Real-Time Data Analysis

Designing and analyzing real-time data systems is one of the essential components of Computer Science and Engineering

Only by actively participating in data-driven activities can students truly understand the challenges and immense possibilities of modern data processing. This activity will introduce first-year students to the core aspects of real-time data engineering and real-world applications involving streaming data, sensor-based systems, live dashboards, and time-critical decision-making.

In this learning activity, students will design and implement a real-time data analysis pipeline using the provided tools and datasets. Students will interact with data streams, observe system behaviors, and understand key data-processing components such as ingestion, filtering, transformation, and visualization.
The group will comprise a maximum of **5 members**.

Faculty Facilitator
Dr. Jinee Goyal
Dr. Vijay Kumari
Dr. Ashish Bajaj

The basic outline of the activity is:

- To learn fundamental concepts of real-time data, data streams, event processing, latency handling, and the difference between batch and stream processing.
- To identify noise, missing values, and anomalies within real-time data and apply filtering or smoothing techniques to improve quality.
- To build a real-time data pipeline using the provided sensors, datasets, or simulation tools (e.g., Kafka producer, Python streams, IoT data). Students must ensure efficient processing; unnecessary components will increase latency and reduce performance.
- To develop and test a real-time dashboard for monitoring, visualization, and decision-making based on live data.

After going through this activity, the students would be able to:

- Understand major components of a real-time data analysis system—including data ingestion, processing engines, and visualization layers.
- Identify different types of data streams (sensor data, log streams, event data, transactional streams).
- Apply essential data-engineering concepts: latency, throughput, event-time vs processing-time, windowing, anomaly detection.
- Recognize the contribution of each pipeline component to the overall efficiency, accuracy, and stability of the system.
- Understand how data quality, system architecture, and flow management affect real-time performance.

Applications can be…

- **Develop a Real-Time Traffic Monitoring System** to analyze live traffic flow, detect congestion, and suggest alternative routes.
- **Design a Real-Time Health Monitoring System** that tracks patient vitals (heart rate, SpO_2, ECG signals) and triggers alerts during abnormal patterns.
- **Build a Real-Time Fraud Detection System** for banking transactions using streaming analytics to detect suspicious activities instantly.
- **Develop a Live Environmental Monitoring System** that reads sensor data (air quality, temperature, humidity) and updates dashboards continuously.
- **Implement a Real-Time Industrial Safety System** to detect sudden changes such as temperature rise, pressure drop, equipment failure, or gas leakage.
- **Create a Smart Retail Analytics System** to monitor customer movement, heat maps, and product engagement in stores through live data feeds.
- **Design a Real-Time Social Media Sentiment Analyzer** that processes ongoing posts/tweets to identify trends, opinions, or emergencies.
- **Build a Live Stock Market Data Analysis System** that processes millisecond-level price updates for trading decisions.
- **Develop a Real-Time Anomaly Detection System** to identify unusual behavior in networks, servers, or IoT systems.
- **Create a Predictive Maintenance System** that continuously monitors machine data and predicts failures before they happen.

These are indicative activities only; you are free to explore to go to the next level.

Assessment Module 1: Exploration & Analysis of Real-Time Data Modalities

Objective:

To help students identify their preferred data modality—*Textual, Image, Audio, or EEG/ECG Signals*—and develop foundational skills in data exploration, preprocessing, and basic analysis based on the chosen dataset.

The student must perform the following tasks:

1. Identify Your Preferred Data Modality
Choose *one* modality based on your interest:

- **Textual Data** (news, chats, logs, documents, social media)
- **Image Data** (photos, medical images, CCTV frames, satellite images)
- **Audio Data** (speech, music, environmental sounds)
- **EEG/ECG Physiological Signals** (health monitoring, brain & heart signals)

2. Search & Select a Real Dataset
The student must explore and choose **one relevant real-world dataset** from sources such as:

- Kaggle
- UCI Machine Learning Repository
- PhysioNet (for EEG/ECG)
- HuggingFace Datasets
- Government open data portals

3. Describe the Dataset Clearly
Students must document:

- **Dataset name & source link**
- **Type of data (text/image/audio/EEG/ECG)**
- **Number of samples/records**
- **Key features or attributes**
- **Problem domain** (healthcare, finance, social media, security, etc.)

4. Perform Basic Exploratory Data Analysis (EDA)
Depending on the modality, students should analyze:
For Textual Data:

- Word frequency extraction
- Stopword identification
- Sample sentence analysis
- Text length distribution

For Image Data:

- Image dimensions, color channels
- Sample visualizations
- Pixel intensity distribution
- Identification of noise or distortions

For Audio Data:

- Waveform visualization
- Spectrogram generation
- Duration statistics
- Peak frequencies

For EEG/ECG Data:

- Signal plotting
- Identification of peaks, rhythms
- Noise or artifact detection
- Basic statistical analysis (mean, variance, RMS)

Assessment Module 2: Preprocessing Data for All Modalities

Objective:

To understand and apply essential preprocessing techniques for four major data modalities—**Text, Image, Audio, and EEG/ECG signals**—to make the data ready for analysis or modeling.

