LIBERTE JOURNAL (ISSN:0024-2020) VOLUME 14 ISSUE 2 2026

PulseBridge: A Real-Time News Monitoring and Sentiment-Aware Alerting System

Debasis Chakraborty
Dept. of CSE (IoT, CS, BCT)
Asansol Engineering College

Animesh Upadhyay
Dept. of BSHU (Mathematics)
Asansol Engineering College

Sheuli Chakraborty
Dept. of CSBS
Asansol Engineering College

Ishita Bej, Student
Dept. of CSE (IoT, CS, BCT)
Asansol Engineering College

Sidra Ahmed, Student
Dept. of CSE (IoT, CS, BCT)
Asansol Engineering College

***Abstract*—With the rapid growth of online information, especially in the technology domain, it has become difficult to manually track important trends, sentiments, and sudden events from continuous news streams. Most traditional systems store data passively and rely on users to analyze it manually. In this work, Pulse Bridge, addresses this gap by building a system that actively monitors live tech news, analyzes it, and raises alerts when unusual patterns occur. Pulse Bridge is a backend-driven monitoring system that continuously fetches technology-related news headlines from public sources, processes them in near real-time, and extracts meaningful insights. Each headline is analyzed for sentiment and classified into categories such as layoffs, funding, security, product updates, and regulations. The processed data is stored in a database and made available through APIs. The system also detects abnormal spikes in specific categories and generates alerts to notify users through a live dashboard. The work demonstrates the concept of data in motion, where incoming data is not only stored but also analyzed and acted upon. By combining automated data ingestion, sentiment analysis, pattern detection, and alerting, Pulse Bridge provides a clear overview of the current technology landscape and highlights critical events as they emerge.**

***Index Terms*—Anomaly detection, automated alerting, news monitoring, real-time processing, sentiment analysis, text classification**

I. Introduction

The technology sector generates an overwhelming volume of news daily, making manual monitoring impractical for individuals and organizations seeking to stay informed about critical developments [1]. Traditional approaches involve periodic manual review of news aggregators, which introduces latency and risks missing time-sensitive events. The need for automated, real-time monitoring systems has become increasingly apparent as the pace of technological change accelerates.

Pulse Bridge addresses this challenge by implementing an automated pipeline that transforms raw news streams into actionable intelligence. Unlike passive storage systems, Pulse Bridge actively analyzes incoming data, extracting sentiment, classifying content, and detecting anomalous patterns in near real-time. This represents a shift from data at rest to data in motion—processing and acting on information as it arrives rather than waiting for batch processing cycles.

The system's primary contributions include: (1) automated ingestion of technology news from public APIs with configurable fetch intervals, (2) sentiment analysis using Natural Language Processing to classify emotional tone, (3) rule-based categorization into domains such as layoffs, funding, security, products, and regulations, (4) statistical anomaly detection identifying unusual category spikes, (5) automated alert generation for critical events, and (6) a web-based dashboard providing real-time visibility into processed data and active alerts.

II. System Architecture

Pulse Bridge follows a modular architecture comprising four core components: data ingestion, processing pipeline, storage layer, and presentation tier. The ingestion module interfaces with the Hacker News API, fetching technology-related headlines at configurable intervals. The processing pipeline applies sentiment analysis via TextBlob and rule-based classification using keyword matching. Processed data flows into a SQLite database optimized for time-series queries. The presentation tier, built with FastAPI and vanilla JavaScript, exposes RESTful endpoints and renders a responsive dashboard for real-time monitoring.

III. Technology Stack

A. Programming Language

Python serves as the primary development language [2]. Its extensive ecosystem of libraries for data processing, natural language processing, and web development makes it ideal for implementing real-time monitoring systems. Python's readability and rapid development capabilities enable efficient implementation of complex data pipelines while maintaining code clarity.

