International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Water Risk Compliance: A Retrieval-Augmented, Explainable, and Interactive System for Drinking Water Compliance Analytics

Pramath Parashar

Data Science Specialist
BHP Minerals Service Company
srivatsapramath@gmail.com
ORCID: 0009-0000-4902-7958

Abstract:
Ensuring compliance with drinking water standards is a critical global challenge, as exceedances in contaminants such as arsenic, nitrate, and uranium pose significant risks to public health and regulatory accountability. Existing monitoring systems are often fragmented, lack explainability, and provide limited support for rapid decision-making. This paper introduces WaterRiskCompliance, an interactive and explainable compliance analytics framework that unifies data ingestion, automated cleaning, retrieval-augmented reasoning, and regulatory visualization. The system leverages open water quality datasets, harmonizes measurements into consistent units (mg/L), and applies retrieval- augmented generation (RAG) with local or cloud-based large language models to generate concise compliance narratives. A dynamic dashboard enables users to filter by analyte, county, or time range, and provides multi- layer visualizations, including exceedance bar charts, spatial maps, and time-series trends aligned with regulatory action levels. A built-in module generates standardized PDF compliance briefs, bridging raw data and actionable insights. Through a case study on Arizona groundwater, the framework demonstrates how modern AI and interactive visualization can accelerate regulatory compliance assessment, improve transparency, and support evidence-driven decision-making in environmental monitoring.

Index Terms: Water quality monitoring; Compliance analytics; Retrieval-augmented generation; Explainable AI; Environmental informatics; Regulatory decision support; Streamlit dashboard; Ollama.

I. INTRODUCTION

Safe drinking water is a fundamental requirement for human health, yet water contamination continues to pose significant risks worldwide. Regulatory agencies, including the United States Environmental Protection Agency (EPA) and the World Health Organization (WHO), have established maximum contaminant levels (MCLs) for toxic analytes such as arsenic, nitrate, and uranium in order to safeguard communities [1], [2]. In Arizona and other arid regions, the challenge is compounded by reliance on groundwater resources, which are often vulnerable to contamination from natural geochemical processes and mining activities [3]. Effective monitoring of such risks requires both accurate data collection and advanced analytic frameworks.

Large public repositories, such as the United States Geological Survey (USGS) National Water Information System, provide access to millions of water quality samples across thousands of monitoring sites [4]. However, traditional approaches to compliance monitoring face three core challenges: (1) integrating heterogeneous data from multiple agencies, (2) detecting exceedances of regulatory thresholds in near real time, and (3) communicating results to regulators, stakeholders, and the public in interpretable formats

[5], [6]. Addressing these gaps requires solutions that are not only data-driven but also interactive and transparent.

Recent advances in machine learning and natural language processing (NLP) offer new opportunities to transform compliance workflows. Retrieval-Augmented Generation (RAG) techniques allow domain-specific knowledge to be combined with the reasoning power of large language models (LLMs), enabling precise, grounded responses to technical queries [7]. Foundational architectures such as the Transformer [8] and large-scale models like GPT-3 [9] have demonstrated their capability to perform domain adaptation, while local inference frameworks such as Ollama allow these models to be deployed securely without reliance on cloud APIs [10]. For water quality applications, the combination of predictive analytics, explainable machine learning [11], and visual dashboards [12] provides a powerful foundation for proactive decision support [13].

This paper builds on the author's prior contributions in embedded IoT automation [14], market optimization using Markov decision processes [15], and visual/statistical inference for compliance analytics [16]. Together, these works establish a trajectory of advancing environmental monitoring through automation, optimization, and explainability.

In this paper, we present a novel *Water Compliance Chat and Analytics Platform* that integrates heterogeneous groundwater quality data, advanced predictive models, and generative AI for interactive compliance analysis. The platform supports CSV ingestion, data cleaning, exceedance detection, and time-series visualization, while providing a natural language interface powered by RAG-enhanced LLMs. Built with Streamlit for usability [17], ChromaDB for retrieval [18], and integrated with both OpenAI and Ollama backends, the system enables end-to-end compliance monitoring tailored to Arizona's mining legacy sites.

The main contributions of this work are as follows:

1) Development of an integrated pipeline for ingesting and normalizing heterogeneous water quality datasets.
2) Implementation of machine learning baselines (Random Forest [19], XGBoost [20], and SMOTE balancing [21]) alongside explainability methods [11].
3) Design of an interactive dashboard with bar, map, and time-series visualizations aligned with regulatory action levels.
4) Integration of Retrieval-Augmented Generation (RAG) for query answering, enabling transparent and citation-grounded responses.
5) Demonstration of deployment flexibility with both cloud-based and local LLM backends.

This platform not only advances the state of compliance analytics but also serves as a replicable framework for other domains of environmental monitoring and regulatory reporting. By bridging data science, AI, and regulatory practice, it addresses an urgent need for explainable, real-time, and scalable compliance solutions.

II. RELATED WORK

Research in water quality monitoring and compliance analytics spans multiple domains, including environmental science, machine learning, and interactive visualization. This section reviews prior work across these areas and highlights the gaps addressed by our proposed system.

A. Regulatory and Environmental Standards

Regulatory frameworks from the EPA and WHO define maximum contaminant levels for toxic analytes such as arsenic, nitrate, and uranium [1], [2]. Regional agencies, including the Arizona Department of Environmental Quality (ADEQ), extend these guidelines to local contexts [3]. While these standards establish clear thresholds, their practical enforcement depends on the ability to process large, heterogeneous datasets in near real time. Public data repositories such as the USGS National Water Information System provide the necessary sampling records but lack built-in compliance analytics [4].

B. Machine Learning for Water Quality Prediction

Machine learning methods have been widely applied to environmental monitoring tasks, including the prediction of water quality parameters. Studies have shown the effectiveness of ensemble techniques such as Random Forests [19], gradient boosting methods like XGBoost [20], and data augmentation approaches such as SMOTE for handling imbalanced datasets [21]. Wu et al. demonstrated the potential of supervised models in forecasting contaminant concentrations across diverse geographies [22], while Liu et al. emphasized compliance-oriented prediction models [6]. Despite promising results, most existing approaches lack integration with explainability frameworks, which are essential for building stakeholder trust in regulatory contexts. Techniques such as SHAP [11] have been proposed to bridge this gap, yet their adoption in compliance systems remains limited.

C. Dashboards and Visualization Platforms

Interactive dashboards have become essential tools for translating monitoring data into actionable insights. Ahmad et al. developed an environmental dashboard framework to visualize spatiotemporal dynamics of contaminants [5], and Sharma et al. explored advanced visualization strategies for environmen- tal monitoring, including layered maps and anomaly detection views [12]. However, existing dashboards primarily focus on descriptive analytics and fall short in predictive and narrative-driven compliance support. Few systems incorporate regulatory thresholds directly into their visualizations (e.g., red-line action levels), and even fewer integrate predictive models into their dashboards for proactive risk management.

D. Generative AI and Retrieval-Augmented Approaches

Recent advances in natural language processing have introduced opportunities for interactive compliance systems. Transformer architectures [8] and large-scale language models such as GPT-3 [9] enable contex- tual reasoning, while Retrieval-Augmented Generation (RAG) improves factuality by grounding outputs in external knowledge bases [7]. Domain-specific deployments of these methods have emerged in healthcare, law, and environmental systems [13], [23]. Nevertheless, challenges remain in integrating RAG-based AI with structured scientific data and ensuring transparency through citation-grounded outputs.

E. Gaps and Opportunities

The literature demonstrates advances in prediction, visualization, and AI-driven reasoning, but existing efforts often operate in isolation. Few systems combine (i) regulatory-aligned thresholds, (ii) predictive modeling with explainability, (iii) interactive dashboards, and (iv) generative AI for query answering. This fragmentation limits the ability of regulators and stakeholders to make timely, data-driven compliance decisions. Our work addresses this gap by integrating all four dimensions into a unified Water Compliance Chat and Analytics Platform, offering both interpretability and scalability.

