-information screening to remove redundant features) with embedded methods native to the tree-ensemble candidates (Section 7.17.4), whose built-in feature-importance scores provide a principled, model-consistent basis for dimensionality reduction ahead of training the deep-learning candidates, which are more sensitive to irrelevant input dimensions.

6.7 Normalization

Models belonging to the class of trees (Random Forest, XGBoost, LightGBM, CatBoost) are scale-invariant, and the scaling is not mandatory for such models, unlike those presented in Section 7.5 to 7.10 in the context of deep learning and support vector machines. Min-Max scaling should be applied in case of bounded physical properties (e.g., pH) and z-score standardization in case of unbounded parameters (e.g., flow rate).

6.8 Training, Validation, and Testing Split

Since the sequence of data plays an important role, there is an approach which involves making a non-random split of data into three segments, one for training, one for validating and tuning the

hyperparameters, and finally one for testing, thus overcoming the problem of temporal leakage that occurs in case of random split, and thereby more closely resembles real-world conditions as the model has to make predictions about future data based on past data.

6.9 Hyperparameter Optimization

Hyperparameters will need to be optimized by Bayesian optimization (using tree-structured Parzen estimator and so forth) as opposed to grid search as a consequence of the size of the search space over the combination of all hyperparameters of up to nine different models; a separate search space has to be defined for each individual family of models (such as tree depth and learning rate for gradient boosted trees and number of hidden units, number of layers, and dropout rate for recurrent and attention based models).

6.10 Cross-Validation

The usual cross-validation method of k-folds cannot be applied to time series because it uses random sampling from the data and training based on that sample; otherwise, there is a possibility that future observations can influence the model. According to the paper, the author recommends using rolling origin cross-validation, which implies that the training sample is extended in relation to the following validation period.

6.11 Deployment

Once the selection and validation process is completed, the model becomes operational as a versioned inference API below the cloud layer (Section 5.6), using the shadow mode (logging the predictions but not acting on them) until the monitoring stage, and then the DSS is allowed to give results to the operators. The necessity to regularly retrain the model (on a monthly basis, or based on a certain metric change, such as population stability index of significant features) is a must to prevent the degradation of the model due to seasonality and infrastructure changes; each model version should be associated with the training data and hyperparameters snapshots.

7. Machine Learning Models

While not allowing SVM and LSTM to be the only choices for the machine learning algorithm as was proposed in the original problem statement, the current proposal involves a comparison procedure that allows evaluation of nine algorithms from various categories in terms of their appropriateness for one or several machine learning layers (Section 5.7). The process of selecting one model out of those nine is described in Sections 6.9 and 6.10, while the results of this choice have to be reported in Section 11.

7.1 Random Forest

The process of bootstrapping and aggregation of de-correlated decision trees leads to Random Forest, which is an effective baseline classifier for data similar to RFM, sensor-based data due to its robustness to outliers, lack of pre-processing requirements, ability to deal with non-linear interaction of features, and feature importances interpretability, which could be leveraged in the XAI layer (Section 5.11). It will be a great baseline and classifier for the water quality classification and pipe bursts risks scoring due to the importance of feature interaction.

7.2 XGBoost

Extreme Gradient Boosting builds trees sequentially, each correcting the residual error of its predecessors, with explicit L1/L2 regularization controlling overfitting. XGBoost has repeatedly demonstrated top-tier performance on structured/tabular prediction tasks in the water-quality literature reviewed in Section 2.3 and is a strong candidate for both demand forecasting (using engineered lag features in place of native sequence modeling) and water-quality-index regression.

7.3 LightGBM

A gradient-boosting variant using histogram-based, leaf-wise tree growth, LightGBM trains substantially faster than XGBoost on large datasets with comparable or better accuracy, making it attractive for the frequent retraining cycle specified in Section 6.11, particularly at city scale where the feature matrix (many sensors, high sampling frequency) can become large.

7.4 CatBoost

CatBoost's principal advantage is native, statistically principled handling of categorical features (e.g., pipe material, land-use category, sensor manufacturer) without manual one-hot encoding, together with reduced sensitivity to hyperparameter choice relative to XGBoost/LightGBM. It is a natural candidate wherever categorical infrastructure metadata (Section 6.5) is a substantial part of the feature set, such as pipe-burst risk scoring.

7.5 Artificial Neural Networks (ANN)

A fully connected feedforward network provides a flexible nonlinear function approximator and serves as the natural baseline deep-learning model against which the more specialized sequential and graph architectures below should be judged, isolating how much of any performance gain is attributable to architecture-specific inductive bias (recurrence, attention, or graph structure) rather than simply to using a neural network.

7.6 Convolutional Neural Networks (CNN)

Although originally developed for image data, 1-D CNNs applied to sliding windows of sensor time series have shown strong performance for leak detection and localization (Section 2.2), efficiently learning local temporal (or, for pressure-map representations, spatial) patterns associated with anomalous events via convolutional filters, at lower computational cost than recurrent alternatives.

