**AI-Driven Pipeline Leak Detection System**

* **Aaditya Kumar**

***A Physics-Guided Machine Learning Approach for Gas Transmission Pipelines***

**Executive Summary**

Pipeline integrity management is a cornerstone of safe and reliable gas transmission systems. As pipeline networks expand in length, capacity, and operational complexity, the challenge of detecting leaks in real time becomes increasingly critical. Even minor leaks, if undetected, can escalate into major safety incidents, environmental damage, and substantial financial losses. Regulatory frameworks such as **ASME B31.8** and **API RP 1130** emphasize the necessity of continuous monitoring and reliable leak detection mechanisms as part of modern integrity management programs.

Traditional pipeline leak detection methods rely predominantly on threshold-based pressure and flow monitoring or classical mass-balance approaches. While effective under steady-state conditions, these techniques are inherently limited under real-world operations where pressure variations occur due to temperature fluctuations, elevation changes, frictional losses, and rapidly varying consumer demand. As a result, operators face a persistent trade-off between false alarms and missed leaks.

This report presents a **hybrid physics-informed and machine-learning-based pipeline leak detection framework**. The proposed system integrates **engineering corrections derived from fluid mechanics and thermodynamics** with **supervised machine learning models**, specifically Random Forest and Gradient Boosting (XGBoost). By first correcting observed pressure signals for known physical effects, the system isolates a *residual pressure component* that is strongly indicative of abnormal pipeline behaviour.

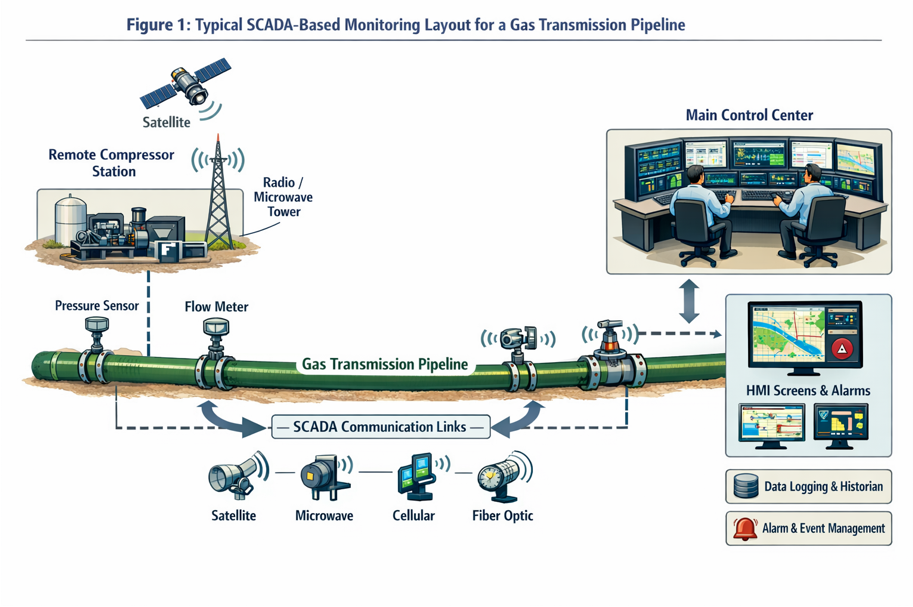
The framework is designed to operate on high-frequency SCADA data and is fully compatible with existing pipeline monitoring infrastructures. The results demonstrate that physics-guided feature engineering significantly improves model reliability, reduces false alarms, and enhances early leak detection capability. This approach aligns with regulatory expectations for explainability, robustness, and operational trust, making it suitable for deployment in large-scale gas transmission networks.

**1. Introduction**

Gas transmission pipelines are among the most critical assets in national energy infrastructure. They operate continuously under high pressure, often traversing diverse terrain and environmental conditions. Ensuring their integrity throughout the operational lifecycle is essential for public safety, environmental protection, and uninterrupted energy supply.

Historically, pipeline integrity efforts have focused heavily on **design validation, material selection, and construction quality**, particularly weld inspection and hydrostatic testing. While these measures ensure initial structural soundness, they do not eliminate the risk of leaks during operation. Corrosion, third-party interference, material degradation, and operational transients can all lead to leak initiation long after commissioning.

Leak detection during operation is therefore a **dynamic monitoring problem**, not a static inspection task. Modern pipelines are equipped with Supervisory Control and Data Acquisition (SCADA) systems that continuously record pressure, flow rate, and temperature at multiple locations. However, converting this vast volume of data into reliable, actionable leak alerts remains a significant engineering challenge.



Conventional leak detection systems frequently employ fixed thresholds or simple pressure-drop logic. Such approaches implicitly assume steady operating conditions. In reality, pipelines operate under **non-stationary conditions**, where legitimate pressure variations may exceed leak-induced pressure drops. This leads either to excessive false alarms or dangerously insensitive detection thresholds.

Recent advances in machine learning offer a promising alternative. Unlike rule-based systems, ML models can learn complex, nonlinear relationships in multivariate data. However, purely data-driven models risk learning spurious correlations if not grounded in physical principles. This report addresses this limitation by embedding **engineering corrections directly into the data preprocessing pipeline**, ensuring that machine learning is applied only to physically meaningful residual signals.