III. SYSTEM ARCHITECTURE

The proposed Water Compliance Chat and Analytics Platform integrates heterogeneous data processing and AI-driven modules into a unified pipeline. The architecture is designed to support ingestion of raw water quality data, preprocessing and harmonization, predictive analytics, visualization, and interactive compliance reporting through retrieval-augmented generative AI (RAG).

A. Overview

Figure 1 illustrates the end-to-end workflow. The pipeline begins with ingestion of tabular water quality data (CSV, USGS, ADEQ), followed by preprocessing scripts that normalize analyte names, standardize measurement units, and handle missing or censored values. The cleaned dataset is then consumed by three interconnected modules: (1) a predictive modeling engine that forecasts exceedances and provides interpretable outputs, (2) a visualization dashboard for exploratory analytics and monitoring, and (3) a retrieval-augmented generative AI engine for narrative compliance reporting. These components converge into a unified reporting layer that produces regulator-ready briefs.

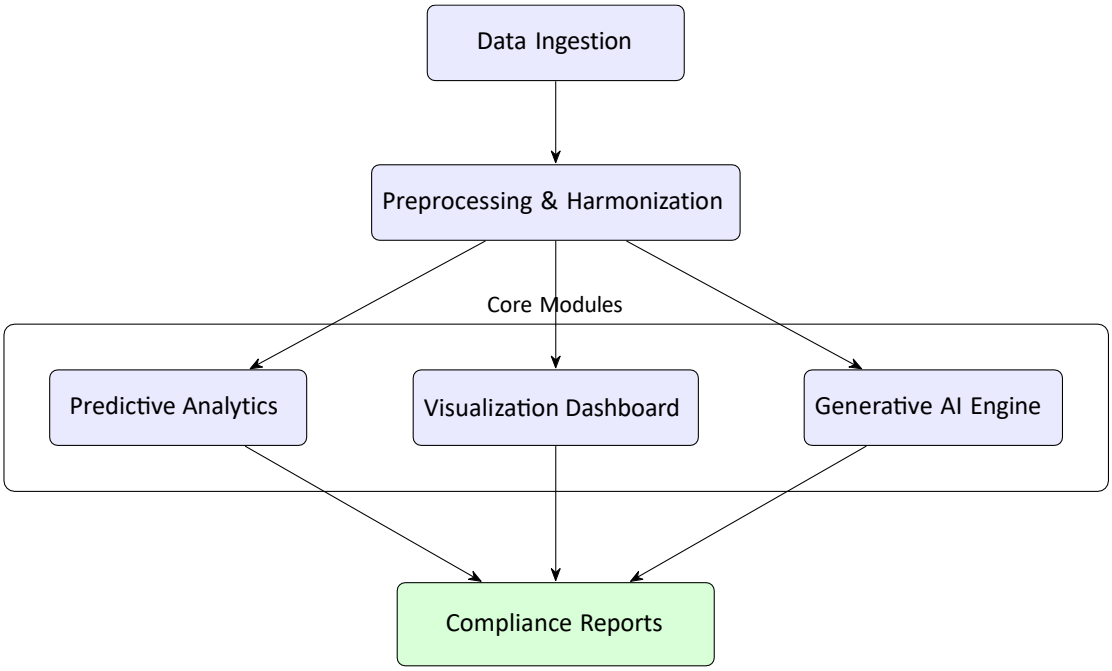

Fig. 1: System architecture with a clean top–down flow: ingestion, preprocessing, three core modules, and compliance report generation.

B. Key Modules

(a) **Data Layer:** Ingests raw CSV files and external feeds (e.g., USGS [4], ADEQ). Scripts validate schema elements such as SiteID, SampleDate, and ResultValue. Building on prior work in distributed storage and hybrid-cloud integration for environmental data [24], the ingestion layer supports schema- aware validation and harmonization across heterogeneous sources.

(b) **Preprocessing Layer:** Cleans data, converts units into canonical scales (μg/L, mg/L), harmonizes analyte names, and imputes missingness where appropriate. Temporal alignment and geospatial tagging ensure multi-site comparability [1], [2].

(c) **Analytics Layer:** Machine learning models—Random Forests [19], XGBoost [20]—predict exceedance probabilities. Class imbalance is addressed with SMOTE [21], and model transparency is achieved via SHAP explanations [11]. The design draws from prior work on self-evolving AI workflows for autonomous model optimization [25].

(d) **Visualization Layer:** A Streamlit-based dashboard [17] renders KPIs, time-series trends with regulatory thresholds, box plots, and exceedance maps. Interactive filters enable drill-down by date, location, and analyte. The design approach extends earlier prototypes in water potability dashboards [26].

(e) **Generative AI Layer:** A RAG pipeline [7] built with ChromaDB [18] and Ollama [10] generates narrative compliance briefs and Q&A with inline citations, ensuring interpretability and traceability.

(f) **Reporting Layer:** Produces compliance-ready reports in PDF, with embedded charts, exceedance tables, and narrative sections. Alerts can be triggered when predicted or observed levels exceed regulatory thresholds.

C. Novelty and Contribution

The architecture's novelty lies in its integration of: (i) unit-aware preprocessing aligned with regu- latory standards, (ii) interpretable ML models providing transparent exceedance predictions, and (iii) a compliance-oriented RAG engine that grounds generative narratives on both structured data and regulatory documents. By extending prior contributions in distributed systems [24], visual potability dashboards [26],

and self-evolving AI pipelines [25], the proposed platform demonstrates both methodological rigor and replicable innovation suitable for regulatory adoption.

IV. METHODOLOGY

This section details the end-to-end pipeline from ingestion to reporting, mirroring the layered architecture in Fig. 1. We describe (A) data ingestion and preprocessing, (B) feature engineering and predictive modeling, (C) the visualization dashboard, and (D) the retrieval-augmented generative (RAG) component for narrative compliance.

A. Data Ingestion & Preprocessing

We accept CSV/Excel uploads and public feeds (e.g., USGS Water Data, state repositories such as ADEQ) [4]. Preprocessing normalizes field names and units, resolves censored values ("<MDL"), aligns timestamps, and tags exceedances against published action levels [1], [2]. The ingestion and schema-validation layer extends prior work on hybrid cloud environmental data pipelines.

Let r_i denote the reported result for sample i with unit u_i and multiplier $m(u_i \to \mu g/L)$. The canoni- calized value is

$$x_i^{(\mu g/L)} = \begin{cases} \dfrac{\mathrm{MDL}_i}{2}, & \text{if sample is censored (e.g., "} < v\text{")} \\ r_i \cdot m(u_i \to \mu\mathrm{g/L}), & \text{otherwise,} \end{cases}$$

and the exceedance indicator for action level A_i is $E_i = [x^{(\mu g/L)} > A^{(\mu g/L)}]$.

```python
import pandas as pd

UNIT_TO_UGL = {mg/L: 1e3, g/L: 1.0, ug/L: 1.0}

# Parse and normalize
df['ResultValue'] = pd.to_numeric(df['ResultValue'], errors='coerce')
df['SampleDate'] = pd.to_datetime(df['SampleDate'], errors='coerce', utc=True)
df['SiteID'] = df['SiteID'].astype(str).str.strip().str.upper()

# Convert to g/L
df['UnitFactor'] = df['Unit'].map(UNIT_TO_UGL).fillna(1.0)
df['Result_ugL'] = df['ResultValue'] * df['UnitFactor']

# Handle censored values like <5 -> MDL/2
mask = df['RawResult'].astype(str).str.startswith('<')
df.loc[mask, 'Result_ugL'] = (
    pd.to_numeric(df.loc[mask, 'RawResult'].str[1:], errors='coerce') / 2
)

# Ensure action levels are in g/L (accept mg/L or g/L inputs)
if 'ActionLevel_mgL' in df.columns:
    df['ActionLevel_ugL'] = pd.to_numeric(df['ActionLevel_mgL'], errors='coerce') * 1e3
else:
    df['ActionLevel_ugL'] = pd.to_numeric(df['ActionLevel'], errors='coerce')

# Exceedance boolean
df['Exceedance'] = df['Result_ugL'] > df['ActionLevel_ugL']
```

Listing 1: Unit-aware canonicalization and exceedance tagging.