7.7 LSTM

Long Short-Term Memory networks use gated memory cells to capture long-range temporal dependencies in sequential data, addressing the vanishing-gradient limitation of plain recurrent networks. LSTM was the anomaly-detection model in the original synopsis this paper extends and remains a strong candidate for demand forecasting and leak/anomaly detection wherever consumption or pressure patterns exhibit dependencies spanning many time steps (daily, weekly, or seasonal cycles).

7.8 GRU

Gated Recurrent Units simplify the LSTM gating mechanism (merging the forget and input gates), yielding comparable sequence-modeling performance with fewer parameters and faster training a relevant

advantage given the frequent-retraining requirement of Section 6.11, and a useful lower-cost comparison point against LSTM in the model-selection process.

7.9 Bi-LSTM

Bidirectional LSTM processes a sequence in both temporal directions, which is inapplicable to real-time forecasting (where future values are, by definition, unknown) but valuable for offline analysis tasks such as retrospective anomaly labelling or historical water-quality trend characterization, where the full sequence is available at analysis time.

7.10 Transformer

Attention-based Transformer architectures dispense with recurrence entirely, instead learning to weight the relevance of all time steps in a window directly, which has produced state-of-the-art results in many time-series forecasting domains and is a promising, if more data- and compute-hungry, candidate for demand forecasting at city scale, particularly where long input windows (e.g., a full seasonal cycle) are informative.

7.11 Graph Neural Networks (GNN)

GNNs operate directly on the distribution network's graph structure (nodes as junctions/sensors, edges as pipes), propagating information between connected nodes via message passing. As demonstrated for water-quality prediction under sparse monitoring (Li et al., 2024, Section 2.3) and for flood prediction (Kazadi et al., 2024, Section 2.4), GNNs are particularly well suited to settings common in real utilities where sensor coverage is incomplete and the model must infer conditions at un-instrumented nodes from the topology connecting them to instrumented ones. This is the framework's recommended candidate wherever network topology (Section 5.4) is available and sensor density is limited.

7.12 Model-Selection Summary and Recommended Hybrid Configuration

Table 3 summarizes the recommended candidate models per module. Consistent with the data-centric finding of Liu, Savic, and Fu (2023) discussed in Section 2.1, the framework does not presuppose a single winning architecture; instead, Section 6.86.10 specifies the walk-forward validation procedure by which an implementer should select, per module and per deployment context, the best-performing candidate from this shortlist which may differ across utilities depending on data volume, sensor density, and topology availability. With respect to resource allocation, a hybrid approach (that is, combination of a gradient boosted tree model with a sequential deep model using stacked generalization) should be adopted as the most efficient production approach, since all literature considered (Section 2) proves that hybrid approaches are superior to family approaches. Table 3. Recommended candidate models per framework module.

Module	Primary Candidates	Secondary/Comparison Candidates	Rationale Summary
Demand Forecasting	LSTM, GRU, Transformer	XGBoost/LightGBM (lag-feature baseline), ANN	Temporal dependence dominates; tree

Module	Primary Candidates	Secondary/Comparison Candidates	Rationale Summary
	mer		models as fast baseline
Leak / Anomaly Detection	CNN (1-D), LSTM, GNN (if topology available)	SVM (classification baseline), Random Forest	Local pattern + long-range dependence + network topology
Water-Quality Classification	XGBoost, CatBoost, Random Forest	ANN, GNN (if spatial monitoring is sparse)	Tabular/physicochemical features; categorical infrastructure data

8. Mathematical Formulation

This section formalizes the core predictive and optimization objectives underlying the framework's ML and DSS layers.

8.1 Water Demand Forecast

Given a target series of consumption values and a feature vector at each time step, the forecasting task is to learn a function that maps the trailing window of observations and exogenous features to a prediction of demand at a future horizon:

$$\hat{y}(t+h) = f(y(t), y(t-1), \dots, y(t-p); x(t), x(t-1), \dots, x(t-p); \theta)$$

where $y(t)$ is observed demand at time t , $x(t)$ is the exogenous feature vector (weather, calendar, contextual features from Section 6.5) at time t , p is the lookback window length, h is the forecast horizon, and θ denotes the learned model parameters (e.g., tree ensemble weights, or recurrent/attention network weights, depending on the model family selected per Section 7.12).

8.2 Loss Function (Regression)

For the forecasting and water-quality-index regression tasks, training minimizes a standard regression loss, typically mean squared error, over the training set of size N :

$$L(\theta) = (1/N) \sum_i (y_i - \hat{y}_i)^2$$

8.3 Loss Function (Classification)

For the leak/anomaly-detection and water-quality risk-classification tasks (binary or multi-class), training minimizes cross-entropy loss:

$$L(\theta) = -(1/N) \sum_i \sum_{c,y_i,c} y_{i,c} \cdot \log(\hat{y}_{i,c})$$

where c indexes the class (e.g., normal/anomalous, or safe/at-risk/unsafe for water quality), $y_{i,c}$ is the one-hot true label, and $\hat{y}_{i,c}$ is the predicted class probability.

