Prototype and Implementation Plan for an Early Warning Climate Disaster System Using Google BigQuery AI and Multimodal Data

---

Executive Summary

The increase in frequency and severity of climate-driven disasters, such as wildfires and floods, highlights the critical need for advanced, reliable, and real-time early warning and response systems. The development of a modern early warning system demands large-scale, multimodal data ingestion — spanning sensor networks, satellite imagery, and third-party APIs — and the ability to generate actionable, AI-powered insights with minimal latency. Google BigQuery AI, now enhanced with state-of-the-art generative AI and time series foundation models, offers a SQL-native, highly scalable platform uniquely suited for this purpose, incorporating multimodal analytics, vector search, and seamless integration with Google Earth AI satellite imagery and both structured and unstructured data sources. This solution delivers a full end-to-end blueprint for a cloud-based, event-driven early warning and disaster routing architecture, focused on real-time prediction and emergency response for wildfires and floods using the latest BigQuery AI advancements. This includes detailed research into the functional capabilities of BigQuery AI, a robust SQL-driven implementation plan with explicit code, data structures, and configuration examples, as well as documented best practices for scaling, security, and cost management.

---

1. Research Report on BigQuery AI Capabilities for Early Warning Climate Disaster Systems

1.1. The Role of AI and Big Data in Modern Climate Disaster Prediction

Recent breakthroughs in AI and big data have transformed climate disaster prediction and management. Traditional forecasting methods, often based on statistical models and historical records, struggle with timeliness, local specificity, and the integration of disparate sensor, imagery, and environmental data. AI, particularly when injected via cloud-native platforms like Google Cloud's BigQuery, allows for the fusion of structured, semi-structured, and unstructured data, providing massive scalability, flexibility, and near real-time analytics for early warning systems.

Early Warning Systems (EWS) Requirements:

- Massive, high-frequency, heterogeneous data ingestion from edge sensors and remote APIs.
- Seamless analytics for both structured tabular formats (sensor readings) and unstructured/multimodal sources (satellite images, sensor metadata).
- Predictive automation, rapid dispatch, and actionable outputs, requiring minimal manual integration or model management.

By integrating these AI capabilities directly into its data warehousing and SQL layer, BigQuery enables business and government teams to make critical decisions, such as routing resources or triggering warnings, based on AI-generated forecasts and event detection across extraordinary data scales.

1.2. Key BigQuery AI Features for Disaster Prediction

1.2.1. AI.FORECAST and TimesFM: Foundation Model Forecasting

- **AI.FORECAST** leverages Google's pre-trained TimesFM model, a transformer-based time series model trained on billions of real-world data points, optimized for zero-shot, highly accurate forecasting without developer-side model tuning.
- With a simple SQL invocation, users can forecast millions of time series in parallel for metrics like temperature, precipitation, river levels, or sensor activation, selecting horizon, identifiers, and confidence intervals directly in SQL.
- The model manages internal tuning, anomaly control, and seasonality, performing robustly even on sparse or irregular datasets — crucial for disaster-exposed, data-scarce regions.

1.2.2. Generative AI and Structured Data Extraction

- **AI.GENERATE_TABLE** and **AI.GENERATE** functions integrate LLMs directly into BigQuery, allowing row-wise or prompt-based extraction, transformation, and augmentation of structured tabular or JSON-formatted data from unstructured sources (e.g., logs, text streams, and imagery via ObjectRef/Blob metadata).
- These functions can parse dispatch alerts, sensor health logs, or third-party forecasts, and extract location tags, event types, and relevant parameters into structured, queryable BigQuery tables.

1.2.3. Multimodal Data and Vector Analytics

- BigQuery's support for **Object Tables** and the **ObjectRef** data type natively brings unstructured data, like images from Google Earth AI or field cameras, into the SQL data universe, enabling composable multimodal workflows.
- **ML.GENERATE_EMBEDDING** produces vector representations for both images and text, facilitating clustering, anomaly detection, and semantic search via **VECTOR_SEARCH**, all within SQL.
- The ability to join weather images, satellite feeds, and sensor metadata using BigQuery's AI embedding and search functions enables complex, cross-modal event prediction.

1.2.4. Direct Integration with Google Earth Engine and Satellite APIs

- **BigQuery and Earth Engine** are now tightly integrated: SQL users can query Google's vast geospatial and climate satellite data repository through functions like **ST_REGIONSTATS**, enabling zonal statistics, geospatial joins, and direct AI analytics on raster/imagery data.
- With object tables and external connections, BigQuery can ingest and process Google Earth Engine data, process real-time satellite images, and cross-reference with sensor and forecast data streams.