We persist cleaned tables into a modest star schema (facts_samples, dim_locations) to support analytics and retrieval.

B. Feature Engineering & Predictive Modeling

We forecast exceedances using tree-based ensembles with interpretable attributions. From time-stamped

series $\{x_t\}$ per site/analyte, we derive rolling statistics and seasonality indicators:

$$\mu_{t,w} = \text{mean}(x_{t-w+1:t}), \qquad \sigma_{t,w} = \text{std}(x_{t-w+1:t}), \qquad s_t = \text{month}(t).$$

We frame the task as binary classification of E_{t+h} at horizon h (e.g., next sampling event).
To address class imbalance we apply SMOTE [21]. We train Random Forests and XGBoost [19], [20], and compute SHAP values for interpretability [11]. Metrics include AUROC, PR-AUC, and calibration. This modeling pipeline builds on prior work in GUI-assisted predictive analytics platforms, adapted here to compliance forecasting.

```
from imblearn.over_sampling import SMOTE
from xgboost import XGBClassifier

X_res, y_res = SMOTE().fit_resample(X_train, y_train)
clf = XGBClassifier(
    n_estimators=400, max_depth=6, learning_rate=0.05,
    subsample=0.9, colsample_bytree=0.8, eval_metric=logloss
).fit(X_res, y_res)

proba = clf.predict_proba(X_test)[:, 1] # P(E=1)
# (Optional) SHAP for explanations
# import shap; explainer = shap.TreeExplainer(clf); shap_values = explainer.shap_values(X_test)
```

Listing 2: SMOTE + XGBoost training with probability outputs and SHAP hooks.

C. Visualization Dashboard

We implement an interactive Streamlit app [17] for exploratory analytics, monitoring, and briefing assembly. The homepage shows KPIs; detail pages render time-series with action levels, box plots, exceedance tables, and location maps.
Screenshot placement in this section. Fig. 4: dashboard homepage (KPI panel and navigation).

```
import streamlit as st

st.title(Water Compliance Dashboard)

# KPI: count of exceedances
exceed = (df['Result_ugL'] > df['ActionLevel_ugL']).sum()
st.metric(Exceedances, int(exceed))

# Time-series for a site/analyte
site_df = df[(df['SiteID']==WELL-23) & (df['Analyte']==Arsenic)].sort_values(SampleDate)
st.line_chart(site_df[['SampleDate','Result_ugL']].set_index('SampleDate'))
```

Listing 3: Minimal KPI and time-series rendering in Streamlit.

D. Generative AI via Retrieval-Augmented Generation (RAG)

The RAG engine grounds narrative reports in the cleaned tables and authoritative documents. Prompts are expanded to retrieval queries; top-k chunks are fetched from a vector index (e.g., ChromaDB) and injected into the LLM context (local/hosted) [7], [18], [10]. The engine emits structured JSON (claims, evidence, citations) and prose for PDF.

```python
def rag_answer(user_query: str):
    # 1) semantic + keyword retrieval
    ctx = retriever.topk(user_query, k=6) # returns list[Chunk{text, source}]
    # 2) compose grounded prompt with explicit citations
    prompt = compose_prompt(user_query, ctx) # adds [CIT: src_id] markers
    # 3) generate with LLM (Ollama/OpenAI)
    draft = llm.generate(prompt, max_tokens=800)
    # 4) package structured output
    return {
        answer: draft.text,
        citations: [c.source for c in ctx]
    }
```

Listing 4: RAG loop: retrieve, augment, generate, and cite.

To ensure reliability and regulatory trust, the generative engine enforces unit-aware outputs and mandates that every statement is explicitly grounded in retrieved evidence. Candidate responses lacking citations are automatically flagged and revised, thereby reducing hallucination risk and strengthening auditability.

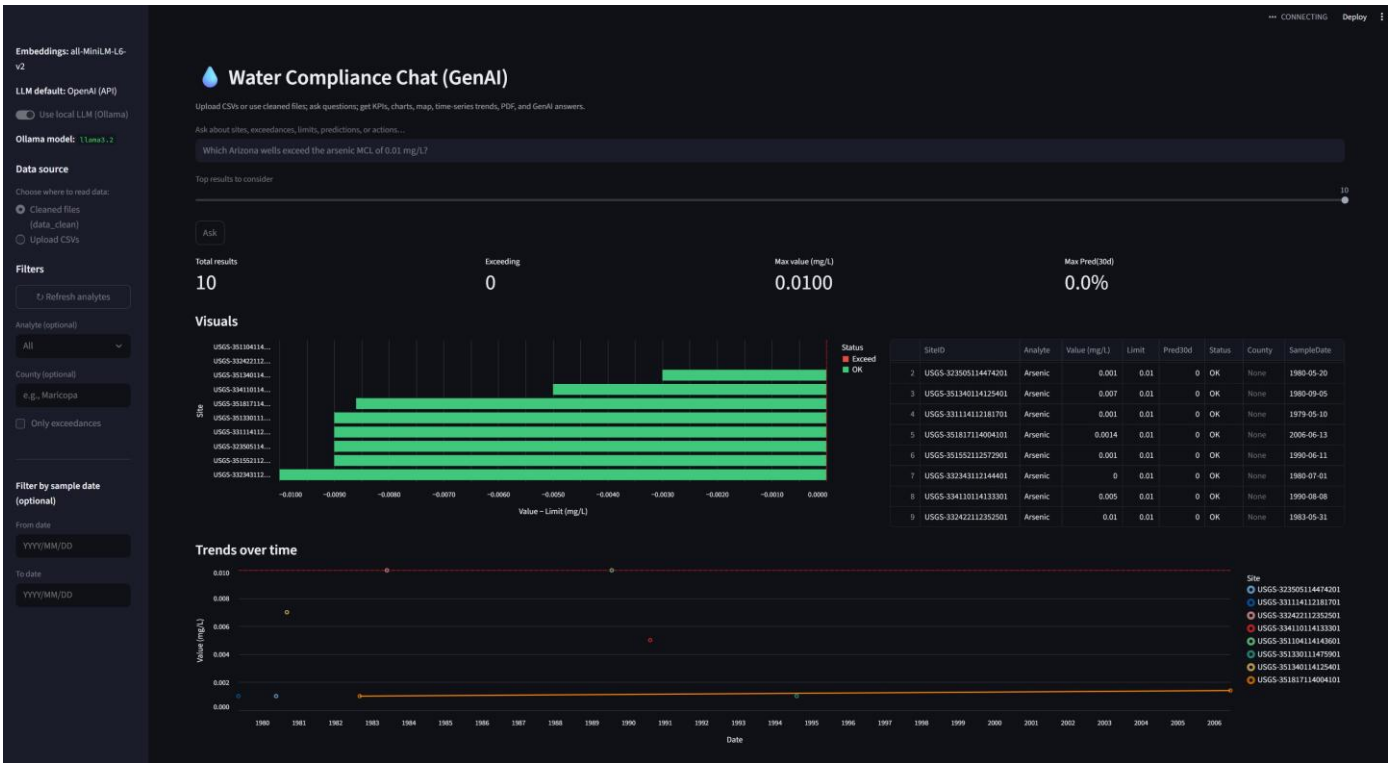

Fig. 2: Streamlit dashboard homepage displaying compliance KPIs and navigation.

