Jal Sarovar: ML-Driven Water Quality Monitoring

Abstract

This research presents a comprehensive machine learning framework for water quality monitoring in India, addressing critical limitations of traditional laboratory testing through a dual-scale monitoring paradigm. At the public scale , the LEGOLAS autonomous robotic platform surveys geographically distributed water bodies (stepwells, tanks, ponds, rivers) with multi-parameter sensors (pH, TDS, turbidity, dissolved oxygen, temperature) and computer vision capabilities for population-level surveillance. At the residential scale , low-cost IoT sensors (8,000-15,000 per household) provide continuous real-time monitoring with 15-minute sampling intervals for immediate contamination alerts. The system integrates five production-ready machine learning models trained on a comprehensive dataset of [TOTAL_SAMPLES] water samples from [TOTAL_SITES] monitoring sites spanning [DATE_RANGE] years. Models achieve 84.6% accuracy for site risk classification , 73.4% for contamination type detection , and R^2=0.94 for time-series forecasting , while demonstrating [COST_REDUCTION]% cost reduction through intelligent resource allocation compared to traditional exhaustive testing. A unique hybrid data architecture enables coexistence of manual data collection (CSV uploads, API integration, laboratory submissions) with autonomous sensor streams (LEGOLAS measurements, IoT readings), weighted by reliability for decision-making. The framework supports multiple government initiatives including Mission Amrit Sarovar (68,000+ water bodies nationally), Jal Jeevan Mission (rural drinking water safety), and state/municipal water quality programs. Following successful proof-of-concept deployment and ML model validation, the system is currently scaling autonomous monitoring infrastructure while maintaining continuous bulk data import from existing monitoring networks. This incremental transition preserves human expertise and laboratory validation as essential components of water safety decision-making.

Contents

Machine Learning Framework Trained on Bulk Imported Data with Phased LEGOLAS Robotic and IoT Sensor Deployment

1 Introduction

1.1 Motivation and Project Objectives

Safe drinking water is fundamental to human health, yet approximately 2 billion people globally lack access to safely managed drinking water services. In India alone, water contamination contributes to 37.7 million cases of waterborne diseases annually. Traditional water quality monitoring approaches face critical limitations: high cost (estimated Rs. 1,000-2,000 per laboratory sample), delayed detection (monthly/quarterly testing misses rapid contamination events), and limited coverage preventing comprehensive monitoring of all water sources.

Jal Sarovar addresses these challenges by supporting the goals of various Government initiatives in India at both State and Central levels, including:

- Mission Amrit Sarovar: Supporting the development and rejuvenation of 75 water bodies in each district (approximately 68,000 water bodies nationwide)
- Jal Jeevan Mission: Enabling water quality monitoring for safe and adequate drinking water to rural households
- Municipal Administration and Urban Development Departments: Providing data-driven insights for urban water body management at State and Central levels
- Water Supply and Drainage Boards: Supporting continuous monitoring and quality assurance for public water supply infrastructure

Beyond supporting these national and state-level initiatives, the project establishes a framework for nationwide collaboration for tracking water quality in both public water bodies and private sites, enabling comprehensive monitoring at multiple scales and supporting evidence-based policy decisions with a truly national and global collaborative approach based on sharing of water quality parameters across regions.

1.2 The Dual-Scale Paradigm

Water quality management inherently operates at two distinct scales, each serving complementary purposes:

- Geographically distributed public water bodies (rivers, stepwells, tanks, ponds)
- Periodic testing via LEGOLAS autonomous robots (weekly to quarterly)
- Population-level surveillance and policy decisions
- Cost-optimized resource allocation
- Commercial and Residential communities
- Continuous real-time monitoring (15-60 minute intervals)
- Water safety assurance and immediate alerts
- Consumer-grade affordability

These scales are complementary: public site monitoring identifies systemic contamination patterns and informs policy interventions, while private site monitoring provides granular safety verification and empowers institutions and individuals. This proof-of-concept validates the first integrated system combining both scales with coordinated ML analytics, supporting multiple initiatives' goals of comprehensive water body rejuvenation and establishing a model for national and global water quality collaboration.

1.3 ML-Driven Capabilities

Jal Sarovar provides a comprehensive ML pipeline with the following capabilities:

- Bulk Data Import & Processing: CSV uploads, API integration, and mobile app submissions from NABL/ISO 17025 accredited laboratory test results

- 5 Production-Ready ML Models: Site Risk Classifier (84.6% accuracy), Contamination Classifier (73.4% accuracy), Water Quality Forecaster (R^2=0.94), Bayesian Cost Optimizer, and Hybrid Anomaly Detector
- LEGOLAS Autonomous Robotic Testing: Multi-parameter water quality sensors with computer vision for algae detection deployed on public water bodies
- IoT Real-Time Monitoring: Low-cost inline sensors (8,000-15,000) for continuous household water quality monitoring with 15-minute sampling intervals
- Hybrid Data Architecture: Multi-source data fusion combining manual laboratory testing with autonomous sensor streams, weighted by reliability (laboratory ¿ LEGOLAS ¿ IoT)

See Section 12.1 Implementation Roadmap for detailed phased deployment timeline.

1.4 Implementation Phases

The Jal Sarovar framework development follows a six-phase incremental approach, progressing from foundational infrastructure to long-term hybrid operations. A detailed Implementation Roadmap with status, milestones, and timelines is provided in Section 12.1 .

See Section 12.1 Implementation Roadmap for complete phase details, technical specifications, and timelines.

2 System Overview

2.1 Data Sources and Collection Methodology

2.1.1 Data Collection Sources

The Jal Sarovar system operates on a hybrid data collection model combining multiple sources. Bulk imported laboratory test results from [TOTAL_SITES] public monitoring sites serve as the primary data foundation, supplemented by autonomous monitoring from LEGOLAS robotic platforms and IoT sensor networks. This comprehensive dataset spans [DATE_RANGE] years ([MIN_DATE] to [MAX_DATE]) and includes [TOTAL_SAMPLES] water samples with [TOTAL_TESTS] parameter measurements.

2.1.2 Data Import Methods

Method	Source	Frequency	Parameters	Status
CSV Upload (Bulk Import)	Government agencies, research institutions, NGOs	Batch imports (daily/weekly/-monthly/quarterly)	Full 45-parameter test results from accredited labs	Primary Source
API Integration	Partner laboratories, government databases	Daily/Weekly sync	Standardized JSON format with WHO/BIS validation	Active
Manual Entry	Field teams, community volunteers	On-demand	Field test kits (pH, turbidity, TDS, chlorine, coliform)	Active
Mobile App	Field operators (manual sampling for lab analysis)	Site visits	GPS-tagged sample collection sent to laboratories	Scaling Up
LEGOLAS Autonomous Robot	Autonomous robotic platform on public water bodies	Continuous autonomous monitoring	Multi-sensor (pH, turbidity, DO, temperature, conductivity)	Scaling Up
IoT Sensors	Inline sensors on residential water systems	Real-time monitoring (15-min intervals)	pH, turbidity, TDS, chlorine, temperature	Scaling Up

2.1.3 Data Quality and Validation

- Laboratory Accreditation: All samples tested by NABL/ISO 17025 certified laboratories What is NABL/ISO 17025? NABL (National Accreditation Board for Testing and Calibration Laboratories): India's premier accreditation body established by the Government of India. NABL accreditation ensures that testing laboratories

meet internationally recognized standards for technical competence, quality management, and reliable testing practices. NABL operates under the Department of Science and Technology. ISO/IEC 17025: International standard titled "General requirements for the competence of testing and calibration laboratories." This standard specifies requirements for: Technical competence of laboratory personnel Validity and reliability of test results Quality management systems Proper equipment calibration and maintenance Traceability of measurements to international standards Examples of NABL-accredited water testing laboratories in India: State Public Health Engineering Departments (PHED) laboratories Central Pollution Control Board (CPCB) regional laboratories State Pollution Control Board (SPCB) laboratories Indian Institute of Technology (IIT) environmental testing labs National Institute of Hydrology (NIH) laboratories Private NABL-accredited laboratories (e.g., Eurofins, SGS India, TÜV SÜD) Municipal corporation water quality testing facilities Why NABL/ISO 17025 certification matters: Ensures that water quality test results are accurate, reproducible, and legally defensible. Government agencies, courts, and regulatory bodies accept test reports only from NABL/ISO 17025 accredited laboratories for compliance verification, legal proceedings, and policy decisions.

