A Real-Time Multi-Source River Gauge Monitoring and Alert System with Predictive Analytics for Recreational Water Sports

Michael Chanslor[1]

[1]Independent Developer, Alabama, United States

michael.chanslor@gmail.com

Abstract: This paper presents a comprehensive real-time river monitoring system designed for recreational water sports enthusiasts, specifically whitewater kayakers. The system integrates multiple data sources including the United States Geological Survey (USGS) Instantaneous Values API, Tennessee Valley Authority (TVA) dam operations data, National Weather Service (NWS) Quantitative Precipitation Forecasts (QPF), Weather Underground Personal Weather Stations (PWS), and the US Drought Monitor. The architecture employs a containerized Flask-based REST API serving both a responsive web dashboard and IoT endpoints optimized for ESP32 microcontrollers. Key innovations include a predictive analytics module that forecasts river runnability based on rainfall predictions and historical response patterns, a cascade correlation visualization for multi-dam river systems, and indefinite historical data storage enabling long-term trend analysis. The system currently monitors 12 river sites across Alabama, Tennessee, North Carolina, and Georgia, processing data every 60 seconds with sub-100ms API response times. This paper details the system architecture, data flow patterns, threshold algorithms, visualization techniques, and deployment infrastructure.

Keywords: River monitoring, USGS API, TVA dam data, IoT, ESP32, Flask REST API, predictive analytics, whitewater kayaking

1 Introduction

Whitewater kayaking and other recreational water sports require precise knowledge of current river conditions to ensure both safety and optimal enjoyment. Rivers can transition from unrunnable to ideal conditions within hours following rainfall events, making real-time monitoring essential. Traditional approaches rely on manually checking government websites or physical staff gauges, which is time-consuming and often provides data without context relevant to paddlers.

This paper presents a unified river monitoring system that addresses these challenges through: (1) aggregation of multiple authoritative data sources, (2) real-time threshold-based status classification, (3) predictive analytics for anticipating runnable conditions, (4) multi-platform delivery via web dashboard and IoT API, and (5) automated email alerts for threshold transitions.

The system has been in production since 2024, serving paddlers across the Southeastern United States with continuous monitoring of rivers ranging from small creeks to dam-controlled releases.

1.1 Problem Statement

Paddlers face several challenges when determining if a river is runnable:

Data Fragmentation: River data exists across multiple agencies (USGS, TVA, private gauges) with no unified interface. Each source has different update frequencies, data formats, and access methods.

Context Deficiency: Raw gauge readings (stage height, discharge) require interpretation. A reading of "450 CFS" is meaningless without knowing what flow rate constitutes runnable conditions for that specific river.

Weather Correlation: Predicting when a river will run requires understanding the relationship between rainfall and river response, which varies significantly between watersheds.

1.2 Contributions

This work makes the following contributions:

- A unified data aggregation layer supporting USGS, TVA, StreamBeam, and proprietary gauge sources
- A threshold-based classification system with configurable multi-level status indicators
- A predictive model correlating QPF rainfall forecasts with historical river response patterns
- A cascade correlation visualization for dam-controlled river systems
- An IoT-optimized API design for resource-constrained devices
- A production-ready containerized deployment architecture

2 System Architecture

The system employs a dual-service containerized architecture running on Fly.io infrastructure.

2.1 Core Components

2.1.1 Background Worker

The primary data processing component (`usgs_multi_alert.py`) executes on a configurable interval (default 60 seconds), performing: (1) data fetching from all configured sources, (2) threshold evaluation and status classification, (3) trend calculation over configurable time windows, (4) state persistence and alert generation, and (5) output file generation (JSON, HTML).

2.1.2 Flask API Server

The API layer (`api_app.py`) provides RESTful endpoints for river data, static file serving for dashboard, CORS support for web applications, and health monitoring endpoints.

2.1.3 Supporting Modules

The system includes specialized modules:

- `qpf.py` – NWS precipitation forecasts
- `pws_observations.py` – Weather Underground PWS
- `observations.py` – NWS airport stations
- `tva_fetch.py` – TVA dam operations
- `tva_history.py` – Historical data storage
- `drought.py` – US Drought Monitor
- `predictions.py` – Predictive analytics
- `site_detail.py` – Detail page generation
- `ocoee_correlation.py` – Dam cascade visualization

2.2 Data Flow

The system processes data through: Sources → Fetch → Normalize → Evaluate → Output. Each iteration produces `gauges.json` (structured data feed), `index.html` (main dashboard), `details/*.html` (per-site detail pages), state database updates, and email alerts when thresholds are crossed.