1.2.5. Streaming Ingestion, Pub/Sub, and Dataflow Pipelines

- Google Dataflow offers serverless, autoscaling ETL and real-time data processing, ingesting from millions of mobile edge sensors, APIs, and external queues via Pub/Sub topics, piping directly into BigQuery with schema validation and transformation for immediate AI analysis.

1.2.6. Real-Time Analytics and Automated Decision Routing

- With BigQuery Continuous Queries and event-driven architectures, real-time insights, triggers, and resource dispatch become possible, even across petabyte-scale streaming data infrastructures.

1.3. Multimodal and Unstructured Data Handling in BigQuery

ObjectRef and Multimodal Workflows:

- **ObjectRef values** now allow individual table rows to contain pointers to GCS objects (e.g., a sensor record with a field image or satellite snapshot), maintaining unified schema/metadata governance with embedded access control and pipeline compatibility.
- **Object Tables** serve as read-only SQL interfaces over buckets of imagery, sensor data, or even PDF field reports, enabling SQL analysis and joining of structured and unstructured data in a single workflow.
- **AI.GENERATE_TABLE** can reference ObjectRef/Blob columns, extracting features or insights from images, letting the analyst operate on raw sensor feeds and satellite inputs as easily as tabular data.

1.4. Advanced AI-Based Disaster Forecasting: Fires and Floods

1.4.1. Wildfire Forecasting

- AI-powered wildfire boundary and propagation modeling, using satellite imagery, field sensor heat signatures, and weather data, now leverages ML.GENERATE_EMBEDDING and image analysis functions in BigQuery, enabling boundary detection and real-time risk assessment directly from raw image data fed into object tables.
- Projects like Google's own WindTL use GCP's stack (BigQuery, Dataflow, Earth AI, Vertex AI) for predictive ember spread and fire severity mapping, with proven impact on emergency response accuracy and efficiency.

1.4.2. Flood Forecasting

- AI models, as deployed by Google Flood Hub and other initiatives, now fill critical data gaps via global hydrology and rainfall model calibration, fusing stream sensor data, satellite imagery, and third-party precipitation records.
- BigQuery-based systems can extend these approaches by fusing Earth Engine data, mobile edge sensor transmissions, and third-party flood APIs, using foundation time series AI for short- and mid-term location-specific flood probability modeling.

1.5. Security, Compliance, and Cost Management in Cloud-Scale AI Pipelines

- **Identity and Access Management (IAM):** Granular, least-privilege access at project, dataset, table, and column levels, with audit logs, role-based security, and integration with organization-wide security policy tools.
- **Encryption:** Full in-transit and at-rest encryption, with optional customer-managed keys (CMEK) for sensitive data.
- **Network Security:** VPC service controls, private IP, and integration with Cloud VPN prevent unauthorized access and data exfiltration.
- **Best Practices:** Use partitioning for large, time-series and regional data; apply clustering and indexing for fast query execution; and monitor with Cloud Logging and Cloud Monitoring for real-time pipeline health and anomaly detection.

---

2. Implementation Plan: System Architecture & Dataflow

2.1. Overall System Architecture

System Components:

1. **Mobile Edge Sensor Layer**:
    - Custom devices/geofenced sensor nodes capturing temperature, precipitation, and pressure.
    - IoT gateways streaming to Pub/Sub topics.
2. **Third-Party and Google Earth API Ingestion**:
    - Weather.gov API for U.S. real-time sensor/flood data.
    - Google Earth Engine for satellite/raster imagery by lat/long.
3. **Google Cloud Dataflow ETL Layer**:
    - Real-time and batch pipelines transform, validate, and normalize data feeds.
    - Routing to appropriate BigQuery tables: fire/flood events, raw sensor events, object tables for imagery.
4. **BigQuery Data Warehousing and AI Layer**:
    - Structured tables for sensor, API, and forecast data.
    - Object tables for GCS buckets of images (e.g., Google Earth, field cameras).
    - AI.* and ML.* functions for forecasting, extraction, embedding, and search.
5. **Routing and Emergency Response Layer**:
    - AI-generated SQL emits probable event hazards by lat/long.
    - Output triggers routing of field sensors (for more data) and emergency teams.

Targeted Locations (Examples for Prototype):

- Sonoma County, California, USA (Wildfires): lat, long: 38.29, -122.46.
- Houston, Texas, USA (Floods): lat, long: 29.76, -95.37.
- Venice, Italy (Floods): lat, long: 45.44, 12.33.