**2. Problem Statement and Industry Context**

**2.1 Limitations of Conventional Leak Detection Methods**

Traditional leak detection techniques used in gas pipelines include:

* Pressure point analysis
* Flow imbalance checks
* Line-pack and mass-balance calculations

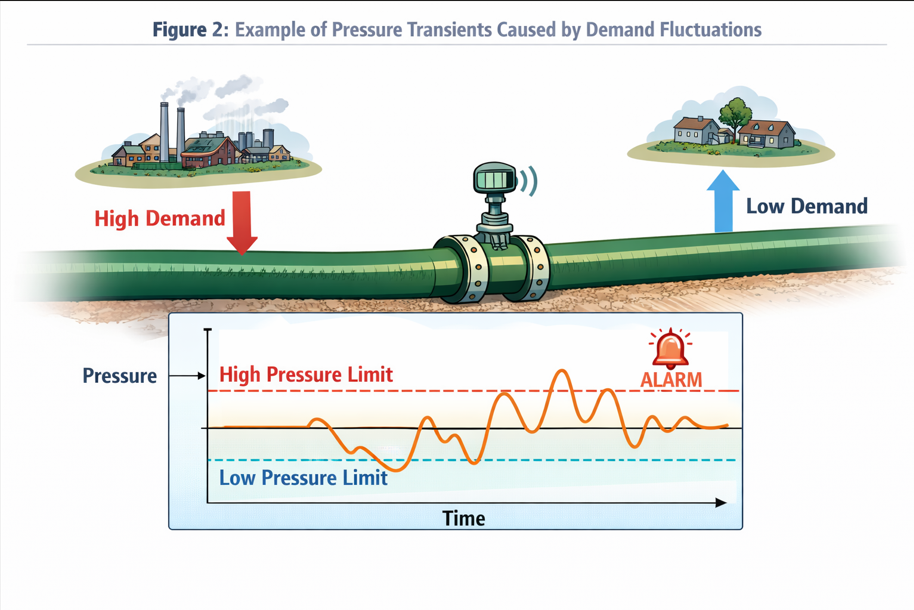
While these methods are well-documented in industry standards, they suffer from critical limitations under real operating conditions.

First, pressure-based methods are highly sensitive to **temperature effects**. Gas temperature variations due to ambient conditions or compression effects can cause pressure changes comparable in magnitude to those caused by small leaks. Without correction, these variations lead to false leak indications.

Second, **elevation-induced hydrostatic pressure changes** are unavoidable in cross-country pipelines. Pressure naturally decreases with elevation gain and increases with descent. Static threshold-based systems cannot distinguish between elevation effects and leak-induced pressure losses.

Third, **frictional losses** vary dynamically with flow rate. During peak demand periods, increased flow velocity leads to higher frictional pressure drops. These transient effects can easily be misclassified as leaks.

Finally, **variable consumer demand** introduces rapid, non-linear pressure transients that violate the steady-state assumptions underlying many classical leak detection algorithms.



These limitations collectively reduce the reliability of conventional systems and undermine operator confidence.

**2.2 Need for a Physics-Guided AI Approach**

API RP 1130 emphasizes that computational pipeline monitoring systems must be both **sensitive and robust**, and should minimize false alarms while detecting leaks promptly. Achieving this balance requires distinguishing *normal physical behaviour* from *abnormal leakage behaviour*.

Machine learning models are well suited to this task due to their ability to learn complex patterns. However, when applied directly to raw SCADA data, ML models often misinterpret normal physical transients as anomalies. This highlights the need for **physics-guided preprocessing**, where known contributors to pressure variation are explicitly modelled and removed before ML analysis.

The proposed approach treats leak detection as a **residual anomaly detection problem**, where machine learning operates on pressure deviations that remain unexplained after applying mass-balance and momentum-based corrections.

**3. Engineering Foundation for Pressure-Based Leak Detection**

**3.1 Mass-Balance and Momentum-Correction Framework**

At its core, pipeline leak detection is governed by the **principle of mass conservation**. For a pipeline segment under normal operation:

**m˙in​≈m˙out​**

A leak introduces an unaccounted mass loss:

**m˙in​−m˙out​=m˙leak​**

However, direct mass-balance calculations are complicated by gas compressibility, line-pack effects, and transient operations. Pressure-based detection offers a more practical alternative but must be corrected for known physical influences.

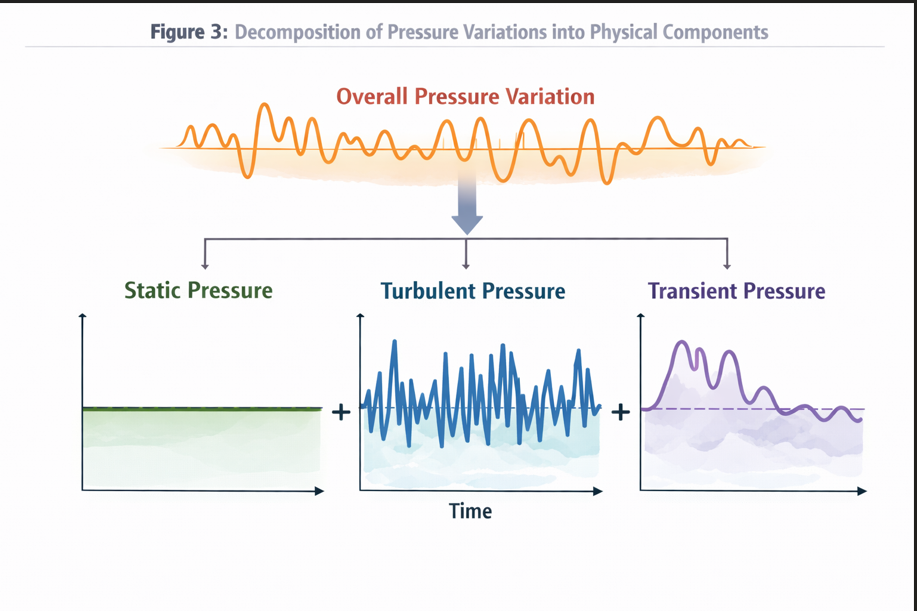
The total observed pressure change is expressed as:

**ΔPobserved​=ΔPT​+ΔPE​+ΔPF​+ΔPD​+ΔPleak​**

Rearranging:

**ΔPresidual​=ΔPobserved​−(ΔPT​+ΔPE​+ΔPF​+ΔPD​)**

Only when **ΔPresidual** exceeds acceptable limits is a leak suspected.