Tasks to be Completed

1. Preprocess Textual Data

Perform **any two** of the following:

- Removal of punctuation, stopwords
- Lowercasing, cleaning unwanted characters
- Tokenization and lemmatization
- Converting text into numerical form (Bag-of-Words / TF-IDF)

2. Preprocess Image Data

Perform **any two** of the following:

- Resizing and normalization of pixel values
- Converting to grayscale
- Removing noise using filters (Gaussian/Median)
- Image augmentation (rotation, flip, brightness adjustments)

3. Preprocess Audio Data

Perform **any two** of the following:

- Noise reduction using filters
- Trimming/slicing audio clips
- Extracting MFCCs or spectrograms
- Converting stereo to mono & resampling

4. Preprocess EEG/ECG Signal Data

Perform **any two** of the following:

- Filtering noise (bandpass or notch filters)
- Normalizing raw signal values
- Smoothing using moving averages
- Segmenting the signal into fixed-length windows

Assessment Module 3: Feature Extraction / Model Building

Objective:

To understand the next step after preprocessing by applying **feature extraction (ML)** or **model building (DL)** on the chosen data modality.

Tasks to be Completed

1. Choose One Approach

Select **either**:

- **Machine Learning Approach (Feature Extraction)**
  OR
- **Deep Learning Approach (Model Building)**

A. If You Choose Machine Learning (ML)

Perform the following:

1. Extract **one meaningful feature set** from your dataset:
    - Text: TF-IDF / Bag-of-Words
    - Image: HOG / Color Histogram
    - Audio: MFCC / Spectral Features
    - ECG/EEG: Peak detection / statistical features
2. Apply a simple ML model (any one):
    - Logistic Regression
    - SVM
    - Decision Tree
3. Submit a short note (6–8 lines) describing:
    - What features you extracted
    - Why these features help in learning
    - Result/accuracy of the basic model

B. If You Choose Deep Learning (DL)

Perform the following:

1. Select a suitable DL architecture:
    - CNN for Images
    - LSTM/Transformer for Text
    - CNN + Spectrogram for Audio

Assessment Module 4: Model Evaluation & Performance Analysis

Objective:

To evaluate the trained Machine Learning or Deep Learning model using standard metrics and analyze its performance on unseen data.

Tasks to be Completed

1. Evaluate Your Model Using Appropriate Metrics

Choose metrics based on your task:

For Classification Tasks:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

For Regression Tasks:

- MAE (Mean Absolute Error)
- MSE / RMSE
- R^2 Score

For Signal / Audio Models:

- Signal-to-Noise Ratio (SNR)
- Error rate
- Detection accuracy

2. Compare Training vs Validation Performance

- Plot **training vs validation accuracy**
- OR **training vs validation loss**
- Identify if the model is **overfitting**, **underfitting**, or **well-balanced**

3. Test on Unseen Data

- Run the model on a **small unseen test set**
- Report how well the model generalizes
- Note any **misclassifications or errors**

Assessment Module 5: Deployment & Real-Time Implementation

Objective:

To understand how a trained model can be deployed and used in real-time or near real-time environments for practical applications.

Tasks to be Completed

1. Choose a Deployment Method

Select **one** method based on your project:

- Deploy as a **web app** (Streamlit / Flask / FastAPI)
- Deploy as a **mobile-friendly system** (simple UI)
- Deploy as a **real-time dashboard** (live graphs, alerts)
- Integrate into a **real-time pipeline** (Kafka / MQTT / continuous input)

2. Simulate Real-Time Data Input

Use any one approach:

- Live data stream (sensor / API)
- User input text/image/audio uploaded through UI
- Video/camera feed
- Streaming physiological signals (ECG/EEG sample streams)

Explain how the model receives and processes the data.

3. Run Your Model in Real-Time

Demonstrate the model working live:

- Predict sentiment for live text
- Detect object in camera frame
- Classify audio in real-time
- Detect heartbeat anomalies from streaming ECG

Record the model's **latency**, **speed**, and **response quality**.

4. Evaluate the Deployment

Write a short analysis (6–8 lines):

- How fast does the system respond?
- Any lag or delay?
- Are predictions stable in real-time?
- What challenges did you face during deployment?

Assessment Module 6: "AI Exploration Experiment" – Try Something Interesting with Your Model

Objective:

To allow students to explore and understand the behavior of their trained model by performing simple, engaging, and insightful experiments.

1. **Test the Model with Unexpected Inputs**

Provide the model with inputs it was not specifically trained on.
Examples:

- Give an image classifier an unrelated object
- Speak in a different tone/accent
- Provide unusual or tricky sentences
- Feed slightly altered ECG/EEG patterns

Students will analyze:

- How the model responds
- Whether predictions change
- Why this behavior might occur

2. **Provide Real-World Inputs from Your Environment**

Use **your own real data** to test the model:

- A photo captured using your phone
- Your own recorded voice
- A sentence you write
- Simple smartwatch signal data (optional)

Students will note:

- How well the model generalizes
- Accuracy differences

3. **Modify One Condition and Observe Changes**

Introduce a small change:

- Rotate or blur an image
- Add low-level noise to audio
- Add emojis/slang to text
- Add minor noise to signals

Students will explain:

- How predictions changed
- Why the model behaved differently