2.2. Dataflow Diagram

Below is an abstract depiction. For a code-oriented prototype, see diagram code further below.

[Dataflow Diagram]

```
         +---------------------------+
         | Mobile Edge Sensors       |
         | (Temp, Precip, Pressure) |
         +-----------+---------------+
                     |
      [Sensor data (JSON/Proto, MQTT/HTTP)]
                     |
         +-----------v------------+
         | Pub/Sub Topics         |
         | (fire, flood, sensors) |
         +-----------+------------+
                     |
         +-----------v------------+         +------------------------+
         | Dataflow ETL Pipeline  |<--------+ Earth AI/Satellite     |
         | (Streaming/Batch)      |         | & Weather APIs         |
         +-----------+------------+         +------------------------+
                     |
     +---------------+-------------------+
     |                                   |
+----v----+                       +------v-------+
| BQ Table|   (Structured         | Object Table |
| Sensors |    events, forecasts) | (Imagery GCS)|
+----+----+                       +------+-------+
     |                                   |
     +----------+  +---------------------+
                |  |
           +----v--v---+
           | BigQuery  |<-------------------------------------+
           | AI Layer  |                                      |
           | (AI.FORECAST,  AI.GENERATE, ML.GENERATE_EMBED)   |
           +----+-----+                                       |
                |                                             |
           +----v-----+                                       |
           | Route/   |<--------------------------------------+
           | Dispatch | (Decision: emergency team, sensor move)
           +----------+
```

2.3. SQL Table Structures and Example Data

2.3.1. Sensor Data Table

Field	Type	Description
id	STRING	Sensor unique ID
sensor_type	STRING	'temp', 'precip', 'pressure'
lat	FLOAT64	Latitude
lon	FLOAT64	Longitude
value	FLOAT64	Reading value
timestamp	TIMESTAMP	UTC timestamp
source	STRING	'edge', 'api', 'earth_engine'
image_ref	STRUCT	Optional: ObjectRef to GCS Earth image

Example:

id	sensor_type	lat	lon	value	timestamp	source	image_ref
sensor_ca_1	temp	38.29	-122.46	44.8	2025-08-20T13:44:00	edge	STRUCT<uri STRING, ...> (if available)
sensor_tx_1	precip	29.76	-95.37	12.7	2025-08-20T13:45:00	earth_ai	STRUCT<uri STRING, ...> (Earth satellite)
...	...	...	...	...	...	...	...

2.3.2. Disaster Event Forecast Table

Field	Type	Description
event_type	STRING	'fire', 'flood'
lat	FLOAT64	Event latitude
lon	FLOAT64	Event longitude
forecast_time	TIMESTAMP	Forecast for this time
forecast_value	FLOAT64	Probability (0-1) or intensity score
confidence	FLOAT64	Confidence of prediction
ai_status	STRING	Success/Error
action_required	BOOL	Route detection

2.3.3. Emergency Routing Output Table

Field	Type	Description
event_type	STRING	'fire', 'flood'
lat	FLOAT64	Target latitude
lon	FLOAT64	Target longitude
route_time	TIMESTAMP	Time routing is issued
dispatch_type	STRING	'emergency_team', 'sensor_mobile'
rationale	STRING	Free text, extracted via AI.GENERATE (optional)

2.3.4. Object Table for Imagery (BigQuery External Table)

Field	Type	Description
uri	STRING	GCS URI of image
ref	STRUCT	ObjectRef metadata struct
lat	FLOAT64	If available, image center lat
lon	FLOAT64	If available, image center lon
tstamp	TIMESTAMP	Associated time

---

3. SQL-Based Implementation: End-to-End Prototype

3.1. Setting Up Object Tables (Imagery/Unstructured Data)

```
CREATE OR REPLACE EXTERNAL TABLE climate_ai.earth_images
WITH CONNECTION DEFAULT
OPTIONS (
  object_metadata = 'SIMPLE',
  uris = ['gs://climate-ai-satellite-2025/*']
);
```

Join Satellite Imagery with Sensor Readings:

```
-- Enrich sensor table with nearest satellite image reference
CREATE OR REPLACE TABLE climate_ai.sensor_data_with_images AS
SELECT
  s.*,
  img.uri AS image_uri,
  img.ref AS image_ref
FROM climate_ai.sensor_data s
LEFT JOIN climate_ai.earth_images img
ON ABS(s.lat - img.lat) < 0.01 AND ABS(s.lon - img.lon) < 0.01
  AND TIMESTAMP_DIFF(s.timestamp, img.tstamp, MINUTE) BETWEEN -10 AND 10
```

3.2. Ingesting, Aggregating, and Tracking Real-time Sensor/Disaster Data

Assuming Dataflow writes directly into table `climate_ai.sensor_data`.