- Parameter Standardization: Measurements converted to WHO/BIS standard units (mg/L, NTU, MPN/100mL)
- Outlier Detection: Statistical validation removes measurement errors and sensor malfunctions
- Geospatial Validation: GPS coordinates verified against known water body locations
- Temporal Consistency: Time series analysis flags unrealistic parameter changes
- NABL (National Accreditation Board for Testing and Calibration Laboratories): India's premier accreditation body established by the Government of India. NABL accreditation ensures that testing laboratories meet internationally recognized standards for technical competence, quality management, and reliable testing practices. NABL operates under the Department of Science and Technology.
- ISO/IEC 17025: International standard titled "General requirements for the competence of testing and calibration laboratories." This standard specifies requirements for: Technical competence of laboratory personnel Validity and reliability of test results Quality management systems Proper equipment calibration and maintenance Traceability of measurements to international standards
- Examples of NABL-accredited water testing laboratories in India: State Public Health Engineering Departments (PHED) laboratories Central Pollution Control Board (CPCB) regional laboratories State Pollution Control Board (SPCB) laboratories Indian Institute of Technology (IIT) environmental testing labs National Institute of Hydrology (NIH) laboratories Private NABL-accredited laboratories (e.g., Eurofins, SGS India, TÜV SÜD) Municipal corporation water quality testing facilities
- Why NABL/ISO 17025 certification matters: Ensures that water quality test results are accurate, reproducible, and legally defensible. Government agencies, courts, and regulatory bodies accept test reports only from NABL/ISO 17025 accredited laboratories for compliance verification, legal proceedings, and policy decisions.
- Technical competence of laboratory personnel
- Validity and reliability of test results
- Quality management systems
- Proper equipment calibration and maintenance
- Traceability of measurements to international standards
- State Public Health Engineering Departments (PHED) laboratories
- Central Pollution Control Board (CPCB) regional laboratories
- State Pollution Control Board (SPCB) laboratories
- Indian Institute of Technology (IIT) environmental testing labs
- National Institute of Hydrology (NIH) laboratories
- Private NABL-accredited laboratories (e.g., Eurofins, SGS India, TÜV SÜD)
- Municipal corporation water quality testing facilities

2.1.4 Role of Data in System Intelligence

Bulk imported historical data serves as the primary training foundation for all ML models, supplemented by preliminary trial data from LEGOLAS and IoT systems (Phase 1). The 20,269+ labeled samples enable the system to:

- Learn contamination risk patterns across different site types and geographic regions
- Recognize chemical signatures of specific contamination sources (sewage, runoff, corrosion)
- Forecast seasonal water quality trends and predict threshold violations
- Optimize resource allocation by understanding detection-cost tradeoffs
- Detect both sudden anomalies and gradual drift in water quality parameters

All 5 production-ready ML models are operational , trained on bulk imported historical laboratory data supplemented by autonomous sensor measurements from LEGOLAS robots and IoT networks. The system continues to import existing laboratory samples while integrating real-time autonomous sensor data for production predictions. The hybrid operational model balances high-reliability laboratory testing with high-frequency autonomous monitoring, weighted by data source reliability (laboratory ¿ LEGOLAS ¿ IoT) for decision-making.

See Section 12.1 Implementation Roadmap for detailed deployment timeline and scaling strategy.

2.2 LEGOLAS: Autonomous Robotic Architecture

LEGOLAS is a low-cost autonomous water quality assessment platform for continuous monitoring of public water bodies (stepwells, tanks, ponds, rivers). The platform integrates multi-parameter sensors with computer vision capabilities and GPS navigation, providing real-time data to production ML models for site risk assessment, contamination detection, and water quality forecasting.

- Raspberry Pi based System
- Fiberglass hull (1.2m $\times$ 0.8m $\times$ 0.4m)
- Dual brushless DC motors (300W each)
- GPS navigation (2.5m accuracy)
- 4G LTE real-time data upload
- Solar charging (14-day autonomy)
- pH sensor ($\pm$0.1 accuracy)
- TDS probe ($\pm$2% accuracy)
- Turbidity sensor (0.1-1000 NTU)
- Dissolved oxygen ($\pm$5%)
- Temperature ($\pm$0.5°C)
- 5MP RGB camera for imaging

Cost Analysis: LEGOLAS reduces estimated per-test cost to Rs. 540 (amortized over 3-year lifespan with 2 tests/week), a 73% reduction compared to manual sampling (estimated Rs. 2,000/test including transport and labor).

2.2.1 Computer Vision for Algae Detection

LEGOLAS integrates a MobileNetV2-based computer vision pipeline for algae classification from RGB surface images. The system performs 4-class classification: No algae (clear water), Green algae (Chlorophyta), Cyanobacteria (blue-green, toxic), and Diatoms (brown/golden algae).

Algae Class	Precision	Recall	F1-Score	Samples
No algae (clear water)	0.91	0.93	0.92	462
Green algae (Chlorophyta)	0.84	0.82	0.83	122

Cyanobacteria (toxic)	0.88	0.85	0.86	75
Diatoms (brown/golden)	0.79	0.75	0.77	20
Weighted Average	0.88	0.87	0.87	679

Bacterial Contamination Estimation: Multi-modal regression combining turbidity, color intensity, and dissolved oxygen achieves R^2=0.79 correlation with laboratory coliform tests (Pearson r=0.89, p¡0.001), enabling instant bacterial safety alerts without 24-48 hour laboratory delays.

2.3 IoT Sensor Architecture

The IoT sensor network provides continuous real-time water quality monitoring for residential and small-scale water systems. Low-cost inline sensors (8,000-15,000 per installation) measure key parameters at 15-minute intervals, enabling immediate contamination alerts and real-time anomaly detection through integrated ML models. This architecture is designed for large-scale deployment across households and community water distribution systems.

- Raspberry Pi or equivalent Microcontroller Based System
- pH sensor
- TDS sensor
- Turbidity sensor
- Temperature sensor
- Chlorine sensor
- IP65-rated enclosure
- Sampling: 15-minute intervals (configurable 5-60 min)
- Averaging: 3 measurements per cycle
- Outlier rejection for data quality
- Automated 2-point calibration every 7 days
- 12-hour battery backup during power outages
- Edge Computing: On-device WQI calculation (sub-100ms)
- Cloud sync via MQTT
- Time-series storage (InfluxDB, 90-day retention)
- SMS/push notifications via SNS (sub-5-second latency)

Deployment Scope: The framework demonstrates viability for nationwide expansion supporting the multiple water supply initiatives and enabling global water quality collaboration across diverse regional water quality characteristics and environmental patterns.

2.4 Data Summary

2.4.1 Total Sites

2.4.2 Water Samples

2.4.3 Date Range

2.4.4 Samples/Site

Metric	Value	Details
Sites Monitored	[TOTAL_SITES]	[STATES]
Water Samples	[TOTAL_SAMPLES]	[MIN_DATE] to [MAX_DATE]
Test Results	[TOTAL_TESTS]	˜[AVG_SAMPLES] samples per site

Site Risk Predictions	[RISK_PRED]	Critical: [RISK_CRIT], High: [RISK_HIGH], Medium: [RISK_MED],
Contamination Predictions	[DATA]	[DATA]
WQI Readings	[DATA]	Avg: [DATA], Classes: [DATA]
Water Quality Forecasts	[DATA]	R^2=[DATA] for pH, TDS, turbidity, temperature
Cost Optimization Results	[DATA]	[COST_REDUCTION]% avg reduction, [DATA]% detection

2.5 Water Quality Parameters

The Jal Sarovar system measures a comprehensive set of 40+ water quality parameters across physical, chemical, and microbiological categories. These parameters are captured in laboratory test results and analyzed by the ML pipeline to assess water safety and contamination patterns.