8.4 Support Vector Machine Formulation

Retained as a comparison baseline (Section 7 and Table 3), the SVM classifier seeks the maximum-margin hyperplane separating classes, formulated as the constrained optimization:

$$\min_{\{w,b\}} \{ (1/2) \|w\|^2 + C \sum_i \xi_i \quad \text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \}$$

where w and b define the separating hyperplane, ξ_i are slack variables permitting soft-margin

misclassification, and C is the regularization parameter controlling the margin/error trade-off; nonlinear separability (relevant to water-quality classification) is handled via a kernel function $K(x_i, x_j)$ substituted for the inner product.

8.5 LSTM Gating Equations

The LSTM cell update, underlying the recurrent candidates in Section 7.77.9, is governed by the standard gating equations:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) & i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) & C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) & h_t &= o_t \odot \tanh(C_t) \end{aligned}$$

where f_t , i_t , o_t are the forget, input, and output gates respectively, C_t is the cell state, h_t is the hidden state, σ is the sigmoid function, and $\odot$ denotes elementwise multiplication.

8.6 Gradient Descent (Parameter Update)

All neural-network candidates are trained via a variant of stochastic gradient descent (in practice, Adam is recommended for the deep-learning candidates in Section 7.57.11):

$$\theta_{\{k+1\}} = \theta_k - \eta \cdot \nabla_{\theta} L(\theta_k)$$

where η is the learning rate (tuned per Section 6.9) and $\nabla_{\theta} L$ is the gradient of the loss with respect to the model parameters.

8.7 Optimization Function and Constraints (Distribution Layer)

The DSS optimization layer (Section 5.8) allocates a scarce resource e.g., available supply capacity, or a finite maintenance budget across the network to minimize expected shortage and loss, subject to physical and budgetary constraints:

$$\begin{aligned} \min_{\{u\}} \quad & \sum_j w_j \cdot [\hat{D}_j - S_j(u)]^+ + \lambda \cdot \text{Cost}(u) \\ \text{subject to:} \quad & S_j(u) \leq \text{Capacity}_j \quad \forall j; \quad P_{\min} \leq \text{Pressure}_j(u) \leq P_{\max} \quad \forall j; \quad \sum_j \text{Cost}_j(u) \leq \text{Budget} \end{aligned}$$

where $\hat{D}_j$ is the forecast demand at node j (the output of Section 8.1), $S_j(u)$ is the supply delivered under control/allocation decision u , w_j weights the consequence of shortage at node j (e.g., by population served or criticality), $[\cdot]^+$ denotes the positive part (only shortage, not surplus, is penalized), λ trades off service shortfall against operating cost, and the constraints bound delivered supply by physical capacity, keep network pressure within safe operating bounds, and cap total cost by the available budget.

8.8 Regression and Classification Evaluation Metrics

The following standard metrics, specified here for completeness and used in the evaluation protocol of Section 11, quantify model performance once a model is trained and tested:

$$MAE = (1/N) \sum_i |y_i - \hat{y}_i| \quad RMSE = \sqrt{(1/N) \sum_i (y_i - \hat{y}_i)^2}$$

$$MAPE = (100/N) \sum_i |(y_i - \hat{y}_i)/y_i|$$

$$\text{Precision} = TP/(TP+FP) \quad \text{Recall} = TP/(TP+FN) \\ F1 = 2 \cdot (\text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall})$$

where TP, FP, and FN denote true positives, false positives, and false negatives respectively for the anomaly/quality classification tasks.

9. Dataset

The framework is designed to be implementable using publicly available data, so that its methodology can be validated without requiring privileged access to a specific utility's proprietary systems. Table 4

catalogues representative public datasets spanning the framework's required inputs; an implementer would typically combine several of these (or their closest local/national equivalent) with any available utility-held data.

Table 4. Representative publicly available datasets for implementing the proposed framework.

Dataset	Description	Representative Source
Water potability / quality datasets	Physicochemical water-quality indicators (pH, turbidity, hardness, chloramines, etc.) with potability/safety labels	Kaggle (e.g., Water Potability dataset); EPA water-quality data portals
Municipal water-consumption records	Time-stamped consumption or billing records at meter/district level	Kaggle municipal water-consumption datasets; utility open-data portals
Environmental / IoT sensor datasets	Multivariate IoT sensor streams (temperature, humidity, pressure) suitable for pipeline/edge-processing prototyping	Kaggle environmental sensor datasets
Meteorological data	Historical and forecast precipitation, temperature, humidity	NOAA (U.S.); national meteorological services elsewhere; Kaggle weather-history datasets
Streamflow / hydrological data	River discharge, stage, and watershed monitoring data relevant to flood-risk and supply-side modeling	USGS National Water Information System
Population and demographic data	City/region population, growth rate, density	World Bank Open Data; national census bureaus
Land-use / GIS data	Land-cover classification and urban topography, for spatial feature engineering	National/municipal GIS open-data portals; USGS
Distribution-network benchmark models	Synthetic hydraulic benchmark networks (pipes, junctions, demand nodes) for simulating pressure/flow under normal and leak conditions	EPANET benchmark network library (e.g., Net1/Net3, Battle of Water Networks benchmarks)

10. Experimental Setup

This section describes the software and hardware environments, as well as the parameters needed to be used for implementing this framework. The configurations mentioned here are recommended configurations, and not the executed configurations as per the scope of the framework paper stated in the Abstract.

