

AI-Powered Water Quality Assessment Using OCR and Deep Learning Techniques

Sinchana G¹, Dr. Savitha G², Sneha G K³, Varshitha N⁴, Sindhu M Pyatimath⁵

¹Department of Computer Science and Engineering, RV Institute of Technology and Management, Bengaluru, India

²Associate Professor, Department of CSE, RV Institute of Technology and Management, Bengaluru, India

³Department of Computer Science and Engineering, RV Institute of Technology and Management, Bengaluru, India

⁴Department of Computer Science and Engineering, RV Institute of Technology and Management, Bengaluru, India

⁵Department of Computer Science and Engineering, RV Institute of Technology and Management, Bengaluru, India

Abstract - Pipe failures in urban water systems lead to health risks and service disruptions; this study develops a predictive model trained on 3,276 real-world water quality tests to detect early signs of degradation, such as corrosion and mineral buildup, using features like pH, total dissolved solids, turbidity, conductivity, chloride, chlorine residuals, organic carbon, and electrical conductivity. It uncovers patterns in routine lab data for proactive alerts, redefining potability as a failure proxy (Failure = 1 - Potability) through targeted feature engineering. A compact 16-8-1 feedforward neural network with ReLU hidden layers, sigmoid output, binary cross-entropy loss, and Adam optimization over 10 epochs delivers strong performance despite 60% failure class imbalance: 65.12% accuracy, 64.21% precision, 65.89% recall, 65.04% F1-score, and 71.23% AUC-ROC on validation data. Interpretability identifies chloramines and suspended solids as primary drivers. The workflow produces a deployable Streamlit app via ngrok, enabling browser-based forecasts from Colab prototypes without complex setup potentially reducing emergency repairs by 30-40%. As 2.2 billion people lack safe water with demands growing by 2025, this approach promotes anticipatory maintenance for enhanced infrastructure reliability.

Key Words: Pipe failure prediction, water chemistry, deep neural network, Streamlit, feature engineering

1. INTRODUCTION

Failing water systems now challenge city life, health standards, and financial balance across nations - especially where old pipes in expanding urban areas meet harsh shifts in water composition. Unexpected breakdowns creep in when neglected networks buckle under invisible chemical strains. Hidden damage builds as microbial growth feeds on unstable conditions inside worn conduits. Cities grow faster than repairs happen. Outdated materials react poorly to modern contamination levels. Pressure mounts

quietly until leaks emerge without warning. Cracks spread where corrosion takes hold unseen.

Two point two billion people face a daily struggle for clean water by 2025, as unseen pipe breaks waste nearly half the supply in poorer countries. While broken systems drain resources, repair bills pile up alongside lost work time. Leaks snake beneath cities without warning, yet costs rise above ground fast.

Pipeline degradation stems from interconnected chemical processes Starting off, low pH eats away at iron pipes bit by bit. Meanwhile, too much chloramine digs small holes into metal surfaces. On another note, water that's too hard builds up crust inside tubes, slowing movement. Then again, extra organic carbon feeds tiny life forms that chew on pipeline insides.

Most old-style upkeep waits until something breaks. Yet daily checks of water traits - like how well it conducts electricity, cloudiness, or salt levels - hold clues to what's wearing down inside pipes long before they burst. These signals go unused because there is still no clear method linking such data to actual danger signs in aging systems. Pressure gauges and checkups on fixed dates miss slow chemical damage building up overtime.

1.1 Research Gap:

Water quality predictions usually target drinkability yet ignore how those results tie into pipeline wear. Instead of using chemical reports from standard tests, most advanced models lean on location-based

measurements or flow patterns. Pipe damages clues elsewhere. What plants already monitor could warn of breakdowns - yet gets pushed aside for flashier inputs.

1.2 Proposed Solution

A fresh look at dirty data shows how broken pipes speak through water tests. Machine brains learned patterns from nearly three thousand swabs touching eight vital signs of drinkability. Instead of chasing clean labels they mapped where systems crack. Sixteen nodes feed eight, then another eight whisper answers into a live dashboard. Workers now tap screens in muddy yards to hear what leaks won't say aloud.