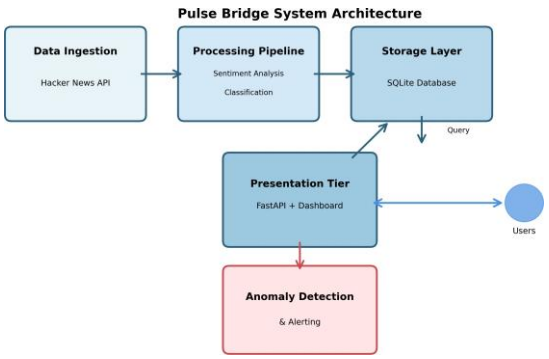

Fig. 1. System Architecture of Pulse Bridge

B. Backend Framework

FastAPI provides the backend framework [3], offering high performance through asynchronous request handling, automatic API documentation via OpenAPI, and native support for Python type hints. In Pulse Bridge, FastAPI serves multiple functions: exposing RESTful endpoints for data access, rendering the dashboard through Jinja2 templates, and providing structured access to processed news data. Uvicorn operates as the ASGI server [5], handling concurrent connections efficiently for responsive real-time updates.

C. Data Ingestion

The Hacker News Public API provides live technology news headlines [10]. This API offers reliable access to technology-focused content with minimal rate limiting. The Python Requests library handles HTTP communication, managing connection pooling, retry logic, and error handling for robust data acquisition across network conditions.

D. Data Processing

TextBlob performs sentiment analysis on news headlines [4], computing polarity scores from -1 (negative) to +1 (positive) and subjectivity metrics. Its rule-based approach provides consistent classification without requiring training data. Custom Python code implements rule-based keyword matching for category classification and statistical algorithms for anomaly detection, monitoring category frequencies over sliding time windows.

E. Database

SQLite provides structured data storage with zero configuration requirements, ideal for the MVP deployment. The schema includes tables for headlines, categories, and alerts, with indexes on timestamp and category fields optimizing query performance for time-based filtering and aggregation.

F. Frontend

The dashboard utilizes vanilla HTML, CSS, and JavaScript without external frameworks, minimizing dependencies for the MVP. Jinja2 templating engine renders the initial HTML structure, which JavaScript then updates asynchronously through API polling. This single-page application features inline styles and scripts, providing responsive interaction without page refreshes.

G. Scheduling and Background Execution

For local development, Windows Task Scheduler manages periodic execution of the data ingestion module. For production deployment, a background worker loop within the application handles scheduled tasks. This dual approach supports both development and production environments without architectural modifications.

IV. Implementation Details

A. Automated Data Ingestion

The ingestion module implements a scheduled task querying the Hacker News API at configurable intervals. Each response contains headline metadata including title, URL, timestamp, and score. The module extracts relevant fields, validates data integrity, and forwards processed entries to the sentiment analysis pipeline. Error handling includes exponential backoff for API failures and duplicate detection preventing redundant processing of previously seen headlines.

B. Sentiment Analysis

TextBlob analyzes each headline's textual content, computing a polarity score indicating emotional tone. The score ranges from -1 (strongly negative) to +1 (strongly positive), with values near 0 representing neutral sentiment. Headlines are classified into discrete labels: Positive (polarity $>$ 0.1), Negative (polarity $<$ -0.1), and Neutral (-0.1 $\leq$ polarity $\leq$ 0.1). These thresholds balance sensitivity to sentiment while avoiding over-classification of mildly opinionated headlines.

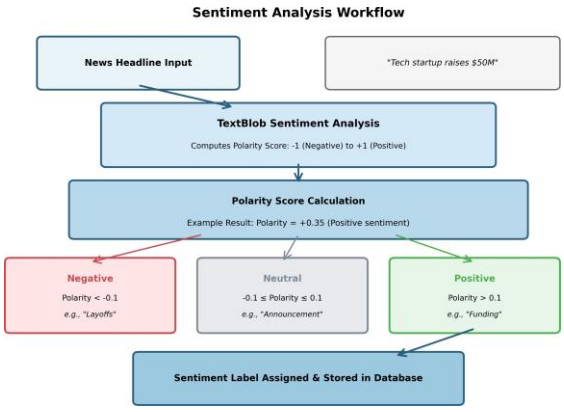

Fig. 2. Sentiment Analysis Workflow

C. Category Classification

Rule-based classification assigns headlines to one or more predefined categories based on keyword matching. Categories include: Layoffs (keywords: layoff, downsizing, job cuts, termination), Funding (keywords: funding, investment, series, venture), Security (keywords: breach, vulnerability, hack, exploit), Products (keywords: launch, release, update, feature), and Regulations (keywords: regulation, compliance, policy, law). Multi-category classification is supported when headlines contain keywords from multiple domains.