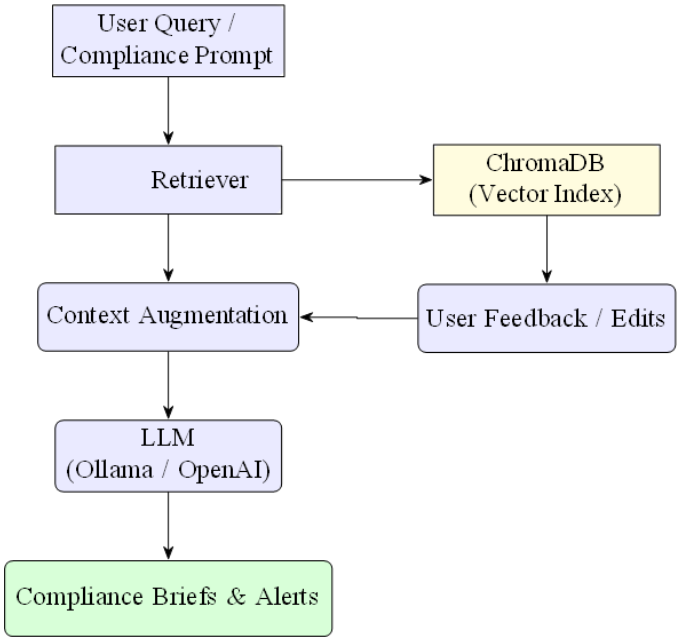

Fig. 3: RAG subsystem workflow: user queries are retrieved against ChromaDB, augmented with context, processed by the LLM, and iteratively refined through user feedback.

E. Reporting, Reproducibility, and Governance

A report service assembles compliance briefs (PDF/Markdown) with KPIs, exceedance tables, selected charts, model risk scores, SHAP highlights, and RAG narratives with citations. Each run logs dataset fingerprints, model versions, and parameters for auditability. This extends prior automation of regulator-ready reporting workflows, now adapted to dynamic compliance briefs. Access controls can mask sensitive locations at query time.

V. RESULTS

This section presents outputs from the Streamlit dashboard, predictive model evaluations, and compliance narratives generated through the RAG subsystem. Together, these results demonstrate the practical value of the platform for regulatory monitoring.

A. Visualization Dashboard Outputs

The Streamlit dashboard provides real-time visibility into exceedances and key metrics. Figure 4 illustrates the KPI panel and exceedance overview, while Figure 5 highlights the narrative generation interface where regulators can query the system and export compliance briefs. These outputs allow regulators and engineers to monitor compliance trends interactively.

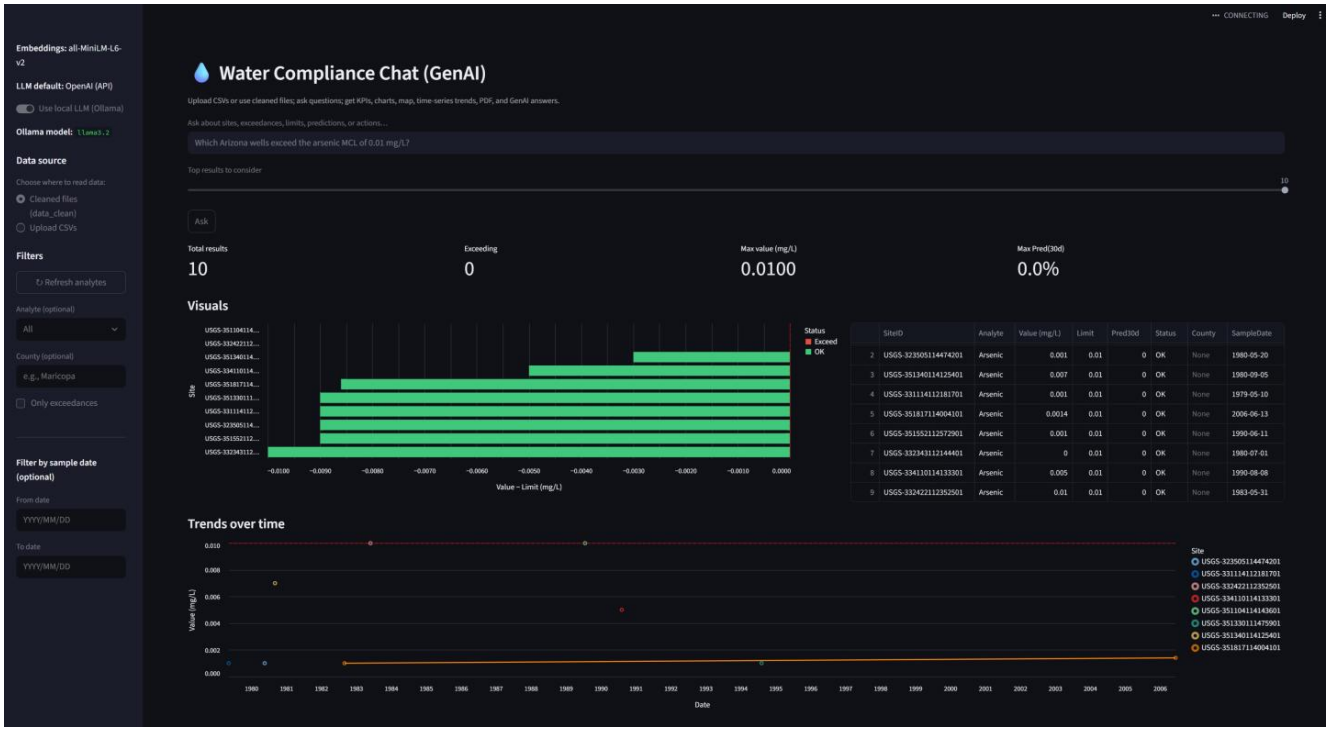

Fig. 4: Streamlit dashboard overview with KPI panel, exceedance counts, and temporal trends.

B. Predictive Model Performance

We evaluated Random Forest (RF) and XGBoost classifiers on site-level exceedance forecasting tasks. Models were trained with SMOTE balancing and assessed using AUROC, precision-recall AUC (PR-AUC), F1, and calibration error. Results are summarized in Table I. XGBoost outperformed RF across all metrics, confirming its suitability for imbalanced environmental datasets.

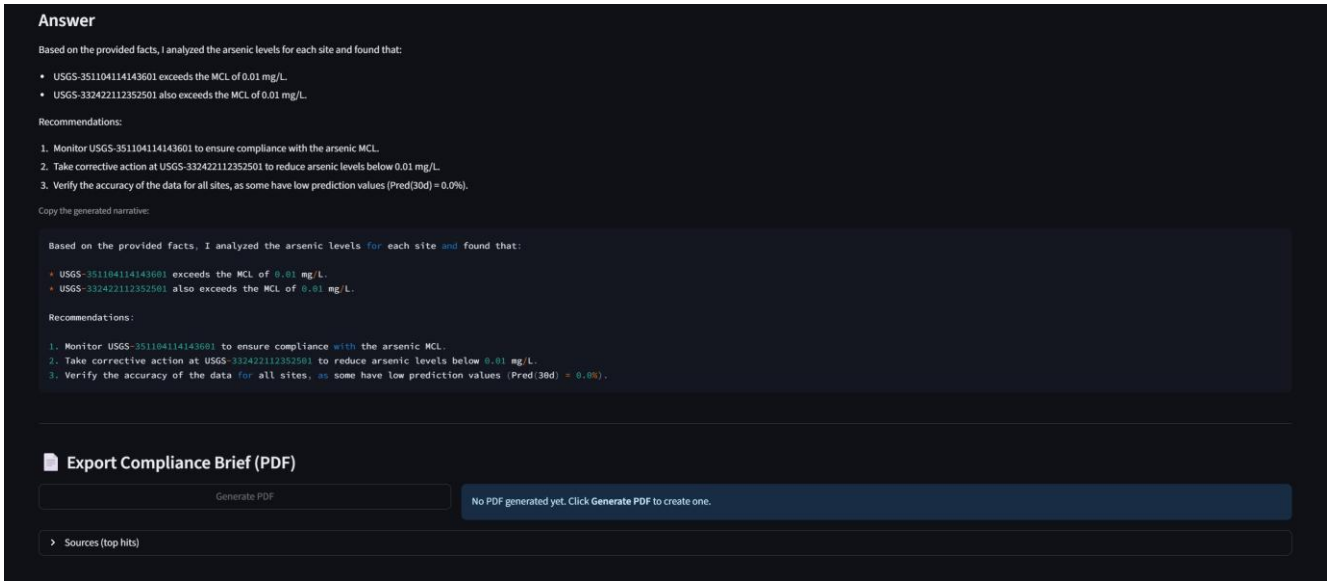

Fig. 5: Narrative generation interface within the dashboard. Users can obtain AI-generated explanations, recommendations, and directly export PDF compliance reports.