3 Data Sources

The system integrates five primary data sources.

3.1 USGS Instantaneous Values API

The USGS maintains the most comprehensive network of stream gauges. The system accesses the Instantaneous Values service:

```
https://waterservices.usgs.gov/nwis/iv/
  ?format=json&sites=02399200
  &parameterCd=00060,00065
```

Parameters include 00065 (gage height in feet) and 00060 (discharge in CFS). Data typically updates every 15 minutes.

3.2 TVA Dam Operations

The Tennessee Valley Authority operates dams affecting paddling destinations. The system discovered undocumented REST endpoints:

```
https://www.tva.com/RestApi/
  observed-data/{SITE_CODE}.json
https://www.tva.com/RestApi/
  predicted-data/{SITE_CODE}.json
```

Data includes reservoir pool elevation, tailwater elevation, average hourly discharge, and 3-day operational forecasts.

3.3 StreamBeam Private Gauges

StreamBeam provides crowd-funded gauges on ungauged creeks. Data requires calibration:

$$h_{actual} = h_{raw} - \Delta_{offset} \quad (1)$$

where Δ_{offset} is determined through field calibration.

3.4 NWS Quantitative Precipitation Forecast

The system parses ISO 8601 intervals and apportions precipitation to local calendar days:

$$P_{day} = \sum_{i \in intervals} P_i \cdot \frac{t_{overlap}}{t_{duration}} \quad (2)$$

QPF data is cached in SQLite with 3-hour TTL.

3.5 Weather Underground PWS

Personal Weather Stations provide hyperlocal conditions. Each river has a configured fallback chain of 4 PWS stations for redundancy.

4 Threshold Classification

4.1 Standard Classification

For most rivers, a three-tier system applies:

Table 1: Standard Threshold Classification

Status	Cond.	Color	Mean.
OUT	$v < v_{min}$	Gray	Not run.
IN	$v_{min} \leq v$	Yellow	Runnable
GOOD	$v \geq v_{good}$	Green	Ideal

When both stage and discharge thresholds are configured:

$$status_{IN} = (h \geq h_{min}) \wedge (Q \geq Q_{min}) \quad (3)$$

4.2 Little River Canyon Classification

Little River Canyon requires specialized classification based on expert paddler knowledge. Six distinct levels capture the nuanced relationship:

Table 2: Little River Canyon Classification

CFS Range	Status	Color
< 250	Not runnable	Gray
$250 - 400$	Good low	Yellow
$400 - 800$	Difficult	Brown
$800 - 1500$	Good medium	Lt Green
$1500 - 2500$	Optimal	Green
> 2500	Too high	Red

4.3 Trend Calculation

The system calculates 8-hour trends:

$$\Delta = v_{current} - v_{-8h} \quad (4)$$

Classification: Rising (↑) if $\Delta > \epsilon$, Falling (↓) if $\Delta < -\epsilon$, Steady (→) otherwise.

4.4 Tailwater Trend Detection

For TVA dam sites, tailwater elevation trends indicate spillway activity:

$$\Delta_{tw} = h_{tw,current} - h_{tw,-4h} \quad (5)$$

Rising tailwater ($\Delta_{tw} > 0.5$ ft) indicates water over the dam spillway.

4.5 Level Prediction Panel

The system provides real-time predictions of when river levels will cross thresholds, enabling paddlers to plan trips based on current conditions and rate of change.

4.5.1 Rate of Change Calculation

The system uses an 8-hour window (32 USGS readings at 15-minute intervals) to calculate a stable rate of change:

$$r = \frac{h_{current} - h_{-8h}}{8 \text{ hours}} \tag{6}$$

where $h_{current}$ is the current gage height and h_{-8h} is the reading from 8 hours prior. The 8-hour window provides stability by smoothing out short-term fluctuations while remaining responsive to actual trends.