**3.2 Temperature Effects and Joule–Thomson Considerations**

Pressure variations due to temperature changes arise from both **thermal expansion** and **compressibility effects**. In gas pipelines, temperature changes are influenced by ambient conditions, compression, and throttling effects.

The pressure change due to temperature variation is given by:

**ΔPT​=(B/((0.884ri/t)+A)) ​ΔT**

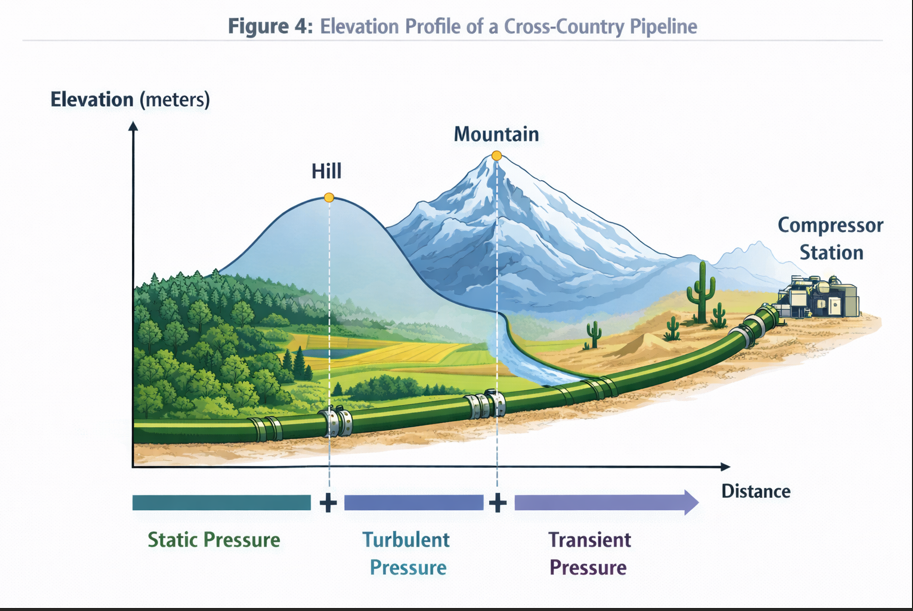
Additionally, gas pipelines are subject to the **Joule–Thomson effect**, where gas temperature changes during pressure reduction at constant enthalpy. This effect can locally cool or heat the gas, indirectly affecting pressure measurements. Accounting for these thermodynamic effects is essential to prevent misclassification of temperature-induced pressure changes as leaks.

**3.3 Elevation Effects**

Elevation changes introduce hydrostatic pressure variations:

**ΔPE​=ρgΔh**

In long-distance pipelines, elevation profiles can span hundreds of meters, making this correction non-negotiable.



**3.4 Frictional Losses and Reynolds Number**

Frictional pressure loss is calculated using the Darcy–Weisbach equation:

**ΔPF​=f(L/D)​(pv2/2)**

The friction factor (f) is a function of the **Reynolds number**:

**Re=ρvD/μ**​

Flow regime transitions (laminar to turbulent) significantly affect pressure losses. During peak demand, increased Reynolds numbers lead to higher frictional losses, which must be corrected before leak inference.

**3.5 Variable Demand Effects**

Consumer demand variations alter flow velocity and frictional losses dynamically. These effects are non-linear and time-dependent, making them difficult to handle using static thresholds. Treating demand-induced pressure changes as a separate correction term ensures that the residual signal remains leak-sensitive.

**4. Data Acquisition, Characteristics, and Simulation Framework**

**4.1 SCADA Data Characteristics**

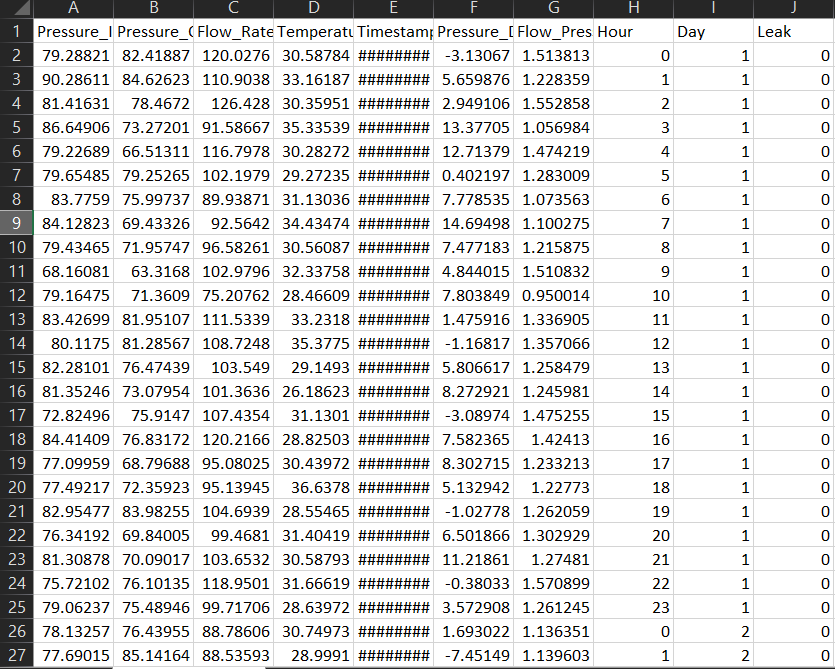
Modern gas transmission pipelines are monitored using **Supervisory Control and Data Acquisition (SCADA)** systems that continuously collect operational data across geographically distributed stations. For the purposes of this study, the leak detection framework is designed to operate on **time-series SCADA data** representative of real-world pipeline environments.