Category	Parameter	Unit
Physical Parameters	pH	-
Temperature	°C	Water temperature in Celsius
Turbidity	NTU	Cloudiness or haziness (threshold: ¡5 NTU)
Color	Hazen	Water color intensity
Odor	1-5 scale	Odor threshold rating
Taste	1-5 scale	Taste quality rating
Conductivity	µS/cm	Electrical conductivity (microsiemens per centimeter)
Chemical - General	TDS (Total Dissolved Solids)	ppm
Total Hardness	mg/L	Calcium and magnesium concentration
Calcium Hardness	mg/L	Calcium-specific hardness
Magnesium Hardness	mg/L	Magnesium-specific hardness
Total Alkalinity	mg/L	Buffer capacity against pH changes
Disinfection	Free Chlorine	mg/L
Total Chlorine	mg/L	Combined free and bound chlorine
Chlorine Residual	mg/L	Residual chlorine after reaction
Anions	Chloride	mg/L
Fluoride	mg/L	Fluoride content (optimal: 0.5-1.5 mg/L)
Sulfate	mg/L	Sulfate concentration
Nitrate	mg/L	Nitrate-nitrogen (threshold: ¡50 mg/L)
Nitrite	mg/L	Nitrite-nitrogen concentration
Phosphate	mg/L	Phosphate concentration
Cations / Metals	Iron	mg/L
Manganese	mg/L	Manganese concentration (threshold: ¡0.1 mg/L)
Copper	mg/L	Copper content
Zinc	mg/L	Zinc concentration
Lead	mg/L	Lead content (threshold: ¡0.01 mg/L)
Arsenic	mg/L	Arsenic concentration (threshold: ¡0.01 mg/L)
Chromium	mg/L	Chromium content
Cadmium	mg/L	Cadmium concentration (threshold: ¡0.003 mg/L)
Mercury	mg/L	Mercury content (threshold: ¡0.001 mg/L)
Nickel	mg/L	Nickel concentration
Aluminum	mg/L	Aluminum content
Sodium	mg/L	Sodium ion concentration
Potassium	mg/L	Potassium content
Nitrogen Compounds	Ammonia	mg/L
Total Nitrogen	mg/L	All forms of nitrogen
Organic Nitrogen	mg/L	Nitrogen in organic compounds
Organic Parameters	Dissolved Oxygen (DO)	mg/L
BOD (Biochemical Oxygen Demand)	mg/L	Oxygen required by bacteria to decompose organic ma
COD (Chemical Oxygen Demand)	mg/L	Oxygen required to chemically oxidize organic matter
TOC (Total Organic Carbon)	mg/L	Total organic carbon content

Microbiological	Total Coliform	MPN/100mL
Fecal Coliform	MPN/100mL	Fecal contamination indicator
E. coli	MPN/100mL	Escherichia coli presence (should be 0)
Total Plate Count	CFU/mL	Total viable bacteria count (colony forming units)
Pesticides/Herbicides	Pesticides Detected	Boolean
Pesticide Types	List	Specific pesticides identified

Parameter Coverage Assessment: The system evaluates data quality based on parameter coverage. Full assessment requires 3 key parameters; partial assessment requires 1 key parameter; insufficient data indicates no key parameters measured. This tiered approach ensures ML predictions are only generated when sufficient data quality is available.

2.6 ML Pipeline Architecture

2.6.1 Machine Learning Models (5 Total - Data-Driven Algorithms)

The following models use machine learning algorithms that learn from data and optimize their behavior.

- Site Risk Classifier: Random Forest (ensemble of decision trees for classification)
- Contamination Classifier: XGBoost (gradient boosted trees for multi-class classification)
- Water Quality Forecaster: Gaussian Process (Bayesian non-parametric regression with uncertainty quantification)
- Bayesian Cost Optimizer: Bayesian Optimization (GP-based black-box optimization with Expected Improvement)
- Hybrid Anomaly Detector: Isolation Forest + CUSUM (unsupervised anomaly detection with drift detection)

2.6.2 Deterministic Algorithm (Rule-Based - NOT Machine Learning)

The following is NOT machine learning . It uses fixed formulas and thresholds defined by WHO/BIS standards. Same inputs always produce the same output.

- Real-time WQI Calculator: Penalty-based scoring using WHO/BIS water quality standards (deterministic calculation)

3 Site Risk Classifier (Random Forest)

Purpose: Predict contamination risk level for water bodies to prioritize testing resources.

Algorithm: Random Forest with feature engineering from site characteristics and historical data.

3.1 How It Functions

3.1.1 Why Random Forest?

Random Forest was chosen for site risk classification because water contamination depends on complex interactions between multiple environmental factors. Unlike a single decision tree that might miss subtle patterns, Random Forest builds 100 independent trees, each learning from different data perspectives. This "wisdom of the crowd" approach handles the complexity of real-world contamination scenarios where industrial proximity might matter more for coastal sites than for inland agricultural areas.

Think of training as teaching 100 different experts to recognize contamination risk. Each expert (decision tree) looks at the same historical data but focuses on different aspects. One expert might learn: "If a stepwell is near industry AND hasn't been tested in 60+ days, it's high risk." Another learns: "Coastal urban sites with 20%+ contamination history are critical." During training on 20,269 historical site assessments, the forest discovers that recent contamination history (42% importance) and testing recency (23% importance) are the strongest signals.

3.1.2 Prediction Workflow

When evaluating a new site, each of the 100 trees votes for a risk category (critical, high, medium, or low). The forest counts votes: if 73 trees vote "high risk" and 27 vote "medium risk," the site is classified as "high" with 73% confidence. The model then recommends bi-weekly testing (26 tests/year) for high-risk sites based on learned associations between risk levels and optimal monitoring frequencies.

3.1.3 Real-World Example

A stepwell near an industrial zone in Gujarat with 35% historical contamination rate and 45 days since last test would trigger majority votes for "high risk" because the model learned this combination often precedes contamination events. The system recommends bi-weekly testing (26 tests/year) to catch problems before they spread.

3.1.4 When Invoked

This model runs during testing schedule optimization to assign monitoring frequencies across all sites in the network.

3.2 Input Parameters

Parameter	Type	Description	Example
site_type	string	Type of water body	"stepwell", "tank", "pond", "lake"
is_industrial_nearby	boolean	Industrial facilities within 5km	true/false
is_agricultural_nearby	boolean	Agricultural land within 2km	true/false
is_coastal	boolean	Within 10km of coastline	true/false
is_urban	boolean	Urban population density ¿500/km^2	true/false
contamination_rate_30d	float	Historical contamination % (30 days)	0-100
days_since_last_test	int	Days since last water test	0-365

3.3 Machine Learning Algorithm Details

Implementation: sklearn.ensemble.RandomForestClassifier

Training Data: 20,269 labeled samples from historical site assessments

3.3.1 Hyperparameters

- n_estimators: 100 decision trees in the ensemble
- max_depth: 8 levels per tree
- min_samples_split: 10 samples required to split a node
- class_weight: 'balanced' (handles class imbalance)
- random_state: 42 (for reproducibility)

3.3.2 Feature Engineering

- One-hot encoding for site_type (4 binary features)
- Boolean to binary: Environmental factors (0/1)
- Numerical features: contamination_rate_30d, days_since_last_test
- Total features: 10 input dimensions

3.3.3 Training Process

1. Split data: 80% training, 20% validation
2. Build 100 decision trees on random subsamples
3. Each tree learns different decision boundaries

4. Ensemble aggregation: majority voting across all trees
5. Calculate class probabilities from vote percentages

3.3.4 Model Performance

- Training Accuracy: 87.3%
- Validation Accuracy: 84.6%
- F1 Score: 0.835

3.3.5 Feature Importance (Learned from Data)

Feature	Importance	Interpretation
contamination_rate_30d	0.42	Strongest predictor
days_since_last_test	0.23	Testing recency matters
is_industrial_nearby	0.15	Industrial proximity risk
site_type_stepwell	0.12	Site-specific patterns
is_urban	0.08	Urban contamination signal

3.4 Output Parameters

Parameter	Type	Description
risk_level	string	"critical", "high", "medium", "low"
risk_score	float	0-100 numerical score
confidence	float	Model confidence percentage
recommended_frequency	string	"weekly", "bi-weekly", "monthly", "quarterly"
tests_per_year	int	Recommended annual tests (52, 26, 12, or 4)

3.5 Pseudocode

4 Contamination Classifier (XGBoost)

Purpose: Classify contamination source type to guide remediation strategies.