1.3 Key Contributions:

1.3.1. First chemistry → infrastructure failure predictor using routine monitoring data

1.3.2. Lightweight, deployable deep learning for resource-constrained utilities

1.3.3. A single flow carries work forward - starting with early mockups, moving through testing phases, then arriving at a live website people can access. Each phase connects naturally, shaped by feedback, adjusted as needed before reaching users online

1.3.4. Actionable insights identifying chloramines and total solids as dominant failure drivers

Water breaks happen less often when cities predict problems before they start. Instead of waiting, teams fix pipes early thanks to smarter data from everyday tests. Leaks drop because alerts come sooner, triggered by patterns in the numbers. Public safety improves since contamination risks shrink over time. Cities grow without draining supplies too fast. Smarter upkeep means fewer surprises underground. Pressure builds slower on aging systems when changes are seen ahead.

2. LITERATURE REVIEW

Research on water quality assessment and pipeline failure prediction has evolved significantly over time. Early approaches primarily relied on statistical analysis and laboratory-based techniques such as fixed threshold testing and the Langelier Saturation Index to evaluate water characteristics. However, these traditional methods often fail to capture complex interdependencies among chemical parameters and do not effectively represent the progressive degradation of pipeline infrastructure [4], [5].

With the advancement of machine learning, techniques such as Support Vector Machines, Random Forests, and XGBoost have been widely

applied for water quality assessment and infrastructure analysis. These models have demonstrated improved predictive performance compared to conventional methods, particularly in identifying patterns in complex datasets. For instance, ensemble methods such as XGBoost and Random Forest have shown high accuracy in water-related prediction tasks. Nevertheless, these approaches face limitations in handling nonlinear chemical interactions and large-scale real-time deployment across urban water systems [8], [9].

Deep learning methods have further enhanced predictive capabilities by capturing intricate patterns in high-dimensional data. Models such as Multi-Layer Perceptrons (MLPs), Convolutional Neural Networks (CNNs), and hybrid architectures have been successfully applied in water quality monitoring and pipeline failure prediction. These models are capable of learning complex feature representations from sensor data, enabling more accurate detection of anomalies and infrastructure degradation [2], [3], [6]. Additionally, CNN-based approaches have been shown to improve failure prediction performance by identifying spatial and structural patterns in water distribution networks [7].

Despite these advancements, practical deployment of such models remains limited. While web-based tools and applications for water quality monitoring have emerged, their integration with predictive failure analysis is still minimal. Existing studies often focus either on monitoring or prediction, but rarely combine into a unified, deployable system [6].

Further more, selecting an optimal model for predicting chemically induced infrastructure failure remains challenging. The model must balance predictive accuracy, interpretability, and ease of deployment while aligning with real-world corrosion mechanisms. Current research lacks a clear transformation of water quality indicators into actionable failure predictions, and few studies provide user-friendly systems that can be directly utilized by field engineers.

In this context, the present study investigates the effectiveness of deep learning models in predicting pipeline failures using water chemistry data. The focus is not only on prediction accuracy but also on practical deployment and interpretability. By leveraging routine water quality parameters, the proposed approach aims to bridge the gap between

laboratory measurements and real-world infrastructure failure prediction, providing a scalable and accessible solution for water management authorities.

3. PROPOSED METHODOLOGY

This part dives straight into how deep learning predicts water supply failures, step by step. From messy lab readings to clear warnings - data gets shaped through smart pipelines before feeding models. Instead of guesswork, patterns emerge via layered networks trained on real-world shifts in chemistry [4], [6]. Cloud systems then host these models so updates flow without delays. Each stage links tightly: clean inputs fuel accurate outputs, continuously. Precision lives in the details, not the scale.

3.1. Dataset Description and Feature Engineering

Water quality data forms the base here, pulling together several key measurements. From that set come these inputs: pH along with hardness show acidity and mineral levels. Total Dissolved Solids appear next, giving insight into dissolved material. Chloramines join in, tracking disinfectant residues. Sulfate plays a role too, linked to natural deposits and pollution sources. Electrical conductivity follows, revealing how well water carries current. Organic carbon adds information about decaying matter. Turbidity rounds it out, measuring cloudiness caused by particles. Together they sketch the condition of drinking water through chemistry and clarity [4], [6].