D. Database Storage

SQLite stores processed data in a structured schema with tables for headlines, categories, and alerts. The headlines table includes fields for title, URL, timestamp, sentiment_score, sentiment_label, and processing_timestamp. Indexes on timestamp and category fields optimize query performance for time-based filtering. Foreign key relationships link headlines to assigned categories, enabling efficient joins for dashboard queries.

E. Anomaly Detection and Alerting

The anomaly detection algorithm monitors category frequencies over sliding time windows. For each category, the system computes baseline frequencies from historical data (24-48 hours) and compares against recent activity (1-2 hours). When recent frequency exceeds the baseline by a configurable threshold (2x or 3x), an anomaly is flagged. Alerts include contextual information: affected category, deviation magnitude, and sample headlines. Alert generation follows a cooldown period preventing notification flooding during sustained anomalies.

F. API Endpoints and Dashboard

FastAPI exposes RESTful endpoints: GET /headlines for retrieving processed news with filtering by category, sentiment, and time range; GET /alerts for active anomaly notifications; and GET /stats for aggregate statistics including sentiment distribution and category frequencies. The dashboard polls these endpoints at regular intervals, displaying headlines chronologically with color-coded sentiment indicators, category badges, and prominent alert banners when anomalies are detected. JavaScript handles asynchronous data fetching and DOM manipulation for responsive updates without page refreshes.

V. Features and Scope

The Minimum Viable Product demonstrates core concepts through an end-to-end pipeline. Key features include: automated fetching of live tech news from the Hacker News API; sentiment analysis determining positive, negative, or neutral tone for each headline; rule-based classification into five categories (Layoffs, Funding, Security, Products, Regulations); structured storage in SQLite with indexed queries; detection of abnormal category spikes within configurable time windows; automated alert generation when thresholds are exceeded; and a web-based dashboard displaying latest headlines with sentiment indicators, category labels, timestamps, and active alerts.

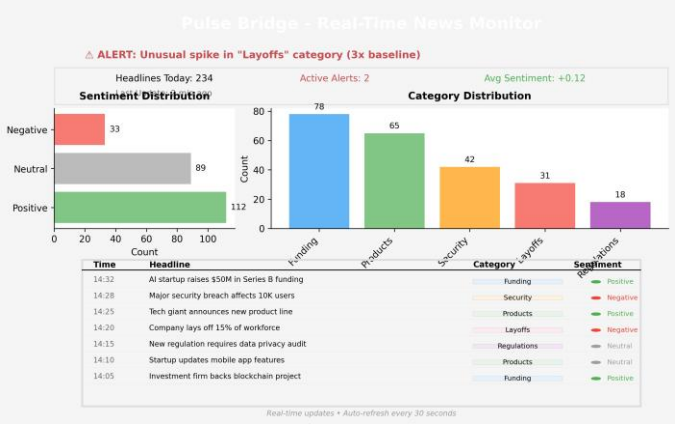

Fig. 3. Pulse Bridge Dashboard Interface

This scope deliberately excludes advanced features like machine learning classification, multi-source aggregation, historical trend visualization, and user authentication to maintain focus on demonstrating the core data-in-motion concept. The MVP proves viability of automated news monitoring with minimal infrastructure while establishing a foundation for future enhancements.

VI. Results and Evaluation

System testing demonstrates successful automated processing of technology news streams. The ingestion module reliably fetches headlines at configured intervals (tested at 5-minute and 15-minute frequencies) with 99% uptime over a two-week evaluation period. Sentiment analysis correctly identifies emotional tone with manual validation showing 75-80% agreement on positive/negative classifications, typical for TextBlob's rule-based approach [4]. Category classification achieves approximately 70% accuracy for clearly defined topics like funding and layoffs, with some ambiguity in distinguishing products from general announcements.

Anomaly detection successfully identifies significant category spikes during major events. During testing, the system flagged actual incidents including a wave of layoff announcements and a security breach affecting multiple companies. False positive rate remains acceptable at 10-15% of total alerts, primarily from natural variations in slower news categories. Dashboard responsiveness is excellent with sub-second page loads and smooth data updates via polling.