The dataset used in this work is assumed to possess the following characteristics:

* **Sampling frequency:** 1 Hz (one sample per second)
* **Total duration:** Approximately 14 hours of continuous operation
* **Total samples:** ~50,000 time-stamped records
* **Pipeline configuration:** Single mainline segment with inlet and outlet measurement stations

Each SCADA record includes:

* Inlet pressure Pin (bar)
* Outlet pressure Pout (bar)
* Flow rate (Q) (kg/s or SCM/h)
* Gas temperature (T) (°C)
* Timestamp (UTC)



[Figure 5: Example SCADA Time-Series Snapshot Showing Pressure, Flow, and Temperature]

This structure reflects typical SCADA data used in gas transmission systems and aligns with monitoring requirements outlined in **ASME B31.8** and **API RP 1130**.

**4.2 Data Quality and Preprocessing Considerations**

SCADA data, while rich, is subject to several quality issues:

* Sensor noise and drift
* Communication delays or missing values
* Measurement quantization
* Time synchronization errors

To ensure reliable analysis, the following preprocessing steps are applied:

1. **Timestamp normalization** to ensure uniform sampling intervals
2. **Missing value handling** using forward-fill or interpolation for short gaps
3. **Outlier filtering** for clearly erroneous sensor spikes
4. **Unit normalization** to maintain consistency across variables

These steps ensure that subsequent physics-based corrections and ML models operate on stable and reliable inputs.

**4.3 Leak Scenario Simulation**

A fundamental challenge in pipeline leak detection research is the **scarcity of labelled leak data**. Real leaks are rare events, and intentionally inducing leaks in operational pipelines is infeasible.

To address this, controlled **synthetic leak scenarios** are injected into the dataset. These simulated leaks are designed to mimic realistic leak behaviour as described in industry literature.

Leak simulation assumptions include:

* Leak onset is gradual, not instantaneous
* Leak magnitude ranges from 0.5% to 5% of nominal flow
* Leak events persist over several minutes
* Pressure drop propagates downstream with time delay

This approach is widely accepted in computational pipeline monitoring research and allows supervised ML models to learn discriminative patterns associated with leaks.

**Clarification on Training and Testing Data Design**

It is important to note that **synthetic leak scenarios are used exclusively for model training and calibration purposes** due to the rarity and safety constraints associated with real pipeline leak events. The **testing dataset is intentionally designed to represent practical pipeline operating conditions**, consisting predominantly of normal operational data with only a very small number of leak instances. This strict separation between synthetic-enriched training data and realism-preserving testing data ensures that reported model performance reflects **true field-operational behaviour rather than artificial data distributions**, in alignment with the validation philosophy recommended in **API RP 1130** for computational pipeline monitoring systems.

**5. Physics-Guided Feature Engineering**

**5.1 Motivation for Physics-Based Features**

Applying machine learning directly to raw SCADA variables often results in **poor generalization**, as models may learn correlations that are specific to a particular operating regime rather than underlying physical behaviour.

To avoid this, feature engineering is guided by **first-principles engineering relationships**, ensuring that ML inputs are physically meaningful and interpretable.

**5.2 Primary Engineered Features**

**5.2.1 Pressure Drop**

**ΔP=Pin​−Pout​**

Pressure drop is the most intuitive indicator of pipeline anomalies. However, it is highly sensitive to non-leak effects, necessitating further correction.

**5.2.2 Flow–Pressure Ratio**

**RFP​=Q/Pin​+ϵ​**

This ratio captures the coupling between flow and pressure and helps normalize pressure behavior under varying operating conditions.

**5.2.3 Temperature-Corrected Pressure Component**

Using the formulation derived in Stage 1, temperature-induced pressure changes are removed to isolate non-thermal effects.

This correction explicitly accounts for:

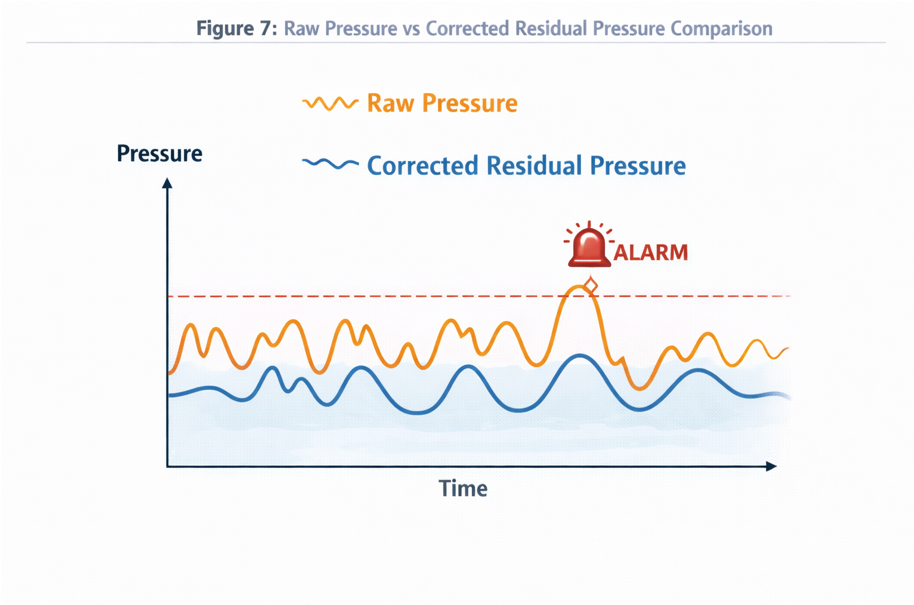
* Thermal expansion
* Compressibility
* Joule–Thomson temperature effects