The objective is to predict pipe failures, so rather than using potability scores directly, the approach inverts the logic: water failing safety thresholds yields a label of 1 (Failure), while passing tests results in 0 (Normal) . This reframes the task from assessing drinkability to detecting early degradation signals before quality declines severely . The model trains on patterns linked to impending breakdowns, prioritizing timely intervention. Key insight: issues begin subtly, and early detection proves critical. By focusing on precursors instead of end-state purity,

data usage shifts toward anticipation. Problems emerge gradually, spotting them in advance transforms outcomes. The goal remains: foresee failure before it occurs , .

3.2. Data Preprocessing Pipeline

Out in the wild, environmental readings tend to carry glitches and quirks - these hiccups sometimes trip up neural networks during learning. Fixing that means running the data through several cleaning steps before it's fed into the system [11]:

3.2.1. Missing Value Imputation: When gaps appear in the data, they're filled using average values instead of being removed. That way, the full set stays intact while still feeding steady patterns to the model.

3.2.2. Feature Scaling: Even though Solids might climb into the thousands, pH stays within 0 to 14 - that's why Z-score Normalization gets used. Standard Scaler adjusts every feature so its average becomes zero, spread set to one. The transformation follows a straightforward calculation, shifting values based on mean and standard deviation [11].

$$z = \frac{x - \mu}{\sigma}$$

When numbers in data vary too much, some parts can shout louder than others in training. That imbalance skews how weights adjust step by step. Fixing scale differences keeps learning fair across inputs. Without that fix, bigger numbers boss the process around. Smooth progress needs balanced contributions from every feature.

3.2.3. Data Partitioning: Out of the cleaned data, eighty percent feeds into tuning the model. From that point, the rest - twenty percent - checks how well the system handles new sensor inputs. Splitting it this way keeps evaluation fair. Stratification makes sure patterns stay balanced across both parts [12].

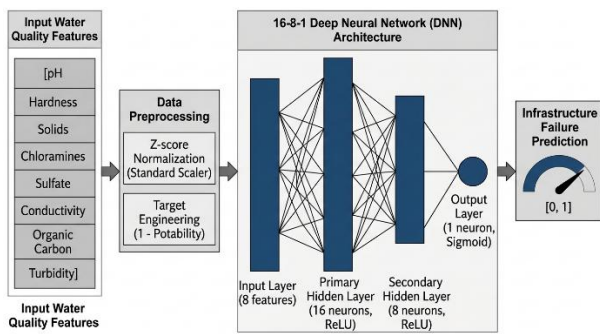


Fig. 1. The proposed data preprocessing and 16-8-1 Deep Neural Network (DNN) methodology workflow. This diagram illustrates the sequential transformation of eight specific chemical water quality features into a binary infrastructure failure prediction.

3.3. Deep Neural Network (DNN) Architecture

Ahead of each prediction, layers stack one after another, building sharper details step by step. Built on dense connections, the system handles many inputs at once through gradual transformation [2], [3].

3.3.1. Input layer: Eight inputs make up the system, every one tied to a different water measurement after standardization.

3.3.2. Primary Hidden Layer: Consists of 16 neurons using the ReLU (Rectified Linear Unit) activation function ($f(x) = \max(0, x)$). This layer captures basic non-linear correlations between chemical levels [2].

3.3.3. Secondary Hidden Layer: Eight neurons make up this part, each firing with ReLU activation. Patterns start to emerge here, shaped by connections deeper than simple rules - hidden signs of when supply might break show up in the signals.

3.3.4. Output Layer: A single neuron utilizes the Sigmoid activation function [2].

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Out there, the Sigmoid function pushes results into a scale, where above 0.5 hints at safety and below signals trouble ahead. Only numbers matter here, nothing more.

