Real-Time Earthquake Monitoring and Prediction System

Mikołaj Malec 298828

High-Level Description

The core business process supports real-time situational awareness, data-driven decision-making, and predictive insights for earthquake monitoring teams.
Users can monitor live seismic activity, apply filters, generate reports, and interpret AI-driven forecasts to support early warnings and operational response.

The system offers a web-based dashboard with two primary modules:
- Analytical Module: Displays real-time and historical earthquake data on an interactive map and data table, with filters for time, magnitude, and location.
- Predictive Module: Visualises machine learning–based predictions of possible future tremors, showing forecasted locations and estimated magnitudes on the same map.

Business Process Diagram (Core Use Case)

Primary Actors:
- Analyst
- Seismologist
- Emergency Operations Officer

Use Case 1: Data Exploration and Trend Analysis
User: Analyst
Goal: Identify temporal and regional earthquake patterns.
Process Flow:
1. The analyst opens the Dashboard (Analytical View).
2. They apply filters, e.g., Time Range: past 5 years, Magnitude: > 4.5.
3. The map and data table update dynamically to display filtered earthquake events.
4. The analyst reviews visualizations such as heatmaps, frequency plots, and magnitude distributions.
5. They export selected charts and insights into a presentation or report.

Outcome: The analyst obtains clear, data-driven visualizations of seismic trends for reporting and analysis.

Use Case 2: Monitoring Active Seismic Zone Real-time
User: Seismologist
Goal: Track live seismic activity for scientific observation and validation.
Process Flow:
1. The seismologist opens the Dashboard (Analytical View).
2. They apply filters to focus on specific active tectonic zones (e.g., the Pacific).
3. The map updates to display current seismic events, color-coded by magnitude and depth.
4. The seismologist inspects individual events for parameters such as depth, magnitude, and fault region.
5. The seismologist reviews visualizations such as heatmaps, frequency plots, and magnitude distributions updated for that region.
6. They note patterns or anomalies that may indicate ongoing geological shifts.

Outcome: The seismologist gains live visibility into active seismic zones, supporting scientific validation and research.

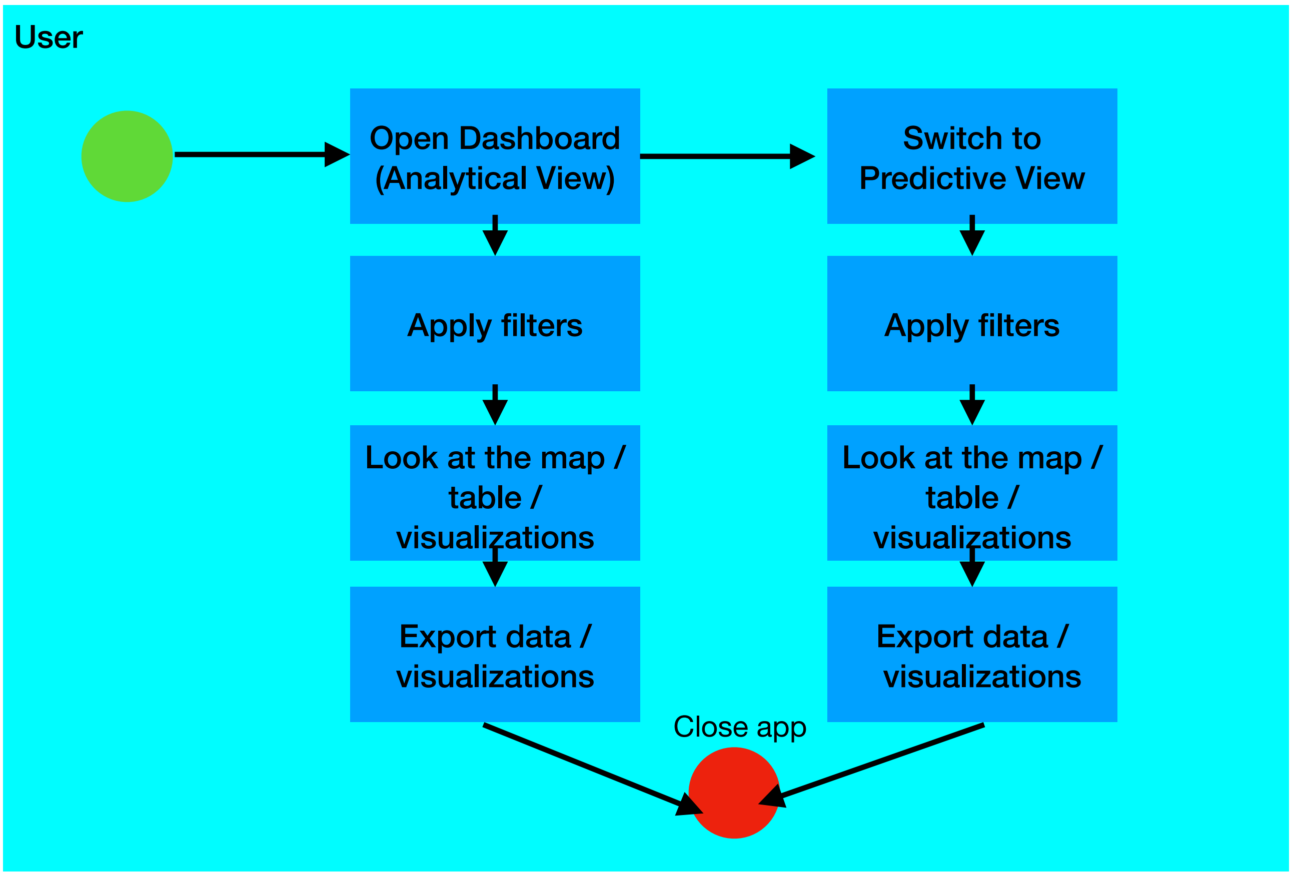
Use Case 3: Real-Time Incident Monitoring
User: Emergency Operations Officer
Goal: Monitor significant ongoing earthquakes and identify potential escalation.
Process Flow:
1. The Emergency Officer opens the Dashboard (Analytical View).
2. They filter the view to show recent activity, e.g., the past 24 hours and magnitudes above 5.0.
3. The map displays active seismic regions such as the Pacific.
4. The officer reviews key details, depth, magnitude, location, and fault line proximity, to assess the severity.
5. If a large or clustered event is detected, the officer notifies supervisors or response teams.