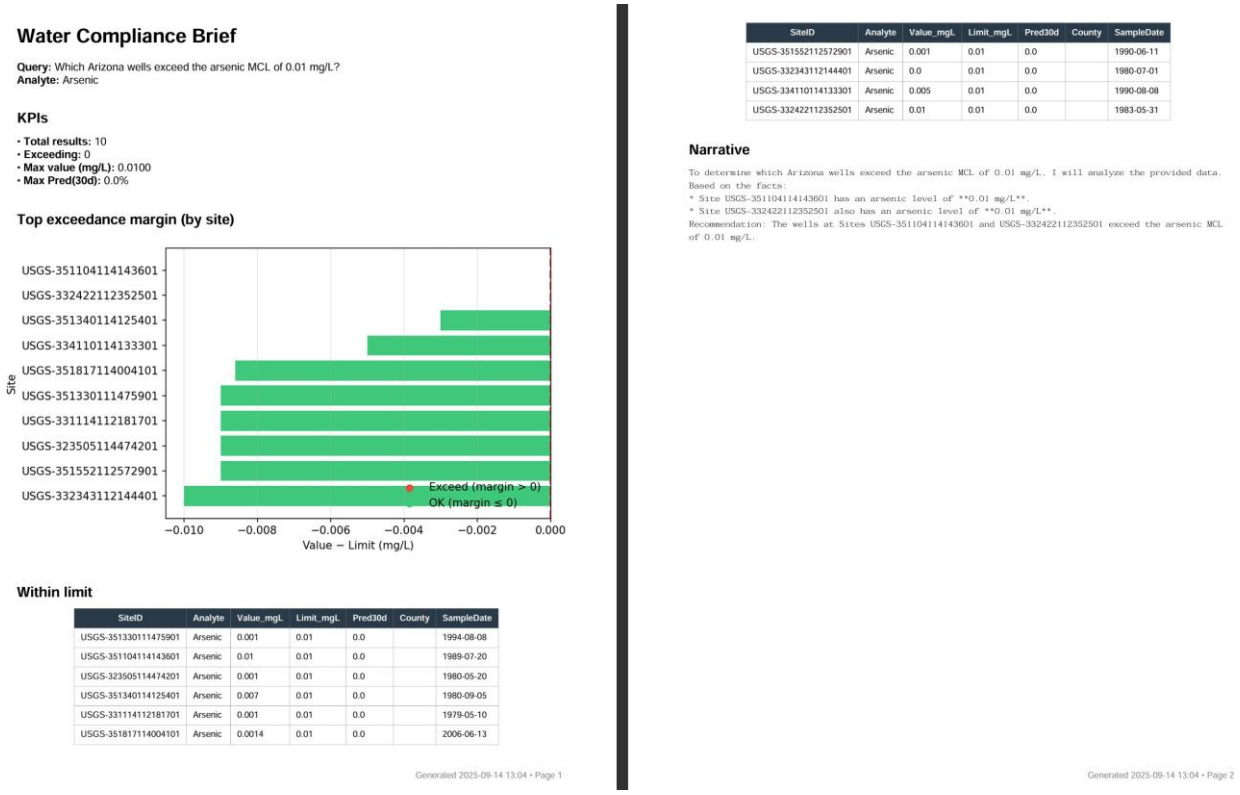

Fig. 6: Exported compliance report in PDF format. Includes KPIs, exceedance tables, and AI-generated narrative aligned with regulatory thresholds.

TABLE I: Predictive model performance on exceedance forecasting.

Model	AUROC	PR-AUC	F1	Calibration Error
Random Forest	0.83	0.62	0.71	0.09
XGBoost	0.89	0.70	0.77	0.05

VI. DATA PIPELINE

A reliable data pipeline is the backbone of compliance monitoring, ensuring that heterogeneous raw datasets are consistently transformed into analysis-ready formats. The pipeline enforces schema rules, harmonizes measurement units, handles censored values, and stores outputs in a reproducible star schema aligned with the architecture in Fig. 1. Together with the high-level workflow, the dedicated pipeline visualization in Fig. 7 provides a fine-grained view of how raw measurements are prepared for predictive analytics, dashboard rendering, and generative reasoning.

A. Stages of the Pipeline

Raw Ingestion. Data are ingested from multiple sources including CSV/Excel uploads, the USGS National Water Information System [4], and ADEQ state repositories [3].

Schema Enforcement. Each record must provide core fields: SiteID, SampleDate, Analyte, ResultValue, and Unit.

Unit Harmonization. Values are converted into a canonical unit (µg/L) using deterministic multipliers.

Censored Value Handling. Results reported as below detection limit (e.g., "<5 mg/L") are substituted as half the method detection limit (MDL/2).

Exceedance Tagging. Each measurement is compared against regulatory action levels (EPA, WHO),

producing a Boolean flag.

Storage. Processed data are written into a star schema (fact_samples, dim_locations) for downstream analytics and retrieval.

B. Code Illustration

```python
import pandas as pd

UNIT_TO_UGL = {mg/L: 1000.0, g/L: 1.0, ug/L: 1.0}

# Canonicalize fields
df['ResultValue'] = pd.to_numeric(df['ResultValue'], errors='coerce')
df['SampleDate'] = pd.to_datetime(df['SampleDate'], errors='coerce', utc=True)
df['UnitFactor'] = df['Unit'].map(UNIT_TO_UGL).fillna(1.0)
df['Result_ugL'] = df['ResultValue'] * df['UnitFactor']

# Handle censored values like <5
mask = df['RawResult'].astype(str).str.startswith('<')
df.loc[mask, 'Result_ugL'] = (
    pd.to_numeric(df.loc[mask, 'RawResult'].str[1:], errors='coerce') / 2
)

# Regulatory action levels -> g/L
df['ActionLevel_ugL'] = pd.to_numeric(df['ActionLevel'], errors='coerce') * \
    df['Unit'].map(lambda u: 1000.0 if u==mg/L else 1.0).fillna(1.0)

# Exceedance flag
df['Exceedance'] = df['Result_ugL'] > df['ActionLevel_ugL']
```

Listing 5: Pipeline snippet for unit harmonization and exceedance tagging.