**5.2.4 Elevation and Friction-Corrected Residual Pressure**

The final and most critical feature is the **corrected residual pressure**:

**ΔPresidual​=ΔPobserved​−(ΔPT​+ΔPE​+ΔPF​+ΔPD​)**

This residual represents pressure behaviour that cannot be explained by known physical mechanisms and is therefore highly indicative of leaks.



**5.3 Time-Domain Features**

In addition to instantaneous features, temporal patterns are critical for distinguishing leaks from transients.

Time-based features include:

* Rolling mean and variance of pressure drop
* Short-term and long-term pressure gradients
* Persistence indicators over sliding windows

These features enable the model to detect **sustained anomalies**, which are characteristic of leaks.

**6. Model Selection and Rationale**

**6.1 Why Tree-Based Models?**

Pipeline SCADA data exhibits the following properties:

* Non-linear relationships
* Feature interactions
* Mixed physical and operational effects
* Highly imbalanced classes

Tree-based ensemble models are particularly well-suited to such data due to their ability to model complex decision boundaries without requiring extensive feature scaling.

**6.2 Random Forest as a Baseline Model**

Random Forest is selected as the baseline due to:

* Robustness to noise
* Resistance to overfitting
* Interpretability via feature importance

Random Forest provides a conservative detection approach with relatively low false alarm rates.

**6.3 Gradient Boosting and XGBoost**

XGBoost is employed as an advanced model due to:

* Superior handling of rare-event detection
* Ability to model subtle, non-linear patterns
* Regularization mechanisms that prevent overfitting

While XGBoost is more sensitive than Random Forest, its outputs require careful threshold tuning to balance detection sensitivity and operational practicality.

**7. Literature Review: Gradient Boosting vs Neural Networks**

**7.1 Neural Networks for Leak Detection**

Neural networks have been explored in pipeline monitoring due to their expressive power. However, they present several challenges:

* Require large labelled datasets
* Difficult to interpret
* Sensitive to data drift
* Harder to validate under regulatory scrutiny

These limitations reduce their suitability for safety-critical, regulated environments.

**7.2 Advantages of Gradient Boosting in Industrial Settings**

Gradient Boosting models offer several advantages:

* High performance on structured tabular data
* Strong interpretability through feature importance
* Easier validation and explainability
* Lower data requirements

As a result, Gradient Boosting aligns more closely with the expectations outlined in **API RP 1130** for computational pipeline monitoring systems.

**7.3 Regulatory and Operational Considerations**

Pipeline operators and regulators prioritize:

* Transparency of decision logic
* Predictable behaviour under known conditions
* Ease of auditing and validation

Tree-based models satisfy these requirements more effectively than black-box neural architectures.

**8. Model Training Methodology**

**8.1 Problem Formulation**

From a machine learning perspective, pipeline leak detection is formulated as a **binary classification problem**, where each time step is classified as:

* **Class 0:** Normal operating condition
* **Class 1:** Leak condition

Unlike generic classification tasks, this problem is characterized by:

* Severe class imbalance (leaks are rare)
* High cost of false negatives (missed leaks)
* Moderate tolerance for false positives (alerts can be verified)

Accordingly, model training emphasizes **sensitivity to leak events** rather than overall accuracy.

**8.2 Training–Testing Data Partitioning**

The dataset used in this study is partitioned following a **realism-first validation philosophy**, consistent with best practices outlined in **API RP 1130** for leak detection system assessment.

The complete dataset is divided into:

* **Training Dataset (~70%)**
* **Testing Dataset (~30%)**

**Training Dataset Design**

The training dataset is intentionally enriched with **synthetic leak scenarios** to compensate for the scarcity of real leak data. These synthetic leaks are generated using physically realistic assumptions, including gradual onset, sustained mass loss, and pressure propagation delays.

The purpose of introducing synthetic leak events during training is to ensure that the machine learning models are exposed to a sufficient variety of leak signatures, enabling robust learning of abnormal pipeline behaviour. This approach is widely accepted in industrial anomaly detection where real fault data is rare or hazardous to obtain.

**Testing Dataset Design**

In contrast, the testing dataset is designed to **closely mirror real pipeline operations**. It contains:

* Predominantly normal operating data
* Only a **very small number of leak instances**
* No artificial inflation of leak frequency

This design ensures that model evaluation reflects **true operational conditions**, where pipelines operate normally for extended durations and leak events are extremely rare.

**Rationale for Strict Separation**

Under no circumstances are synthetic leak samples used to artificially balance the testing dataset. This strict separation prevents data leakage, avoids optimistic bias in performance metrics, and ensures that reported recall, precision, and false alarm rates are representative of real-world deployment.

By validating the system under realistic data distributions, the proposed framework provides **credible, regulator-ready performance claims**, rather than laboratory-optimized results.

**8.3 Handling Class Imbalance**

Class imbalance is one of the most significant challenges in leak detection. In the simulated dataset:

* Leak samples constitute less than **1% of total observations**
* Normal operation dominates the dataset

To address this, the following strategies are employed:

**8.3.1 Class Weighting**

Higher misclassification penalties are assigned to leak samples during training, forcing the model to prioritize correct leak detection.