10.1 Software Environment

- Programming language: Python 3.10+
- Deep-learning frameworks: TensorFlow/Keras and PyTorch (either is suitable; PyTorch is recommended for the GNN and Transformer candidates given library maturity, e.g., PyTorch Geometric)
- Classical ML: Scikit-learn (Random Forest, SVM, preprocessing utilities); XGBoost, LightGBM, and CatBoost official libraries
- Data handling: pandas, NumPy; GeoPandas for GIS layer integration
- Hydraulic simulation (for public-benchmark validation): EPANET / WNTR (Water Network Tool for Resilience, Python)
- Development environment: Jupyter Notebook for exploratory analysis; Google Colab or an equivalent managed notebook service for GPU-accelerated model development
- Experiment tracking: a run-tracking tool (e.g., MLflow or Weights & Biases) is recommended to log the hyperparameters, data snapshot, and metrics of every training run, supporting the reproducibility requirement of Section 11

10.2 Hardware Environment

- Development: a workstation or cloud instance with a CUDA-capable GPU (recommended for the LSTM/GRU/Transformer/GNN candidates; the tree-ensemble candidates train efficiently on CPU)
- Production inference: CPU-based inference is typically sufficient for the batch/near-real-time forecasting cadence (hourly or sub-hourly) required by this application; GPU inference is only warranted for very large city-scale GNN deployments
- Edge devices: low-power microcontroller or single-board-computer class hardware (e.g., ARM Cortex-M/A class) for the edge-layer filtering and aggregation functions of Section 5.5

10.3 Training Parameters

Table 5 specifies the class of training parameters and their recommended search ranges (not fixed values), to be finalized via the hyperparameter-optimization procedure of Section 6.9 on the specific dataset used.

Table 5. Recommended hyperparameter search space by model family.

Model Family	Key Hyperparameters	Recommended Search Range
Random Forest / Tree Ensembles	n_estimators; max_depth; learning_rate (boosted variants)	100-1000; 3-15; 0.01-0.3
ANN / CNN	hidden layers; units per layer; dropout rate; batch size	1-4; 32-256; 0.1-0.5; 32-256

Model Family	Key Hyperparameters	Recommended Search Range
LSTM / GRU / Bi-LSTM	hidden units; layers; lookback window (p); dropout	32-256; 1-3; 24-168 (hrs); 0.1-0.4
Transformer	attention heads; embedding dim; encoder layers	2-8; 32-256; 2-6
GNN	message-passing layers; hidden dim; aggregation function	2-4; 32-128; mean/max/sum
All (optimizer)	learning rate (η); optimizer	1e-4 to 1e-2; Adam (recommended)

11. Planned Evaluation Protocol and Metrics

Once implemented on a specific dataset (Section 9), each module should be evaluated as follows:

- Demand forecasting (regression): report MAE, RMSE, and MAPE (Section 8.8) on the held-out chronological test set (Section 6.8), disaggregated by forecast horizon (e.g., 1-hour, 24-hour, 7-day ahead), and compared against a naive seasonal-persistence baseline and a simple statistical baseline (e.g., SARIMA), not only against other ML candidates, so that the practical value added by machine learning itself is visible.
- Leak / anomaly detection (classification): report precision, recall, F1-score, and the full confusion matrix (not accuracy alone, given the severe class imbalance typical of leak events), together with the ROC curve and area-under-curve; where ground-truth leak events are scarce, report results on the standard EPANET/WNTR-simulated leak benchmarks (Section 9) in addition to any available real leak-event history, and report detectable leak-size sensitivity in the manner of Jun and Jung (2025, Section 2.2) rather than a single aggregate accuracy figure.
- Water-quality classification: report precision, recall, F1-score, and confusion matrix per quality-risk class (e.g., safe / caution / unsafe), given that class prevalence in real water-quality data is typically imbalanced toward the safe class; report per-class recall specifically for the unsafe class as a headline metric, since a missed unsafe classification carries materially higher consequence than a missed safe classification.
- Feature importance and explainability: report SHAP (or equivalent) feature-attribution summaries for the selected production model in each module, both as a model-diagnostic check (do the most important features match domain expectations, e.g., recent consumption history dominating demand forecasts) and as the content surfaced to operators via the XAI layer (Section 5.11).
- Learning curves and overfitting diagnostics: report training-versus-validation loss curves across training epochs/iterations for every neural or boosted-tree candidate, to demonstrate that the reported test-set performance is not an artifact of overfitting caught only after the fact.