Algorithm: XGBoost multi-class classifier with probability outputs.

4.1 How It Functions

4.1.1 Why Gradient Boosting?

XGBoost was selected for contamination classification because different pollution sources create overlapping chemical signatures that require progressive refinement. Sewage might elevate both coliform and ammonia, while pipe corrosion raises iron and manganese but not bacteria. Gradient boosting excels at these nuanced distinctions by building trees sequentially, where each new tree focuses on correcting the mistakes of previous trees—learning the subtle differences between contamination types.

Imagine 100 increasingly specialized detectives investigating contamination. The first detective makes broad guesses: "High coliform = sewage." But this misses some cases. The second detective studies the first's errors: "High coliform WITHOUT elevated ammonia = agricultural runoff, not sewage." Each subsequent detective (tree) becomes an expert in the remaining confusing cases. By tree 100, the ensemble achieves 78.9% training accuracy on 20,508 labeled samples, with total_coliform (28% importance), turbidity (19%), and free_chlorine (17%) emerging as the strongest discriminators.

4.1.2 Prediction Workflow

For a new water sample, all 100 trees vote with weighted scores. Each tree applies learned decision rules like "if coliform ¿ 15 AND ammonia ¿ 0.8 → sewage_ingress (0.42 probability)." The final prediction aggregates these weighted votes through softmax conversion: sewage_ingress (65%), pipe_corrosion (22%), runoff_sediment (8%), others (5%). The model returns the top prediction (sewage) with 65% confidence.

4.1.3 Real-World Example

A residential water sample shows high turbidity (12 NTU), elevated coliform (25 MPN/100mL), high ammonia (1.2 mg/L), but normal iron levels. The model distinguishes this as sewage contamination (not pipe corrosion) because the ammonia-coliform combination without metallic signatures matches learned sewage patterns. This guides operators to inspect sewer lines rather than water pipes.

4.1.4 When Invoked

This model runs immediately after contamination detection (when WQI falls below thresholds) to identify the pollution source and guide targeted remediation.

4.2 Input Parameters

Parameter	Type	Unit	WHO/BIS Threshold
ph	float	-	6.5-8.5
turbidity	float	NTU	¡5
tds	float	ppm	¡500
chlorine	float	mg/L	0.2-5.0
iron	float	mg/L	¡0.3
manganese	float	mg/L	¡0.4
coliform	float	MPN/100mL	0
ammonia	float	mg/L	¡0.5
chloride	float	mg/L	¡250

4.3 Machine Learning Algorithm Details

Implementation: xgboost.XGBClassifier

Training Data: 20,508 labeled contamination samples

4.3.1 Hyperparameters

- n_estimators: 100 boosted trees
- max_depth: 6 levels per tree
- learning_rate: 0.1 (eta parameter)
- subsample: 0.8 (80% data per tree)
- colsample_bytree: 0.8 (80% features per tree)
- objective: 'multi:softprob' (multi-class probabilities)

4.3.2 Feature Engineering

- Water quality parameters (9 features): pH, turbidity, TDS, free_chlorine, iron, manganese, total_coliform, ammonia, chloride
- Environmental context (2 features): rained_recently, is_coastal
- Feature scaling: StandardScaler normalization (mean=0, std=1)
- Total features: 11 input dimensions

4.3.3 Training Process

1. Encode contamination types to integers (0-7)
2. Gradient boosting iteration: Fit each tree to residual errors of previous trees Each tree corrects ensemble mistakes Apply learning_rate to prevent overfitting Use subsample for randomness
3. Ensemble prediction: weighted sum of all 100 tree predictions
4. Softmax conversion to probabilities

- Fit each tree to residual errors of previous trees
- Each tree corrects ensemble mistakes
- Apply learning_rate to prevent overfitting
- Use subsample for randomness

4.3.4 Model Performance

- Training Accuracy: 78.9%
- Validation Accuracy: 73.4%
- Macro F1 Score: 0.68
- Best Detected: sewage_ingress (F1: 0.82)

4.3.5 Feature Importance (Learned via Gain)

Feature	Importance
total_coliform	0.28
turbidity	0.19
free_chlorine	0.17
TDS	0.12
ammonia	0.10
chloride	0.08
pH	0.06

4.4 Output Parameters

Parameter	Type	Description
predicted_type	string	Contamination source type
confidence	float	Probability × 100
prob_runoff_sediment	float	0-1 probability
prob_sewage_ingress	float	0-1 probability
prob_salt_intrusion	float	0-1 probability
prob_pipe_corrosion	float	0-1 probability
prob_disinfectant_decay	float	0-1 probability

4.5 Pseudocode

5 Water Quality Forecaster (Gaussian Process)

Purpose: Predict future water quality parameter values with uncertainty bounds.

Algorithm: Gaussian Process regression with trend and seasonality components.

5.1 How It Functions

5.1.1 Why Gaussian Process Instead of Linear Regression?

Water quality doesn't follow simple straight-line trends—pH might spike after monsoons, TDS increases gradually from salt intrusion, and turbidity shows seasonal patterns. Gaussian Processes (GP) excel here because they're non-parametric : instead of forcing data into a predetermined equation, GP learns the underlying pattern's shape from data. Crucially, GP provides Bayesian uncertainty quantification , telling operators "pH will be 7.2 ± 0.3" rather than just "7.2"—critical for proactive intervention planning.

Think of Gaussian Processes as drawing a "cloud of possible futures" based on historical patterns. The model learns a similarity function (kernel): measurements close in time should have similar values. Training on historical time series [(Jan 1: pH 7.4), (Jan 15: pH 7.3), (Feb 1: pH 7.1)...], the GP discovers that pH typically changes smoothly over ˜30-day periods (the learned length scale). The RBF kernel captures this: recent data points influence predictions more than distant ones, creating smooth, realistic forecasts.

5.1.2 Prediction Workflow

To forecast 90 days ahead, the GP computes similarity between the target date and all historical observations. It asks: "This future date is 45 days from our last reading—what did pH typically do after 45-day gaps historically?" The model outputs a probability distribution: mean prediction (most likely value) plus confidence bounds. For example: "30 days ahead, pH will be 7.3 ± 0.15 (95% confidence)" where uncertainty widens for distant predictions (90 days: ± 0.42).

5.1.3 Real-World Example

A coastal site shows TDS gradually increasing from 380 ppm (January) to 450 ppm (March). The GP forecast predicts TDS will reach 520 ppm by June (exceeding the 500 ppm threshold) with 82% confidence. This triggers proactive desalination planning rather than waiting for reactive contamination alerts—preventing 3 months of unsafe water.

5.1.4 When Invoked

This model runs weekly for monitored sites with sufficient historical data (minimum 10 samples) to generate 90-day forecasts for pH, TDS, and turbidity.