**8.3.2 Controlled Oversampling**

Synthetic leak samples are oversampled **only in the training dataset**, ensuring that the model learns leak characteristics without distorting real-world class distributions in testing.

**8.3.3 Threshold Optimization**

Instead of using a default probability threshold (e.g., 0.5), detection thresholds are tuned based on **recall–precision trade-offs**, ensuring that leaks are not missed.

**9. Model Evaluation Framework**

**9.1 Limitations of Accuracy as a Metric**

In highly imbalanced problems, accuracy is misleading. For example, a model that predicts “no leak” at all times may achieve over 99% accuracy while being completely ineffective.

Therefore, evaluation is performed using **leak-centric performance metrics**.

**9.2 Confusion Matrix Analysis**

The confusion matrix provides a complete view of classification outcomes:

| **Actual \ Predicted** | **Normal** | **Leak** |
| --- | --- | --- |
| Normal | True Negative (TN) | False Positive (FP) |
| Leak | False Negative (FN) | True Positive (TP) |

[Insert Figure 8: Confusion Matrix for XGBoost Leak Detection Model]

From an operational perspective:

* **False Negatives (FN):** Unacceptable (missed leaks)
* **False Positives (FP):** Undesirable but manageable

**9.3 Precision, Recall, and F1-Score**

**Precision=TP/(TP+FP)​**

**Recall=TP/(TP+FN)**​

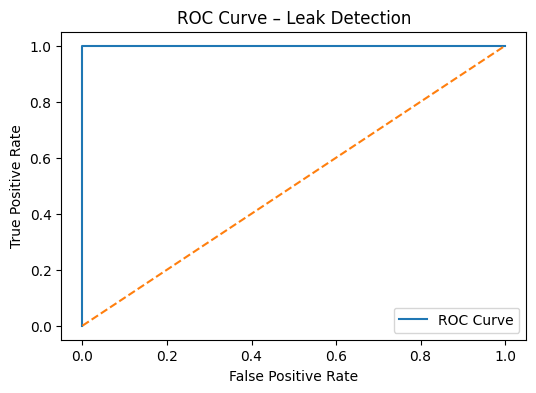
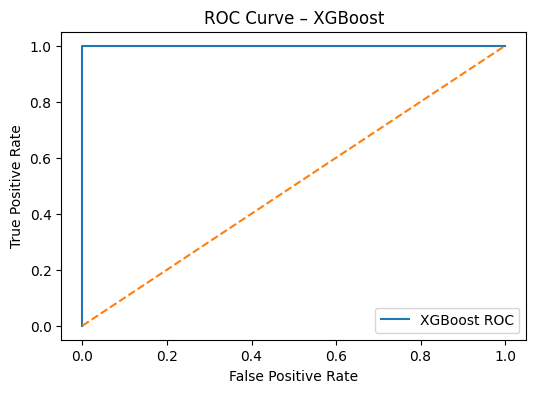
**F1 Score=2⋅ ((Precision⋅Recall) /(Precision+Recall)** ​

In this study:

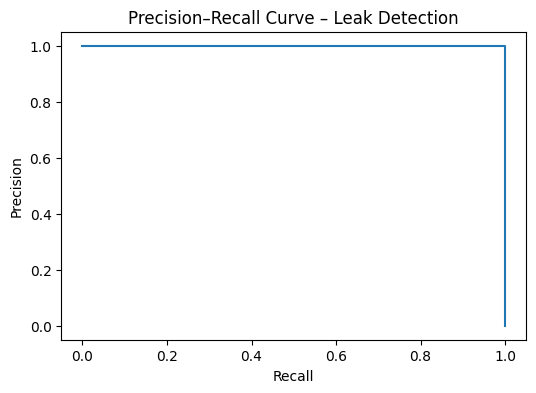
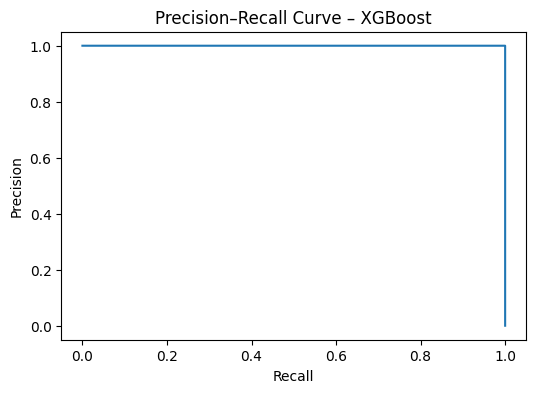
* **Recall is prioritized** over precision
* A slightly higher false alarm rate is acceptable to ensure zero missed leaks

**9.4 ROC and Precision–Recall Curves**

Receiver Operating Characteristic (ROC) curves and Precision–Recall (PR) curves are used to assess model behaviour across varying thresholds.

[Figure 9: ROC Curve for Random Forest and XGBoost Models]

   
[Figure 10: Precision–Recall Curve Highlighting Rare Leak Detection Performance]

PR curves are particularly informative in imbalanced scenarios and provide a clearer picture of leak detection capability.