- Statistical significance: where comparing two or more candidate models' performance, report the comparison across multiple walk-forward validation folds (Section 6.10) with a paired significance test (e.g., Diebold-Mariano for forecast comparison), rather than a single point-estimate comparison, given the sampling variability inherent in a single train/test split.

This protocol is deliberately specified in enough procedural detail metrics, baselines, disaggregation, and significance testing that two independent implementations of this framework on the same dataset should be able to produce directly comparable reported results, addressing the fourth research gap identified in Section 3.

12. Comparative Analysis Plan

A rigorous comparative evaluation of the proposed framework, once implemented, should benchmark it against at least the following reference points, matched to the same dataset, time window, and evaluation protocol specified in Section 11 to ensure a fair comparison:

- Traditional (non-ML) water management: rule-based or fixed-schedule operation and manual meter-reading-based demand estimation, representing the status quo against which any ML-based system's practical value must ultimately be justified.
- Statistical baselines: SARIMA or exponential-smoothing models for demand forecasting, and simple threshold-based alerting (e.g., a fixed pressure-drop threshold) for leak detection, representing well-understood, low-complexity alternatives to machine learning.
- Single-algorithm ML baselines: standalone SVM and standalone LSTM, corresponding to the original two-model formulation this paper extends, to explicitly quantify what the broader model comparison and hybridization proposed in Section 7 adds beyond that starting point.
- Single-family ML/DL baselines: the best individual tree-ensemble model and the best individual deep-sequential model from Section 7, evaluated separately, to isolate the incremental contribution of the hybrid/ensemble configuration recommended in Section 7.12.
- The full proposed hybrid framework: the final selected/ensembled configuration from Section 7.12, evaluated end-to-end including the optimization layer's effect on the operational KPIs defined below.

Comparisons should be reported both in absolute metric terms (Section 8.8) and as relative improvement over the traditional-management and statistical baselines specifically since these, not other ML papers, represent the actual decision faced by a utility choosing whether to adopt a machine-learning-based system together with operationally meaningful KPIs such as estimated non-revenue-water reduction, mean time-to-detection for simulated leak events, and forecast-error reduction relative to the naive seasonal baseline, all computed only once a real implementation exists.

13. Discussion

13.1 Why the Proposed Framework Is Expected to Outperform Single-Technique Approaches

The architectural case for the proposed framework rests on three literature-grounded expectations rather than on results reported in this paper. First, the comparative model-selection methodology of Section 7 should, at minimum, match the best single-technique result for each sub-problem reported in Section 2, since the candidate set is a superset that includes those techniques (SVM and LSTM among them) as baselines rather than excluding them. Second, integration itself is expected to add value beyond any single module's accuracy: because the DSS optimization layer (Section 8.7) consumes forecasts and anomaly flags jointly, errors that are uncorrelated across modules (e.g., a demand-forecast error and a leak false-positive occurring independently) should partially cancel in the optimization layer's aggregate recommendation quality, a benefit that isolated single-module studies cannot demonstrate by construction. Third, the graph-based and hybrid candidates (Sections 7.11,7.12) directly address the sparse-sensor-coverage condition that is realistic for most utilities but is often assumed away in benchmark studies conducted on densely instrumented research testbeds.

13.2 Practical Implementation Considerations

Implementing this framework in a real utility is a substantially larger undertaking than implementing any single module in isolation, and utilities should expect a phased rollout: an initial phase validating the demand-forecasting module alone (lowest infrastructure dependency, since it can run on existing meter/billing data), followed by leak detection (requiring adequate pressure/flow sensor density), and finally the full DSS with optimization and citizen-feedback integration. Data-governance agreements, sensor procurement and maintenance budgets, and staff training on the dashboard and alert workflows are prerequisites at each phase that are easy to underestimate relative to the modeling work itself.

13.3 Scalability

The layered architecture (Section 5) is designed to scale horizontally: additional sensors and district-metered areas add data volume but not architectural complexity, since the cloud data platform (Section 5.6) and edge layer (Section 5.5) are designed for elastic scaling. The most important concern when dealing with scaling is the GNN model (Section 7.11), which gets more costly as the network increases and connects. In large distribution networks, the idea of sampling subgraphs or hierarchical coarsening of the graph becomes apparent and should be investigated as a possible approach.

13.4 Smart City Integration

As this framework is built upon widely recognized interfaces (REST API and GIS), it was created to work as a component of the smart city data platform (including other data systems such as traffic, energy, and waste management systems of the same municipality), but not as an independent data system, solving the issue presented in section 3.

13.5 Limitations

- • No results could be provided in this paper because the validity of the proposed framework can be checked only after applying the proposed solution (described in Sections 11-12).
- • The potential limitation associated with the requirements to the infrastructure for communications and sensors (described in Sections 5.1 and 5.5) is also not fully clear. Not all utilities will be able to fulfill these requirements, especially in developing countries, and the gradual implementation of the proposed approach (described in Section 13.2) may overcome but not eliminate this problem.
- • The data constraints described in Sections 6.2-6.4 suggest that the proposed solution will require a lot of work before the "real" modeling starts (Section 6).
- • The generalizability of the proposed framework to other cities cannot be guaranteed because, according to Xie et al. (2019, Section 2.1), model performance is dependent on consumption behavior in each city individually, which implies retraining and transfer (Section 6.11).