5.2 Input Parameters

Parameter	Type	Description
site_id	int	Unique site identifier
parameter	string	"ph", "turbidity", "tds", "chlorine", "temperature"
historical_data	array	List of {date, value} pairs (minimum 10 samples)
days_ahead	int	Forecast horizon (default: 90 days)

5.3 Machine Learning Algorithm Details

Implementation: sklearn.gaussian_process.GaussianProcessRegressor

Approach: Non-parametric Bayesian regression on time series

5.3.1 Kernel Function

RBF (Radial Basis Function) + White Noise:

- Formula: $k(x, x') = {}^{2}\_signal \times \exp(-\text{——}x - x'\text{——}^{2} / (2 \times l^{2})) + {}^{2}\_noise \times (x, x')$
- ^{2}_signal: Signal variance (learned from data)
- l: Length scale - determines smoothness (30 days)
- ^{2}_noise: Noise variance - measurement uncertainty (0.1)

5.3.2 Hyperparameters

- Kernel: RBF(length_scale=30) + WhiteKernel(noise_level=0.1)
- n_restarts_optimizer: 10 (avoid local optima)
- alpha: 1e-10 (numerical stability)
- normalize_y: True (zero-mean targets)

5.3.3 Training Process

1. Input: Historical time series [(t, y), (t, y), ..., (t, y)]
2. Compute Gram matrix K: K_ij = k(t, t) for all observation pairs
3. Maximum likelihood estimation: Optimize kernel hyperparameters
4. Maximize log p(y — X,) = -$\frac{1}{2}$ yK^{1}y - $\frac{1}{2}$ log—K— - n/2 log(2)
5. Store learned kernel parameters

5.3.4 Prediction Process (Bayesian Inference)

1. For new time t*: Compute k* = [k(t*, t), k(t*, t), ..., k(t*, t)] Compute k** = k(t*, t*)
2. Posterior mean (prediction): (t*) = k* K^{1} y
3. Posterior variance (uncertainty): ^{2}(t*) = k** - k* K^{1} k*
4. 95% confidence interval: [(t*) - 1.96(t*), (t*) + 1.96(t*)]

- Compute k* = [k(t*, t), k(t*, t), ..., k(t*, t)]
- Compute k** = k(t*, t*)

5.3.5 Model Performance

- R^{2} Score: 0.94 (94% variance explained)
- MAE (pH): 0.12 units
- MAE (TDS): 15.3 ppm
- MAE (Turbidity): 0.8 NTU
- MAE (Temperature): 1.2°C
- Calibration: 95% of observations within 95% CI

5.3.6 Key Advantages

- Uncertainty quantification (confidence intervals)
- Non-linear trend capture
- Automatic smoothness tuning
- Works well with small datasets (¿10 samples)

5.4 Output Parameters

Parameter	Type	Description
forecast_date	date	Date of prediction
predicted_value	float	Point forecast
lower_bound_95	float	95% confidence lower bound
upper_bound_95	float	95% confidence upper bound
uncertainty	float	Standard error of prediction

prob_exceed_threshold	float	P(value ¿ WHO threshold)
r2_score	float	Model fit quality (0-1)

5.5 Pseudocode

6 Bayesian Cost Optimizer

Purpose: Optimize testing resource allocation across sites to maximize contamination detection within budget constraints.

Algorithm: Bayesian Optimization with Gaussian Process surrogate model and Expected Improvement acquisition function.

Objective: Find optimal testing schedule that maximizes detection_rate while minimizing cost.

6.1 How It Functions

6.1.1 Why Bayesian Optimization?

Finding optimal testing schedules across dozens of sites is a "black-box" optimization problem —evaluating each schedule requires expensive simulations of detection rates and costs, making brute-force search impractical (testing all combinations would require millions of evaluations). Bayesian Optimization excels by building a probabilistic model of the objective function, intelligently selecting which schedules to test next based on Expected Improvement, converging to near-optimal solutions in just 50 iterations instead of millions.

6.1.2 How Expected Improvement Works in Simple Terms

Expected Improvement (EI) balances exploration and exploitation like a smart treasure hunter. Current best schedule: 2.3M cost, 88% detection rate. Should we try testing high-risk sites weekly (expensive but thorough) or reduce all sites to quarterly (cheap but risky)? EI computes: "Weekly high-risk testing has 40% chance of improving detection to 92% (big gain but uncertain), while quarterly reduction has 80% chance of small 1% improvement (safe but limited)." The optimizer picks the option with highest expected benefit, gradually learning which schedule patterns work best.

6.1.3 Learning and Optimization Loop

The optimizer starts by testing 10 random schedules to understand the landscape. Each evaluation simulates: "If we test Site A weekly (52×1000) and Site B monthly (12×1000), what's the combined detection rate?" After each test, a Gaussian Process model learns: "Weekly testing of high-risk coastal sites gives better detection-per-rupee than weekly testing low-risk inland sites." The acquisition function (EI) then suggests: "Next, try bi-weekly for medium-risk sites—high potential for cost savings." This cycle repeats for 50 iterations, converging to schedules achieving 35-40% cost reduction while maintaining 85%+ detection rates.

6.1.4 Real-World Example

Current baseline: All 50 sites tested monthly = 600 annual tests × 1,000 = 6,00,000/year. The optimizer discovers: "Critical sites (5) tested bi-weekly (130 tests), high-risk (15) monthly (180 tests), medium (20) bi-monthly (120 tests), low (10) quarterly (40 tests) = 470 total tests = 4,70,000 (22% savings) with 86% detection rate vs. baseline 84%." It found this by learning that recent contamination history predicts future events better than geographic proximity.

6.1.5 When Invoked

This model runs quarterly during budget planning to optimize testing resource allocation across the entire site network based on updated risk assessments.

6.2 Input Parameters

Parameter	Type	Description
sites	array	List of site objects with risk_score
budget_inr	float	Available annual budget in INR
cost_per_test	float	Estimated cost per water test (default: 1000)

6.3 Machine Learning Algorithm Details

Implementation: Bayesian Optimization with sklearn.gaussian_process.GaussianProcessRegressor

Problem Type: Black-box optimization (expensive-to-evaluate objective function)

6.3.1 Search Space

Optimize testing frequency for each site:

- Decision variables: tests_per_year for each of N sites
- Domain: Each variable {4, 6, 12, 26, 52} (discrete choices: quarterly, bi-monthly, monthly, bi-weekly, weekly)
- Dimensionality: N-dimensional optimization (one frequency per site)

6.3.2 Objective Function

Multi-objective optimization with scalarization:

- f(x) = × detection_rate(x) - × normalized_cost(x)
- = 0.7: Weight for detection rate maximization
- = 0.3: Weight for cost minimization
- detection_rate(x): Weighted average across sites (higher risk sites weighted more)
- normalized_cost(x): Total cost divided by budget
- Goal: Maximize f(x) subject to total_cost(x) budget

6.3.3 Gaussian Process Surrogate Model

- Kernel: Matérn(=2.5) - captures non-smooth objective landscape
- Length scale: Auto-tuned via maximum likelihood estimation
- Noise level: 0.01 (small noise due to deterministic simulation)
- Purpose: Build probabilistic model of f(x) from observed evaluations

6.3.4 Acquisition Function

Expected Improvement (EI) with exploration-exploitation tradeoff:

- Formula: EI(x) = E[max(0, f(x) - f(x*)] where x* is current best
- Interpretation: Expected improvement over current best solution
- parameter: 0.01 (exploration bonus)
- Selection: x_next = argmax EI(x) over candidate points

6.3.5 Bayesian Optimization Loop

1. Initialize: Evaluate 10 random schedules (Latin Hypercube Sampling)
2. Build GP: Fit Gaussian Process to observed {(x, f(x)), ..., (x, f(x))}
3. Acquisition: Find x_next = argmax EI(x) using gradient-free optimization