3.3. Forecasting Disaster Probability with BigQuery AI.FORECAST

Predicting Next 6 Hours for Fires in Sonoma County

```
SELECT * FROM AI.FORECAST(
  TABLE climate_ai.sensor_data,
  data_col => 'value',
  timestamp_col => 'timestamp',
  id_cols => ['sensor_type', 'lat', 'lon'],
  model => 'TimesFM 2.0',
  horizon => 6,
  confidence_level => 0.95
)
WHERE sensor_type = 'temp' AND lat BETWEEN 38.25 AND 38.35 AND lon BETWEEN -122.50
AND -122.40
```

- This produces forecasted time series for temperature sensors in Sonoma County, ideal for detecting fire risk thresholds.

Example Interpretation

- Use forecast_value/prediction_interval_upper_bound to flag areas with extreme heat or weather anomalies that correlate with fire or flood outbreaks.
- Output rows with high-risk locations can be joined with recent precipitation and pressure forecasts to corroborate risk for fire or flood.

3.4. Generative AI for Extracting and Structuring Disaster Reports or Sensor Alerts

Suppose we receive unstructured API logs or sensor events as blobs or JSON.

AI.GENERATE_TABLE:

```
SELECT * FROM AI.GENERATE_TABLE(
  MODEL `climate_ai.gemini_flash`,
  (SELECT alert_text as prompt FROM climate_ai.sensor_alert_logs),
  STRUCT("alert_type STRING, action_required BOOL, affected_area STRING" AS
output_schema)
)
```

- This parses unstructured logs, extracting incident types, regions, and whether action is required.

3.5. Multimodal Analysis: AI Embeddings of Satellite Imagery

Step 1: Generate Embeddings for Images

```
CREATE OR REPLACE MODEL `climate_ai.multimodal_embedding_model`
REMOTE WITH CONNECTION DEFAULT
OPTIONS (ENDPOINT = 'multimodalembedding@001');

CREATE OR REPLACE TABLE `climate_ai.earth_image_embeddings` AS
SELECT * FROM ML.GENERATE_EMBEDDING(
  MODEL `climate_ai.multimodal_embedding_model`,
  (SELECT * FROM `climate_ai.earth_images` WHERE content_type = 'image/jpeg')
);
```

Step 2: Vector Search for Fire/Flood Patterns

Suppose we have embedding for a "fire signature" or "flood pattern" text.

```
CREATE OR REPLACE TABLE `climate_ai.fire_signature_query_embedding` AS
SELECT * FROM ML.GENERATE_EMBEDDING(
  MODEL `climate_ai.multimodal_embedding_model`,
  (SELECT "visible wildfire signature: plume, heat, smoke" AS content)
);

-- Semantic search for similar images (wildfire visual cues)
CREATE OR REPLACE TABLE `climate_ai.fire_image_candidates` AS
SELECT base.uri AS gcs_uri, distance
FROM VECTOR_SEARCH(
  TABLE `climate_ai.earth_image_embeddings`,
  'ml_generate_embedding_result',
  TABLE `climate_ai.fire_signature_query_embedding`,
  'ml_generate_embedding_result',
  top_k => 5
);
```

- Similar process applies for flood image templates.

3.6. Real-Time Routing and Automated Emergency Notification

Suppose the risk/forecast is above threshold:

```
-- Flag likely fire or flood disaster
CREATE OR REPLACE TABLE climate_ai.emergency_routing AS
SELECT
  'fire' AS event_type,
  lat,
  lon,
  CURRENT_TIMESTAMP() AS route_time,
  'emergency_team' AS dispatch_type,
  CONCAT('AI forecast high fire risk. Temperature:', CAST(forecast_value AS
STRING)) AS rationale
FROM climate_ai.fire_forecast
WHERE forecast_value > 0.85
```

- Emergency operators can subscribe or poll this SQL table for live routing recommendations.

3.7. Visualizing and Monitoring Results

- Use BigQuery Geo Viz to plot forecasted event locations, average temperature anomalies, or confidence intervals across the monitored region.
- Stream results to Pub/Sub or Cloud Functions for integration with alerting dashboards and response notification systems.