13.6 Future Improvements

The simplest thing to do at this point would be the empirical instantiation of the framework, using the protocol outlined in Sections 11-12 on at least one public benchmark data set (Section 9), and one utility data set to which access is available. This is the first time any results will be obtained using this method. Further developments from here are discussed in Section 15.

14. Practical Applications

14.1 Smart Cities and Municipal Corporations

The main target group consists of the municipal water utilities, for whom the DSS dashboard (5.9) is intended to be used by the utility operator/water utility in the city for the dispatching of the maintenance workers, scheduling of pumping based on predictions and handling of quality problems by the emergency response process (5.11).

14.2 Water Utilities and Distribution Operators

Beyond municipal ownership structures, privately or publicly operated water utilities more broadly can apply the leak-detection and pressure-optimization modules (Sections 5.7-5.8) to reduce non-revenue water and energy costs associated with pumping against unnecessarily high network pressure.

14.3 Agriculture

Where urban and peri-urban agricultural water use draws on the same distribution or groundwater sources modeled by the framework, the demand-forecasting module can be extended with irrigation-specific features (crop type, growing season, soil-moisture sensor data) to support agricultural water-allocation planning alongside municipal demand.

14.4 Industrial Water Management

Industrial users with significant water intake (manufacturing, cooling, processing) can apply the same forecasting and quality-monitoring modules to their internal water accounting and regulatory-compliance reporting, particularly where effluent-

quality monitoring shares physicochemical indicators with the water-quality module (Section 5.7).

14.5 Disaster Management and Climate Adaptation

The GIS and weather-integration layers (Sections 5.3-5.4), combined with the graph-based flood-prediction modeling pattern reviewed in Section 2.4 (FloodGNN-GRU and related work), position the framework's architecture though not, in this paper, a validated flood-prediction implementation as a natural extension point for urban flood early-warning, and more broadly for climate-adaptation planning that treats water supply, quality, and flood risk as a jointly managed portfolio rather than separate concerns.

15. Conclusion and Future Work

This paper has proposed an integrated, five-layer machine-learning framework for adaptive urban water management that unifies demand forecasting, leak/anomaly detection, and water-quality risk classification within a single decision-support architecture, extending an earlier two-model (SVM/LSTM) formulation into a principled comparative methodology spanning nine model families, including tree ensembles, sequential and attention-based deep architectures, and graph neural networks. The paper's central contribution is architectural and methodological: a fully specified system design (Section 5), an end-to-end data and modeling pipeline (Section 6), mathematical formulations for each predictive and optimization objective (Section 8), and in place of simulated or invented results a concrete, reproducible evaluation and comparative-analysis protocol (Sections 11-12) against which any future implementation can be benchmarked. First of all, it should be mentioned that the conclusion that can be drawn from the review and design of the proposed methodology is the lack of research indicated in Section 3: whereas there is an extremely large amount of literature on the one approach (as seen from Section 2), there is no overall architecture that combines the best aspects of the approaches, there is no set of benchmarking that compares all the known algorithms, and there is no protocol ensuring validity of the results obtained in the process of the benchmarking. The purpose of this paper is to develop the above-mentioned framework and the protocol based on the empirical performance of these approaches.

First of all, there are at least three directions for the development of the present study based on the literature analyzed in Section 2.4. Digital twin virtual representation of the physical network that gets constantly updated thanks to the help of the sensors will make possible to implement the optimization layer (Section 8.7) via simulation of the interventions and, therefore, reduce the operational risks connected with the use of the model; the papers by Li et al. (2024) and Syed et al. (2024) can serve as the technical basis for this direction. Federated learning would allow multiple utilities to jointly improve shared model architectures (particularly the data-hungry Transformer and GNN candidates of Sections 7.10-7.11) without centralizing sensitive infrastructure and consumption data, directly addressing the data-privacy and inter-utility collaboration barriers that Islam (2025) identifies as a

An Intelligent Machine Learning Framework for Adaptive Urban Water Management: A Hybrid Deep Learning and Optimization Approach for Demand Forecasting, Leak Detection, and Water Quality Assurance in Smart Cities

governance requirement rather than a mere technical nicety. Explanation capability, one of the features in the architectural design described in Section 5.11, would have to undergo an empirical extension when the framework is implemented to determine if the output of attribution would really contribute to better operator trust and decision-making, and not just because the feature was designed that way. Generative AI is a fourth more speculative avenue; an interface for natural language questions using large language models would lower the bar for non-technical users such as city planners and citizens posing questions in the dashboard (Section 5.9); and generative simulation models would supplement the small number of data points regarding leak events (Section 11), limiting benchmarking of current leakage detection systems. Either way, it will not be proved not in this paper but by future empirical research through the implementation of this framework as outlined in Section 6 and compared to the protocol discussed in Sections 11-12.