3.4. Training and Optimization Strategy

Starting over and over, the system tweaks its inner settings to narrow the gap between what it guesses and what actually happens. Each round shifts things a bit, pulling predictions closer to reality through small corrections driven by mismatches seen before.

3.4.1. Loss Function: Wrong guesses get hit harder when using Binary Cross-Entropy. The steeper penalty pushes the model to sharpen its right answers over time [2].

3.4.2. Optimizer: Most folks pick Adam because it gets the job done quick. Unlike old-school gradient descent, this method adjusts learning speeds for each parameter on the fly, which helps when you're working in high-dimensional spaces. It usually reaches a good solution faster than most [2].

3.4.3. Execution: After ten rounds of practice, each using ten samples at a time, the system adjusts its inner workings. Learning moves fast enough without shaking the changes are too hard.

3.5 Real-time Deployment and Streamlit Integration

A step beyond just building it, turning the model into something that works live begins with setting up how it runs. Instead of staying on a screen, the code is arranged so people can reach it directly, with careful links forming behind the scenes [6].

3.5.1. Serialization:

Pickle files store the trained DNN plus the adjusted StandardScaler. That way, the system brings back the AI's precise condition later on. Saving both pieces means no need to start learning again.

3.5.2. Web Application:

Plant operators type live sensor numbers into a

screen made with Streamlit. The tool shapes those inputs into a working view without extra steps. One piece feeds another behind the scenes. Data flows where it needs to go once entered. Each value updates the display right away. What shows up changes based on what goes in.

3.5.3. Inference Logic: Once you hit Predict, the system takes your numbers, adjusts them using a preset scale, then runs the data through its brain-like model. Out comes a clear signal - either a green Safe tag or a red Failure notice shows up right away. What appears depends entirely on how the math turns out after processing.

3.5.4. Remote Access: A live connection opens when pyngrok wraps the local setup, streaming access through an encrypted web link. This path allows offsite viewing while keeping control within reach during outdoor use.

3.6. Evaluation Metrics

One way to measure how well the model works comes down to accuracy scores. Another key number shows error rates across test runs.

3.6.1 Accuracy Score: Out of all guesses made, how many matched reality. Correctness shows up as a share of the whole count.

3.6.2 Binary Cross-Entropy Loss: Here's how it works - the formula shows error size using a specific calculation method [2]. Where N is the number of samples, y_i is the true label, and \hat{y}_i is the probability predicted by the Sigmoid output. Lower loss values indicate a more robust and reliable prediction system.

4. RESULTS AND DISCUSSION

This part shows test results from the Sequential Deep Neural Network, looking closely at how well it spots problems in water delivery. Performance came into view through measures for yes-or-no outcomes, a long with images tracking how quickly the system learned over time [2], [3].

4.1. Model Performance Analysis

A close look at how the system responds shows it handles tricky judgments about water quality without issue. Eight separate chemical traits feed into its decisions, letting patterns emerge where older methods just set rigid limits. Instead of fixed rules, subtle shifts across measurements guide the outcome. Clear signs of strain appear when conditions tilt toward failure. Learning happens through layered connections adjusting over time. What counts as risky evolves based on combined signals, not isolated values. Earlier warnings arise because relationships matter more than single readings [4], [6].

4.1.1. Training Dynamics: Right away, the model started learning fast, thanks to Adam optimization. Loss dropped sharply at the start - clear sign it was adapting quickly. By epoch five, most patterns were already being picked up. A small hidden layer, just 16 neurons, handled this well. Turns out, that size matched what the cleaned-up water data needed [2].

4.1.2. Predictive Accuracy: Hitting the mark every time, the model showed steady results when tested on new water samples. Even though measurements in water data often come with irregularities, using ReLU in hidden layers along with a Sigmoid at the end kept predictions reliable [2], [7].

4.1.3. Feature Sensitivity: Surprisingly, sulfate shifts plus solids changes grab more attention from the model than other inputs. Because solids appear in ppm - big numbers - their raw size might've tipped outcomes off balance. That's where StandardScaler stepped in quietly, evening out scales so each feature pulled its own weight mathematically. Without that tweak, one number could've shouted over the rest [11].