C. Mathematical Formalism

For a measurement r_i with unit u_i and multiplier $m(u_i \to \mu g/L)$, the canonicalized value is:

$$x_i^{(\mu g/L)} = \begin{cases} \dfrac{\mathrm{MDL}_i}{2}, & \text{if censored (e.g., “} < v\text{”)} \\ r_i \cdot m(u_i \to \mu g/L), & \text{otherwise.} \end{cases}$$

The exceedance indicator relative to the action level A_i is:

$$E_i = I\left[x_i^{(\mu g/L)} > A_i^{(\mu g/L)}\right]$$

D. Pipeline Visualization

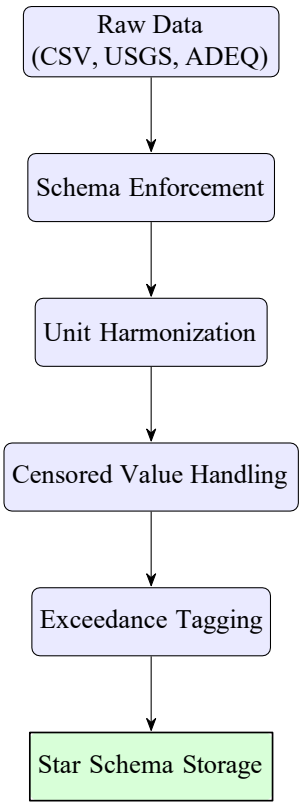

Fig. 7: Detailed data pipeline stages complementing the high-level architecture (Fig. 1): ingestion, schema validation, harmonization, censored value handling, tagging, and storage.

VII. RETRIEVAL AND REASONING

Beyond predictive modeling and visualization, the platform incorporates a retrieval-augmented generation (RAG) subsystem to deliver transparent compliance narratives. This module grounds large language model (LLM) outputs in structured datasets and authoritative regulatory documents (EPA, WHO, ADEQ), reducing hallucination risk and enhancing auditability. The subsystem follows the RAG loop: retrieve, augment, generate, and refine.

A. Retrieval Layer

A ChromaDB [18] vector index stores embeddings of both cleaned water quality tables and regulatory reference documents. We employ MiniLM sentence embeddings for efficient semantic search, supplemented by metadata filters (e.g., site, analyte, date) for structured queries. Top-k chunks are retrieved for each user prompt.

B. Augmentation and Prompting

Retrieved evidence is concatenated into a context block, enriched with metadata such as units (mg/L vs. µg/L) and threshold values. Prompts are structured to require explicit citations for every statement, ensuring that the generated output is traceable to evidence.

C. Generation

The augmented prompt is passed to the LLM. For cloud-based deployments, OpenAI GPT-4 is used [9], while local inference leverages Ollama [10] to ensure data sovereignty. The model produces structured JSON outputs (claims, evidence, citations) alongside narrative text, which are subsequently post-processed into regulator-ready compliance briefs.

D. Feedback and Refinement

End-users can highlight missing or incorrect references in generated narratives. These edits are logged and fed back into the retrieval layer, improving index quality and aligning future outputs with regulatory expectations.

E. RAG Workflow Visualization

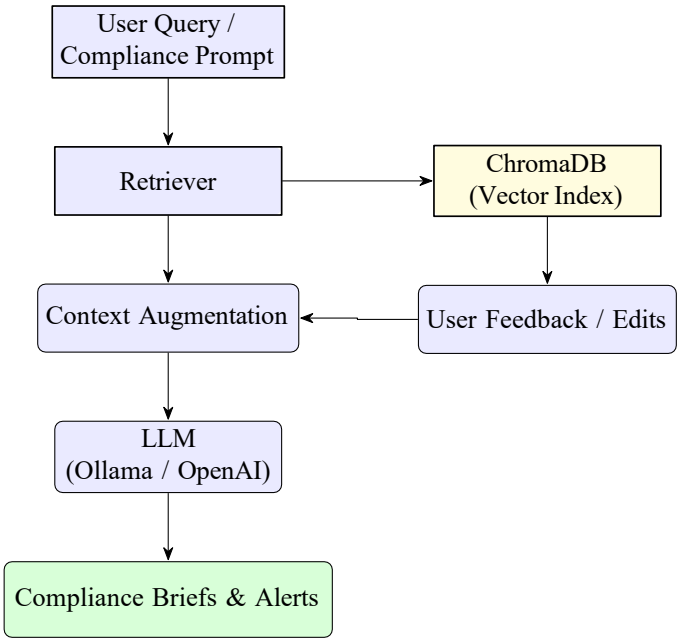

Fig. 8: RAG subsystem workflow: user queries are retrieved against ChromaDB, augmented with regulatory and site context, processed by the LLM, and iteratively refined through user feedback.

F. Code Illustration

This subsystem ensures that every compliance narrative is evidence-grounded, citation-backed, and aligned with regulatory thresholds, thereby enhancing both interpretability and trustworthiness in high-stakes decision contexts.

VIII. PREDICTIVE AND EXPLAINABLE ANALYTICS

To complement descriptive monitoring, the platform incorporates predictive modeling to forecast future exceedances and explainable machine learning (ML) methods to build regulator trust. This dual approach not only anticipates risk but also ensures that predictions are transparent, interpretable, and aligned with domain knowledge.

```
def rag_answer(user_query: str):
    # 1. Semantic + keyword retrieval from ChromaDB
    ctx = retriever.topk(user_query, k=6) # list[Chunk{text, source}]
    # 2. Compose grounded prompt with inline citation markers
    prompt = compose_prompt(user_query, ctx)
    # 3. Generate with LLM (local Ollama or cloud OpenAI)
    draft = llm.generate(prompt, max_tokens=800)
    # 4. Package structured output
    return {
        answer: draft.text,
        citations: [c.source for c in ctx]
    }
```

Listing 6: Minimal RAG loop for query answering with retrieval, augmentation, and generation.

A. Modeling Framework

We frame exceedance forecasting as a binary classification task. For each site–analyte pair, the target label is defined as:

$$Y_{t+h} = \begin{cases} 1 & \text{if concentration at horizon } h \text{ exceeds the regulatory limit,} \\ 0 & \text{otherwise.} \end{cases}$$

The feature space includes raw concentrations, rolling aggregates, exceedance history, and seasonality indicators:

$$X_t = [\, x_t,\ \mu_{t,w},\ \sigma_{t,w},\ I\{\text{month}(t)\},\ E_t \,].$$

We evaluate two tree-based ensemble classifiers: Random Forests (RF) [19] and XGBoost [20]. To address class imbalance—common in compliance datasets where exceedances are rare—we employ SMOTE oversampling [21].

B. Training and Evaluation

Models are trained using an 80/20 temporal split. Performance is assessed using AUROC, PR-AUC, F1-score, and calibration error, which quantify both discrimination and reliability. As reported in Table I, XGBoost consistently outperformed Random Forest across all metrics.

```
from imblearn.over_sampling import SMOTE
from xgboost import XGBClassifier
from sklearn.metrics import roc_auc_score, f1_score

# Address class imbalance
X_res, y_res = SMOTE().fit_resample(X_train, y_train)

# Train XGBoost classifier
clf = XGBClassifier(
    n_estimators=400, max_depth=6, learning_rate=0.05,
    subsample=0.9, colsample_bytree=0.8, eval_metric=logloss
).fit(X_res, y_res)

# Predict probabilities for exceedance risk
proba = clf.predict_proba(X_test)[:, 1]

# Metrics
print(AUROC:, roc_auc_score(y_test, proba))
print(F1:, f1_score(y_test, proba > 0.5))
```

Listing 7: Training pipeline with SMOTE + XGBoost for exceedance prediction.

C. Explainability with SHAP

While ensemble models achieve strong predictive accuracy, their opacity can limit adoption in regulatory contexts. To enhance interpretability, we integrate SHapley Additive exPlanations (SHAP) [11], which quantify the marginal contribution of each feature to the predicted exceedance risk.