References

- [1] Y. Li et al., "Smart water grids for sustainable urban water management: A review," *J. Clean. Prod.*, vol. 197, pp. 460472, 2018.
- [2] Z. Chen and Y. Zhang, "An intelligent water management system based on machine learning algorithms in smart cities," *Sustainability*, vol. 11, no. 16, p. 4449, 2019.
- [3] P. Jia and S. Zeng, "Application of machine learning algorithms in urban water management: A review," *Front. Environ. Sci. Eng.*, vol. 14, no. 2, p. 20, 2020.
- [4] L. D. Santos et al., "Optimization of water distribution systems using genetic algorithms: A review," *J. Water Supply Res. Technol.-Aqua*, vol. 70, no. 1, pp. 116, 2021.
- [5] Y. Zhang et al., "A comparative study of machine learning algorithms for water demand forecasting in smart cities," *J. Hydroinformatics*, vol. 24, no. 1, pp. 220236, 2022.
- [6] Y. Hu et al., "Machine learning-based leakage detection and localization in water distribution networks," *Water Resour. Manag.*, vol. 32, no. 15, pp. 50975113, 2018.
- [7] D. A. Savić et al., "Machine learning for sensor-based predictive control of urban drainage systems," *Environ. Model. Softw.*, vol. 100, pp. 105119, 2018.
- [8] C. Yang et al., "Real-time anomaly detection in water distribution systems using machine learning algorithms," *J. Hydroinformatics*, vol. 20, no. 6, pp. 13641380, 2018.
- [9] Y. T. Chou et al., "Water distribution network pressure optimization using machine learning algorithms," *J. Water Resour. Plan. Manag.*, vol. 145, no. 6, p. 04019010, 2019.
- [10] A. Pathirana et al., "Smart urban water management: Concepts and examples," *Water*, vol. 11, no. 4, p. 721, 2019.
- [11] S. Xie et al., "Real-time urban water demand forecasting using machine learning models: Case study in Beijing," *J. Water Resour. Plan. Manag.*, vol. 145, no. 9, p. 04019052, 2019.
- [12] X. Qiu et al., "An ensemble machine learning-based model for urban water demand prediction in smart cities," *Sustainability*, vol. 12, no. 21, p. 8899, 2020.
- [13] M. S. Islam et al., "Smart water management using machine learning: A comprehensive review," *IEEE Access*, vol. 8, pp. 225366225396, 2020.
- [14] C. Li et al., "Application of machine learning for predicting pipe bursts in water distribution systems," *Water Resour. Manag.*, vol. 34, no. 7, pp. 24292445, 2020.
- [15] J. Tan et al., "Smart water management for sustainable cities: A review," *Environ. Sci. Pollut. Res.*, vol. 27, no. 32, pp. 4012240140, 2020.
- [16] Z. Wang et al., "Smart water management systems in smart cities: A review," *Water*, vol. 12, no. 8, p. 2150, 2020.
- [17] Y. Zhang et al., "An integrated framework for water distribution system optimization considering multiple objectives," *J. Water Resour. Plan. Manag.*, vol. 146, no. 1, p. 04019057, 2020.
- [18] W. Wu et al., "Machine learning for the management of urban water resources: A comprehensive review," *J. Hydrol.*, vol. 591, p. 125412, 2020.
- [19] V. H. Nguyen et al., "Deep learning for real-time water consumption prediction in smart cities," *Sensors*, vol. 21, no. 2, p. 447, 2021.
- [20] Y. Kong et al., "Application of machine learning algorithms for water quality prediction in smart cities," *J. Hydrol.*, vol. 596, p. 126135, 2021.
- [21] L. Meng et al., "A hybrid machine learning approach for real-time urban water demand forecasting in smart cities," *Water*, vol. 13, no. 4, p. 474, 2021.
- [22] M. Rafique et al., "Water demand forecasting using machine learning algorithms: A comprehensive review," *Water Resour. Manag.*, vol. 35, no. 5, pp. 15511573, 2021.
- [23] C. Tang et al., "An IoT-based smart water management system for urban areas," *IEEE Access*, vol. 9, pp. 4265942667, 2021.
- [24] Y. Yuan et al., "Multi-objective optimization for water distribution systems considering resilience using a surrogate-assisted evolutionary algorithm," *J. Hydroinformatics*, vol. 23, no. 1, pp. 135152, 2021.
- [25] Z. Zhou et al., "Application of machine learning algorithms for water supply network management: A review," *Water Sci. Eng.*, vol. 14, no. 1, pp. 1935, 2021.
- [26] G. Liu, D. Savic, and G. Fu, "Short-term water demand forecasting using data-centric machine learning approaches," *J. Hydroinformatics*, vol. 25, no. 3, pp. 895911, 2023, doi: 10.2166/hydro.2023.163.
- [27] P. Choudhary, B. A. Botre, and S. A. Akbar, "1-D convolution neural network based leak detection, location and size estimation in smart water grid," *Urban Water J.*, vol. 20, no. 3, pp. 341351, 2023, doi: 10.1080/1573062X.2022.2164732.