Outcome: Early identification of unusual seismic clusters that may signal larger upcoming events, enabling faster response coordination.

Use Case 4: Checking AI-Based Earthquake Predictions
User: Emergency Operations Officer
Goal: Utilize AI-driven forecasts to anticipate possible earthquake occurrences and enhance preparedness.
Process Flow:
1. The Emergency Officer opens the Dashboard (Analytical View)
2. The Emergency Officer switches to the Predictive View on the dashboard.
3. The system displays predicted earthquake hotspots on the same map, marked with expected magnitude ranges and confidence levels.
4. The officer filters the predictions by region or time window (e.g., next 12 hours).
5. They review high-risk zones.
6. If elevated risk is detected, the officer initiates preventive alerts or readiness protocols within the emergency response unit.

Outcome: The officer proactively plans resource allocation and readiness measures based on predictive insights, improving disaster preparedness.

Business process diagram



The results of a preliminary investigation of the sources and datasets to be utilised in the project

Source	API Name	Format / Structure	Key Attributes	Base Data Volume	Velocity (Update Rate)
TerraQuake API	api.terraquakeapi.com	JSON (REST API)	event_id, timestamp, latitude, longitude, depth_km, magnitude, region, source_type, alert_level	~15,000–25,000 events per day globally (~3–6 MB/day in JSON format)	Updated every 2–10 seconds (depending on regional seismic network reporting latency)
USGS Earthquake Hazards Program API	earthquake.usgs.gov/fdsnws/event/1/	GeoJSON	id, time, latitude, longitude, depth, mag, place, type, status, tz	~0.5–2 MB/ hour (thousands of events daily)	Updated every few seconds to minutes, depending on the detection and verification cycle

- Data Type: Both sources provide semi-structured JSON/GeoJSON data that can be parsed easily into NoSQL formats.
- Granularity: Each record represents an individual earthquake or seismic event, uniquely identified by an event ID and timestamp.
- Geospatial Nature: Both datasets contain latitude, longitude, and depth attributes, enabling spatial analytics and visualization on maps.
- Quality: Data from both APIs is near real-time but may include minor reporting delays (a few seconds to minutes).

Key assumptions regarding data processing steps

Real-Time and Batch Data Integration

The system processes both streaming (real-time) and historical (batch) earthquake data to ensure comprehensive situational awareness and long-term trend analysis.

Data Ingestion and Preprocessing:

Incoming seismic event data, received in JSON/GeoJSON format from external APIs, will be ingested with fault-tolerant delivery. Preprocessing includes validation, normalization of attribute fields (magnitude, depth, coordinates, timestamps), and schema alignment to ensure data consistency across the pipeline.

Low-Latency Event Handling:

The system provides near real-time processing of streaming data, ensuring that recent events are quickly reflected in the dashboard visualizations.

Historical Data Storage and Model Training:

The system manages long-term data storage and supports large-scale transformations and machine learning workflows for predictive modeling.

Scalability and Fault Tolerance:
The system architecture is designed to scale horizontally to handle varying data loads while maintaining robustness against data loss or processing failures.

An outline of the proposed solution infrastructure

The proposed system follows a Lambda Architecture, integrating streaming and batch data processing components into a cohesive data pipeline.

Input Layer (Preprocessing)
Technologies: Apache NiFi Streams, Apache Kafka
Function: Collects real-time seismic data from APIs, performs initial validation and enrichment, and forwards it to processing layers via Kafka topics.
Output: Cleaned and standardized data ready for further processing.

Speed Layer
Technologies: Apache Spark Structured Streaming, Apache Cassandra
Function: Performs real-time computations on streaming data (e.g., detecting recent seismic activity, calculating moving averages, and updating dashboards).
Output: Low-latency updates stored in Cassandra for immediate dashboard visualization.

Batch Layer
Technologies: Apache Spark, PySpark, HDFS, Hive
Function: Handles large-scale transformations, aggregation of historical data, and machine learning model training for predictive analytics.
Output: Processed datasets and prediction models stored in Hive for long-term access.

Serving Layer
Technologies: Hive / Cassandra (depending on aggregation scale)
Function: Integrates data from both Speed and Batch Layers to provide a unified, queryable interface for the presentation layer.
Output: Aggregated datasets accessible via APIs or SQL queries.

Presentation Layer
Technology: Web-based Dashboard / Visualization Tool
Function: Displays real-time and historical earthquake data, predictive insights, and visual analytics through an interactive user interface.
Output: Actionable insights for analysts, seismologists, and emergency officers.