4.5.2 ETA Estimation

When the river is trending toward a threshold, the system calculates estimated time of arrival:

$$t_{ETA} = \frac{|h_{current} - h_{threshold}|}{|r|} \tag{7}$$

Two scenarios are handled:

Falling toward threshold (currently runnable, level dropping): When $h_{current} > h_{threshold}$ and $r < 0$, the system displays "ETA to Drop Below Threshold."

Rising toward threshold (currently too low, level rising): When $h_{current} < h_{threshold}$ and $r > 0$, the system displays "ETA to Reach Threshold."

4.5.3 Display Format

The ETA is formatted for readability:

- $t_{ETA} < 1$ hour: displayed as minutes
- $1 \leq t_{ETA} < 24$ hours: displayed as hours
- $t_{ETA} \geq 24$ hours: displayed as days

4.5.4 Level Prediction Panel UI

The detail page displays a 4-card prediction panel showing:

1. **Current Level**: Gage height with trend icon (↑, →, ↓)
2. **8h Change**: Magnitude of change with rate per hour
3. **Distance to Threshold**: How far above/below runnable level
4. **ETA**: Estimated time to cross threshold (or "N/A" if steady)

The panel uses color coding: green background when above threshold (runnable), yellow when approaching threshold. This provides paddlers with actionable intelligence—not just current conditions, but when conditions will change.

5 Predictive Analytics

5.1 River Response Characteristics

Each river has empirically determined response parameters:

Table 3: River Response Parameters

River	Resp.	Rain
Short Creek	12 hrs	0.65"
Town Creek	32 hrs	1.25"
Tellico River	24 hrs	1.50"
Little River	33 hrs	1.75"
Locust Fork	33 hrs	1.75"
South Sauty	33 hrs	2.00"
Mulberry Fork	33 hrs	2.25"

5.2 Likelihood Calculation

The prediction algorithm computes likelihood:

$$L = f\left(\frac{P_{forecast}}{P_{needed}}\right) \times 100\% \tag{8}$$

The function f maps rainfall ratio to likelihood with values ranging from 70%+ when $P_{forecast} \geq P_{needed}$ to <15% when forecast is below 50% of needed rainfall.

5.3 Peak Window Estimation

$$t_{peak} = t_{rain} + \tau_{response} \tag{9}$$

where t_{rain} is the centroid of forecasted rainfall and $\tau_{response}$ is the river's characteristic response time.

6 Dam Cascade Correlation

For multi-dam systems like the Ocoee, understanding cascade relationships is essential.

6.1 Ocoee River System

The Ocoee has three sequential dams:

1. **Ocoee #3 (Upper)** – OCCT1 – 1,432 ft
2. **Ocoee #2 (Middle)** – OCBT1 – 1,096 ft
3. **Ocoee #1 (Lower)** – OCAT1 – 828 ft

Water released from #3 flows to #2, then #1, with characteristic lag times.

6.2 Correlation Visualization

The cascade page provides three modes:

- **Overlapping Lines:** All dams on single axis
- **Multi-Panel Synced:** Stacked charts with synchronized x-axes
- **Full Metrics:** Discharge, pool, tailwater on dual y-axes

A dedicated endpoint aggregates all three dams:

```
GET /api/tva-history/ocoee/combined?days=7
```

7 REST API Design

7.1 Endpoint Structure

Table 4: API Endpoints

Endpoint	Purpose
`GET /`	Dashboard
`GET /api`	API docs
`GET /api/health`	Health check
`GET /api/river-levels`	All rivers
`GET /api/predictions`	Predictions
`GET /api/usgs-history/{id}`	USGS hist.
`GET /api/tva-history/{c}`	TVA hist.

7.2 ESP32 Optimization

The API includes pre-formatted display lines:

```
{
  "display_lines": [
    "Little River Canyon",
    "450 cfs ^ rising",
    "QPF Today: 0.15\"",
    "Tom:0.45\" Day3:0.00\"",
    "Temp:49F Wind:8.5 NW"
  ]
}
```

This format maps directly to 5-line OLED displays on Heltec ESP32 devices.

8 Historical Data Storage

8.1 TVA History Module

The `tva_history.py` module provides indefinite storage using SQLite with automatic deduplication via unique constraints on (site_code, timestamp).