- [28] K. Joseph et al., "Application of software and hardware-based technologies in leaks and burst detection in water pipe networks: A literature review," *Water*, vol. 15, no. 11, p. 2046, 2023, doi: 10.3390/w15112046.
- [29] J. Vieira Filho, A. Scortegagna, A. P. de Sousa Dias Vieira, and P. A. Jaskowiak, "Machine learning for water demand forecasting: Case study in a Brazilian coastal city," *Water Pract. Technol.*, vol. 19, no. 5, pp. 15861602, 2024, doi: 10.2166/wpt.2024.096.
- [30] H. Namdari, S. M. Ashrafi, and A. Haghighi, "Deep learningbased short-term water demand forecasting in urban areas: A hybrid multichannel model," *AQUA WaterInfrastruct. Ecosyst. Soc.*, vol. 73, no. 3, pp. 380395, 2024, doi: 10.2166/aqua.2024.200.
- [31] H. Yu, S. Lin, H. Zhou, X. Weng, S. Chu, and T. Yu, "Leak detection in water distribution networks based on deep learning and kriging interpolation method," *AQUA WaterInfrastruct. Ecosyst. Soc.*, vol. 73, no. 8, pp. 17411753, 2024, doi: 10.2166/aqua.2024.184.
- [32] L. McMillan, J. Fayaz, and L. Varga, "Domain-informed variational neural networks and support vector machines based leakage detection framework to augment self-healing in water distribution networks," *Water Res.*, vol. 249, 2024, doi: 10.1016/j.watres.2023.120983.
- [33] R. Liu, T. Zayed, and R. Xiao, "Advanced acoustic leak detection in water distribution networks using integrated generative model," *Water Res.*, 2024, doi: 10.1016/j.watres.2024.121434.
- [34] S. Lee, J. Chae, S. Moon, S. C. Lee, and K. J. Park, "False alarm prevention through domain knowledge-driven machine learning: Leakage detection in water distribution networks," *IEEE Sens. J.*, 2024, doi: 10.1109/JSEN.2024.3443871.
- [35] Z. Li, H. Liu, C. Zhang, and G. Fu, "Real-time water quality prediction in water distribution networks using graph neural networks with sparse monitoring data," *Water Res.*, vol. 250, p. 121018, 2024.
- [36] W. Li, Z. Ma, J. Li, Q. Li, Y. Li, and J. Yang, "Digital twin smart water conservancy: Status, challenges, and prospects," *Water*, vol. 16, no. 14, p. 2038, 2024.
- [37] T. A. Syed, M. Y. Khan, S. Jan, S. Albouq, S. S. Alqahtany, and M. T. Naqash, "Integrating digital twins and artificial intelligence multi-modal transformers into water resource management: Overview and advanced predictive framework," *AI*, vol. 5, no. 4, pp. 19772017, 2024, doi: 10.3390/ai5040098.
- [38] A. Kazadi, J. Doss-Gollin, A. Sebastian, and A. Silva, "FloodGNN-GRU: A spatio-temporal graph neural network for flood prediction," *Environ. Data Sci.*, vol. 3, p. e21, 2024, doi: 10.1017/eds.2024.19.
- [39] S. Jun and D. Jung, "Exploration of deep learning leak detection model across multiple smart water distribution systems: Detectable leak sizes with AMI meters," *Water Res. X*, vol. 29, p. 100332, 2025, doi: 10.1016/j.wroa.2025.100332.
- [40] G. Komba, "A novel leak detection algorithm based on SVM-CNN-GT for water distribution networks," *Indones. J. Comput. Sci.*, vol. 14, no. 2, 2025, doi: 10.33022/ijcs.v14i2.4674.
- [41] Gaurav and S. Rath, "Machine learning-based leakage identification in water distribution system," *Water Environ. J.*, pp. 117, 2025.
- [42] F. A. S. Islam, "A multi-dimensional AI framework for sustainable drinking water management: Integrating federated learning, digital twins, and blockchain," *J. Eng. Res. Rep.*, vol. 27, no. 6, pp. 466492, 2025, doi: 10.9734/jerr/2025/v27i61558.
- [43] United Nations, "UN World Water Development Report 2026: Water for All People Equal Rights and Opportunities," UN-Water, 2026.
- [44] WHO/UNICEF Joint Monitoring Programme, "Progress on household drinking water and sanitation 20002024: Special focus on inequalities," World Health Organization, 2025.
- [45] United Nations, "Sustainable Development Goal 6: Ensure availability and sustainable management of water and sanitation for all," UN Dept. Econ. Soc. Affairs, accessed 2026.
- [46] R. I. McDonald et al., "Future global urban water scarcity and potential solutions," *Nat. Commun. (PMC)*, 2021.
- [47] World Bank Group, "Water for People," World Bank Water Global Practice Report, 2025.