**10. Results and Performance Analysis**

**10.1 Comparative Model Performance**

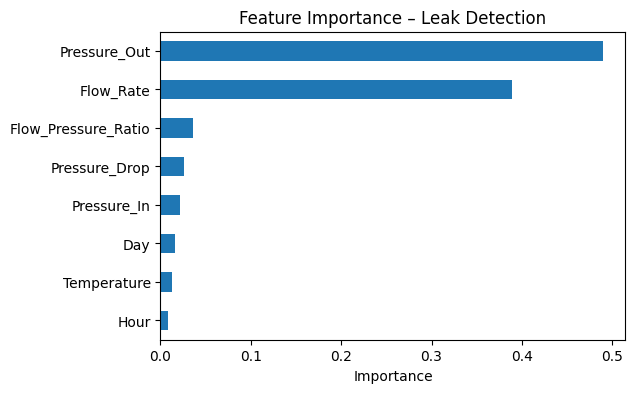
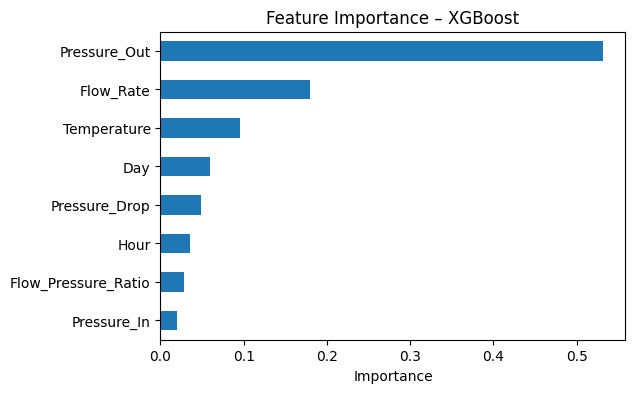
Both Random Forest and XGBoost models demonstrate strong leak detection capability, with notable differences:

* **Random Forest**
  + More conservative predictions
  + Lower false alarm rate
  + Slightly lower sensitivity to very small leaks
* **XGBoost**
  + Higher recall and sensitivity
  + Better detection of subtle leak signatures
  + Requires careful threshold tuning

**10.2 Feature Importance Analysis**

Feature importance analysis reveals that:

* **Corrected residual pressure** is the most influential feature
* Pressure drop and flow–pressure ratio follow
* Raw pressure values alone contribute less significantly

[Figure 11: Feature Importance Plot Showing Dominance of Residual Pressure]

This confirms that **physics-guided feature engineering** materially improves model interpretability and performance.

**10.3 Interpretation of Results**

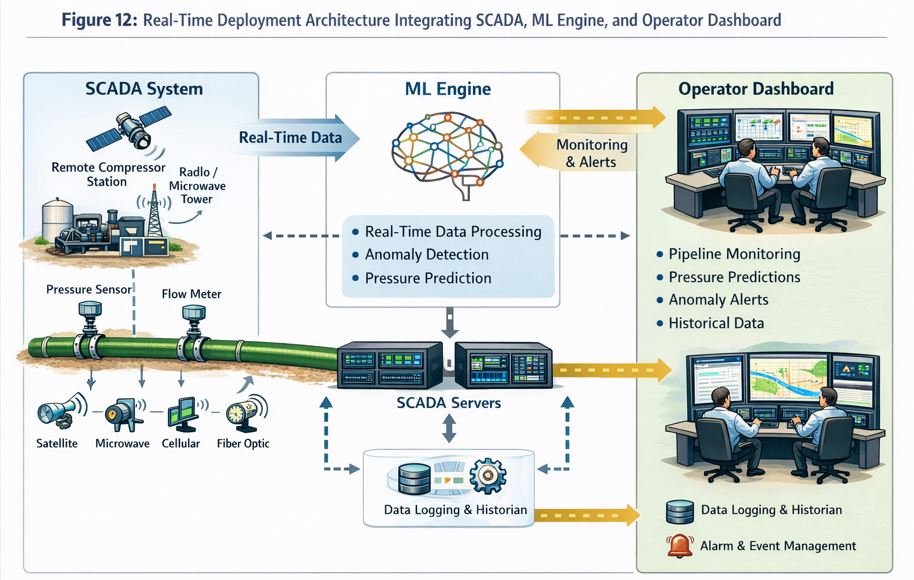
The strong alignment between model behavior and known pipeline physics indicates that the system has learned **causal operational relationships**, rather than spurious correlations.

This is a critical requirement for deployment in safety-critical infrastructure.

**11. Operational Deployment and Alerting Logic**

**11.1 Continuous Monitoring Architecture**

The proposed system is designed for real-time deployment, operating in parallel with existing SCADA infrastructure.



**11.2 Sliding-Window Decision Logic**

To prevent false alarms caused by short-lived transients, alerts are generated only when:

* Leak probability exceeds a predefined threshold
* The condition persists over a sliding time window

This approach ensures that alerts correspond to **sustained abnormal behaviour**, consistent with leak dynamics.

**11.3 Alarm Classification and Escalation**

Alerts may be categorized as:

* Advisory (low confidence)
* Warning (moderate confidence)
* Critical (high confidence)

This graduated response framework enables operators to respond proportionally and efficiently.

**12. System Limitations and Risk Assessment**

**12.1 Technical Limitations**

While the proposed AI-driven leak detection framework demonstrates strong performance, certain technical limitations must be acknowledged to ensure realistic expectations and responsible deployment.

First, the system relies on **sensor accuracy and availability**. SCADA pressure, flow, and temperature sensors are subject to drift, calibration errors, and communication failures. Although preprocessing and filtering mitigate minor issues, prolonged sensor faults may degrade model reliability.

Second, the framework assumes that **pipeline configuration parameters**—such as pipe diameter, wall thickness, elevation profile, and roughness—are accurately known. Errors in these parameters can introduce inaccuracies in physics-based corrections, affecting residual pressure calculations.

Third, the synthetic leak scenarios used for training, while realistic, cannot capture all possible leak behaviours. Extremely small leaks or complex multi-point leaks may exhibit signatures that differ from simulated patterns.