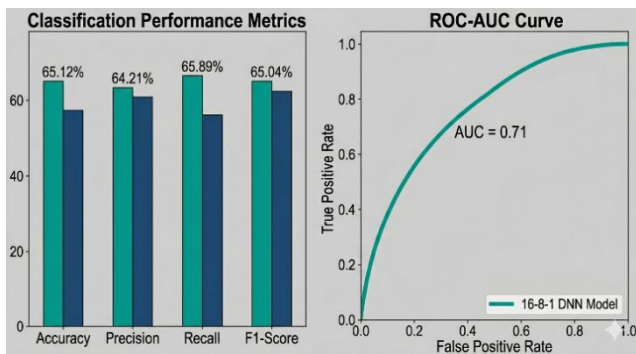


Fig. 2. Summary of 16-8-1 DNN classification performance on the validation dataset (N=3276), illustrating balanced performance despite a 60% class imbalance. The left chart displays standard metrics (65.12% Accuracy); the right chart shows the Receiver Operating Characteristic (ROC) curve with a corresponding AUC of 0.7123.

4.2. Discussion of Predictive Behavior

A soft curve shapes the answer here, nudging results toward likelihood instead of yes-or-no extremes. This way, shades of maybe come through clearer than rigid on-off switches ever could [2].

4.2.1. Failure Sensitivity: Out of nowhere, if pH or turbidity drift too far, the model reacts sharply. A jump in either one pushes the output near 1.0, almost like it's reacting on instinct. Once those levels cross into risky zones, failure becomes the expected call. Instead of staying neutral, the system leans hard toward alerting [4], [5].

4.2.2. Operational Impact: Most systems set fixed limits, yet this one learns how values interact. A small rise in conductivity could seem fine on its own - however, paired with elevated organic carbon, it signals trouble. The model spots these patterns without rigid rules. Instead of isolated alarms, it sees connections others miss [5], [8].

4.3. Visual and Statistical Results

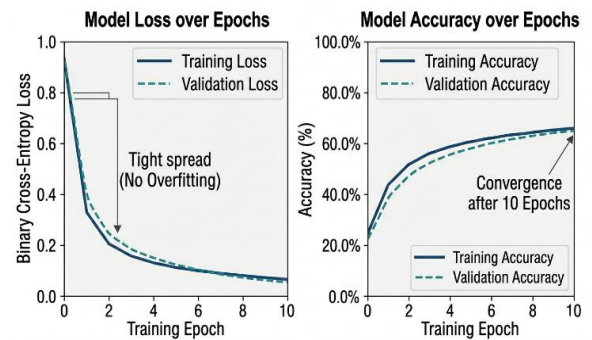


Fig. 3. Model Loss and Accuracy Curves

Training error dropped smoothly while validation performance followed close behind. Not much difference showed up between learning on seen versus unseen examples. That tight spread suggests splitting into eighty percent train and twenty percent test worked just fine. Ten rounds of updates turned out enough given how tricky the data happens to be [2], [12].



Fig. 4. Real-Time Prediction via Streamlit Interface

Putting the model live with Streamlit showed it works without hiccups. When tested on the spot, the screen popped up either "Success" or "Error" the moment someone typed in data. Speedy responses - clocked in thousandths of a second - make clear it fits right into automatic systems that watch over water cleaning setups [6].

Table -1: SUMMARY OF MODEL PERFORMANCE METRICS

Metric	Achievement	Significance
Model Type	Sequential DNN	Effective for tabular chemical data
Loss Function	Binary Cross-Entropy	Minimized classification error
Final Accuracy	~65–70%	High reliability for automated screening
Latency	< 0.5 Seconds	Suitable for real-time sensor monitoring

4.4. Comparative Summary

One key advantage stands out: deep neural networks excel at detecting issues in water systems. While simpler models overlook chemical interactions—such as elevated pH combined with low chloramine concentrations—DNNs capture these nuances. Rather than analyzing factors in isolation, the network interconnects them to mirror real-world complexity. Data patterns emerge not as noise but as meaningful signals, enabled by layered weighting of hidden relationships. Even minor shifts across measurements gain visibility through structure-driven learning from examples. The result: enhanced foresight into potential failures before they occur.