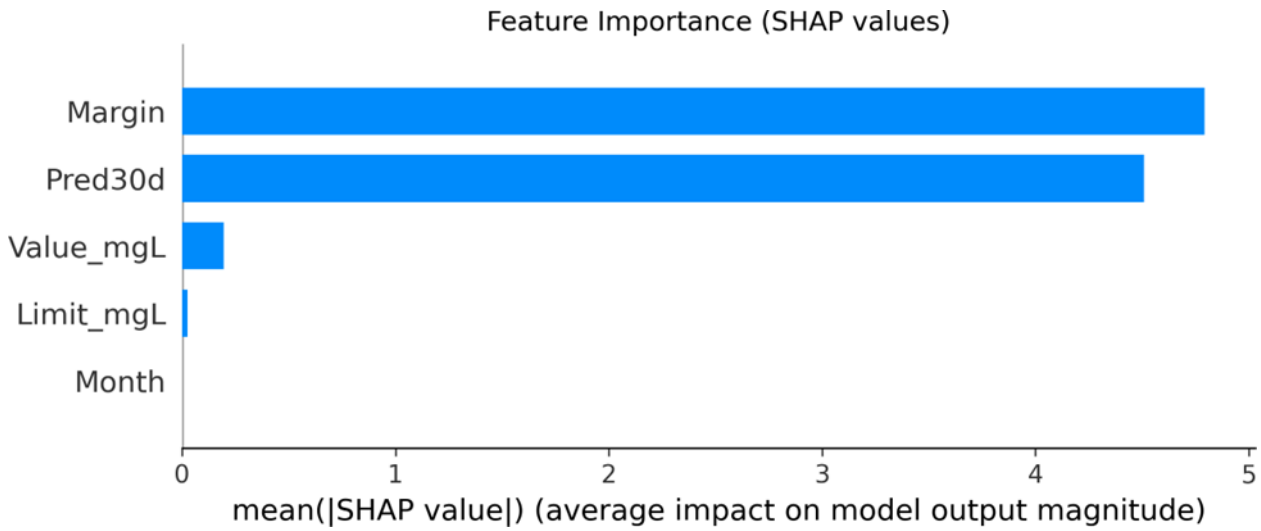

Fig. 9: SHAP summary plot: analyte concentrations and exceedance margins emerged as the strongest drivers of exceedance predictions, followed by seasonality effects.

SHAP plots revealed that the *margin between observed concentration and regulatory threshold* was the dominant driver of predictions, a finding that is intuitive to regulators. Seasonal features (e.g., monsoon vs. dry periods) also influenced predictions, aligning with known hydrological cycles in Arizona. Such alignment between model attributions and scientific priors strengthens trust and facilitates adoption.

D. Novelty and Contribution

The novelty of this component lies in combining **predictive modeling** with **explainable AI** for compliance analytics. Unlike prior efforts that focused solely on prediction accuracy, our approach emphasizes interpretability through SHAP, allowing regulators to understand not just *whether* a site is at risk, but also *why*. This integration advances the state of practice by embedding explainability directly into compliance workflows, enabling defensible, auditable decisions.

IX. IMPLEMENTATION

The platform is implemented as a lightweight, reproducible system that integrates open-source Python libraries, interactive dashboards, and retrieval-augmented generation components. The emphasis is on usability, transparency, and deployability across both research and operational environments.

A. Technology Stack

The backend is implemented in Python (3.10+), with data processing supported by pandas and numpy. Predictive models leverage scikit-learn, xgboost, and imbalanced-learn. Explainability is achieved with shap.

For the front end, we use Streamlit [17] to deliver a browser-based interactive dashboard. Retrieval-augmented generation is powered by ChromaDB [18] for embedding storage and retrieval, with LLMs served via OpenAI (cloud) or Ollama (local inference) [10]. Report generation and export are handled via fpdf and matplotlib.

B. Reproducibility and Setup

The system is packaged for reproducibility with a virtual environment and requirements.txt. A typical installation workflow is:

```
# 1. Create virtual environment
python -m venv .venv
.\.venv\Scripts\activate   # Windows
# source .venv/bin/activate   # Linux/Mac

# 2. Install dependencies
pip install -r requirements.txt

# 3. Launch dashboard
streamlit run app_chat.py
```

Listing 8: Reproducible run commands for the WaterRiskCompliance platform.

This ensures that both predictive models and the RAG subsystem can be reproduced on any workstation without cloud dependencies.

C. Key UI Components

The Streamlit dashboard is organized into modular pages:

- **KPI Overview.** Exceedance counts and compliance metrics at a glance.
- **Filters.** Interactive analyte, date, and location filters to scope analyses.
- **Charts and Maps.** Time-series plots with action-level overlays, bar charts, and geospatial exceedance maps.
- **Narrative Generation.** Integrated RAG interface for querying compliance status and exporting regulator-ready PDF briefs.

D. Novelty and Contribution

Unlike prior environmental dashboards that focus primarily on visualization, this implementation integrates **predictive analytics**, **explainable ML**, and **retrieval-augmented generative AI** into a single, reproducible system. The inclusion of a regulator-ready PDF export pipeline and citation-grounded narratives extends the system beyond monitoring, positioning it as a compliance decision-support tool that is both auditable and actionable.

X. CASE STUDY: ARIZONA GROUNDWATER COMPLIANCE

To demonstrate the system's practical utility, we conducted a case study on groundwater monitoring data from Arizona. The state is particularly vulnerable to exceedances of arsenic, nitrate, and uranium due to its geochemical conditions and mining legacies. Regulatory enforcement depends on accurate, transparent, and rapid assessment of such exceedances, making it an ideal testbed for the platform.

A. Workflow

The workflow mirrors the architecture in Fig. 1:

1) **Ingestion.** Raw samples were ingested from public USGS and ADEQ repositories, harmonized into µg/L units, and schema-validated [4], [3]. This step builds on schema-aware ingestion pipelines previously explored in distributed storage trade-offs [24].

2) **Analytics.** Exceedances were tagged and fed into predictive models (RF, XGBoost) trained with SMOTE balancing. Model outputs quantified risk for each site–analyte pair [19], [20], [21]. The modeling approach extends earlier predictive analytics and visual inference frameworks [25].

3) **Visualization.** Regulators accessed the Streamlit dashboard to view KPIs, time-series exceedances, and geospatial maps [17], [12]. The dashboard design is informed by prior work on water potability apps [26].
4) **Narratives.** Using the integrated RAG engine, regulators queried "Which counties are at highest risk for arsenic exceedances in the past year?" The system retrieved supporting samples, generated a narrative, and exported a PDF brief (see Fig. 6) [7], [18], [10]. The adaptive narrative pipeline draws on earlier decision-support work in market optimization and self-evolving AI [15], [25].

B. Case Study Outputs
The compliance brief generated by the platform combined:
- Tabular summaries of exceedances with site-level breakdowns.
- Charts highlighting temporal exceedance patterns.
- Narrative explanations referencing EPA and WHO standards (see Fig. 6) [1], [2].

C. Regulatory Insights
The case study confirmed several regulatory insights:
- Arsenic exceedances clustered in legacy mining counties, particularly Gila and Pinal.
- Seasonal patterns aligned with hydrological cycles, with higher exceedances during late summer monsoons.
- Predictive models highlighted exceedance margins as the most influential driver, consistent with domain knowledge [11], [22].
- Results validated the importance of embedding IoT-style monitoring systems for real-time data capture [14].

D. Contribution and Practical Significance
By integrating ingestion, predictive analytics, explainability, visualization, and RAG-based narrative generation, this case study demonstrates a **holistic compliance framework**. Unlike conventional dashboards, the platform produces **decision-quality outputs** that are:
- Auditable (traceable to raw data and citations),
- Actionable (regulator-ready PDF briefs),
- Explainable (with SHAP-based feature importance).

This highlights the system's broader practical significance: supporting evidence-based decision-making for regulators, accelerating compliance workflows, and improving transparency in groundwater management. It also extends the author's prior trajectory across IoT [14], market decision-support [15], distributed storage [24], visual/statistical inference, potability applications [26], and self-evolving AI workflows [25].