---

4. Documentation: How SQL Drives Decision-Making

- **Ingestion:** All sensor and API data enters the pipeline structured for direct SQL querying, ensuring cleanliness, consistency, and AI compatibility.
- **Forecasting:** AI.FORECAST automates disaster probability prediction, enabling a SQL-based, model-free workflow that targets time and location granularity.
- **Multimodal Insight:** With OBJECTREFs and image embeddings, SQL queries can directly join alerts, imagery, and event data for anomaly detection, even across modalities.
- **Automated Routing:** Routing tables and AI-driven alerts serve as the backbone for informing mobile sensor movement and emergency team dispatch, all as structured, queryable outputs.
- **Decision Transparency:** AI-generated rationale and forecast confidence are stored and can be audited as part of each routing/dispatch decision, ensuring traceability and model explainability.

---

5. Summary: AI-Driven Early Warning Climate Disaster Systems with BigQuery

"AI-Powered Early Warning and Response for Climate Disasters: Real-Time Multimodal Analytics and Routing with Google BigQuery AI"

Abstract

As climate disasters intensify globally, real-time prediction and response are critical to saving lives and resources. This article details the design and prototype implementation of an early warning system using Google BigQuery AI, which offers SQL-native, multimodal analytics across massive, live sensor streams and satellite imagery. By integrating AI forecasting, generative extraction, object table connectors, and vector search, the system enables timely, automated routing of both sensors and emergency teams to fire and flood hot spots. The solution, focused on high-risk areas in North America and Europe, demonstrates the fusion of cloud-scale event processing, geospatial AI, and automated operations needed for the next generation of disaster resilience.

Key Technologies

- **BigQuery AI (SQL-native):** AI.FORECAST, AI.GENERATE, AI.GENERATE_TABLE, ML.GENERATE_EMBEDDING, VECTOR_SEARCH.
- **Google Earth AI/Engine:** Satellite and raster data integration.
- **Dataflow/PubSub:** Real-time sensor ingestion, streaming ETL.
- **Object Tables/ObjectRef:** Seamless handling of images and unstructured data within analytics pipelines.

Prototype Highlights

- Fused edge sensor, weather API, and cloud satellite data (using SQL object/join patterns).
- AI-generated, location-specific forecasts for wildfires and floods with TimesFM in SQL.
- Rapid semantic search over live imagery for fire/flood signature detection using vector embeddings.
- Emergency routing table as live, AI-driven SQL output ready for dashboarding and real-time integration.
- Full cost, scaling, and security compliance by design (partitioning, clustering, row-level, and column-level governance).

Sample SQL Code

```
-- AI-based time series forecasting of fire risk
SELECT * FROM AI.FORECAST(
  TABLE climate_ai.sensor_data,
  data_col => 'value',
  timestamp_col => 'timestamp',
  id_cols => ['sensor_type', 'lat', 'lon'],
  model => 'TimesFM 2.0',
  horizon => 6,
  confidence_level => 0.95
)
WHERE sensor_type = 'temp' AND lat BETWEEN 38.25 AND 38.35 AND lon BETWEEN -122.50
AND -122.40
```

Recommended Best Practices

- Use partitioned tables on timestamp/region for high-volume sensor data.
- Enforce IAM, VPC Service Controls, CMEK on all sensitive datasets.
- Monitor pipeline health and anomalies via Cloud Monitoring, and set up alert policies for high-risk forecasts.
- Apply row-level and column-level security for privacy and compliance.
- Leverage dead-letter queues and audit logs for pipeline reliability and post-mortem diagnostics.

Links and References

- BigQuery AI Reference
- BigQuery AI.FORECAST
- BigQuery Multimodal Data
- BigQuery Object Tables
- Google Earth Engine Integration
- WindTL Wildfire AI Platform
- Incident Insight AI: Wildfire Knowledge Base
- BigQuery Security Best Practices
- BigQuery API Documentation

---

6. Conclusion

The prototype and plan provided herein demonstrate a real-world-ready, fully cloud-native, SQL-driven system for early warning, disaster prediction, and response. By leveraging the latest BigQuery AI, Earth Engine, and Google Cloud Dataflow/PubSub streaming platforms, organizations can now integrate edge data with multimodal analytics—ranging from text logs and sensor feeds to live satellite imagery—and make split-second, AI-driven routing and event management decisions with full compliance, auditability, and scalable performance.

This blueprint serves as both a jump-start for academic research and a practical implementation guide for public safety agencies, environmental monitoring companies, and governments seeking to build resilient, adaptive, and cost-efficient climate disaster response systems fit for the realities of the new climate era that started within last years.