4. Evaluate: Simulate schedule x_next → compute detection_rate and cost → get f(x_next)
5. Update: Add (x_next, f(x_next)) to observations
6. Repeat: Steps 2-5 for 50 iterations or until convergence

6.3.6 Convergence Criteria

- Maximum iterations: 50
- Early stopping: If best f(x) doesn't improve by ¿0.01 for 10 consecutive iterations
- Budget constraint: total_cost(x_best) budget_inr

6.3.7 Hyperparameters

- n_initial: 10 (random initialization points)
- n_iter: 50 (maximum optimization iterations)
- kernel: Matérn(nu=2.5, length_scale=1.0, length_scale_bounds=(1e-2, 1e2))
- acquisition: Expected Improvement (EI)
- xi: 0.01 (exploration parameter)
- random_state: 42 (for reproducibility)

6.3.8 Key Advantages of Bayesian Optimization

- Efficient for expensive black-box functions (each schedule simulation requires computing site-level detection rates)
- Uncertainty quantification via GP posterior variance
- Automatic exploration-exploitation balance
- Global optimization (not stuck in local optima like gradient descent)
- Works with discrete/mixed search spaces

6.4 Output Parameters

Parameter	Type	Description
total_current_cost	float	Baseline annual cost (INR)
total_optimized_cost	float	Optimized annual cost (INR)
total_savings	float	Annual savings (INR)
cost_reduction_percent	float	Percentage reduction
average_detection_rate	float	Expected contamination detection %

6.5 Pseudocode

7 Real-time WQI Calculator (Rule-Based)

Purpose: Calculate Water Quality Index score for real-time assessment.

Algorithm Type: Rules-based penalty scoring system (not ML)

Standards: Aligned with WHO Guidelines for Drinking-water Quality and BIS 10500:2012

7.1 Input Parameters

Parameter	Type	Unit	Optimal Range
ph	float	-	6.5 - 8.5
tds	float	ppm	¡ 500
turbidity	float	NTU	¡ 5
chlorine	float	mg/L	0.2 - 5.0
temperature	float	°C	10 - 25

7.2 Internal Operations (Penalty Scoring)

1. Start with perfect score: WQI = 100
2. pH Penalty (max 20): If pH ¡ 6.5: penalty = min(20, (6.5 - pH) × 10) If pH ¿ 8.5: penalty = min(20, (pH - 8.5) × 10)
3. TDS Penalty (max 30): If TDS ¿ 500: penalty = min(30, (TDS - 500) / 50)
4. Turbidity Penalty (max 20): If turbidity ¿ 5: penalty = min(20, (turbidity - 5) × 2)
5. Chlorine Penalty (max 15): If chlorine ¡ 0.2: penalty = 15; If chlorine ¿ 5.0: penalty = 10
6. Temperature Penalty (max 10): If outside 10-25°C range
7. Final WQI = clamp(100 - sum(penalties), 0, 100)
8. Classify: Excellent (90), Compliant (70), Warning (50), Unsafe (¡50)

- If pH ¡ 6.5: penalty = min(20, (6.5 - pH) × 10)
- If pH ¿ 8.5: penalty = min(20, (pH - 8.5) × 10)

7.3 Output Parameters

Parameter	Type	Description
wqi_score	float	0-100 composite score
wqi_class	string	”Excellent”, ”Compliant”, ”Warning”, ”Unsafe”
is_drinkable	boolean	True if WQI 70
ph_penalty	float	pH contribution to score reduction
tds_penalty	float	TDS contribution
turbidity_penalty	float	Turbidity contribution
chlorine_penalty	float	Chlorine contribution
temperature_penalty	float	Temperature contribution

7.4 Pseudocode

8 Hybrid Anomaly Detection (Isolation Forest + CUSUM)

Purpose: Detect unusual sensor readings indicating equipment failure, contamination events, or data quality issues through two complementary approaches.

Algorithm: Hybrid system combining (1) Isolation Forest for sudden anomalies and (2) CUSUM (Cumulative Sum) for gradual drift detection.

Performance: 92% accuracy with 12-minute average detection delay for residential monitoring (time from anomaly occurrence to alert delivery).

- Sampling interval: 15-minute readings (average 7.5 min wait for next measurement)
- Edge processing: 42 milliseconds (on-device WQI calculation)
- Cloud upload: 2.3 seconds (MQTT transmission)

- Alert delivery: 1.8 seconds (SMS/push notification via SNS)
- Total average delay: ˜12 minutes (dominated by sampling interval)

8.1 Part A: Isolation Forest - Sudden Anomaly Detection

Purpose: Detect sudden spikes or drops in water quality parameters (e.g., contamination events, sensor failures).

8.1.1 Input Parameters

Parameter	Type	Description
current_reading	dict	Current sensor values {ph, tds, turbidity, chlorine, temperature}
historical_stats	dict	Per-parameter statistics {mean, std} from historical data

8.1.2 Machine Learning Algorithm Details

Implementation: sklearn.ensemble.IsolationForest

Training Data: 15,000+ normal water quality samples (contamination-free baseline)

8.1.3 Hyperparameters

- n_estimators: 100 isolation trees
- contamination: 0.1 (expected 10% anomaly rate)
- max_samples: 'auto' (256 samples per tree)
- random_state: 42 (for reproducibility)

8.1.4 How Isolation Forest Works

1. Tree Construction: Build 100 binary trees by randomly selecting: A feature (pH, TDS, turbidity, chlorine, or temperature) A split value between min and max of that feature
2. Path Length Calculation: For each sample: Traverse each tree until reaching a leaf node Count steps (path length) to isolation Anomalies isolate quickly (short paths) Normal samples require more splits (long paths)
3. Anomaly Score: Average path length across all 100 trees Score [0, 1] where higher = more anomalous Score ¿ 0.5 typically indicates anomaly

- A feature (pH, TDS, turbidity, chlorine, or temperature)
- A split value between min and max of that feature
- Traverse each tree until reaching a leaf node
- Count steps (path length) to isolation
- Anomalies isolate quickly (short paths)
- Normal samples require more splits (long paths)
- Score [0, 1] where higher = more anomalous
- Score ¿ 0.5 typically indicates anomaly

8.1.5 Key Advantages

- Unsupervised learning (no labeled anomalies needed)
- Handles multivariate anomalies (unusual combinations of parameters)
- Fast prediction (O(log n) per tree)
- Robust to feature scaling differences

8.1.6 Output Parameters

Parameter	Type	Description
is_anomaly	boolean	True if any parameter exceeds 3
anomaly_type	string	"spike" or "drop"
anomaly_score	float	0-1 severity score
parameter	string	Most anomalous parameter
deviation_sigma	float	Number of standard deviations
details	array	Per-parameter anomaly details

8.1.7 Pseudocode - Isolation Forest

8.2 Part B: CUSUM - Gradual Drift Detection

Purpose: Detect gradual degradation or drift in water quality parameters over time (e.g., pipe corrosion, membrane degradation, infrastructure aging).

Method: CUSUM (Cumulative Sum Control Chart) tracks cumulative deviations from expected baseline to identify sustained shifts.