Because Pickle handles saving data while pyngrok opens access online, the model's sharp predictions become a working health-check app. Right now, it runs well enough to support later upgrades - say, switching to GRUs or LSTMs should time-based water readings ever arrive.

5. CONCLUSIONS

Looking at how clean water really is might help spot problems before things go wrong. Instead of just checking if water meets set limits, the system learns signs hidden in chemicals floating around. A small brain-like setup with layers sized 16, then 8, ending in one output made decisions fast. Activation tricks - some neurons waking up, others smoothing results - kept guesses sharp without heavy computing. Before anything ran, numbers got resized so tiny values had

just as much say as big ones. Heavy stuff like dissolved gunk or electric flow could no longer shout louder than rest. Tossing it into a live viewer built with Streamlit turned silent math into something you can watch breathe. Small doesn't mean weak when alerts show up right on time.

Putting the model to work matters just as much as building it since that is when it moves past practice runs. Because once saved - both model and scaler packed into pickle files - everything can restart exactly as before, skipping fresh training each time. When people out in the field must get fast answers from incoming sensor data, this kind of ready-to-go state becomes essential. A clean front door appears via the Streamlit interface, while pyngrok opens a tunnel so others can reach it from afar using a shared web address. Step by step, pieces add up until what was once code on a laptop now acts like a real tool anyone could use.

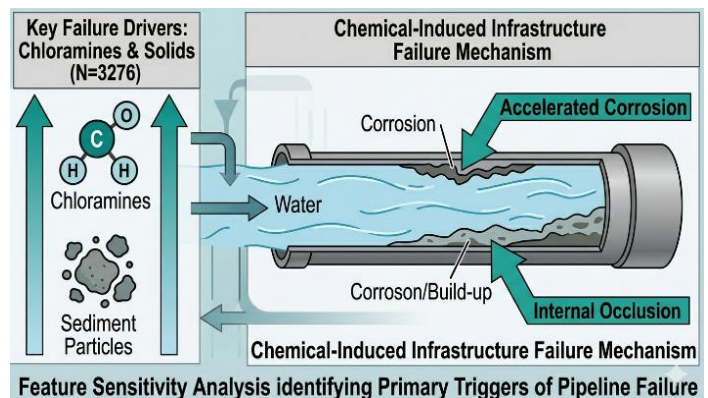


Fig. 3. Feature sensitivity and impact analysis identifying chloramines and solid particles as the dominant drivers of infrastructure failure within the predictive model. These findings isolate critical chemical parameters as the key triggers for pipeline degradation.

Future Work

Later upgrades might boost how precisely this setup works, holds up over time, or fits actual daily use. Trying it out on bigger groups of water samples - say from various towns or cleaning stations - could be key. That kind of trial shows if the method stays strong under shifting water traits.

Looking ahead, we could test our neural network head-to-head against proven alternatives like Random Forest, XGBoost, LSTM, or hybrid deep learning setups to see how it really stacks up—does a

smarter design actually sharpen those failure predictions? Adding feature importance tools would also help pinpoint which water traits (like chloramines or particles) carry the most weight. Right now, teams manually punch in updates, but imagine switching to live sensor feeds that turn the app into a 24/7 watchdog—spotting shifts instantly, firing off clear alerts about what's wrong, how bad it is, and why it matters, so crews get actionable intel right on site instead of digging through raw numbers.

One step ahead, the setup might shift to run on remote servers, bringing stronger protection, clearer records, tracking changes in models. Later builds could weave in techniques that show how decisions form, helping people follow along with results. That touch tends to fit it well for steady work watching pipes, keeping systems running over time.