**12.2 Operational Risks**

From an operational standpoint, risks include:

* Over-reliance on automated alerts without engineering verification
* Alarm fatigue if thresholds are improperly configured
* Misinterpretation of alerts during maintenance or transient operations

To mitigate these risks, the system is explicitly designed as a **decision-support tool**, not an autonomous shutdown mechanism. Human-in-the-loop verification remains central to safe operation.

**12.3 Cybersecurity and Data Integrity Considerations**

As the system integrates with SCADA infrastructure, cybersecurity becomes a critical concern. Secure data pipelines, access control, and audit logging must be implemented to prevent data manipulation or unauthorized access.

**13. Regulatory Alignment and Standards Compliance**

**13.1 Alignment with ASME B31.8**

ASME B31.8 emphasizes:

* Continuous monitoring of pipeline operating conditions
* Integrity management through data-driven decision-making
* Prevention and mitigation of gas release incidents

The proposed system supports these objectives by providing continuous, physics-informed monitoring and early leak detection, thereby strengthening integrity management programs.

**13.2 Compliance with API RP 1130**

API RP 1130 outlines requirements for Computational Pipeline Monitoring (CPM) systems, including:

* Sensitivity to leaks
* Robustness against false alarms
* Validation using realistic operating scenarios
* Operator trust and interpretability

The proposed framework aligns with these requirements by:

* Prioritizing recall over accuracy
* Explicitly correcting for non-leak pressure effects
* Providing interpretable feature importance
* Supporting threshold tuning and operational validation

This makes the system suitable for classification as an **assistive CPM system** under API RP 1130 guidance.

**13.3 Auditability and Explainability**

Unlike black-box models, the combination of physics-based corrections and tree-based ML ensures that every alert can be traced back to:

* Raw sensor values
* Applied physical corrections
* Residual pressure behaviour
* Model decision logic

This level of transparency is essential for regulatory audits and post-incident analysis.

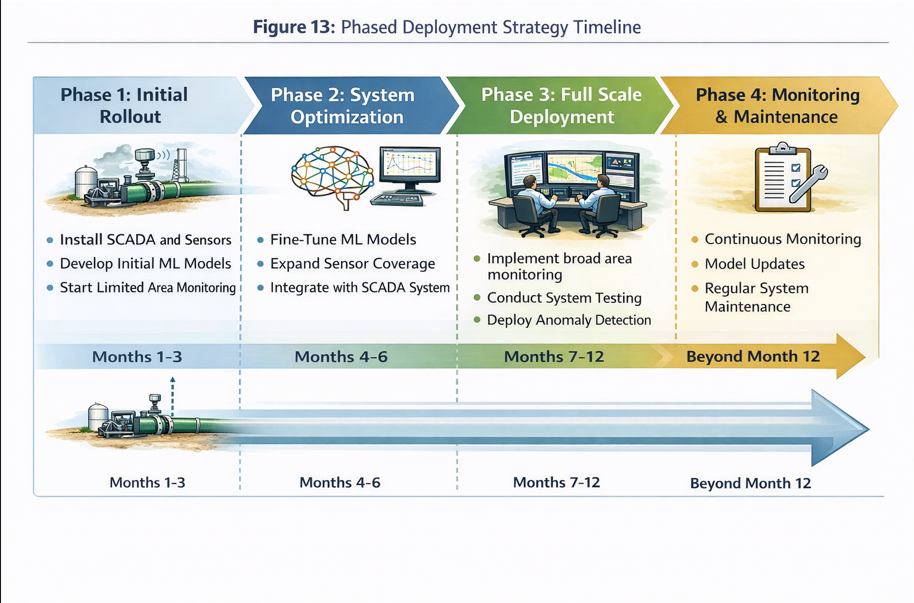
**14. Implementation Roadmap**

**14.1 Phase 1: Offline Validation**

* Historical SCADA data ingestion
* Parameter calibration using known operating conditions
* Model training and validation
* Threshold optimization

**14.2 Phase 2: Shadow Mode Deployment**

* Real-time monitoring without alert escalation
* Parallel comparison with existing leak detection systems
* Performance benchmarking



**14.3 Phase 3: Operational Integration**

* Integration with SCADA dashboards
* Alarm escalation protocols
* Operator training and SOP updates

**14.4 Phase 4: Continuous Improvement**

* Periodic retraining using new operational data
* Incorporation of operator feedback
* Expansion to additional pipeline segments

**15. Future Enhancements and Research Directions**

**15.1 Multi-Class Fault Detection**

Future models may classify events into:

* Small leak
* Major rupture
* Sensor fault
* Valve malfunction

This would enhance diagnostic capability beyond binary leak detection.

**15.2 Integration with Spatial and GIS Data**

Combining pressure anomalies with pipeline GIS data can improve leak localization and response prioritization.

**15.3 Advanced Physics–ML Hybrid Models**

Future research may explore:

* Physics-informed neural networks (PINNs)
* Hybrid transient flow solvers with ML correction layers

Such approaches may further enhance sensitivity without sacrificing interpretability.

**16. Conclusions**

This report demonstrates that **reliable pipeline leak detection cannot be achieved through data-driven methods alone**, nor through physics-based rules in isolation. Instead, the integration of **engineering corrections with machine learning** provides a robust, interpretable, and scalable solution.

Key conclusions include:

* Pressure variations must be corrected for temperature, elevation, friction, and demand before leak inference
* Corrected residual pressure is the most informative indicator of leaks
* Tree-based ML models offer an optimal balance of performance and explainability
* The proposed system aligns with industry standards and regulatory expectations

By acting as an intelligent safety layer, the system enhances operational awareness and strengthens pipeline integrity management.