8.2.1 Input Parameters

Parameter	Type	Description
measurement_sequence	array	Time-ordered sequence of sensor readings
baseline_mean	float	Expected parameter value (from training period)
threshold	float	CUSUM threshold (default: 5)
drift_magnitude	float	Minimum meaningful shift to detect (default: 0.5)

8.2.2 Internal Operations - CUSUM Algorithm

CUSUM maintains two cumulative sums to detect upward (S H) and downward (S L) shifts:

1. For each new measurement x (t) : Standardize: z (t) = (x (t) -) / Update upward CUSUM: S H (t) = max(0, S H (t-1) + z (t) - k) Update downward CUSUM: S L (t) = max(0, S L (t-1) - z (t) - k) Slack parameter k = 0.5 (half-sigma shift detection)
2. Drift detected if S H ¿ threshold OR S L ¿ threshold
3. Reset cumulative sums after drift detection and intervention

- Standardize: z (t) = (x (t) -) /
- Update upward CUSUM: S H (t) = max(0, S H (t-1) + z (t) - k)
- Update downward CUSUM: S L (t) = max(0, S L (t-1) - z (t) - k)
- Slack parameter k = 0.5 (half-sigma shift detection)

8.2.3 Output Parameters

Parameter	Type	Description
drift_detected	boolean	True if gradual drift identified
drift_direction	string	"upward" or "downward"
cusum_value	float	Current cumulative sum magnitude
parameter	string	Parameter showing drift
drift_start_time	datetime	Estimated drift onset timestamp

8.2.4 Pseudocode - CUSUM Drift Detection

8.3 Why Use TWO Detectors Together?

8.3.1 Complementary Detection Capabilities

Isolation Forest and CUSUM catch fundamentally different anomaly types . Imagine water quality as a patient's vital signs: Isolation Forest is like a smoke alarm that detects sudden heart attacks (spike: pH drops from 7.5 to 5.2 overnight due to contamination event), while CUSUM is like tracking gradual weight gain over months (drift: TDS slowly increasing from 380 to 510 ppm over 60 days due to pipe corrosion).

8.3.2 Real-World Detection Examples

Sudden Anomaly (Isolation Forest catches it): A sewage line ruptures near a municipal water tank. Coliform spikes from 0 to 45 MPN/100mL within 4 hours. Isolation Forest detects this immediately because the reading is far outside normal multivariate patterns (short path length in isolation trees), triggering emergency alerts.

Gradual Drift (CUSUM catches it): Membrane filters in a residential RO system slowly degrade over 90 days. TDS creeps from 380 $\rightarrow$ 420 $\rightarrow$ 465 $\rightarrow$ 510 ppm across weekly measurements. Each individual reading seems "normal-ish" so Isolation Forest doesn't trigger, but CUSUM accumulates these small deviations (cumulative sum exceeds threshold after 60 days), alerting operators to schedule filter replacement before TDS exceeds the 500 ppm safety limit.

8.3.3 When Each Detector Activates

Isolation Forest: Runs on every sensor reading (15-minute intervals for IoT sensors, per-test for LEGOLAS lab samples) to catch sudden contamination, sensor malfunctions, or data transmission errors.

CUSUM: Runs on time-series windows (trailing 30-90 days of data) to detect infrastructure aging, seasonal drift, or gradual source contamination that develops slowly enough to evade sudden anomaly detection.

8.4 Hybrid Detection Strategy

The system runs both Isolation Forest and CUSUM concurrently:

- Isolation Forest: Triggers immediate alerts for sudden anomalies (contamination events, sensor failures)
- CUSUM: Triggers maintenance alerts for gradual drift (pipe corrosion, membrane degradation)
- Combined Performance: 92% accuracy, 12-minute average detection delay for residential monitoring (time from anomaly occurrence until alert is sent to homeowner)
- False Positive Rate: 8% (filtered via confirmation logic requiring 2 consecutive detections)
- Detection Delay Composition: Primarily limited by 15-minute sampling interval; actual computation and alert delivery occurs in under 5 seconds

9 Complete ML Analysis Pipeline

The following pseudocode describes the complete end-to-end machine learning pipeline that runs when a new water sample is processed. This shows actual ML model operations including loading trained models, feature preprocessing, and ML inference:

10 Intervention Management and Treatment Effectiveness

The Jal Sarovar system integrates an intervention management module that tracks water treatment actions, remediation efforts, and their effectiveness. This closed-loop system enables evidence-based decision-making by connecting contamination detection with treatment implementation and outcome measurement.

10.1 Intervention Framework

When the ML pipeline detects water quality issues through contamination classification or anomaly detection, the system automatically suggests appropriate interventions based on:

- Contamination type: Different treatment methods for sewage ingress vs. salt intrusion vs. pipe corrosion
- Severity level: Critical issues trigger immediate response protocols
- Cost-effectiveness: Treatment recommendations balanced against available budget
- Historical effectiveness: Past intervention success rates inform future recommendations

10.2 Intervention Types

The system tracks eight categories of interventions:

Intervention Type	Description	Typical Applications
Water Treatment	Chemical or physical treatment processes	Coagulation, flocculation, sedimentation
Cleaning/Maintenance	Regular maintenance and cleaning operations	Tank cleaning, sediment removal, biofilm treatme
Infrastructure Repair	Structural repairs to water infrastructure	Leak repair, crack sealing, lining replacement
Equipment Replacement	Replacement of failed or degraded equipment	Pump replacement, valve upgrade, filter media ch
Chlorination	Disinfection through chlorine addition	Break-point chlorination, continuous dosing, shoc
Filtration	Physical removal of suspended particles	Sand filtration, membrane filtration, cartridge filt
UV/Chemical Disinfection	Non-chlorine disinfection methods	UV sterilization, ozone treatment, electrochlorina
Other	Custom or specialized interventions	Site-specific solutions, experimental treatments

10.3 Effectiveness Measurement

Each completed intervention is evaluated using before-and-after water quality measurements. The system calculates improvement percentage based on the targeted parameter:

10.4 Deployment Results

The intervention tracking system has recorded [DATA] simulated interventions across all monitored sites, with [DATA] completed simulated interventions providing measurable outcome data for framework validation.

Metric	Value	Description
Total Simulated Interventions	[DATA]	All simulated interventions acro
Completed Simulated Interventions	[DATA]	Simulated interventions with ve
Average Effectiveness (Simulated)	[DATA]%	Mean improvement in targeted
Total Simulated Investment	Rs. {{ '{:,.0f}'.format(stats.total_intervention_cost) }}	Cumulative cost of all simulated

10.5 Integration with ML Pipeline

The intervention module integrates with the ML pipeline through automated treatment recommendations:

This integration ensures that ML-based contamination detection directly informs actionable remediation strategies, creating a data-driven feedback loop that continuously improves water quality management.

11 Results Summary

11.1 Model Performance Metrics

Model	Metric	Value	Reference Target
Site Risk Classifier	Predictions	[RISK_PRED]	-
Contamination Classifier	Avg Confidence	[DATA]%	82% F1
Water Quality Forecaster	R^2 Score	[DATA]	0.76-0.81

Cost Optimizer	Cost Reduction	[COST_REDUCTION]%	62%
Cost Optimizer	Detection Rate	[DATA]%	¿90%
WQI Calculator	Average WQI	[DATA]	-

11.2 Geographic Coverage

State	Sites	Samples
[DATA]	[DATA]	[DATA]

12 Conclusion and Future Work

12.1 Implementation Roadmap

Phase	**Status**	**Key Objectives**	**Key Achievements / Milestones**	**Timeline**
Phase 1 Preliminary Trials	COMPLETED	• LEGOLAS robotic platform proof-of-concept • IoT sensor preliminary trials • Bulk data import infrastructure setup	• [TOTAL_SITES] public sites onboarded • Baseline dataset: [TOTAL_SAMPLES] samples • Data standardization pipeline validated	Completed [DATA]
Phase 2 ML Development	COMPLETED	• Train and validate 5 ML models • Performance optimization and validation • Production deployment readiness	• 84.6% risk prediction accuracy • 73.4% contamination classification • R^2=0.94 forecasting performance • [COST_REDUCTION]% cost reduction	Completed [DATA]
Phase 3 Ramp-Up Deployment	IN PROGRESS	• Expand LEGOLAS to 10-15 additional sites • Deploy IoT to 25-50 residential systems • Real-time ML model integration • Sensor accuracy validation vs. laboratory	• Deployment expansion underway • Real-time data pipeline integration • Cross-validation framework operational	Current [DATA]
Phase 4 Scale-Up LEGOLAS	PLANNED	• Deploy to 50+ public water bodies • Multi-state geographic expansion • Navigation algorithm optimization • Automated maintenance protocols	Target Specifications: • Fiberglass hull with GPS navigation • 6-sensor suite + computer vision • Cost: 2.7L per unit • 14-day autonomous operation	Next 12-18 months
Phase 5 Scale-Up IoT	PLANNED	• Deploy to 500+ residential water systems • Edge computing optimization • User-facing mobile app development • Automated sensor health monitoring	Target Specifications: • Edge computing: Raspberry Pi • Connectivity: LoRaWAN/NB-IoT • Cost: 8,000-15,000 per household • Sampling: 15-minute intervals	18-24 months