8.2 Historical Charting

Detail pages provide Chart.js visualizations with selectable time ranges (7d, 30d, 90d, 1yr). Large datasets are downsampled to approximately 200 points:

$$n_{samples} = \min(n_{original}, 200) \qquad (10)$$

9 Alert System

9.1 State Machine

Each site maintains state: current status, last alert timestamp, last observed value, and cooldown expiration.

Rising Mode: Alert only on OUT→IN transitions (default).

Any Mode: Alert whenever thresholds are met.

9.2 Cooldown Logic

$$\text{alert\_allowed} = (t_{current} - t_{last}) > \tau_{cooldown} \qquad (11)$$

Default cooldown is 4 hours, configurable per-site.

10 Dashboard Visualization

10.1 Visual Design

Key elements include:

- **Color Coding:** Background colors indicate status
- **Sparklines:** 12-hour trend graphs with threshold lines
- **Weather Indicators:** Temperature <55°F (blue), wind >10 mph (yellow)
- **Drought Status:** D0-D4 levels with USDM colors

10.2 Detail Pages

Each site has a dedicated page with current conditions, 3-day forecast (TVA), historical chart, statistics, and external links.

11 Deployment Architecture

11.1 Container Configuration

Ubuntu 22.04 base with Python 3, Flask, and required packages. Non-root execution (UID 10001) provides security.

11.2 Persistent Storage

Fly.io volumes at `/data`:

- `state.sqlite` – Alert state
- `qpf_cache.sqlite` – QPF cache
- `drought_cache.sqlite` – Drought cache
- `tva_history.sqlite` – Historical data

12 Monitored Sites

Table 5: Monitored River Sites

River	Src	Min	Good
Mulberry	USGS	5.0 ft	10.0 ft
Locust Fork	USGS	1.70 ft	2.5 ft
Town Creek	USGS	180 cfs	250 cfs
Little River	USGS	300 cfs	500 cfs
Short Creek	SB	0.5 ft	1.0 ft
Rush South	USGS	4k cfs	8k cfs
Hiwassee	TVA	3k cfs	5k cfs
Ocoee #3	TVA	1k cfs	1.25k
Ocoee #2	TVA	1k cfs	1.25k
Ocoee #1	TVA	800 cfs	1k cfs

13 Performance Metrics

Table 6: System Performance

Metric	Value
Update interval	60 seconds
API response (p95)	< 50 ms
Dashboard load	< 500 ms
Sites per cycle	12
Cycle duration	8-12 sec
Memory usage	$\sim$150 MB

14 Future Work

Planned enhancements include: mobile application with push notifications, machine learning predictions trained on historical data, community annotations for current conditions, expanded river coverage, and water temperature integration.

15 Conclusion

This paper presented a comprehensive real-time river monitoring system for recreational water sports. The system integrates multiple data sources (USGS, TVA, StreamBeam, NWS, PWS, USDM) into a unified platform.

Key achievements:

- Sub-minute refresh with sub-100ms API response
- Predictive analytics correlating rainfall to river response
- Cascade correlation for multi-dam systems
- Production deployment serving 12 river sites
- ESP32-optimized API for IoT displays

The system has been operational since 2024, providing paddlers with actionable information previously scattered across multiple government websites.

Acknowledgments

The author thanks Adam Goshorn for Little River Canyon classifications and the StreamBeam project for enabling ungauged creek monitoring.

Availability

Production system available at:

https://docker-blue-sound-1751.fly.dev/

References

[1] U.S. Geological Survey, "USGS Water Services," https://waterservices.usgs.gov/, 2025.

[2] Tennessee Valley Authority, "TVA Lake Levels," https://www.tva.com/environment/lake-levels, 2025.

[3] National Weather Service, "Weather.gov API," 2025.

[4] National Drought Mitigation Center, "US Drought Monitor," https://droughtmonitor.unl.edu/, 2025.

[5] StreamBeam, "Community Stream Gauges," https://www.streambeam.net/, 2025.

[6] American Whitewater, "National Whitewater Inventory," https://www.americanwhitewater.org/, 2025.

[7] Pallets Projects, "Flask Web Framework," 2024.

[8] Fly.io, "Deploy App Servers Close to Your Users," 2025.

Revised: January 4, 2026