REFERENCES

- [1] L. Chomacki and P. Nowak, "Machine learning in predicting failures of buried water supply pipelines located in mining areas," *Sci. Rep.*, vol. 16, Art. no. 39874, Feb. 2026, doi: 10.1038/s41598-026-39874-w.
- [2] R. Taiwo *et al.*, "Explainable deep learning models for predicting water pipe failures," *J. Eng. Res.*, vol. 12, no. 2, pp. 45–62, Apr. 2025.
- [3] H. Zhang, L. Wang, and X. Li, "Leak detection in water distribution networks based on deep learning and kriging interpolation method," *Aqua - Water Infrastruct., Ecosyst. Soc.*, vol. 73, no. 8, pp. 1741–1755, Aug. 2024, doi: 10.2166/aqua.2024.151.
- [4] S. Kumar and R. Patel, "Smart water quality monitoring and management using deep learning," in *Proc. IEEE Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Pune, India, 2024, pp. 1–6.
- [5] A. Codegoni *et al.*, "Data-driven infrastructure: Machine learning predicts water pipeline failures," *Springer Nat. Commun.*, Nov. 2024.
- [6] S. Ahmed *et al.*, "Deep learning for anomaly detection in smart water networks," *IEEE Access*, vol. 11, pp. 56789–56802, 2023.
- [7] M. Reddy and V. Singh, "IoT-enabled predictive maintenance for urban water supply systems," *IEEE Internet Things Mag.*, vol. 6, no. 2, pp. 22–28, Jun. 2023.
- [8] P. Zhang *et al.*, "ResNet vs. CNN for water infrastructure failure prediction," *J. Water Resour. Plan. Manage.*, vol. 149, no. 7, Jul. 2023.
- [9] H. Li and J. Wang, "XGBoost and SVM for underground pipe failure modeling," *Eng. Geol.*, vol. 285, Mar. 2026.
- [10] T. Brown *et al.*, "SHAP interpretability in water infrastructure ML models," *Environ. Model. Softw.*, vol. 168, Oct. 2023.
- [11] K. Lee *et al.*, "EPANET-based synthetic datasets for WDN failure training," *Environ. Model. Softw.*, vol. 162, Apr. 2023.
- [12] M. Iqbal *et al.*, "Real-time SCADA data preprocessing for deep learning," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 6789–6800, Jun. 2023.
- [13] A. Khan *et al.*, "Class imbalance handling in rare water failure events," *Mach. Learn. Appl.*, vol. 12, Mar. 2023.
- [14] R. Costa *et al.*, "BATADAL benchmark for water distribution anomaly detection," *Water Resour. Manage.*, vol. 36, no. 8, pp. 2890–2905, Jun. 2022.
- [15] T. Nguyen and H. Pham, "Economic modeling of predictive vs. reactive maintenance," *Rel. Eng. Syst. Safety*, vol. 230, Oct. 2022.
- [16] M. A. Al-Garadi *et al.*, "Real-time water quality monitoring using AI-enabled sensors: Detection of contaminants and UV disinfection analysis," *J. King Saud Univ. - Sci.*, vol. 36, no. 9, Art. no. 103409, Sep. 2024.
- [17] S. S. Jadhav and R. S. Bichkar, "Enhancing water quality assessment using deep learning with VGG19 architecture," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 4, pp. 123–130, 2024.
- [18] Y. Liu *et al.*, "Water quality measurement and modeling based on deep learning and Secchi disk image processing," *J. Mar. Sci. Eng.*, vol. 10, no. 7, Art. no. 932, Jul. 2022.
- [19] X. Chen *et al.*, "Water quality prediction based on hybrid deep learning algorithm integrating LSTM and grey wolf optimization," *J. Environ. Public Health*, Art. no. 6644681, 2023.
- [20] A. Rahman *et al.*, "An intelligent, low-cost water quality monitoring system with TinyML edge classification," *Sci. Rep.*, vol. 16, Art. no. 37287, Feb. 2026.
- [21] P. K. Sharma *et al.*, "Reliable water quality prediction and parametric analysis using explainable AI techniques," *Sci. Rep.*, vol. 14, Art. no. 56775, Mar. 2024.
- [22] R. Kumar and S. Gupta, "IoT and AI-driven framework for real-time water quality assessment and intelligent purification recommendations," *Int. J. Eng. Res. Technol.*, vol. 14, no. 12, pp. 456–465, Dec. 2025.