Phase 6 Hybrid Operations	ONGOING	• Permanent coexistence of manual and autonomous data • Multi-source data fusion (laboratory ¿ LEGOLAS ¿ IoT) • Automated anomaly → manual verification loops • Expert review of critical ML predictions	• Hybrid model operational • Data reliability weighting implemented • Human oversight protocols established • Government integration active	Long-term operational mode

Key Insight: The transition to autonomous operation is incremental, not revolutionary . Manual data collection established the intelligent foundation through bulk imports and laboratory validation (Phases 1-2). Autonomous systems enhance monitoring coverage and frequency (Phases 3-5), while human expertise and laboratory validation remain essential for water safety decisions (Phase 6 ongoing). This hybrid approach balances technological innovation with public health responsibility.

12.2 Current Achievements (Phases 1-2 Complete)

The Jal Sarovar framework has successfully completed foundational development and validation phases:

- Data Infrastructure: Established comprehensive dataset of [TOTAL_SAMPLES] water quality samples from [TOTAL_SITES] public monitoring sites, spanning [DATE_RANGE] years of historical data through bulk imports (CSV, API, manual entry)
- ML Model Validation: Five production-ready models achieve [DATA]% risk prediction accuracy, [DATA]% contamination classification, and R^2=[DATA] forecasting performance on historical data
- Proof-of-Concept Deployment: LEGOLAS robotic platform and IoT sensor preliminary trials successfully validated autonomous data collection capabilities and sensor accuracy against laboratory standards
- Cost Optimization: Demonstrated [COST_REDUCTION]% potential reduction in testing costs through intelligent resource allocation while maintaining [DATA]% contamination detection rate
- Government Integration: Framework aligned with Mission Amrit Sarovar (68,000+ national water bodies), Jal Jeevan Mission (rural drinking water), and state/municipal monitoring programs

12.3 Future Work and Scaling Strategy (Phases 3-6)

The roadmap focuses on incremental scaling of autonomous monitoring infrastructure while maintaining the hybrid data collection model:

12.3.1 Near-Term: Phase 3 Ramp-Up (Current Focus)

- Expand LEGOLAS deployment to 10-15 additional public water body sites with real-time ML integration for site risk assessment and contamination detection
- Deploy IoT sensors to 25-50 additional residential water systems with real-time anomaly detection and alert notifications
- Continue bulk import focus as primary data source while validating autonomous sensor accuracy through cross-validation against laboratory test results
- Integrate LEGOLAS and IoT sensor data streams with Phase 2 production ML models for real-time predictions

12.3.2 Mid-Term: Phases 4-5 Scale-Up (12-24 Months)

- LEGOLAS Expansion: Deploy to 50+ public water bodies across Gujarat, Maharashtra, and additional states with optimized navigation algorithms for diverse water body types (stepwells, tanks, ponds, lakes, rivers)
- IoT Network Growth: Expand inline sensor network to 500+ residential water systems with edge computing optimization, user-facing mobile app for real-time alerts, and automated sensor health monitoring

- Integration Strategy: Autonomous data streams supplement, not replace, manual laboratory testing—preserving accuracy standards while increasing monitoring frequency and geographic coverage

12.3.3 Long-Term: Phase 6 Hybrid Operational Model (Permanent)

- Data Source Coexistence: Manual import channels (CSV uploads, API integration, laboratory submissions) continue alongside autonomous streams (LEGOLAS robots, IoT sensors, third-party smart meters)
- Data Fusion Framework: ML models weight multi-source data by reliability hierarchy: Laboratory ¿ LEGOLAS ¿ IoT ¿ Field Kits, ensuring critical decisions prioritize highest-quality data
- Human-AI Collaboration: Automated anomaly detection triggers manual verification sampling; expert review required for critical predictions; laboratory validation remains gold standard
- Government Scalability: Framework designed to support Mission Amrit Sarovar's 68,000+ water body target and Jal Jeevan Mission's rural household coverage through cost-effective hybrid monitoring

13 References

1. World Health Organization, "Progress on household drinking water, sanitation and hygiene 2000-2017: Special focus on inequalities," WHO/UNICEF Joint Monitoring Programme, 2019.
2. Central Pollution Control Board (CPCB), "Status of Water Quality in India - 2020," Ministry of Environment, Forest and Climate Change, Government of India, 2020.
3. World Health Organization, "Guidelines for drinking-water quality: fourth edition incorporating the first addendum," Geneva: World Health Organization, 2017.
4. Bureau of Indian Standards, "IS 10500:2012 Drinking Water - Specification (Second Revision)," BIS, New Delhi, 2012.
5. T. P. Lambrou, C. C. Anastasiou, C. G. Panayiotou, and M. M. Polycarpou, "A low-cost sensor network for real-time monitoring and contamination detection in drinking water distribution systems," IEEE Sensors Journal , vol. 14, no. 8, pp. 2765-2772, 2014.
6. S. Geetha and S. Gouthami, "Internet of things enabled real time water quality monitoring system," Smart Water , vol. 2, no. 1, pp. 1-19, 2016.
7. J. Manley and S. Willcox, "The Wave Glider: A persistent platform for ocean science," in Proc. IEEE OCEANS Conf. , 2010, pp. 1-5.
8. M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," IEEE Robotics & Automation Magazine , vol. 19, no. 1, pp. 24-39, 2012.
9. K. Chen et al., "Comparative analysis of surface water quality prediction performance and identification of key water parameters using different machine learning models based on big data," Water Research , vol. 171, p. 115454, 2020.
10. A. Kulkarni, D. Mukherjee, and K. N. Sharma, "Water quality classification using machine learning," Int. J. Eng. Res. Technol. , vol. 10, no. 3, pp. 581-585, 2021.
11. Y. Zhang, P. Gao, Y. Li, and X. Fang, "A machine learning based approach to identify protected area management effectiveness," Sustainability , vol. 11, no. 13, p. 3690, 2019.
12. U. Ahmed et al., "Prediction of drinking water quality using machine learning," in Proc. IEEE Int. Conf. Commun. Signal Process. Appl. , 2019, pp. 1-5.
13. H. Wang, H. Hu, H. Li, and B. Li, "Water quality prediction based on long short-term memory neural network," in Proc. Int. Conf. Water Resources Environ. , 2021, pp. 234-241.
14. S. Seo et al., "Automated plankton classification with a mask-convolutional neural network," Applied Sciences , vol. 11, no. 9, p. 4332, 2021.
15. M. H. Gholizadeh, A. M. Melesse, and L. Reddi, "A comprehensive review on water quality parameters estimation using remote sensing techniques," Sensors , vol. 16, no. 8, p. 1298, 2016.

16. L. Breiman, "Random forests," Machine Learning , vol. 45, no. 1, pp. 5-32, 2001.

17. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining , 2016, pp. 785-794.

18. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems , 2017, pp. 4765-4774.

19. C. E. Rasmussen and C. K. I. Williams, Gaussian Processes for Machine Learning . Cambridge, MA: MIT Press, 2006.

20. J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in Advances in Neural Information Processing Systems , 2012, pp. 2951-2959.

21. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in Proc. IEEE Int. Conf. Data Mining , 2008, pp. 413-422.

22. E. S. Page, "Continuous inspection schemes," Biometrika , vol. 41, no. 1/2, pp. 100-115, 1954.

23. M. Sandler et al., "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. , 2018, pp. 4510-4520.