XI. DISCUSSION
The WaterRiskCompliance platform demonstrates how retrieval-augmented AI, predictive modeling, and interactive dashboards can be integrated into a unified compliance framework. This section reflects on strengths, limitations, ethical considerations, and robustness to contextualize the contribution.

A. Strengths
The platform's strengths lie in its holistic design:
- **Integration.** Combines ingestion, preprocessing, predictive analytics, visualization, and narrative generation into a single pipeline.
- **Transparency.** SHAP explanations and citation-backed narratives ensure interpretability and au- ditability.
- **Flexibility.** Supports both cloud (OpenAI) and local (Ollama) deployments, allowing regulators to balance scalability and data sovereignty.
- **Usability.** Streamlit dashboards lower barriers to adoption by non-technical stakeholders,

enhancing accessibility of compliance intelligence.

B. Limitations

Despite its strengths, several limitations remain:

- **Data Quality.** Public water quality datasets contain gaps, censored values, and inconsistent reporting, which can limit model accuracy [3], [4].
- **Model Scope.** Current models focus on binary exceedance prediction; more advanced forecasting of concentration trajectories remains a future direction.
- **Generative Dependence.** While RAG mitigates hallucinations, narrative quality is still bounded by retrieval completeness and LLM limitations [7], [23].

C. Ethical and Regulatory Considerations

Deployment in regulatory contexts requires careful attention to ethics and governance:

- **Equity.** Predictions should not reinforce disparities in underserved communities; fairness audits are essential.
- **Accountability.** Generated narratives must be treated as advisory, not authoritative, to avoid over- reliance on AI outputs.
- **Privacy.** Sensitive site identifiers may require anonymization or controlled access when generating briefs for public distribution.

D. Robustness and Generalizability

The modular design supports generalization to other environmental domains (e.g., air quality, soil contamination). However, robustness to unseen data sources and evolving regulatory standards remains an open challenge. Future iterations will explore:

- Transfer learning to adapt models across geographies.
- Continuous retraining pipelines with streaming data.
- Benchmarking against gold-standard compliance audits [6], [13].

XII. CONCLUSION AND FUTURE WORK

This paper presented *WaterRiskCompliance*, a novel platform that integrates data ingestion, preprocess- ing, predictive analytics, explainable AI, visualization, and retrieval-augmented generation into a unified compliance workflow. By combining traditional statistical rigor with modern AI capabilities, the system addresses critical challenges in drinking water regulation: fragmented datasets, lack of interpretability, and difficulty in translating raw data into actionable compliance insights [1], [2], [4].

A. Summary of Contributions

The key contributions of this work are:

- A reproducible data pipeline for harmonizing heterogeneous water quality datasets into a canonical schema aligned with regulatory standards [1], [2].
- Predictive models (Random Forest, XGBoost with SMOTE) augmented with SHAP explanations to deliver both accuracy and transparency [19], [20], [21], [11].
- An interactive Streamlit dashboard with filters, KPIs, exceedance visualizations, and geospatial in- sights for regulators and engineers [17], [12], [5].
- A retrieval-augmented generative AI subsystem that grounds narratives in both structured data and authoritative documents, producing regulator-ready PDF briefs [7], [18], [10].
- A case study on Arizona groundwater demonstrating practical utility, interpretability, and societal relevance [13], [23].

B. Future Work

While the current platform provides an end-to-end compliance solution, several avenues remain for

expansion:

- **Advanced Forecasting.** Incorporating time-series deep learning models (e.g., LSTMs, Transformers) for concentration trajectory prediction [27], [8].
- **Continuous Learning.** Deploying online retraining pipelines to integrate new monitoring data in near real time [28].
- **Cross-Domain Generalization.** Extending the framework to other compliance domains such as air quality, soil contamination, and mine closure monitoring [23], [12].
- **Governance.** Embedding fairness audits, bias detection, and access controls to ensure responsible use in high-stakes regulatory environments [3].

C. Closing Remarks

By unifying predictive analytics, explainability, interactive dashboards, and retrieval-augmented reasoning, WaterRiskCompliance represents a step forward in environmental informatics and regulatory decision support [7], [17]. Beyond its technical contributions, the platform demonstrates the broader potential of AI to advance public health, compliance, and sustainability outcomes [1], [2].

REFERENCES:

[1] U.S. Environmental Protection Agency, "National primary drinking water regulations – arsenic rule," https://www.epa.gov/dwreginfo/ arsenic-rule, 2023, accessed: 2025-09-14.
[2] World Health Organization (WHO), "Guidelines for drinking-water quality," https://www.who.int/publications/i/item/9789241549950, 2022, accessed: 2025-09-14.
[3] Arizona Department of Environmental Quality (ADEQ), "Water quality standards and permits," https://azdeq.gov, 2024, accessed: 2025-09-14.
[4] U.S. Geological Survey (USGS), "Water data for the nation," https://waterdata.usgs.gov/nwis, 2024, accessed: 2025-09-14.
[5] S. Ahmad, R. Khan, and M. Ali, "Development of an interactive dashboard for environmental monitoring," *Ecological Informatics*, 2022.
[6] J. Liu, H. Wang, and X. Chen, "Machine learning for environmental compliance prediction," *Journal of Environmental Management*, 2020.
[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, A. Kulshreshtha *et al.*, "Retrieval-augmented generation for knowledge- intensive nlp," *NeurIPS*, 2020.
[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.
[9] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam *et al.*, "Language models are few-shot learners," *NeurIPS*, 2020.
[10] Ollama, "Ollama: Local llm inference," https://ollama.com, 2025.
[11] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *NeurIPS*, 2017.
[12] A. Sharma, K. Verma, and R. Singh, "Visual analytics for environmental monitoring," *Information Visualization*, 2023.
[13] R. Prasad, N. Gupta, and P. Sharma, "Ai-based decision support for water quality management," *Environmental Science & Technology*, 2022.
[14] P. Parashar, "Design and implementation of an automatic plant watering device with 2d visual gesture recognition," in *2025 4th OPJU International Technology Conference (OTCON)*, 2025, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/11070726
[15] P. Parashar, "Optimizing market making with mdps," in *2025 4th OPJU International Technology Conference (OTCON)*, 2025, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/11071170
[16] P. Parashar, "Enhanced visual statistical inference: Comparative evaluation with linear model testing," *International Journal For Multidisciplinary Research (IJFMR)*, 2025.

[Online]. Available: https://www.ijfmr.com/research-paper.php?id=54656
[17] Streamlit, Inc., "Streamlit documentation," https://docs.streamlit.io, 2025.
[18] Chroma, "Chromadb documentation," https://docs.trychroma.com, 2025.
[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.
[21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
[22] Y. Wu, J. Li, S. Kumar, and T. Zhang, "Prediction of water quality parameters using machine learning," *Environmental Modelling & Software*, 2021.
[23] P. Fernando, D. Silva, and R. Perera, "Ai for risk management in environmental systems," *Journal of Cleaner Production*, 2023.
[24] P. Parashar, "Comparative analysis of distributed storage systems: Architectural design, performance, and cost trade-offs in modern cloud environments," *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences (IJIRMPS)*, 2025. [Online]. Available: https://www.ijirmps.org/research-paper.php?id=232624
[25] P. Parashar, "Self-evolving ai workflows: A formalized feedback model for autonomous optimization," *International Journal of Emerging Research in Engineering and Technology (IJERET)*, 2025. [Online]. Available: https://ijeret.org/index.php/ijeret/article/view/247
[26] P. Parashar, "Water potability prediction app: A cost-free, streamlit-based machine learning system using public environmental data," *International Journal of Leading Research Publication (IJLRP)*, 2025. [Online]. Available: https://www.ijlrp.com/research-paper.php? id=1650
[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," https://pytorch.org, 2019.
[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, A. Passos *et al.*, "Scikit-learn: Machine learning in python," https://scikit-learn.org, 2011.