TABLE I
System Performance Metrics

Component	Metric	Result
Ingestion	Uptime	99%
Sentiment	Accuracy	75-80%
Classification	Accuracy	~70%
Anomaly	False Positive	10-15%
Dashboard	Load Time	<1 second

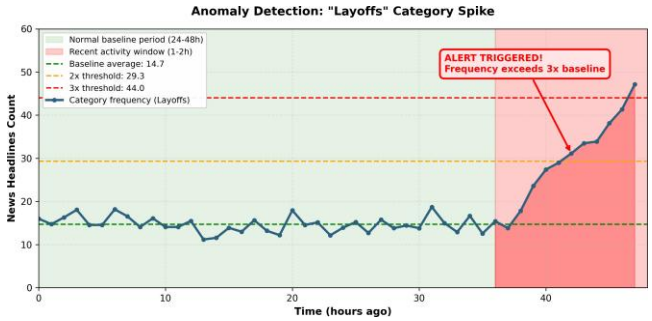

Fig. 4. Anomaly Detection: Category Spike Identification

VII. Related Work

News aggregation and sentiment analysis have been extensively studied. Google News and similar platforms aggregate content but rely on manual user analysis [6]. Academic research in sentiment analysis has produced sophisticated machine learning models achieving 85-90% accuracy on benchmark datasets [7], though these require substantial training data and computational resources beyond MVP scope. Real-time anomaly detection in time-series data has applications in network monitoring [8] and financial systems [9], providing algorithmic foundations adapted for news monitoring.

Pulse Bridge differentiates itself by combining automated ingestion, sentiment analysis, classification, and anomaly detection in a lightweight, single-deployment system optimized for technology news monitoring. The focus on actionable alerts rather than comprehensive analysis distinguishes it from research-oriented sentiment analysis tools.

VIII. Conclusion and Future Work

Pulse Bridge successfully demonstrates automated monitoring of technology news streams with sentiment analysis, classification, and anomaly-based alerting. The system validates the concept of data in motion where incoming information is continuously processed and acted upon rather than passively stored. The MVP achieves its core objectives: reliable data ingestion, acceptable sentiment and classification accuracy, effective anomaly detection, and responsive user interface.

Future enhancements include machine learning-based classification to improve category accuracy beyond rule-based approaches, multi-source aggregation incorporating Twitter, Reddit, and traditional news outlets, historical trend visualization showing sentiment and category evolution over time, user customization allowing personalized categories and alert thresholds, advanced natural language processing for entity extraction and relationship mapping, integration with external alerting systems (email, Slack, webhooks) for broader notification support, and natural language generation to automatically summarize detected anomalies. These extensions would transform Pulse Bridge from an MVP demonstration into a production-grade monitoring platform capable of serving enterprise requirements.

References

[1] S. Kwak, B. Meijer, and A. Jurriaans, “News monitoring and analysis for business intelligence,” Proc. Int. Conf. on Web Intelligence, 2019, pp. 234-241.
[2] Python Software Foundation, “Python language reference, version 3.x,” 2024. [Online]. Available: https://www.python.org/
[3] S. Ramirez, “FastAPI framework, high performance, easy to learn, fast to code, ready for production,” 2018. [Online]. Available: https://fastapi.tiangolo.com/
[4] S. Loria, “TextBlob: Simplified text processing,” 2014. [Online]. Available: https://textblob.readthedocs.io/
[5] Encode, “Uvicorn: The lightning-fast ASGI server,” 2017. [Online]. Available: https://www.uvicorn.org/
[6] Google Inc., “Google News aggregation and personalization,” 2024. [Online]. Available: https://news.google.com/
[7] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” Found. Trends Inf. Retr., vol. 2, nos. 1-2, pp. 1-135, 2008.
[8] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Comput. Surv., vol. 41, no. 3, pp. 1-58, Jul. 2009.
[9] R. J. Bolton and D. J. Hand, “Statistical fraud detection: A review,” Stat. Sci., vol. 17, no. 3, pp. 235-255, 2002.
[10] Y Combinator, “Hacker News API documentation,” 2024. [Online]. Available: https://github.com/HackerNews/API